**AUTOMATION OF WEB APPLICATION USING A GENERIC & REUSABLE AUTOMATION FRAMEWORK**

**A PROJECT REPORT**

***Submitted by***

**S.R.DHATCHINAMOORTHI**

**R.SABARISAN**

**R.CHARUMATHI**

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

***in***

**COMPUTER SCIENCE AND ENGINEERING**

**PSNA COLLEGE OF ENGINEERING AND TECHNOLOGY, DINDIGUL.**

**ANNA UNIVERSITY: CHENNAI 600 025**

**APRIL 2016**

### BONAFIDE CERTIFICATE

### Certified that this project report *“Automation of Web Application using a generic & reusable automation framework”* is the bona-fide work of S.R.DHATCHINAMOORTHI, R.SABARISAN, R.CHARUMATHI who carried out the project work under my supervision.

### SIGNATURE SIGNATURE

### HEAD OF THE DEPARTMENT SUPERVISOR

**ABSTRACT**

Success of any web application depends on the smooth functionality of the application without any intervention. A web application containing broken web elements can create problem and dissatisfaction among the users. Now it becomes obligatory for the organization to perform validation of the web elements present in their application. This paper provides a generic and reusable automation framework to perform web elements validation. Selenium is used to support the feature of cross browser testing.

**TABLE OF CONTENTS**

**CHAPTER NO. TITLE PAGE NO.**

**ABSTRACT**

**LIST OF TABLE**

**LIST OF FIGURES**

**LIST OF SYMBOLS**

**1 INTRODUCTION**

* 1. AIM
  2. OBJECTIVE
  3. PROJECT OVERVIEW

**2 REQUIREMENT SPECIFICATION**

2.1 SYSTEM REQUIREMENT 2.2 FUNCTIONAL AND NON FUNCTIONAL  REQUIREMENT 2.3 HARDWARE AND SOFTWARE SPECIFICATION

**3 SYSTEM STUDY**

3.1 SYSTEM ANALYSIS 3.2 SYSTEM STUDY

3.3 SYSTEM FEATURES 3.4 EXISTING SYSTEM 3.5 PROPOSED SYSTEM 3.6 SYSTEM DESIGN

3.7 DATABASE

**4 SYSTEM DESIGN**

**?**

**5 SNAPSHOTS**

**AUTOMATION OF WEB APPLICATION USING A GENERIC & REUSABLE AUTOMATION FRAMEWORK**

**CHAPTER 1**

**INTRODUCTION**

* 1. **TESTING AUTOMATION**

  Test automation means using a software tool to run repeatable tests against the application to be tested. For regression testing this provides that responsiveness.

**1.2 AIM**

To implement an automated testing in order to reduce human resource and enhance reusability.

**1.3 SCOPE OF AUTOMATION**

Scope of automation is the area of our Application Under Test which will be automated. Following points help determine scope:

* Feature that are important for the business
* Scenarios which have **large amount of data**
* **Common functionalities** across applications
* Technical feasibility
* Extent to which business components are reused
* **Complexity** of test cases
* Ability to use the same test cases for cross browser testing

**1.4 OBJECTIVES**

In this project, we have to write the script for automating the functional testcases

- To reduce Testing Cost and Time.  
- To reduce Redundancy.  
- To speed up the Testing Process.  
- To help improve Quality.  
- To improve Test coverage.  
- To reduce Manual Intervention.

**1.5 OVERVIEW**

Some software testing tasks, such as extensive low-level interface regression testing, can be laborious and time consuming to do manually. In addition, a manual approach might not always be effective in finding certain classes of defects. Test automation offers a possibility to perform these types of testing effectively. Once automated tests have been developed, they can be run quickly and repeatedly. Many times, this can be a cost-effective method for regression testing of software products that have a long maintenance life. Even minor patches over the lifetime of the application can cause existing features to break which were working at an earlier point in time.

**CHAPTER 2**

**LITERATURE SURVEY**

**EXISTING SYSTEM**

**Manual testing** is the process of manually testing software for defects. It requires a tester to play the role of an end user and use most of all features of the application to ensure correct behavior. To ensure completeness of testing, the tester often follows a written test plan that leads them through a set of important test cases. For small scale engineering efforts (including prototypes), exploratory testing may be sufficient. With this informal approach, the tester does not follow any rigorous testing procedure, but rather explores the user interface of the application using as many of its features as possible, using information gained in prior tests to intuitively derive additional tests. The success of exploratory manual testing relies heavily on the domain expertise of the tester, because a lack of knowledge will lead to incompleteness in testing. One of the key advantages of an informal approach is to gain an intuitive insight to how it feels to use the application.

Large scale engineering projects that rely on manual software testing follow a more rigorous methodology in order to maximize the number of defects that can be found.

**DRAWBACKS OF EXISTING SYSTEM**

1. Manual testing is slow and costly. Because it is very labor-intensive, it takes a long time to complete tests. Increasing headcount increases cost.

