410247: Laboratory Practice IV

410245(D)Software Testing and Quality Assurance

Group 1:

Practical No 1:

Problem statement: Write TEST Scenario for Gmail Login Page.

Objectives: To Write TEST Scenario for Gmail Login Page.

Theory:

Test Cases of a Login Page (Test Scenarios Login Page):

Following are the test cases for User Login Page. The list consists of both Positive and Negative test scenarios of login page along with UI test cases, Security test cases and so on.

UI Test Scenarios of Login Page

1. Verify that the login screen contains elements such as Username, Password, Sign in button, Remember password check box, Forgot password link, and create an account link.

2. Verify that all the fields such as Username, Password has a valid placeholder

3. Verify whether all the text boxes have a minimum and maximum length.

4. Verify that the labels float upward when the text field is in focus or filled (In case of the floating label).

5. Verify to see if the font style and size of the labels, as well as the text on each object, are clearly visible.

6. Verify that the application‘s user interface (UI) is responsive, so it will adapt to different screen resolutions and devices.

7. Verify the login page and all the fields in the login page are displaying without any break in different browsers.

Functional Test Scenarios of Login Page

1. Verify that cursor is focused on the ―Username‖ text box on the page load (login page)

2. Verify that tab functionality is working properly or not

3. Verify that Enter/Tab key works as a substitute for the Sign-in button

4. Verify that the User is able to Login with Valid Credentials

5. Verify that the User is not able to Login with an invalid Username and invalid Password

6. Verify that the User is not able to Login with a Valid Username and invalid Password

7. Verify that the User is not able to log in with an invalid Username and Valid Password

8. Verify that the User is not able to log in with a blank Username or Password

9. Verify that the User is not able to Login with inactive credentials

10. Verify that the reset button clears the data from all the text boxes in the login form

11. Verify that the login credentials, mainly password stores in a database in an encrypted format

12. Verify that clicking on the browser back button after successful login should not take the User to log out mode.

13. Verify that validation message is displayed in the case when User leaves Username or Password as blank

14. Verify that validation message is displayed in case of exceeding the character limit of the Username and Password fields

15. Verify that validation message is displayed in case of entering special character in the Username and password fields

16. Verify that the ―Keep me logged in‖ checkbox is unselected by default (depends on business logic, it may be selected or unselected)

17. Verify that the timeout of the login session (Session Timeout)

18. Verify that the logout link is redirected to login/home page

19. Verify that User is redirected to appropriate page after successful login

20. Verify that the User is redirected to the Forgot password page when clicking on the Forgot Password link

21. Verify that the User is redirected to the Create an account page when clicking on the Signup / Create an account link

22. Verify that the User should be able to login with the new password after changing the password

23. Verify that the user should not be able to login with the old password after changing the password

24. Verify that spaces should not be allowed before any password characters attempted

25. Verify whether the user is still logged in after a series of actions such as sign-in, close the browser, and reopen the application.

26. Verify that the ways to retrieve the password if the user forgets the password

Non-functional Security Test Cases for Login Page

1. Verify that clicking on the browser back button after successful logout should not take the User to a logged-in mode

2. Verify that there is a limit on the total number of unsuccessful login attempts (No. of invalid attempts should be based on business logic. Based on the business logic, User will be asked to enter the captcha and try again or user will be blocked)

3. Verify that the password is in encrypted form (masked format) when entered in the password field.

4. Verify the password can be copy-pasted. System shouldn‘t allow users to copy paste password.

5. Verify that encrypted characters in the —Password‖ field should not allow deciphering if copied

6. Verify that the —Remember password‖ checkbox is unselected by default (depends on business logic, it may be selected or unselected).

7. Verify whether the login form is revealing any security information by viewing the page source

8. Verify that the login page is vulnerable to SQL injection.

9. Verify whether Cross-site scripting (XSS) vulnerability works on a login page. XSS vulnerability may be used by hackers to bypass access controls.

Performance Test Cases for Login Page

Verify that how much time the application is taking to load the home page after entering the valid user name and password in the login page.

Test Cases for CAPTCHA & Cookies (If there is a CAPTCHA on the login page)

1. Verify that whether there is a client-side validation when the User doesn‘t enter the CAPTCHA

2. Verify that the refresh link of CAPTCHA is generating the new CAPTCHA

3. Verify that the CAPTCHA is case sensitive

4. Verify whether the CAPTCHA has audio support to listen

5. Verify whether virtual keyboard is available and working properly to enter login credentials in case of banking applications.

6. Verify two-way authentication through OTP is working properly incase of banking applications.

7. Verify SSL certificate is implemented or not

8. Verify that the user is able to login when the browser cookies are cleared. When the cookies are cleared, system should not allow user to login automatically.

9. Verify the login functionality when the browser cookies are turned off.

Conclusion: Hence, we have learned to write the test cases for Gmail login page.

Assignment No:2

Problem statement:- Write Test cases in excel sheet for Social Media application or website .

Objectives:
To Write Test cases in excel sheet for Social Media application or website .

Theory:

What are Test Cases?
A test case is a test scenario to test functionality with a different set of input and parameters. We test the expected Result of the test case with the expected one. The test case is marked as passed if the output matches and is marked as failed if the output of expected and doesn‘t match. Test cases can be executed manually or via automation.
Test cases are maintained in test management tools like Jira, HP QC, and others. Let‘s look at the main components of test cases:

- Test Case ID
- Test Case Description
- Assumptions
- Test Data
- Pre-Condition
- Test Steps
- Expected Test Result
- Actual Test Result
- Status Pass/Fail
- Comments

Types of Test Cases

There are different test cases based on the testing methodologies you are following in your project. Some are Functional, API, Performance, Security, Usability, UI, Database, and Unit Test Cases. Functional Test Cases are the ones that are written to test every functionality of the application against user requirements defined by stakeholders. API Test Cases are the ones that are executed before UI is developed for functionality. We test various APIs against different sets of inputs and conditions. Performance and Security Test cases are used to test the application‘s load capacity and vulnerabilities, respectively.
Usability Test Cases are executed to evaluate how user-friendly is our application. Database test cases are mostly SQL or No SQL-based testing. Unit test cases are written by developers to test their code effectively.

