OUTPUT:

TEST case Scenario for Gmail login page

Gmail Login Screen Test Scenario

Test Scenarios For Gmail Login Page

As we don't have the official Google requirements for the login screen. We'll come up with our own requirement set for the login screen. We'll write the scenario based on these requirements.

- The username should contain a letter, number, and period.
- The username should not be left blank.
- The username should not be more than 40 characters.
- The username should not start with or contain any symbols.
- The password should be at least 6 characters.
- The password should contain a combination of letters, numbers, and symbols.
- The password should not contain spaces and periods.
- The password should not be more than 40 characters.

OUTPUT:

TEST case social media website :

Sr No.	Test Cases	Feature	Description	Steps To Execute	Expected Results
1	TC-01	User Interface	Check all the text boxes and buttons	Check Page	• UI should be perfect • Text boxes and button should be aligned
2	TC-02	Required Fields	Check the required fields by not filling any data.	1. Enter invalid username 2. Enter correct password 3. Click on Login Button	User should not log in and should show proper error message
3	TC-03	User Login	Check When passing a correct username and invalid password	1. Enter valid username 2. Enter incorrect password 3. Click on Login Button	User should not log in and should show proper error message
4	TC-04	User Interface	Check Keeping Password	2. Do not enter password 3. Click on Login Button	User should not log in and should show proper error message
5	TC-05	User Login	Check when pass correct email and password	1. Enter valid username 2. Enter valid password 3. Click on Login Button	User should log in
6	TC-06	User Login	Check if the password is entered in encrypted	2. Enter password 3. Click on Login Button	Password is entered in encrypted form
7	TC-07	Signup Option for new users	Check whether the signup link for the new user is working Verify user should get an error message	Click Signup link 1. Click on the Forgot password link.	Clicking signup link takes the user to signup page successfully

Test Case For Facebook Timeline Login

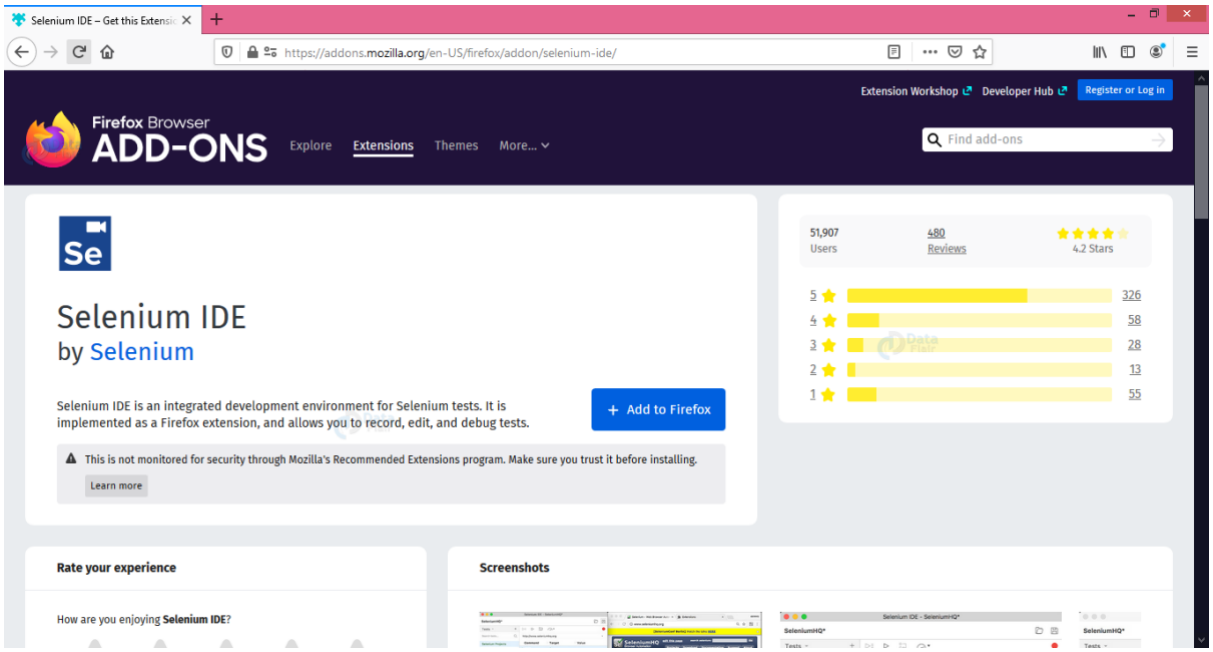
OUTPUT:

Defect report:

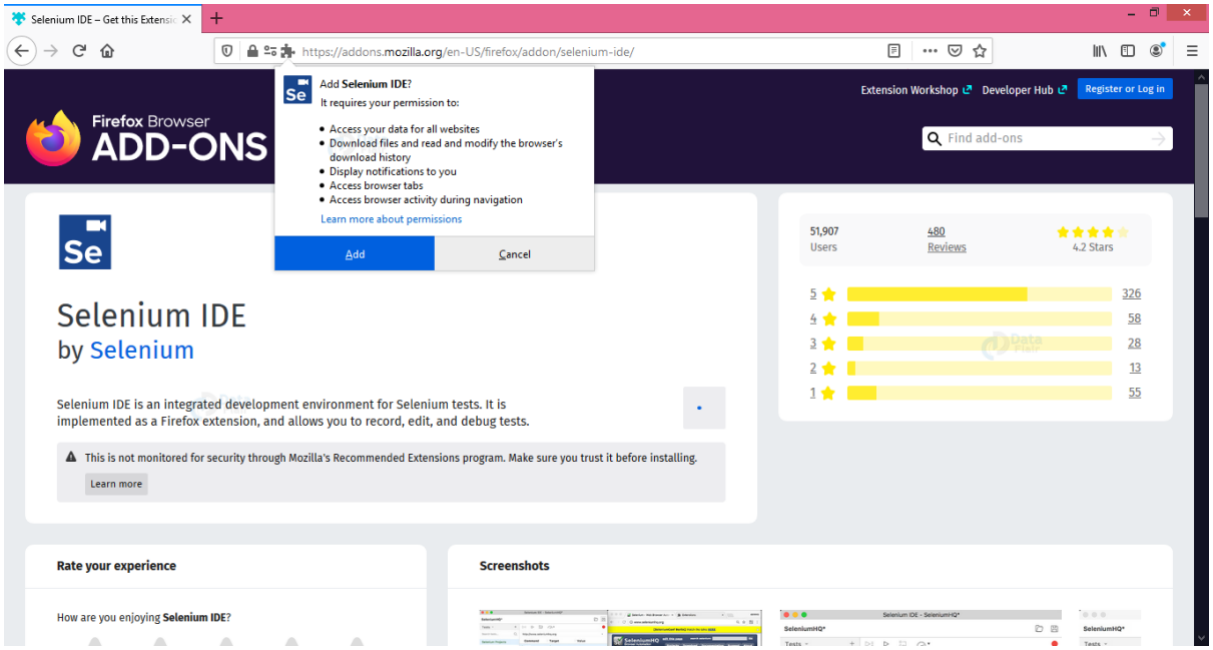
	A	B	C
1	**Category**	**Label**	**Value**
2	Bug ID	ID number	#123
3		Name	CART - Unable to add new item to my cart
4		Reporter	Mike A
5		Submit Date	03/04/16
6	Bug overview	Summary	When my cart contains one item, I am unable to add a second item via the add to cart button on a product page
7		URL	www.example.com/product/abc
8		Screenshot	www.example.com/screenshot123
9	Environment	Platform	Macintosh
10		Operating System	OS X 10.12.0
11		Browser	Chrome 53
12	Bug details	Steps to reproduce	add one item to cart > go to product abc via the search bar > add new item to cart via "add to cart" button (see screenshot) > go to cart
13		Expected result	The cart should contain 2 items
14		Actual result	The cart contains only 1 item
15		Description	/
16	Bug tracking	Severity	Major
17		Assigned to	/
18		Priority	High
19	Notes	Notes	/

Steps to Install Selenium IDE

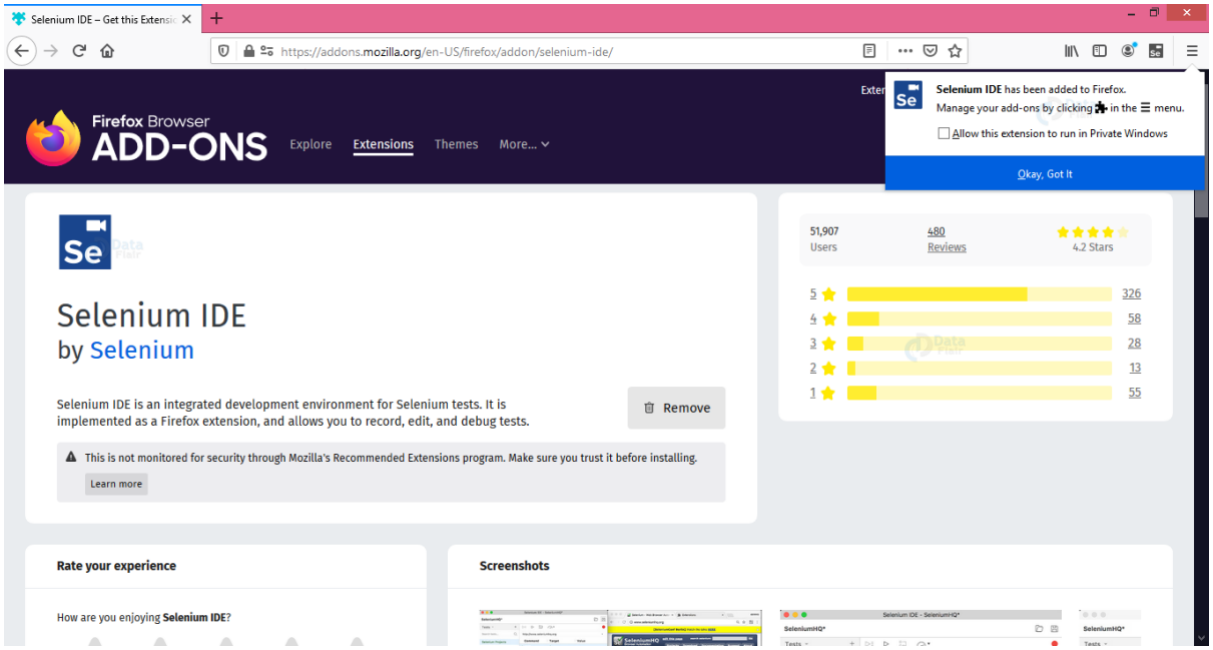
- Open the Firefox Browser and redirect to https://addons.mozilla.org/en-US/firefox/addon/selenium-ide/. A window will open up. Click ‘Add to Firefox.’



- Wait for the download to complete. After that, click “Add.”



- Click OK once the installation completes.



- Then you will see a Selenium IDE icon on the desktop; just double click on it to open.

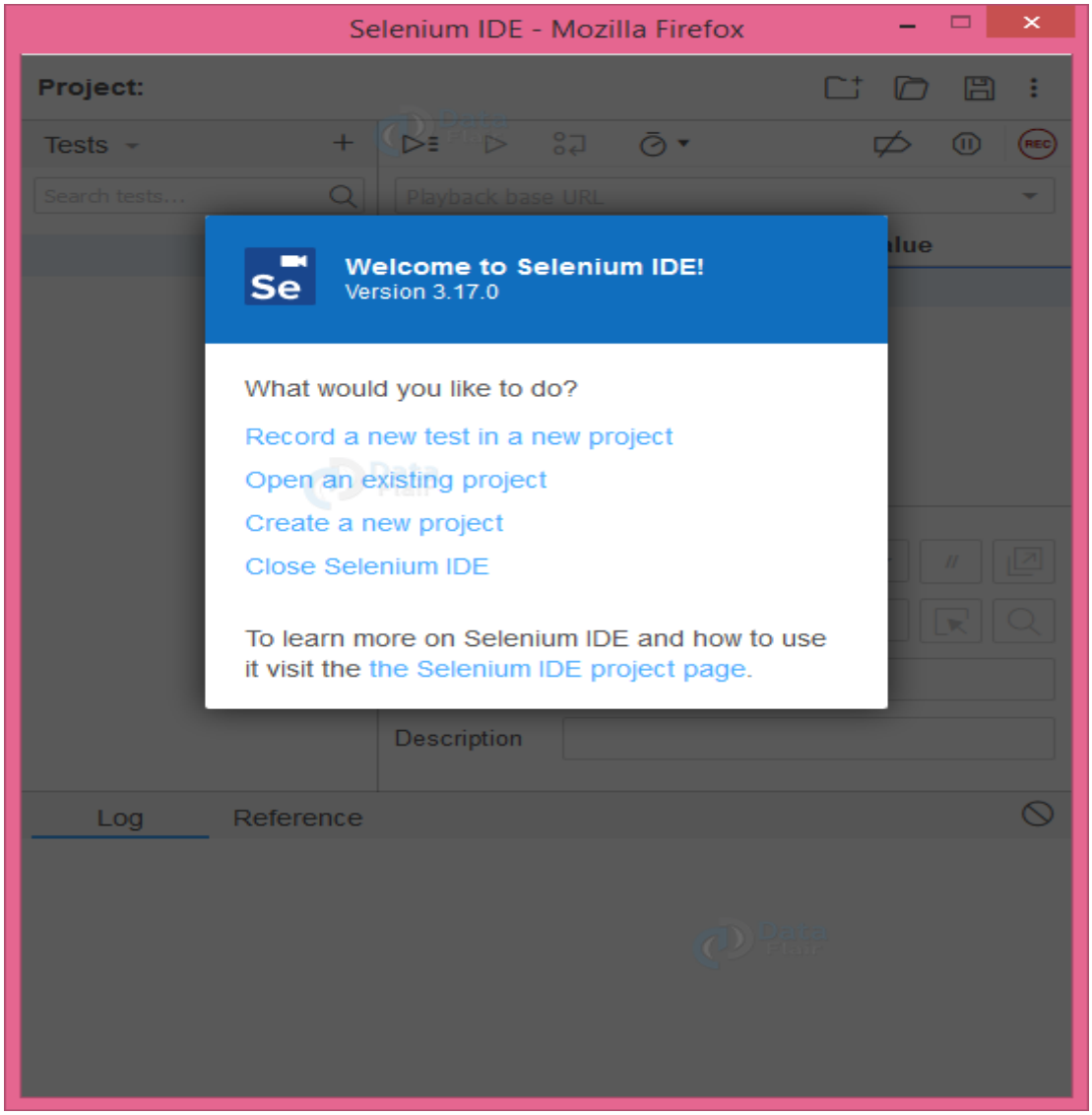


Table of Contents

Table of Contents for a SRS Document

1.INTRODUCTION

1.1 PURPOSE

The purpose of this document is to build an online system to manage flights and passengers to ease the flight management.

1.2 DOCUMENT CONVENTIONS

This document uses the following conventions.

DB	Database
DDB	Distributed Database
ER	Entity Relationship

1.3 INTENDED AUDIENCE AND READING SUGGESTIONS

This project is a prototype for the flight management system and it is restricted within the college premises. This has been implemented under the guidance of college professors. This project is useful for the flight management team and as well as to the passengers.

1.4 PROJECT SCOPE

The purpose of the online flight management system is to ease flight management and to create a convenient and easy-to-use application for passengers, trying to buy airline tickets. The system is based on a relational database with its flight management and reservation functions. We will have a database server supporting hundreds of major cities around the world as well as thousands of flights by various airline companies. Above all, we hope to provide a comfortable user experience along with the best pricing available.

1.5 REFERENCES

- https://krazytech.com/projects
- Fundamentals of database systems by ramez elmarsi and shamkant b.navathe

2. OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

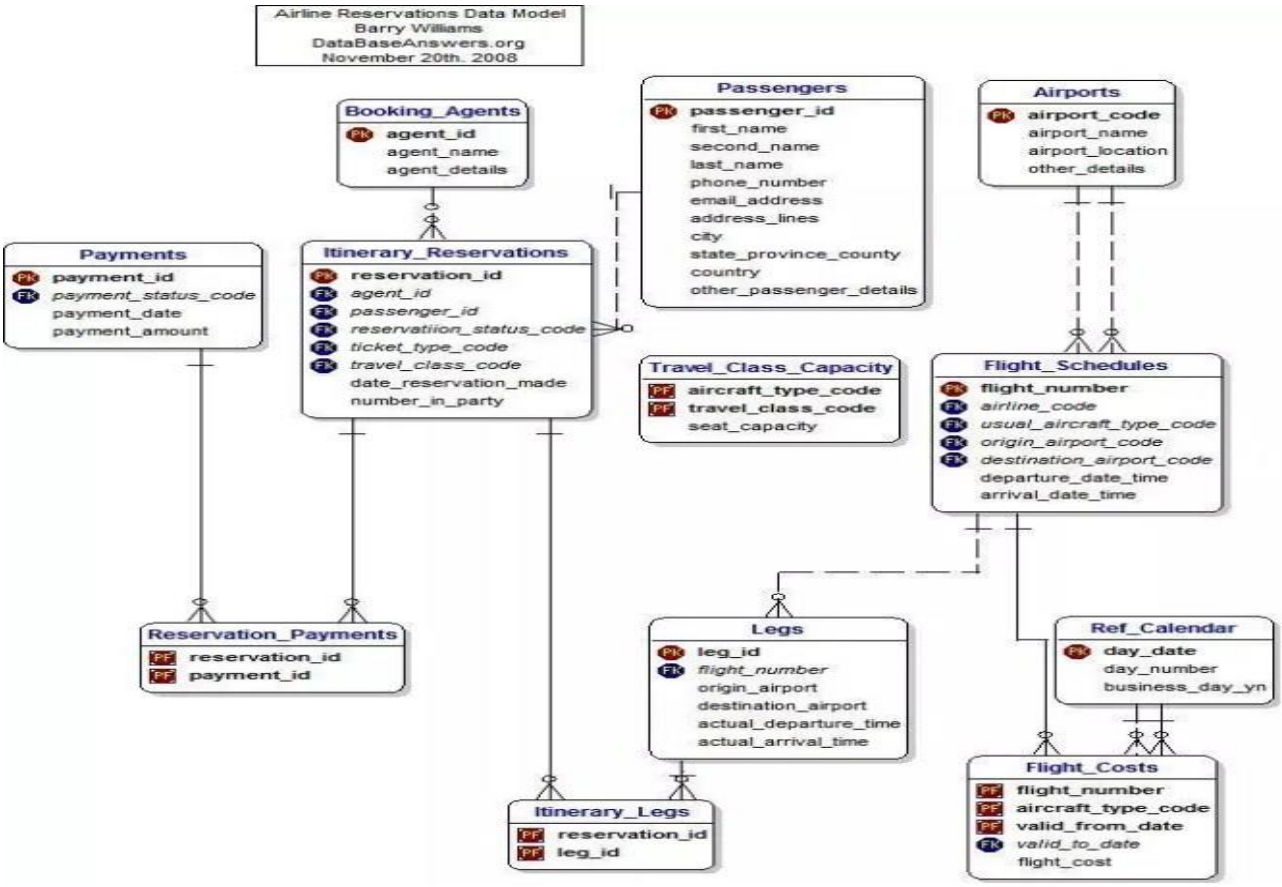
A distributed airline database system stores the following information.

- Flight details:
  It includes the originating flight terminal and destination terminal, along with the stops in between, the number of seats booked/available seats between two destinations etc.
- Customer description:
  It includes customer code, name, address and phone number. This information may be used for keeping the records of the customer for any emergency or for any other kind of information.
- Reservation description:
  It includes customer details, code number, flight number, date of booking, date of travel.

2.2 PRODUCT FEATURES

The major features of airline database system as shown in below entity relationship model (ER model)

The diagram shows the layout of airline database system



– entity–relationship model

2.3 USER CLASS and CHARACTERISTICS

A route from city A to city B is a sequence of connecting flights from A to B such that: a) there are at most two connecting stops, excluding the starting city and destination city of the trip, b) the connecting time is between one to two hours. The system will support two types of user privileges, Customer, and Employee. Customers will have access to customer functions, and the employees will have access to both customer and flight management functions. The customer should be able to do the following functions:

- Make a new reservation
  - One-way
  - Round-Trip
  - Multi-city
  - Flexible Date/time
  - Confirmation
- Cancel an existing reservation
- View his itinerary

The Employee should have following management functionalities:

- CUSTOMER FUNCTIONS.
  - Get all customers who have seats reserved on a given flight.
  - Get all flights for a given airport.
  - View flight schedule.
  - Get all flights whose arrival and departure times are on time/delayed.
  - Calculate total sales for a given flight.
- ADMINISTRATIVE
  - Add/Delete a flight
  - Add a new airport
  - Update fare for flights.
  - Add a new flight leg instance.
  - Update departure/arrival times for flight leg instances.

Each flight has a limited number of available seats. There are a number of flights which depart from or arrive at different cities on different dates and time.

2.4 OPERATING ENVIRONMENT

- distributed database
- client/server system
- Operating system: Windows.
- database: sql+ database
- platform: vb.net/Java/PHP

2.5 DESIGN and IMPLEMENTATION CONSTRAINTS

1. The global schema, fragmentation schema, and allocation schema.
2. SQL commands for above queries/applications
3. How the response for application 1 and 2 will be generated. Assuming these are global queries. Explain how various fragments will be combined to do so.
4. Implement the database at least using a centralized database management system.

2.6 ASSUMPTION DEPENDENCIES

Let us assume that this is a distributed airline management system and it is used in the following application:

- A request for booking/cancellation of a flight from any source to any destination, giving connected flights in case no direct flight between the specified Source-Destination pair exist.
- Calculation of high fliers (most frequent fliers) and calculating appropriate reward points for these fliers.

3. SYSTEM FEATURES

DESCRIPTION and PRIORITY

The airline reservation system maintains information on flights, classes of seats, personal preferences, prices, and bookings. Of course, this project has a high priority because it is very difficult to travel across countries without prior reservations.

