BRAMBLES PRICE MANAGEMENT SYSTEM

Project report submitted in partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

By

Simran Chauhan

UNDER THE GUIDANCE OF

Ms. Murtaza Hussain

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

May 2019

TABLE OF CONTENTS

DECLARATION

I hereby declare that the work reported in the B-Tech thesis entitled **"BRAMBLES PRICE MANAGEMENT SYSTEM"** submitted at **Jaypee University of Information Technology, Waknaghat India,** is an authentic record of my work carried out under the supervision of **Ms.Murtaza Hussain.** I have not submitted this work elsewhere for any other degree or diploma.

Name **Signature**

Simran Chauhan

Department of Electronics and Communication

Jaypee University of Information Technology, Waknaghat, India

Date: 23/5/2019

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to **GENPACT** that gave me an opportunity to undergo this internship program. I am also thankful to all those who supported me throughout the entire course of project and without whose guidance it would have been quite a difficult task.

I am highly thankful to **Ms.Murtaza Hussain, Senior Manager and Ms.Ramesh Bobbili, Project Manager** for their valuable supervision in this project, whose constant support and motivation encouraged me to do the project efficiently.

Date: 23/05/2019

Simran Chauhan(151105)

ABSTRACT

Brambles Price Management System is a project that is based on finding the cost of per pallet and provides services to customers in several industrial and retail supply chains. This web application aims as simplifying the complex pricing structure for the customers. This report revolves around the Brambles pricing system and talks about the concept of Quality Assurance that is an important part of the Software Testing Life Cycle. In this project I have learned various technologies and framework that help in automating the web applications and their results have been presents in this report.

CHAPTER 1

BRAMBLES PRICE MANAGEMENT SYSTEM

Brambles Limited is an Australian company that specializes in providing various pooling services, mainly focused on outsourced management of pallets. A pallet is a piece of equipment that helps in handling of goods for lifting purposes and transportation. Brambles Price Management System is all about managing the price of the pallets.

SAP AND SIEBEL are the two important parts of the project. SAP stands for System Application and Products. It is ERP (Enterprise Resource Planning). There are different interactions that take place between different departments. Let us consider a customer approaches the sales department for a product. Sales department approaches the inventory if the product is out of stock. If the product is out of stock, then production is reached to manufacture the product. The production checks if the raw materials that are required to manufacture the product are available in the inventory. All this process takes a lot of time, high costs are involved and maybe the chance of losing out the customer.

Now SAP makes use of ERP, here all the updation are done in the central system and when the customer approaches sales, the sales has access to products in the inventory that is already updated by the inventory in ERP. If production is required then sales update ERP which auto updates product teams and checks the availability of raw materials in ERP updated by inventory and all other department update in the ERP.

Siebel is CRM Customer Relationship Management which is used to automate the aspects of the business and allow an enterprise to perform and coordinate tasks over the internet. It is a software that improves efficiency of business process. It is used where we have a huge customer base for example in banking a customer keeps contacting bank daily so CRM is implemented manage their requests and keep track of the status.

Siebel is involved in design, development of the CRM applications

1.1 CHEP

CHEP (Commonwealth Handling Equipment Pool) is a logistics company, owned by Brambles, that deals in pallet and container pooling services and provides services to customers in several industrial and retail supply chains. The current pricing structure is complex for the customers. This causes recurrent complaints and pricing disputes leading to credit and invoice adjustments. The complexity is most noticeable in a large number of low revenue generating fees (transportation charges) and penalties originally designed to protect CHEP's pallets. The pricing complexity also causes concerns internally as the system design allows only a low level of flexibility and leads to labor intensive, ad-hoc fixes for customers, moving away from standard country-level pricing. This web application is developed to simplify the pricing and reduce complexity internally as well as for the customers. It delivers revenue and efficiency gains across multiple areas and lower number of manual intervention.

The following are the advantages of newly developed web based system

- Pricing nomenclature aligned with supply chain terminology (e.g. rebranding of Base Transfer Fee to Outbound fee).
- Simplified marginal fee and penalty structure by merging or removing low revenue fee.
- Headline fee structure left as par current date in order to reduce the implementation revenue risk.
- Structured approach for assigning billing options to customers in order to provide pricing flexibility where requested by the customers and simplicity elsewhere i.e. providing the right level of complexity to the right customers.
- Simplified internal structures- mainly concerning systems, processes, pricing control and governance.

1.2 PURPOSE

There are two types of customers existing and new in the system. The purpose of the application is to calculate the price for pallet that is to be charged to the customer.

1.3 MODULES OF THE APPLICATION

There are three major screens of this web application the sales user, the finance user, the master data user.

1.3.1 SALES USER

- **DASHBOARD:-** This is the landing screen of the application of the sales user.

- **RATE CARD BUILDER:-** The user is able to create a rate card in the arte card builder screen.

- **QUESTIONS:-** The user needs to answer data level \rate card level questions in the questions screen.

- **SCENARIO BUILDER:-** The user is able to create a Quote from the existing Default Scenario in the Scenario Builder screen.

- **REVIEW SCENARIOS:-** The scenarios that are created are displayed in the Review Scenarios screen.

- **QUOTES:-** The created quote is sent to finance for approval/rejection by the Sales in the Quotes screen.

- **RATE CARD:-** Rate card is the pricing that is generated for the quote based on billing options in the rate card screen

- **CONTRACT:-** The contract is generated for the customer in Contract screen.

The sales user can access the entire above screen in the application. The sales user can create a new customer in the system. By clicking create new project in the dashboard screen, we can create a new prospect. Filling in the entire necessary information click submit to create a new prospect. After successfully submitting the prospect details, the cases will move to the rate card builder. The sales user can work on existing quotes, contracts, rate cards. The user can search for an existing Quote/Rate Card/Contract and work on the same. The user can start working on a new quote from the dashboard screen. In the dashboard screen, new prospect is clicked to navigate to search screen.
In the search screen customer can be searched based on the name of their global id.
You can add customer to the existing quote.

The to-do list section displays the queue of the quote along with the count of the same is displayed.

- **ONGOING**:-The quote that is in work in progress and and are being worked on are displayed here.

- **APPROVED**:- The quote that has been approved by finance/customer is displayed in this queue.

- **FINANCE RECOMMENDATIONS**:-Only the quote that has been received from finance and is in process of rejection is displayed here.

- **PENDING APPROVAL**:- The quote that has been sent for approval to finance is displayed here.

- **WITH CUSTOMER**:- The quote that has the response pending from the customer is displayed here.

- **ALERTS**:- Once the alert is ready by the user, it gets displayed in the alerts section in the To do list.

- **CONTRACTS**:- In this queue the contracts rejected by the customers are displayed.

The rate card builder is the starting screen. Here we can mention the size and volume of the product and also mention if the receiving end is domestic or not and if no then any additional country can be added.

Next screen is the scenario builder screen. Here a user can create a quote from the existing scenario and can customize the options like apply discount and change billing options. A user can also review the scenarios. The scenarios that have been created can be edited in this screen. By clicking edit scenario we can edit.

Once the scenario is created a quote is created from the scenario and sent to finance for approval or rejection from the sales if not pre approved. If the quote is pre approved then it is sent to customer for approval. After the quote has been approved a rate card is built. Rate card is generated for those customers that have been approved by the customer. After all this a contract is generated and approved or rejected.

1.3.2 FINANCE USER- The finance user is responsible for approving or rejecting the contracts and making necessary recommendations.

- **DASHBOARD:-** This is the landing screen of the application for the finance user.

- **REVIEW THE DOCUMENTS:-** Finance reviews the documents that are sent by the sales in the review the documents screen

- **REVIEW THE QUOTE:-** The quote that is sent by the sales user is displayed in the review the quote screen.

- **PLAY AND VIEW P&L:-** In the Play P&L screen. The arte card scenario data is fetched and accordingly displayed on the screen. The user is able to change the data with play with P&L section. The user is able to view the created scenario in the View P&L Scenarios screen.

The stage of the quote along with the count of the same is displayed in the To do list

- **OVERALL**:- All the quotes that are received from sales for approval are displayed.

- **PENDING**:- The quotes that are approved by finance and are yet to be sent to sales are displayed here.

- **APPROVED**:- The quote which are approved by finance and sent to sales.

- **RECOMMENDATION**:-Quotes with some suggestion are displayed here.

- **AUTO-APPROVED**:- The pre approved quote that is not sent to finance gets displayed here.

Finance reviews the documents that are sent by sales. The quote is self assigned to be able to work on it. Finance user can also suggest some necessary changes. After approving the quote finance user can send it to sales.

1.3.3 MASTER DATA USER

- **DASHBOARD:-** This is the landing screen of the application for the Master Data user where a deal is created.

The navigation of the quote through various queues is explained in the To-do list.

- **NEW RATE CARDS GENERATED**:- The quote that is approved by the customer allows the sales to generate the rate card that gets displayed in this queue.

- **MY QUEUE**:- All deals that are self assigned are displayed here.

Here the user can self assign the deals so that rate card can be downloaded. The user is able to download the active rate cards for that specific data range entered in excel

format. If the user does not enter any data range, then all the generated active rate cards are downloaded in excel format. If the user enters the from date and and does not enter the to date field, then all the generated Active Rate Cards from that specific date upto the current date get downloaded in excel format.

CHAPTER 2

QUALITY ASURANCE

Quality can be defined as the ability to meet the expectations of the customers in terms of product, design, reliability and price. Assurance is a kind of certainty which gives confidence that the product will meet the expectations of the customers and work without any problems.

Quality Assurance also called Quality Testing is done to ensure that the need and expectations of the customers are being catered to and best possible product is being provided to them. The main aim of QA is to provide is to deliver Quality products to the customers by improving the quality of the processes at each and every step. An organization has to make sure that every process is up to the mark and quality products are delivered to the clients.

2.1 SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

Quality assurance or Testing is one the important part of the Software Development Life Cycle. There are various phases of SDLC

- **Requirement analysis**

This stage gives the scope of the project. It is the first stage of the SDLC process. A proper plan for requirements of Quality Assurance and the risks involved with it are done at this phase.

- **Feasibility Analysis**

In this phase define the software needs. It includes everything that should be developed and designed during the project life cycle. It is used to check if the project can be completed in the budget or not also if the software is supported by the computer.

- **Design**

In this phase the documents are prepared as per the requirement. Here two types of design documents are created High level design and low level design.

- **Coding**

This phase comes after the design phase is done. In this phase the developers start writing the code in a chosen programming language. Here different programming tools are used and proper guidelines are followed.

- **Testing**

Here all the testers start working to ensure that the product is up to the mark. Various testing techniques are followed and it is ensured that the product is up to the mark.

- **Installation/Deployment**

Once the system has been tested and it is ensured that there are no errors in the system the deployment starts

- **Maintenance**

Once the product has been developed, the customer will start using the product. Some new activities occur like up gradation to the newer versions, enhancing the software by adding some new features.

2.2 POPULAR SDLC MODELS

- **WATERFALL MODEL**

In the waterfall model every step is done sequentially. The first phase in the waterfall model is the requirements phase in which detailed test plan is made. In this model only when the first phase is completed then move to the next phase. You cannot move to the next phase if the first phase is not done.

- **ITERATIVE MODEL**

In this model, a project is divided into small parts and on each part waterfall model is run, and at the end entire models are combined together and then tested. One more advantage of this model is that here it is not the case that first one phase will be completed and then moves to another. In this model a feedback is obtained at the end of each cycle.

- **AGILE METHODOLOGY**

The above two models work on the concept that the requirements do not change throughout the project but as the complexity increases there is a constant change in the requirements. There are times when customer is not sure about the requirements and he can make changes in the requirements. Therefore if a customer demands certain changes in the requirements the above two models cannot be used but Agile model is used so that as per the requirements changes can be made whenever required.

2.2 SOFTWARE TESTING LIFE CYCLE (STLC)

It is used to define the steps that can be used in the process of software testing

- **Requirement Analysis**

In this phase requirements are understood by interacting with the client. The requirements can be understood in a manner that what a client expects the system to do or how much secure or what is the expected performance of the software

- **Test Planning**

In this phase test strategy is determined and the test plan is also made by estimating the cost of the project.

