

K.S.R. COLLEGE OF ENGINEERING (Autonomous) –TIRUCHENGODE.

Vision of the Institution

- We envision to achieve status as an excellent educational institution in the global knowledge hub, making self-learners, experts, ethical and responsible engineers, technologists, scientists, managers, administrators and entrepreneurs who will significantly contribute to research and environment friendly sustainable growth of the nation and the world.

Mission of the Institution

- To inculcate in the students self-learning abilities that enable them to become competitive and considerate engineers, technologists, scientists, managers, administrators and entrepreneurs by diligently imparting the best of education, nurturing environmental and social needs.
- To foster and maintain a mutually beneficial partnership with global industries and Institutions through knowledge sharing, collaborative research and innovation.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Vision of the Department

- To create ever green professionals for software industry, academicians for knowledge cultivation and researchers for contemporary society modernization.

Mission of the Department

- To produce proficient design, code and system engineers for software development.
- To keep updated contemporary technology and fore coming challenges for welfare of the society.

Programme Educational Objectives (PEOs)

PEO1: Figure out, formulate, analyze typical problems and develop effective solutions by imparting the idea and principles of science, mathematics, engineering fundamentals and computing.

PEO2: Competent professionally and successful in their chosen career through life-long learning.

PEO3: Excel individually or as member of a team in carrying out projects and exhibit social needs and follow professional ethics

A.Program Outcomes (POs)Engineering Graduates will be able to :	
PO1	Engineering knowledge: Ability to exhibit the knowledge of mathematics, science, engineering fundamentals and programming skills to solve problems in computer science.
PO2	Problem analysis: Talent to identify, formulate, analyze and solve complex engineering problems with the knowledge of computer science. .
PO3	Design/development of solutions: Capability to design, implement, and evaluate a computer based system, process, component or program to meet desired needs.
PO4	Conduct investigations of complex problems: Potential to conduct investigation of complex problems by methods that include appropriate experiments, analysis and synthesis of information in order to reach valid conclusions.
PO5	Modern tool Usage: Ability to create, select, and apply appropriate techniques, resources and modern engineering tools to solve complex engineering problems.
PO6	The engineer and society: Skill to acquire the broad education necessary to understand the impact of engineering solutions on a global economic, environmental, social, political, ethical, health and safety.
PO7	Environmental and sustainability: Ability to understand the impact of the professional engineering solutions in societal and Environmental contexts and demonstrate the knowledge of, and need for sustainable development.
PO8	Ethics: Apply ethical principles and commit to professional ethics and responsibility and norms of the engineering practices.
PO9	Individual and team work: Ability to function individually as well as on multi-disciplinary teams.
PO10	Communication: Ability to communicate effectively in both verbal and written mode to excel in the career.
PO11	Project management and finance: Ability to integrate the knowledge of engineering and management principles to work as a member and leader in a team on diverse projects.
PO12	Life-long learning: Ability to recognize the need of technological change by independent and life-long learning.
B. Program Specific Outcomes (PSOs)	
PSO1	Develop and Implement computer solutions that accomplish goals to the industry, government or research by exploring new technologies.
PSO2	Grow intellectually and professionally in the chosen field

K.S.R. COLLEGE OF ENGINEERING (Autonomous)

R 2016 SEMESTER -VII

16CS721

SOFTWARE TESTING LABORATORY

L T P C 0 0 3 2

Prerequisite:

Basic knowledge in C#.NET and Java programming (16CS422,16CS621)

Objectives:

- To learn software testing techniques for online systems.
- To be familiar with GUI object testing.
- To learn exception concept in software testing.

List of Experiments:

1. Design and Document the Employee Information System.
2. Practice the Data Flow Diagram.
3. Create a use Case Design, class diagram, sequence diagram and activity diagram for online reservation system.
4. Using Test Link create Test case along with Test plan, create users and assign roles.
5. Using jmeter analyze the performance of web application.
6. Creating a Bug-report in Bugzilla and Prepare a Graphical Reports.
7. Create a simple TestNG project.

Software Requirements:

Java eclipse, Xampp, Apache Server, Bugzilla, Jmeter, MySQL, ActivePerl and Selenium IDE.

Total : 45 Periods

Course Outcomes:

On Completion of this course , the student will be able to

- Execute open test tool selenium.
- Apply software testing techniques for various applications.
- Implement test cases for GUI objects.
- Implement the RegressionTesting.
- Develop exception testing.

		16CS721- Software Testing Laboratory													
CO	Course Outcomes	Programme Outcomes													
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
C405.1	Execute open test tool selenium.	2	2	3	3	3	-	1	-	-	3	2	-	3	3
C405.2	Apply software testing techniques for various applications.	2	3	3	3	2	-	1	-	-	3	2	-	3	3
C405.3	Implement test cases for GUI objects.	2	2	3	3	3	-	1	-	-	3	2	-	3	3
C405.4	Implement the Regression Testing.	2	2	3	3	3	-	1	-	-	3	2	-	3	3
C405.5	Develop exception testing.	2	2	3	3	3	-	1	-	-	3	2	-	3	3
Maximum Appeared value		2	2	3	3	3	-	1	-	-	3	2	-	3	3

Index

S.NO	Date	Name of the Program	Page No	Marks	Staff Signature
1.		To Design and Document the employee Information System	11		
2.		Data Flow Diagram	27		
3.		Create a use case design class diagram sequence diagram activity diagram for online reservation system	39		
4.		Using Test Link create Test case along with Test plan, create users and assign roles.	47		
5.		How to Use JMeter for Performance & Load Testing	67		
6.		Bugzilla	83		
7.		Create a Simple TestNG Project	117		
		Average:			

Ex No.1.	To Design and Document the employee Information System
Date:	

AIM:

To Design and Document the employee Information System

ALGORITHM:

- 1.Create a Login Table contains user name and password
- 2.Create a Field Name with Data Type/Size Description
- 3.Create ADD DEPARTMENT Table which contains details of different departments
- 4.Create ADD ROLES Table contains details of different roles
5. Create ATTENDEELOGIN-This table is used to enter the details of the participants who will be participating in the programs
- 6.Stop

INTRODUCTION & OBJECTIVE

1. Employee information system is a database which keeps the record of all the details pertaining to employees in an organization.
2. Organizations have always had to collect and use personal information from and about employees to comply with tax and other laws, to administer benefits, to operate their businesses.
3. The objective of the project is to improve and enhance the “Employee information system” module available in IDRBT’s intranet for deployment on virtualized server.
4. The same could not be migrated to the virtualized environment owing to incompatibility of the existing code with the virtualized environment

Project Scope:

The scope of the project is to store and access the database consisting of employee's personal details and the database can be shared with the concerned department to complete joining formalities

We can also store and access the database consisting of program details.

Project Deliverables

We can store the employee's personal details for internal accessibility.

We can post the program details on intranet and we can also prepare the program timetable.

Project Overview

The intranet consist of many different modules like Management, Activities, formats, Employee module, etc.

All the above mentioned modules also have many sub-modules inside each of them. Thus, to store all the data, the database used is oracle.

Most of the data tables are interlinked with the employee module and hence the employee module is developed first.

The employees will be having limited access to the website based on their department, designation and role.

The program module keeps a record of all the upcoming programs and the program timetables

This module is also linked with employee module
The admin will have all the rights and full accessibility to intranet i.e he has the right to add, modify ,or delete any data .

EMPLOYEE

All the above added fields are stored in the database and are retrieved in dropdown boxes in the employee table.
The employee table contains the following fields and after entering these details a unique employee id will be generated for every employee .

INPUT

Department
Designation
Name
Gender
Date of birth
Address
Email
Contact number
Category
Joining date
Valid date
Deputation from
Deputation to
Work area
Reporting to

PROGRAM MODULE

The program module consists of all the details about the upcoming programs.
These details include program name, coordinator, objective of the program, nature of the program, starting and ending dates of the program, fee structure, etc

INPUT

Program name
Coordinator
Objective of the program
Nature of the program
From date
To date
Timetable
Feedback
Fee structure
Amount

There are options for modifying or deleting the program table. These options are available only for admin.

PROGRAM MANAGEMENT MODULE

This module is used to prepare timetables for different programs.

The admin is required to enter the following details first

INPUT

Program name

Number of sessions

Depending on the number of sessions entered the page is accordingly redirected

For example if the number of sessions entered is 4, the page is redirected to a page which has 4 textboxes for 4 sessions.

The input details are as follows

Duration (number of days)

Select program date

Session1

Faculty1

Session2

Faculty2

Session3

Faculty 3

Session 4

Faculty 4

Duration of the program is calculated by subtracting from date and to date which are entered in the program module.

The program date is dropdown column with options as dates from ‘from date’ to date’.

Faculty is also a dropdown column which has the employee names from the employee table as options

ADD PARTICIPANT

The details of participants who are participating in the programs can be added.

The input details are as follows

Pass number

Date

Participant Name

Program name

Status

Program code

Organization

Start date

Designation

End date

Address

City

Vehicle number

Time

PUBLISH

Every program will have dates allotted for group photograph and special lunch .

All the related details will be entered into publish table

The input details are as follows

Program name

According to the program name entered, the page will be redirected to another page which checks the program name from the program table and the values of from date ,to date and the program coordinator are returned to the page.

After this the following details should be given as input

GROUP PHOTOGRAPH

Date

Time

Venue

SPECIAL LUNCH

Date

Time

Venue

INPUT AND OUTPUT SCREENS

SCREENS

LOGIN PAGE

1)Login Table contains user name and password

Field Name Data Type/Size Description

UNAME Varchar(20) User name

PASSWORD Varchar(20) Password

HOME PAGE

ADD DEPARTMENT

DEPMT Table contains details of different departments

Field Name Data Type/Size Description

DEPTID Number Department id

DEPT Varchar(20) Department

VIEW DEPARTMENT

ADD DESIGNATION

DESGN Table contains details of different designations

Field Name Data Type/Size Description

DESIGID Number Designation id

DESIG Varchar(20) Designation

ADD ROLES

ROLES Table contains details of different roles

Field Name Data Type/Size Description

SNO Number Serial number

ROLES Varchar(20) Role

ADD EMPLOYEE

EMPLOYEE4 Table contains the details of employees

Field Name Data Type/Size Description
EID Number Employee id
DEPT Varchar(20) Department
DESIG Varchar(20) Designation
ENAME Varchar(20) Employee name
GENDER Varchar(20) Gender
DOB Varchar(20) Date of birth
ADDRESS Varchar(50) Address
MAIL Varchar(20) Email address
CONT Varchar(20) Contact number
CATEGORY Varchar(20) Category
JD Varchar(20) Joining date
VALIDDATE Varchar(20) Valid date
DEPTFROM Varchar(20) Deputation from
DEPTTO Varchar(20) Deputation to
WORKAREA Varchar(20) Work area
REPTTO Varchar(20) Reporting to

EMPLOYEE VIEW

ADD PROGRAMS

PROGRAMS Table contains the details of programs
Field Name Data Type/Size Description
ID Number Serial Number
PROGRAM Varchar(20) Program name
CORD Varchar(20) Coordinator
OBJ Varchar(20) Objective
NATURE Varchar(20) Nature of the program
FD Date From Date
TD Date To Date
TT Varchar(20) Timetable
FB Varchar(20) Feedback
FEE Varchar(20) Fee
AMOUNT Varchar(20) Amount

MODIFY PROGRAM

PROGRAM MANAGEMENT

PREPARE PROGRAM TIMETABLE BY ENTERING NUMBER OF SESSIONS.

