

Lab. class C)

Exams week - Test program



Test ID	Test - Name	args	case	Output	Status
TC-01	TestCaseTest	admin	manager	User - Rama (Pass)	Pass

Note :-

- while writing webdriver all the API should import from "org. apache. poi. ss..usermodel" package.
- menu providing file location, file extension should be .xlsx

All the java code available inside the poi folder should be imported to java project.

library should be written back to Excel. Basically while writing data back to Excel, library opened workbook should be closed.

WORK

(used in code reusability)

login code can be written once and everyone can use it by using framework.

WORK

use Test

④ Test
public void executeTest()

③ To execute
test procedure

Execution code
same code for
public void executeTest()

use in Exe
next in Exe
 \equiv → Apache poi code

Excel work - Reusability

Test class
use Test

Excel file = true
ExcelUtil

④ Test
public void setDataUtil()

String str = driver.getattribute("value");
String str = driver.getattribute("value");
String str = driver.getattribute("value");
String str = driver.getattribute("value");

setExcelData(String str,
String name, int row,
int col, String data);

④ Test
public void executeTest()

for (Row row : sheet)
for (Cell cell : row)
if (cell.getCellType() == CellType.STRING)
str = cell.getStringValue();
else if (cell.getCellType() == CellType.NUMERIC)
str = cell.getNumericCellValue();
else if (cell.getCellType() == CellType.BLANK)
str = null;
setExcelData(str, name, row, col);

ExcelLib

```

public class ExcelLib
{
    String filePath = "C:\Users\Deepu\OneDrive\Documents\Takta FizmaTech\testData.xlsx";
    FileOutputSteam fes = new FileOutputStream("file poach");
    Cell.setCellValue("data");
    fes.write("poach");
    fes.close();
}

public String getExcelData (String sheetName, int
    rowNum, int column) throws EncryptedException,
    IOException, InterruptedException
{
    FileInputStream fis = new FileInputStream(filePath);
    Workbook wbo = WorkbookFactory.create(fis);
    Sheet sh = wbo.getSheet(sheetName);
    Row row = sh.getRow(column);
    String data = row.getCell(column).getStringValue();
    Cell.setCellValue();
    return data;
}

```

FileInputSteam

```

public void getExcelData (String sheetName, int
    rowNum, int column) throws EncryptedException,
    IOException, InterruptedException
{
    FileInputStream fis = new FileInputStream(filePath);
    Workbook wbo = WorkbookFactory.create(fis);
    Sheet sh = wbo.getSheet(sheetName);
    Row row = sh.getRow(column);
    String data = row.getCell(column).getStringValue();
    Cell.setCellValue();
    return data;
}

```

```

public void getExcelData (String sheetName, int
    rowNum, int column) throws EncryptedException,
    IOException, InterruptedException
{
    FileInputStream fis = new FileInputStream(filePath);
    Workbook wbo = WorkbookFactory.create(fis);
    Sheet sh = wbo.getSheet(sheetName);
    Row row = sh.getRow(column);
    String data = row.getCell(column).getStringValue();
    Cell.setCellValue();
}

```

```

public void getExcelData (String sheetName, int
    rowNum, int column) throws EncryptedException,
    IOException, InterruptedException
{
    FileInputStream fis = new FileInputStream(filePath);
    Workbook wbo = WorkbookFactory.create(fis);
    Sheet sh = wbo.getSheet(sheetName);
    Row row = sh.getRow(column);
    String data = row.getCell(column).getStringValue();
}

```

FileOutputSteam

```

public class FileOutputSteam
{
    public void write (String s)
    {
        byte [] arr = s.getBytes();
        for (int i = 0; i < arr.length; i++)
        {
            fes.write(arr[i]);
        }
    }
    fes.close();
}

public void write (String s)
{
    byte [] arr = s.getBytes();
    for (int i = 0; i < arr.length; i++)
    {
        fes.write(arr[i]);
    }
}

```

```

public void write (String s)
{
    byte [] arr = s.getBytes();
    for (int i = 0; i < arr.length; i++)
    {
        fes.write(arr[i]);
    }
}

```

```
password);
driver.findElement(By.id("loginButton")).click();
Thread.sleep(2000);
```

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

```
driver.quit();
```

```
Write data back to excel
```

```
File.writeExcelData("Sheet1", 1, 5, "PASS");
```

```
Output
```

Test-Id	Test-Name	Expo	Expo	Expo	Status
TC-101	User-test	admin	manager	user-test	Pass

All multiple testcase has to be run in different browser if we declare launching of browser in each test case then it will be a problem because to execute them in browser instead we execute it in class like Driver in generic libary like consider a code to launch browser so to run in different browser we can only change code in driver class rather than changing in all test cases

In Driver class , every time waiting two lines of code to launch a browser instead we declare an interface Constants which contains a variable instance

changes we want to run test case in different browser (ie. use in different browser) instead executing every time we can move it to constants therefore driver class use can store instance of password via constant instead getting from excel source file use global constants see we store in constants Interface (no user can change code no user can change password)

constant Interface (no user can change code no user can change password)

```
class Test {
    static WebDriver driver;
    static String url = "http://www.qaclickacademy.com/practice.html";
    static String title;
}
```

```
public class Test {
    static WebDriver driver;
    static String url = "http://www.qaclickacademy.com/practice.html";
    static String title;
}
```

```
public class Test {
    static WebDriver driver;
    static String url = "http://www.qaclickacademy.com/practice.html";
    static String title;
}
```

```
public class Test {
    static WebDriver driver;
    static String url = "http://www.qaclickacademy.com/practice.html";
    static String title;
}
```

```
public class Test {
    static WebDriver driver;
    static String url = "http://www.qaclickacademy.com/practice.html";
    static String title;
}
```