2. Manual tests don’t scale well. As the complexity of the software increases, the complexity of the testing problem grows exponentially. This leads to an increase in the total time devoted to testing as well as the total cost of testing.

3. Manual testing is not consistent or repeatable. Variations in how the tests are performed are inevitable, for various reasons. One tester may approach and perform a certain test differently from another, resulting in different results on the same tests, are not being performed identically.

4. Lack of training is a common problem, although not unique to manual software testing.

5. Testing is difficult to manage. There are more unknowns and greater uncertainty in testing than in code development. Modern software development practices are well-structured, but if you don’t have sufficient structure in testing it will be difficult to manage.

**PROPOSED SYSTEM**

**SYSTEM DESIGN**

Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. It is the most crucial phase in the developments of a system. The logical system design arrived at as a result of systems analysis is converted into physical system design. Normally the design proceeds in two stages:

**Preliminary or General Design**

In the preliminary or general design, the features of the new system are specified. The costs of implementing these features and the benefits to be derived are estimated. If the project is still considered to be feasible, we move to the detailed design stage.

**Structured Or Detailed Design**

In the detailed designed stage, computer oriented work begins in earnest. At this stage, the design of the system becomes more structured. Structure design is a blue print of a computer system solution to a given problem having the same components and inter relationships among the same components as the original problem. Input, output, databases, forms, codification schemes and processing specifications are drawn up in detail.

In the design stage, the programming language and the hardware and software platform in which the new system will run are also decided. There are several tools and techniques used for describing the system design of the system. These tools and techniques are:

* Flowchart
* Data Flow Diagram (DFD)
* Data Dictionary

**CHAPTER 3**

**REQUIREMENT SPECIFICATIONS**

**FUNCTIONAL AND NON FUNCTIONAL REQUIREMENTS**

**Functional Requirement**

**A functional requirement describes***what***a software system should do.** It is basically describing the behavior of the system as it relates to the system's functionality.The functional requirement here is an appliance(hardware and software) or software that can handle the data in Enterprise Risk Integration(ERI).The system must be able to process traditional data warehouse as well as OLTP workloads .

**Non functional requirement**

The non-functional requirement elaborates a performance characteristic of the system. The nonfunctional requirements of this system are to efficient inserts, updates , deletes and other similar queries. The system must have high availability and recovery option during the occurrences of disasters. It also provide high concurrency, efficient ,storage and compression ,ease of integration with external libraries, sql support and ease of increasing the nodes.

**HARDWARE AND SOFTWARE SPECIFICATION**

**Hardware Requirements**

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirement list is often accompanied by a Hardware Compatibility List(HCL), especially in case of operating systems. An HCL lists tested, compatible and sometimes incompatible hardware devices for a particular operating system or application.

* SYSTEM : IBM-Compatible PC
* PROCESSOR : Pentium IV
* SPEED : 2.0 GHz
* MEMORY : 256 MB RAM
* HARD DISK DRIVE : 40 GB and above.

**Software Specification**

Software requirements deal with defining software resource requirements and requirements that need to be installed on a computer to provide optimal functioning of an application. These requirements are generally not included in the software installation package and need to be install separately before the software is installed.

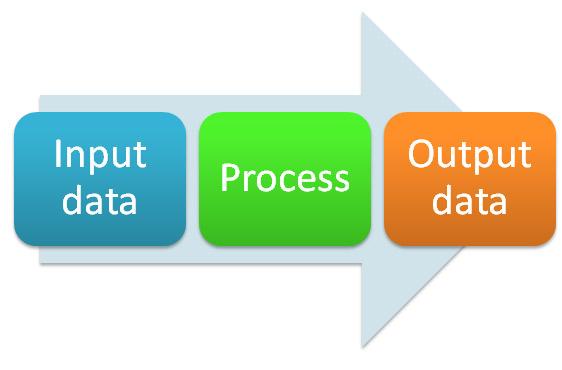
* PLATFORM : Windows 7
* TOOLS USED : Eclipse, Selenium, JDK1.8

TestNG, MS Excel

**CHAPTER 4**

**SYSTEM ANALYSIS**

Systems are created to solve problems. One can think of the systems approach as an organized way of dealing with a problem. In this dynamic world, the subject system analysis and design, mainly deals with the software development activities. A collection of components that work together to realize some objectives forms a system. Basically there are three major components in every system, namely input, processing and output.



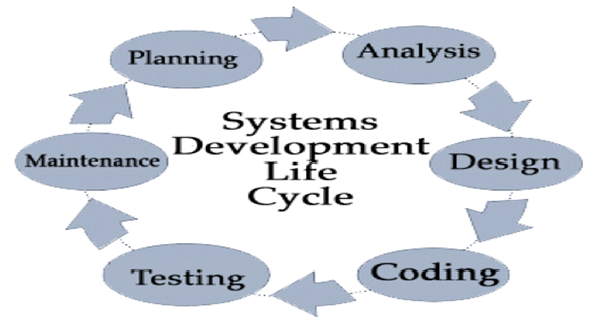
In a system the different components are connected with each other and they are interdependent. The objective of the system demands that some output is produced as a result of processing the suitable inputs. A well designed system also includes an additional element referred to as ‘control’ that provides a way feedback to achieve desired objectives of the system.