PREPARE PROGRAM TIMETABLE

SESS4-This table is used to manage programs when there are 4 sessions
Field Name Field Name Description

PROGRAM Varchar(20) Program name
DURATION Varchar(20) Duration of the program
PROGDATE Date Program Date
SESS1 Varchar(20) Session 1
FAC1 Varchar(20) Faculty
SESS2 Varchar(20) Session2
FAC2 Varchar(20) Faculty
SESS2 Varchar(20) Session
FAC2 Varchar(20) Faculty
SESS3 Varchar(20) Session
FAC3 Varchar(20) Faculty
SESS4 Varchar(20) Session
FAC4 Varchar(20) Faculty

ADD PARTICIPANT

ATTENDEELOGIN-This table is used to enter the details of the participants who will be participating in the programs
Field Name Field Name Description
PNO Varchar(20) Program name
DATE1 Date Duration of the program
PNAME Varchar(20) Program Date
PROGNAME Varchar(20) Session 1
STATUS Varchar(20) Faculty
CODE Varchar(20) Session2
ORG Varchar(20) Faculty
SD Date Session
DESIG Varchar(20) Faculty
ED Date Session
ADDRESS Varchar(20) Faculty
CITY Varchar(20) Session
VNO Varchar(20) Faculty
TIMETimeStamp(6) Session
FAC5 Varchar(20) Faculty

These are the other tables used in my project
TERMINATE Table contains the details of employees
Field Name Data Type/Size Description
EID Varchar(20) Employee id
DEPT Varchar(20) Department
DESIG Varchar(20) Designation
ENAME Varchar(20) Employee name
GENDER Varchar(20) Gender
DOB Varchar(20) Date of birth
ADDRESS Varchar(50) Address
MAIL Varchar(20) Email address

CONT Varchar(20) Contact number
CATEGORY Varchar(20) Category
JD Varchar(20) Joining date
VALIDDATE Varchar(20) Valid date
DEPTFROM Varchar(20) Deputation from
DEPTTO Varchar(20) Deputation to
WORKAREA Varchar(20) Work area
REPTTO Varchar(20) Reporting to
PROG1 –This table is used to add any new programs
Field Name Field Name Description
SNO Number Serial Number
PROGRAM Varchar(20) Program name
NATURE Varchar(20) Nature of the program
DURATION –This table is used to store the duration of the programs.
Field Name Field Name Description
SNO Number Serial Number
DURATION Varchar(20) Duration of the program

SESS5-This table is used to manage programs when there are 5 sessions
Field Name Field Name Description
PROGRAM Varchar(20) Program name
DURATION Varchar(20) Duration of the program
PROGDATE Date Program Date
SESS1 Varchar(20) Session 1
FAC1 Varchar(20) Faculty
SESS2 Varchar(20) Session2
FAC2 Varchar(20) Faculty
SESS2 Varchar(20) Session
FAC2 Varchar(20) Faculty
SESS3 Varchar(20) Session
FAC3 Varchar(20) Faculty
SESS4 Varchar(20) Session
FAC4 Varchar(20) Faculty
SESS5 Varchar(20) Session
FAC5 Varchar(20) Faculty
SESS6-This table is used to manage programs when there are 6 sessions
Field Name Field Name Description
PROGRAM Varchar(20) Program name
DURATION Varchar(20) Duration of the program
PROGDATE Date Program Date
SESS1 Varchar(20) Session 1
FAC1 Varchar(20) Faculty
SESS2 Varchar(20) Session2
FAC2 Varchar(20) Faculty
SESS2 Varchar(20) Session

FAC2 Varchar(20) Faculty
SESS3 Varchar(20) Session
FAC3 Varchar(20) Faculty
SESS4 Varchar(20) Session
FAC4 Varchar(20) Faculty
SESS5 Varchar(20) Session
FAC5 Varchar(20) Faculty
SESS6 Varchar(20) Session
FAC6 Varchar(20) Faculty

PUBLISH-This table is used to add the date ,time and venue for group photograph and special lunch.

Field Name Field Name Description

PROG Varchar(20) Program name
CORD Varchar(20) Duration of the program

FD Date Program Date

TD Date Session 1

DATE1 Date Faculty

TIME1TimeStamp(6) Session2

VENUE1 Varchar(20) Faculty

DATE2 Date Session

TIME2TimeStamp(6) Faculty

VENUE2 Varchar(20) Session

SYSTEM TESTING

SOFTWARE TESTING TECHNIQUES:

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, designing and coding.

Observation	25	
Record	10	
Total	35	
Signature		

Result:

Thus the Design and Document the employee Information System is documented successfully.

Ex No.2.

Data Flow Diagram

Date:

Aim

To data-flow diagram is a way of representing a flow of a data of a process or a system

Algorithm

Step1 : start the program

Step2 : To create new DFD, select Diagram > New from the toolbar.

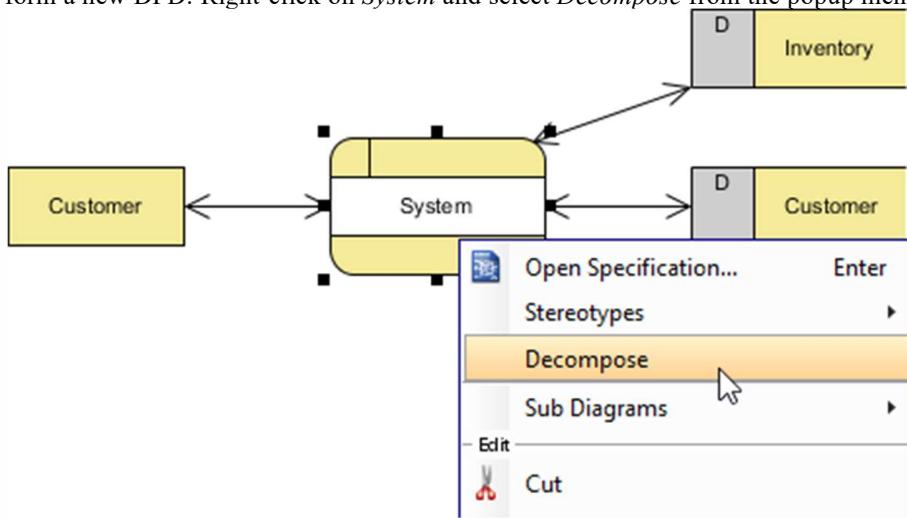
Step3 : In the New Diagram window, select Data Flow Diagram and click Next.

Step4 : Enter Context as diagram name and click OK to confirm

Step5 : We'll now draw the first process. From the Diagram Toolbar, drag Process onto the diagram. Name the new process System.

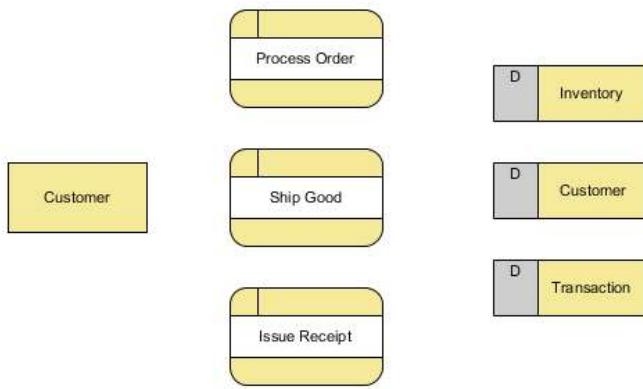
Procedure:

- Instead of creating another diagram from scratch, we will decompose the *System* process to form a new DFD. Right-click on *System* and select *Decompose* from the popup menu.



- The data stores and/or external entities connected to the selected process (*System*) would be referred to in the level 1 DFD. So when you are prompted to add them to the new diagram, click **Yes** to confirm.
Note: The new DFD should look very similar to the Context diagram initially. Every element should remain unchanged, except that the *System* process (from which this new DFD decomposes) is now gone and replaced by a blank space (to be elaborated).
- Rename the new DFD. Right-click on its background and select **Rename...**. In the diagram's name box, enter *Level 1 DFD* and press **ENTER**.

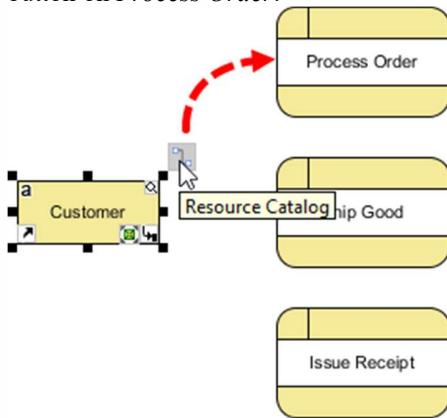
4. Create three processes (Process Order, Ship Good, Issue Receipt) in the center as shown below. That is the old spot for the *System* process and we place them there to elaborate *System*.



Wiring with connection lines for data flows

The remaining steps in this section are about connecting the model elements in the diagram. For example, *Customer* provides order information when placing an order for processing.

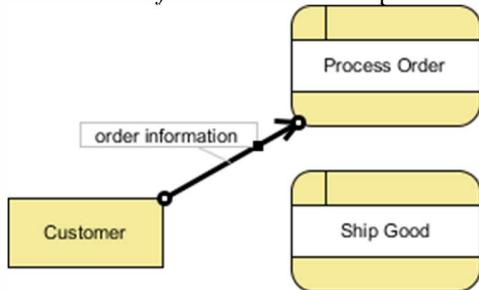
1. Place your mouse pointer over *Customer*. Drag out the **Resource Catalog** icon and release your mouse button on *Process Order*.



2. Select Data Flow from Resource Catalog.

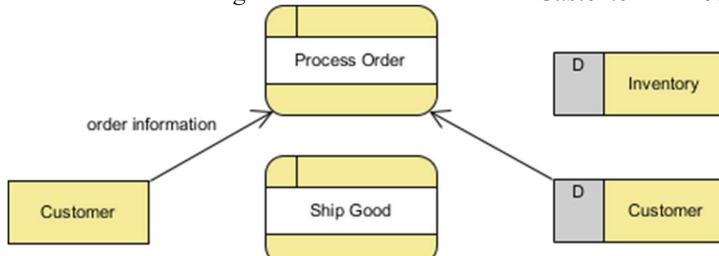


3. Enter *order information* has the caption of flow.



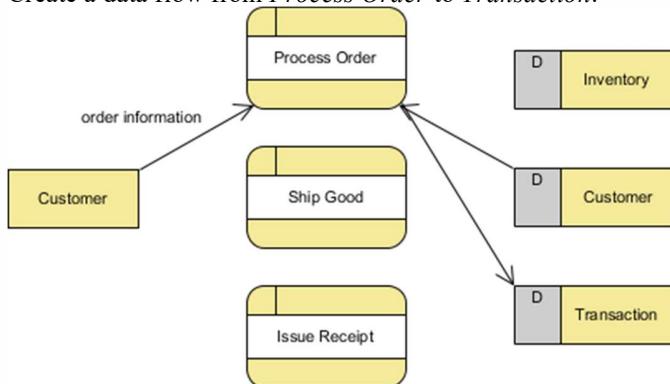
4. Meanwhile, the **Process Order** process also receives customer information from the database in order to process the order.

Use Resource Catalog to create a data flow from *Customer* to *Process Order*.



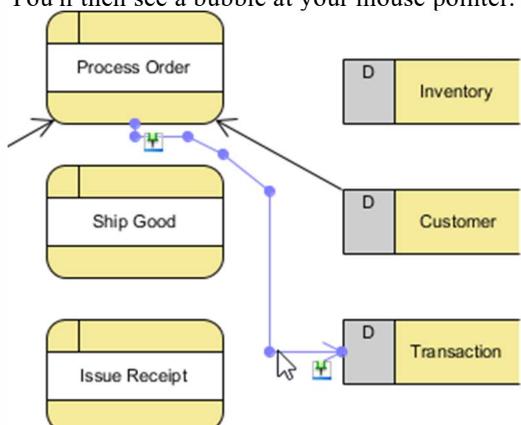
Optional: You can label the data flow "customer information" if you like. But since this data flow is quite self-explanatory visually, we are going to omit it here.

5. By combining the order information from *Customer* (external entity) and the customer information from *Customer* (data store), *Process Order* (process) then creates a transaction record in the database. Create a data flow from *Process Order* to *Transaction*.