Program

```
public interface Constants {
    String browser = "firefox";
    String url = "http://deepu.com";
    String userTest = "admin";
    String password = "manager";
}

public class Driver {
    static WebDriver driver = null;
    static String exePath = "C:\\Users\\DEEPU\\Desktop\\New folder\\Selenium\\chromedriver.exe";
    static String chromePath = "C:\\Users\\DEEPU\\AppData\\Local\\Programs\\chromium\\chromedriver.exe";
    static String iePath = "C:\\Users\\DEEPU\\AppData\\Local\\Programs\\internet explorer\\IEDriverServer.exe";
    static String edgePath = "C:\\Users\\DEEPU\\AppData\\Local\\Programs\\Microsoft\\Edge\\MicrosoftWebDriver.exe";
    static String firefoxPath = "C:\\Users\\DEEPU\\AppData\\Local\\Programs\\Mozilla Firefox\\geckodriver.exe";
}

public static void main(String[] args) {
    System.out.println("Hello Selenium");
    Constants constants = new Constants();
    Driver driver = new Driver();
    driver.getBrowser(constants.chrome);
    driver.getDriver();
    driver.getURL(constants.url);
    driver.getIE();
    driver.getEdge();
    driver.getFirefox();
    driver.getIE();
    driver.getEdge();
    driver.getFirefox();
}

class Browser {
    public void getBrowser(String browserName) {
        if (browserName.equals("chrome")) {
            System.setProperty("webdriver.chrome.driver", "C:\\Users\\DEEPU\\Desktop\\New folder\\Selenium\\chromedriver.exe");
            WebDriver driver = new ChromeDriver();
            driver.get("http://www.google.com");
        } else if (browserName.equals("ie")) {
            System.setProperty("webdriver.ie.driver", "C:\\Users\\DEEPU\\AppData\\Local\\Programs\\internet explorer\\IEDriverServer.exe");
            WebDriver driver = new InternetExplorerDriver();
            driver.get("http://www.google.com");
        } else if (browserName.equals("edge")) {
            System.setProperty("webdriver.edge.driver", "C:\\Users\\DEEPU\\AppData\\Local\\Programs\\Microsoft\\Edge\\MicrosoftWebDriver.exe");
            WebDriver driver = new EdgeDriver();
            driver.get("http://www.google.com");
        } else if (browserName.equals("firefox")) {
            System.setProperty("webdriver.firefox.driver", "C:\\Users\\DEEPU\\AppData\\Local\\Programs\\Mozilla Firefox\\geckodriver.exe");
            WebDriver driver = new FirefoxDriver();
            driver.get("http://www.google.com");
        }
    }
}
```

driver.sendKeys(By.name("pwd")), sendKeys

(constants predefined);

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

driver.quit();

elbow setExcelData ("sheet 1", 1, 3, "Pass");

sheet 1

row 1

cell 1

cell 2

cell 3

Test-Id	Test Name	arg0	arg1
TC-01	User Test	User - Pass	RAN

ArrayList → Fixed in size, collection of similar object

Collection → Dynamic in size

collection of dissimilar objects

ArrayList

String name [0] = new String();

name [0]

allows duplicate values
values

int id [0] = new int [] { }

id [0]

double fee [0] = new double [] { }

fee [0]

Alexaykist

Emp["RAM",0,10,8]

List allows duplicate values

Set

ArrayList

Collection

Collection

ArrayList is a collection

or structure structure

(a) homogeneous data

(b) heterogeneous data

ArrayList is fixed in size

the size to decide the size can grow

the size time of execution

accession

collection

list

set

queue

linked list

list	set	queue	linked list
[]	{ }	[]	[]
→	→	→	→

list	set	queue	linked list
[]	{ }	[]	[]
→	→	→	→

list	set	queue	linked list
[]	{ }	[]	[]
→	→	→	→

ArrayList is a collection

or structure structure

(a) homogeneous data

(b) heterogeneous data

ArrayList is not fixed in size

the size to decide the size can grow

the size time of execution

accession

list allows duplicate values

Set does not allow duplicate values

Set is Index Based

* In order to get the data from list we go for get() method

* Insertion Order is Preserved

Traversing Order is maintained

using for loop

Difference between Map & List

→ In list if we a element in middle then same element get shifted

o business : admin
o manager : user

o manager : manager
3 will
→ using range (index)

Map

is key value based

Key	Value
business	for
manager	admin
password	manager
lock	map //

List

- * To get the data from list we go for get() method
- * Traversing Order is maintained
- * In order to add element to list use add() method
- * In order to add element to list use push() method
- * To get the data from list use get(index) method
- * To get the data from list use get(key)

Difference between Set & Map

Set

- * Set is not Enclosed * Map is Key Value Mapped
- * To get data from set use Iterator
- * In Set there are no duplicate values
- * Unique

Set

- o business
- o manager
- o user
- 3 will
- using range (index)

Key	Value
business	for
manager	admin
password	manager
lock	map //

Map

- * To get the data from map we go for get() method
- * Traversing Order is maintained
- * In order to add element to map use put(key, value)
- * In order to add element to map use push() method
- * To get the data from map use get(key)

Map

- * To get data from map use get(index)
- * To get data from map use get(key)
- * In map, values can be duplicates but key is unique

- In every test case, when move to navigating from one page to another page we need to write simpler wait | explicit wait instead of writing | when embedding test syntax, we can use a class `WebDriverCommonLib` in generic lib which writes `ImplicitWait` or `wait statements` and also `return values`.
- Re-usability
- can used to change the time

WebDriverCommonLib

`waitForPageToLoad()` → Implicit wait
`waitForXPathPresent()` → Explicit wait

Instead of implementing separate code pass XPath & wait explicitly by adding idea

↓
overrided.