STIMULUS/RESPONSE SEQUENCES

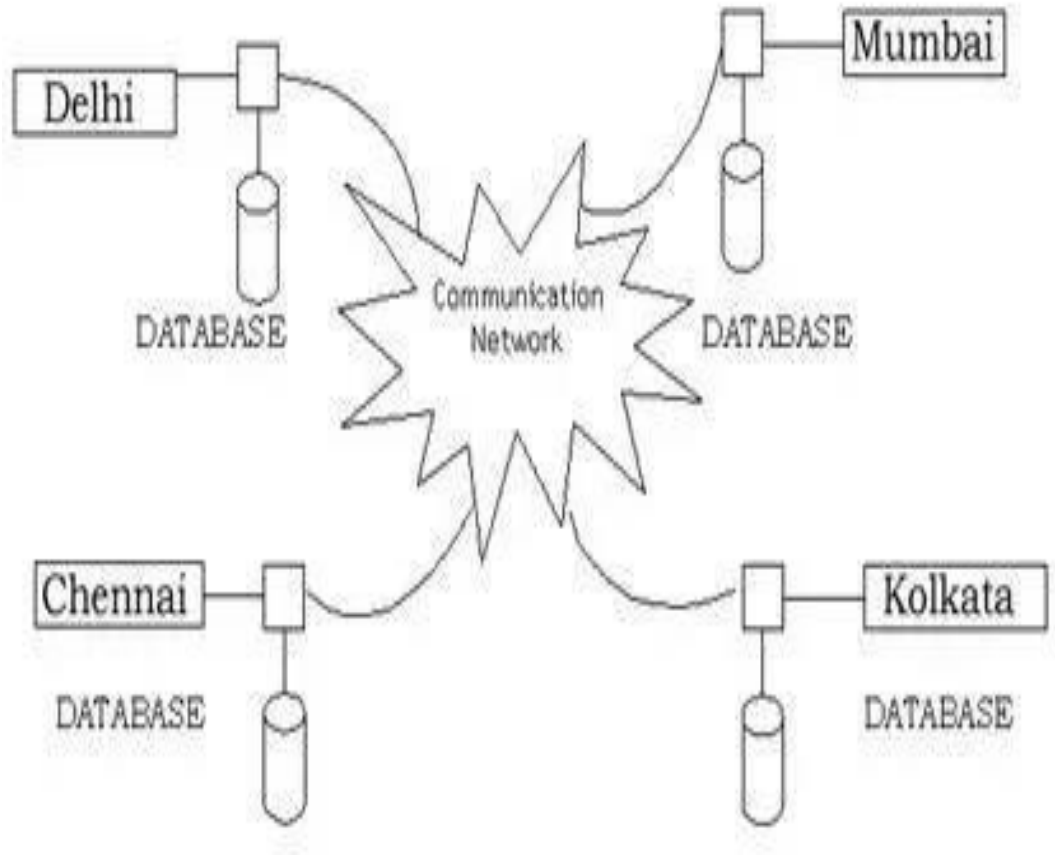
Search for Airline Flights for two Travel cities
Displays a detailed list of available flights and make a "Reservation" or Book a ticket on a particular flight.
Cancel an existing Reservation.

FUNCTIONAL REQUIREMENTS

Other system features include:

DISTRIBUTED DATABASE:

Distributed database implies that a single application should be able to operate transparently on data that is spread across a variety of different databases and connected by a communication network as shown in below figure.



CLIENT/SERVER SYSTEM

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).
A client/server system is a distributed system in which,

- Some sites are client sites and others are server sites.
- All the data resides at the server sites.
- All applications execute at the client sites.

4.EXTERNAL INTERFACE REQUIREMENTS

4.1 USER INTERFACES

- Front-end software: Vb.net version
- Back-end software: SQL+

4.2 HARDWARE INTERFACES

- Windows.
- A browser which supports CGI, HTML & Javascript.

4.3 SOFTWARE INTERFACES

Following are the software used for the flight management online application.

Software used	Description
Operating system	We have chosen Windows operating system for its best support and user-friend
Database	To save the flight records, passengers records we have chosen SQL+ database.
VB.Net	To implement the project we have chosen Vb.Net language for its more interact support.

4.4 COMMUNICATION INTERFACES

This project supports all types of web browsers. We are using simple electronic forms for the reservation forms, ticket booking etc.

5. NONFUNCTIONAL REQUIREMENTS

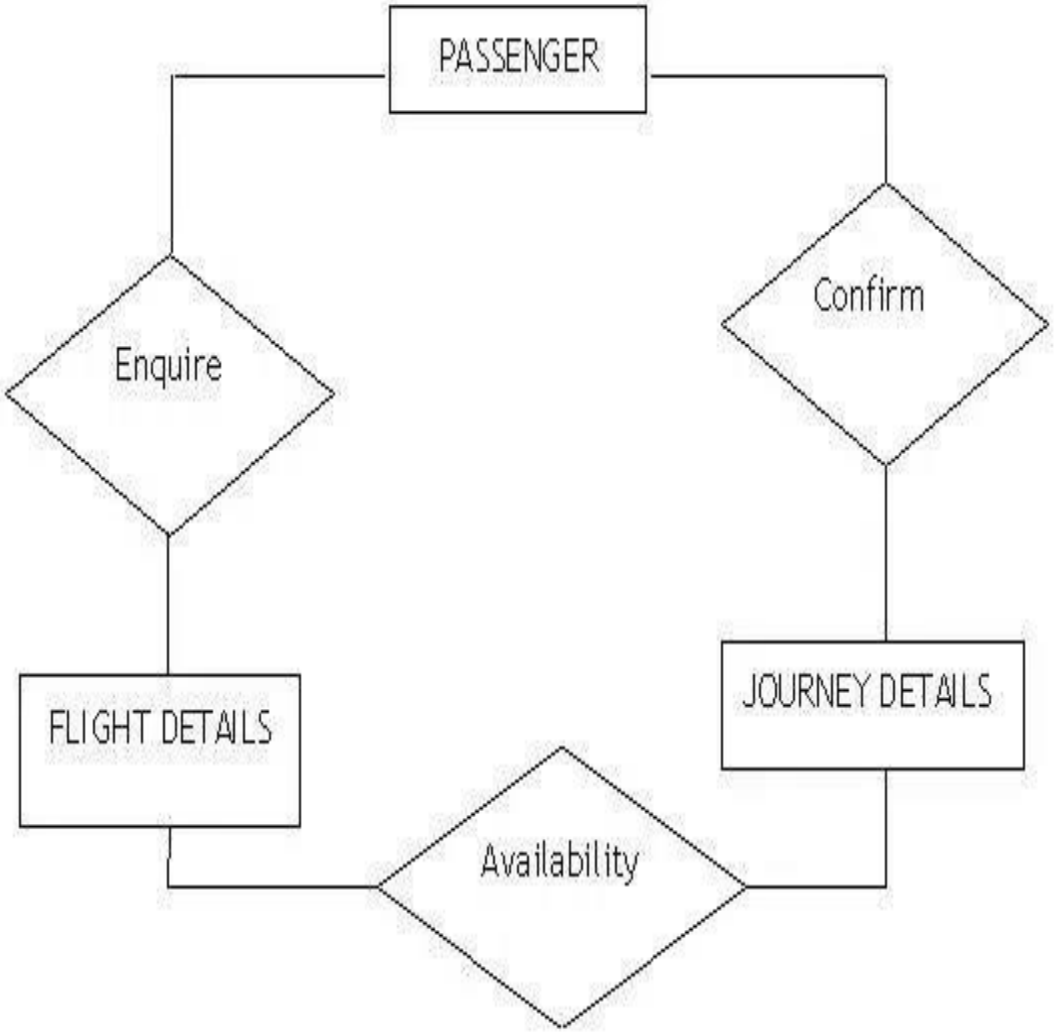
5.1 PERFORMANCE REQUIREMENTS

The steps involved to perform the implementation of airline database are as listed below.

A) E-R DIAGRAM

The E-R Diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.

- **ENTITIES:** Which specify distinct real-world items in an application.
- **PROPERTIES/ATTRIBUTES:** Which specify properties of an entity and relationships.
- **RELATIONSHIPS:** Which connect entities and represent meaningful dependencies between them.



B) NORMALIZATION:

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most practical purposes. It should be considered only after a thorough analysis and complete understanding of its implications.

5.2 SAFETY REQUIREMENTS

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

5.3 SECURITY REQUIREMENTS

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

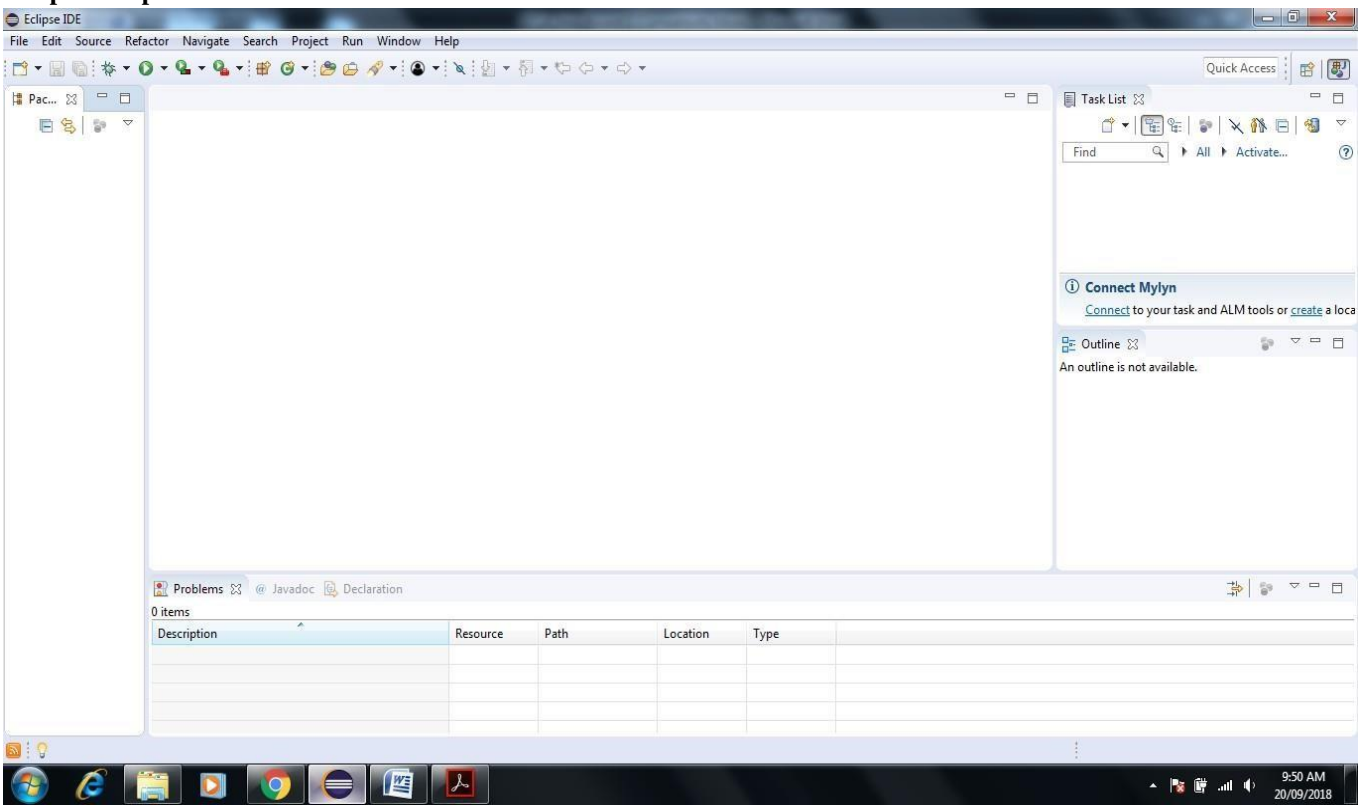
5.4 SOFTWARE QUALITY ATTRIBUTES

- **AVAILABILITY:** The flight should be available on the specified date and specified time as many customers are doing advance reservations.
- **CORRECTNESS:** The flight should reach start from correct start terminal and should reach the correct destination.
- **MAINTAINABILITY:** The administrators and flight in chargers should maintain correct schedules of flights.
- **USABILITY:** The flight schedules should satisfy a maximum number of customer needs.

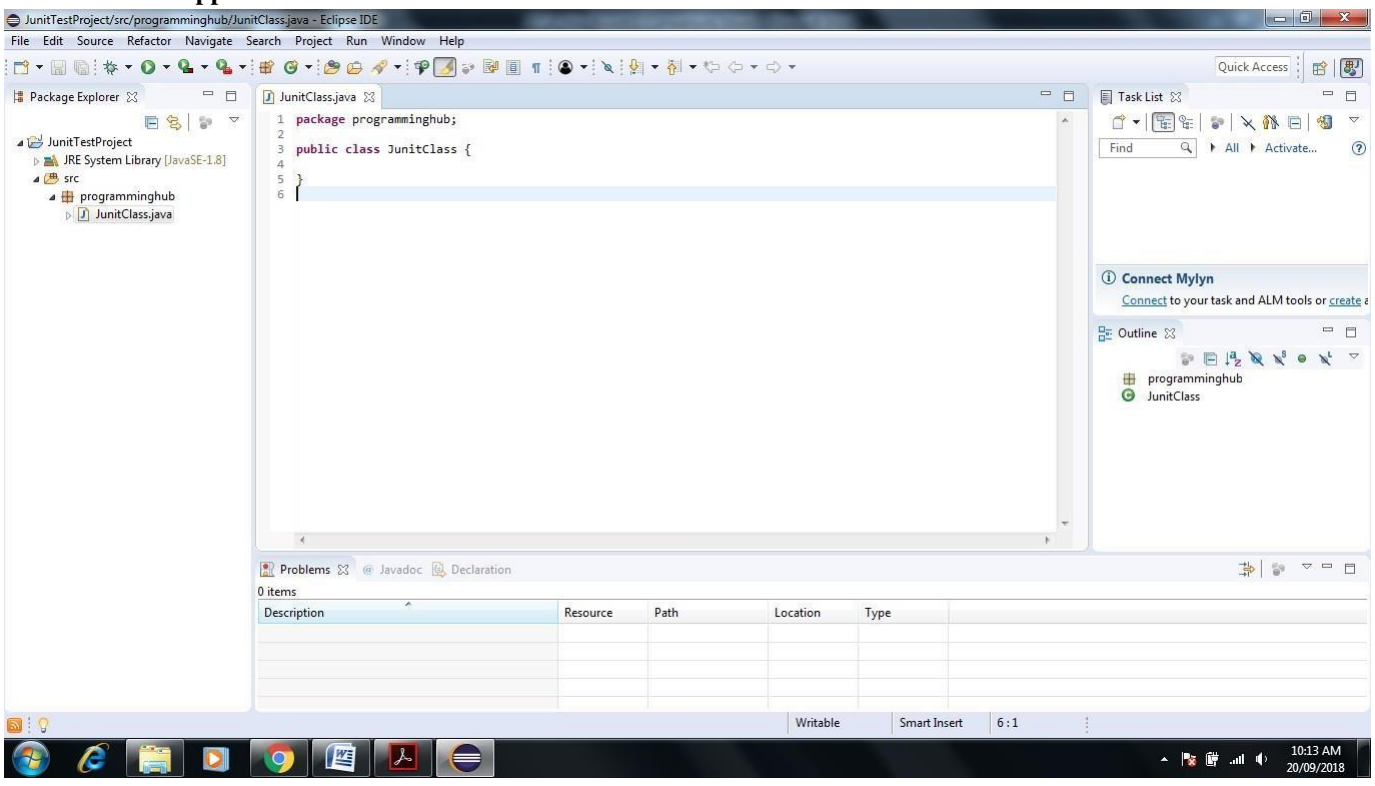
How to Create Simple Junit Test in Eclipse IDE