- **Test Case Development**

In this phase test cases and test scripts are created, test cases are verified and test cases are reworked if needed.

- **Test Environment Setup**

It is used to determine the conditions under which the software or hardware is tested. Here architecture is to be understood and for the test environment hardware and software is prepared.

- **Test Execution**

Based on the test cases and test plans that have been created testers carry out testing. If there are any errors they are sent to developers for correction and again testing is carried out.

- **Test Closure**

Here the testing team analysis strategies that must be implemented in future or the lessons that might have been learned from current scenario. The idea is that the same mistakes should not be repeated.

2.3 WEB TESTING

Web testing is a testing technique that is used to test the web sites for errors or bugs. A web system needs to be checked properly so that the end system that is delivered to the client is up to the mark and meets the requirements. There are various types of testing that are included in Web Testing

2.3.1 Functionality Testing

In functionality testing, all the links, fonts, color, font size, picture size. If there are any links in the web application then those links should be checked. If there are any internal links, external links, check if these links are working properly.

If there are any forms on the web application then we have to see if forms are receiving correct information from the user. All the fields should be checked properly ensure that correct values are being inserted in the fields. If username field is there then make sure that correct values are inserted into this field. There should be options to modify, edit or update the forms.

Cookies can also be tested. An application can be tested by enabling and disabling the cookies. You can also check the application by deleting the cookies and what effect does it leave on the application.

Database testing can also be done. Testing a database involve checking if the database queries are correct. While executing create, delete and update query expected changes are being made in the database.

2.3.2 Usability Testing

It is used to check how easy the application to use is. It sees how easy an end user finds the application to use. First test is the test for navigation. It means how the user surfs web pages how easily he can access the buttons or navigate through the application. Content should be proper, for this content checking is done. Check is the application is grammatically correct, avoidance in using dark colors as they might be the reason of distraction to the user. The content hat is given in then application should be meaningful and the images should be placed at the correct positions. A user should not find difficulty in memorizing the steps in the application and if a user has used the application several times then the next time he should automatically remember the steps. This means that the application should be easy enough to understand and memorize.

2.3.3 Interface Testing

In web testing there are two major interfaces that need to be tested

- Web server and application server
- Application server and database server

Testing these interfaces mean that communication between the servers are proper and if there are any errors then that can be handled properly.

2.3.4 Compatibility Testing

This testing is used to check that the application is compatible enough to work on different browsers, different operating systems.

Browser compatibility means that the applications are not dependent on a single browser. For compatibility testing an application should run on different browsers like internet explorer, Firefox, opera etc.

The application should run on different operating systems like UNIX, MAC, Linux, Solaris etc.

A web application should not only run on the desktops, it should be compatible enough to run on mobile applications.

If you are providing a print option then make sure that fonts, graphics are printed properly. The size that is mentioned should match the paper size mentioned for printing.

2.3.5 Performance Testing

This type of testing is done to ensure that web application can tolerate if there are situations of high load. It can include two types of testing Load and Stress testing.

- **Load Testing** is used to see that if there are many users that are accessing the application at the same time then the system should be able to handle the load in the peak times. If there are large inputs from the users or if there are multiple user requests then that should be easily taken care of.

- **Stress Testing** means testing the capability of the system by extending the system beyond its threshold point. It is used to see that by giving stress to the system what is the reaction of the system and if the system crashes then how easily does the system recover.

2.3.6 Security Testing

Testing how secure the system is called Security Testing. For example if you are logged into a URL using one username and password the if you try to change the URL of the user, then an error message should be displayed and access should be denied as per security testing. If there are any security breaches then proper messages should be displayed.

CHAPTER 3

QAF

QUALITY ASSURANCE FRAMEWORK

QAF is a robust set of tools that supports rapid development of test automation for **web-based applications.**

QAF provides a rich set of testing functions specifically geared to the needs of testing of a web application

QAF allows many options for finding UI elements since QAF operations are highly flexible and aids in comparing expected test results with actual application behavior.

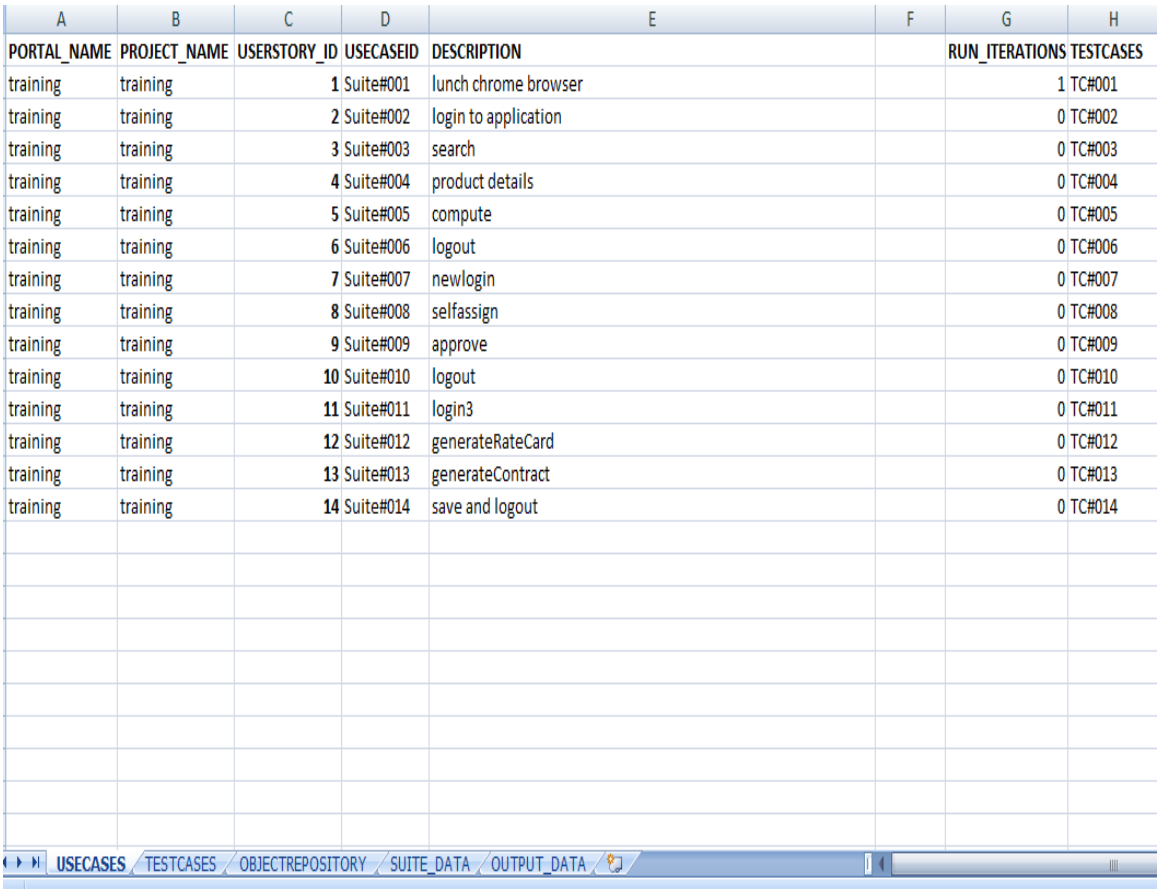
QAF Test automation is the use of software

- To set test preconditions.
- To control the execution of tests.
- To compare the actual results to predicted outcomes.
- To report the execution status.

Commonly, test automation involves automating a process that is manual in nature.QAF automation framework is used when there is repetition. If an application has been tested and there are some problems and then the application is sent back to the developer to make some changes, and then a developer resends it for testing. In such a case QAF automation framework is used. It runs on JAVA script, there is no need of writing a code in this framework. There are some keywords and with those keywords some actions are associated which have a java code. So no need of writing a code only some keywords have to be given.
QAF supports cross browser testing. QAF test cases run on multiple browsers.
Get Text statements provide an efficiency way of comparing expected and actual results and it has an inbuilt reporting mechanism.

3.1 AUTOMATION FRAMEWORK

QAF is an automation framework the various elements of this framework include suite data, use cases, test cases, object repository, and output data. The detailed explanation has been given below



A	B	C	D	E	F	G	H
PORTAL_NAME	PROJECT_NAME	USERSTORY_ID	USECASEID	DESCRIPTION		RUN_ITERATIONS	TESTCASES
training	training	1	Suite#001	lunch chrome browser		1	TC#001
training	training	2	Suite#002	login to application		0	TC#002
training	training	3	Suite#003	search		0	TC#003
training	training	4	Suite#004	product details		0	TC#004
training	training	5	Suite#005	compute		0	TC#005
training	training	6	Suite#006	logout		0	TC#006
training	training	7	Suite#007	newlogin		0	TC#007
training	training	8	Suite#008	selfassign		0	TC#008
training	training	9	Suite#009	approve		0	TC#009
training	training	10	Suite#010	logout		0	TC#010
training	training	11	Suite#011	login3		0	TC#011
training	training	12	Suite#012	generateRateCard		0	TC#012
training	training	13	Suite#013	generateContract		0	TC#013
training	training	14	Suite#014	save and logout		0	TC#014

Figure3.1: Automation Framework

3.1.1 OBJECT REPOSITORY: It contains the reference of the various objects. Using inspect element we find the elements by making use of the locator techniques. You can mention the elements details by ID, Xpath, Name, Link text and CSS.

- **LABEL**-Not a compulsory field but you can mention the label name of the element.
- **LOCATORID**-Mention the name of the locator you want to use.
- **LOCATORTYPE**-Mention the locator type you are using.
- **LOCATOR ELEMENT**- Mention locator details.

A	B	C	D	E	F
SCREENNAME	LABEL	LOCATORID	LOCATORTYPE	LOCATORELEMENT	
		reports	xpath	//*[@id="ms_subjectcategory"]/a	
		username	xpath	//*[@id="username"]	
		password	xpath	//*[@id="password"]	
		submit	xpath	//*[@id="btnlogin"]	
		search	xpath	//*[@id="ongoingDetailtable_filter"]/label/input	
		clickon	xpath	//*[@id="ongoingDetailtable"]/tbody/tr/td[1]	
		editquote	xpath	//*[@id="QuoteCompareSection"]/div[2]/div[3]/div/button	
		forcastactivity	id	3848686	
		inbound	xpath	//*[@id="inbound"]/td[1]/span	
		price	xpath	//*[@id="customerMetrcsDetailsDynamictableId"]/table/tbody[1]/tr[2]/td[8]	
		totalspend	xpath	//*[@id="customerMetrcsDetailsDynamictableId"]/table/tbody[1]/tr[2]/td[9]	
		error	xpath	//*[@id="error_message"]/font/b/div[1]	
		add	xpath	//*[@id="addBtnContainerId"]/a	
		selc	xpath	//*[@id="s2id_zoneId_"]/a	
		scenario	xpath	//*[@id="saveNewButtonid"]	
		scname	xpath	//*[@id="saveNemeTextid"]	
		save	xpath	//*[@id="savescenario"]/div[2]/div[3]/div/button[2]	
		err	xpath	//*[@id="swal2-title"]	
		ok1	xpath	/html/body/div[16]/div/div[10]/button[1]	
		nscname	xpath	//*[@id="scnarionameTextid"]	
		review	xpath	//*[@id="viewAllQuotesId"]	
		up	xpath	//*[@id="containerDataId"]/div[3]/a	
		quote	xpath	//*[@id="createQuoteButtonId"]	
		chek	xpath	//*[@id="ongoingDetailtable"]/tbody/tr/td[1]	
		customer	xpath	//*[@id="buttonSendToCclient"]	
		error1	xpath	//*[@id="swal2-title"]	
		ok1	xpath	/html/body/div[9]/div/div[10]/button[1]	
		finance	xpath	//*[@id="buttonCheckedQuotes1"]	

USECASES | TESTCASES | OBJECTREPOSITORY | SUITE_DATA | OUTPUT_DATA

Figure3.2: Object Repository

3.1.2 USE CASES: In us case sheet the project name is mention along with the portal name. Test case number is also mentioned in this sheet along with the use case id.