`waitForText()`

→ to verify the expected result with actual result instead of writing `if else` in each test case, we just needed to

verify the text by passing web elements to `waitForText()`

→ `waitForElementPresent()` → Explicit wait
`waitForIdPresent()` → we use to wait until element present if we

start to wait until the element present by name and we just needed

Program "Generic Lib"

public class WebDriverCommonLib

Duges

① Re-usability

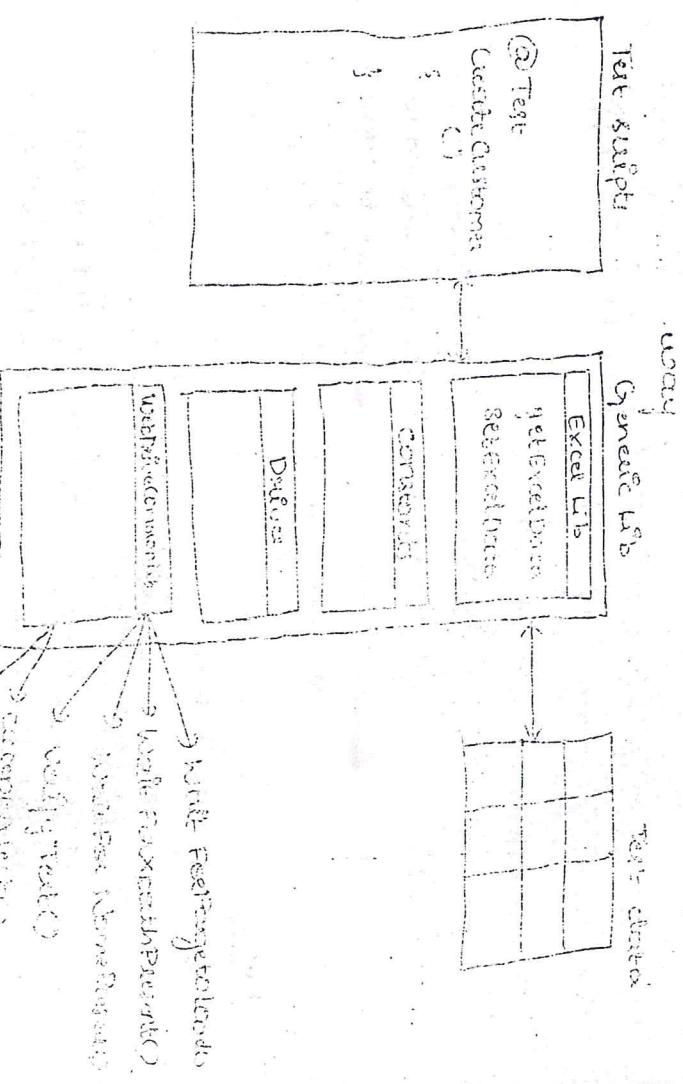
② No need to remember Selenium API's

Advantage of using our common code in one place means that executing a object code can use only once and those to execute

Disadvantage of reusing code methods in one place
① If object required, and other logic methods will be created and occupies free memory space.

Ex: size time of class `CustomerDetails` & `Variables` occupies free memory space so unnecessary memory spaces will be wasted.

② It executes confusing if use in test scenarios



```
public void waitForPageLoad()
```

```
{  
    WebDriver driver = manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
```

```
    log.info("Page loaded");
```

```
    System.out.println("ExpectedText + " + data + " is verified  
    = " + pageText);
```

```
} else
```

```
    public void waitForXpathVisible(By xpath, String errorMessage)  
{  
    }
```

```
    WebDriverWait wait = new WebDriverWait(driver, 30);  
    driver, 30);
```

```
    wait.until(ExpectedConditions.presenceOfAll  
    - ElementsLocatedBy(By.xpath("//xpath")));
```

```
    public void waitForNameVisible(By name, String errorMessage)  
{  
    }
```

```
    WebDriverWait wait = new WebDriverWait(driver, 30);  
    driver, 30);
```

```
    wait.until(ExpectedConditions.presenceOfAll  
    - ElementsLocatedBy(By.name("name")));
```

```
    public boolean verifyText.WebElement webElement, String  
    expectText)  
{  
    }
```

```
    boolean flag = false;  
    String actual = webElement.getText();
```

```
    if (expectText.equals(actual))  
        flag = true;
```

```
    public void userText
```

```
    ExcelLib elib = new ExcelLib();  
    WebElement commonLib = new WebElement -
```

- Common Lib()

④ Test

private void createUserTest() throws EncryptedException, IOException, ElementException, IOException, InvalidFormatException

Exception

String userEncr = elib.getExcelData("Sheet1", 1, 2);

String password = elib.getExcelData("Sheet1", 1, 3);
String lastName = elib.getExcelData("Sheet1", 1, 4);

String fName = elib.getExcelData("Sheet1", 1, 5);

String emailId = elib.getExcelData("Sheet1", 1, 6);
String expectedMsg = elib.getExcelData("Sheet1", 1, 7);

// step1 : login to app

driver.findElement(By.name("username")).sendKeys("Admin");

driver.findElement(By.name("password")).

sendKeys(password);

driver.findElement(By.name("lastName")).sendKeys(lastName);

driver.findElement(By.name("fName")).sendKeys(fName);

driver.findElement(By.name("email")).sendKeys(emailId);

driver.findElement(By.name("expectedMsg")).sendKeys(expectedMsg);

// step2 : navigate to user page

String fNameText = driver.findElement(By.xpath("//div[text()='User']")).

getText();

String fNameExpectedText = driver.findElement(By.xpath("//div[2]/text()")).

getText();

steps : Create New User
driver.findElement(By.xpath("//input[@type='button'][@value='Create New User']")).click();

driver.findElement(By.name("username")).sendKeys("Admin");
driver.findElement(By.name("password")).sendKeys(password);
driver.findElement(By.name("lastName")).sendKeys(lastName);

driver.findElement(By.name("fName")).sendKeys(fName);
driver.findElement(By.name("email")).sendKeys(emailId);
driver.findElement(By.name("expectedMsg")).sendKeys(expectedMsg);

driver.findElement(By.xpath("//input[@type='button'][@value='Create User']")).click();

Assesst. current frame (status, "msg" is not working)

= `fail()`;

driver.quit();

If write driver back to Excel

elbow. setExcelData ("Sheet1", 1, 3, "msg")

3

4

Output

Test_Id	Test_Name	arg0	arg1	arg2	arg3	Kafka
Tc_01	userTest	User1 - RAM	password	RAM		
		arg4	arg5			

Result @ymca.com user account has been successfully
Created

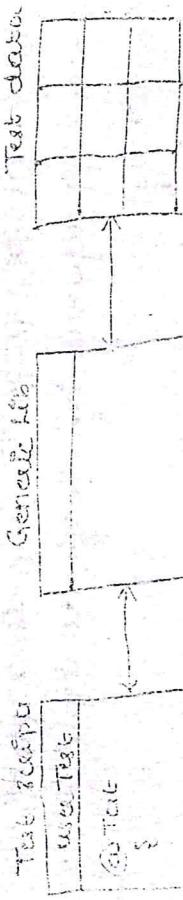
Debug
pass

→ the xpath is used coded, if UI changes may
be xpath many changes then we need to change
in user test case