1. Download JDK 1.8 and Eclipse latest version here we are using eclipse-java-photon-R-win32.

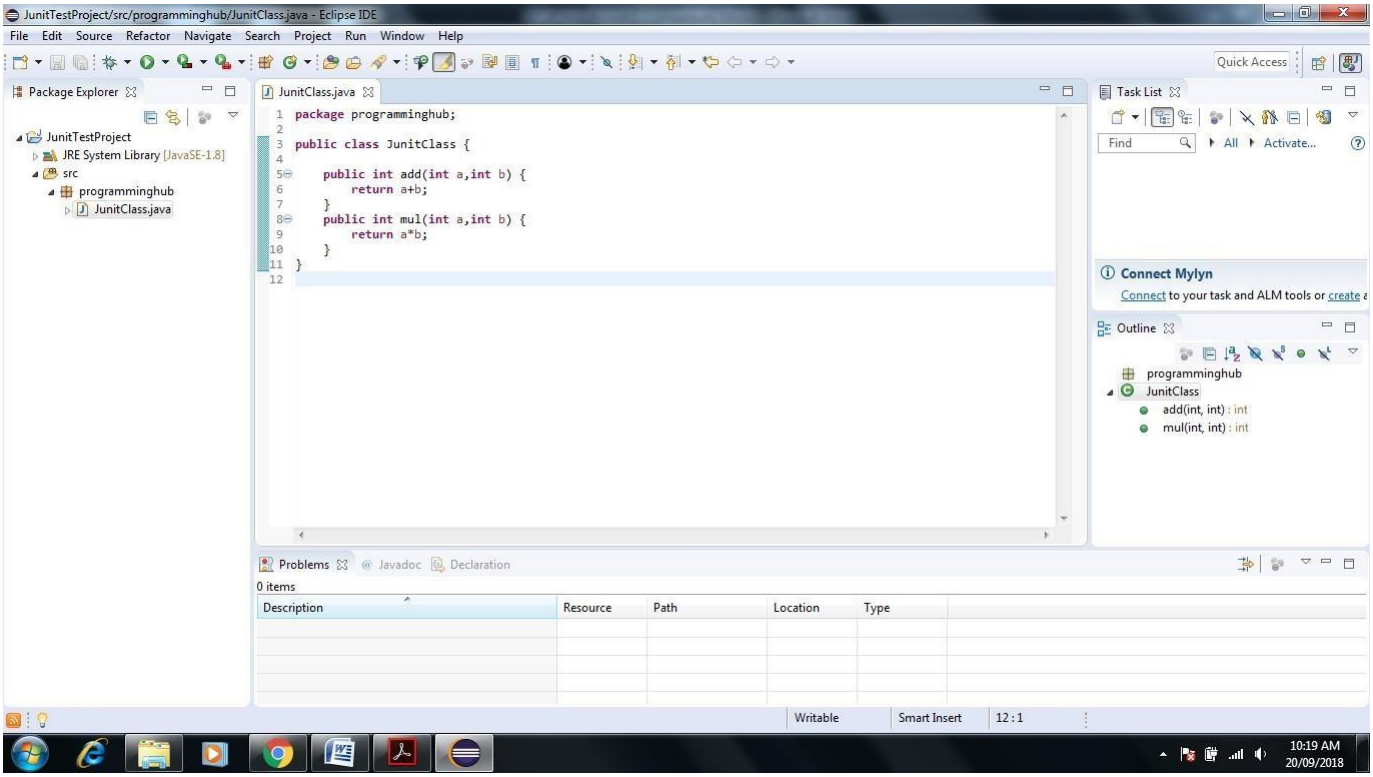
2. Open Eclipse IDE



Next screen will appear

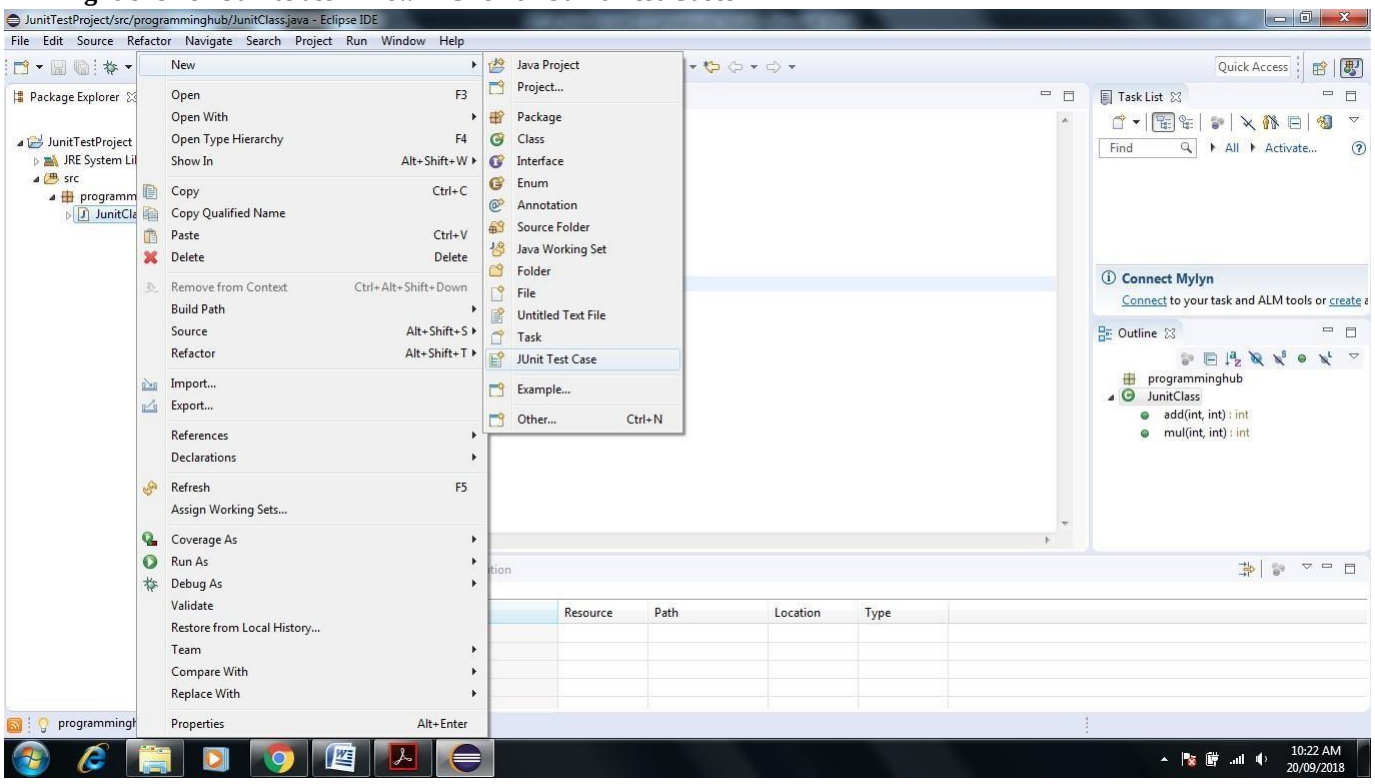


Write a small program with only two functions Add and Multiplication

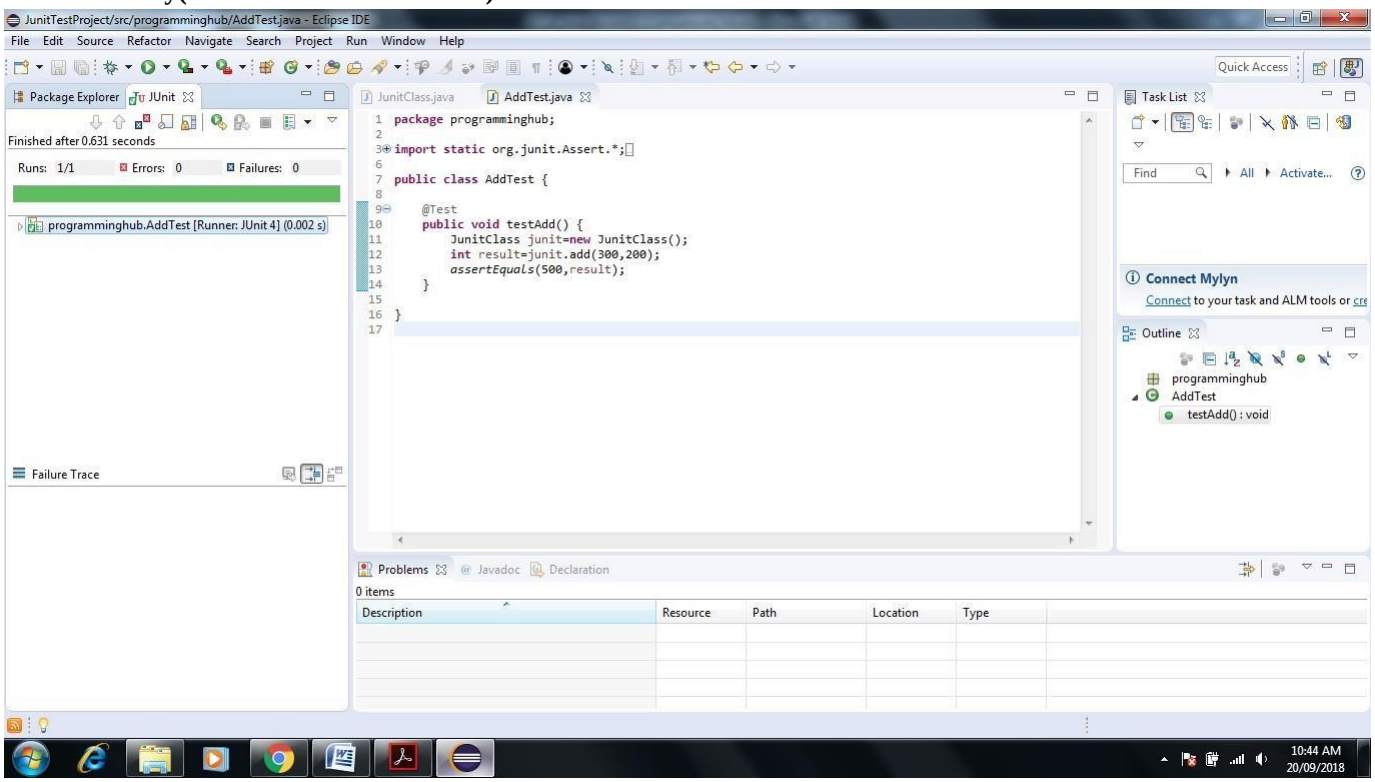


Write Test Cases for Java Program

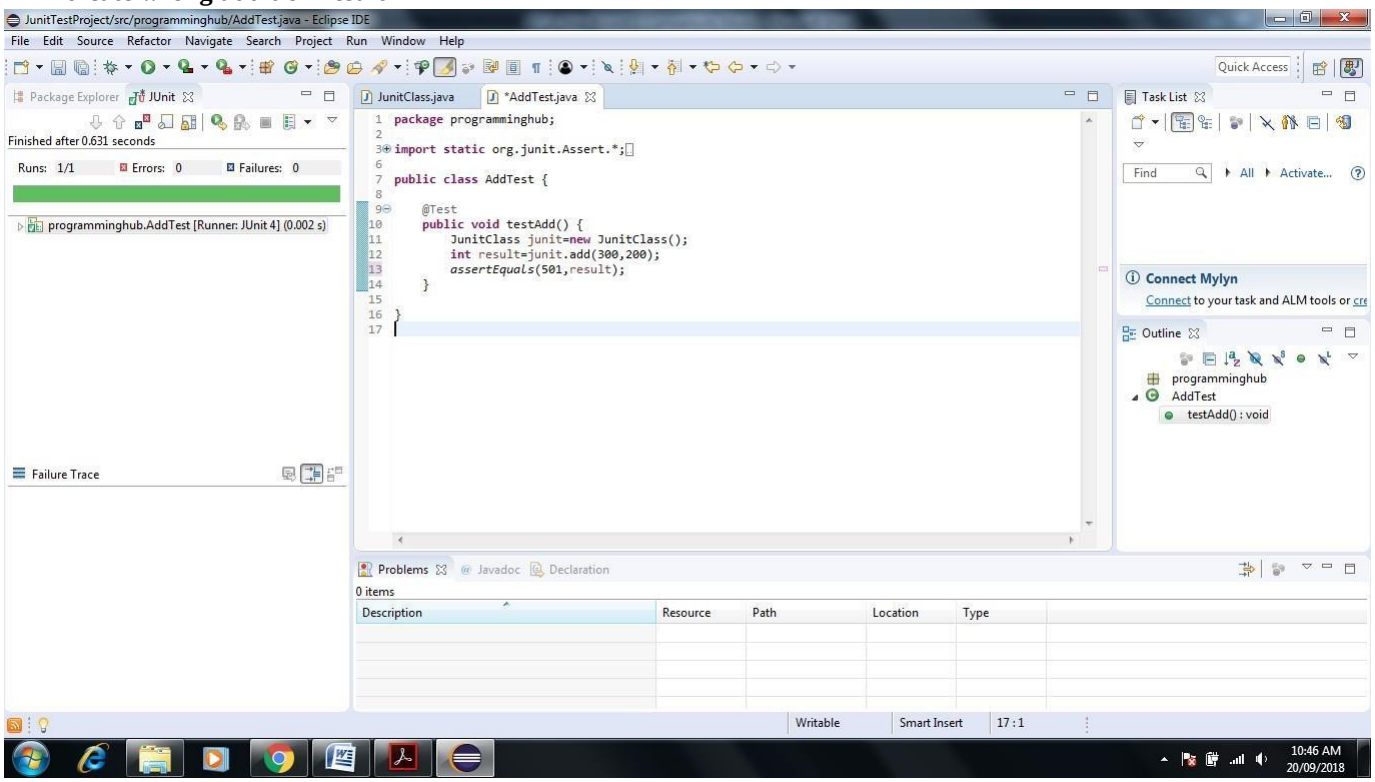
Right click on Junitclass-> New-> Click on Junit Test Cases



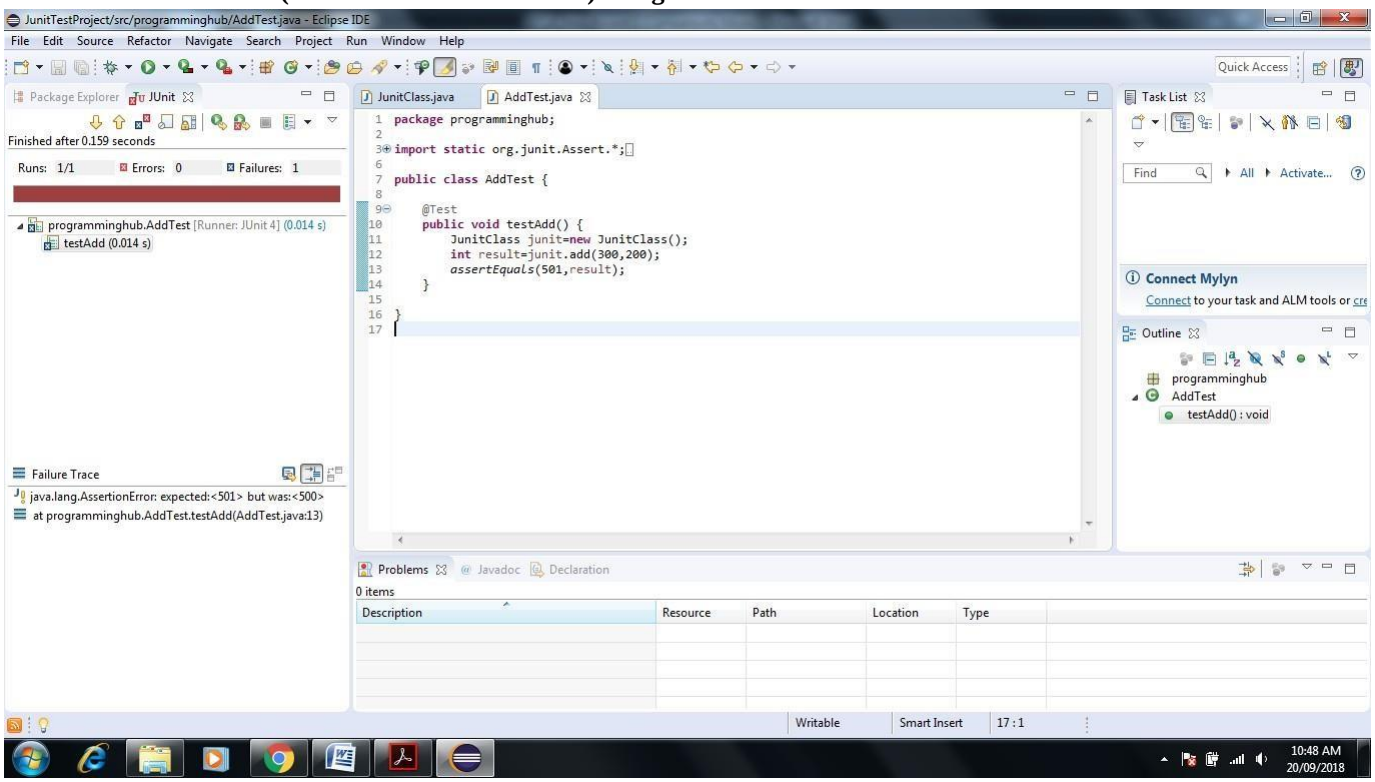
Result of test case is as follows. It shows 0 error and 0 failure and green color test bar which means that test case has run successfully(Green Color Bar Indicate)



Let us purposely give wrong input in assertEquals method or unexpected result here we write 501 instead of 500 indicate wrong addition result

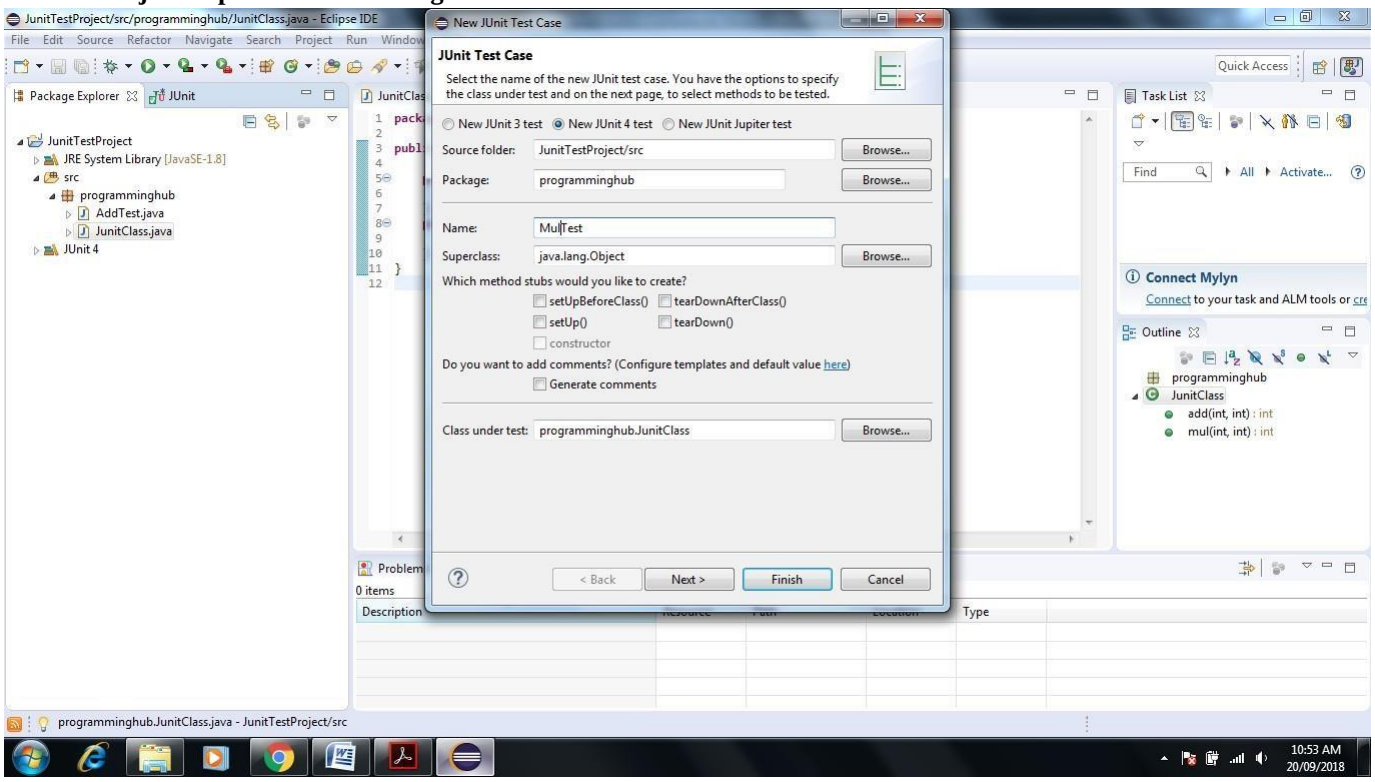


Now test case should fail.(Brown Color Bar Indicate) So again run AddTest as follows

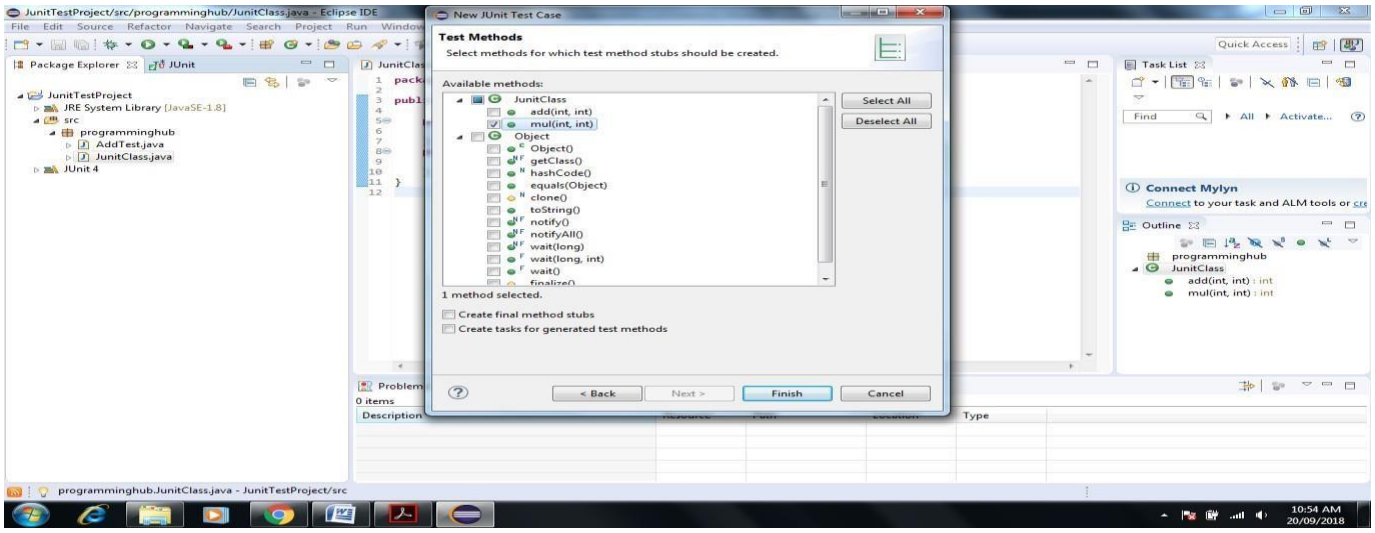


Similarly you can Create Test case for Multiplication Function

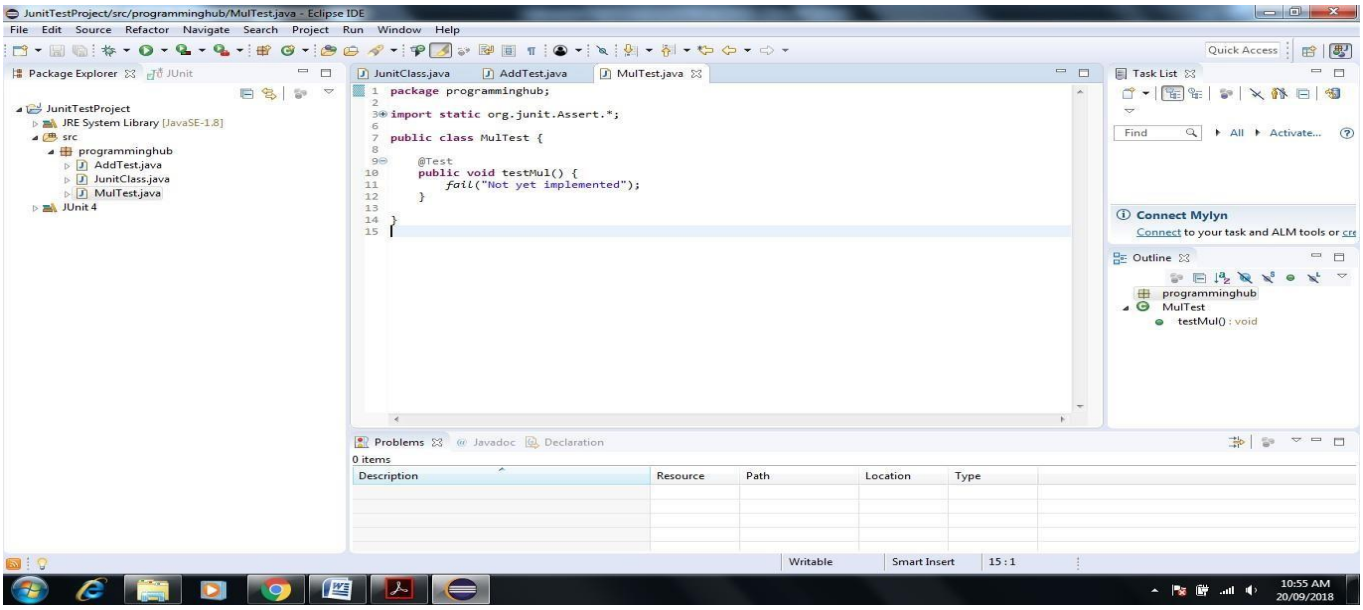
Click on Project Explorer Screen-> Right Click on JunitClass->New->JUnit Test Case-> Give name MulTest.