- **PORTAL_NAME**-Mention the project name.
- **PROJECT_NAME**-Mention the sub project name or project name
- **USERSTORY_ID**- Mention the user story or requirement.
- **USECASEID**-Mention the use case ID for your requirement.
- **DESCRIPTION**- gives details about your test case what you want to test.
- **RUN_ITERATIONS**- mention how many times you want to run the test cases.
- **TESTCASES**- mention the test cases id which you are going to map with test cases sheet.

A	B	C	D	E	F	G	H
PORTAL_NAME	PROJECT_NAME	USERSTORY_ID	USECASEID	DESCRIPTION		RUN_ITERATIONS	TESTCASES
training	training	1	Suite#001	lunch chrome browser		1	TC#001
training	training	2	Suite#002	login to application		0	TC#002
training	training	3	Suite#003	search		0	TC#003
training	training	4	Suite#004	product details		0	TC#004
training	training	5	Suite#005	compute		0	TC#005
training	training	6	Suite#006	logout		0	TC#006
training	training	7	Suite#007	newlogin		0	TC#007
training	training	8	Suite#008	selfassign		0	TC#008
training	training	9	Suite#009	approve		0	TC#009
training	training	10	Suite#010	logout		0	TC#010
training	training	11	Suite#011	login3		0	TC#011
training	training	12	Suite#012	generateRateCard		0	TC#012
training	training	13	Suite#013	generateContract		0	TC#013
training	training	14	Suite#014	save and logout		0	TC#014

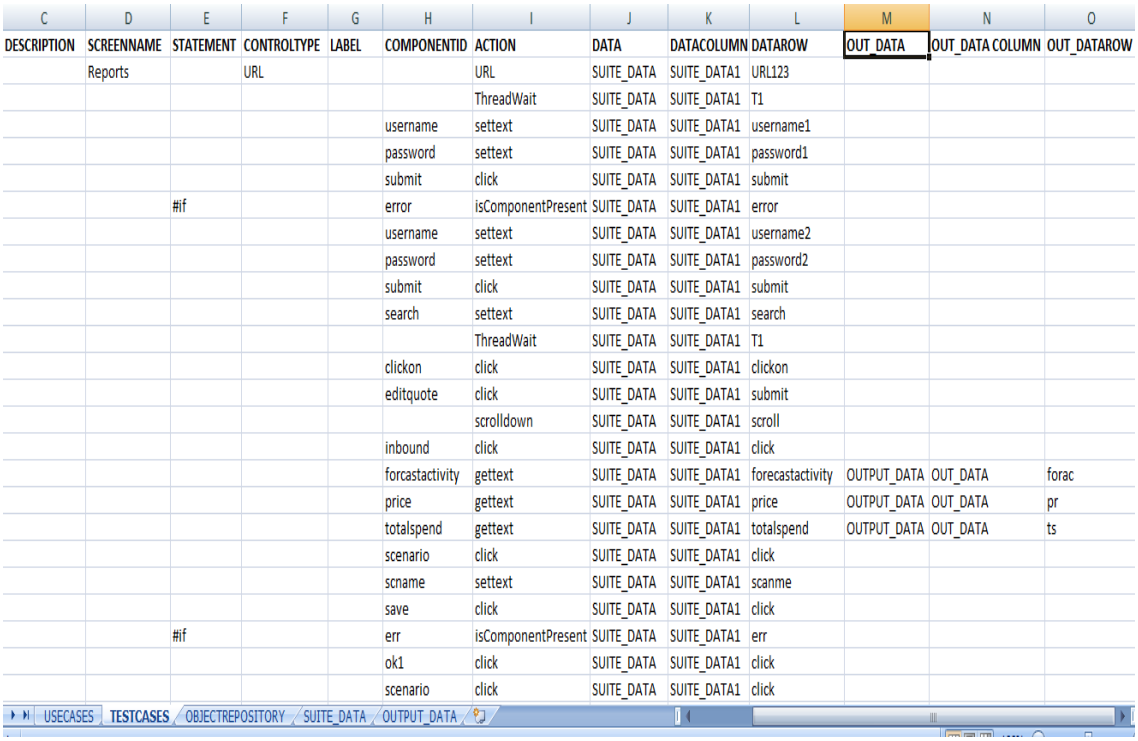
USECASES | TESTCASES | OBJECTREPOSITORY | SUITE_DATA | OUTPUT_DATA

Figure3.3: Use Cases

3.1.3 TEST CASES: Here the test cases are written. If login has to be made then detailed steps ranging from opening the browser to closing are written along with proper description. The sequence in which the test cases are written plays an important role as the steps execute in sequence.

- **TESTCASEID**- Mention the test cases id which is present under the use case.
- **TESTCASENO**- Mention test step number.
- **DESCRIPTION**- Mention test step description.
- **SCREENNAME**-Describe the screen name for the test case.
- **STATEMENT**-Describe the #IF #ELSE #END if required.
- **CONTRIOLTYPE**- Here the element type can be mentioned
- **LABEL**-You can mention the label name of the element.
- **COMPONENTID**-mention the element ID for taking the action.
- **ACTION**- Mention the action name.

- **DATA**- sheet name from where we are getting the input data.
- **DATACOLUMN**-Column name from where we are getting the input data.
- **DATAROW**-Row name from where we are getting the input data.
- **OUT_DATA**-Sheet name from where we are getting the data for compare.
- **OUT_DATACOLUMN**-Row name from where we are getting the data for compare.
- **OUT_DATAROW**-Row name from where we are getting the data for compare.
-



C	D	E	F	G	H	I	J	K	L	M	N	O
DESCRIPTION	SCREENNAME	STATEMENT	CONTROLTYPE	LABEL	COMPONENTID	ACTION	DATA	DATACOLUMN	DATAROW	OUT_DATA	OUT_DATA COLUMN	OUT_DATAROW
	Reports		URL			URL	SUITE_DATA	SUITE_DATA1	URL123			
						ThreadWait	SUITE_DATA	SUITE_DATA1	T1			
					username	settext	SUITE_DATA	SUITE_DATA1	username1			
					password	settext	SUITE_DATA	SUITE_DATA1	password1			
					submit	click	SUITE_DATA	SUITE_DATA1	submit			
		#if			error	isComponentPresent	SUITE_DATA	SUITE_DATA1	error			
					username	settext	SUITE_DATA	SUITE_DATA1	username2			
					password	settext	SUITE_DATA	SUITE_DATA1	password2			
					submit	click	SUITE_DATA	SUITE_DATA1	submit			
					search	settext	SUITE_DATA	SUITE_DATA1	search			
						ThreadWait	SUITE_DATA	SUITE_DATA1	T1			
					clickon	click	SUITE_DATA	SUITE_DATA1	clickon			
					editquote	click	SUITE_DATA	SUITE_DATA1	submit			
						scrolldown	SUITE_DATA	SUITE_DATA1	scroll			
					inbound	click	SUITE_DATA	SUITE_DATA1	click			
					forcastactivity	gettext	SUITE_DATA	SUITE_DATA1	forecastactivity	OUTPUT_DATA	OUT_DATA	forac
					price	gettext	SUITE_DATA	SUITE_DATA1	price	OUTPUT_DATA	OUT_DATA	pr
					totalspend	gettext	SUITE_DATA	SUITE_DATA1	totalspend	OUTPUT_DATA	OUT_DATA	ts
					scenario	click	SUITE_DATA	SUITE_DATA1	click			
					scname	settext	SUITE_DATA	SUITE_DATA1	scanme			
					save	click	SUITE_DATA	SUITE_DATA1	click			
		#if			err	isComponentPresent	SUITE_DATA	SUITE_DATA1	err			
					ok1	click	SUITE_DATA	SUITE_DATA1	click			
					scenario	click	SUITE_DATA	SUITE_DATA1	click			

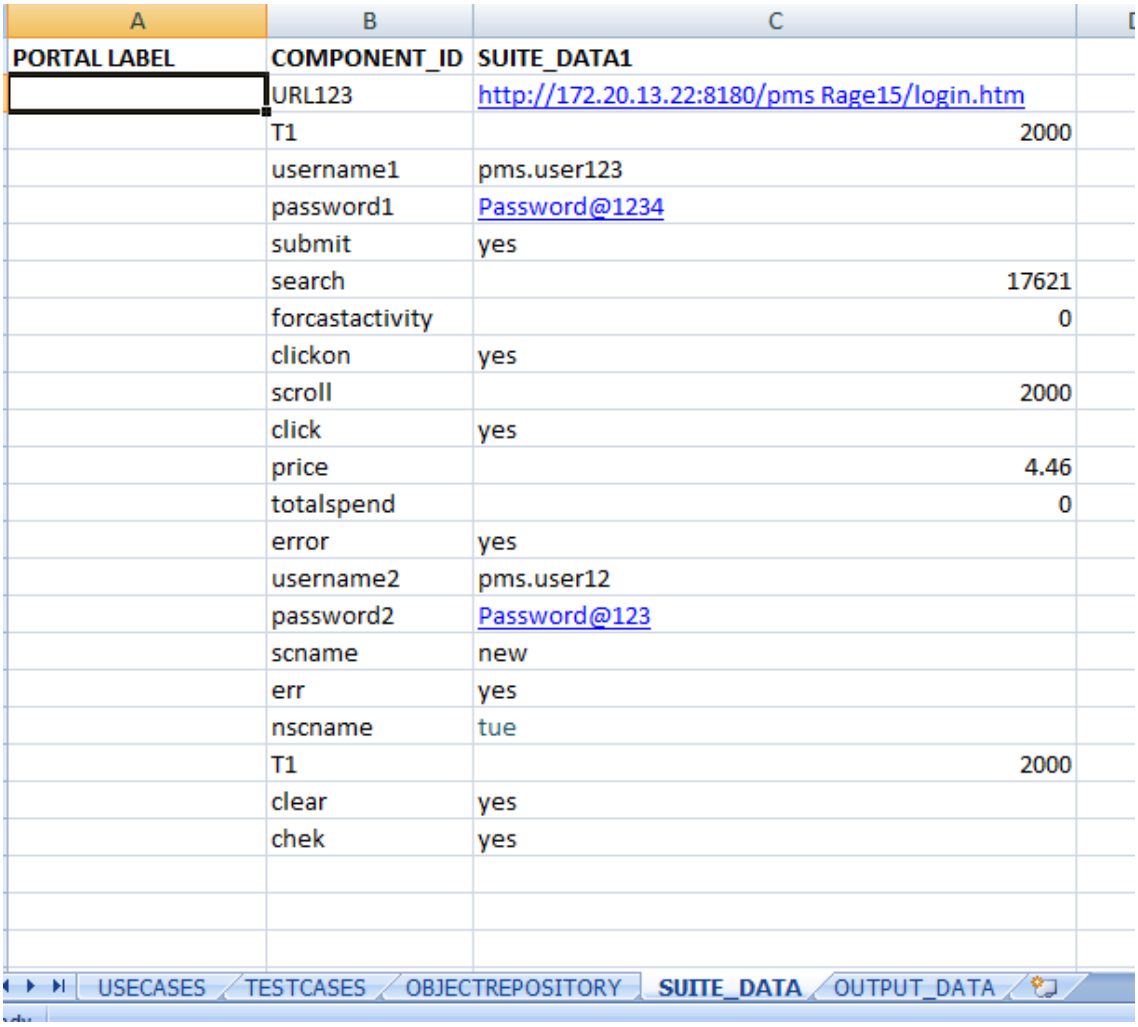
USECASES | TESTCASES | OBJECTREPOSITORY | SUITE_DATA | OUTPUT_DATA

Figure3.4: Test cases

3.1.4 SUITE DATA: Suite data contains the input. If a username has to be entered then its value has to be given in the suite data.