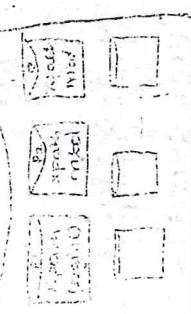
Generic Lib

Generic Lib

```
class Test:  
    def __init__(self):  
        self.driver = webdriver.Chrome()  
    def test_login(self):  
        self.driver.get("http://www.google.com")
```



Page Object Representer file



Generic Lib
Page object writer

Page Page concepts

Java Page concepts

Page Object Representer

Generic Lib

→ does more concern separation
→ figure consider this example of design if
common lib reuse to other project

NOTE

→ this per file association issue x path ignored
note we moved concern between a test script
first and we should get true x path from
external source, External resource might be C
Xu, * properties, .java

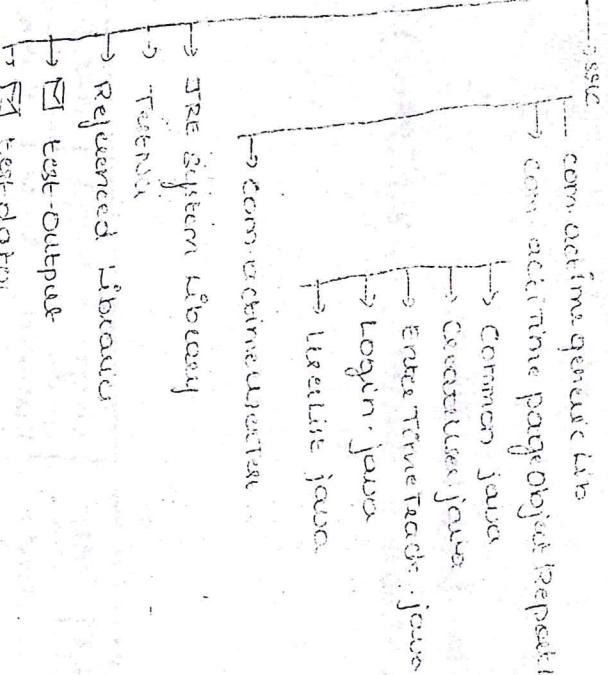
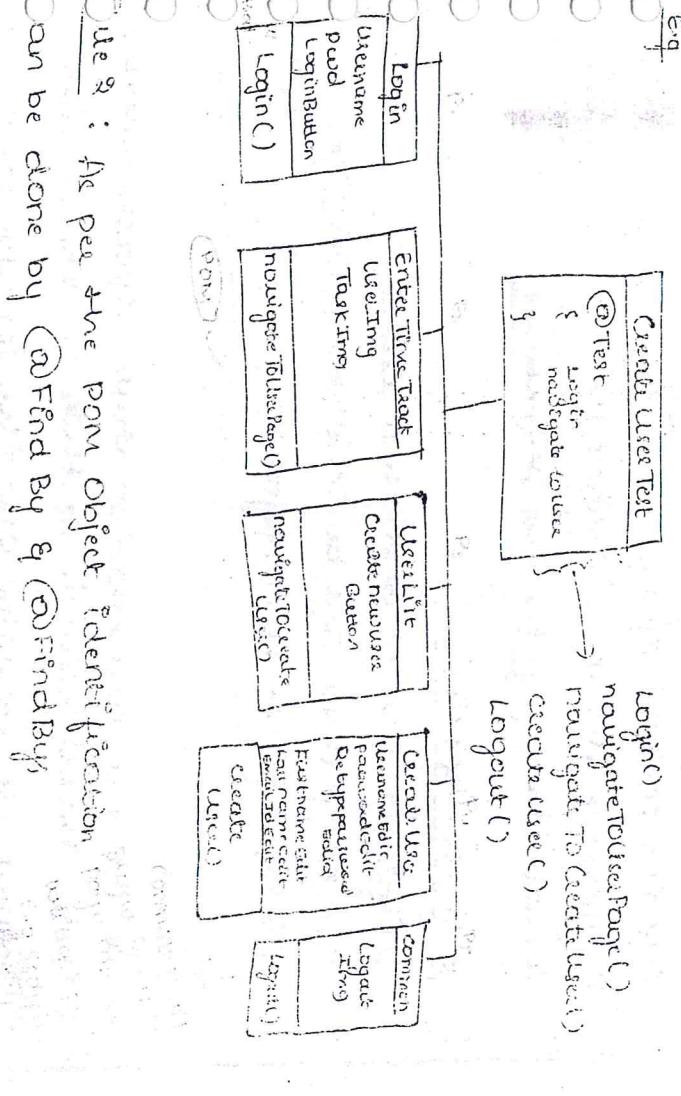
TAKUFAZONE WORK

Page Object Model (POM) Page Factory as a Java
Designed pattern proposed by Google to implement
Object Repository.

Page Object Repository is a collection of web element
Depth Eg Business Lib.

Page Object Repository techniques / POM techniques

Rule 1: As per the POM create a dedicated java
file for each page available in application and
store page specific web element (xpath) in a
particular page.