System analysis is a process of collecting factual data, understand the process involved, identifying problems and recommending feasible suggestions for improving the system functioning. This involves studying the business processes, gathering operational data, understand the information flow, finding out bottlenecks and evolving solutions for overcoming the weaknesses of the system so as to achieve the organizational goals. System analysis also includes subdividing of complex process involving the entire system, identification of data store and manual processes. The major objectives of system analysis are to find answers for each business processes: What is being done, How is it being done, Who is doing it, When is he doing it, Why is it being done and How can it be improved? It is more of a thinking process and involves the creative skills of the system analyst.

It attempts to give birth to a new efficient system that satisfies the current needs of the user and has scope for future growth within the organizational constraints. The result of this process is a logical system design, System analysis is an iterative process that continues until a preferred and acceptable solution emerges.

**3.1.1 System Life Cycle**

System life cycle is an organizational process of developing and maintaining systems. It helps in establishing a system project plan, because it gives overall list of processes and sub-processes required for developing a system. System development life cycle means combination of various activities. In other words we can stay that various activities put together are referred as system development life cycle. In the system analysis and design terminology, the system development life cycle also means software development life cycle. The different phase of system development life cycle is shown in this diagram.



**SYSTEM STUDY**

**JDK 1.8**

The Java Development Kit(JDK) is an implementation of either one of the Java SE, Java EE or Java ME platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, Mac OS X or Windows.

The JDK includes a private JVM and a few other resources to finish the recipe to a Java Application. Since the introduction of the Java Platform, it has been by far most widely used Software Development Kit(SDK). On 17 November 2006, sun announced that it would be released under the GNU General Public License(GPL), thus making it free software.

**Eclipse Luna**

In computer programming, Eclipse is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications. By means of various plug-ins, Eclipse may also be used to develop applications in other programming languages: Ada, ABAP, C, C++, COBOL, Fortran, Haskell, JavaScript, Lasso, Lua, Natural, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Scala, Clojure, Groovy, Scheme, and Erlang. It can also be used to develop packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

The initial codebase originated from IBM Visual Age. The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.

Released under the terms of the Eclipse Public License, Eclipse SDK is free and open source software (although it is incompatible with the GNU General Public License). It was one of the first IDEs to run under GNU Class path and it runs without problems under Iced Tea.

**SYSTEM FEATURES**

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere” (WORA) meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to byte code that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2015, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since merged into Oracle Corporation) and released in 1995 as a core component of Sun Microsystems’ Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

Five primary goals in the creation of Java language :

* Simple, object-oriented and familiar
* Robust and secure
* Architecture-neutral and portable
* High performance
* Interpreted, threaded, and dynamic

Hyper Text Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. It is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent empty elements and so are unpaired, for example < img >. The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags).

Web browsers can read HTML files and compose them into visible or audible web pages. Browsers do not display the HTML tags and scripts, but use them to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

Web browsers can also refer to Cascading Style Sheets (CSS) to define the look and layout of text and other material. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicit presentational HTML.

In computer programming, Eclipse is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications. By means of various plug-ins , Eclipse may also be used to develop applications in other programming languages: Ada, ABAP, C, C++, COBOL, Fortran, Haskell, JavaScript, Lasso, Lua, Natural, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Scala, Clojure, Groovy, Scheme, and Erlang. It can also be used to develop packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

The initial codebase originated from IBM Visual Age. The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.

Released under the terms of the Eclipse Public License, Eclipse SDK is free and open source software (although it is incompatible with the GNU General Public License). It was one of the first IDEs to run under GNU Class path and it runs without problems under Iced Tea.

**TestNG**

TestNG is a testing framework designed to simplify a broad range of testing needs, from unit testing (testing a class in isolation of the others) to integration testing (testing entire systems made of several classes, several packages and even several external frameworks, such as application servers).

Writing a test is typically a three-step process:

* Write the business logic of your test and insert TestNG annotations in your code.
* Add the information about your test (e.g. the class name, the groups you wish to run, etc...) in a testng.xml file or in build.xml.
* Run TestNG

You can find a quick example on the Welcome page.

The concepts used in this documentation are as follows:

* A suite is represented by one XML file. It can contain one or more tests and is defined by the <suite> tag.
* A test is represented by <test> and can contain one or more TestNG classes.
* A TestNG class is a Java class that contains at least one TestNG annotation. It is represented by the <class> tag and can contain one or more test methods.
* A test method is a Java method annotated by @Test in your source.