- **LABEL**- Mention the label name of the element.
- **COMPONENT_ID**- Mention the element id for input data.
- **SUITE_DATA1**-Mention the input data which you want to enter to the apliaction.

- **SUITE_DATA2**-Mention the input data for 2nd iteration which you want to enter to the application.
- **SUITE_DATA2**- Mention the input data for 3rd iteration which you want to enter to the application.



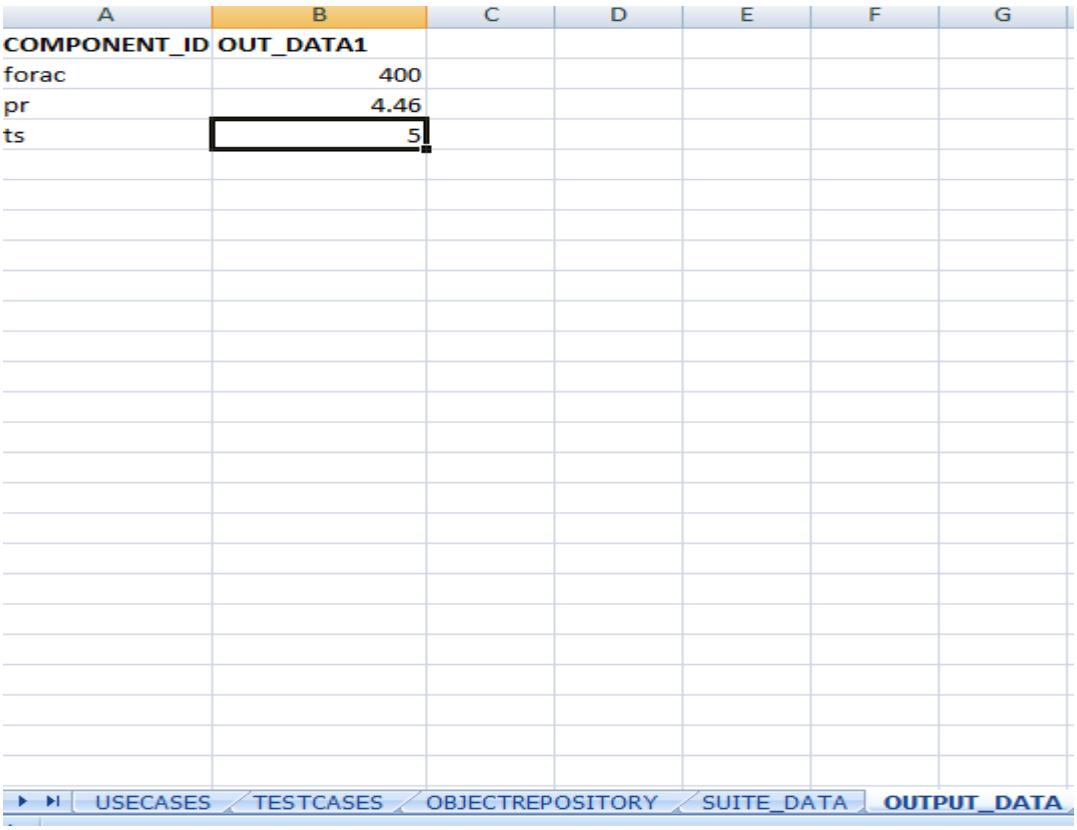
	A	B	C
1	PORTAL LABEL	COMPONENT_ID	SUITE_DATA1
2		URL123	http://172.20.13.22:8180/pms Rage15/login.htm
3		T1	2000
4		username1	pms.user123
5		password1	Password@1234
6		submit	yes
7		search	17621
8		forcastactivity	0
9		clickon	yes
10		scroll	2000
11		click	yes
12		price	4.46
13		totalspend	0
14		error	yes
15		username2	pms.user12
16		password2	Password@123
17		scname	new
18		err	yes
19		nscname	tue
20		T1	2000
21		clear	yes
22		chek	yes

Figure3.5: Suite Data

3.1.5 OUT DATA: Output data contains those values which are the expected output. It helps in drawing a comparison between the values in the suite data to that of output data and analyzing if the output is as desired.

- **COMPONENT_ID**- Mention the element id for input data.

- **OUT_DATA1**- Mention the output data which you want to compare with the application.



	A	B	C	D	E	F	G
1	COMPONENT_ID	OUT_DATA1					
2	forac	400					
3	pr	4.46					
4	ts	5					

USECASES | TESTCASES | OBJECTREPOSITORY | SUITE_DATA | OUTPUT_DATA

Figure3.6: Output Data

Using QAF first use cases are written, where we specify the steps and the description of each step. The steps involve opening a web browser, navigating through it, entering some text etc. The test case column in this sheet is mapped to the test case sheet.

In test case sheet the keywords associated with the following actions are given. There are different keywords for each n every action.

In object repository locator type of the elements are given. The locator types are given using xpath. Using Xpath the locator elements are picked up.

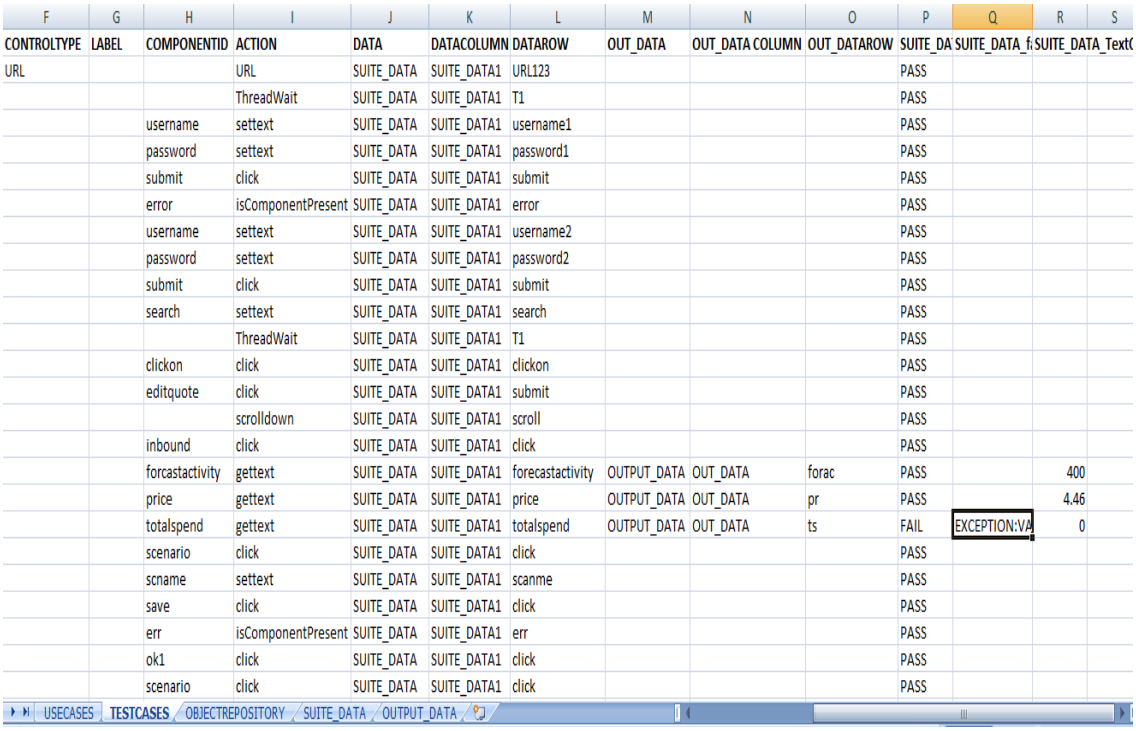
In suite data, input is given for example the URL of the link, if there is username and password field then that is given.

In the Out data we write values for comparison. The values that are to be compared with the input values in the suite data are given here.

The results after testing the web application can be obtained with the help of the report that shows how many test cases pass or fail.

C	D	E	F	G	H	I	J	K
USERSTORY_ID	USECASEID	DESCRIPTION		RUN_ITERATIONS	TESTCASES	SUITE_DATA1_PASS	SUITE_DATA_FAIL	SUITE_DATA_EXCEPTION
1	Suite#001	lunch chrome browser		1	TC#001	18	1	1
2	Suite#002	login to application		0	TC#002			
3	Suite#003	search		0	TC#003			
4	Suite#004	product details		0	TC#004			
5	Suite#005	compute		0	TC#005			
6	Suite#006	logout		0	TC#006			
7	Suite#007	newlogin		0	TC#007			
8	Suite#008	selfassign		0	TC#008			
9	Suite#009	approve		0	TC#009			
10	Suite#010	logout		0	TC#010			
11	Suite#011	login3		0	TC#011			
12	Suite#012	generateRateCard		0	TC#012			
13	Suite#013	generateContract		0	TC#013			
14	Suite#014	save and logout		0	TC#014			
				TestCaseSummary		18	1	

Figure3.7: Result



CONTROLTYPE	LABEL	COMPONENTID	ACTION	DATA	DATACOLUMN	DATAROW	OUT_DATA	OUT_DATA COLUMN	OUT_DATAROW	SUITE_DA	SUITE_DATA_f	SUITE_DATA_Text(
URL			URL	SUITE_DATA	SUITE_DATA1	URL123				PASS		
			ThreadWait	SUITE_DATA	SUITE_DATA1	T1				PASS		
		username	settext	SUITE_DATA	SUITE_DATA1	username1				PASS		
		password	settext	SUITE_DATA	SUITE_DATA1	password1				PASS		
		submit	click	SUITE_DATA	SUITE_DATA1	submit				PASS		
		error	isComponentPresent	SUITE_DATA	SUITE_DATA1	error				PASS		
		username	settext	SUITE_DATA	SUITE_DATA1	username2				PASS		
		password	settext	SUITE_DATA	SUITE_DATA1	password2				PASS		
		submit	click	SUITE_DATA	SUITE_DATA1	submit				PASS		
		search	settext	SUITE_DATA	SUITE_DATA1	search				PASS		
			ThreadWait	SUITE_DATA	SUITE_DATA1	T1				PASS		
		clickon	click	SUITE_DATA	SUITE_DATA1	clickon				PASS		
		editquote	click	SUITE_DATA	SUITE_DATA1	submit				PASS		
			scrolldown	SUITE_DATA	SUITE_DATA1	scroll				PASS		
		inbound	click	SUITE_DATA	SUITE_DATA1	click				PASS		
		forcastactivity	gettext	SUITE_DATA	SUITE_DATA1	forecastactivity	OUTPUT_DATA	OUT_DATA	forac	PASS		400
		price	gettext	SUITE_DATA	SUITE_DATA1	price	OUTPUT_DATA	OUT_DATA	pr	PASS		4.46
		totalspend	gettext	SUITE_DATA	SUITE_DATA1	totalspend	OUTPUT_DATA	OUT_DATA	ts	FAIL	EXCEPTION:VA	0
		scenario	click	SUITE_DATA	SUITE_DATA1	click				PASS		
		scname	settext	SUITE_DATA	SUITE_DATA1	scanme				PASS		
		save	click	SUITE_DATA	SUITE_DATA1	click				PASS		
		err	isComponentPresent	SUITE_DATA	SUITE_DATA1	err				PASS		
		ok1	click	SUITE_DATA	SUITE_DATA1	click				PASS		
		scenario	click	SUITE_DATA	SUITE_DATA1	click				PASS		

USECASES | TESTCASES | OBJECTREPOSITORY | SUITE_DATA | OUTPUT_DATA

Figure3.8: Result

3.2 SELENIUM WEB TESTING

Selenium is software that is used for testing of web applications across different platforms and browsers. The main advantage is that it is free or open source and used as an automated testing suite for web application. Testing that is carried using Selenium is called Selenium web testing. Selenium is not just a single tool but it has some components that cater to the needs of the organization. The four components of Selenium are:-

- Selenium Integrated Development Environment(IDE)
- Selenium remote control(RC)
- Web Driver
- Selenium Grid

Selenium was primarily created by Jason Huggins in 2004. An engineer in some company was working on some application that required some sort of repeated testing. Now since there was repeated manual testing the application was becoming more and more inefficient, so in order to control the browser actions he created a JavaScript so that the browsers actions could be automatically controlled. This program that he created was named as “JavaScriptTestRunner”. When saw the potential in this script to automate web applications he made it free and this was renamed to Selenium Core.