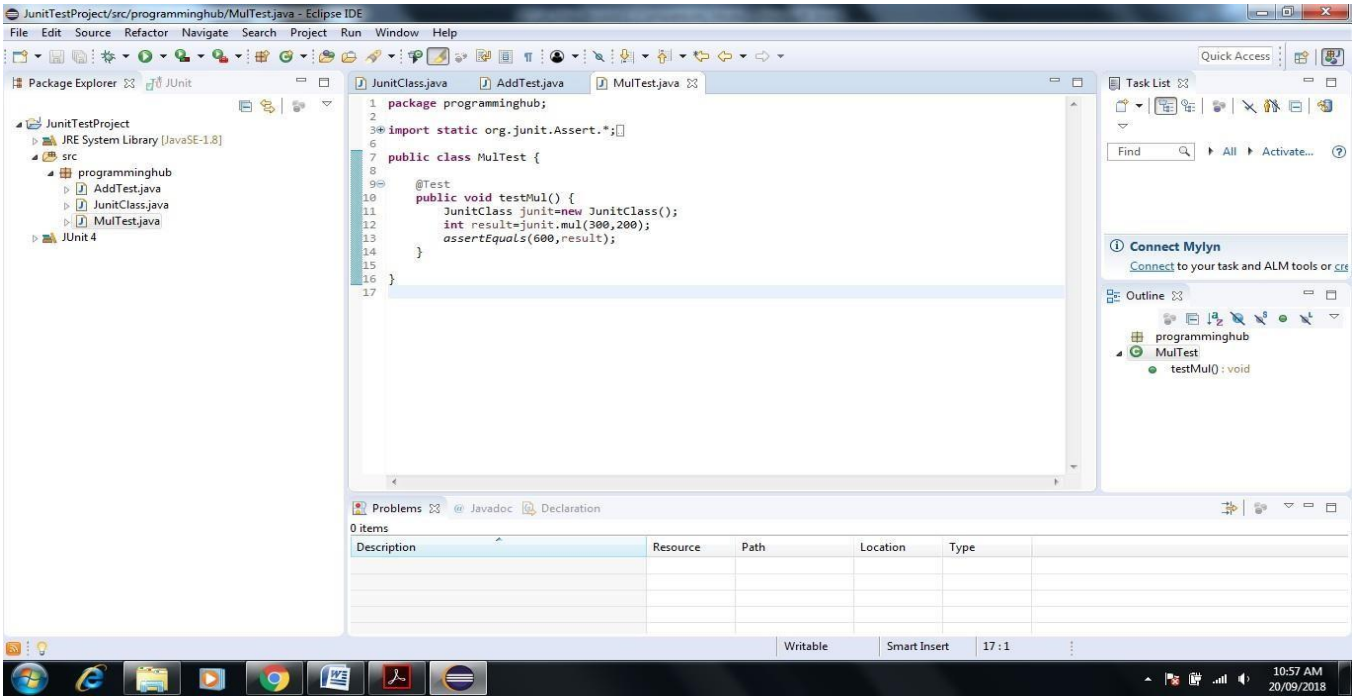
Click on Next ->Select Mul Check Box -> Click Finish



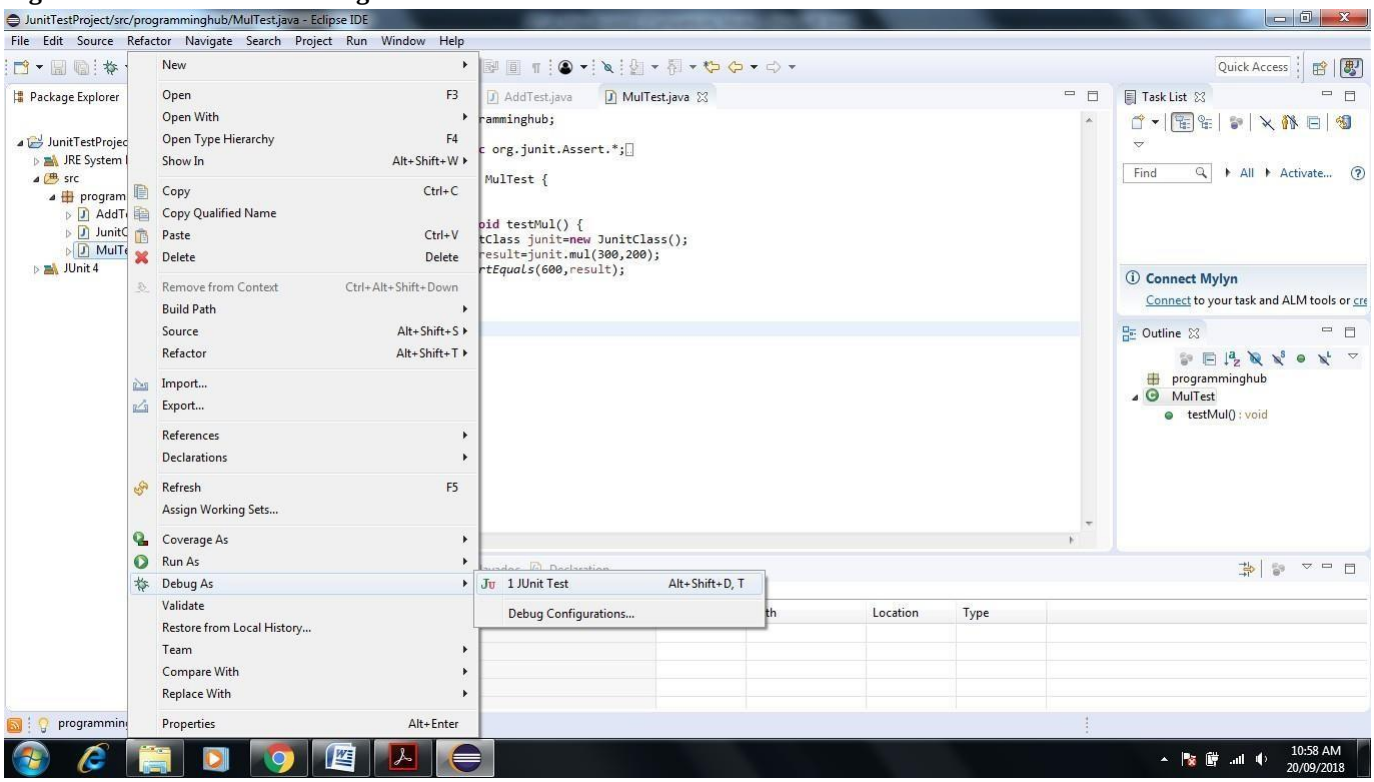
Next Screen will appear



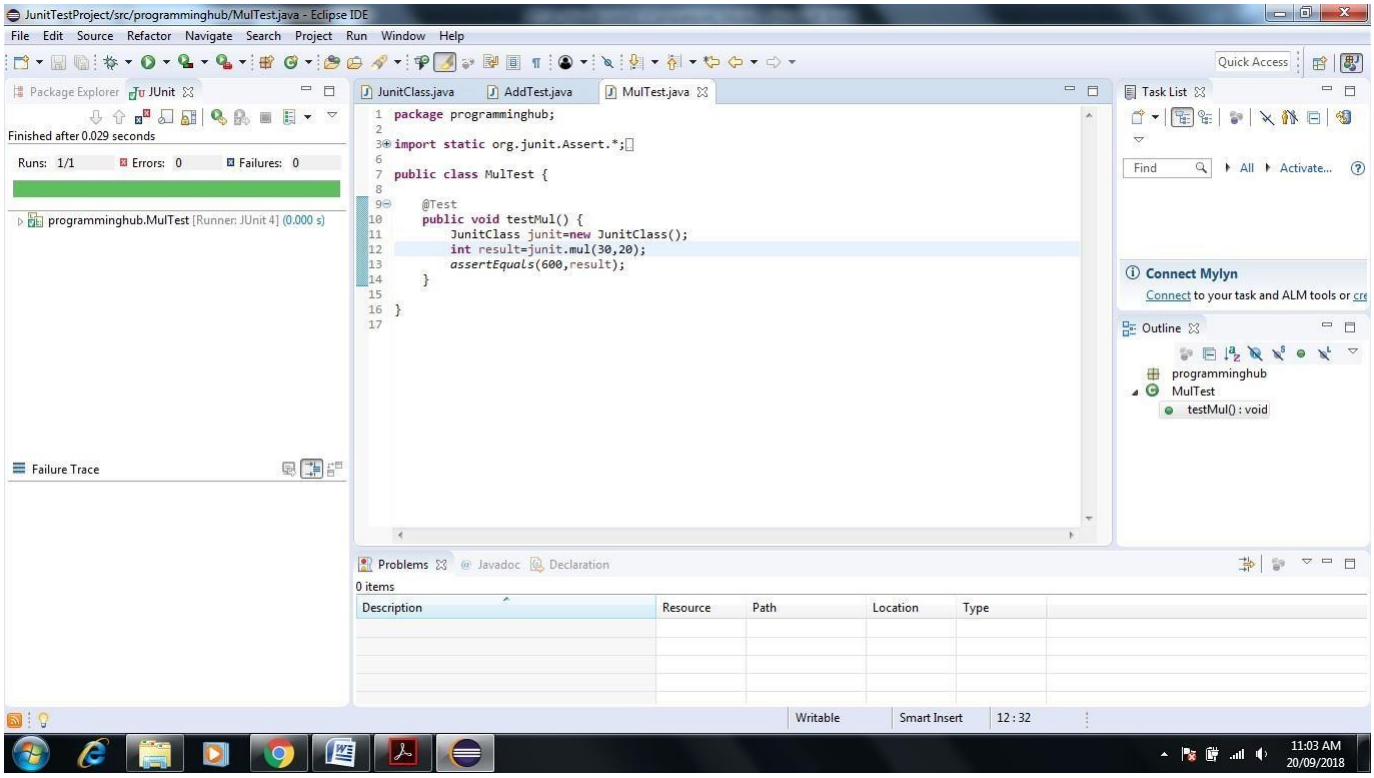
Write a Test Case Code inside MulTest method



Right Click on MulTest->Debug->JUnit Test



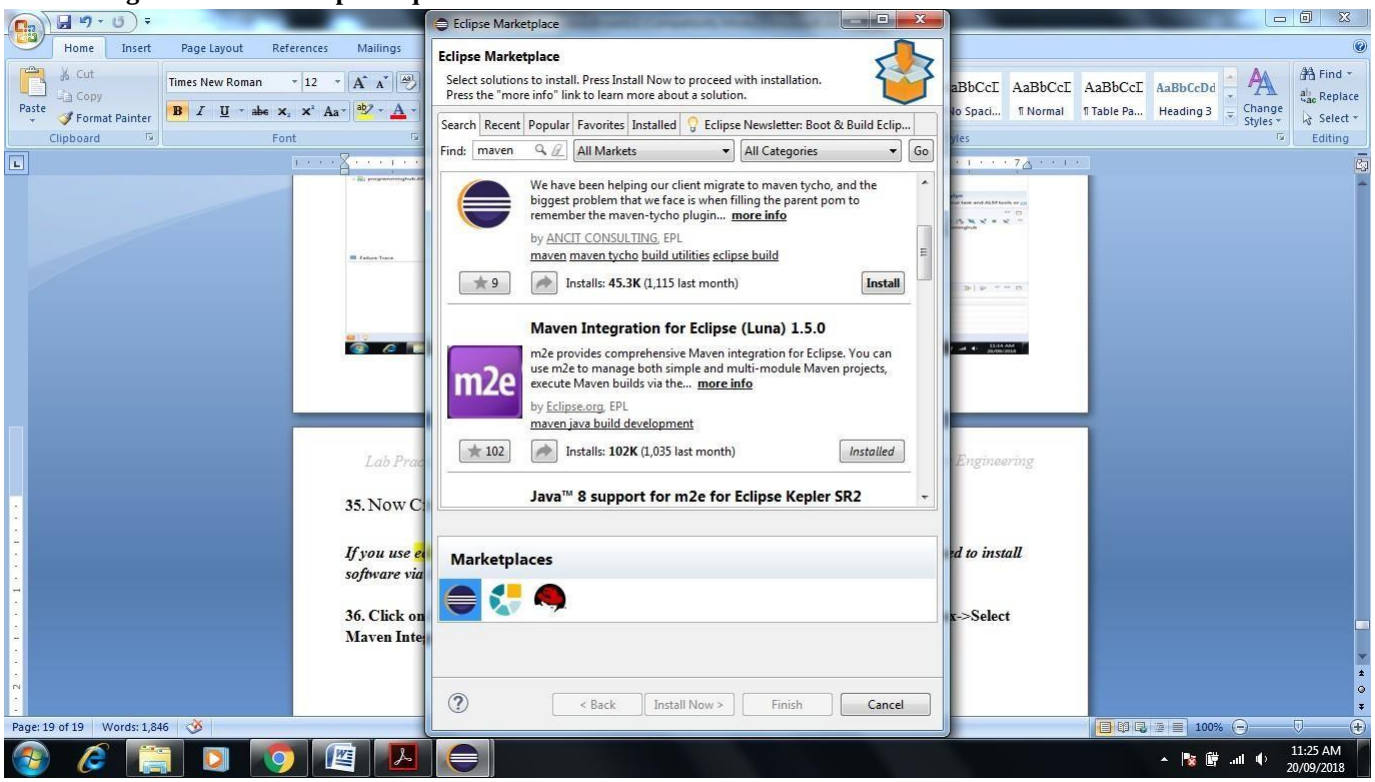
Execute Test



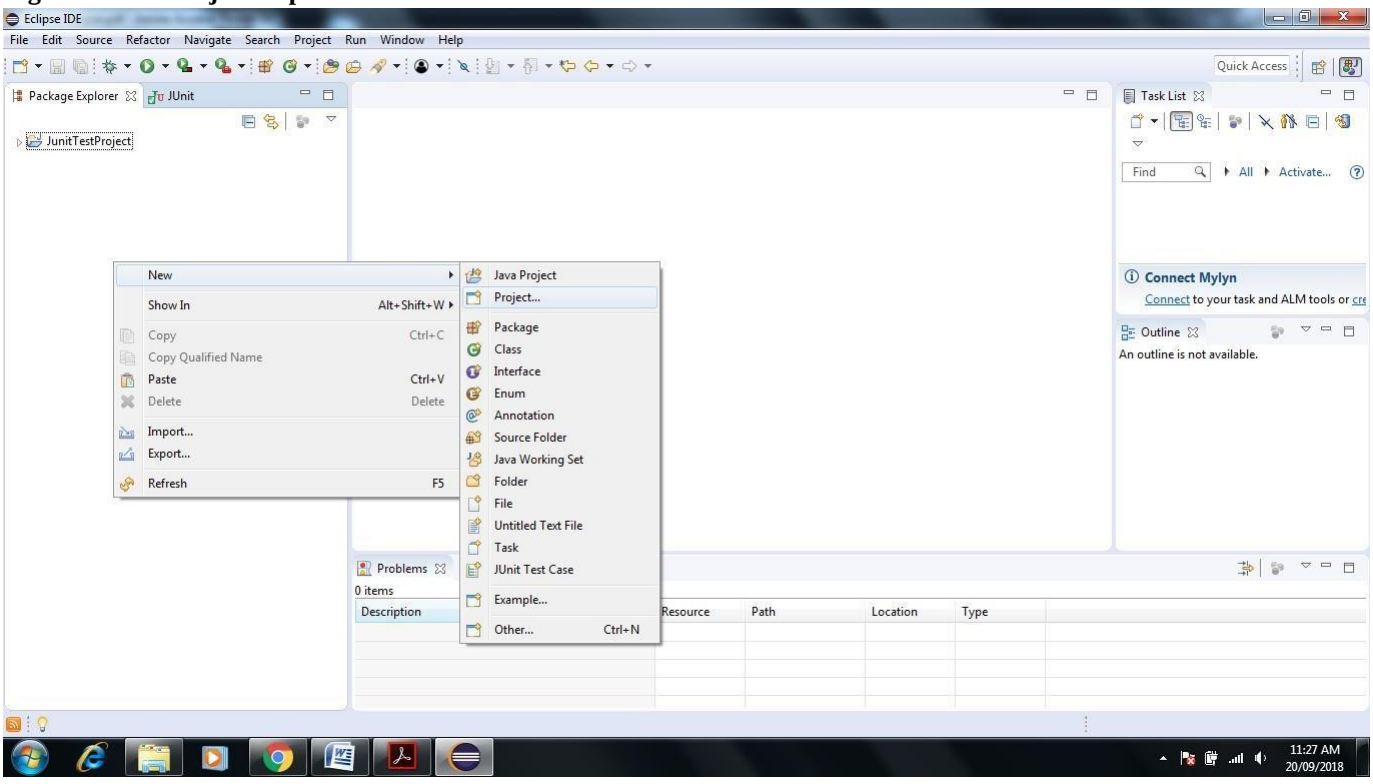
Now Create **Test Report** Using **Apache Maven**

Click on Help in Eclipse->Eclipse Marketplace->Enter Maven Keyword in Search box->Select

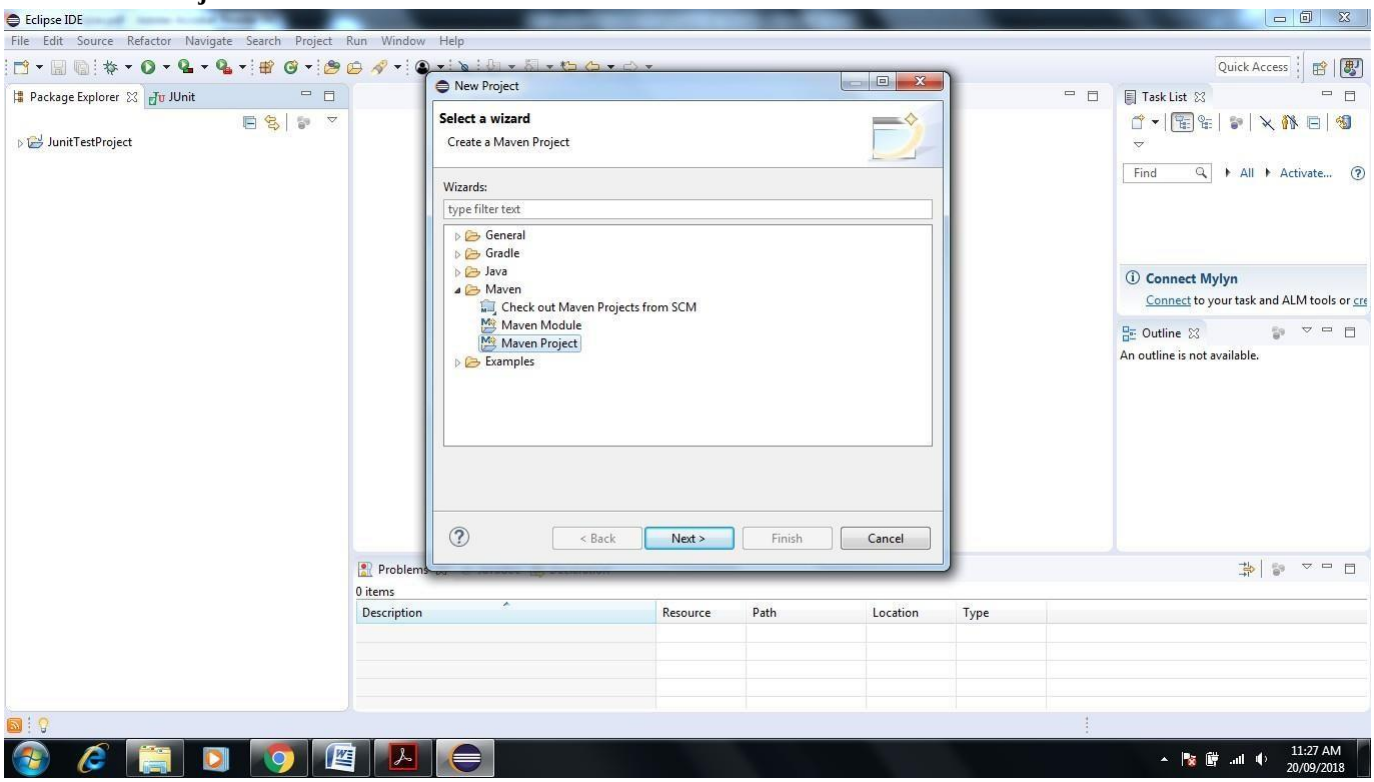
Maven Integration version as per requirement->Click on Install



