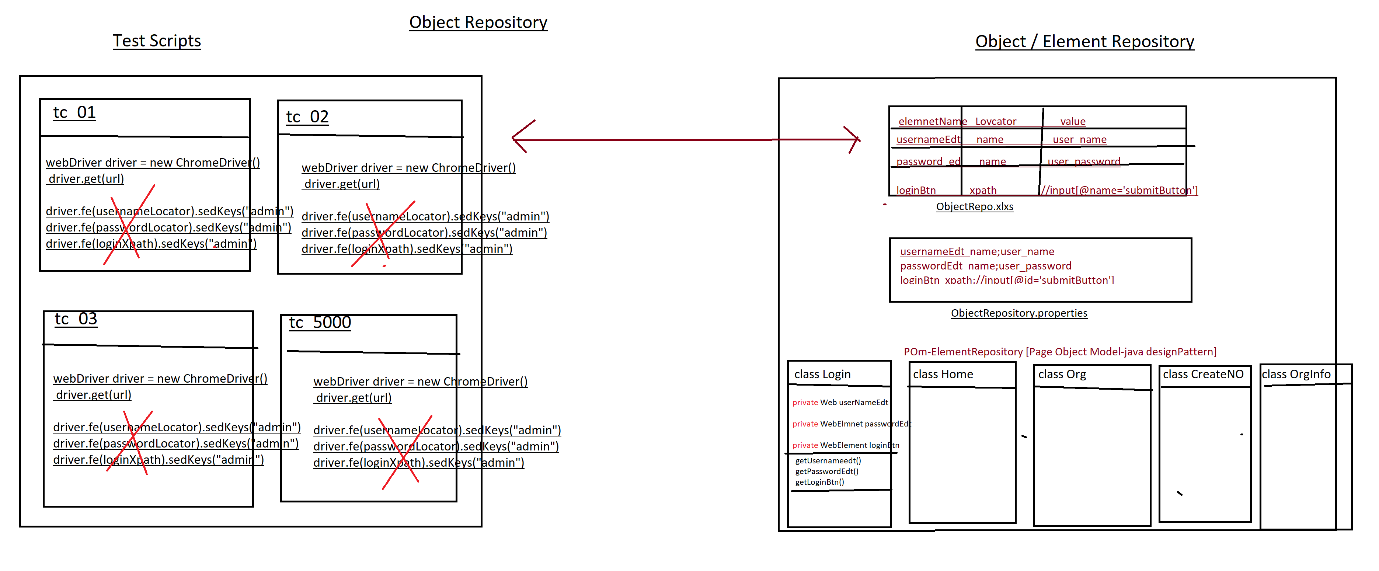
1. What is Object/Elements/POM Repository

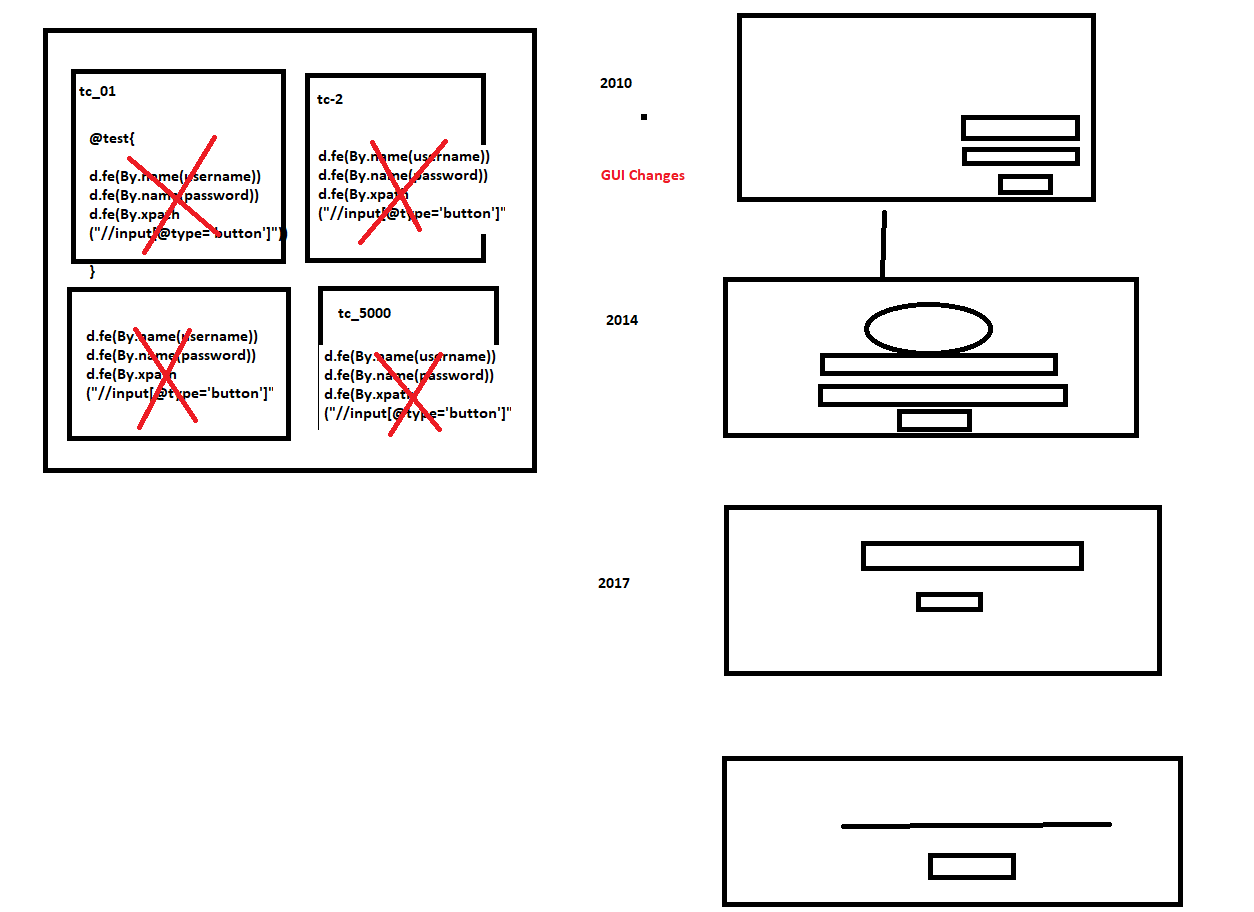
Its collection of elements locators & business libraries in one place & its developed using POM design pattern