Another issue that was faced was Same Origin Policy Issue. In this issue JavaScript code could not access elements from a domain that is different from where this was launched. For example if a code was created in google.com and it uses a java script program then that script program will access pages only with google.com. It cannot access pages from different sites such as yahoo as they are from a very different domain.

In selenium we are provided with a test domain-specific language called selenese so that test cases can be written in a number of different languages like C#, Java, Python. Selenium can run on windows, Linux, Mac platforms.

3.2.1 SELENIUM IDE

Selenium IDE (Integrated Development Environment) is the simplest tool in Selenium suite. It has a record and play functionality that helps create test cases very quickly. It is not used as an overall solution for developing and maintaining complex test cases therefore used only a prototyping tool. User needs to be familiar with HTML, JavaScript even if he does not have a prior knowledge in programming. The Selenium IDE supports two purposes

- It helps the tester to enter commands with greater ease and more quickly.
- It also does not allow a user to enter invalid commands.

Features of Selenium IDE

- **Menu bar** – It is located on top and file, edit, options are main menus. Using file menu a user can create test cases, save existing test cases. Test cases can be exported in any programming language.

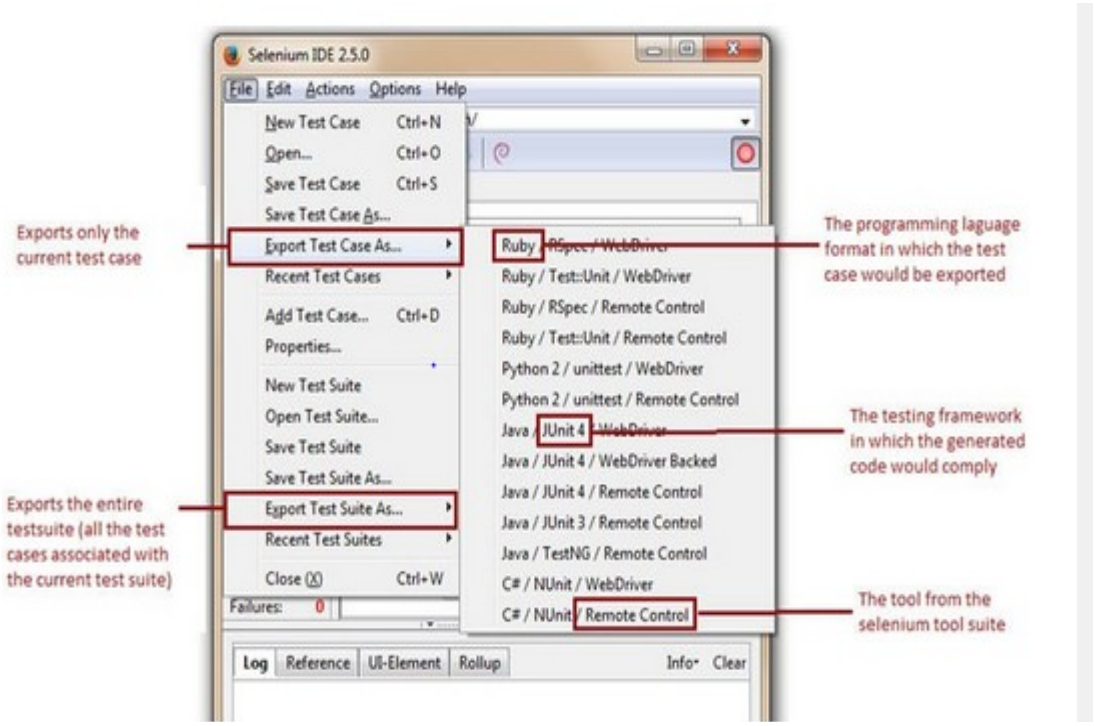


Figure3.9: Selenium IDE Menu bar[1]

- **Edit menu**- It has various options like copy, undo, select. Some of them are insert that allows inserting the new test cases or editing within previous test cases or current test cases.

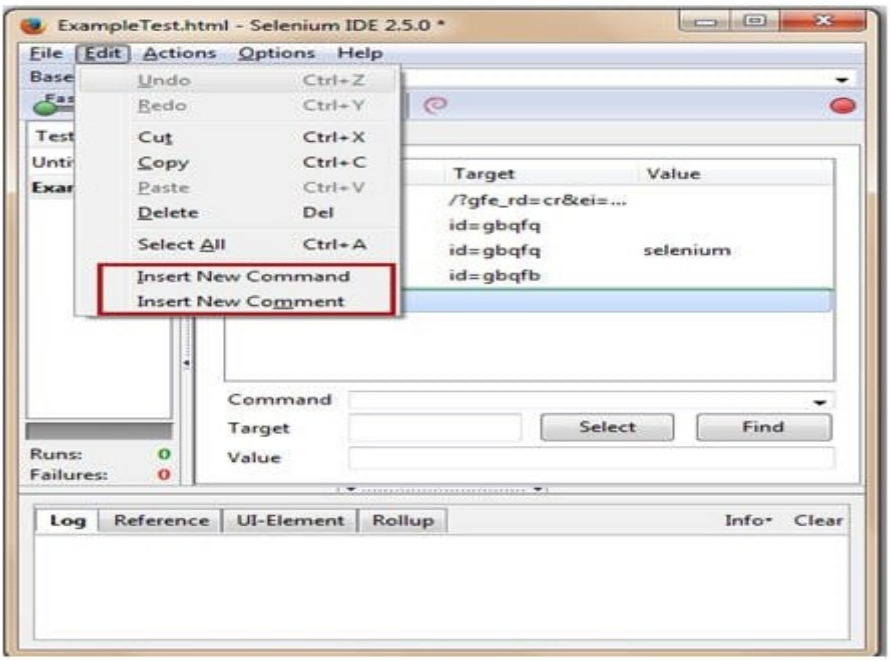


Figure3.10: Edit menu[1]

It allows the facility of inserting new comments and new commands .We can insert the new command in the current test cases

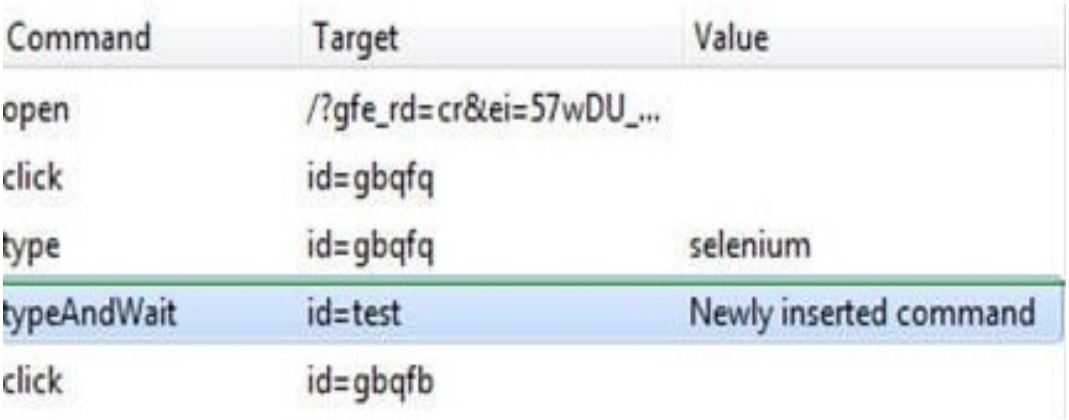


Figure3.11: Inserting new commands[1]

- **Actions menu:-**It provides options like record, pause, resume, play current test case, play entire suite.

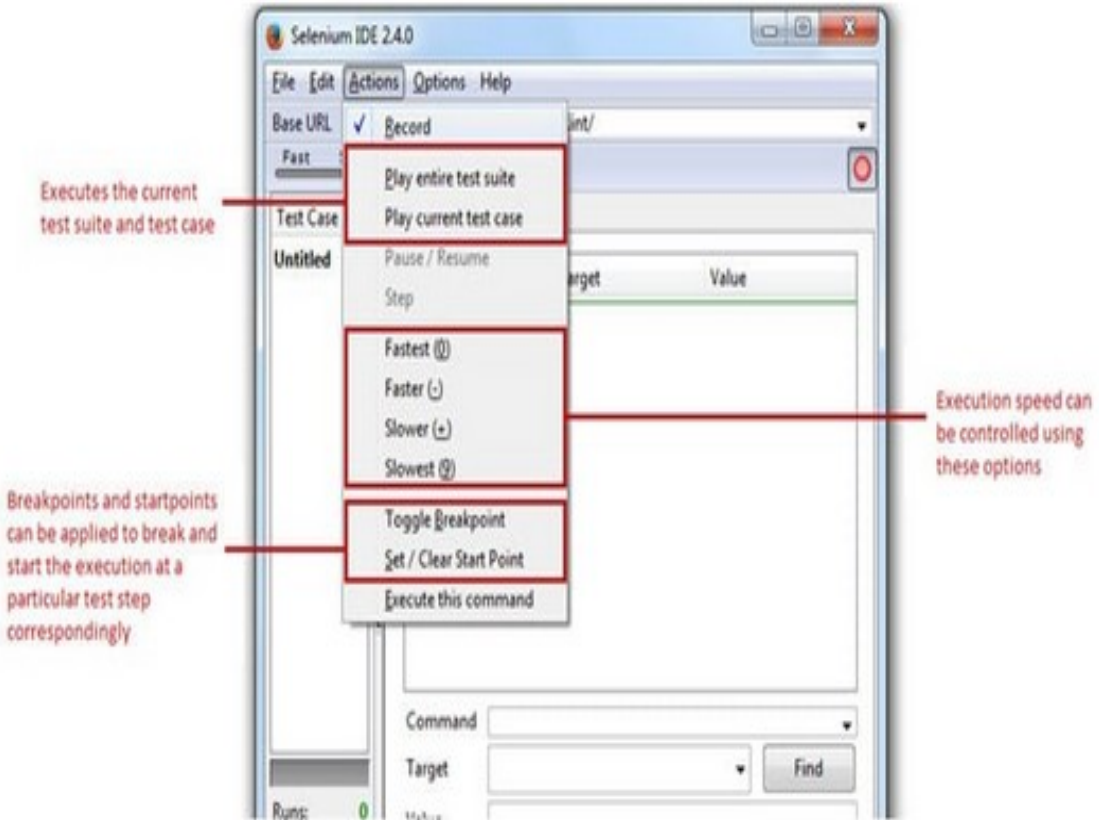


Figure3.12: Actions menu[1]

These are the main menus in Selenium IDE. In selenium the test case can be created using three steps.

Step #:1 Recording- When the browser is opened all the interactions that are done with the browser are recorded and this recording is termed as Selenium IDE Script.