A TestNG test can be configured by @BeforeXXX and @AfterXXX annotations which allows to perform some Java logic before and after a certain point, these points being either of the items listed above.

The rest of this manual will explain the following:

* A list of all the annotations with a brief explanation. This will give you an idea of the various functionalities offered by TestNG but you will probably want to consult the section dedicated to each of these annotations to learn the details.
* A description of the testng.xml file, its syntax and what you can specify in it.
* A detailed list of the various features and how to use them with a combination of annotations and testng.xml.

**Apache POI**

**Apache POI**, a project run by the Apache Software Foundation, and previously a sub-project of the Jakarta Project, provides pure Java libraries for reading and writing files in Microsoft Office formats, such as Word, PowerPoint and Excel.

**WEB ELEMENT**

Selenium sees everything on the page like Textbox,button,link,dropdown as a web element.For simple ones or for most of them,web element type is used.For complex ones, separate classes are used.Web Element Interface:

**IDENTIFYING WEB ELEMENT IN WEBPAGE**

For many selenium commands, a target is required.Target identifies a web element uniquely in the content of Web Application.Target consists of the locating Strategy and format like

### SELENIUM

Selenium is a set of different software tools each with a different approach to supporting test automation. Most Selenium QA Engineers focus on the one or two tools that most meet the needs of their project, however learning all the tools will give you many different options for approaching different test automation problems. The entire suite of tools results in a rich set of testing functions specifically geared to the needs of testing of web applications of all types. These operations are highly flexible, allowing many options for locating UI elements and comparing expected test results against actual application behavior. One of Selenium’s key features is the support for executing one’s tests on multiple browser platforms.

Selenium is a portable software testing framework for web application.It provides a record/playback tool for authoring tests without learning a test scripting language.It also provides a test domain-specific language to write tests in a number of popular programming language including java,c#,groovy,perl,PHP,python and ruby.

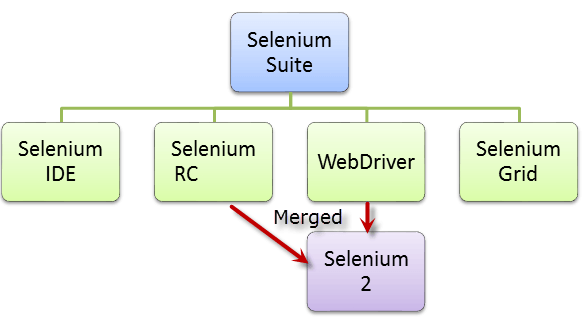
The test can then be run against most modern web browsers. Selenium deploys on windows, linux and Macintosh platforms. It is open-source software, released under the Apache 2.0 licensed, and can be downloaded and used without charge.

**WHY THE NAME SELENIUM**

The Name Selenium came from a joke which Jason cracked one time to his team. Another automated testing framework was popular during Selenium's development, and it was by the company called Mercury Interactive (yes, the company who originally made QTP before it was acquired by HP). Since Selenium is a well-known antidote for Mercury poisoning, Jason suggested that name. His teammates took it, and so that is how we got to call this framework up to the present.

**COMPONENTS OF SELENIUM**

Selenium had four components earlier, later its reduced to three by combining Selenium RC and Selenium Web Driver.

****

* Selenium Integrated Development Environment(IDE)
* Selenium RC
* Selenium Webdriver
* Selenium Grid

Selenium RC and Web Driver are merged into a single framework to form **Selenium 2**. Selenium 1, refers to Selenium RC.

**Selenium IDE**

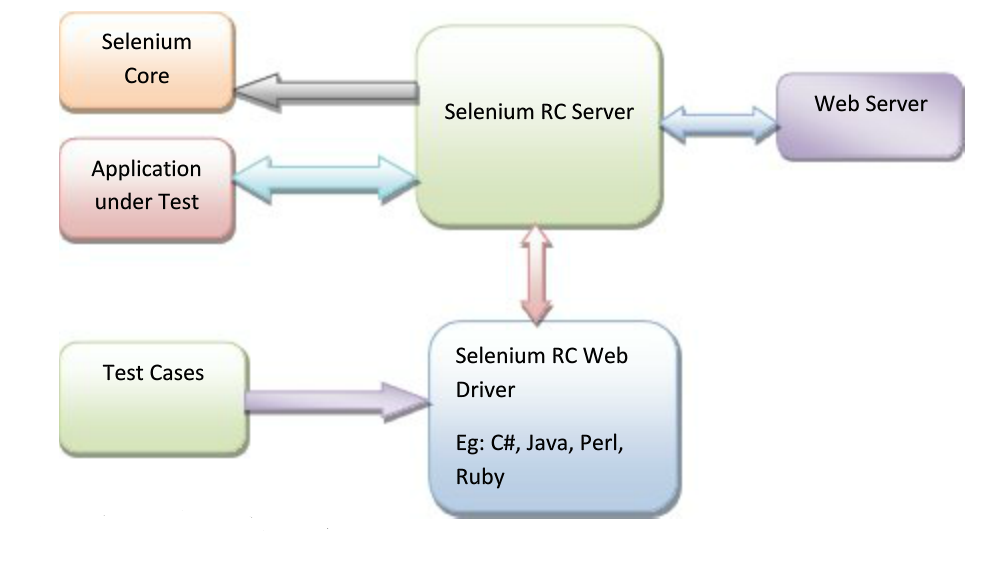
Selenium IDE (Integrated Development Environment) is a prototyping tool for building test scripts. It is a Firefox plugin and provides an easy-to-use interface for developing automated tests. Selenium IDE has a recording feature, which records user actions as they are performed and then exports them as a reusable script in one of many programming languages that can be later executed.