1. Why Object repository ?

As per the rule of the automation, we should not hardcode[fixed]elements with in test Scripts, instead we should get elements from Object Repository , because in Agile process due to frequent requirement changes , modification & maintenance of elements is tedious job

EG : below is the example of Gmail application GUI changes



1. What is the advantages repository?
2. Reusability of elements, no need to write xpath & other locators again & again
3. Modification in Repository is easy, when GUI changes frequently
4. Maintenance is easy, because all the elements we kept in one place
5. Test Script Code Optimized via business reusable libraries
6. More Readability
7. Test Script development is faster due to business lib
8. Test Script is more robust
9. Handle Stale Elements Exception.
10. What is POM?

POM is a java design pattern preferred by google to develop object repository.

1. Why POM ?

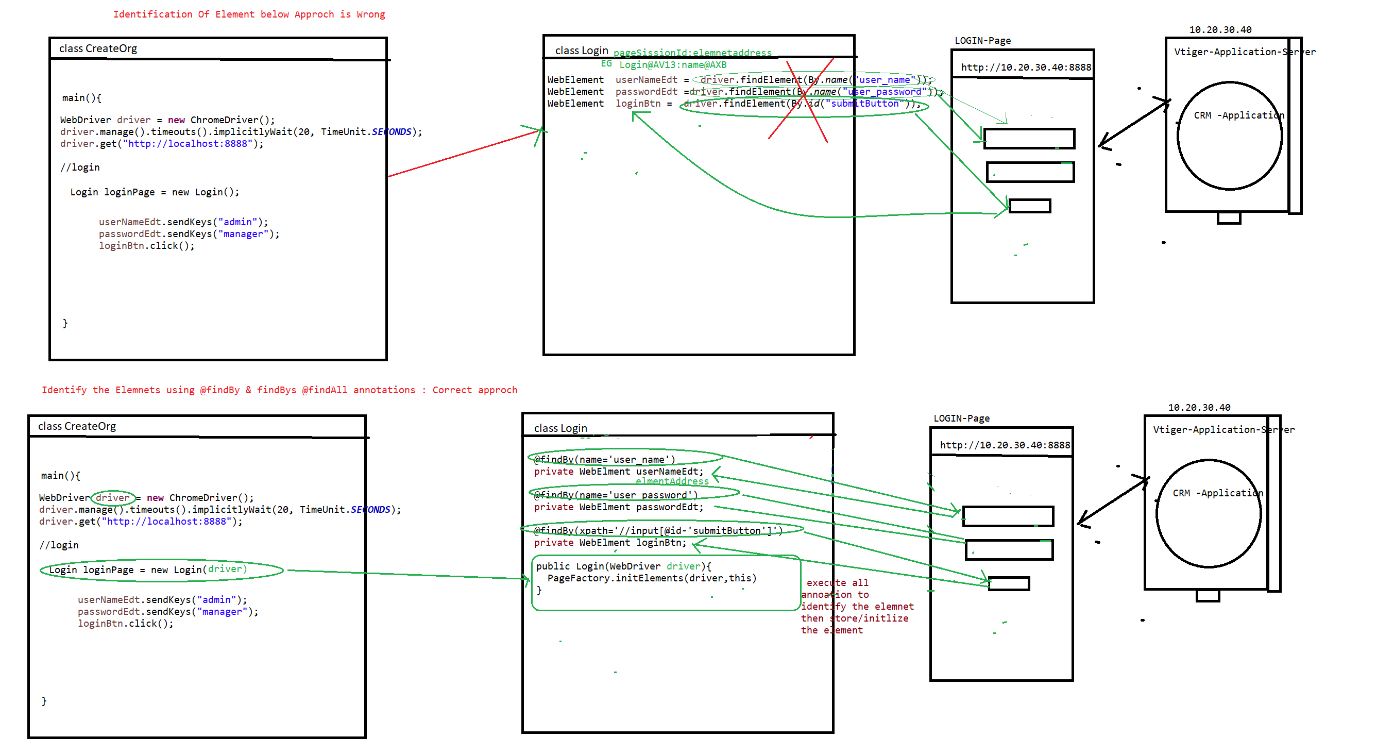
It’s a well-organized structured design pattern, where we can maintain all the web elements in page wise, due to POM design pattern maintains & modification is easy & faster.