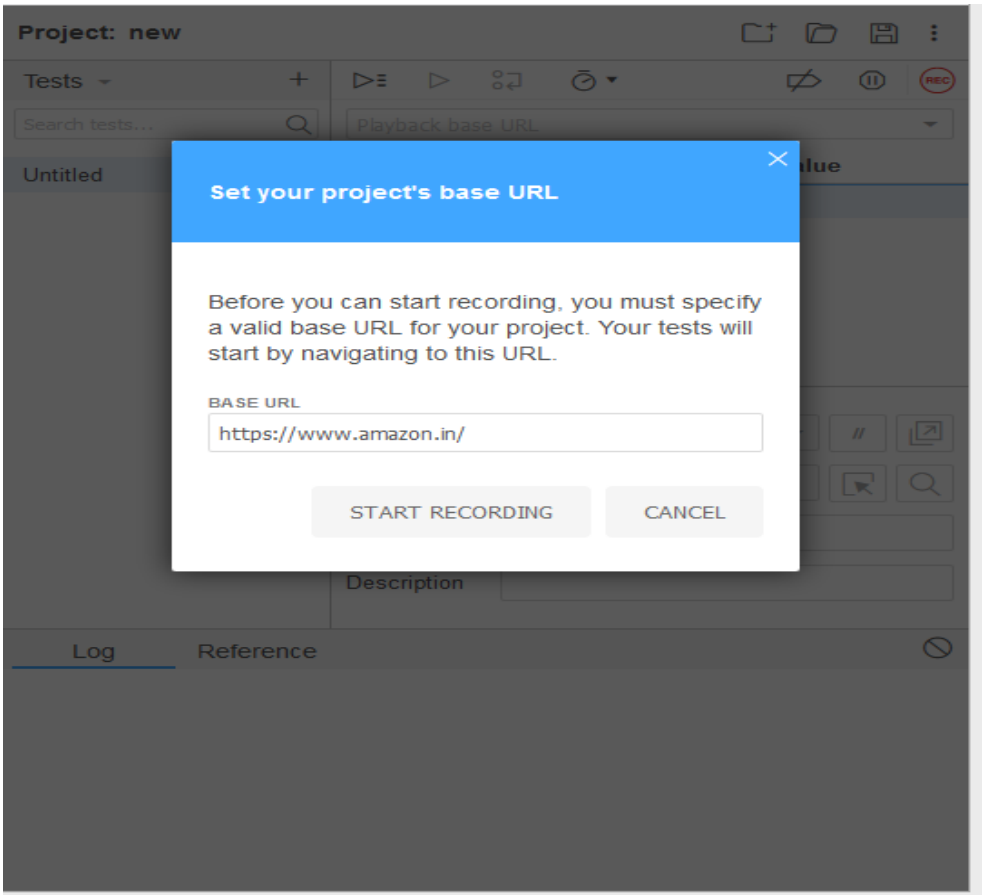


Figure3.13: Starting the Script.

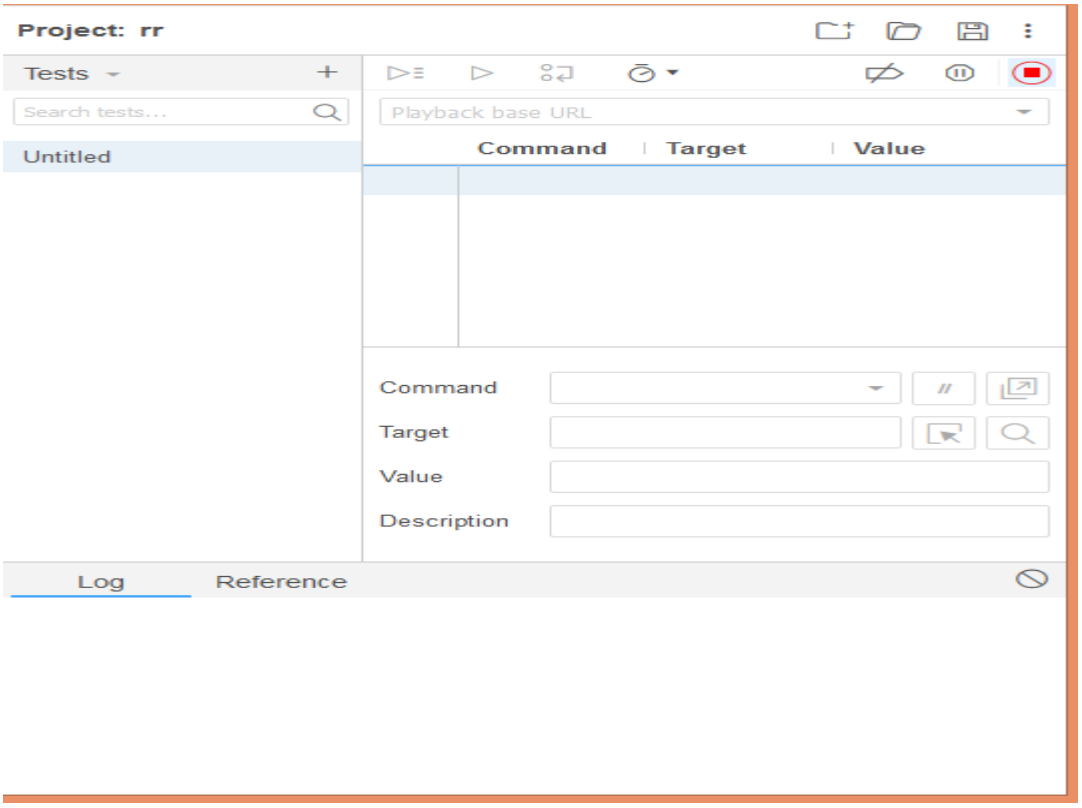


Figure3.14: Selenium IDE is recording

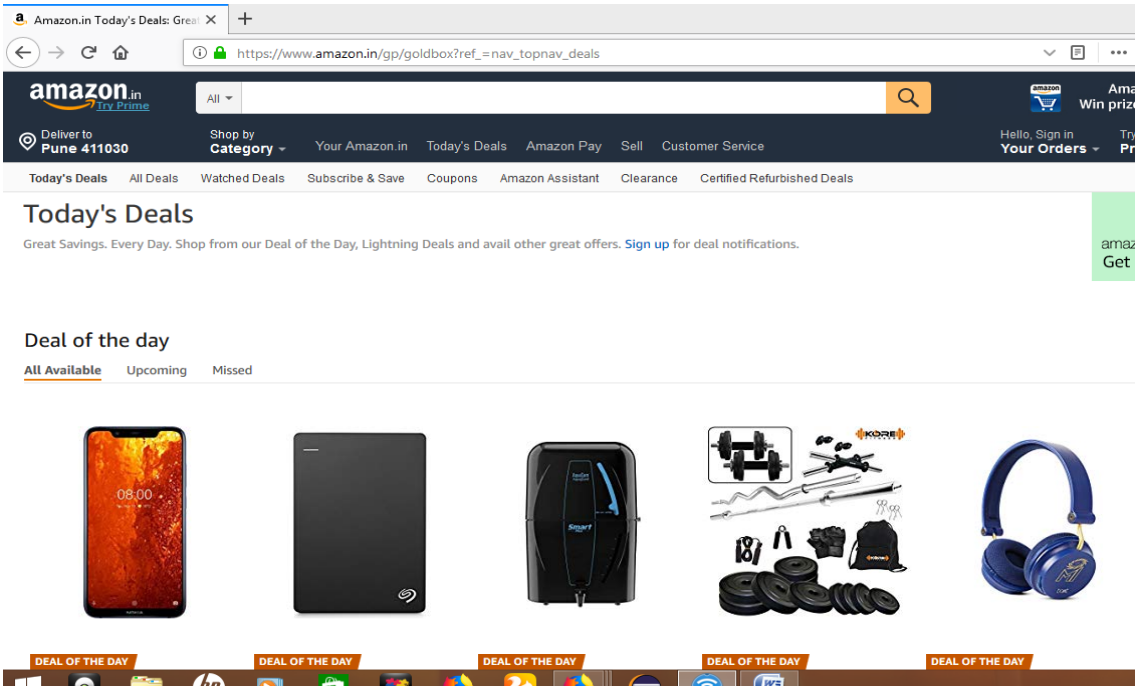


Figure3.15: Performing the actions

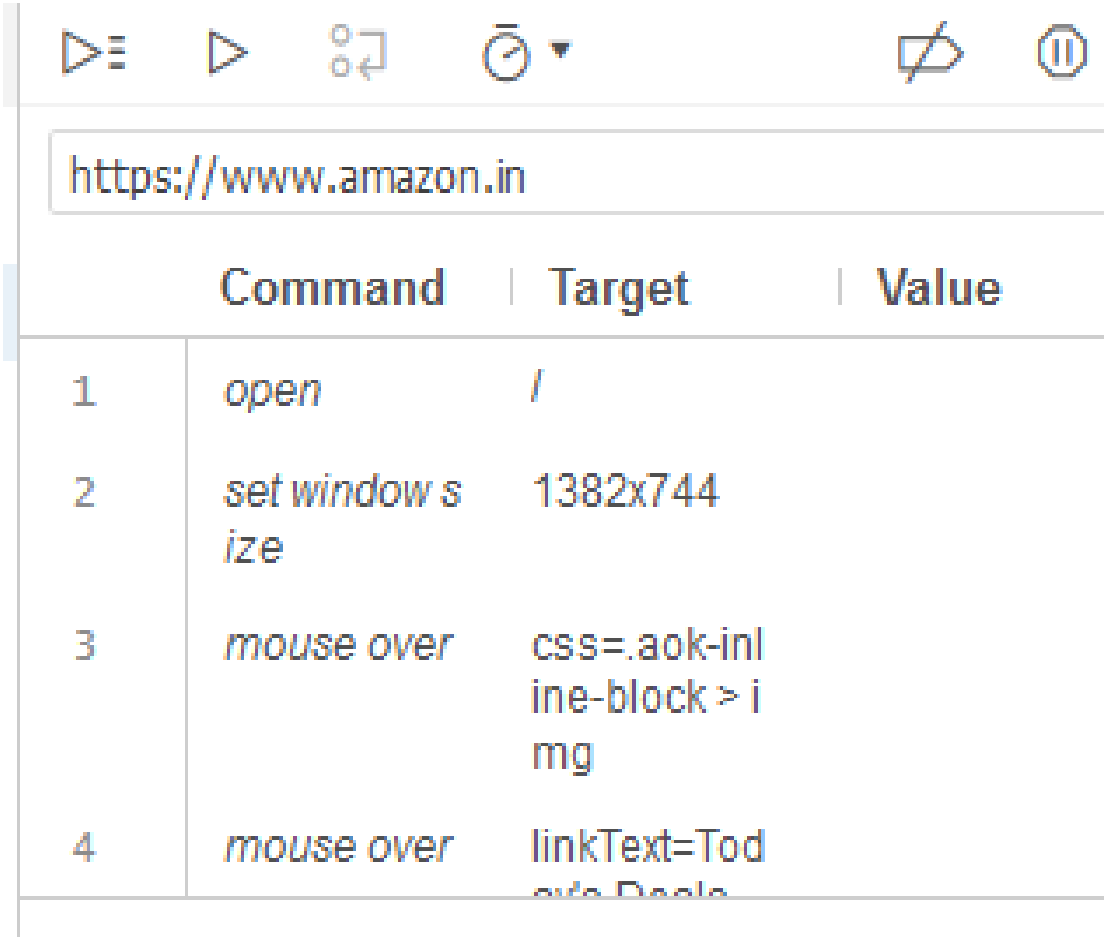


Figure3.16: Recording the actions

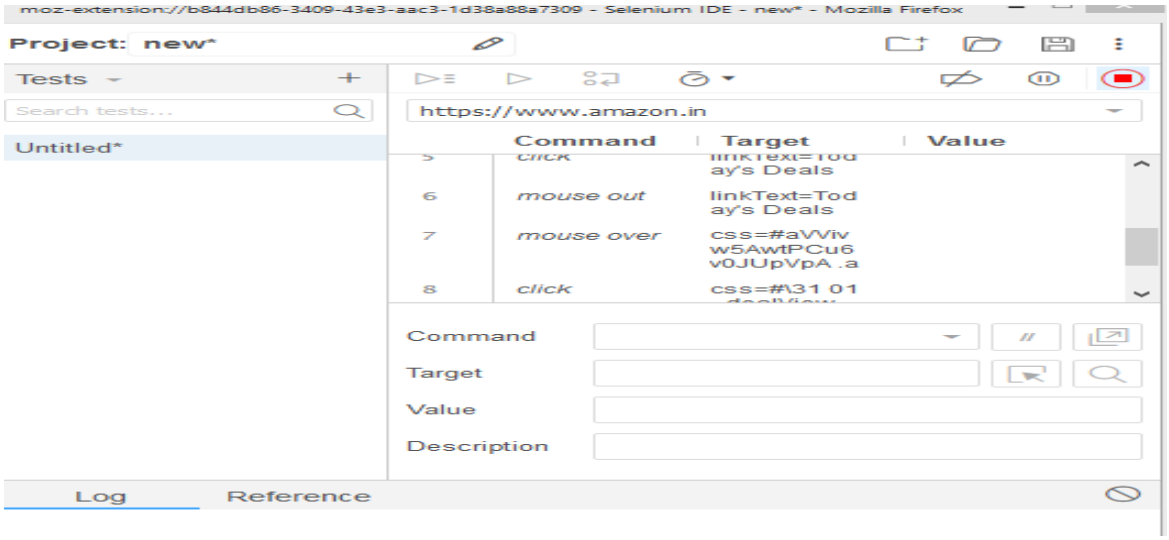


Figure3.17: Recording the actions

Step #:2 Playing back - Here the recorded steps are played to check the success and stability.