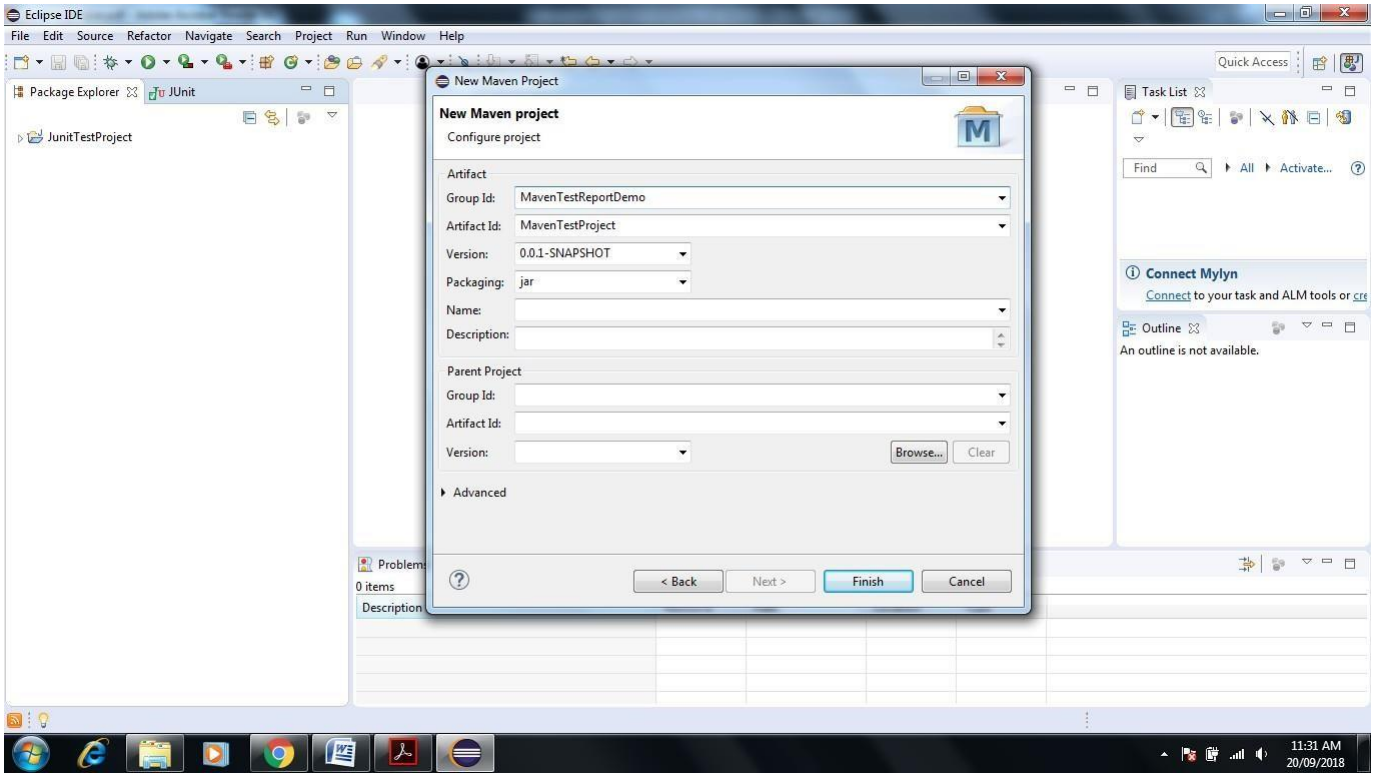
Right Click in Project Explorer Window



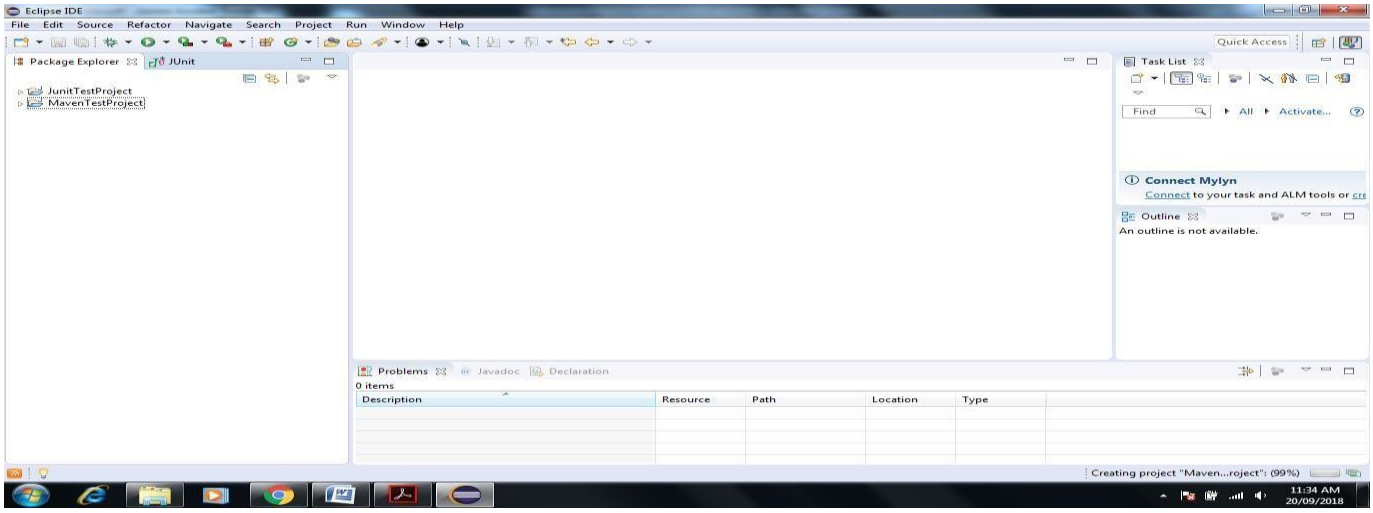
Go to Maven Project-> Click Next



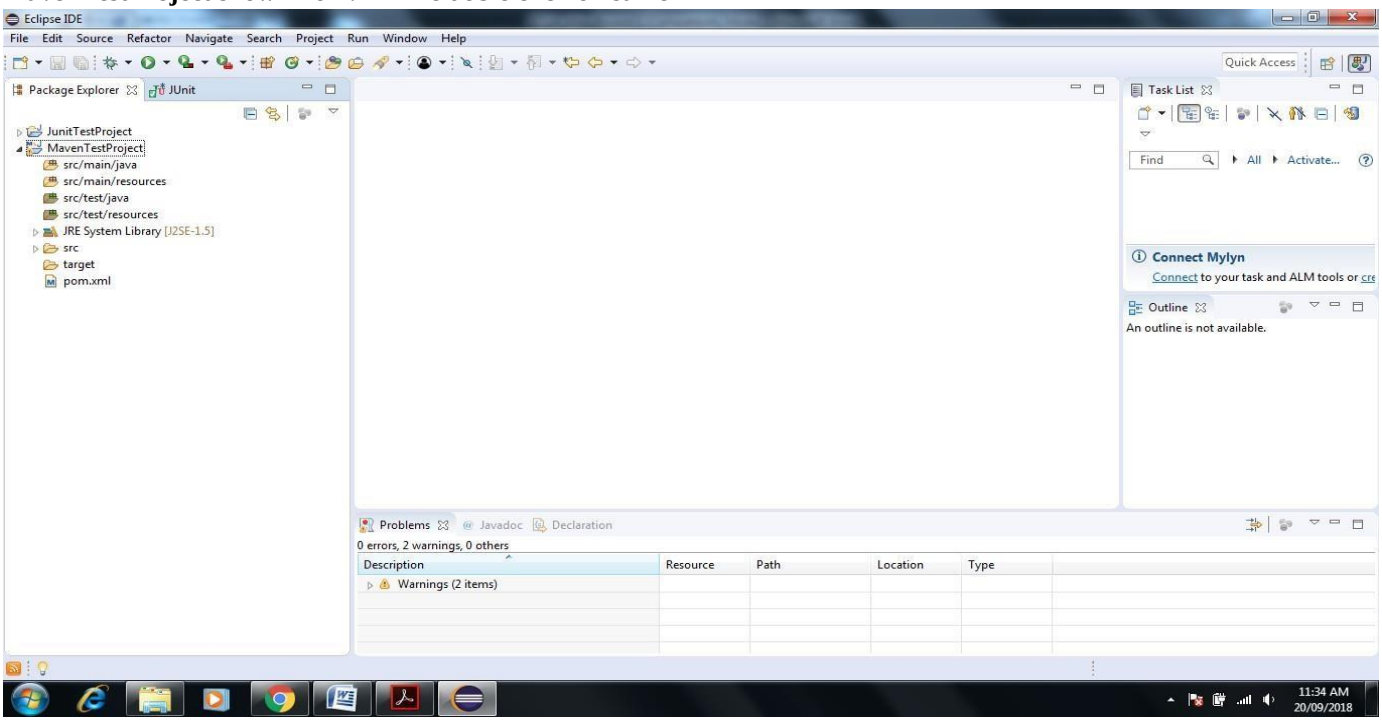
Select Check Box Create Simple Project-> Click Next-> Give Group Id and Artifact name



Click on Finish-> Next Screen Appear

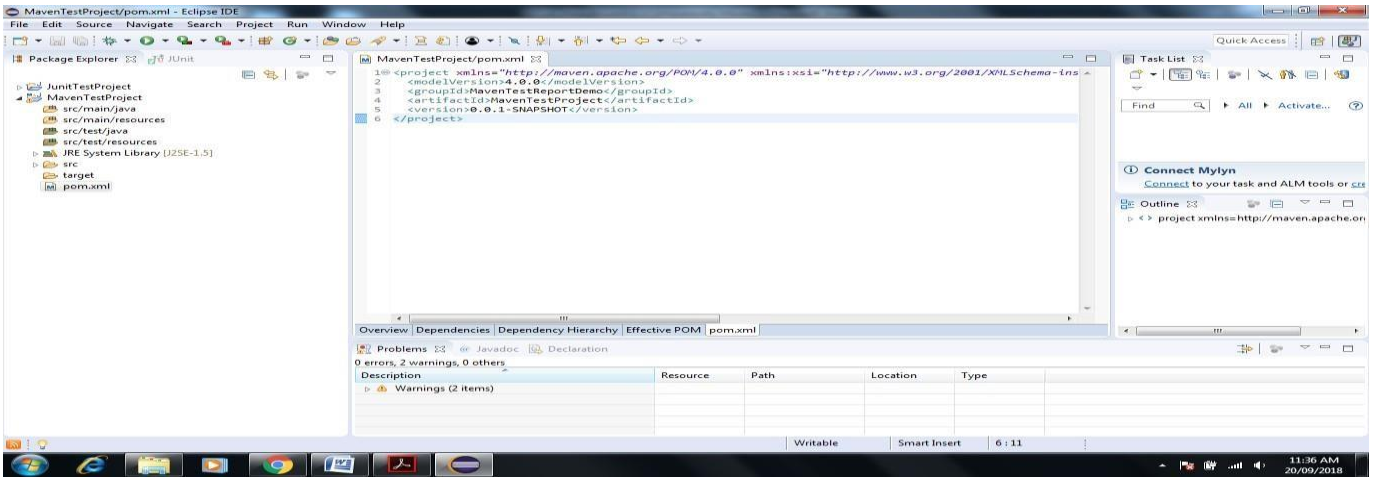


MavenTestProject shown Pom.xml file doble click on same

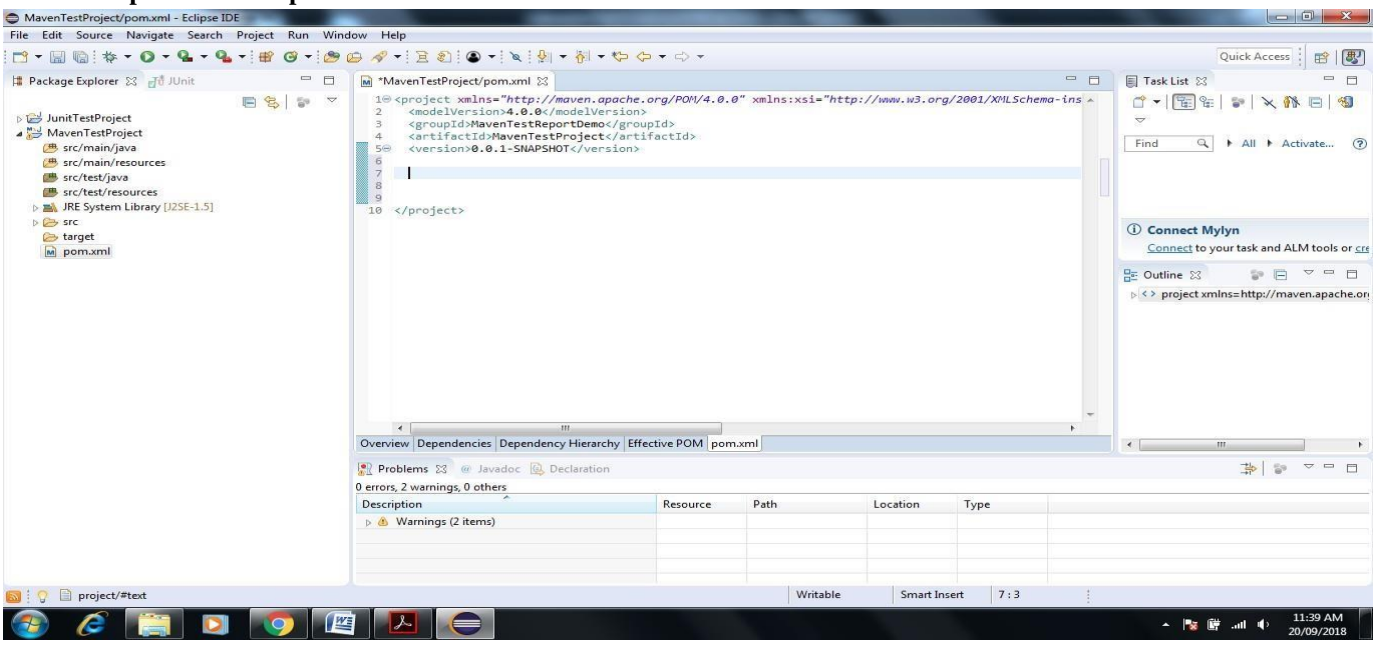


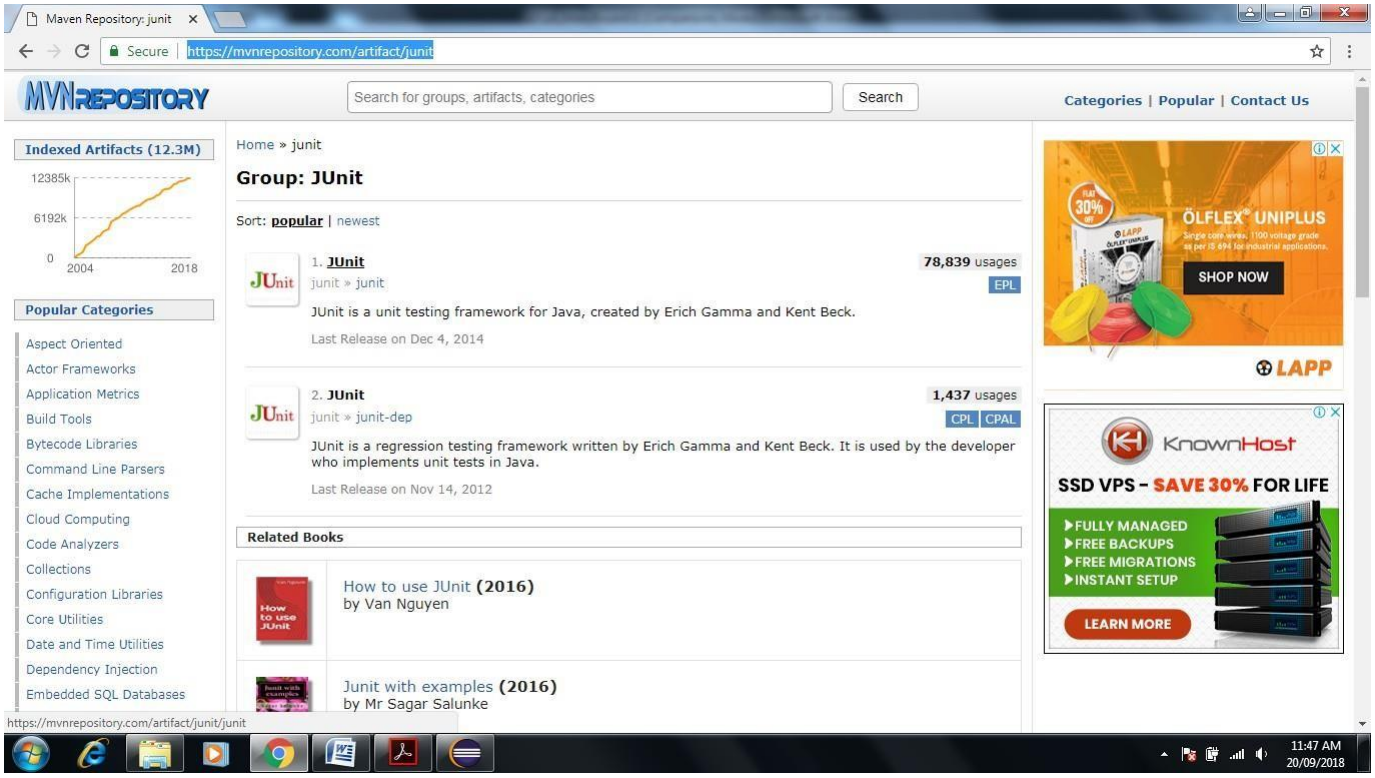
it shown some description like

```
<modelVersion>4.0.0</modelVersion>
<groupId>MavenTestReportDemo</groupId>
<artifactId>MavenTestProject</artifactId>
<version>0.0.1-SNAPSHOT</version>
```

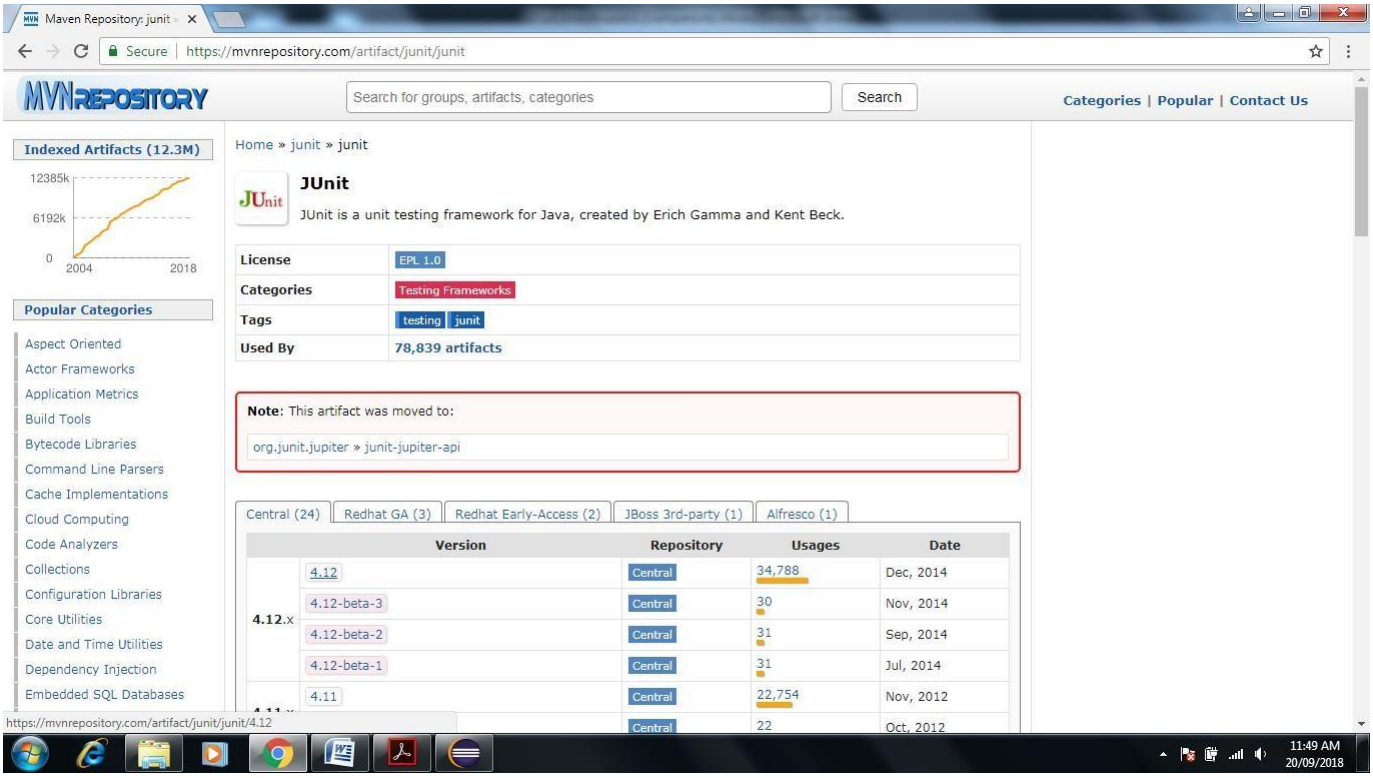


We add dependencies to pom.xml of Junit and Selenium

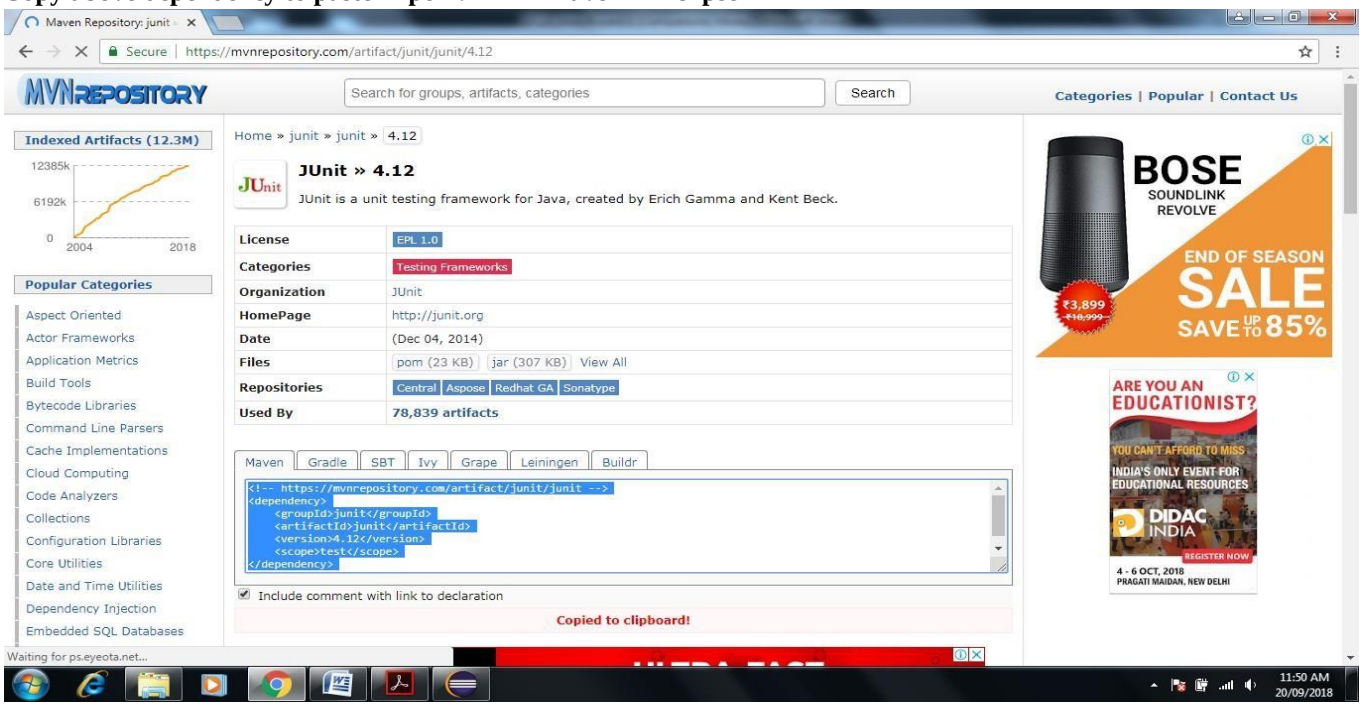




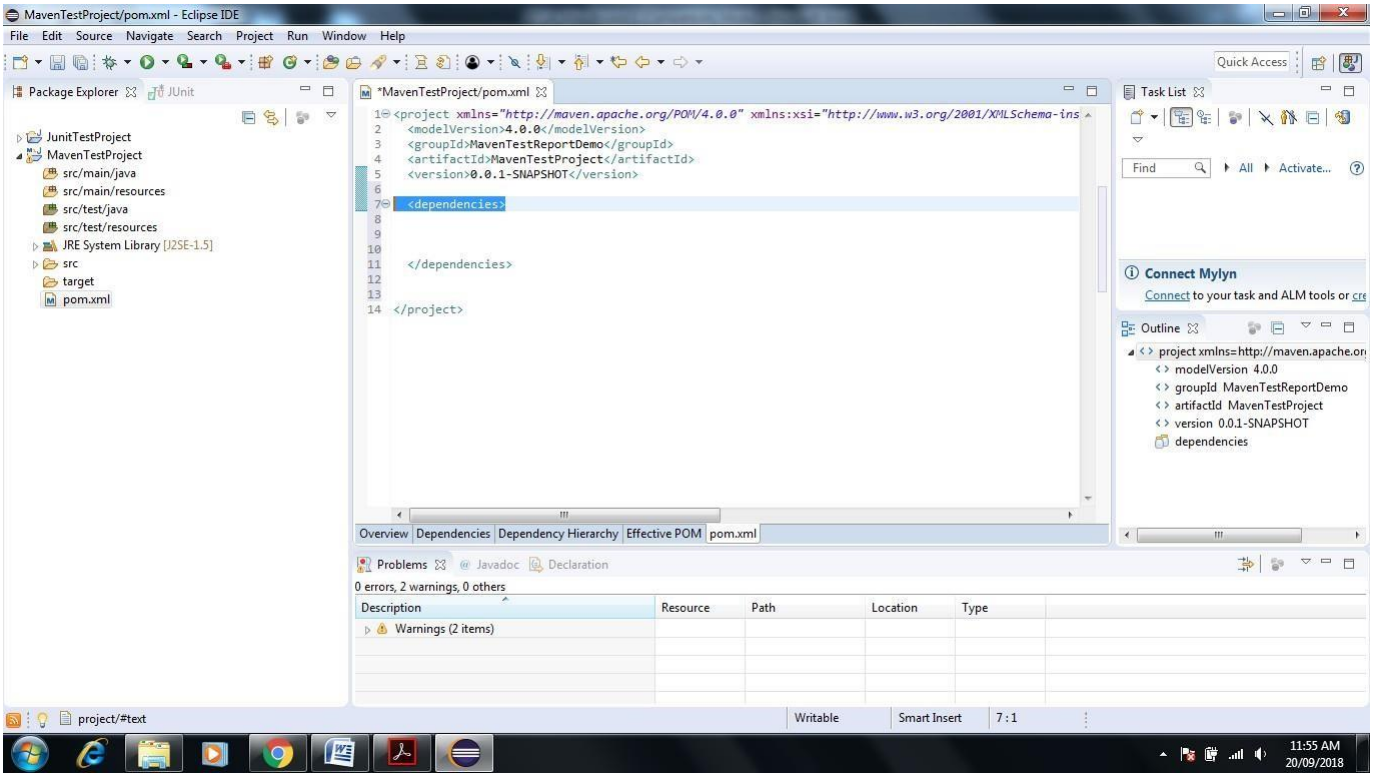
Click on JUnit-> Open and click on latest version as shown below (here 4.12x)