1. Advantages of POM:
2. Well organized structure
3. Handle stale element exception.
4. maintains & modification of element is easy
5. We can directly store Web Elements in java class
6. Better fit for Agile processes
7. Support Auto heal feature
8. Why @FindBy(locator) annotation instead of driver.findElement(“locator”)

Ans : to avoid staleElementReferenceException

1. What is staleElementReferenceException ?

It’s one of selenium Execption , whenever webdriver try to identify an element , element was available in GUI, but at time of performing an action on the elements element was not recognized due to page got refreshed or elements may become old or element not attached to page in such case we get staleElementReferenceException



1. Rules of POM

Rule 1 : create separte java class for every page in a application & class name should be same page name

Rule 2 : Identify all the elements using @findBy & @findAll , @findbys annotations & store them in speific pom / java class (Element declartion)

Rule3 : For Every POM class create Constructor to get an Object of the class & initialize the Page Elements , in order to initialize all the page Elements we should use Pagefactory.initElement() (Element initialization)

Rule 4 : declare all the WebElements as private & provide getters methods to accesses elements in testScripts class [this processes is called Encapsulation]

Note : to create getters mtds inside the java class fallow below steps

🡺place cursor inside the class🡺 Right click🡺source 🡺generate getters & setters 🡺 select the getters check box 🡺 click on ok button

Rule 5 : Go to every page & identify the reusable business libraries & implement them in same POM class

1. Difference between POM & PageFactory design pattern?

POM is java design pattern, where will maintain all the Web element locator in well-organized manner

Pagefactory it’s an extended design pattern of POM , which is used to create an Object to POM classes , & at the time of object creation it will execute all @findBy @findbys annotation then initialize all the elements

Difference between @findBy , @findAll &@findBys annotation

All annotation available in Selenium webdriver, its traditional ways to identify the elements in GUI.

@findBy : used to identify the element using one locator or one condition

@findAll : it contains multiple @findBy annotation , it mean we can identify the same element using multiple locator (multiple conditions) , it will use OR condition during execution of locator

@findALL({ @findBy(@id=’username’) , @findBy(name=’user’)})

Private Webelements userNAmeEdt;