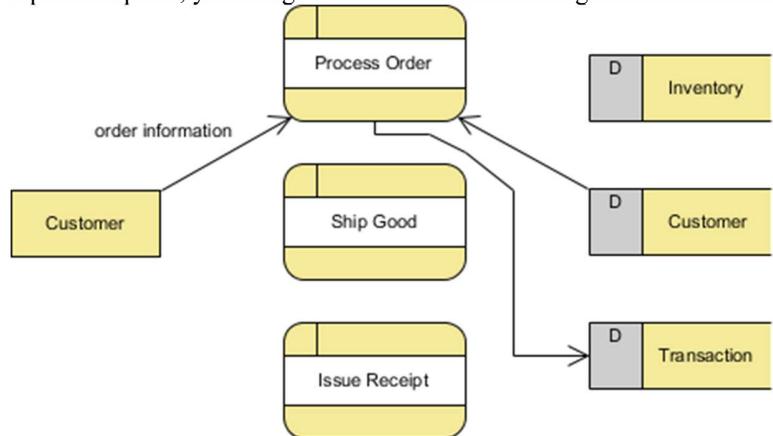


Drawing Tips:

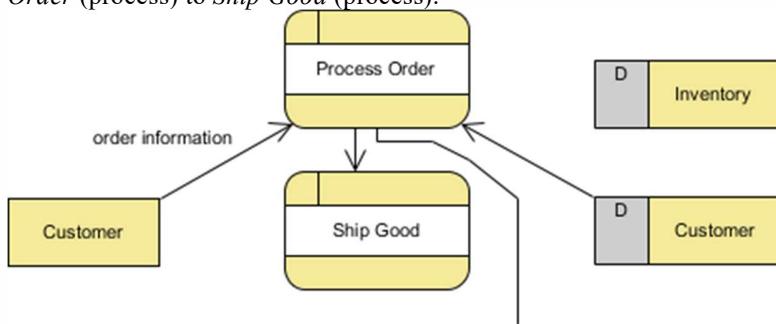
To rearrange a connection line, place your mouse pointer over where you want to add a pivot point to it. You'll then see a bubble at your mouse pointer. Press and drag it to the position desired.



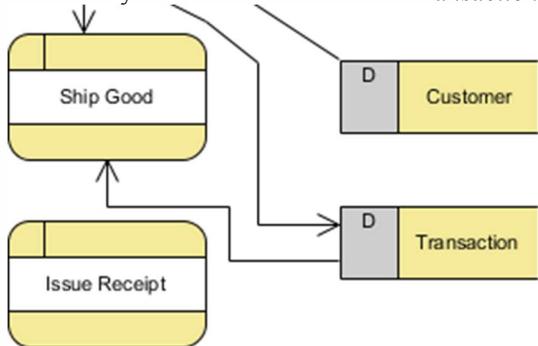
Up to this point, your diagram should look something like this.



6. Once a transaction is stored, the shipping process follows. Therefore, create a data flow from *Process Order* (process) to *Ship Good* (process).

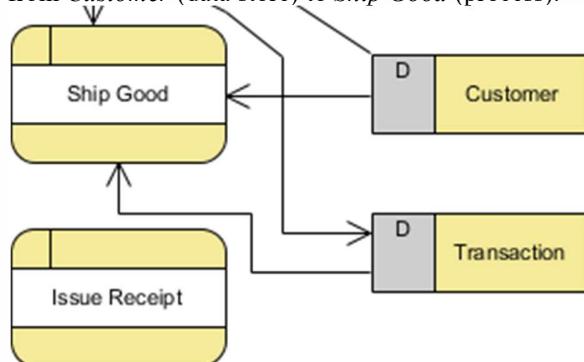


7. *Ship Good* needs to read the transaction information (i.e. The order_) in order to pack the right product for delivery. Create a data flow from *Transaction* (data store) to *Ship Good* (process).

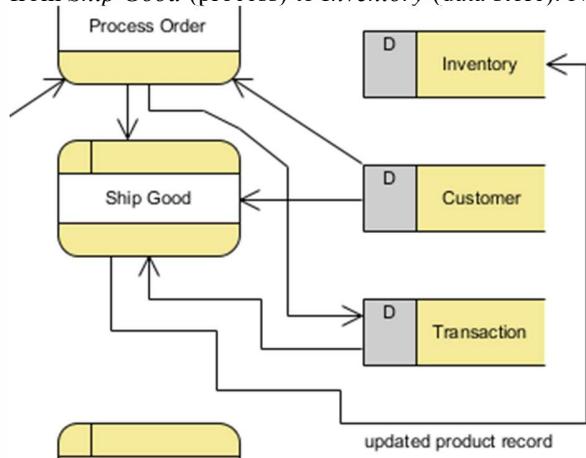


Note: If there is a lack of space, feel free to move the shapes around to make room.

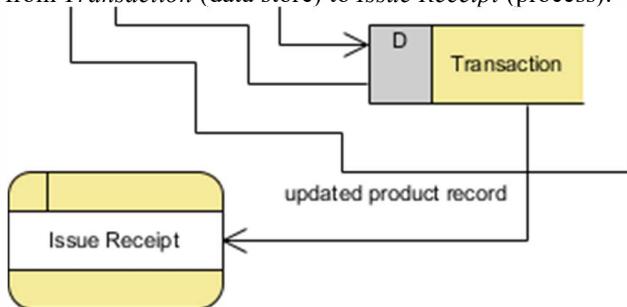
8. *Ship Good* also needs to read the customer information for his/her shipping address. Create a data flow from *Customer* (data store) to *Ship Good* (process).



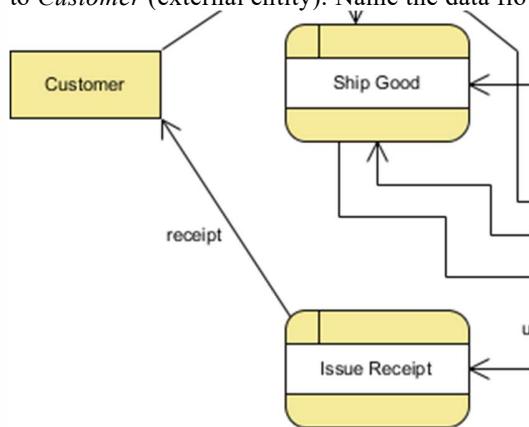
9. *Ship Good* then updates the *Inventory* database to reflect the goods shipped. Create a data flow from *Ship Good* (process) to *Inventory* (data store). Name it *updated product record*.



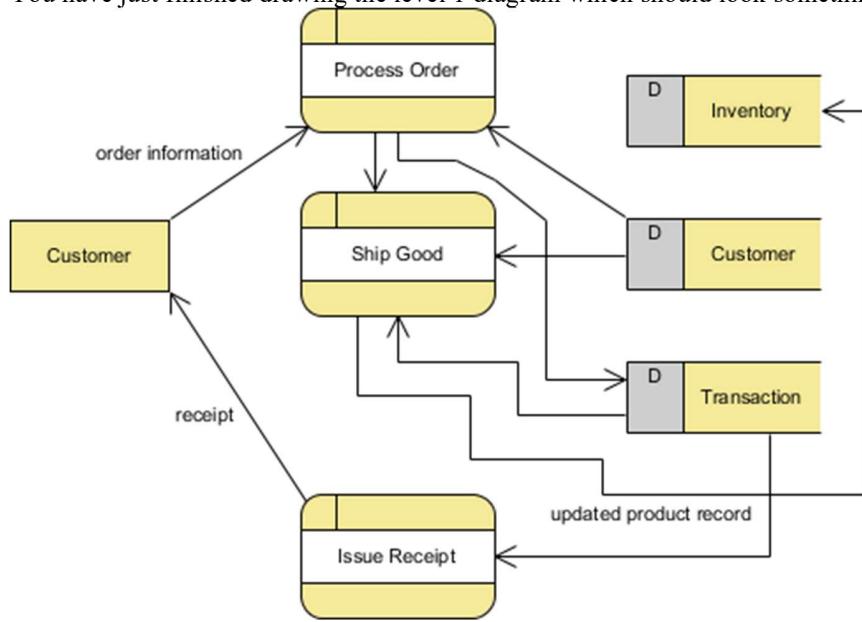
10. Once the order arrives in the customer's hands, the *Issue Receipt* process begins. In it, a receipt is prepared based on the transaction record stored in the database. So let's create a data flow from *Transaction* (data store) to *Issue Receipt* (process).



11. Then a receipt is issued to the customer. Let's create a data flow from *Issue Receipt* (process) to *Customer* (external entity). Name the data flow *receipt*.



You have just finished drawing the level 1 diagram which should look something like this.



Observation	25	
Record	10	
Total	35	
Signature		

Result:

Thus Data flow diagram is a way of representing a flow of a data is studied successfully.

Ex No.3.	Create a use case design class diagram sequence diagram activity diagram for online reservation system
Date:	

Aim:

To create a use case design class diagram sequence diagram activity diagram for online reservation system.

Algorithm:

Step 1: There are two users which are customer and clerk. Both are accessing the Railway System and refund money.

Step 2: They are five ways through accessing the Railway System which are Enquire Ticket Availability, Fill Form, Book Ticket, Cancel Ticket, Pay Fare Amount.

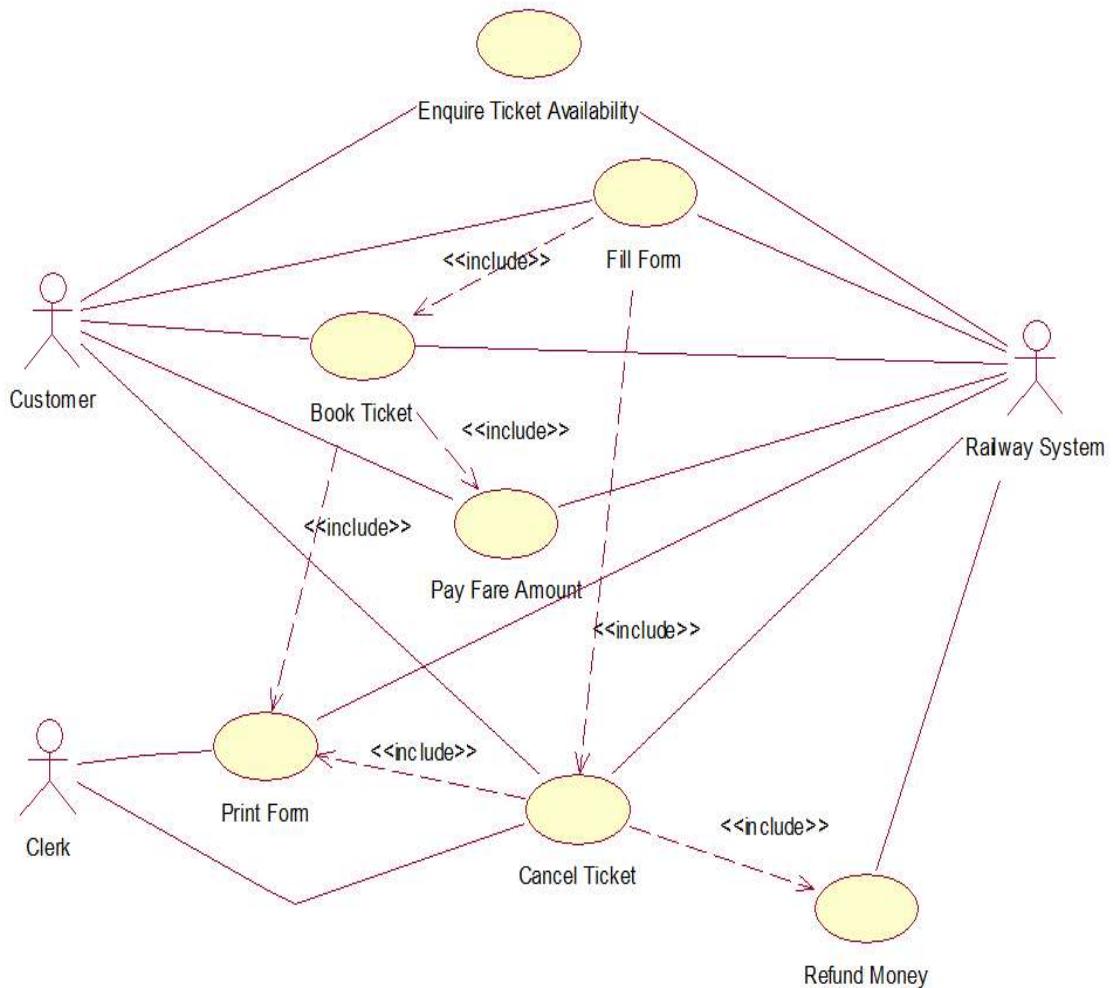
Step 3: Clerk can also access the Refund Money through the Print Form, Cancel Ticket.

Step 4: In the Fill Form we are interacting with the Book Ticket and Cancel Ticket. In the Book Ticket we are interacting with the Pay Fare Amount and Print Form.