**Selenium Remote Control (Selenium RC)**

Selenium RC was the **flagship testing framework** of the whole Selenium project for a long time. This is the first automated web testing tool that **allowed users to use a programming language they prefer**

**Selenium RC Architecture**

* Selenium RC interacts with Browsers with the help of Selenium RC server and RC server communicates with the help of Selenium core java scripts commands.
* RC server receives the selenium commands called Selenese from test program using simple HTTP GET/POST requests.



**Selenium RC Component**

Selenium RC consists of the below mentioned components

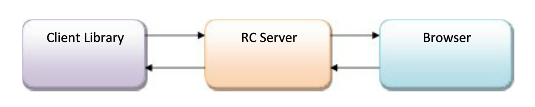
* RC Server
* Client Library
* Selenium core
* Application Under test.

**RC Server**

Selenium RC Server interprets selenium commands.[Selenium RC server is a collection of request in the form of JavaScript, so that your browser can understand].It launches and kills the browser session. It inject selenium core into the browser. It also passes the selenium commands to the browser using selenium core java script commands. RC server communicates using HTTP, so any programming language can be used.

**Selenium Client Library**

Client library is a language which is used for writing selenium code, it can be java library, C# library, python library etc. We can download client library from selenium official Website.



**Selenium Core**

It is a request getting passed from RC Server and inject into browser in the form of JavaScript. Selenium RC injects selenium core into browser when you open the browser programmatically.

**Application Under Test**

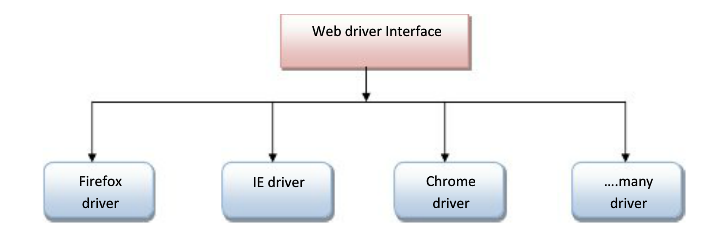
AUT will be web application. Selenium can automate any web application regardless of any technology it is built in. Selenium cannot automate Desktop application.

**Selenium-Web Driver**

The Web Driver proves itself to be **better than both Selenium IDE and Selenium RC** in many aspects. It implements a more modern and stable approach in automating the browser's actions. Web Driver, unlike Selenium RC, does not rely on JavaScript for automation. **It controls the browser by directly communicating to it. The** supported languages are the same as those in Selenium RC.

* Java
* C#
* PHP
* Python
* Perl
* Ruby

When a class implements an interface, it provide the methods defined in that interface with that, We can use same method names with different browsers.



**Selenium Grid**

Selenium Grid is a tool **used together with Selenium RC to run parallel tests** across different machines and different browsers all at the same time. Parallel execution means running multiple tests at once.

**Features:**

* Enables **simultaneous running of tests**in **multiple browsers and environments.**
* **Saves time**enormously.
* Utilizes the **hub-and-nodes** concept. The hub acts as a central source of Selenium commands to each node connected to it.

**Choosing Selenium Tool**

**Selenium IDE**

* In order to learn about concepts on automated testing and Selenium, including the
* Selenese commands such as type, open, clickAndWait, assert, verify, etc.
* Locators such as id, name, xpath, css selector, etc.
* Executing customized JavaScript code using runScript
* Exporting test cases in various formats.
* To create tests with little or no prior knowledge in programming.
* To create simple test cases and test suites that you can export later to RC or WebDriver.
* To test a web application against Firefox only.

**Selenium RC**

* To design a test using a more expressive language than Selenese
* To run your test against different browsers (except HtmlUnit) on different operating systems.
* To deploy your tests across multiple environments using Selenium Grid.
* To test your application against a new browser that supports JavaScript.
* To test web applications with complex AJAX-based scenarios.