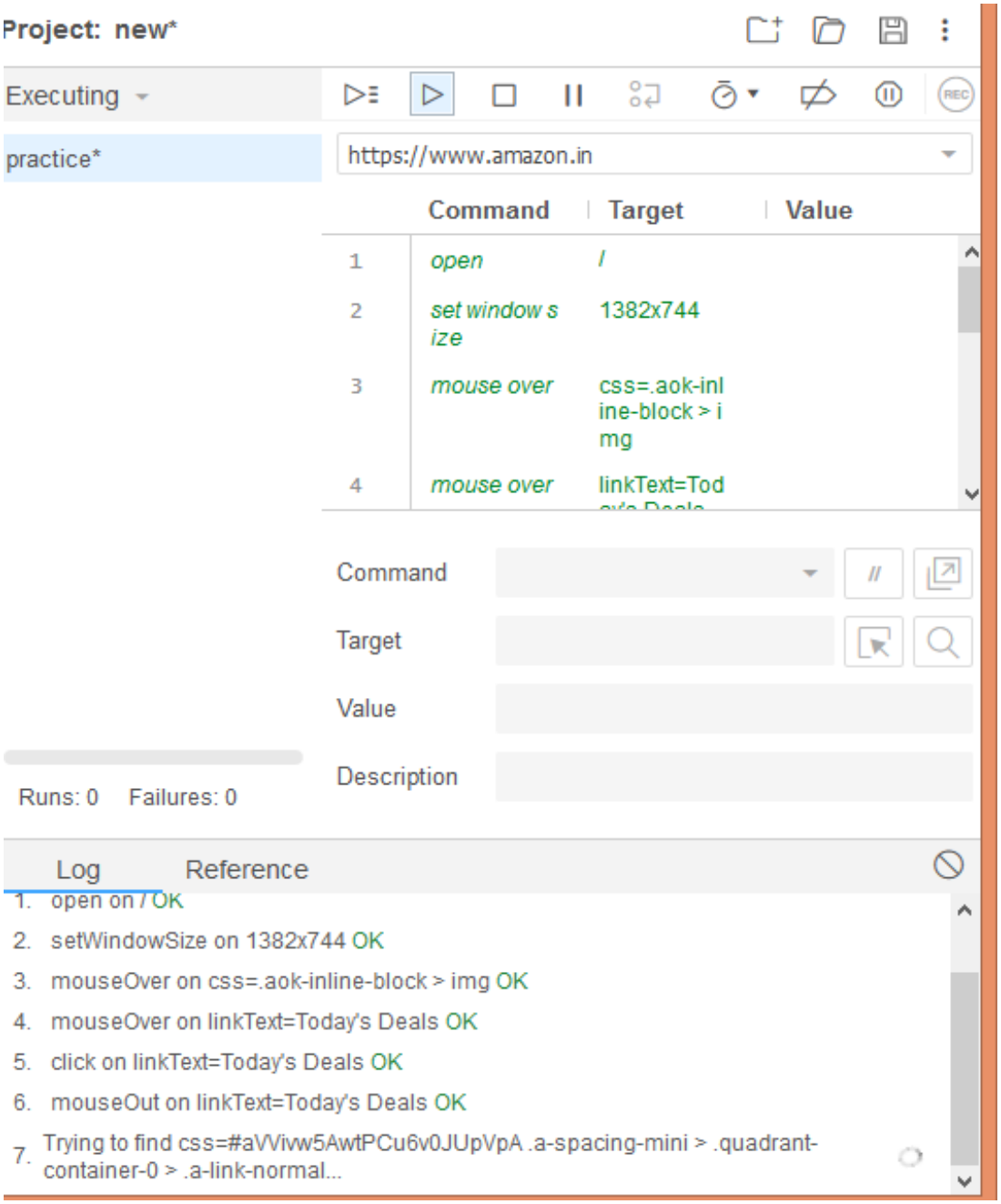


Figure3.18: Playing the actions

Step#:3 Saving – The recorded script can be saved.

3.2.2 SELENIUM RC (REMOTE CONTROL)

This was the main project of selenium that was for a long time before web driver came into exixtance.With the help of programming languages such as Java, C++, C#, Python we can write we can write automated web applications, UI tests and more test cases such as writing or reading files, emailing test results. Selenium can help us to communicate where it passes commands to Selenium Server for execution. The selenium server helps in launching and killing the browser. It also works in executing and running Selenese commands. Using selenium RC we can write test cases in different language. Since RC is slow it is not used much. It was a server to make the system believe that code and the system being tested are on the same domain. First server is started so that the proxy gets injected (to make believe that same domain exists). Then instructions are provided to the server and then the instructions go to the browser. Since there are a lot of steps involved it is a very slow system.

3.2.3 SELENIUM WEB DRIVER

Selenium used to create and execute test cases. Test cases are created using elements (object locator /web driver methods). We need elements to automate, to find web page elements we have to use locator techniques web driver lets you perform actions.

Web Driver is a powerful tool of Selenium. Selenium web driver comes with a lot of advantages and it can work for different browsers unlike Selenium IDE that supported only Firefox. Web driver supports a large number of browsers and can work on different versions. Web driver also works for the advantage of Android Driver and I phone Driver, therefore supports web based testing.

While using Selenium RC and Selenium IDE, we created scripts by recording the steps and playing those steps or they could be created manually using selenium commands called Selenese. There were certain limitations with Selenium IDE. Selenium IDE ahd no support for loops, error handling, conditional statements, These limitations were however overcome by Selenium Web driver as it provided the facility to the user to work on different programming languages and create the test script in whatever language the user wants to create.

There are various languages supported by Selenium Web driver:-

- Java
- C#
- PHP
- Pearl
- Ruby
- Python

Based on which language the user is compatible, he/she can select any programming language and can start building scripts.

Another big advantage of Selenium Web Driver is that it is a very fast and efficient tool. There is a direct communication between web driver client libraries and web browser. This is however an advantage with other tools as they do not communicate directly. First there is communication with the server which in turn communicated with the selenium core and that further communicates with the browser. So this is a long process and reduces the execution speed. This drawback is however overcome by Web driver. For working on web driver first Java has to be downloaded, the Eclipse IDE has to be downloaded and after that Selenium has to be downloaded after all these have been successfully downloaded and configured, we can start web testing on Eclipse.

3.2.4 SCRIPT CREATION

The various steps in the test creation will be

- Launching the browser and opening the URL.
- Performing various actions
- Closing the browser.

anote.java | fbtest.java | Facbook.java | Grid.java | Track.java | Day1.java | Prat.java | WebDriver.class

```
1  package com.edurekhaselenium.webdriver.basic;
2
3  import java.util.concurrent.TimeUnit;
4
5  import org.openqa.selenium.By;
6
7  import org.openqa.selenium.WebDriver;
8  import org.openqa.selenium.chrome.ChromeDriver;
9
10
11 public class Facbook {
12     WebDriver driver;
13     static String nodeUrl;
14
15     public void invokeBrowser() {
16         try {
17             System.setProperty("webdriver.chrome.driver", "C:\\Users\\HP First\\Desktop\\chromedriver.exe");
18             driver = new ChromeDriver();
19             driver.manage().window().maximize();
20             driver.manage().deleteAllCookies();
21             driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
22             driver.manage().timeouts().pageLoadTimeout(10, TimeUnit.SECONDS);
23             driver.get("https://www.youtube.com/");
24         } catch (Exception e) {
25             // TODO Auto-generated catch block
26             e.printStackTrace();
27         }
28     }
```

Figure3.19: Writing the Scripts

note.java | fbtest.java | Facbook.java | Grid.java | Track.java | Day1.java | Prat.java | WebDriver.class

```
    public void checkr() {
        try {
            driver.findElement(By.id("search")).sendKeys("Selenium");
        driver.findElement(By.id("search-icon-legacy")).click();
        String cururl = driver.getCurrentUrl();
        System.out.println("The current url is "+cururl);
        driver.navigate().back();
        String tit = driver.getTitle();
        System.out.println("the title is"+tit);
        driver.navigate().to("https://www.facebook.com/");
        driver.close();
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

    }

    public static void main(String[] args) {
        try {

Facbook myobj = new Facbook();
myobj.invokeBrowser();
myobj.checkr();
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

    }
```

Figure3.20: Writing the Script

First a script is created, to create a script we first create a package edurekha and under that package a class is created named Facebook.

3.2.5 CODE WALKTHROUGH

- org.openqa.selenium.WebDriver – This command is used to instantiate the web browser.
- Org.openqa.selenium.chrome.chromeDriver- It indicates that chrome driver is being instantiated.
- Org.openqa.selenium.WebElement- It is used to refer to a class that is sued to instantiate the web elements.
- org.openqa.selenium.By – It is used to refer to the by class on which the locator type is called.

OBJECT INSTALLATION

- Web Driver driver();
- Driver = new Chrome Driver();

First a variable for web driver is created which is driver. This variable is instantiated using Chrome Driver class. Now a default chrome profile will get launched.

LAUNCHING THE WEB BROWSER

- Driver.get(url)- Here a get method is used that helps in launching the application URL.

MANAGE METHODS

- Driver.manage().window().maximize()-As soon as the application URL is opened maximize() method is used to maximize the browser window.
- Driver.Manage().deleteAllcookies() – This methods enables us to delete all cookies.
- Driver.manage().timeouts().implicitlyWait(10, Timeunit.SECONDS)- As soon as the browser is launched it waits for certain seconds that is for 10 seconds.

- Driver.manage().timeouts().pageLoadTimeout(10, Timeunit.SECONDS)-This is also used to put some wait before opening the URL.

FETCH THE PAGE TITLE

- Driver.getTitle() – The getTitle() method is used to get the title of the current web page.

WEB ELEMNT INSTALLATION

- Driver.findElement(By.id("search)).sendkeys("Selenium");

Using Inspect element we will find the element. In this code we will first find the search button using inspect element.