Rule 2: As per the POM object identification can be done by `@FindBy` & `(@FindBy)`

passwOrdEcT. SeturName (username)

LoginBtn. Click()

}

Rule 3: In order to create here object to your
classes available in page object pool, we
will use PageFactory class.
should take an object

Program

PageObjectRepository

public class Login

{ @FindBy (name = "username")

private WebElement userNmaeEdit;

@FindBy (name = "password")

private WebElement passwordEdit;

@FindBy (id = "loginButton")

private WebElement loginBtn;

public void loginToApp (String username, String password, String url)

{

Driver.driver.get(url);

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

UserTest Script

public class UserTest

{ ExcelLib excel = new ExcelLib();

WebdriverCommonLib webLib = new WebdriverCommonLib();

lib();

EDIT

public void CreateUserTest () throws Exception,
DocumentException, IOException, InterruptedException

Exception

{

// get browser object

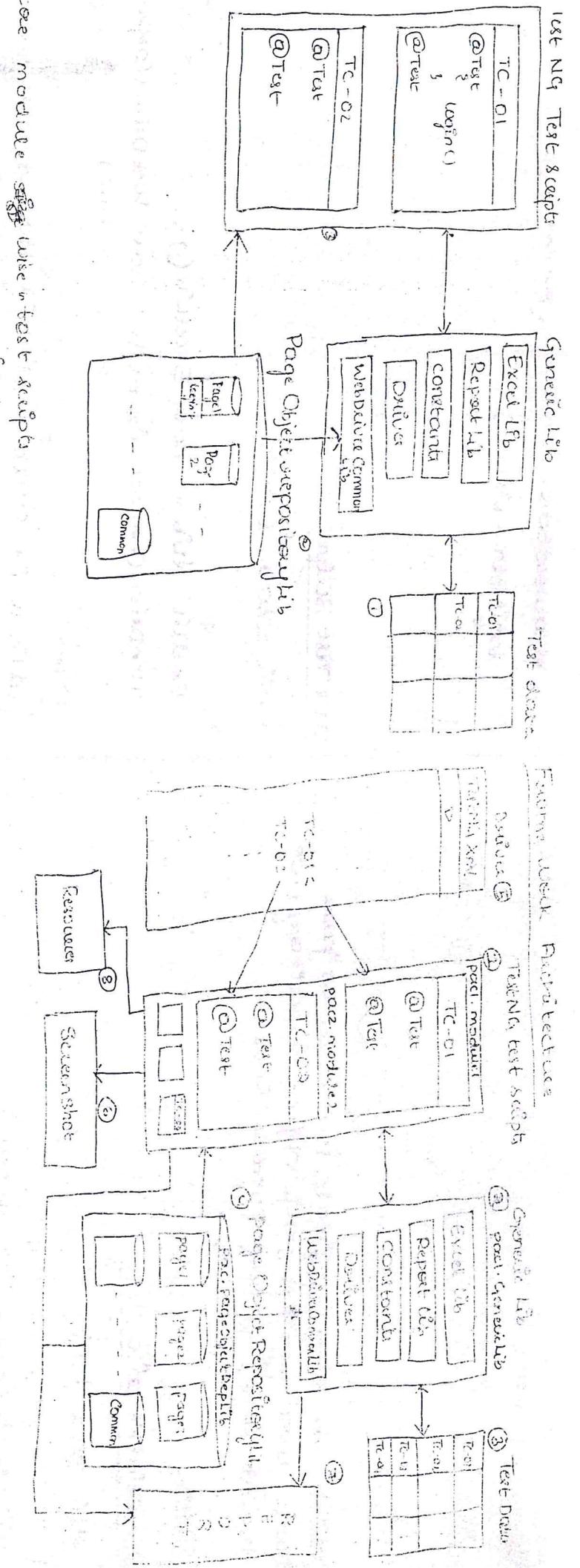
webdriver driver = Driver.getBrowser();

// step1 : login to app

login LoginPage = PageFactory.getPageElement(driver, LoginPage.class);

LoginPage.loginToApp(Constants.userId, Constants.password, Constants.url);

Test NG Test Scripts



Store module wise in test scripts

→ maintenance will be easy

RAME WORK

- well organized structure of package, one driver script will take care of entire batch execution
- without manual interaction

Testing test scripts is used to write test scripts

including to the module using testing construction. Generic lib connects to all the reusable methods. Driver is used for only one driver it contains excel lib → which is used to fetch the database. The driver is used to read the excel sheet.

The driver is used to take screenshot. TestNG Test Scripts is a process followed by different companies based on their requirement to execute the test scripts successfully without any manual interaction.

rame week is a custom tool developed by business developer, which provides lots of custom scenario method, that makes automation test enginee

Constants → const of all global parameters
like url, browser, usernames, password.
If we want test script to execute in diff. browser then it can be configured here.

Defining → used to launch the database based on the requirements.
Driver, Connection → consists of our reusable statements.
in test script like wait for project to load, wait for xpath present

Project Object Repository lib is developed using POM concept where it consists of xpath of business objects like login page on the application, page object repository is specific to project. Test driver, all the test data is stored in the application must be stored in Excel. Each scenario must have one test case.

Repetet fields, once the test execution is completed all the sapports generated will be stored in the folder.

Screenshot folder, whenever the test case fails, screenshot is taken & stored in this folder.

Resources folder is used store all the files of execution file

Main uses of Frame work

① Re-usability

③ Batch Execution

Reporting

- * whenever we run test suite with 100's of test scripts, skipping feature might be useful to handle errors.
- * handle migration issue.
- * There are two reasons test script migration issue because of product bug or test script bugs because of test script modification (modify test script)

Product issue → cause a defect

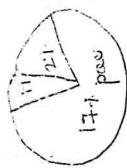
- * If test script fails because of product issue, log in QC if cause a defect.
- * If test script fails due to test script issue, modify in test script.
- * Reporting issue of a type
 - ① High level reporting
 - ② Low level reporting

Normal reporting

Support (jira)

It is an add-on tool which is used to generate report in piechart form.

Test	Pass	Fail	skip	Total
QC	17	3	1	-



low level suspending

→ sop X (don't need code in test script)

→ Repetee [testNG]

→ login

In real time, we won't use sop statements instead

use susceptor class

↳ used to generate low level report

@Test
public void sampleTest

eg

```
public void sampleTest  
{  
    Repetee.log("login to app");  
    Repetee.log("create user");  
    Repetee.log("verify");  
    Repetee.log("logout");  
}
```

) executed & it will not generate in console
) but generate in html report when clicked on

test case name, low level suspect will

be generated

menu suspect

↳ sample test 1

high level reporting → customer point of view
low level suspending → test engg point of view

Interview Question

1. Which frame work you have used for this project?

→ hybrid driven framework

(combination of data driven & medium driven)

To generate low level report in nemereport
uses
Eclipse conga

3. Explain facade pattern?

4. What is repository?

→ Collection of web element present in a single place is called repository.