Note : using above concepts we can achieve **Autohealing** technique

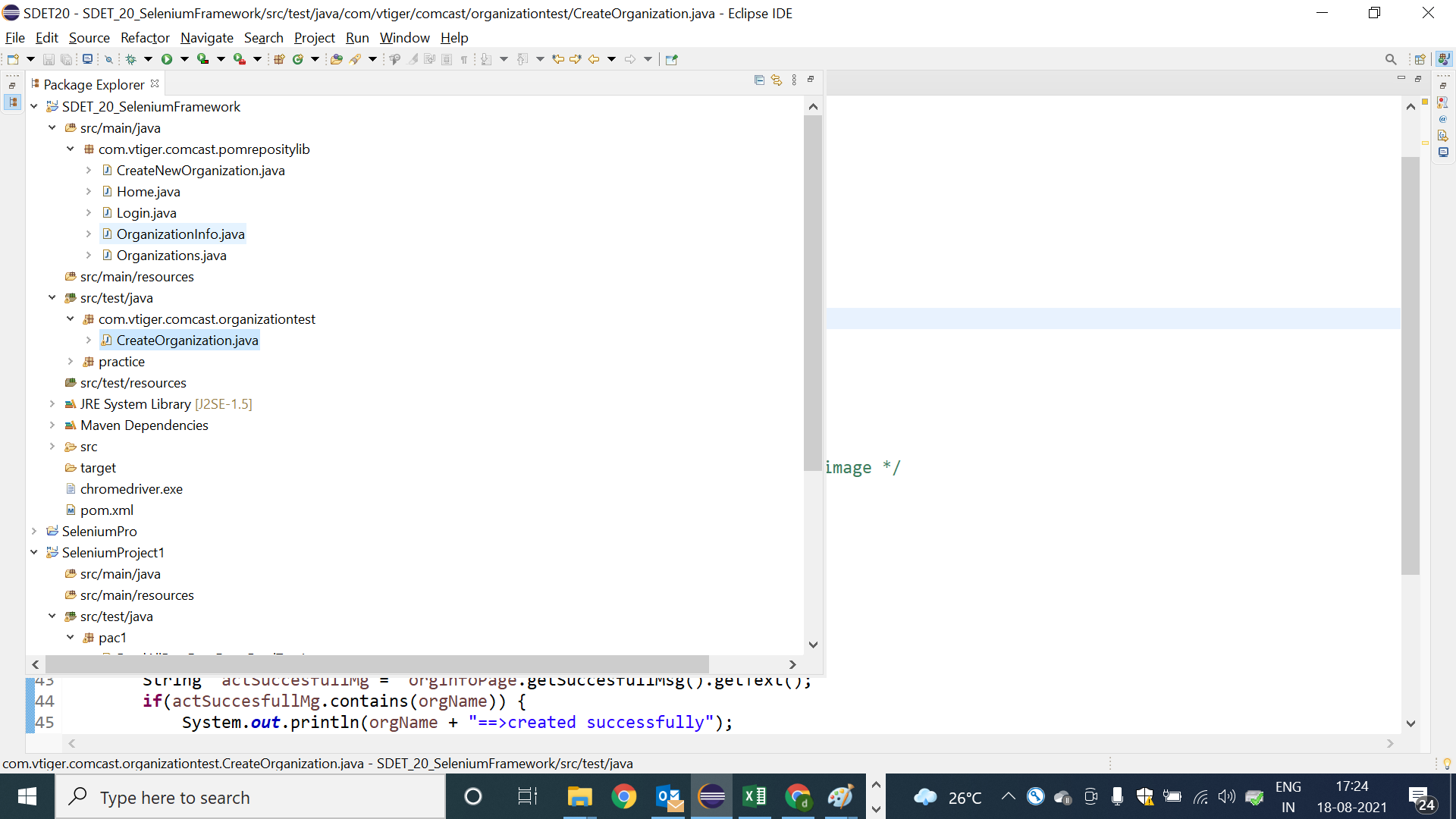
**AutoHealing** : during execution , if one locator fails to identify the element , it will retry to identify the same element using another locator

@FindBys : it contains multiple @findBy annotation , it mean we can identify the elements using multiple locator (multiple conditions) , it will use AND condition to during execution of locator