How to Write Test Cases in an Excel Sheet

For writing test cases in excel you need to make an excel sheet. There is no specific template

for writing test cases that have 10 different columns – Test Case ID, Test Case Description, Assumptions, Test Data. Pre-Condition, Test Steps, Expected Result, Actual Result, Status, and Comments. Suppose you want to write a test case to test a login functionality.

Test Case ID: Test Case ID will be the test case number of story number in JIRA.

Test Case Description: The description will be a short description of the functionality.

Assumptions: Assumptions should be mentioned in the assumptions columns if any.

Test Data: Test Data is the data with which you are performing the testing.

Pre-condition: Pre-condition should be anything that is done before the execution of the test case.

Test Steps: In Test Steps, you must mention steps like Login to application, Enter Username, and password, click the login button, Verify the page redirects to splash page.

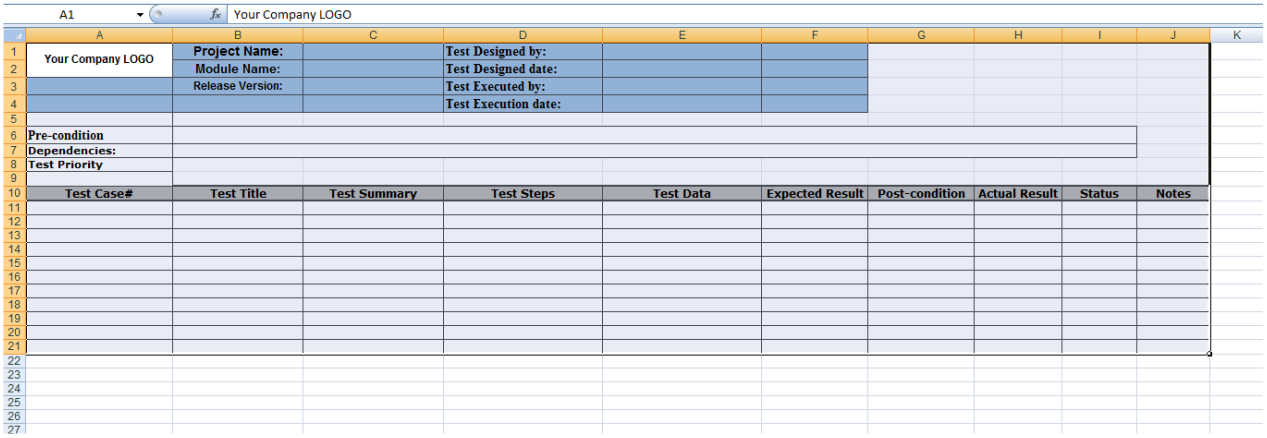
Expected Result: The expected Result should indicate the behavior of the application after the execution of the test case. For successful Login expected Result would be a redirection to the splash page.

Actual Result: The actual Result indicates the actual behaviour of the application on the execution of the test case.

Status: Status can be marked as passed or fail depending on the actual Result.

Comments: Last is the optional comment.

Example :
Link : https://www.softwaretestinghelp.com/test-case-template-examples/



Below are the Test Cases for the same:

Test Scenario ID	Login-1	Test Case ID	Login-1A
Test Case Description	Login – Positive test case	Test Priority	High
Pre-Requisite	A valid user account	Post-Requisite	NA

Test Execution Steps:

S.No	Action	Inputs	Expected Output	Actual Output	Test Browser	Test Result	Test Comments
1	Launch application	https://www.facebook.com/	Facebook home	Facebook home	IE-11	Pass	[Priya 10/17/2017 11:44 AM]: Launch successful
2	Enter correct Email & Password and hit login button	Email id : test@xyz.com Password: ******	Login success	Login success	IE-11	Pass	[Priya 10/17/2017 11:45 AM]: Login successful

Test Scenario ID	Login-1	Test Case ID	Login-1B
Test Case Description	Login – Negative test case	Test Priority	High
Pre-Requisite	NA	Post-Requisite	NA

Test Execution Steps:

S.No	Action	Inputs	Expected Output	Actual Output	Test Browser	Test Result	Test Comments
1	Launch application	https://www.facebook.com/	Facebook home	Facebook home	IE-11	Pass	[Priya 10/17/2017 11:44 AM]: Launch successful
2	Enter invalid Email & any Password and hit login button	Email id : invalid@xyz.com Password: ******	The email address or phone number that you've entered doesn't match any account. Sign up for an account.	The email address or phone number that you've entered doesn't match any account. Sign up for an account.	IE-11	Pass	[Priya 10/17/2017 11:45 AM]: Invalid login attempt stopped

Conclusion: we have studied how to Write Test cases in excel sheet for Social Media application.

Assignment No: 3

Problem statement: Create Defect Report for bug application

Objectives:
To Create Defect Report for bug application.

Theory:

Qualities of a Good Software Bug Report
Anyone can write a Bug report. But not everyone can write an effective Bug report. You should be able to distinguish between an average bug report and a good bug report.

How to distinguish between a good and bad Bug Report? It‘s very simple, apply the following characteristics and techniques to report a bug.

Characteristics and Techniques

1) **Having a clearly specified Bug Number:** Always assign a unique number to each bug report. This, in turn, will help you identify the bug record. If you are using any automated bug-reporting tool then this unique number will be generated automatically each time you report a bug. Note the number and a brief description of each bug that you reported.

2) **Reproducible:** If your bug is not reproducible, then it will never get fixed. You should clearly mention the steps to reproduce the bug. Do not assume or skip any reproducing steps. The bug which is described Step by step is easy to reproduce and fix.

3) **Be Specific:** Do not write an essay about the problem. Be Specific and to the point. Try to summarize the problem in minimum words yet in an effective way. Do not combine multiple problems even if they seem to be similar. Write different reports for each problem.

How To Report A Bug?