What is POM?

→ It is a Java designed pattern, we have used to implement Object Repository.

5. Difference between POM & Page factory?

POM is a collection of page class which contains xpath & Business Library.

→ Pagefactory is a class available in webDiver which is used to create an object to page classes to access web elements.

6. Advantage of Page Object Repository

→ whenever UI is getting changed requirement changes, modification of a path in repository is very easy.

→ maintenance will be very easy.

→ whenever UI getting changed for ex login page got modified, we know which class to go & directly modify the xpath.

→ In page factory classes we can directly store web elements using @FindBy & @FindBy annotation.

7. What kind of supporting feature you have used in Automation?

→ These are two types of repository we have generated

① High level repository → To generate huge report we have used embed emuiObject-report generation in test case.

② Low level repository → To generate low level report we have used initial suppose class reusable in testNG.

Repository class is used to generate low level report in initial step.

Q. What is Generic Library & advantages of Generic library.
→ Generic library contains reusable methods that can be used in any project.

Advantages:

- * can get data from excel
 - * can get business logic based on requirement
 - * Inherit user statement are available
 - * global parameters are available.
- Q. What is Business Management Library?
→ Business Library containing reusable methods that can be used specific to the project.

→ Business library is a building block of test case.

Q. What are the steps followed in creating generic library?
→ First we have to create generic library then after creating generic library we have to create generic class which contains generic methods.

Screen Shots

[Event Firing WebDriver]

→ getScreenshot (Output Type: File)

→ true paste in MS paint

File Screenshot = getScreenshotAs (OutputType: File)

E.g.

public class SampleTest

{

@Test

public void screenshot () throws IOException

WebElement driver = new FirefoxDriver();

driver.get ("http://katalon-qa/login.aspx");

// create a object to "WebDriver Driver"

Listener class to get a screenshot

Event FiringWebDriver eDriver = new EventFiring

WebDriver (driver);

// capture file type of ss using getScreenshot

As() method

File Screenshot = eDriver.getScreenshotAs (OutputT

-ype: FILE);

// create a new file in screenshot folder in framework

File dstFile = new File ("C:\\workspace - OCM30\\

How to take screen shot?

Press Print Screen

True Paste in

// using Apache POI , copy the image to word file .

FileInputStream copyFile (String , String)

→ Press Print Screen

→ True Paste in

File Screenshot = getScreenshotAs (OutputType: File)

E.g.

INSTANCE

→ [Test Listener]

→ [Test Result]

→ [Test Suite]

WebDriver Listener → used to take screenshot

TestNG Listener → used to monitor the test & takes screenshot of fail

public class SampleTest

- implements TestListener

(@Listeners (page, SampleTest))

class SampleTest

(@Test public void onTestFailure

(Exception e) {

public void onTestFailure

(ITestResult result) {

public void code to

take screenshot

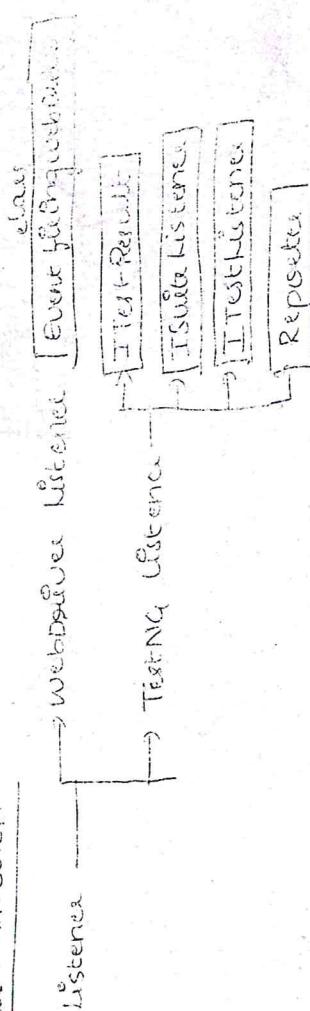
File file = new

File ("c:\\") ;

public void onTestFailure

(Exception e) {

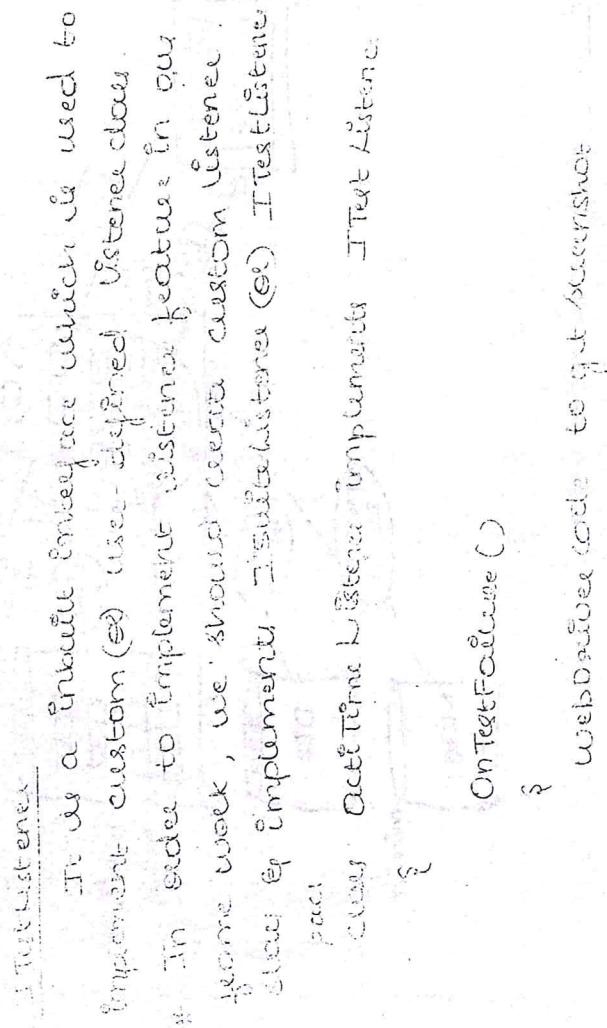
Other Listener



- * Listener is a feature to monitor entire test execution events in the run time
 - e.g. Listener can print test results until loc of test script, to monitor the entire test execution we go for Listener annotation, whenever test fails, Listener annotation can invoke a WebDriver Listener class to take a screenshot in the run time.
- * There are two types of Listener
 - ① WebDriver Listener
 - ② TestNG Listener
- * WebDriver Listener
 - EventListener WebDriver class is a WebDriver class it is used to capture the screenshot in junit.
- * TestNG Listener
 - This class multiple Listener class available in TestNG