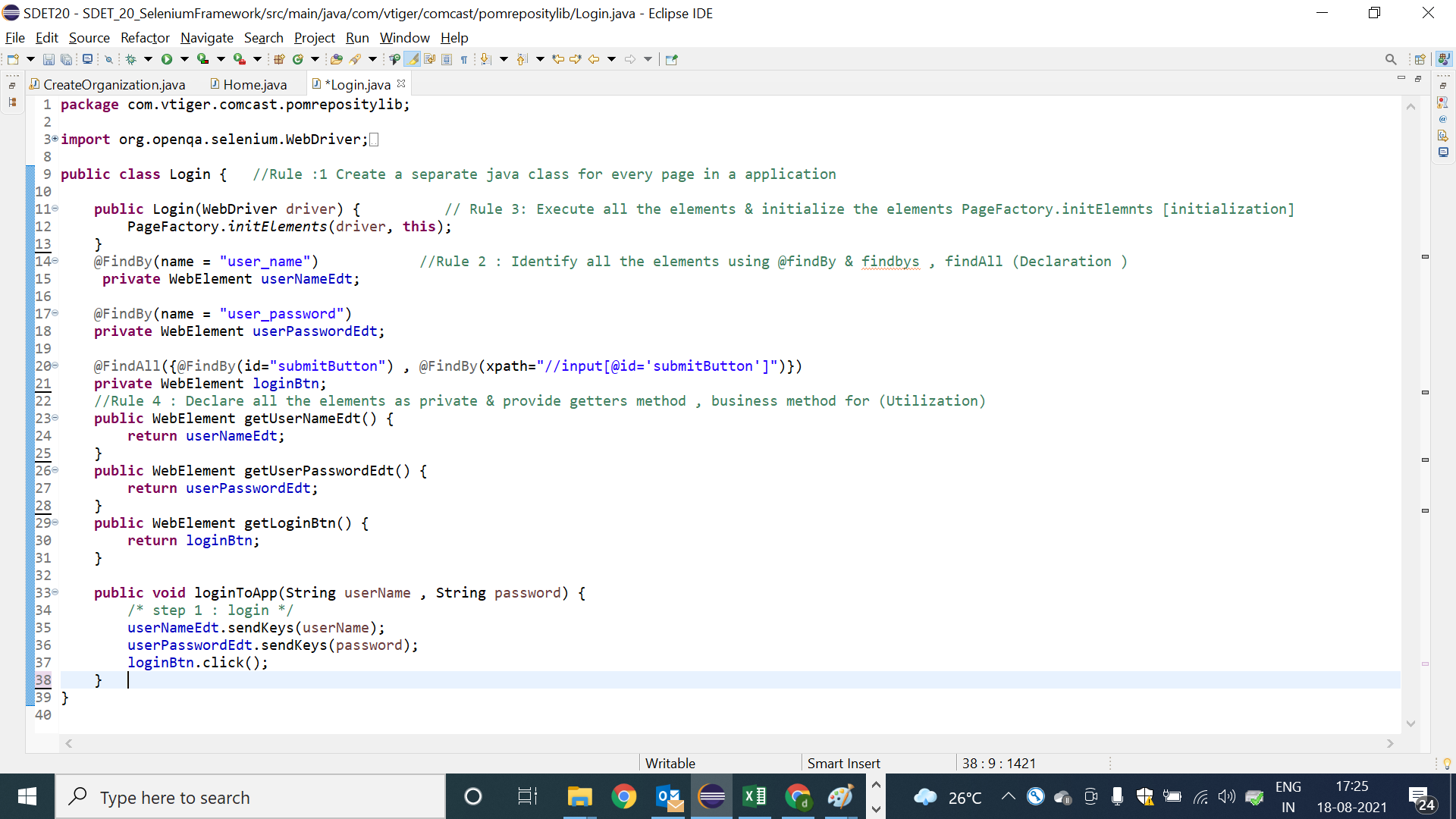
@findBys({ @findBy(@id=’username’) , @findBy(name=’user’)})

Private Webelements userNAmeEdt;