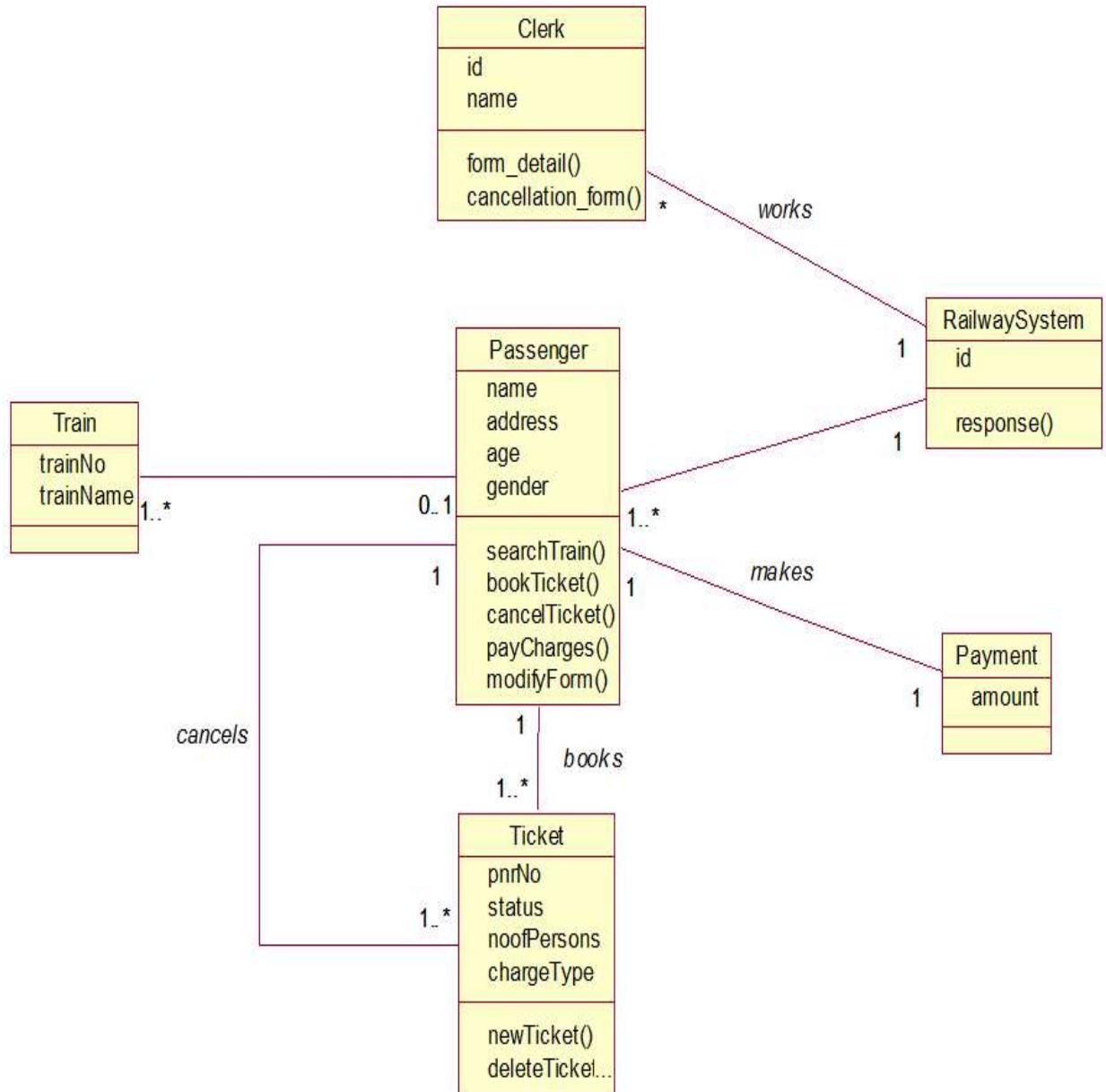
Step 5: In the Cancel Request we are interacting with the print form and the refund money. And from the Clerk it can be connected with the Cancel Ticket.

Procedure:

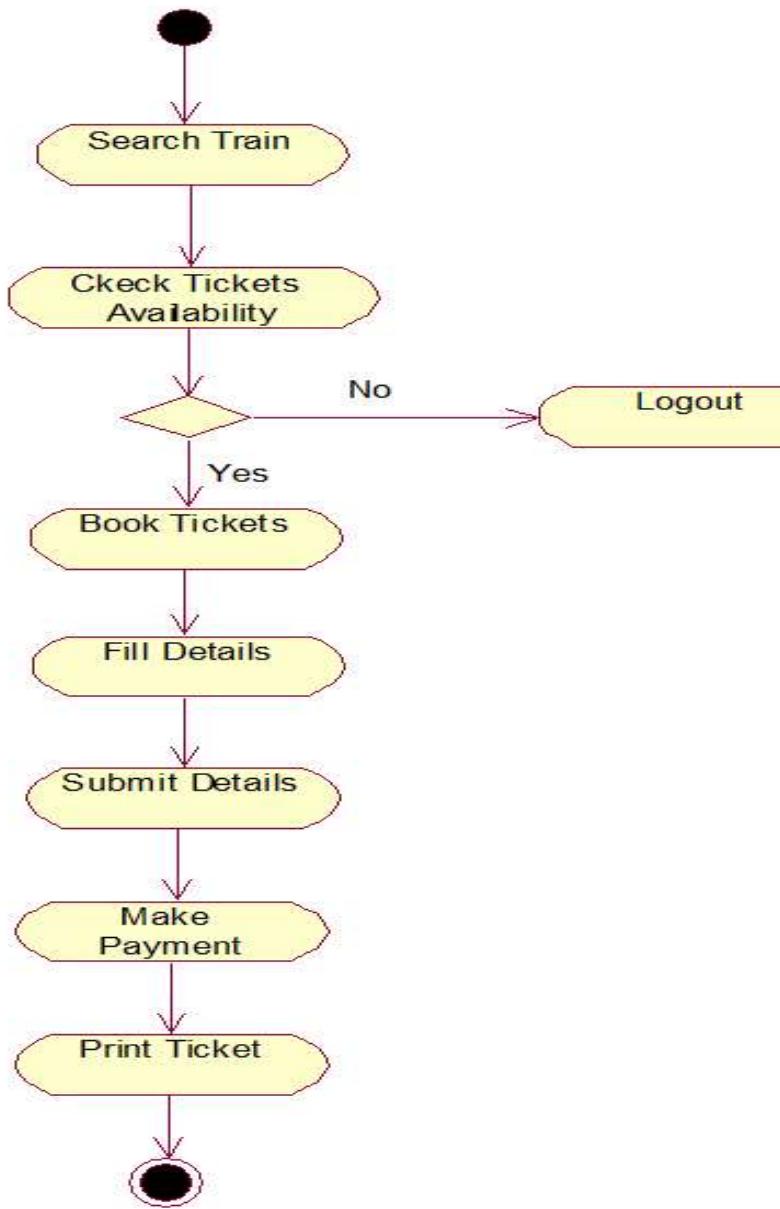
i) case design



ii) class diagram



iii)Activity diagram



Observation	25	
Record	10	
Total	35	
Signature		

Result:

Thus creating use case design, class diagram, sequence diagram, activity diagram for online reservation system was successfully done.

Ex No.4.	Using Test Link create Test case along with Test plan, create users and assign roles.
Date:	

Aim:

Using Test Link create Test case along with Test plan, create users and assign roles.

Algorithm:

Step 1 : Open the Testlink home-page and enter the login details

1. Enter the userID – admin
2. Enter the password
3. Click on the login tab

Step 2: Click on the tab "create" to create a new project.

Step 3: Enter all the required fields in the window like a category for a test project, name of the project, prefix, description, etc. After filling all necessary details, click on tab "Create" at the end of the window.

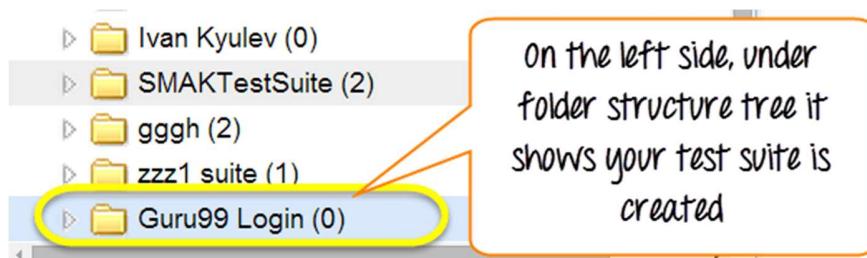
Step 4 :End

Procedure:

Creating a Testcase

Testcase holds a sequence of test steps to test a specific scenario with an expected result. Below steps will explain how to create a test-case along with test steps.

Step 1: Click on the test suite folder on the left side of the panel under a folder tree structure



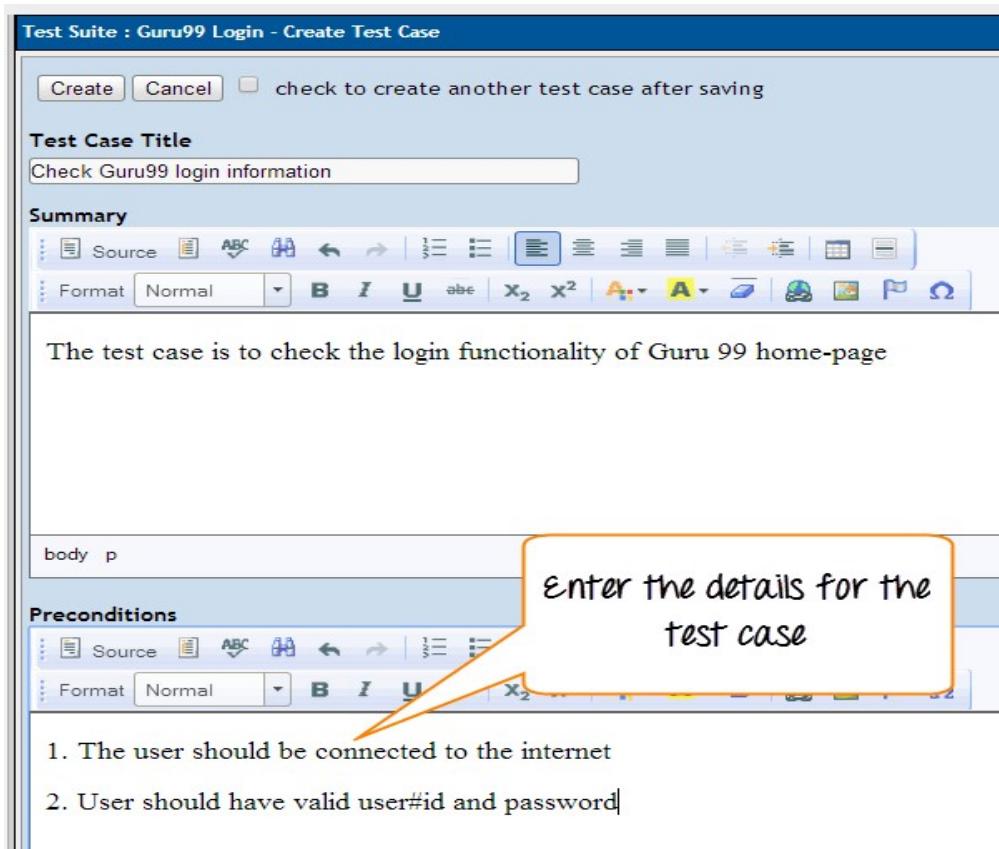
Step 2: Click on the setting icon in the right side panel. List of test case operations will be displayed on the right side panel



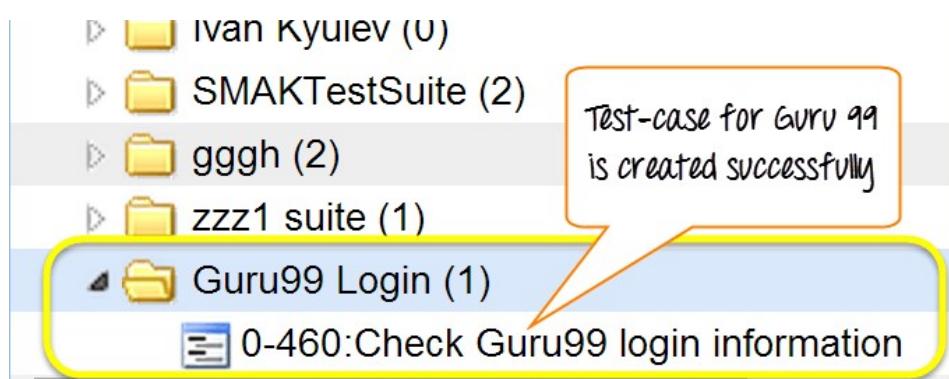
Step 3: New window will open, to create test cases click on create a button in test-case operations