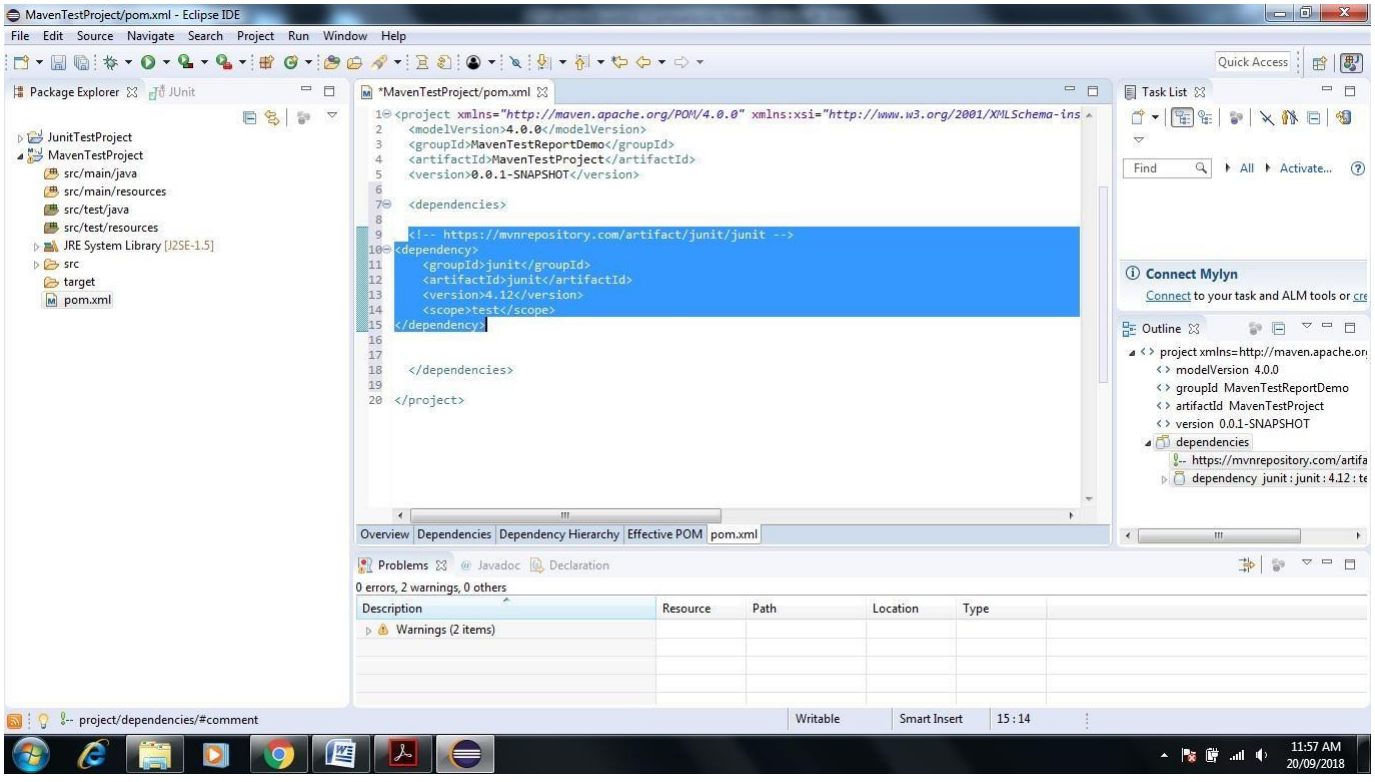
Copy above dependency to paste in pom.xml in Maven in Eclipse

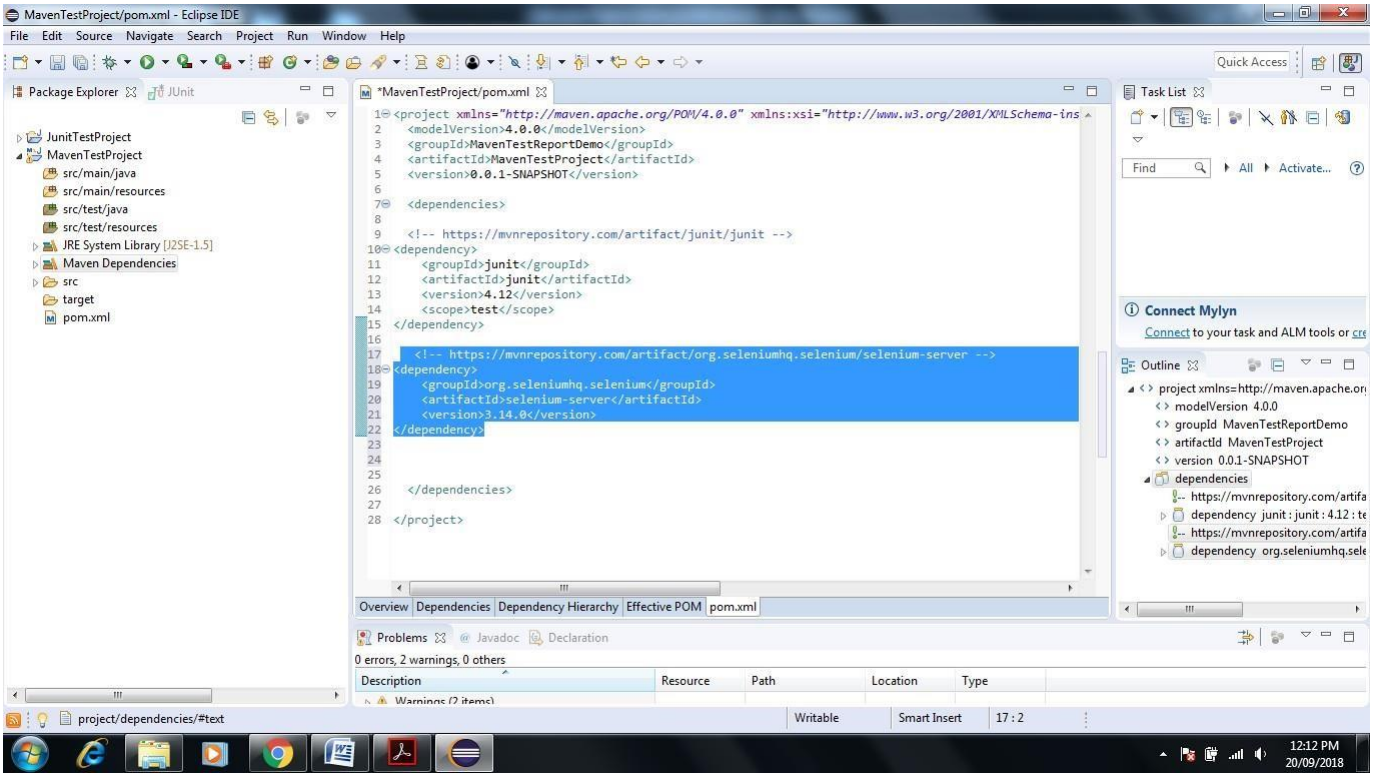


Add <dependencies> tag before pasting as shown below



Now Paste the above code in between <dependencies> tag then save pom.xml file <!--
https://mvnrepository.com/artifact/junit/junit -->
<dependency>
<groupId>junit</groupId>
<artifactId>junit</artifactId>
<version>4.12</version>
<scope>test</scope> </dependency>

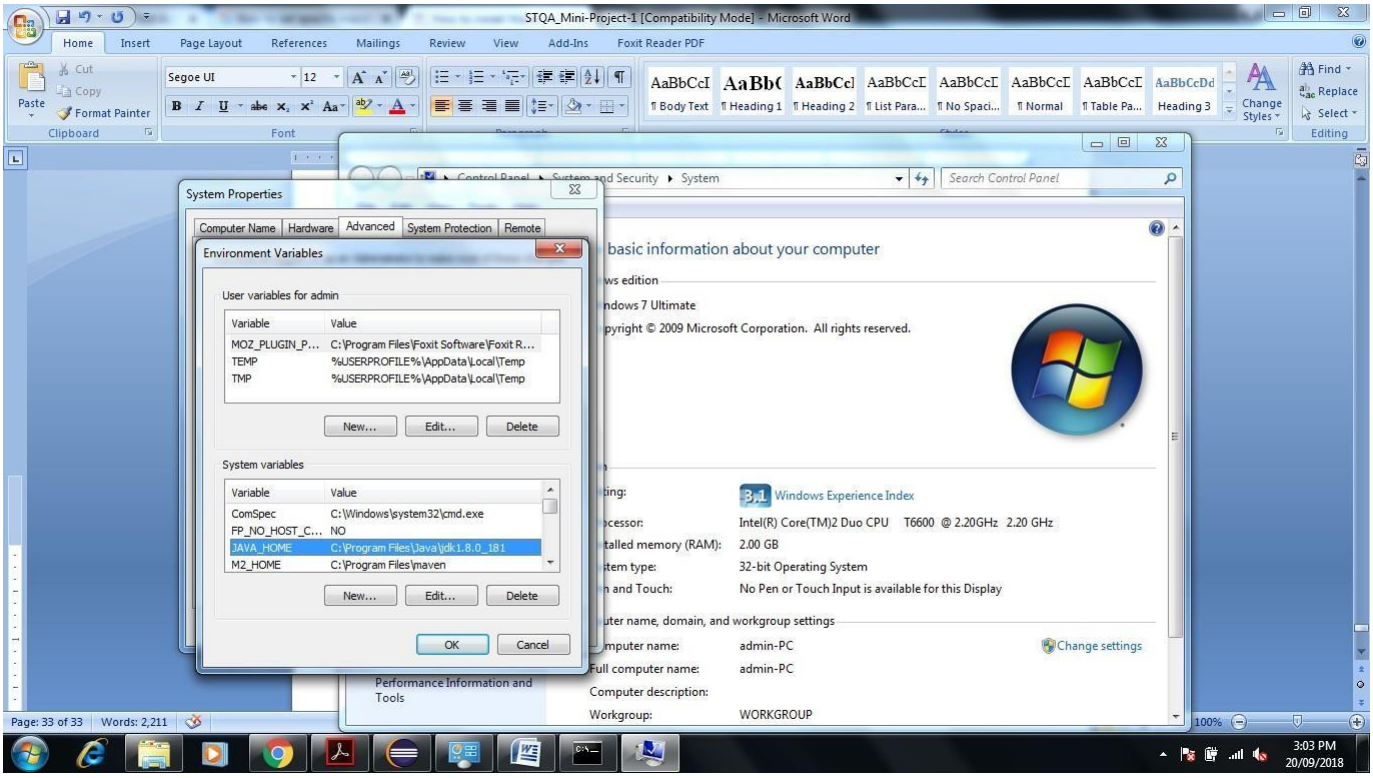




Environment Setup Very Important Steps to Generate Report

1. JDK and JAVA_HOME

Make sure JDK is installed, and "**JAVA_HOME**" variable is added as Windows environment variable. Our JDK installed in Program File ->JAVA->JDK 1.8.0



2. Set Path of Add **M2_HOME** and **MAVEN_HOME**

Create new system variable name M2_HOME and **MAVEN_HOME separately set variable value➐ C:\Program Files\maven**

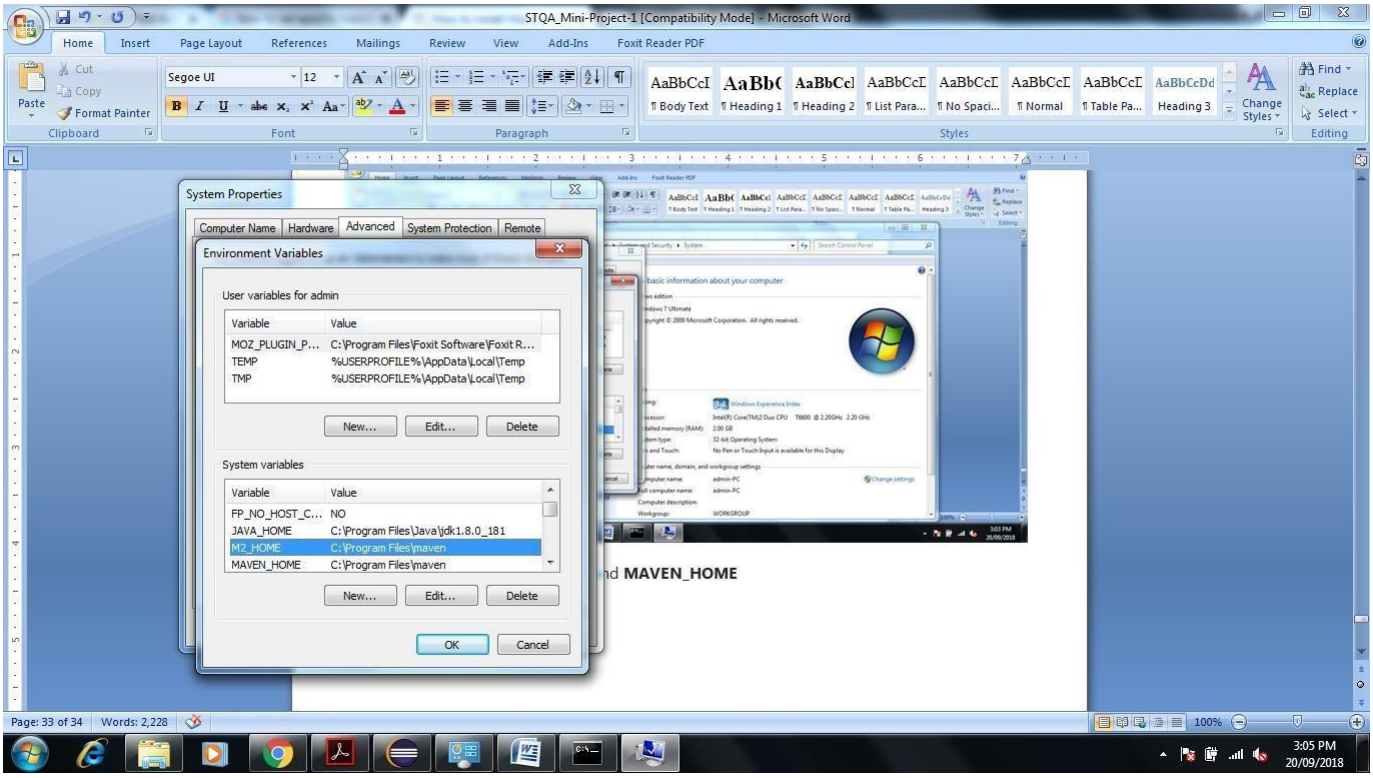


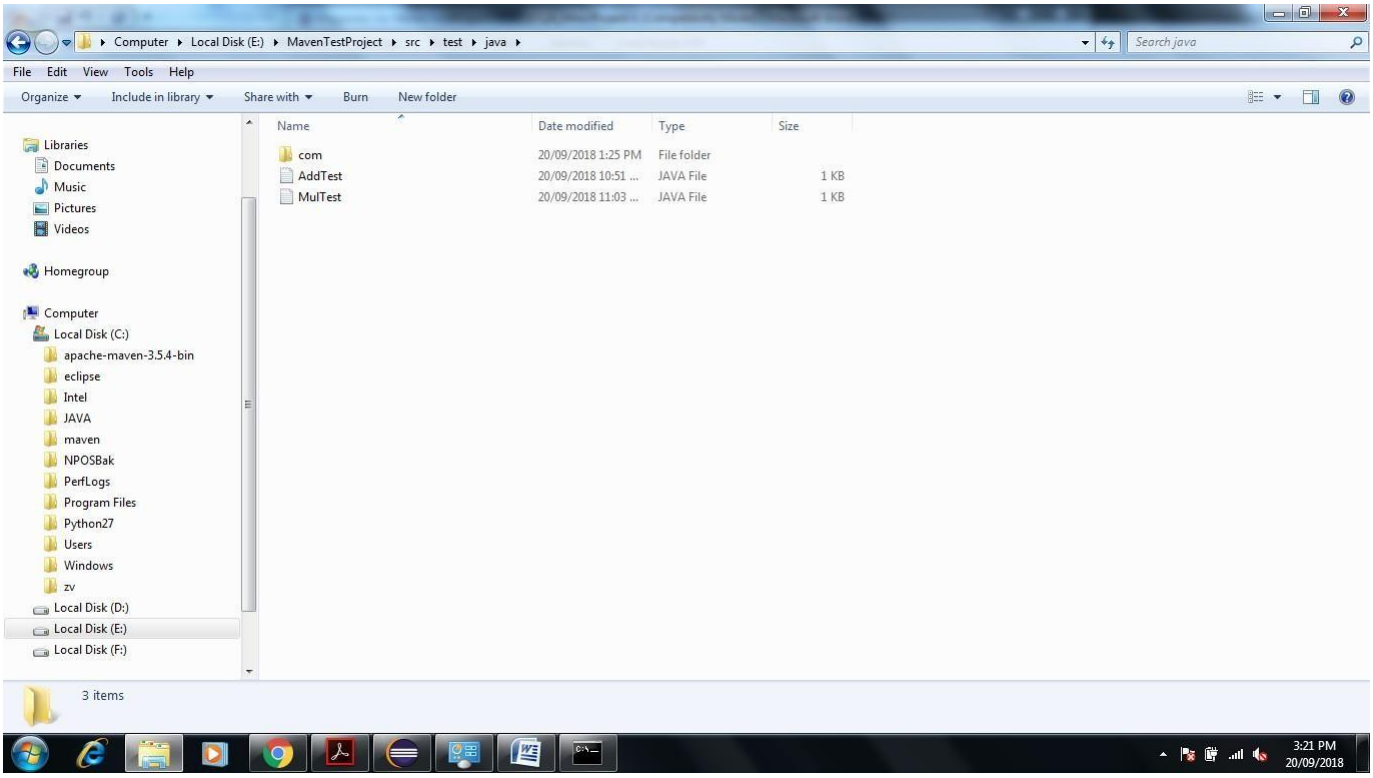
Figure Shows the Path of M2_HOME & MAVEN_HOME same.