**WebDriver**

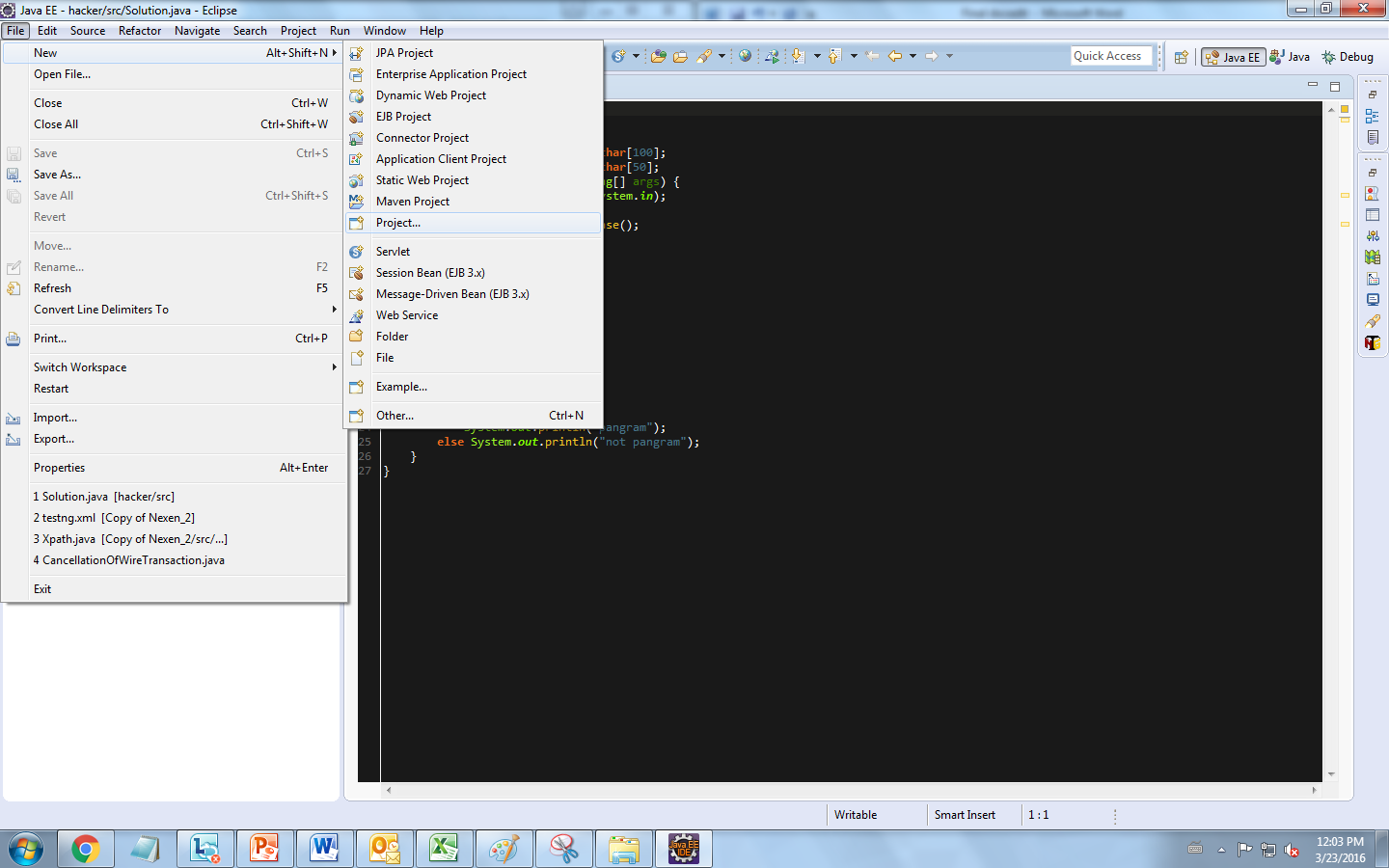
* To use a certain programming language in designing your test case.
* To test applications that are rich in AJAX-based functionalities.
* To execute tests on the HtmlUnit browser.
* To create customized test results.

**Selenium Grid**

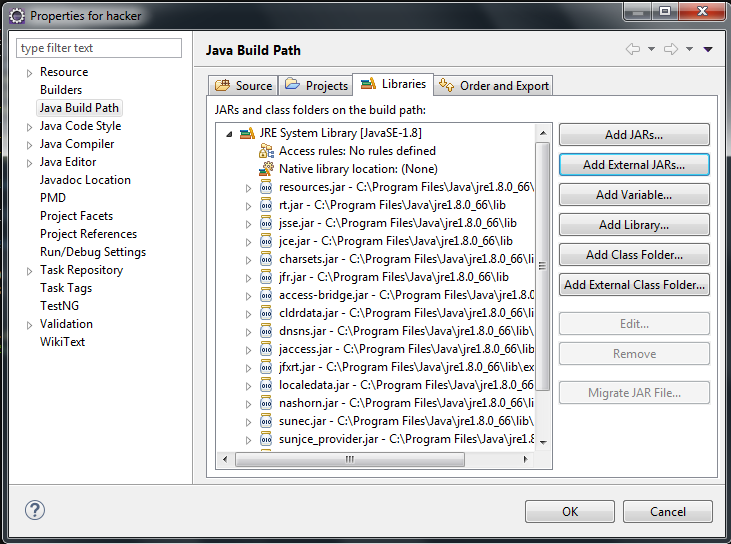
* To run your Selenium RC scripts in multiple browsers and operating systems simultaneously.
* To run a huge test suite, that need to complete in soonest time possible.