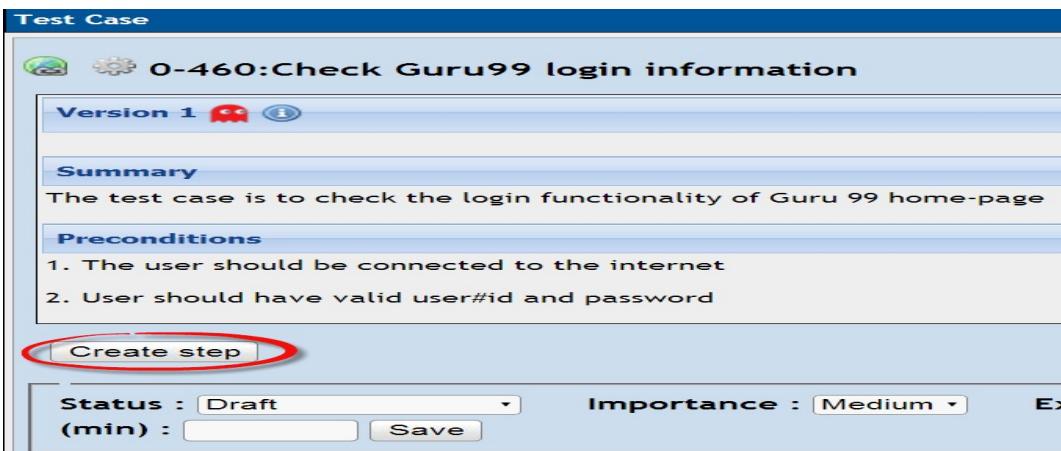
Step 4: Enter the details in the test case specification page



Step 5: After entering the details, click on "create" button to save the details. The test-case for Guru99 is created successfully



Step 6: Click on test-case from the folder, as shown above, it will open a window. Click on "create steps" button in a test case. It will open a test case step editor



Step 7) It will open another window on the same page, in that window you have to enter the following details

1. Enter the step-action for your test case
2. Enter the details about the step action
3. Click save it and add another step action OR click save and exit tab if there is no more test step to add

The screenshot shows a 'Create Step - Test Case: 0-460:Check Guru99 login information - Version: 1' dialog box. It has sections for 'Preconditions' (with items 1 and 2 highlighted by a yellow box) and 'Step actions' (with a table showing an action to open the Guru 99 website). A callout bubble points to the Preconditions section with the text: 'You can add and save test-step by entering details in the step action and expected result box'. Buttons at the bottom include 'Save', 'Save & exit', and 'Cancel'.

#	Step actions	Expected Results	Execution
1	① Open Guru 99 website body p	② The website should be opened body p	Manual

Step 8) Once you save and exit the test step, it will appear like this

Version 1 🧨 ⓘ

Summary

The test case is to check the login functionality of Guru 99 home-page

Preconditions

1. The user should be connected to the internet
2. User should have valid user#id and password

#	Step actions	Expected Results	Execution
1	Open Guru 99 website	The website should be opened	Manual  
2	Enter username in the username textbox	Text-box should accept the entered data	Manual  
3	Enter password in the password text-box	Text-box should accept the entered data	Manual  
4	Click on "sign in" button	Login should success and navigate to the home page	Manual  

Assigning test case to test plan

For test case to get executed, it should be assigned to test plan. Here we will see how we can

assign a test-case to test plan.



Step 1) Click on the setting icon on the test panel. It will show the list of operations.

Step 2) Click on "Add to Test Plans"

Test Case

0-460:Check Guru99 login information

Edit Delete Move / Copy New sibling Export Print view New version Deactivate this version Add to Test Plans

Execution History

Version 1 🧨 ⓘ

Summary

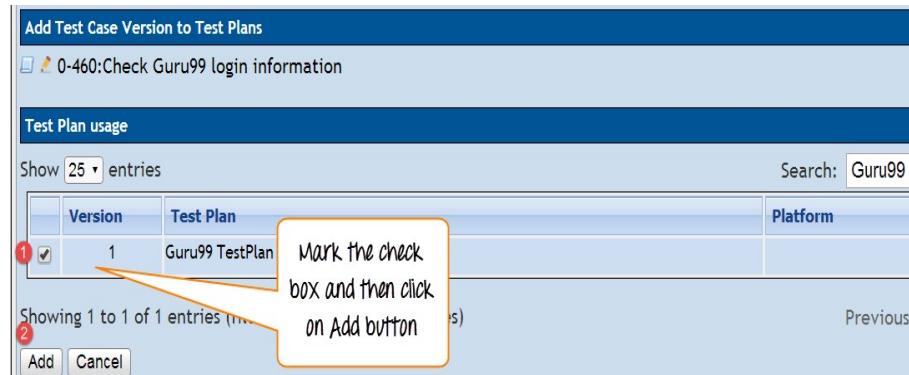
The test case is to check the login functionality of Guru 99 home-page

Preconditions

From the given list of operations, select "Add to Test Plans"

Step 3) New window will open, search your project "Guru99"

1. Mark the checkbox against your test plan
2. Click on add button



This will add your test case to your Test Plan.

Creating Users and Assigning Roles in TestLink

Testlink provides User management and authorization features.
Below is list of default roles in Testlink and their rights -

Role	Test Cases	Test Metrics
Guest	View	View
Tester	Execute	View
Senior Tester	Edit & Execute	View
Leader & Admin	Edit & Execute	Edit & Execute

Step 1: From the Testlinks home-page, click on users/roles icon from the navigation bar



Step 2: Click Create

ztrimble	Zach	Trimble
zztest	zhang	zhong
zzz	zzz	zzz
	Create	Export

Step 3: Fill out all the users details and click the "Save" button

User Management - Account Settings

New User	View Users	View roles	Assign Test Project roles	Assign Test Plan roles
User details				
Login	Jamesguru			
First Name	Jamesguru			
Last Name	Bryan			
Password	*****			
Email	jamesguru@gmail.com			
Role	Senior Test Engineer			
Locale	English (wide/UK)			
Authentication method	Default(DB)			
Active	✓			
Fill out all the details about the user and click on save button				
Save	Cancel			

Here in the list, we can see the users have been created

jafgc	João	joao@fgc@gmail.com	leader
james	jamer	abc@163.om	leader
Jamesguru	Jamesguru	jamesguru@gmail.com	Senior Test Engineer
jan	jan	sivaprakasam	leader
		jan.sivaprakasam@cancer.or...	leader

Step 4: Allotting test project role to the user,

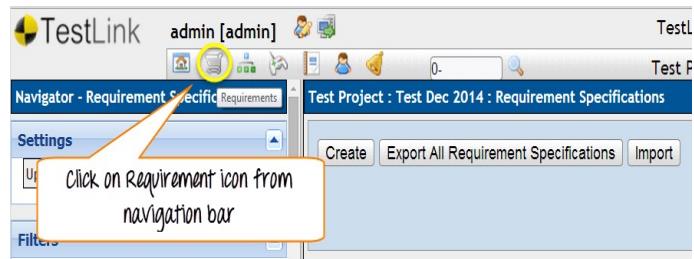
1. Click on "**Assign Test Project Roles**" tab
2. Choose the project name
3. Select the user role from the drop down

User Management - Assign roles

View Users	View roles	Assign Test Project roles	Assign Test Plan roles
Test Project Guru 99 -			
Set roles to Senior Test Engineer			
Show 25 entries			
User	Test Project Role (Guru 99 -)		
Jamesguru (Jamesguru Bryan)	<inherited> Senior Test Engineer		

Writing Requirements:

Step 1: From the navigation bar select the "Requirements Link", it opens the Requirement page.



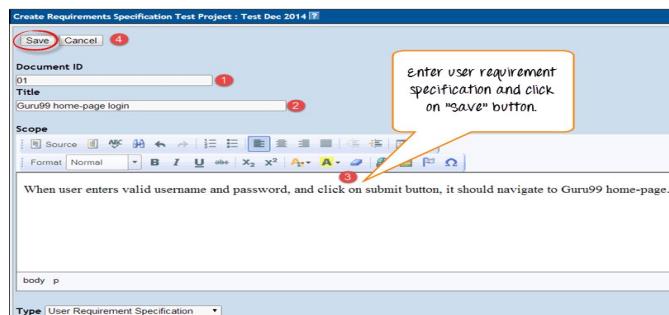
Step 2: From the requirement page, on the right side of the panel click on "create" button



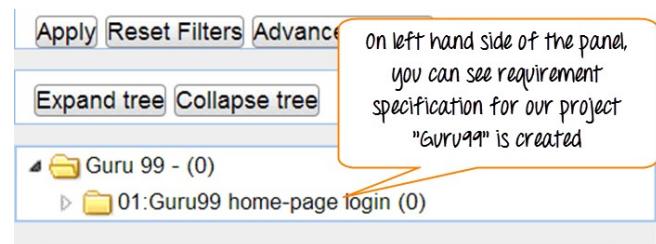
Step 3: A new window will open, enter all the details like

1. Document ID
2. Title name
3. Requirement Description
4. And Click "Save" button

For the type, you can choose the option from the drop-down- here we chose "**User Requirement Specification**"



Step 4: It should create Requirement specification and displayed on the left side panel under the project "Guru99".



Step 5: Select the setting button from requirements specification home-page. It will open another window.



Step 5: Click "Create" tab under Requirement Operations.



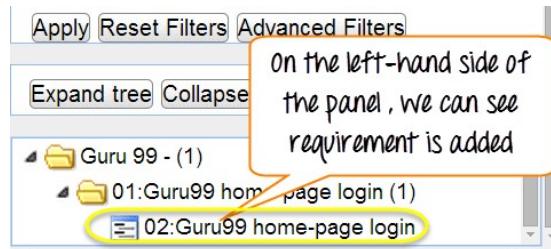
Step 6: Fill out all the specified details and click the "Save" button

1. Enter the document ID
2. Enter the title name
3. Enter the description
4. Enter the status-whether it's in draft, rework, review, not testable, etc. Here we chose valid
5. Enter the type – user interface, non-functional, informational, feature, etc. Here we chose use case
6. Enter the number of test cases needed
7. Enter "Save" button at the end

The dialog box contains fields for Document ID (02), Title (Guru99 home-page login), and Scope. It also includes a note about password encryption and navigation to the Guru99 home-page. At the bottom, there are buttons for Save, Cancel, and a checkbox for creating another requirement after saving.

Note: To add more requirements you can mark the check-box and click save button

On the left side of the panel, we can see that the requirement is added.



Observation	25	
Record	10	
Total	35	
Signature		

Result:

Thus using test link, test case along with test planning, creating users and assigning roles are done successfully.

Ex No.5.	How to Use JMeter for Performance & Load Testing
Date:	

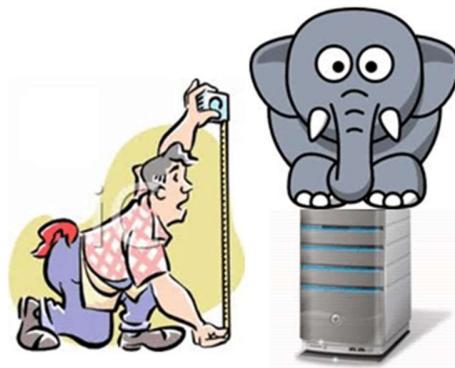
AIM:

To create jmeter analyse the performance of web application

PROCEDURE:

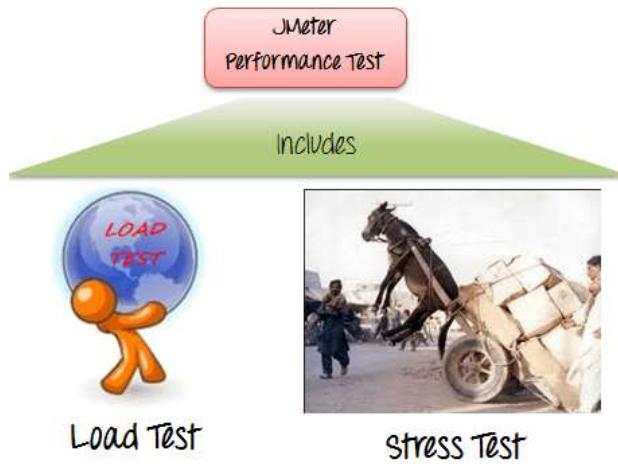
- Step:1 Create a Performance Test Plan in JMeter
- Step:2 Add Thread Group
- Step:3 Adding JMeter elements
- Step:4 Adding Graph result
- Step:5 Run Test and get the test result
- Step:6 Stop

Performance [Testing](#) is crucial to determine that the web application under test will satisfy **high load** requirements. It can be used to analyze overall server performance under heavy load.