Use the following simple Bug report template:
This is a simple Bug report format. It may vary depending upon the Bug report tool that you are using. If you are writing a bug report manually then some fields need to be mentioned specifically like the Bug number – which should be assigned manually.

Reporter: Your name and email address.
Product: In which product you found this bug.
Version: The product version, if any.
Component: These are the major sub-modules of the product.
Platform: Mention the hardware platform where you found this bug. The various platforms like _PC‘, _MAC‘, _HP‘, _Sun‘ etc.

Operating system: Mention all the operating systems where you found the bug. Operating systems like Windows, Linux, Unix, SunOS, and Mac OS. Also, mention the different OS versions like Windows NT, Windows 2000, Windows XP etc, if applicable.

Priority: When should a bug be fixed? Priority is generally set from P1 to P5. P1 as —"fix the bug with the highest priority" and P5 as "Fix when time permits".

Severity: This describes the impact of the bug.

Types of Severity:

- **Blocker:** No further testing work can be done.
- **Critical:** Application crash, Loss of data.
- **Major:** Major loss of function.
- **Minor:** Minor loss of function.
- **Trivial:** Some UI enhancements.
- **Enhancement:** Request for a new feature or some enhancement in the existing one.

- **Application testing scenario :** Let's assume in your application you want to create a new user with his/her information, for that you need to logon into the application and navigate to USERS menu > New User, then enter all the details in the User form like, First Name, Last Name, Age, Address, Phone etc. Once you enter all these need to click on SAVE button in order to save the user and you can see a success message saying, —"New User has been created successfully".
  Now you entered into your application by logging in and navigate to USERS menu > New user, entered all the information and clicked on SAVE button and now the application crashed and you can see one error page on the screen, now you would like to report this BUG.

Now here is how we can report bug for above scenario:

- **Bug Name:** Application crash on clicking the SAVE button while creating a new user.
  Bug ID: The BUG Tracking tool will automatically create it once you save this.
  Area Path: USERS menu > New Users
  Build Number:/Version Number 5.0.1
  Severity: HIGH
  Priority: HIGH (High/Medium/Low) (High/Medium/Low)

- **Assigned By:** Your Name **To:** Developer-X
  Created On: Date
  Reason: Defect

- **Status:** New/Open/Active – Depends on the Tool you are using
  Environment: Windows 2003/SQL Server 2005

- **Description:** Application crash on clicking the SAVE button while creating a new user, hence unable to create a new user in the application.

Steps To Reproduce:

1. Logon into the application
2. Navigate to the Users Menu > New User
3. Filled all the fields
4. Clicked on 'Save' button
5. Seen an error page —"ORA1090 Exception: Insert values Error…"
6. See the attached logs for more information
7. And also see the attached screenshot of the error page

- **Expected:** On clicking SAVE button should be prompted to a success message —"New User has been created successfully".

- Save the defect/bug in the BUG TRACKING TOOL.

Conclusion: Hence, we have successfully studied concept of how to write defect report for the application.

Assignment No: 4

Problem Statement: Installation of Selenium grid and selenium Web driver java eclipse (automation tools).

Objectives: To Install Selenium grid and selenium Web driver java eclipse (automation tools).

Theory:
What Is Selenium Grid?

Selenium Grid is a part of the Selenium Suite that specializes in running multiple tests across different browsers, operating systems, and machines in parallel. It is achieved by routing the commands of remote browser instances where a server acts as a hub. A user needs to configure the remote server in order to execute the tests. Selenium Grid has 2 versions – the older Grid 1 and the newer Grid 2. We will only focus on Grid 2 because Grid 1 is gradually being deprecated by the Selenium Team.
Selenium Grid uses a hub-node concept where you only run the test on a single machine called a **hub**, but the execution will be done by different machines called **nodes**.

When to Use Selenium Grid?

You should use Selenium Grid when you want to do either one or both of following:

- Run your tests against different browsers, operating systems, and machines all at the same time. This will ensure that the application you are testing is fully compatible with a wide range of browser-O.S combinations.
- Save time in the execution of your test suites. If you set up Selenium Grid to run, say, 4 tests at a time, then you would be able to finish the whole suite around 4 times faster.

What Is Selenium WebDriver(Selenium 2.0)?

Selenium WebDriver allowed you to directly interact with the browsers through your automation test scripts. Java, PHP, C#, Python, Ruby, Perl, and Javascript are some of the programming languages it supports. The browsers it supports include Mozilla Firefox, Google Chrome version 12.0.712.0 and above, Internet Explorer, Safari, Opera version 11.5 and above, and HtmlUnit version 2.9 and above. As for operating systems, Selenium WebDriver supports Windows, Linux, Mac OS, and Solaris. Selenium WebDriver is also known as Selenium 2.

Selenium WebDriver does not handle window component, but this limitation can be overcome by using external tools such as AUTO IT tool, Sikuli etc. It has different location strategies as well such as ID, Name, Link text, Partial link text, Class name, CSS selector and Xpath. It also has better support dynamic web pages like Ajax, where elements of the web page may change without the page itself being reloaded. By using different jar files, we can also test API, Database Test etc. using Selenium WebDriver.