**Selenium Configuration With Java**

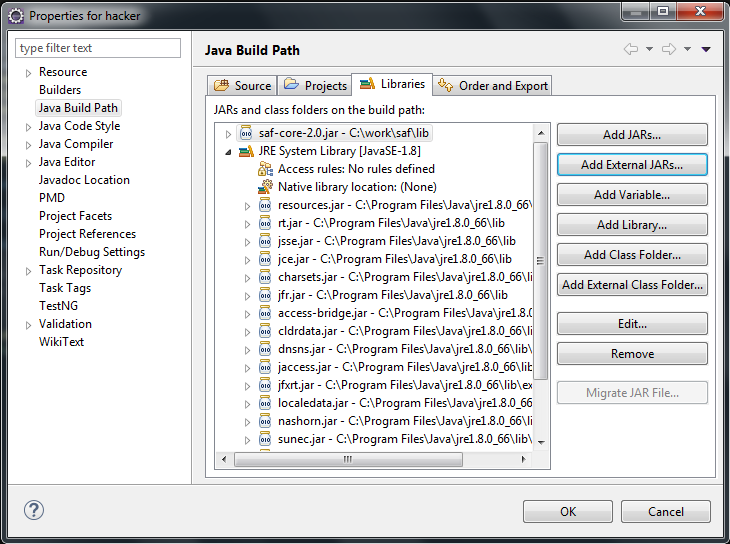
* Eclipse-It is an editor, where we can write scripts.
* JDK/JRE-Java Runtime Environment which is used to open Eclipse.
* Selenium Standalone jar file-used for the combination 9of Selenium RC/Webdriver/Grid.
* Open Eclipse.
* Go to File🡪new 🡪Java Project.



* Enter Projectname.
* Projectname🡪JRE System Library🡪Rightclick.
* Select Build path🡪Configure Build path.
* Now click on ” Add External jar” button and browse for standalone server window.

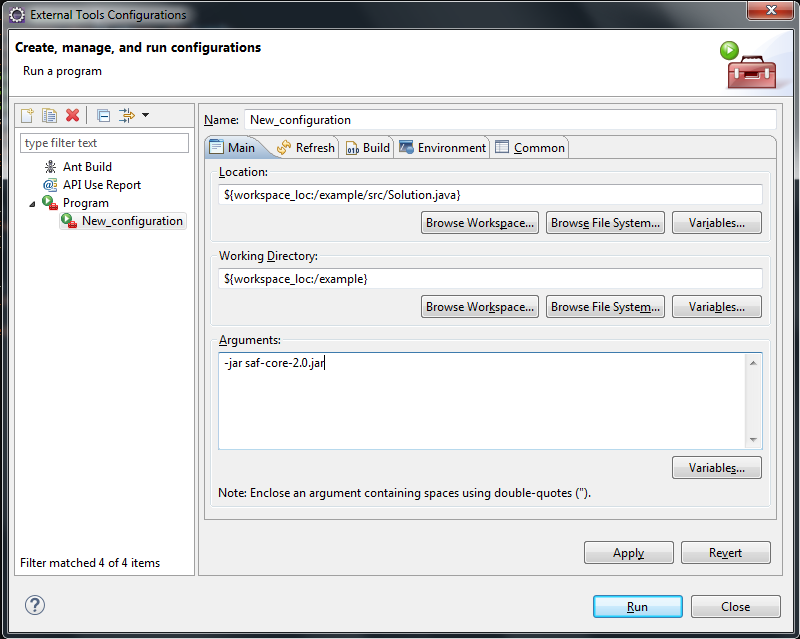


* Now jar files will be added.



Upto this step It is enough for Selenium web driver but for Selenium RC we needed to do one more step.

* Click on Run Menu🡪External tools🡪External toolConfiguration.
* Double Click on Programs.
* Provide below details and click on Run.
* Specify Location of Java exe file, Working Directory, Arguments.
* Click on Apply and Click Run Button



* Once Click on Run-Console wiil be opened-Launching Standalone Server…

With This Selenium Configuration ended.

**Web Elements**

* Selenium sees everything on the page like Textbox, button, link, dropdown as a Web element.
* For simple ones or for most of them, Web element type is used. E.g.:-
* Textbox, button, link.
* For complex ones, separate classes are used. E.g.:- Dropdown,alerts/pop-ups,frames.
* Web Element Interface: We can use same method names for Web Element with different browsers. E.g.:- Click(), getAttribute(), SendKeys().