Apache JMeter testing tool offers following **benefit** in [Performance Testing](#)

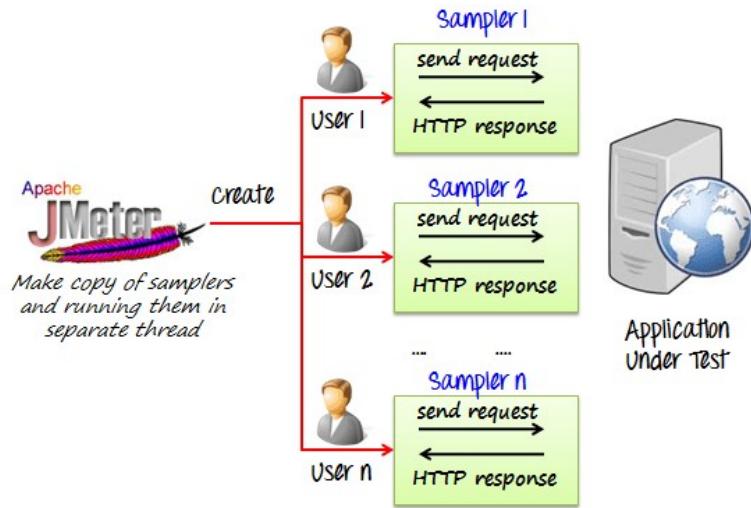
- JMeter can be used to test the performance of both **static** resources such as [JavaScript](#) and HTML, as well as **dynamic** resources, such as JSP, Servlets, and AJAX.
- JMeter can **discover** maximum number of concurrent users that your website can handle
- JMeter provides a variety of graphical analyses of performance reports.



JMeter Performance Testing includes:

- **Load Testing:** Modeling the expected usage by simulating multiple user access the [Web services](#) concurrently.
- **Stress Testing:** Every web server has a maximum load capacity. When the load goes beyond the limit, the web server starts responding slowly and produce errors. The purpose of the [Stress Testing](#) is to find the maximum load the web server can handle.

The figure below shows how JMeter load Testing simulates the heavy load:



Create a Performance Test Plan in JMeter

In this tutorial, we are doing a performance analysis of Google.com for 1000 users. Before testing the performance of target web application, we should determine-

- **Normal Load:** Average number of users visit your website
- **Heavy Load:** The maximum number of users visit your website
- What is your **target** in this test?

Here is the **roadmap** of this practical example

Add Thread Group

Add JMeter elements

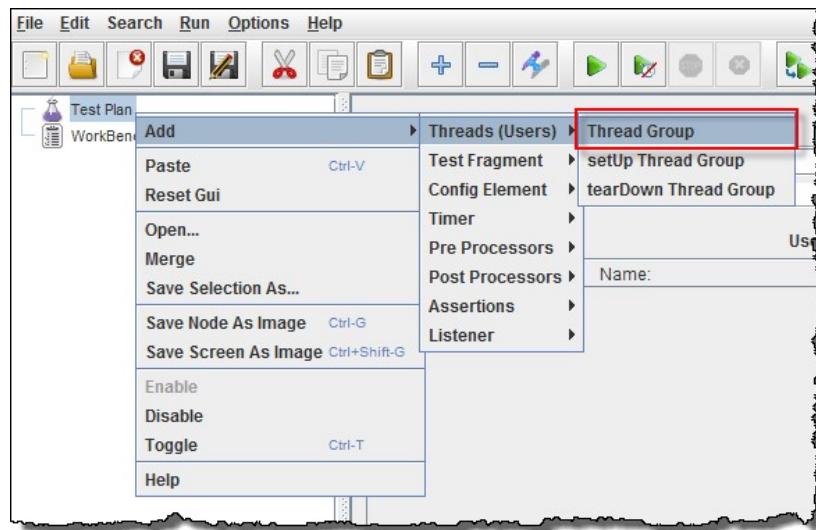
Add Graph result

Run Test & Get Result

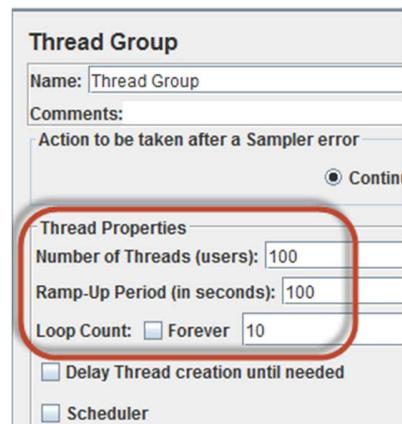
Step 1) Add Thread Group

1. Start JMeter
2. Select **Test Plan** on the tree
3. Add **Thread Group**

Right click on the "Test Plan" and add a new thread group: **Add -> Threads (Users) -> Thread Group**

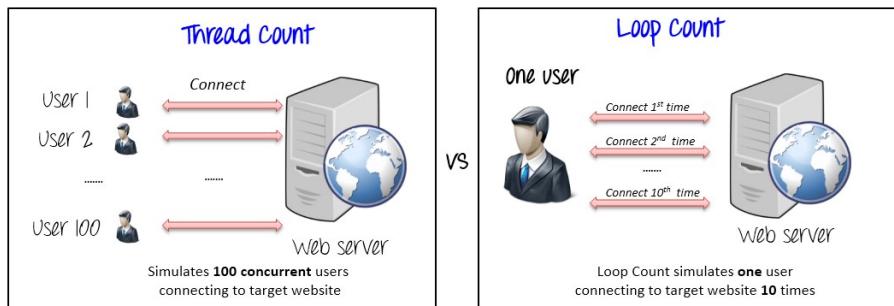


In the Thread Group control panel, enter Thread Properties as follows:



- **Number of Threads:** 100 (Number of users connects to the target website: 100)
- **Loop Count:** 10 (Number of time to execute testing)
- **Ramp-Up Period:** 100

The Thread Count and The Loop Counts are **different**.



Ramp-Up Period tells JMeter how long to **delay** before starting the next user. For example, if we have 100 users and a 100-second Ramp-Up period, then the delay between starting users would be 1 second (100 seconds / 100 users)

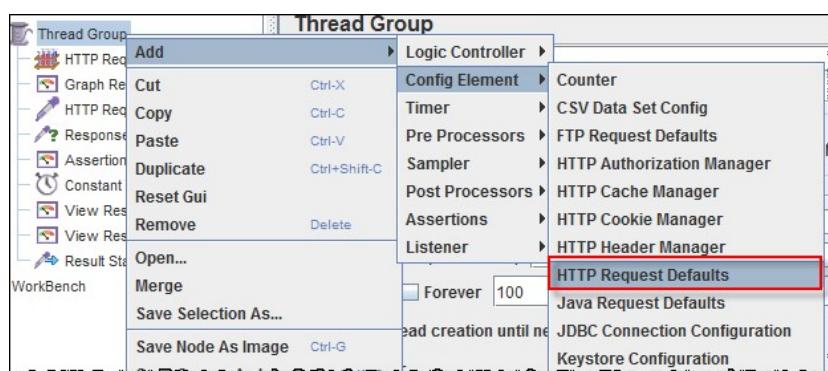


Step 2) Adding JMeter elements

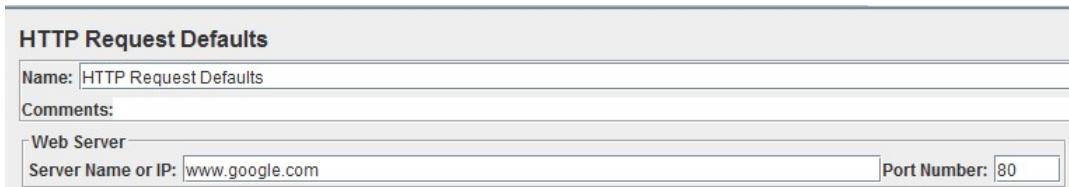
Now we determine what JMeter elements in this test. The elements are

- **HTTP request Default**

This element can be added by right-clicking on the Thread Group and selecting: **Add -> Config Element -> HTTP Request Defaults**.

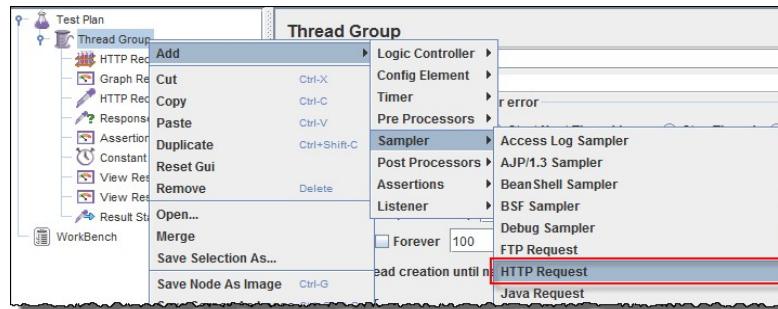


In the HTTP Request Defaults control panel, enter the Website name under test (<http://www.google.com>)

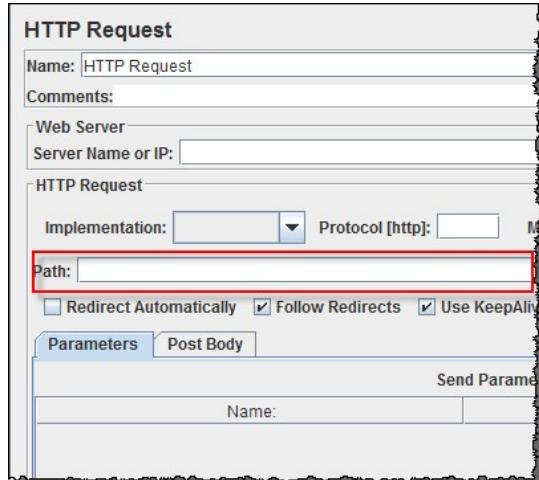


- **HTTP Request**

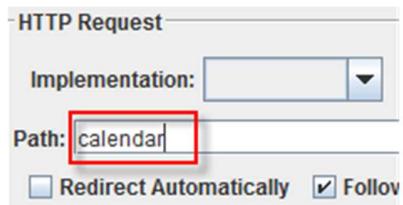
Right-click on Thread Group and select: **Add -> Sampler -> HTTP Request**.



In HTTP Request Control Panel, the Path field indicates which **URL request** you want to send to Google server.