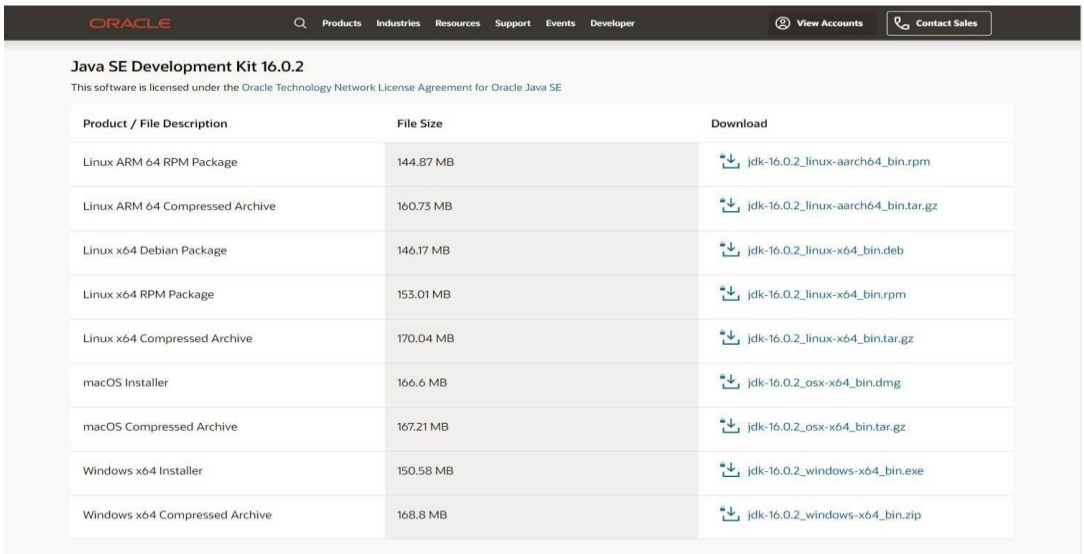
Link : https://www.browserstack.com/guide/how-to-setup-selenium-in-eclipse

- **Prerequisites for configuring Selenium in Eclipse**

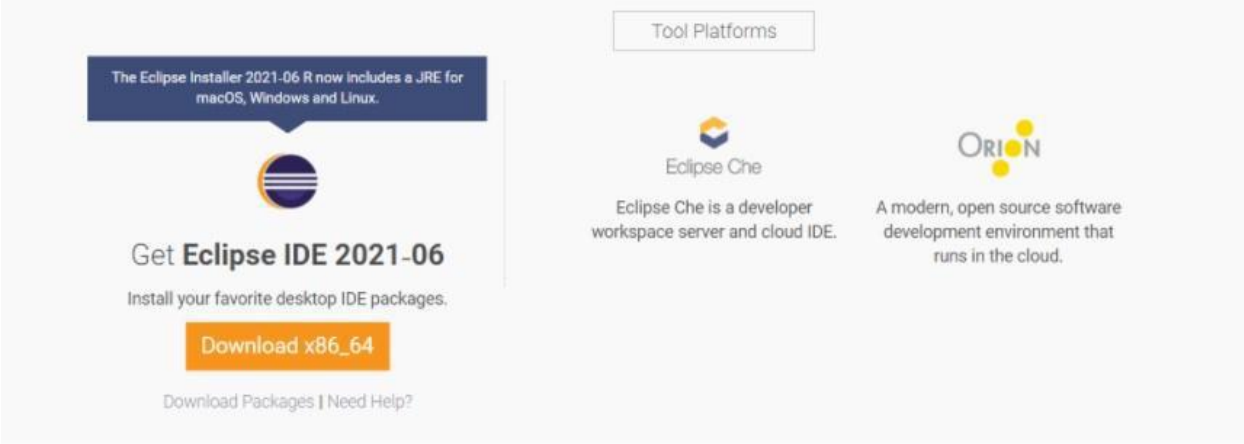
- **Install Java** Download Java SE Development Kit 16.0.2 according to the Windows, Linux,

- Run the JDK Installer by double-clicking on the file name in the download location and following the instructions on the instruction wizard. Alternatively, silently install JDK by entering the following command:

  jdk.exe /s

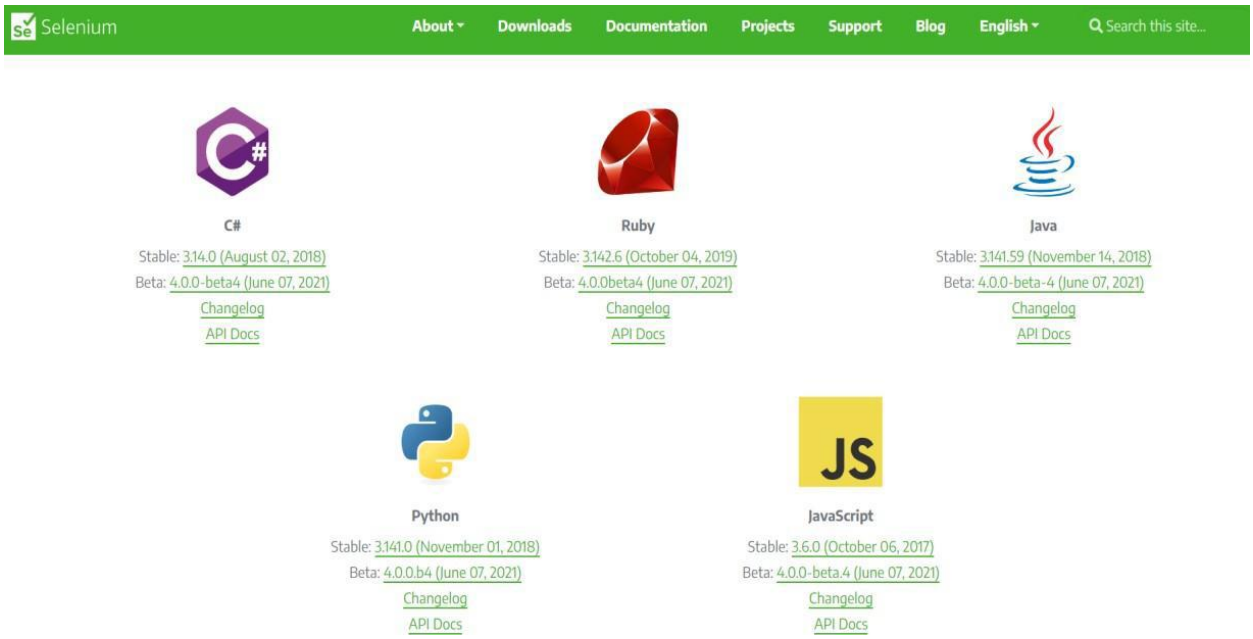


Install Eclipse IDE:



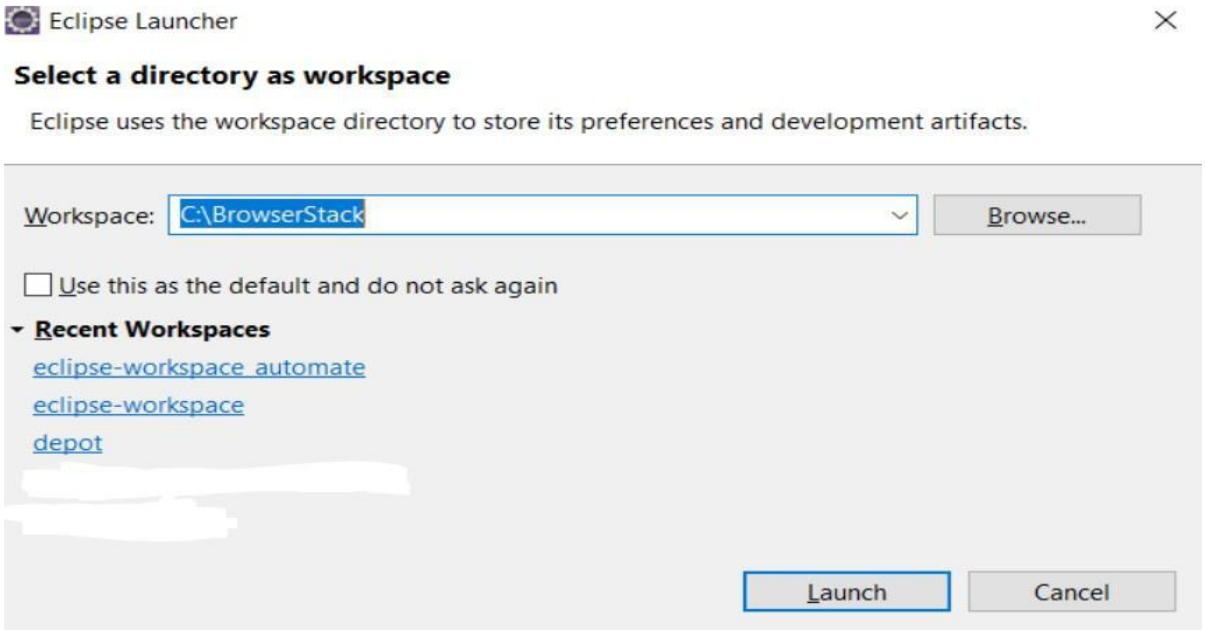
Install Selenium

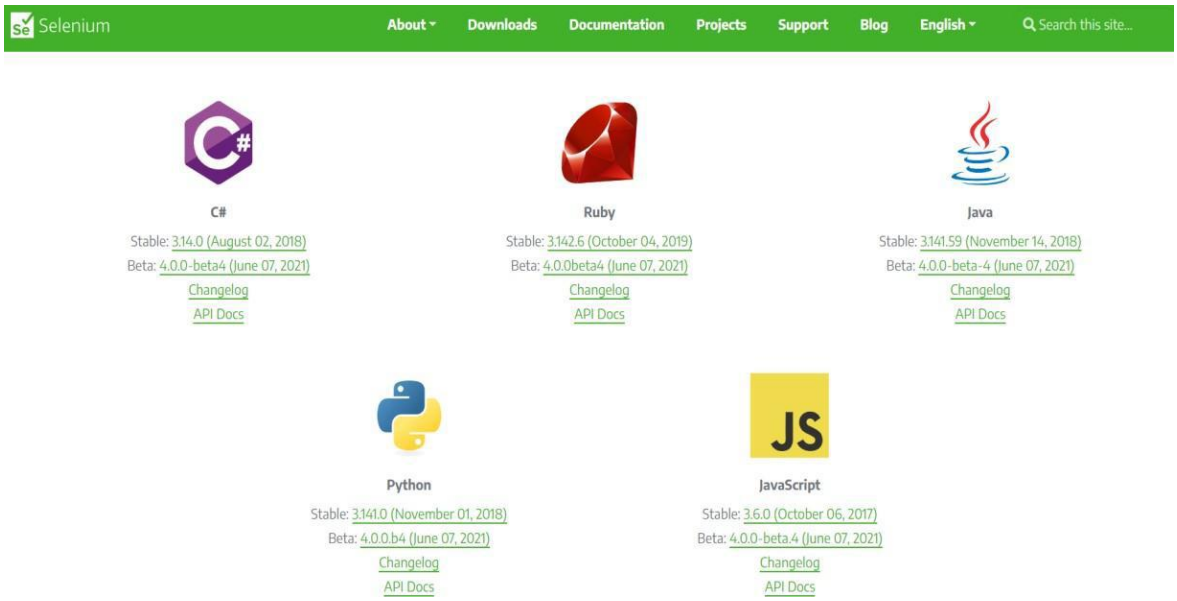
- Download and Install Selenium to be set up in Eclipse.
- **Install Browser Driver**
- For Cross Browser Testing, download the relevant Browser Driver – Chrome Driver (for Chrome),
- Gecko Driver (for Firefox), Safari Driver(for Safari), and Internet Explorer Driver and MS Edge Driver
- IE and Edge respectively
- Place these Browser Driver files in a directory that is part of the environment PATH. This will allow a command-line call to the programs to execute them irrespective of the working directory.
- **Install Java Language Bindings**
  - Version 3.141.59 (2018)
  - Changelog
  - API Docs