3. Update PATH Variable as per folowing

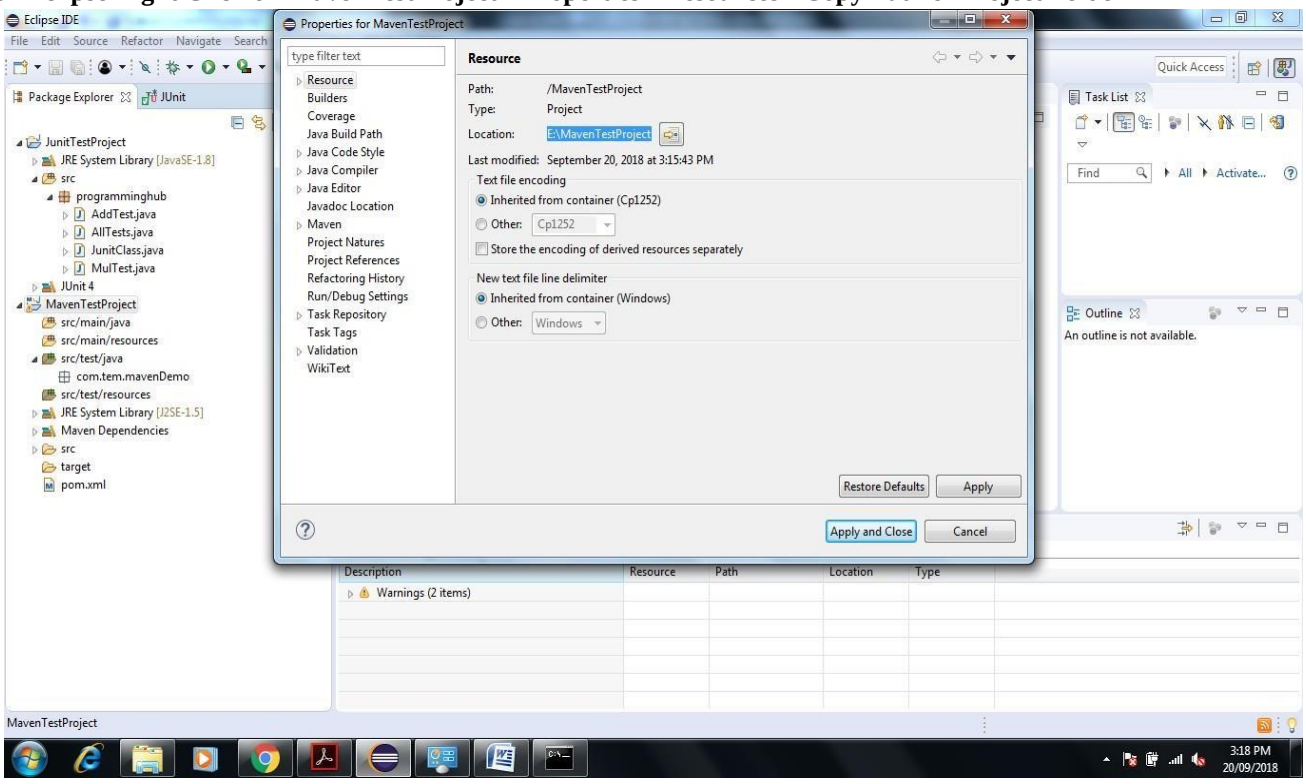
C:\Program Files\maven%MAVEN_HOME%\bin;%M2_HOME%\bin;

4. Verification

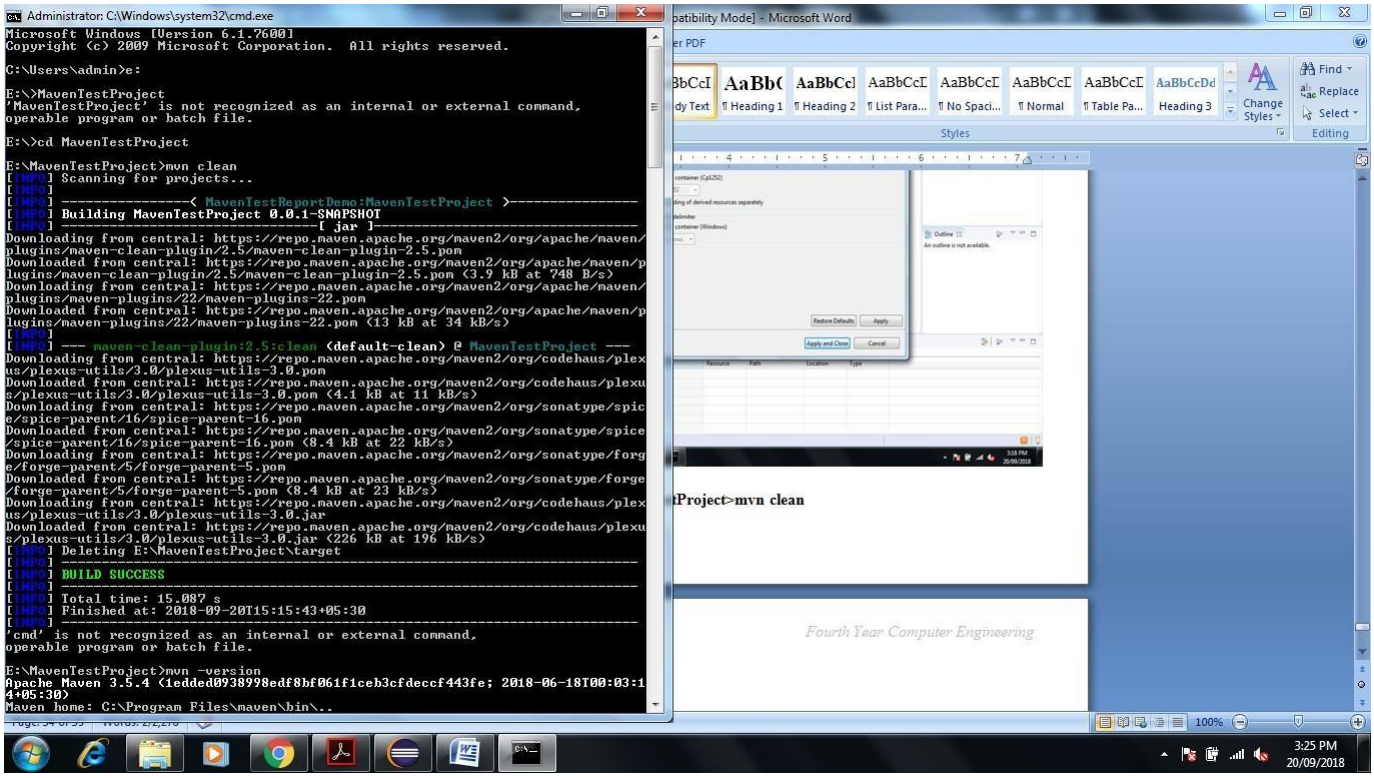
Now copy Previous Created JUnit Test case java file Add Test and Mul Test Paste Externally in E:\MavenTestProject\src\test\java



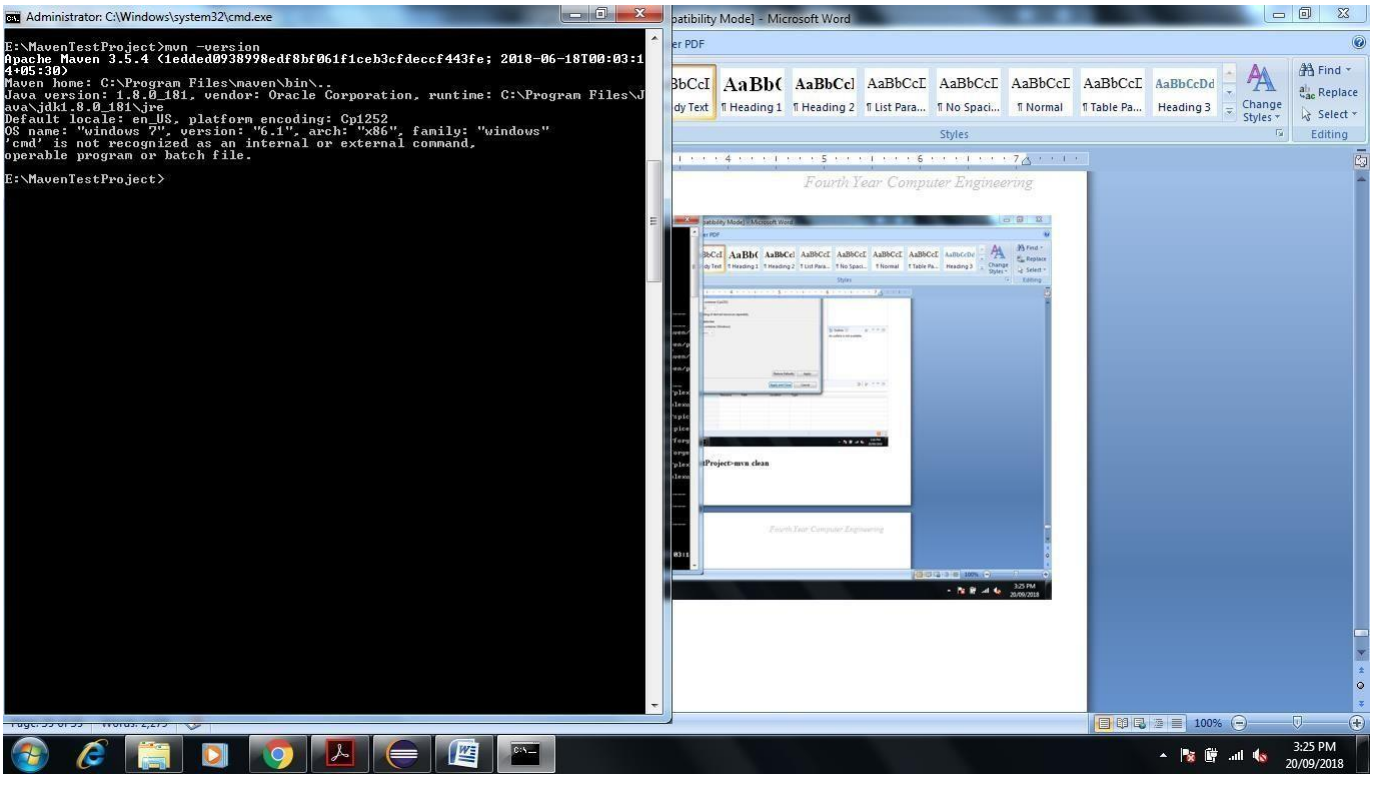
Now Open Eclipse Right Click on MavenTestProject->Properties➐Resources➐Copy Path of Project Folder



Now go to command prompt➐E:\MavenTestProject>mvn clean



Enter E:\MavenTestProject>mvn –version

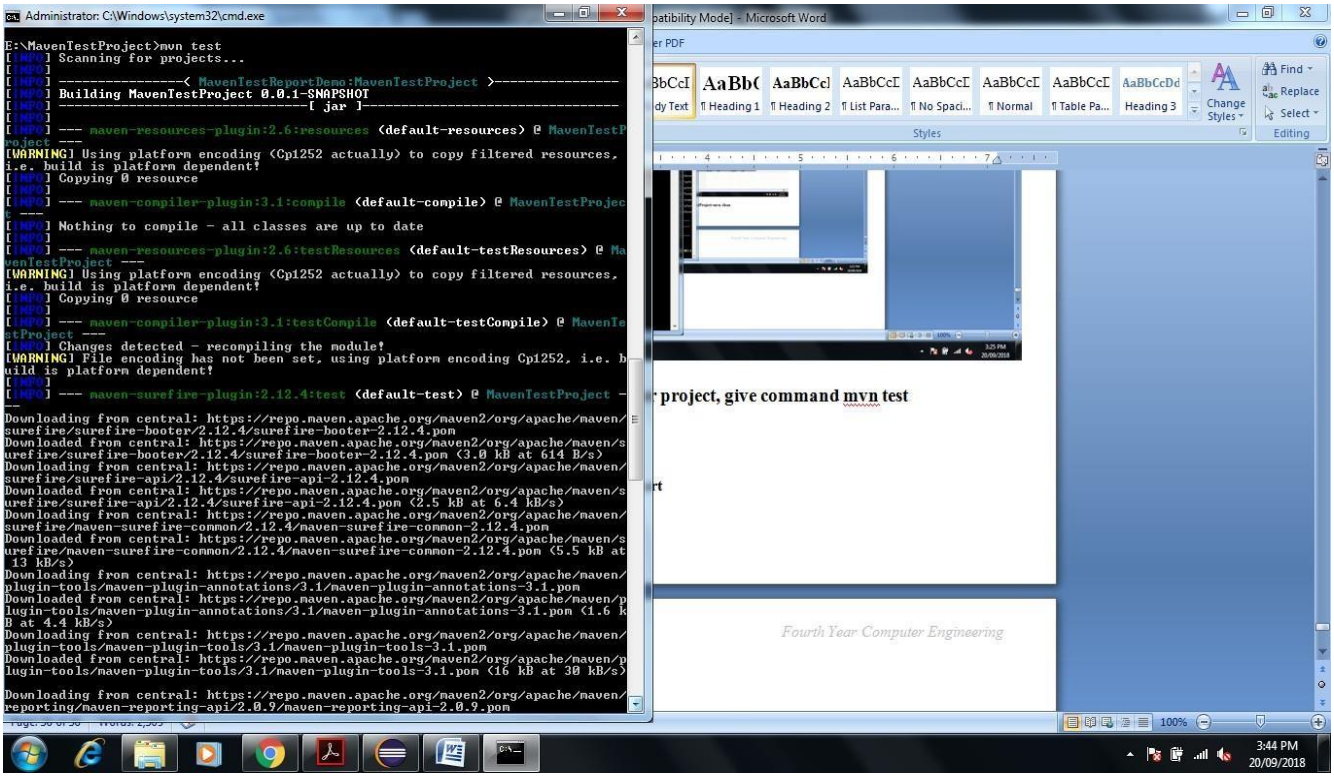


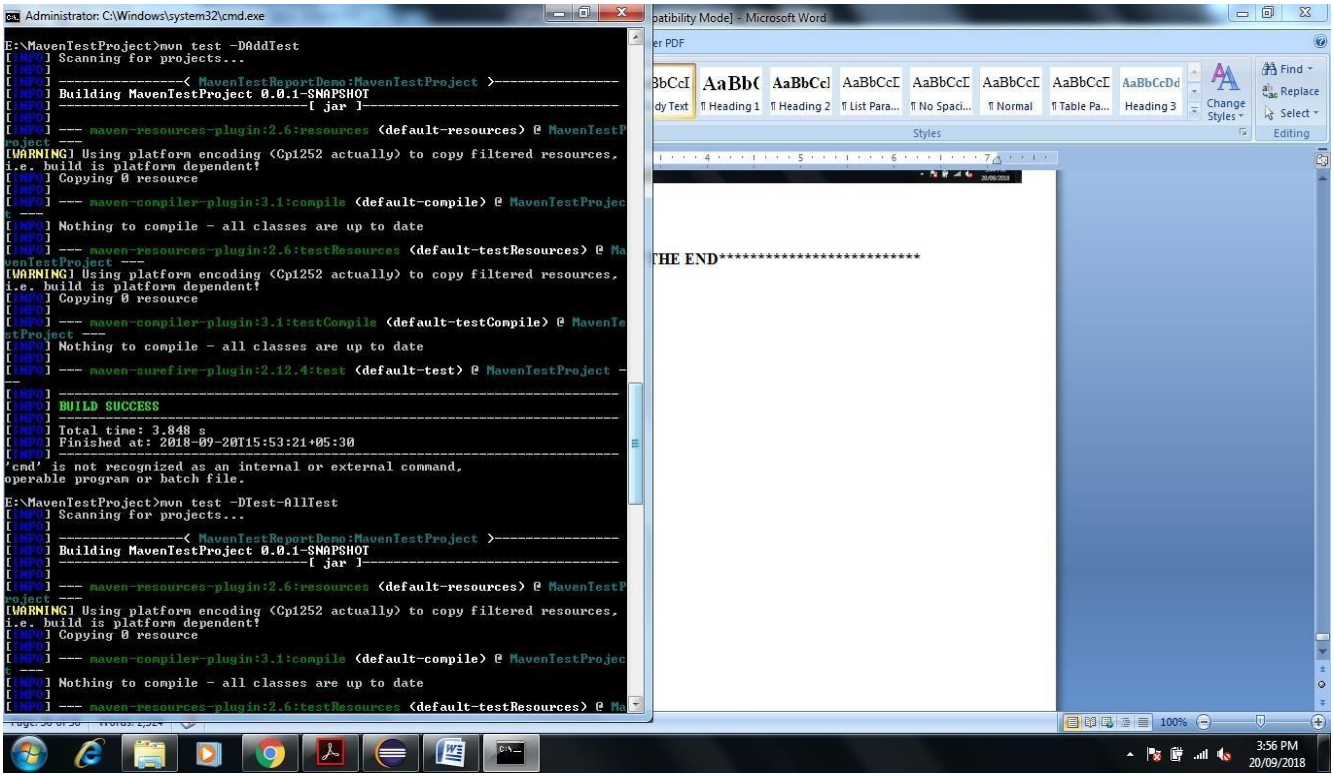
To run test suite or all test cases under project, give command mvn test

Enter E:\MavenTestProject>mvn test

This Command is Used to See the Test Report

You can even run individual test cases. Give command mvn test –Dtesttestcasename
Eg. mvn test –Dtest-AllTest



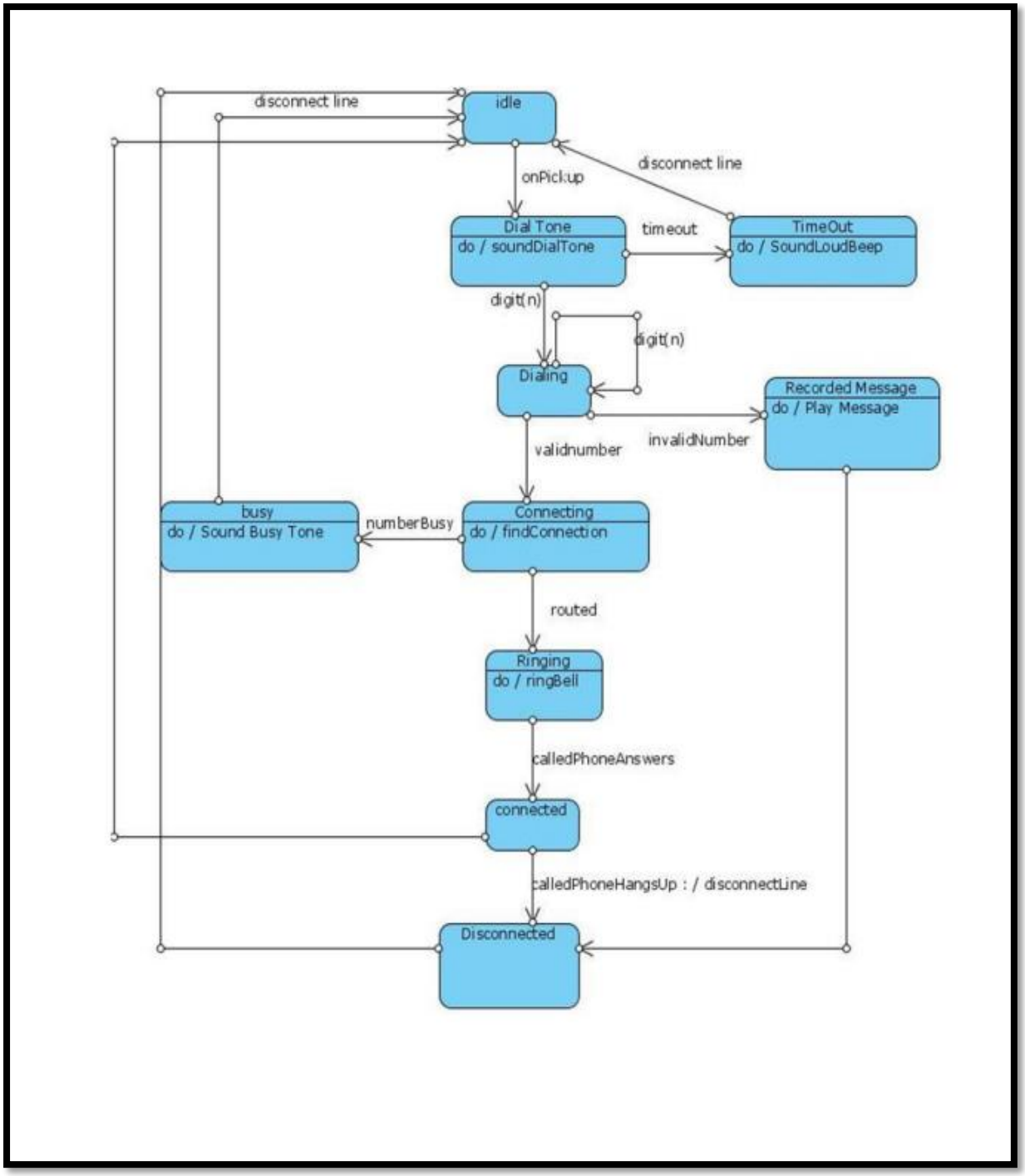


Conclusion

In this way using JUnit and Maven Automation tool we are Perform Unit Testing and Prepare Test Report of same.