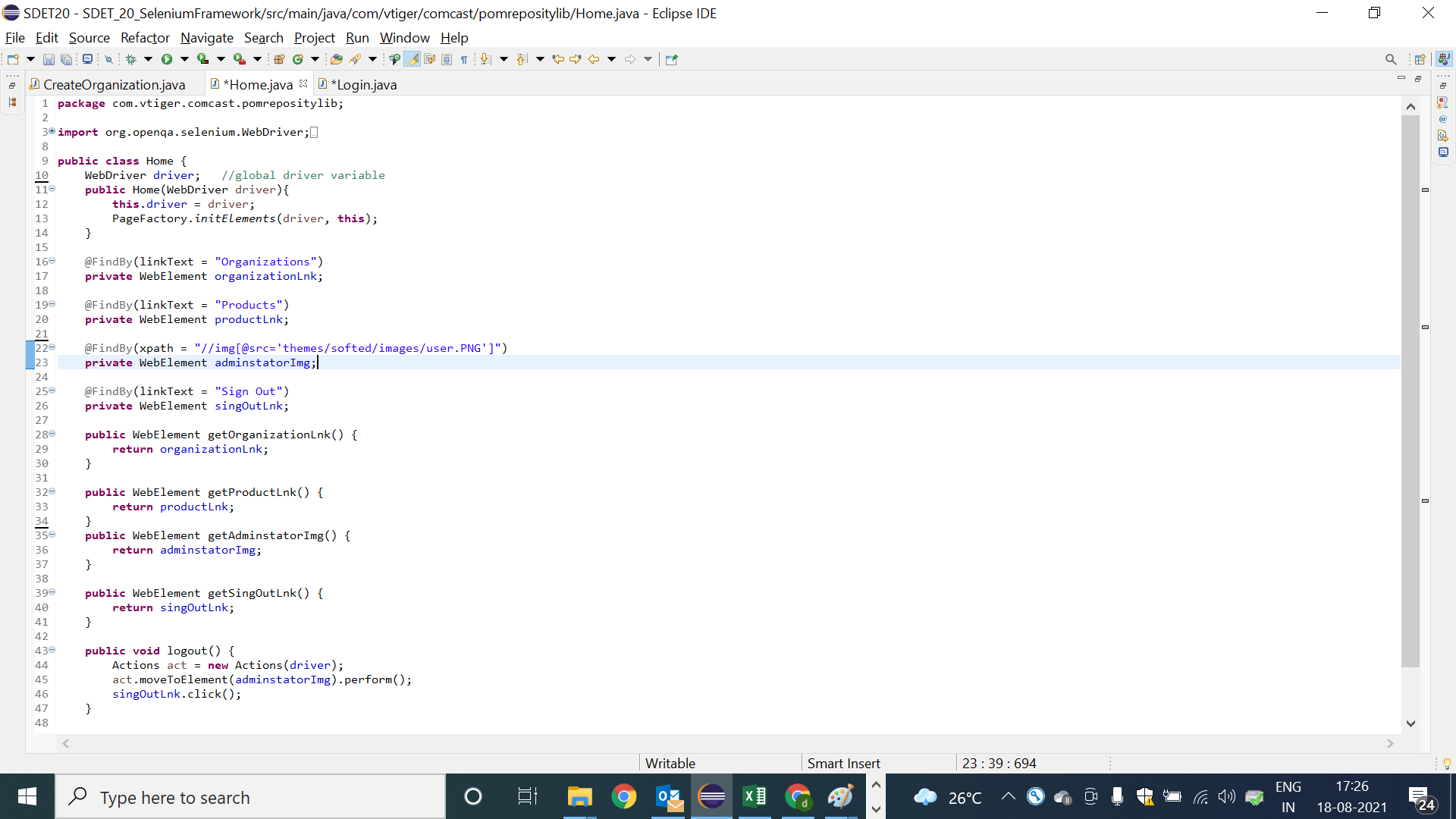
Project Structure



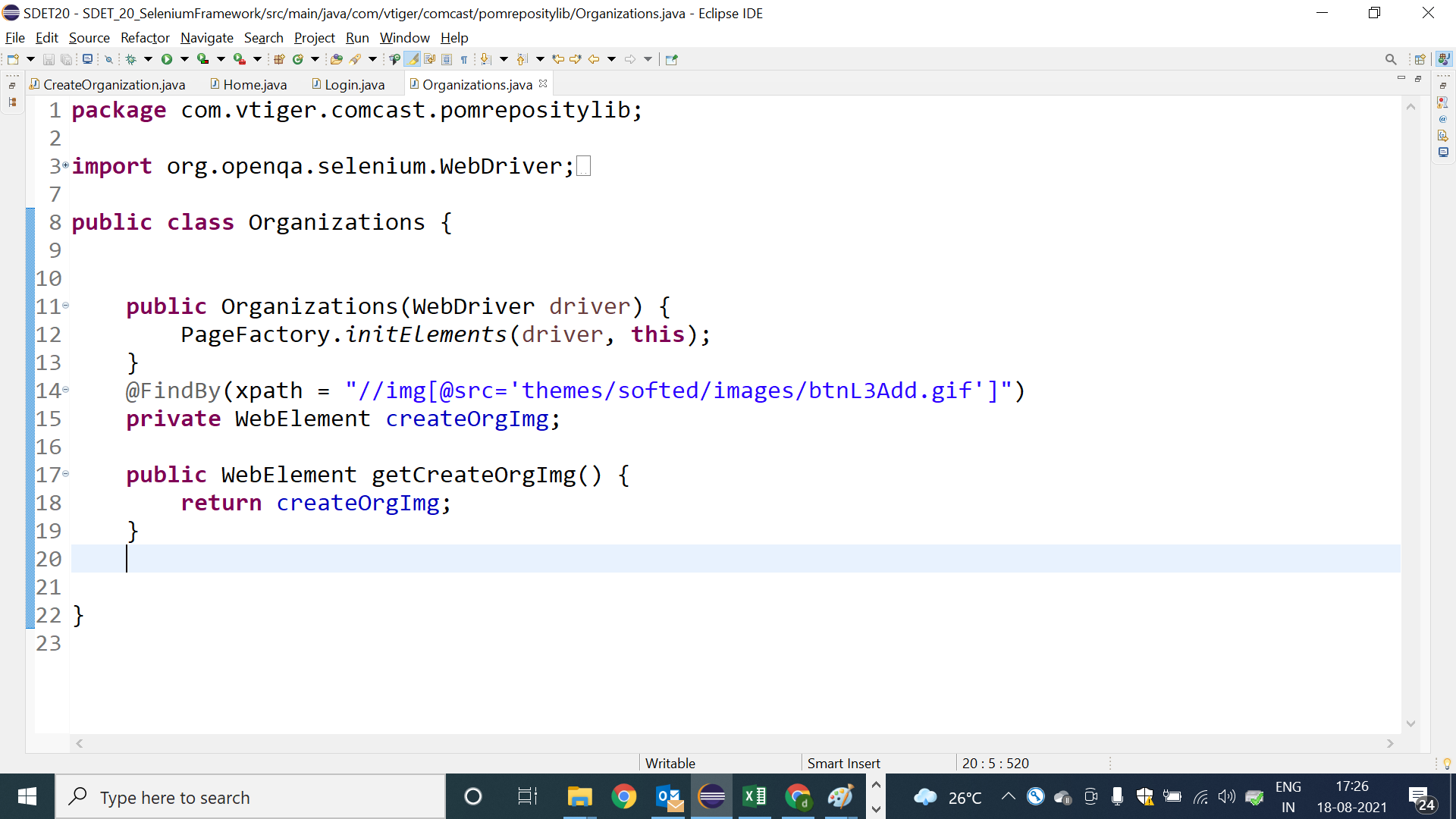
Pom Classes : Login