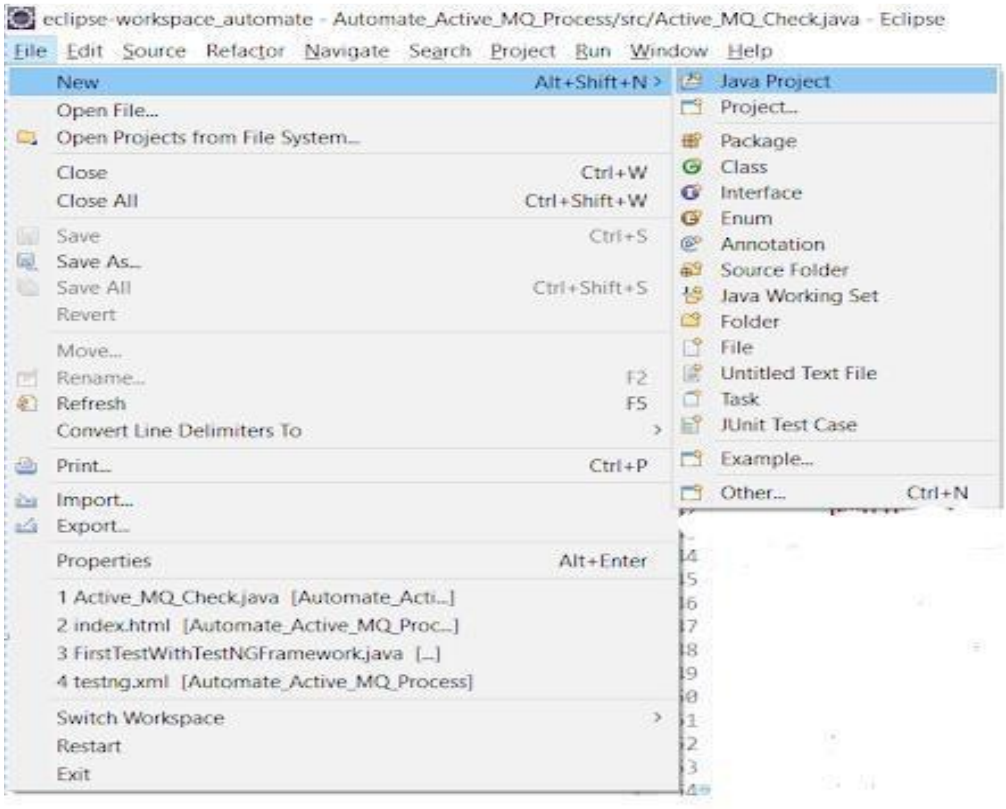
How to configure Selenium in Eclipse

- Here are the steps to configure Selenium Webdriver with Eclipse:
- **Step 1: Launch Eclipse**
- To launch Eclipse double click on the **eclipse.exe** file in the download location.
- **Step 2: Create Workspace in Eclipse**
- This workspace named ―**C:\BrowserStack**‖ is like any other folder, which will store all the test scripts.
- Launch the BrowserStack workspace.



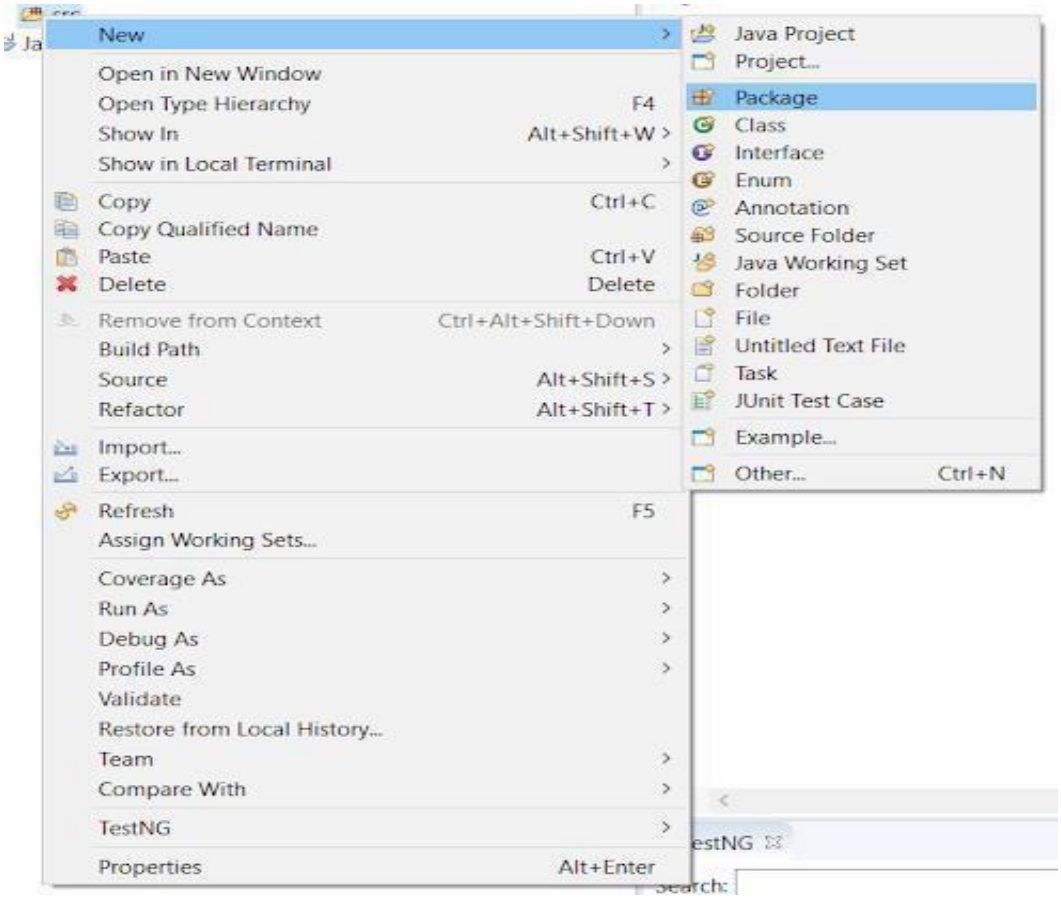


- **Step 3: Create New Java Project in the BrowserStack Workspace**
- Create a new Java Project by clicking on **File > New > Java Project** and name it.



- **Creating a new Java Project**
  Step 4: Create Package and Class under the Java Project
- By clicking on the **src folder** (which is the source folder), create a new package and name it.

- Then right-click on the package name and create a class.



Conclusion: Hence, we have successfully studied Installation of Selenium grid and selenium Web driver java eclipse.

Assignment No: 5

Problem Statement: Prepare Software requirement specification for online education portal

Objectives:
To Prepare Software requirement specification for online education portal

Theory:

What Is a Software Requirements Specification (SRS) Document?