Reporter API

Reporter Listener API
It is used to generate log in run time



- * It is a built-in interface which is used to implement custom (e.g) user-defined Listener class.
- * In order to implement Listener interface in our Listener, we should create custom Listener class (e.g.) TestListener class by implementing Listener interface.
- * Listener Listener implements TestListener interface.
- * OnTestFailure()
 - WebDriver code to take screenshot
- * In order to use Listener interface in test script we should invoke Listener class before the class declaration.

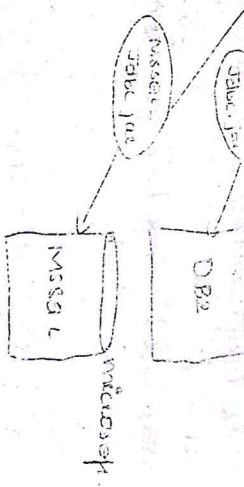
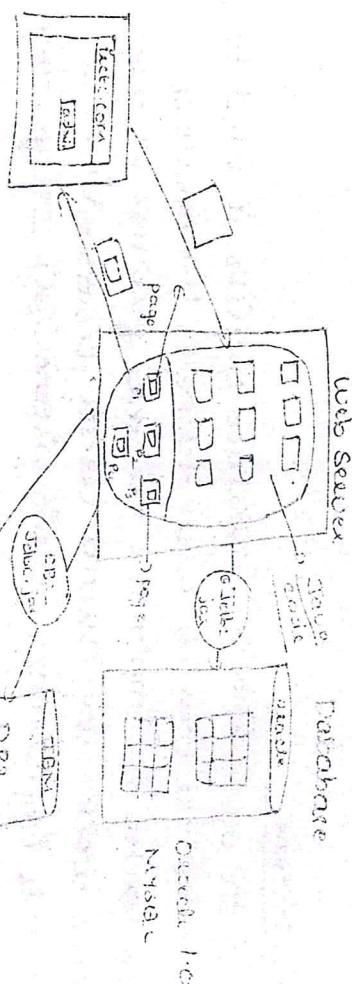
new @Listener (for Listener class)

class Listener

Test Request

- * It is used to capture test execution status or test name in the run time
 - ① Test
 - ② Test

Telco - Java [Java data base connection]



Case 1

In case 1, SQL code would be uploaded by user through JDBC interface to test the application. So JDBC driver engine is used by user to execute the query.

In case 2, if user is not matching SQL or SQL after to check, JDBC will be used to verify the statement.



In case 1, SQL code would be uploaded by user through JDBC interface to test the application. So JDBC driver engine is used by user to execute the query.

In case 2, if user is not matching SQL or SQL after to check, JDBC will be used to verify the statement.

In case 1, SQL code would be uploaded by user through JDBC interface to test the application. So JDBC driver engine is used by user to execute the query.

In case 2, if user is not matching SQL or SQL after to check, JDBC will be used to verify the statement.

In case 1, SQL code would be uploaded by user through JDBC interface to test the application. So JDBC driver engine is used by user to execute the query.

In case 2, if user is not matching SQL or SQL after to check, JDBC will be used to verify the statement.

In case 1, SQL code would be uploaded by user through JDBC interface to test the application. So JDBC driver engine is used by user to execute the query.

In case 2, if user is not matching SQL or SQL after to check, JDBC will be used to verify the statement.

In case 1, SQL code would be uploaded by user through JDBC interface to test the application. So JDBC driver engine is used by user to execute the query.

In case 2, if user is not matching SQL or SQL after to check, JDBC will be used to verify the statement.

In case 1, SQL code would be uploaded by user through JDBC interface to test the application. So JDBC driver engine is used by user to execute the query.

In case 2, if user is not matching SQL or SQL after to check, JDBC will be used to verify the statement.

In case 1, SQL code would be uploaded by user through JDBC interface to test the application. So JDBC driver engine is used by user to execute the query.

In case 2, if user is not matching SQL or SQL after to check, JDBC will be used to verify the statement.

In case 1, SQL code would be uploaded by user through JDBC interface to test the application. So JDBC driver engine is used by user to execute the query.

In case 2, if user is not matching SQL or SQL after to check, JDBC will be used to verify the statement.

In case 1, SQL code would be uploaded by user through JDBC interface to test the application. So JDBC driver engine is used by user to execute the query.

In case 2, if user is not matching SQL or SQL after to check, JDBC will be used to verify the statement.

Different types of Feature tests

test - script

1. Data driven framework
2. Key based driven framework
3. Modular driven framework
4. Hybrid driven framework

Claus Account

Test data	
Acc1	psw1
Acc2	psw2
Acc3	psw3

```
@Test (dataProvider = "getdata")
public void CreateAccount
{
    ① login
    ② create account
    ③ verify
    ④ logout
}
```

Banking Application User Driven Framework

Data driven Feature test

- * Whenever application ideal user huge amount of data we prefer update driven feature test

* Getting the data from the external source in same user test case with different set of data is called data driven testing, also called as parameterization

* In order to achieve parameterization, we should use @dataProvider annotation method for each test in a class

* Data provider annotation will be used to get run time user test case with different set of requirements

Claus Account

- ① login to banking App
- ② create 100 account
- ③ verify account
- ④ logout

Claus Account

Test data	
Acc1	psw1
Acc2	psw2
Acc3	psw3

```
@Test (dataProvider = "getdata")
public void CreateAccount
{
    ① login
    ② create account
    ③ verify
    ④ logout
}
```

⑤ Data provider
public void getdata()

Data Provider	
Acc1	psw1
Acc2	psw2
Acc3	psw3

Claus Account

(Big Application uses this type of feature test)