For example, if you enter "*calendar*" in Path field. JMeter will create the URL request <http://www.google.com/calendar> to Google server



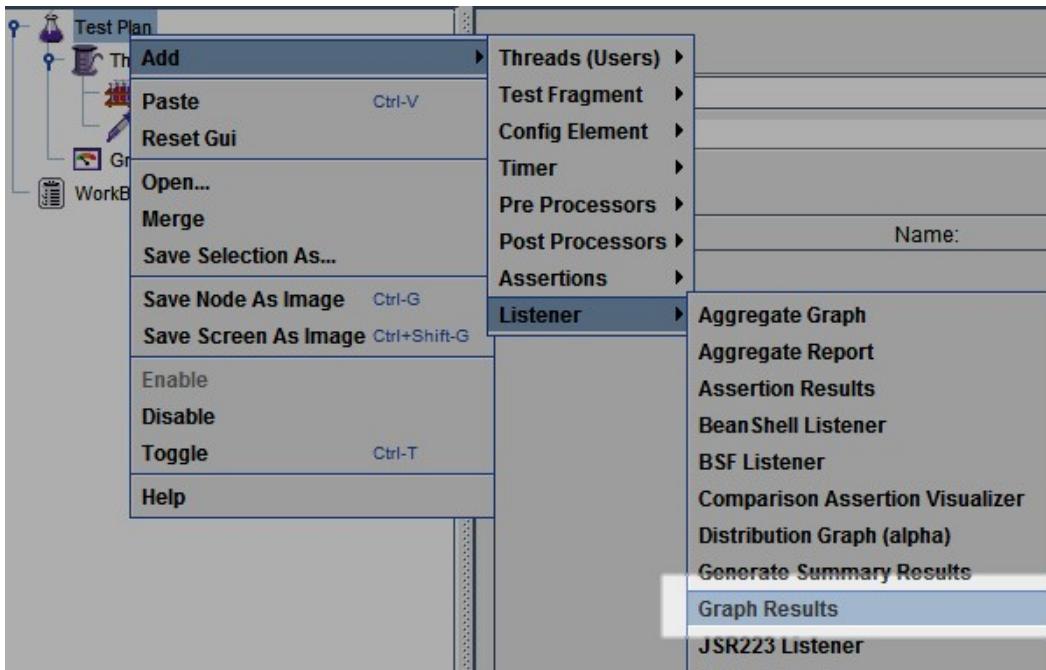
If you keep the Path field **blank** JMeter will create the URL request <http://www.google.com> to Google server.

In this test, you keep the Path field blank to make JMeter create the URL request <http://www.google.com> to Google server.

Step 3) Adding Graph result

JMeter can show the test result in Graph format.

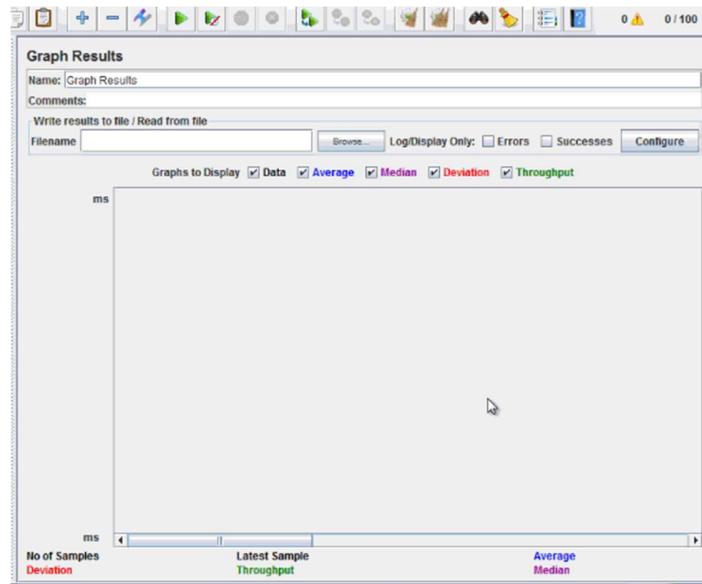
Right click Test Plan, Add -> Listener -> Graph Results



Step 4) Run Test and get the test result

Press the **Run** button (Ctrl + R) on the Toolbar to start the software testing process. You will see the test result display on Graph in the real time.

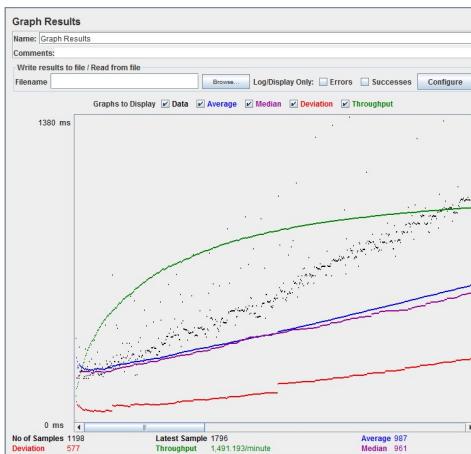
The picture below presents a graph of a test plan, where we simulated 100 users who accessed on website www.google.com.



At the bottom of the picture, there are the following statistics, represented in colors:

- Black: The total number of current samples sent.
- Blue: The current average of all samples sent.
- Red: The current standard deviation.
- Green: Throughput rate that represents the number of requests per minute the server handled

Let analyze the performance of Google server in below figure.



To analyze the performance of the web server under test, you should focus on 2 parameters

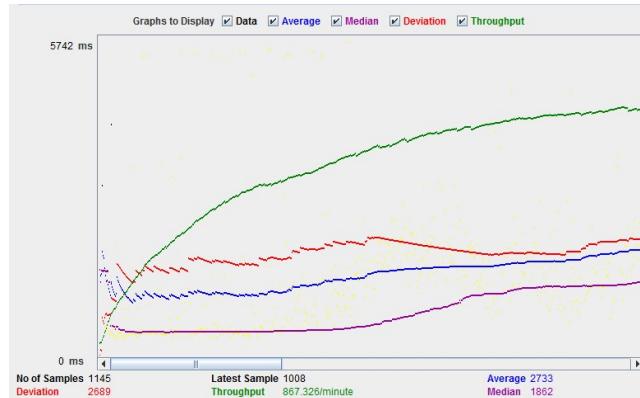
- **Throughput**
- **Deviation**

The **Throughput** is the most important parameter. It represents the ability of the server to handle a heavy load. The **higher** the Throughput is, the **better** is the server performance.

In this test, the throughput of Google server is 1,491.193/minute. It means Google server can handle 1,491.193 requests per minute. This value is quite high so we can conclude that Google server has good performance

The **deviation** is shown in red - it indicates the deviation from the average.
The **smaller the better**.

Let compare the performance of Google server to other web servers. This is the performance test result of website <http://www.yahoo.com/> (You can choose other websites)



The throughput of a website under test <http://www.yahoo.com> is 867.326/minutes. It means this server handle 867.326 requests per minute, lower than Google.

The deviation is 2689, much higher than Google (577). So we can determine the performance of this website is less than a Google server.

NOTE: The above values depend on several factors like current server load at Google, your internet speed, your CPU power etc. Hence, it's very unlikely that you will get the same results as above. So don't panic!

Troubleshooting:

If you face the issue while running the above scenario ... do the following

1. Check whether you are connecting to the internet via a proxy. If yes, remove the proxy.
2. Open a new instance of Jmeter
3. Open the [PerformanceTestPlan.jmx](#) in Jmeter
4. Double Click on Thread Group -> Graph Result
5. Run the Test

Observation	25	
Record	10	
Total	35	
Signature		

RESULT:

Thus the jmeter analyse the performance of web application was executed successfully.

Ex No.6.	Bugzilla
Date:	

AIM:

To create a Bug report in Bugzilla and prepare a Graphical reports.

ALGORITHM:

Step 1) To create a new bug in Bugzilla, visit the home-page of Bugzilla and click on **NEW** tab from the main menu

Step 2) In the next window

1. Enter Product
2. Enter Component
3. Give Component description
4. Select version
5. Select severity
6. Select Hardware
7. Select OS
8. Enter Summary
9. Enter Description
10. Attach Attachment
11. Submit

Step 3) Bug is created ID# 26320 is assigned to our Bug. You can also add additional information to the assigned bug like URL, keywords, whiteboard, tags, etc. This extra-information is helpful to give more detail about the Bug you have created.

Step 4) Stop

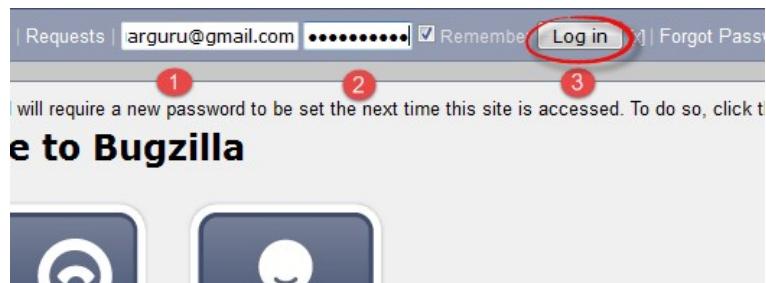
PROCEDURE:

Step 1) Use the following [link](#) for your handons. To create an account in Bugzilla tool or to login into the existing account go to **New Account** or **Log in** option in the main menu.



Step 2) Now, enter your personal details to log into Bugzilla

1. User ID
2. Password
3. And then click on "Log in"



Step 3) You are successfully logged into Bugzilla system



Creating a Bug-report in Bugzilla

Step 1) To create a new bug in Bugzilla, visit the home-page of Bugzilla and click on **NEW** tab from the main menu



Step 2) In the next window

1. Enter Product
2. Enter Component
3. Give Component description
4. Select version,
5. Select severity

6. Select Hardware
7. Select OS
8. Enter Summary
9. Enter Description
10. Attach Attachment
11. Submit

NOTE: The above fields will vary as per your customization of Bugzilla

The screenshot shows the Bugzilla bug creation form with the following field values:

- ① * Product: Sam's Widget
- ② Component: Widget Gears
- ③ Component Description: Gears for Sam's widgets
- ④ Version: unspecified
- ⑤ Severity: normal
- ⑥ Hardware: PC
- ⑦ OS: Windows NT
- ⑧ * Summary: Gears for sams widget twisted
- ⑨ Description: (empty)
- ⑩ Attachment: Add an attachment
- ⑪ Submit Bug

NOTE: The mandatory fields are marked with *.

In our case field's

- Summary
- Description

Are mandatory

If you do not fill them you will get a screen like below

The screenshot shows the Bugzilla bug creation form with validation errors:

- ① * Summary: Gears for sams widget twisted

You must enter a Summary for this bug.
- ② Description: The widget gears are twisted at the end and not showing correct signal

PUT your description overhere

Possible Duplicates:	Bug ID	Summary	Status	Action
	2776	when using the Widget Gears, the mV signal unexpectedly goes to 0	CONFIRMED	Add Me to the CC List
	2777	Widget Gears causes wrong mV signal to appear	CONFIRMED	Add Me to the CC List
	12431	Widget Gears cannot start	IN_PROGRESS	Add Me to the CC List
	12480	The Gear of sams widgets failed its validation	CONFIRMED	Add Me to the CC List
	15407	Sams Widget came pipe	CONFIRMED	Add Me to the CC List
	21019	Gears are bound up	CONFIRMED	Add Me to the CC List
	21841	Widget gears are stuck	CONFIRMED	Add Me to the CC List

Step 4) Bug is created ID# 26320 is assigned to our Bug. You can also add additional information to the assigned bug like URL, keywords, whiteboard, tags, etc. This extra-information is helpful to give more detail about the Bug you have created.

1. Large text box
2. URL
3. Whiteboard
4. Keywords
5. Tags
6. Depends on
7. Blocks
8. Attachments

Bug ID number is assigned to newly created bug

Bug 26320 - Gears for sams widget twisted (edit)

Status: CONFIRMED (edit)

Product: Sam's Widget

Component: Widget Gears

Version: unspecified

Hardware: PC

Importance: P2 - normal

Assigned To: cam.felt-williams (edit) (Take)

QA Contact: (edit) (Take)

① URL:

② Whiteboard:

③ Keywords:

④ Tags:

⑤ Depends on:

⑥ Blocks:

⑦ Show dependency tree / graph

Reported: 2015-01-07 02:50 PST by James

Modified: 2015-01-07 03:10 PST (history)

CC List: 1 user including you (edit)

See Also: (add)

Large text box:

A multiple-select box: Always Appears
Also Always Appears
Third Value: Always

Drop Down List: --

Date time:

Flags: None yet set (set flags)

Orig. Est.: Current Est.: Hours Worked: Hours Left: %Complete: Gain: Deadline:

0.0	0.0	0.0 + 0	0.0	0	0.0	2015-01-09
-----	-----	---------	-----	---	-----	------------

Summarize time (including time for bugs blocking this bug)

Attachments: Add an attachment (proposed patch, testcase, etc.)