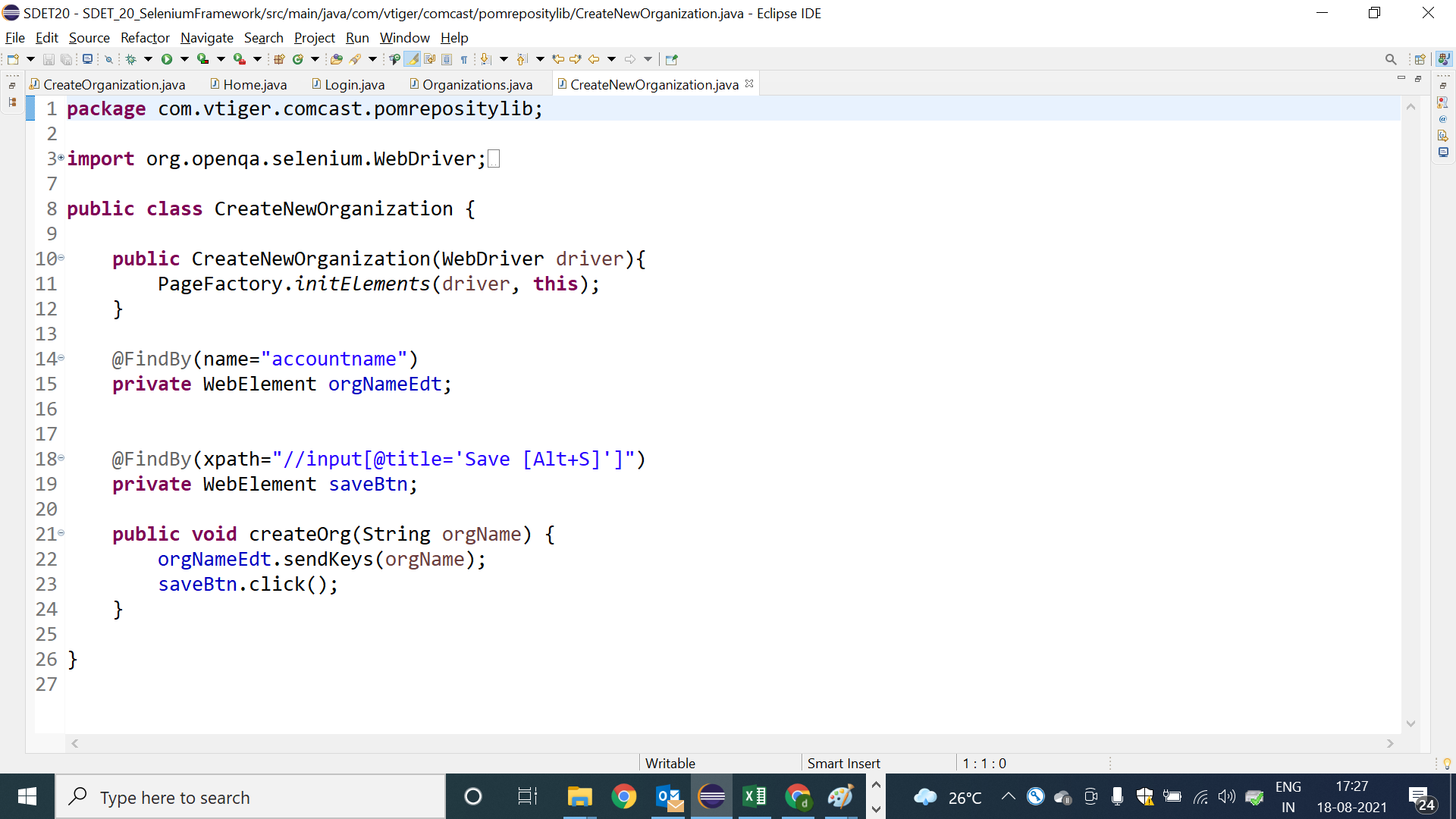
Pom Classes : Home



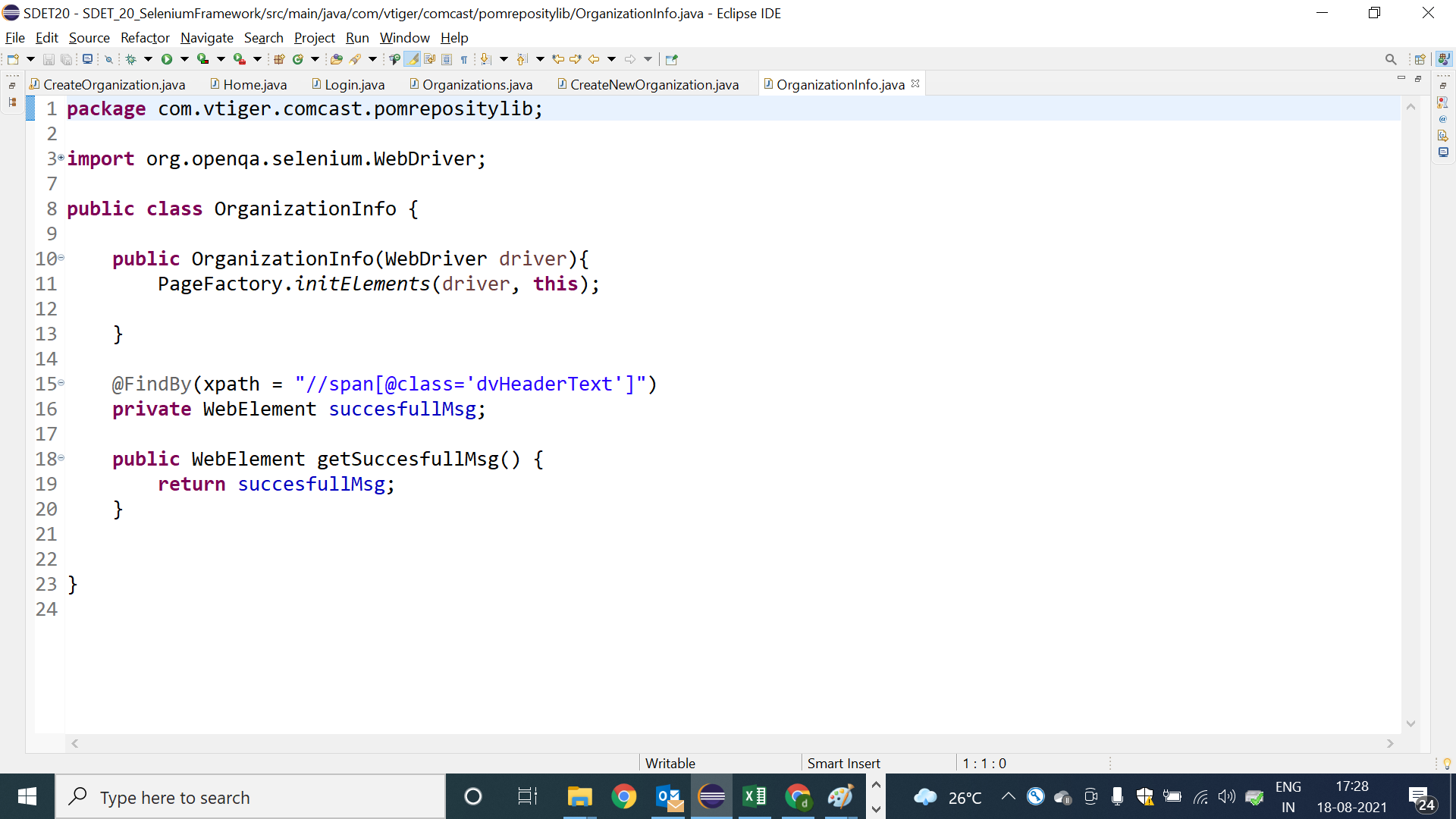
POM class : Organization



POM class : Create new Organization Page



POm class : OrganizationInfo



Test Scripts using POM class