**Identifying the web elements in Web Page**

* For many Selenium commands, a target is required.
* Target identifies a web element uniquely in the content of Web Application.
* Target consists of the locating strategy and format like ***By.LocatorStrategyType(“Value”)***
* Whenever we use findElement(), it will always finds the first web element using the given locating strategy.
* findElements() - it returns all the elements matching to the given locating strategy. E.g.: It can be used in tables to get the cell value.

**Locating Strategy**

**1. Locating By Id**

* Locates the first element with matching Id attribute.
* *driver.findElement(By.id(“UserId”)).sendKeys(“Username”);*

**2. Locating By Name**

* Locates the first element with matching name attribute.
* *driver.findElement(By.name(“pwd”)).sendKeys(“Password”);*

**3. Locating By LinkText**

* Locates the first hyperlink with matching link attribute (it use full link text).
* This is a simple method of locating a hyperlink in your web page by using the text of the link.
* *driver.findElement(By.linkText(“Sign In”)).Click();*

**4. Locating By PartialLinkText**

* Locates the first hyperlink which contains specified link text.
* *driver.findElement(By.PartialLinkText(“Can’t access”)).Click();*

**5. Locating By Xpath**

* Locates the first element as specified by Xpath Strategy
* *driver.findElement(By.Xpath(“//input[@placeholder=’Email’]”)).sendKeys(“UserName”);*

**6. Locating By CssSelector**

* Locates the first element as specified by CSS selector strategy.
* *driver.findElement(By.cssSelector(“input.input\_submit”)).Click();*

**7. Locating By TagName**

* Useful to get all elements with a given tag.
* *driver.findElement(By.tagName(“a”)).Clicksize();*

**8. Locating By ClassName**

* Useful to get all elements with a given class or display style.
* *driver.findElement(By.ClassName(“input\_text”)).size();*

**Xpath**

XPath is designed to allow the navigation of XML documents, with the purpose of selecting individual elements, attributes, or some other part of an XML document for specific processing.

**XML**  
The Extensible Markup Language (XML) is the context in which the XML Path Language, XPath, exists.

XML provides a standard syntax for the markup of data and documents.

XML documents contain one or more elements. If an element contains content, whether other elements or text, then it must have a start tag and an end tag. The text contained between the start tag and the end tag is the element’s content.

<Element> *//Start tag*

Element content goes here.*//Element Content*

</Element>*//End Tag*

An element may have one or more attributes, which will provide additional information  
about the element type or its content.  
  
Below is the sample XML:

<?xml version='1.0'?>

<**Catalog**>

<**Book**>

<**Title**>Selenium</**Title**>

<**Author**>Selenium Automation</**Author**>

</**Book**>

</**Catalog**>

It can also be written as:

<?xml version='1.0'?>

<**Catalog**>

<**Book** Title=" Selenium " Author=" Selenium Automation ">

</**Book**>

</**Catalog**>

XPath can be viewed as a way to navigate round XML documents. Thus XPath has similarities to a set of street directions.

When you need to search for address, you should know what is your starting point to reach your destination.

In Xpath the starting point is called the context node.

**Absolute XPath**  
Absolute XPath starts with the root node or a forward slash (/).  
The advantage of using absolute is, it identifies the element very fast.  
Disadvantage here is, if anything goes wrong or some other tag added in between, then this path will no longer works.

**Example:**  
If the Path we defined as  
1. html/head/body/table/tbody/tr/th

If there is a tag that has added between body and table as below  
2. html/head/body/form/table/tbody/tr/th

The first path will not work as 'form' tag added in between

**Relative Xpath**  
A relative xpath is one where the path starts from the node of your choice - it doesn't need to start from the root node.

It starts with Double forward slash(//)

**Syntax:**  
//table/tbody/tr/th

Advantage of using relative xpath is, you don't need to mention the long xpath, you can start from the middle or in between.

In our project, we have automated the …. website’s launch and checked for its features and functionalities. We achieved this through the public methods and classes under one common package. The automation process included selecting the hyperlink, entering the text, clicking the elements, picking the dates from date-picker, mouse motions and mouse event handling, getting element text, switching to other frames, etc.,

**Project automation flow proceeds as follows:-**

* When the .xml file is made to run using the TestNG suite, it triggers the driver.
* WebDriver object invokes the mentioned browser and launches the URL.
* Laterally, the .xml file inputs values from the spreadsheet.
* Spreadsheet contains the values that calls the methods needed for the specified operation.
* Based on the inputs from the spreadsheets, action is performed by the driver.
* For e.g.:- When the driver launches the……. url, the spreadsheet gives in the login command for which the login operation is performed by the driver, by calling the respective classes and methods.
* The following operations proceed in the same way.

**CONCLUSION AND FUTURE SCOPE**

This project aims in implementing a tool that reduces human effort in doing repeatable tasks. This can be applied, not only for testing the web applications, but also to all types of tasks which are least complex and require a large amount of human labor and less human knowledge.