A software requirements specification (SRS) is a document that describes what the softwar and how it will be expected to perform. It also describes the functionality the product needs to stakeholders (business, users) needs.

An SRS can be simply summarized into four Ds:

- Define your product's purpose.
- Describe what you're building.
- Detail the requirements.
- Deliver it for approval.

We want to DEFINE the purpose of our product, DESCRIBE what we are building, DETAIL individual requirements, and DELIVER it for approval. A good SRS document will de everything from how software will interact when embedded in hardware to the expectat when connected to other software. An even better SRS documents also account for real-life u and human interaction. The best SRS documents define how the software will interact w embedded in hardware — or when connected to other software. Good SRS documents account for real-life users.

Why Use an SRS Document?

An SRS gives you a complete picture of your entire project. It provides a single source of every team involved in development will follow. It is your plan of action and keeps all you from development to maintenance — on the same page (no pun intended).

This layout not only keeps your teams in sync but it also ensures that each requirement is hit. It can ultimately help you make vital decisions on your product‘s lifecycle, such as when an obsolete feature.

The time it takes to write an SRS is given back in the development phase. It allows for better understanding of your product, team, and the time it will take to complete.

Software Requirements Specification vs. System Requirements Specification :

A **software requirements specification (SRS)** includes in-depth descriptions of the software that will be developed. A **system requirements specification (SyRS)** collects information on the requirements for a system. —"Software" and —"system" are sometimes used interchangeably as SRS. But, a software requirement specification provides greater detail than a system requirements specification.
>> Need to prove compliance? Here's how to create a traceability matrix >>

How to Write an SRS Document?

Writing an SRS document is important. But it isn't always easy to do. Here are five steps you can follow to write an effective SRS document.

1. Define the Purpose with an Outline (Or Use an SRS Template)

Your first step is to create an outline for your software requirements specification. This may be something you create yourself. Or you may use an existing SRS template. If you're creating this yourself, here's what your outline might look like:

1. Introduction

- 1.1 Purpose
- 1.2 Intended Audience
- 1.3 Intended Use
- 1.4 Scope
- 1.5 Definitions and Acronyms

2. Overall Description

- 2.1 User Needs
- 2.2 Assumptions and Dependencies

3. System Features and Requirements

- 3.1 Functional Requirements
- 3.2 External Interface Requirements
- 3.3 System Features
- 3.4 Nonfunctional Requirements

This is a basic outline and yours may contain more (or fewer) items. Now that you have an outline, let's fill in the blanks.

Download a white paper on best practices for writing requirements >>

2. Define your Product's Purpose

This introduction is very important as it sets expectations that we will hit throughout the SRS. Some items to keep in mind when defining this purpose include:

Intended Audience and Intended Use

Define who in your organization will have access to the SRS and how they should use it. This may include developers, testers, and project managers. It could also include stakeholders in other departments, including leadership teams, sales, and marketing. Defining this now will lead to less work in the future.

Product Scope

What are the benefits, objectives, and goals we intend to have for this product? This should relate to overall business goals, especially if teams outside of development will have access to the SRS.

Definitions and Acronyms

It's important to define the risks in the project. What could go wrong? How do me mitigate these risks? Who is in charge of these risk items?
For example, if the failure of a medical device would cause slight injury, that is one level of risk.
Taking into account the occurrence level and the severity, we can come up with a strategy to mitigate this risk.
>> Need to create a PRD? Here's a how-to with examples >>

3. Describe What You Will Build

Your next step is to give a description of what you're going to build. Is it a new product? Is it an add-on to a product you've already created? Is this going to integrate with another product? Why is this needed? Who is it for? Understanding these questions on the front end makes creating the product much easier for all involved.

User Needs

Describe who will use the product and how. Understanding the user of the product and their needs is a critical part of the process. Who will be using the product? Are they a primary or secondary user? Do you need to know about the purchaser of the product as well as the end user? In medical devices, you will also need to know the needs of the patient.

Assumptions and Dependencies

What are we assuming will be true? Understating and laying out these assumptions ahead of time will help with headaches later. Are we assuming current technology? Are we basing this on a Windows framework? We need to take stock of these assumptions to better understand when our product would fail or not operate perfectly. Finally, you should note if your project is

dependent on any external factors. Are we reusing a bit of software from a previous project? This new project would then depend on that operating correctly and should be included.

4. Detail Your Specific Requirements

In order for your development team to meet the requirements properly, we MUST include as much detail as possible. This can feel overwhelming but becomes easier as you break down your requirements into categories. Some common categories are:

Functional Requirements

Functional requirements are essential to your product because, as they state, they provide some sort of functionality. Asking yourself the question —"does this add to my tool's functionality?" Or —"What function does this provide?" can help with this process. Within Medical devices especially, these functional requirements may have a subset of risks and requirements. You may also have requirements that outline how your software will interact with other tools, which brings us to external interface requirements.

External Interface Requirements

External interface requirements are specific types of functional requirements. These are especially important when working with embedded systems. They outline how your product will interface with other components. There are several types of interfaces you may have requirements for, including:

- User
- Hardware
- Software
- Communications

System Features

System features are types of functional requirements. These are features that are required in order for a system to function.