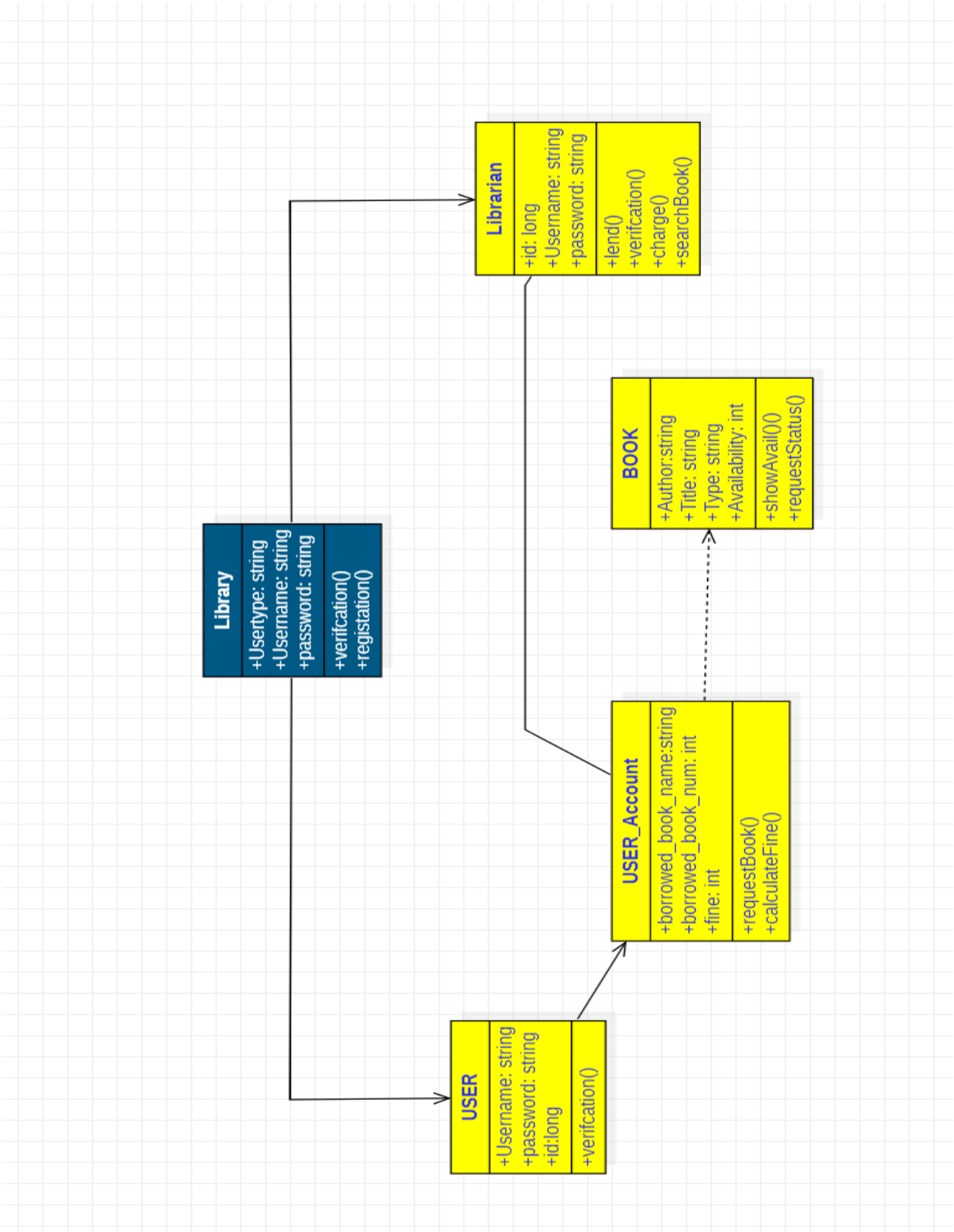
OUTPUT:

State model/diagram for telephone line



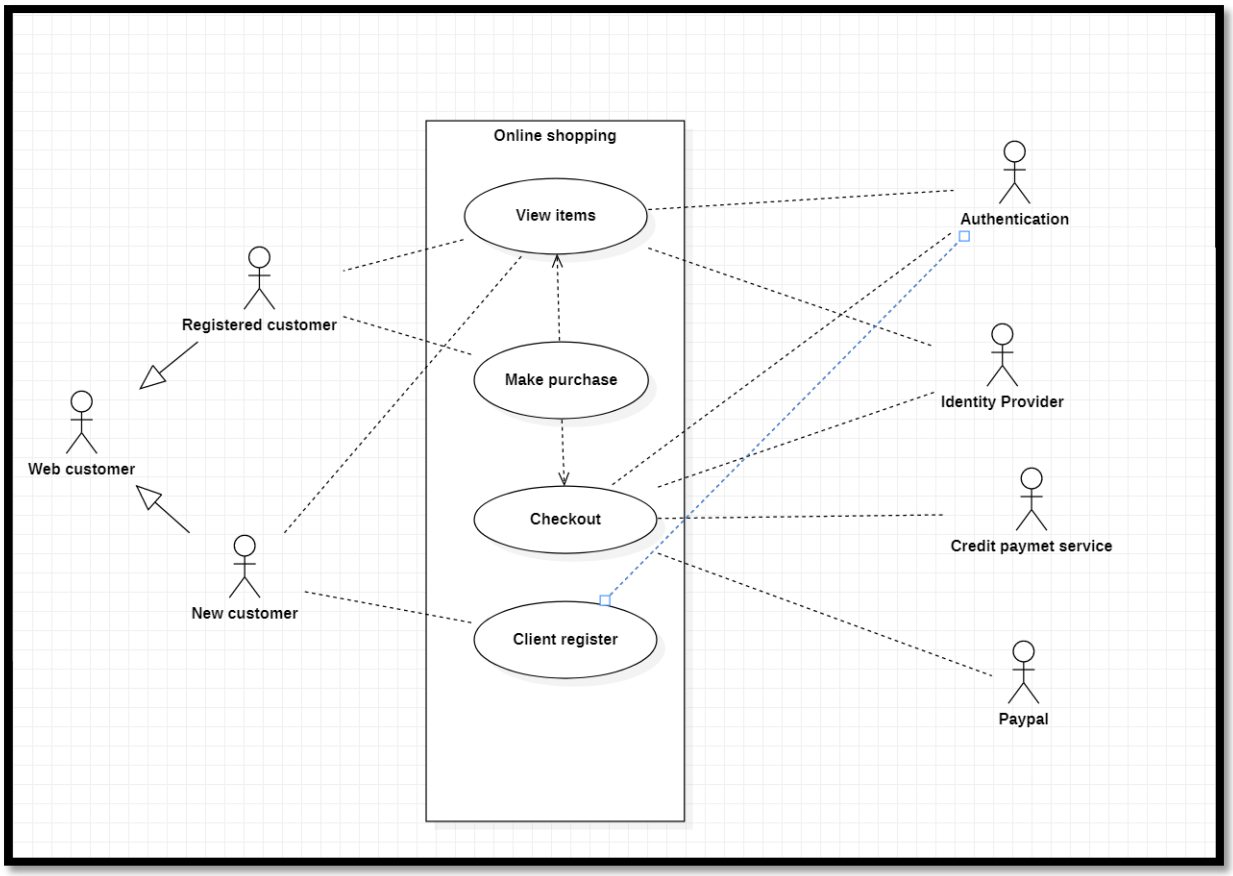
OUTPUT:

Class diagram for library management system



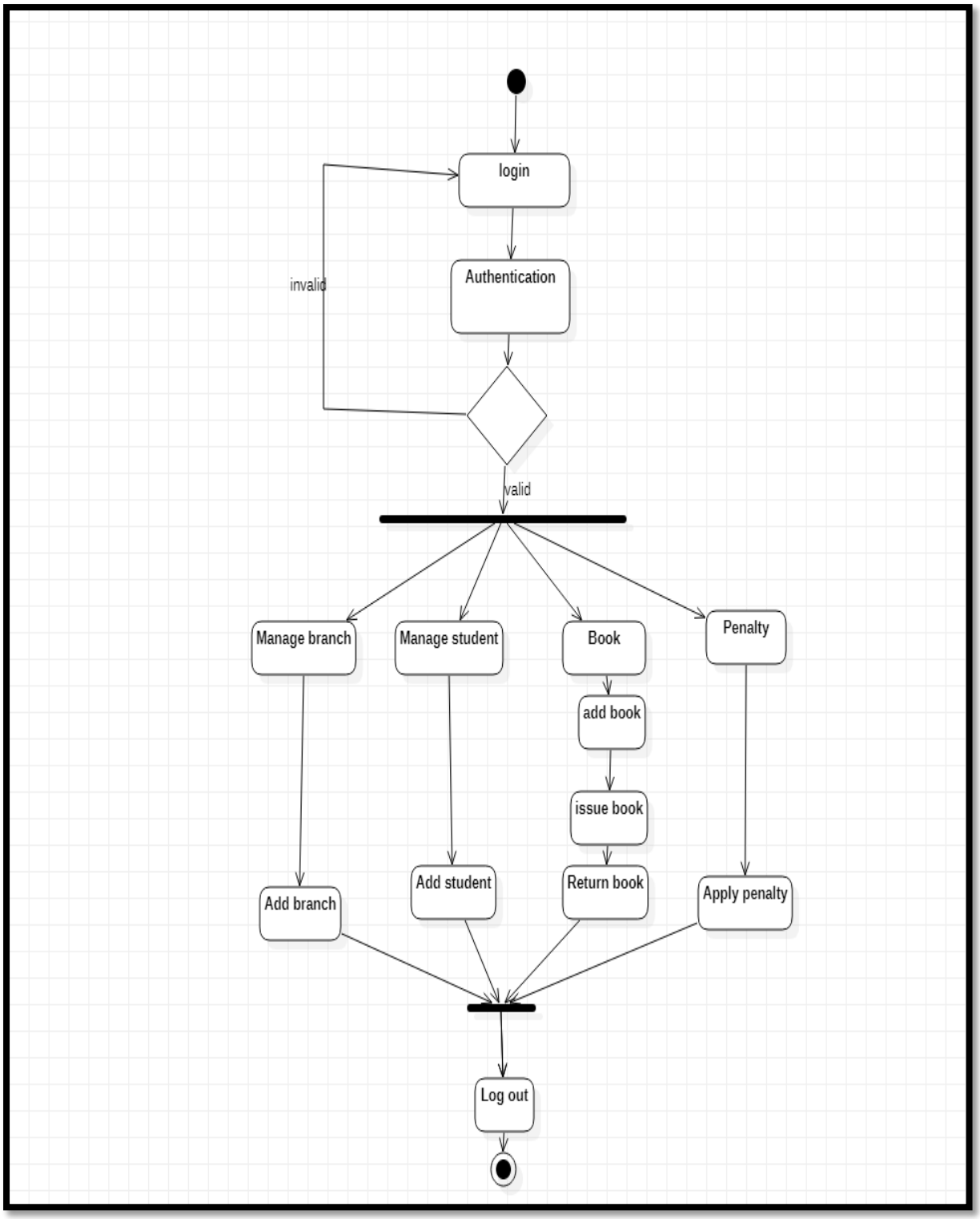
OUTPUT:

Use case diagram for Online shopping



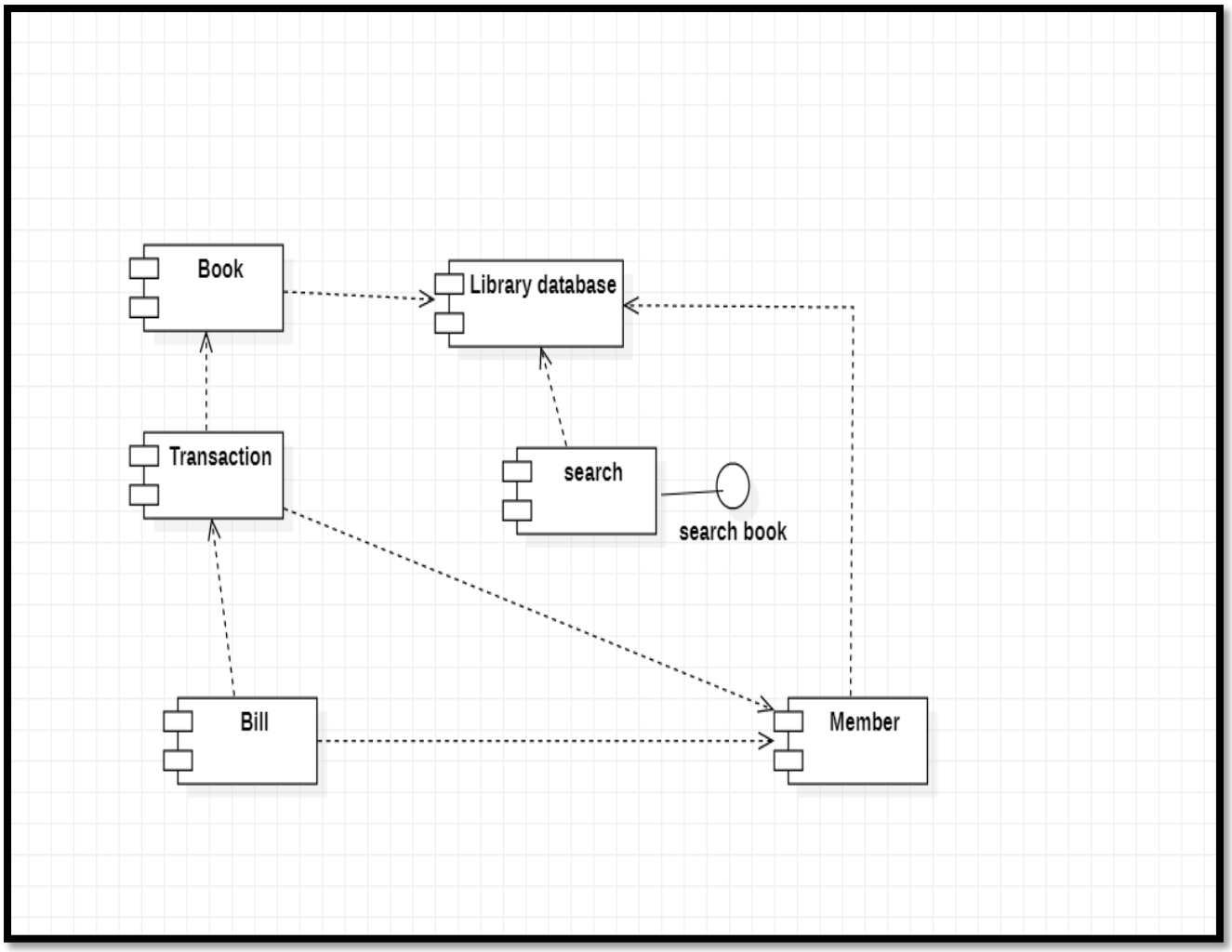
OUTPUT:

Activity diagram for library management



OUTPUT:

Component diagram for library management



MINI PROJECT [OOMD]

TITLE: UML Diagrams For Hotel Management System

Activity UML diagram

Activity UML diagram of Hotel Management System which shows the flows between the activity of Hotel, Payments Rooms, Booking Customers. The main activity involved in this UML Activity Diagram of Hotel Management System are as follows:

Hotel Activity

Payments Activity

Rooms Activity

Booking Activity

Customers Activity

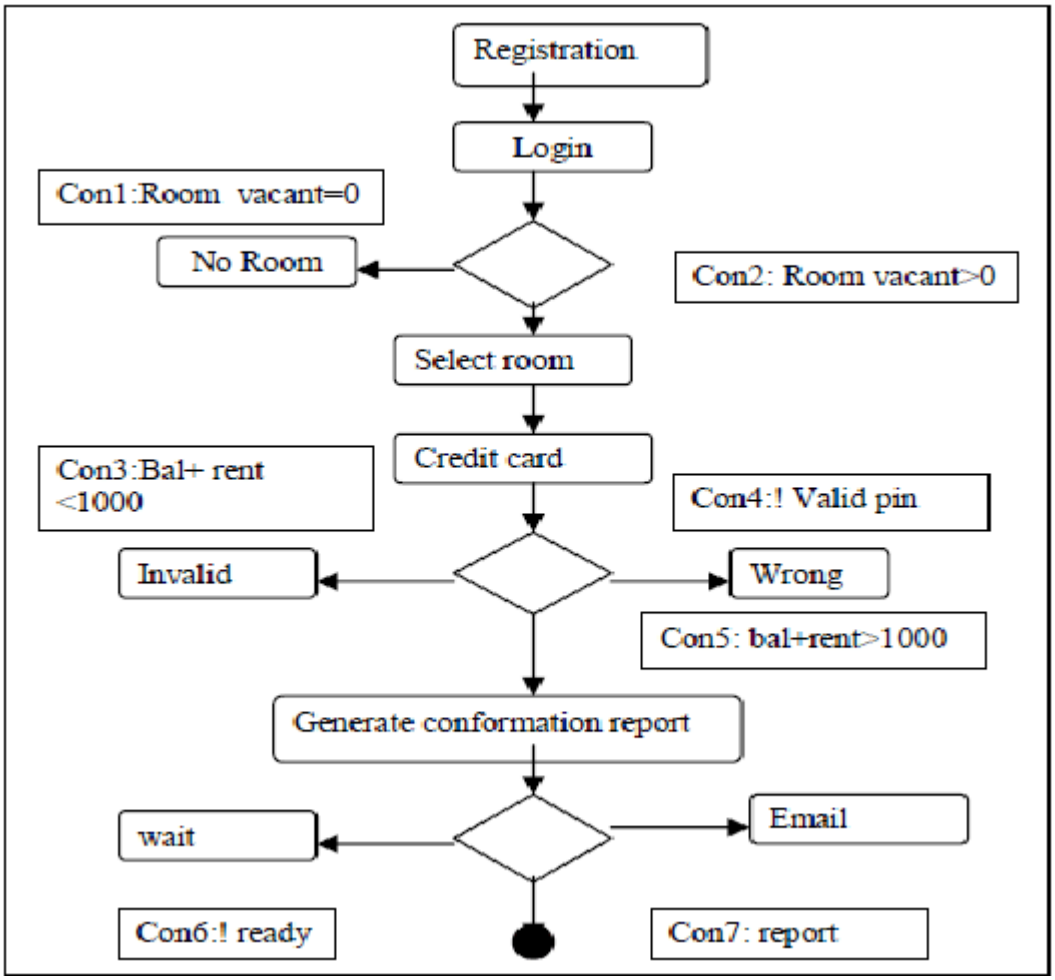


Fig. Activity Diagram

Features Of The Activity UML Diagram Of Hotel Management System

Admin User can search Hotel, view description of a selected Hotel, add Hotel, update Hotel and delete Hotel

Its shows the activity flow of editing, adding and updating of Payments
User will be able to search and generate report of Rooms, Booking Customers

All objects such as (Hotel, Payments, Customers) are interlinked its shows the full description and flow of Hotel, Booking, Customers, Rooms, Payments

Class Diagram

Hotel Management System Class Diagram describes the structure of a Hotel Management System casses their artes operation for metod and the relationships among objects. The main classes of the Hotel Management System are Hotel Room, Services Payments, Booking C

Classes of Hotel Management System Class Diagram:
Hotel Class: Manage all the operations of Hotel
Rooms Class: Manage all the operations of Rooms. Services Class Manage all the operations of Services

Payments Class Manage all the operations of Payments
Booking Class Manage all the operations of Booking Customers Class Manage all the operations of Customers
Hotel Methods: addHotel(), editHotel(), deleteHotel(), updateHotel(), saveHotel), searchHote , addRooms(), editRooms(), deleteRooms(), updateRooms(), saveRoomall(), searchRooms()
Services Methods: addServices(), editServices(), deleteServices(), updateServices(), saveServices), searchServices()
Payments Methods: addPayments(), editPayments(), deletePayments(), updatePayments(), savePayments()
Booking Methods : addBooking(), editBooking(), delete Booking(), updateBooking(), savelocking() searchBooking()
Customers Methods: addCustomers(), editCustomers(), deleteCustomers(), updateCustomers()

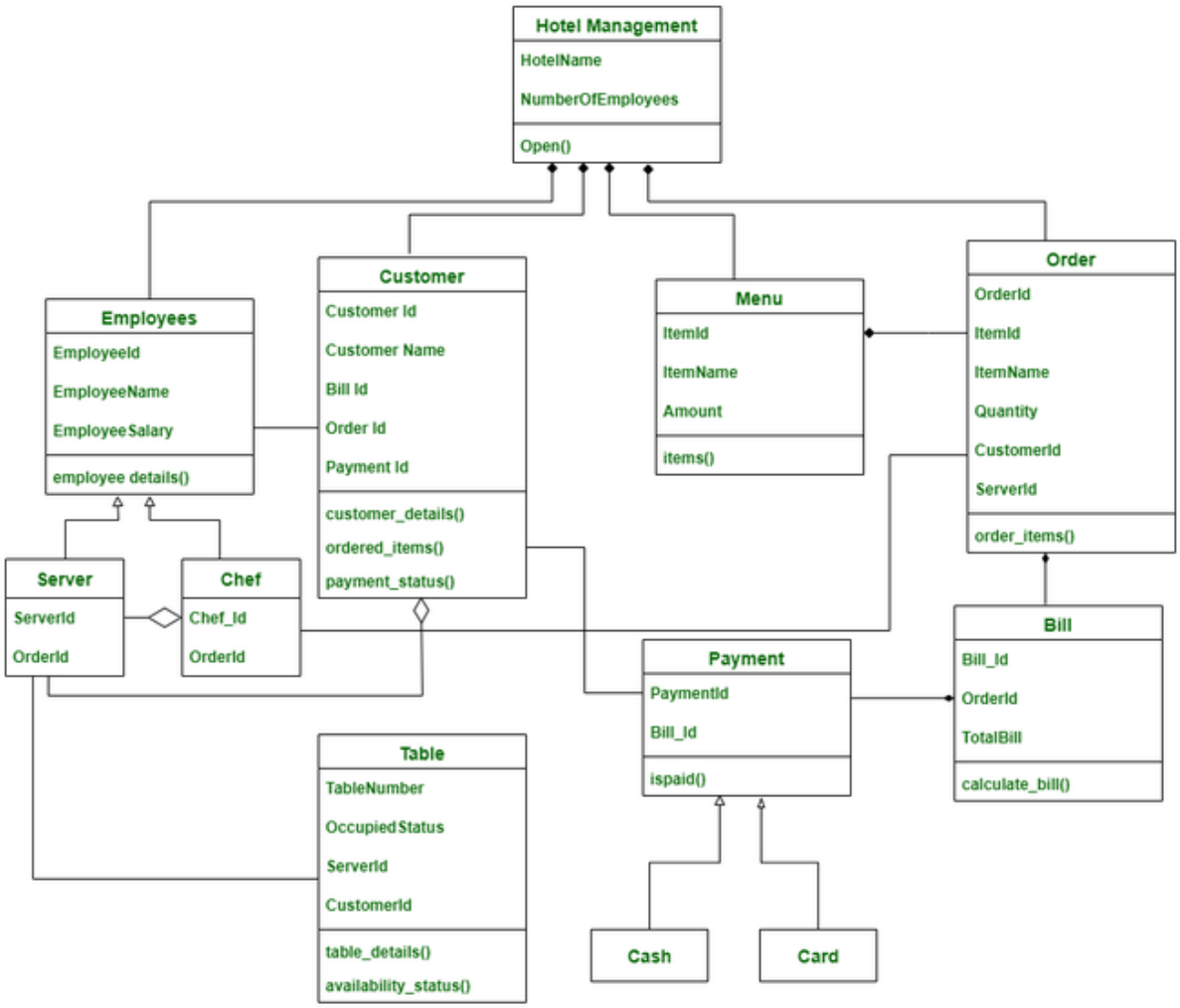


Fig. Class Diagram

Component Diagram

Component diagram of Hotel Management System which shows components, provided and required interfaces, ports, and relationships be tween the Services, Booking, Rooms, Hotel and Customers. This type of diagrams is used in Component-Based Development (CBD) to describe sys tems with Service-Oriented Architecture (SOA) Hotel Management System UML component diagram describes the organization and wiring of the physical components in a system

Components of UML

Services Component

Booking Component

Rooms Component

Hotel Component

Customers Component

Features of Hotel Management System Component Diagram

You can show the models the components of Hotel Management System.

Model the database schema of Hotel Management System

Model the executables of an application of Hotel Management System Model the system's source code of Hotel Management System