Like Gmail, CRM

- * maintenance is very hard & tough
- * if per two module deliver feature weekly by feature component should be maintained in module wise, called module driven framework

Modular deliverable frame work is very old

frame week, nowdays company always prefer hybrid frame week instead of modular

deliverable frame week, because modular didn't become week is very difficult to maintain all

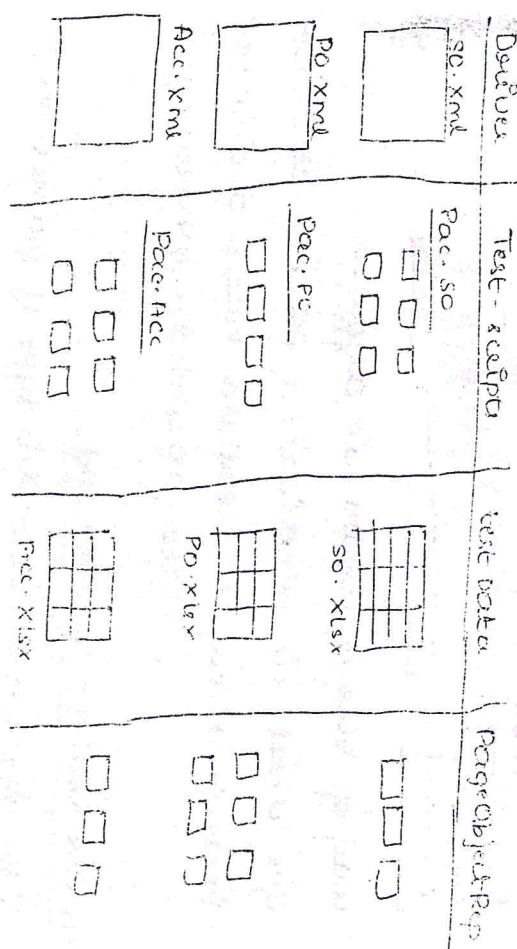
the components.

E.g

CRM			
SO	PO	Acc	Contact
100	200	300	400

SO			
PO	Acc	Contract	1
200	300	400	1

Medium deliverable frame week



Hybrid deliverable frame week

combination of any two frame work is called

Hybrid deliverable frame week

E.g
Our frame week is combination of data driven & modular deliverable

→ Arranging all test cases in module wise

SO modules driven

→ Getting the data from excel here it is

Data driven

Keyword driven framewark

→ use keyword framework

→ writing test case on Excel sheet

↓
manual to writes the test case

In Excel

Selenium → nobody uses this frame In AT&T, this frame week is used.

* whenever we need user friendly framewark (when

manual test engg needs to write a automation test script with less knowledge on automation

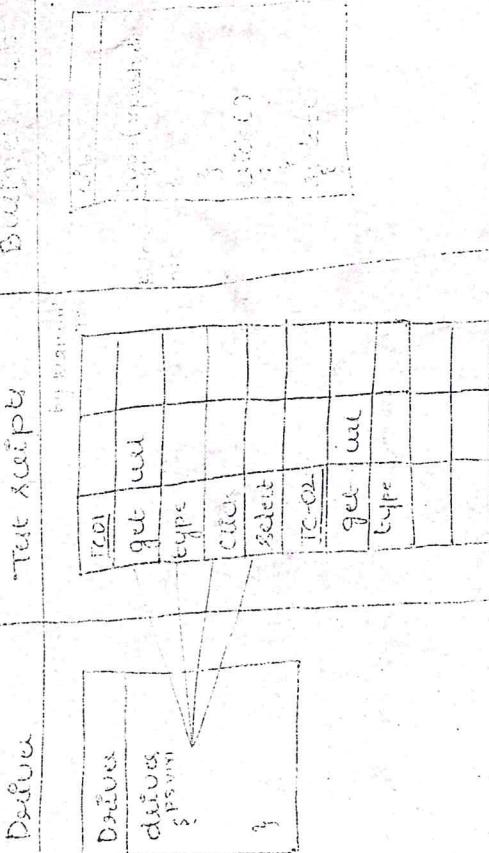
took a programming language) we prefer keyword driven frame work.

* As per this frame work, all the test script should be written in Excel

* But the keyword (method) like type, click, select is implemented by frame week developer

SUN

Device



Test Scripts

Driver	Test	Result
divs	get url	success
type	type	success
click	click	success
select	select	success
TC-02	get url	success

Business Logic



SUN Web Server

- * SUN Web Server is a open source enterprise web server.
- * It is based on Apache, Java, and MySQL.
- * It is used for web development, testing, and deployment.
- * It is available for free download.
- * It is used for web development, testing, and deployment.
- * It is available for free download.

SUN

- * SUN is a open source tool which is used to host web application.
- * SUN is a component of SUN ONE.

- * SUN is a source control tool we can store entire version from version 1 to a single place.

* There are two types of software available in SUN

- A) SUN Web Server
- B) SUN Client

- * SUN Server is a software written in C/C++ installed on server machine, and we can check or view repository in SUN server, to get the entire selenium home work.

- * SUN Client is also a software installed in C/C++ and an environment

Java

- * Java is a programming language developed by Sun Microsystems.
- * It is a platform-independent language.
- * It is used for web development, enterprise application development, and mobile application development.
- * It is available for free download.
- * It is used for web development, enterprise application development, and mobile application development.
- * It is available for free download.

Automation Testing

- * Automation Testing is a process of testing software applications to verify that they behave as expected.
- * It is used to test web applications, mobile applications, and enterprise applications.
- * It is available for free download.
- * It is used to test web applications, mobile applications, and enterprise applications.
- * It is available for free download.

Selenium

- * Selenium is a open source tool which is used to host web application.
- * SUN is a component of SUN ONE.

- * SUN is a source control tool we can store entire version from version 1 to a single place.

* There are two types of software available in SUN

- A) SUN Web Server
- B) SUN Client

- * SUN Server is a software written in C/C++ installed on server machine, and we can check or view repository in SUN server, to get the entire selenium home work.

- * SUN Client is also a software installed in C/C++ and an environment