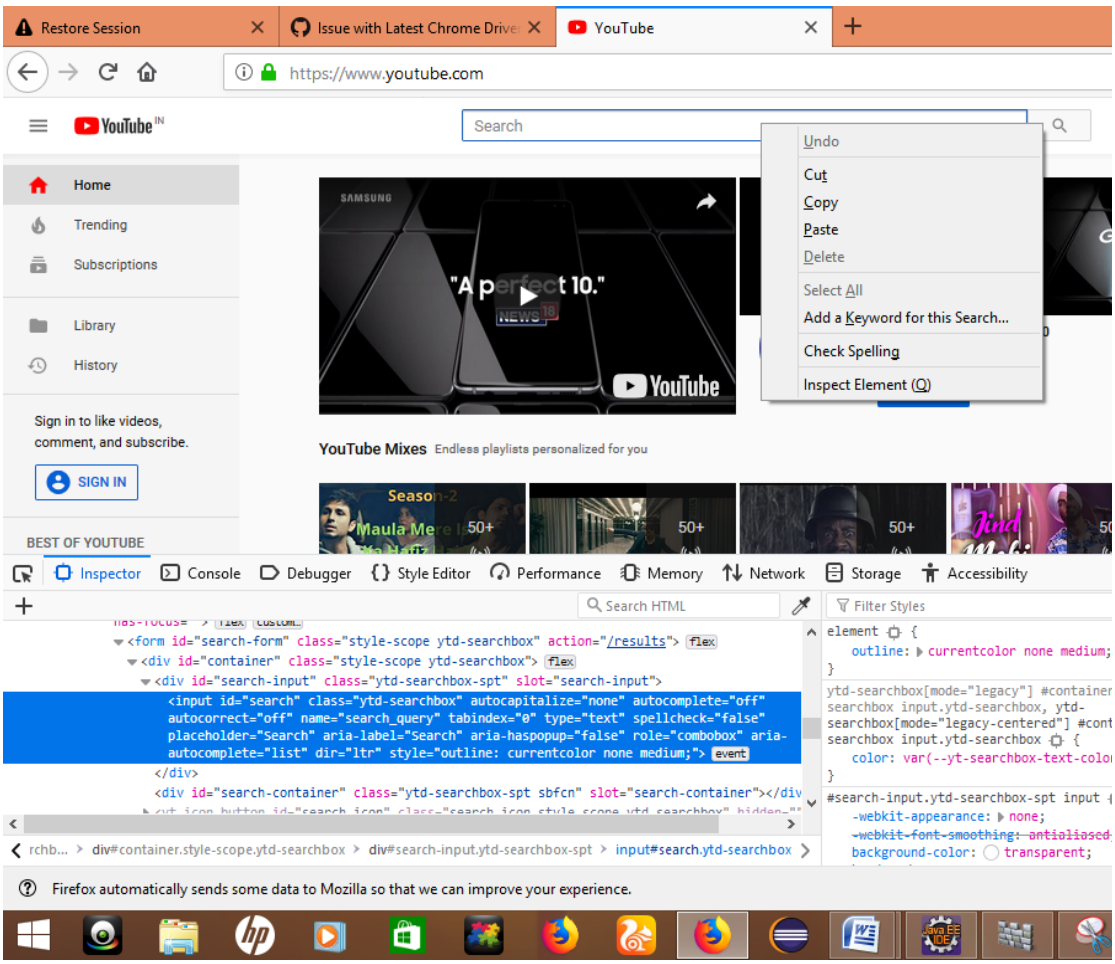


Figure3.21: Finding the element.

Using Inspect element we will find the element. In this code we will first find the search button using inspect element

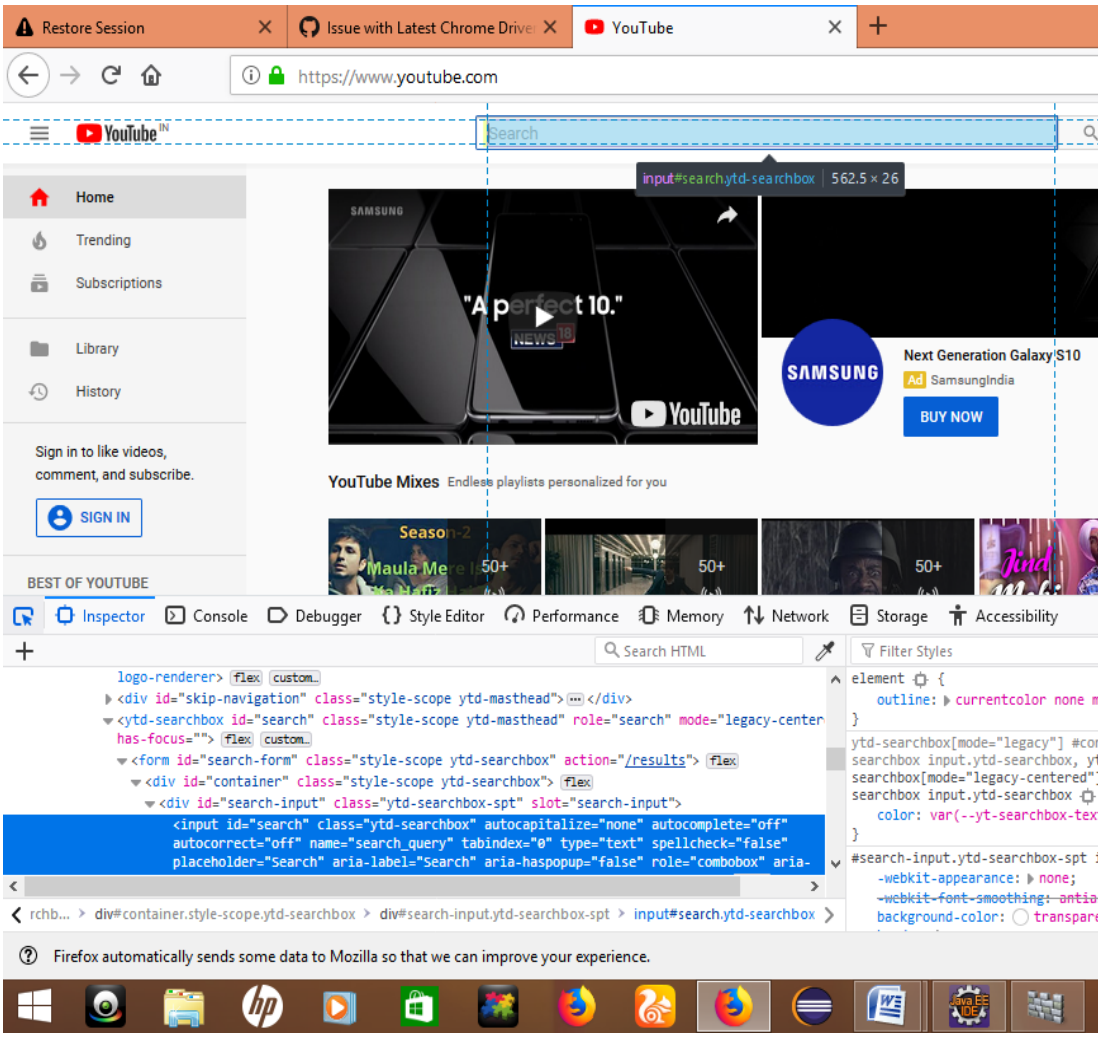


Figure3.22: Finding Xpath using inspect element.

SENDKEYS COMMAND

- Sendkeys(“selenium”)- This command is used to enter the value that is present within the parenthesis in the textbox. Now here the send keys button enters the selenium into the search box.

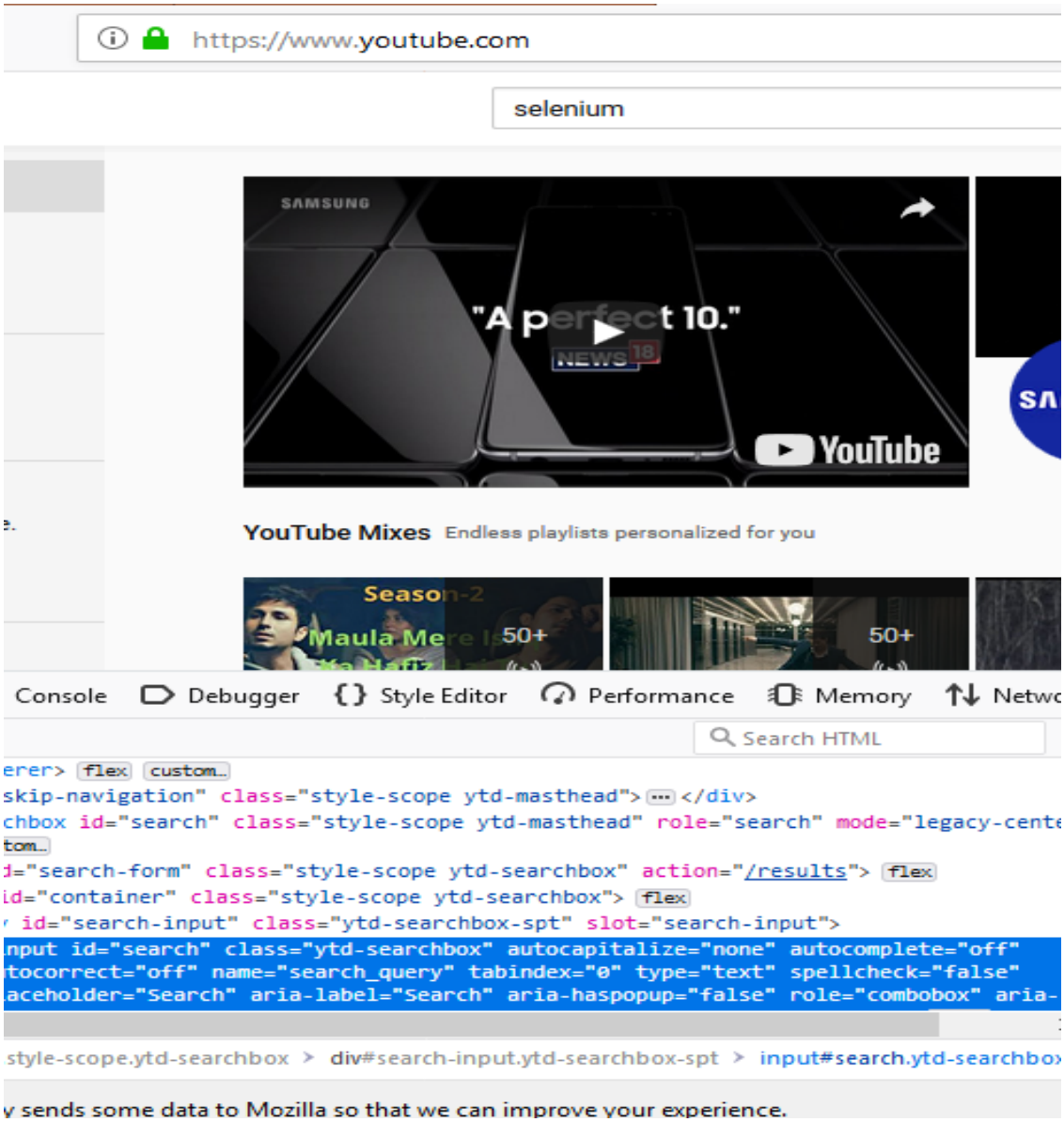


Figure3.23: Input the values

CLICK COMMAND

- driver.findElement(By.id("search-icon-legacy")).click()- This commands will first find the search icon using inspect element. After finding the search icon it will click on it using the click () command. In other words it is used to click on the web element present on the web page.

CURRENT URL

- String cururl = driver.getCurrentUrl() – This command is used to fetch the current URL of the page and store it in cururl.
- System.out.println("The current url is "+cururl) – This command is used to print the current URL.

NAVIGATE

- driver.navigate().back() – This is a navigation command that is used to navigate back to the previous page
- driver.navigate().to("https://www.facebook.com/")- This helps in navigating to the new URL from the current URL.

Figure3.24: Navigating to next URL

OBJECT CREATION

- Facbook myobj = new Facbook()- This is used to create a new object
- myobj.invokeBrowser();/myobj.checkr(); - These two are used to instantiate invokebrowser and chekr.

CLOSE COMMAND

- Driver.close()- This command is used to close the current browser window.

Now using surround with from the menu we can surround the code with try catch block.

CONCLUSION

The internship program has been a great learning experience. I got acquainted with new technologies that I had never used before. The involvement in the project gave an outlook of how things go about in a corporate environment. I learnt about various frameworks. I was involved in the live project and all over it was a good enriching experience.

REFERENCES

[1] www.seleniumhq.org/

[2]softwaretestinghelp.com/web-application-testing/

[3] www.softwaretestinghelp.com/test-automation-frameworks-selenium-tutorial-20/

[4]Brambles Project Guide.

[5] www.evoketechnologies.com/blog/test-automation-framework-design/

[6] www.seleniumhq.org/docs/02_selenium_ide.jsp

[7]chrome.google.com/webstore/detail/seleniumide/mooikfkahbdckldjjndioackbalphokd?hl=en

[8] www.guru99.com/introduction-selenuim-ide.html

[9] www.seleniumhq.org/projects/webdriver/

[10] www.seleniumhq.org/docs/03_webdriver.jsp