Other Nonfunctional Requirements

Nonfunctional requirements can be just as important as functional ones. These include:

- Performance
- Safety
- Security
- Quality

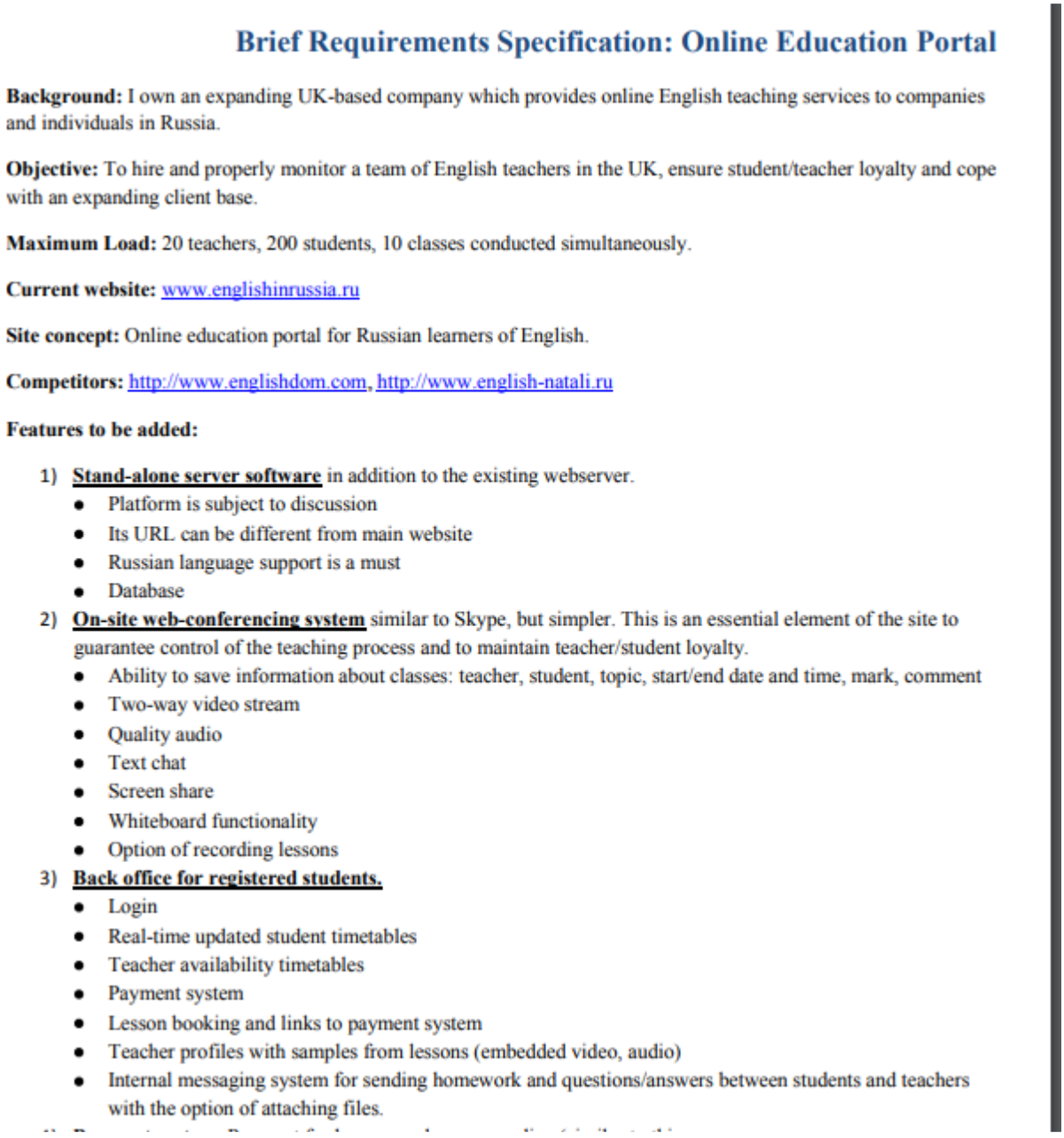
The importance of this type of requirement may vary depending on your industry. In the medical device industry, there are often regulations that require the tracking and

accounting of safety. IEEE also provides guidance for writing software requirements specifications, if you're a member.

5. Deliver for Approval

We made it! After completing the SRS, you'll need to get it approved by key stakeholders. This will require everyone to review the latest version of the document.

Link : https://impressit.io/blog/software-requirements-specification-guide



Brief Requirements Specification: Online Education Portal

Background: I own an expanding UK-based company which provides online English teaching services to companies and individuals in Russia.

Objective: To hire and properly monitor a team of English teachers in the UK, ensure student/teacher loyalty and cope with an expanding client base.

Maximum Load: 20 teachers, 200 students, 10 classes conducted simultaneously.

Current website: www.englishinrussia.ru

Site concept: Online education portal for Russian learners of English.

Competitors: http://www.englishdom.com, http://www.english-natali.ru

Features to be added:

1) **Stand-alone server software** in addition to the existing webserver.
   - Platform is subject to discussion
   - Its URL can be different from main website
   - Russian language support is a must
   - Database
2) **On-site web-conferencing system** similar to Skype, but simpler. This is an essential element of the site to guarantee control of the teaching process and to maintain teacher/student loyalty.
   - Ability to save information about classes: teacher, student, topic, start/end date and time, mark, comment
   - Two-way video stream
   - Quality audio
   - Text chat
   - Screen share
   - Whiteboard functionality
   - Option of recording lessons
3) **Back office for registered students.**
   - Login
   - Real-time updated student timetables
   - Teacher availability timetables
   - Payment system
   - Lesson booking and links to payment system
   - Teacher profiles with samples from lessons (embedded video, audio)
   - Internal messaging system for sending homework and questions/answers between students and teachers with the option of attaching files.

with the option of attaching files.

4) **Payment system.** Payment for lessons and courses online (similar to this: http://www.englishdom.com/en/cost).
5) **Online test functionality.** Interactive exercises in the form of multiple choice questions (similar to this: http://cliomsk.com/on-line-test). This should be created as a customisable template, which can be used to create other interactive exercises by study topic.
6) **Reports.**
   - Ability to monitor who is currently online, who is in the virtual classrooms.
   - Recorded lessons with start date/time and duration

   - Payments list within a selected period, teacher or student.
   - Teacher workload statistics for every teacher
   - Report on student: classes with marks
   - Online test results.
   - Message logs.
   - System log.
7) **Configuration.**
   - User page
   - Student profiles
   - Teacher profiles
   - Online test templates
   - Hourly rates for every teacher

Conclusion: Prepared the software requirement specification document successfully.

Group 2: Mini Project

Software Testing and Quality Assurance Mini Project Dynamic website of covid-19 information using HTML, CSS, JAVASCRIPT And PHP, MySQL database used to store user account, comment, and registration form details. Regular Expression test cases for testing purpose.