Step 5) In the same window if you scroll down further. You can select deadline date and also status of the bug. **Deadline in Bugzilla usually gives the time-limit to resolve the bug in given time frame.**

Deadline:

January 2015

Su	Mo	Tu	We	Th	Fr	Sa
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

You can select deadline for your bug-report

Status: CONFIRMED (edit)

CONFIRMED
IN_PROGRESS
RESOLVED

James 2015-01-07 02:50:31 PST

Description [reply] [-]

Save Changes

Collapse All Comments
Expand All Comments

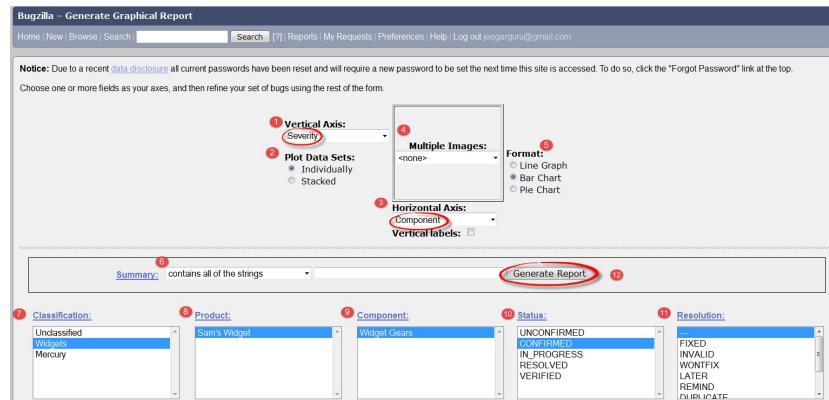
The widget gears are twisted at the end and not showing correct signal

Create Graphical Reports

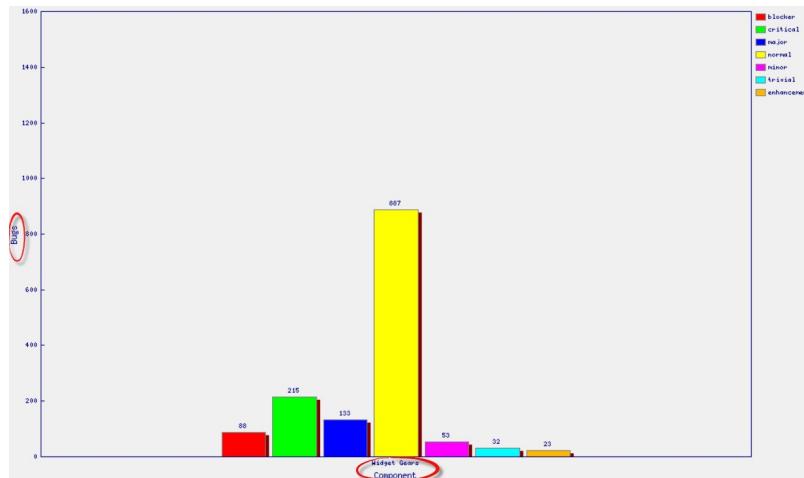
Graphical reports are one way to view the current state of the bug database. You can run reports either through an HTML table or graphical line/pie/bar-chart-based one. The idea behind

graphical report in Bugzilla is to define a set of bugs using the standard search interface and then choosing some aspect of that set to plot on the horizontal and vertical axes. You can also get a 3-dimensional report by choosing the option of "Multiple Pages".

Reports are helpful in many ways, for instance, if you want to know which component has the largest number of bad bugs reported against it. In order to represent that in the graph, you can select severity on X-axis and component on Y-axis, and then click on generate a report. It will generate a report with crucial information.



The graph below shows the Bar chart representation for the Bugs severity in component "**Widget Gears**". In the graph below, the most severe bug or blockers in components are 88 while bugs with normal severity are at the top with 667 number.



Likewise, we will also see the line graph for **%complete Vs Deadline**

Step 1) To view your report in a graphical presentation,

- Click on Report from Main Menu
- Click on the Graphical reports from the given option

Bugzilla – Reporting and Charting Kitchen

Home | New | Browse | Search | Search | [?] Reports | My Requests | Preferences | Help

Notice: Due to a recent [data disclosure](#) all current passwords have been reset and will require a new password to be set

Bugzilla allows you to view and track the state of the bug database in all manner of exciting ways.

Current State

- [Search](#) - list sets of bugs.
- [Tabular reports](#) - tables of bug counts in 1, 2 or 3 dimensions, as HTML or CSV.
- ② [Graphical reports](#) - line graphs, bar and pie charts.
- [Duplicates](#) - list of most frequently reported bugs.

Change Over Time

- [Old Charts](#) - plot the status and/or resolution of bugs against time, for each product in your database.
- [New Charts](#) - plot any arbitrary search against time. Far more powerful.

Step 2) Let's create a graph of % Complete Vs Deadline

In here on the vertical axis we chose **% Complete** and on our horizontal axis we chose **Deadline**. This will give the graph of amount of work done in percentage against the set-deadline.

Now, set various option to present reports graphically

1. Vertical Axis
2. Horizontal Axis
3. Multiple Images
4. Format- Line graph, Bar chart or Pie chart
5. Plot data set
6. Classify your bug
7. Classify your product
8. Classify your component
9. Classify bug status
10. Select resolution
11. Click on generate a report

Bugzilla – Generate Graphical Report

Home | New | Browse | Search | Search | [?] Reports | My Requests | Preferences | Help | Log out jeegarguru@gmail.com

Notice: Due to a recent [data disclosure](#) all current passwords have been reset and will require a new password to be set the next time this site is accessed. To do so, click the "Forgot Password" link at the top.

Choose one or more fields as your axes, and then refine your set of bugs using the rest of the form.

Vertical Axis: %Complete

Horizontal Axis: Deadline

Multiple Images: Severity

Format:

- Line Graph
- Bar Chart
- Pie Chart

Plot Data Sets:

- Individually
- Stacked

Vertical labels:

Summary: contains all of the strings Generate Report

Classification: ⑥

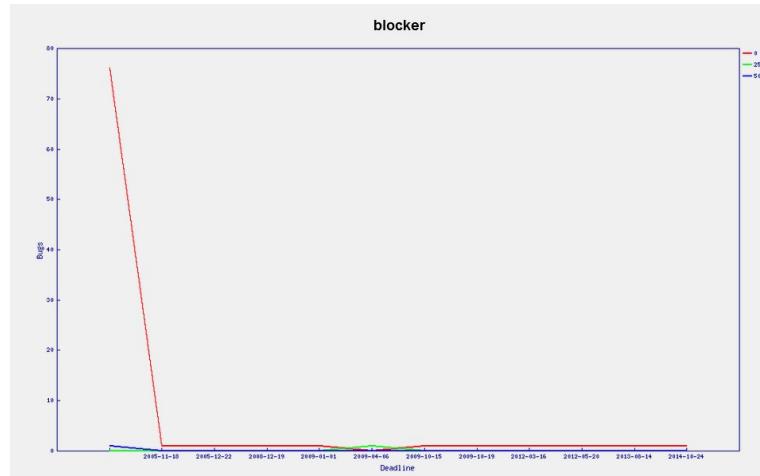
Product: ⑦

Component: ⑧

Status: ⑨

Resolution: ⑩

The image of the graph will appear somewhat like this



Browse Function

Step 1) To locate your bug we use browse function, click on **Browse** button from the main menu.



Step 2) As soon as you click on browse button a window will open saying "**Select a product category to browse**" as shown below, we browse the bug according to the category.

- After clicking the browse button
- Select the product "Sam's Widget" as such you have created a bug inside it

Bugzilla – Browse

Home | New **Browse** Search | Search 1

Notice: Due to a recent [data disclosure](#) all current passwords have been reset and will require a new password. To do so, click the "Forgot Password" link at the top.

Select a product category to browse:

Unclassified: Unassigned to any classifications

- [FoodReplicator](#): Software that controls a piece of hardware
- [Spider Sécretions](#): Spider secretions

Widgets: All widgets get classification of widget

2 [Sam's Widget](#) Special SAM widgets

Mercury: Because classifications do exist

- [MyOwnBadSelf](#): feh.
- [WorldControl](#): A small little program for controlling things from another place using the WorldControl API to extend control over other programs

Step 3) It opens another window, in this click on component "widget gears". Bugzilla Components are sub-sections of a product. For instance, where our product is **SAM'S WIDGET** whose component is **WIDGET GEARS**.

Bugzilla – Components for Sam's Widget

Home | New | Browse | Search | Search | [?] | Reports | My Requests |

Notice: Due to a recent [data disclosure](#) all current passwords have been reset and will require a new password. To do so, click the "Forgot Password" link at the top.

Select a component for the product that component: It'll take you to the bugs in this component. [Bug list created](#)

Components Default Assignee Default QA Contact

[Widget Gears](#) sam_folk-williams Gears for Sam's widgets

Step 4) when you click on the component, it will open another window. All the Bugs created under particular category will be listed over here. From that Bug-list, choose your Bug#ID to see more details about the bug.

Wed Jan 7 2015 20:59:10 PST						
Resolution: --- Component: Widget Gears Product: Sam's Widget						
This result was limited to 500 bugs. See all search results for this query.						
ID	Product	Comp	Assignee	Status	Resolution	Summary
1256	Sam's Wi	Widget G	justdave@syndicomm.com	CONF	---	summary
4219	Sam's Wi	Widget G	justdave@syndicomm.com	CONF	---	sdadasad
4742	Sam's	Click on Bug ID	stdave@syndicomm.com	CONF	---	just a test
5509	Sam's	number to see the details	stdave@syndicomm.com	CONF	---	Test Bug
2566	Sam's	details	ndfill@gavinsharp.com	CONF	---	test
6504	Sam's Wi	Widget G	mabst45@gmail.com	CONF	---	Won't run
3010	Sam's Wi	Widget G	mickesnow@yahoo.com.mx	CONF	---	buhigs
24741	Sam's Wi	Widget G	neha.malik028@gmail.com	CONF	---	cancel button not working
						Changed
						2014-10-23
						2012-06-13
						2009-01-26
						2014-05-04
						2009-11-04
						2010-09-21
						2012-03-15
						2014-10-16

It will open another window, where information about your bug can be seen more in detail. In the same window, you can also change the assignee, QA contact or CC list.

Bug 1256 - summary (edit)

Status: CONFIRMED ([edit](#))

Product: Sam's Widget

Component: Widget Gears

Version: unspecified

Hardware: HP Linux

Importance: P1 normal

Target Milestone: ---

Assigned To: [Dave Miller](#) ([take](#))

QA Contact: ([edit](#)) ([take](#))

URL: [\[redacted\]](#)

Whiteboard:

Reported: 2003-05-27 13:28 PDT by [Jason McCallum](#)

Modified: 2014-10-23 08:09 PDT ([History](#))

CC List: Add me to CC list
6 users ([edit](#))

See Also: ([add](#))

Large text box:

free text:

A multiple-select box: Always Appears
Also Always Appears
Third Value, Always

Save Changes

How to use Simple search option in Bugzilla

Bugzilla provides two ways of searching for bugs, they are **Simple Search** and **Advance Search** methods.

Step 1) We will first learn the "**Simple Search**" method. Click on search button from the main menu and then follow these steps

1. Click on "Simple Search" button
2. Choose the status of the Bug – choose Open if you are looking the bug in Open status and closed for bug in closed status
3. Choose your category and component, and you can also put keywords related to your bug
4. Click on the search

Bugzilla - Simple Search

Home | New | Browse | **Search** | ① | Search | [?] | Reports | My Requests | Preferences | Log out jeegarguru@gmail.com

Notice: Due to a recent [data disclosure](#) all current passwords have been reset and will require a new password to be set the next time this site is accessed. To do so, click the "Forgot Password" link at the top.

Simple Search ② | **Advanced Search** ③

Find a specific bug by entering words that describe it. Bugzilla will search bug descriptions and comments for those words and return a list of matching bugs sorted by relevance. All ④

Unclassified FoodReplicator Spider Sécrétions

For example, if the bug you are looking for is a crash when you go to a secure web site with an embedded Flash animation, you might search for "crash secure SSL flash".

Status: Open ⑤ | Closed | All

Product: Sam's Widget | Mercury | MyOwnBadSelf | WorldControl

Words: Widgets gears

Search ⑥

Step 2) Here we will search for both option **open** and **closed** status, first we have selected closed status for bug and clicked search button.

Simple Search

Find a specific bug by entering words that describe it. Bugzilla will search bug descriptions and comments for those words and return a list of matching bugs sorted by relevance.

For example, if the bug you are looking for is a crash when you go to a secure web site with an embedded Flash animation, you might search for "crash secure SSL flash".

Status: Closed ① | **Product:** All | **Words:** widget gears ②

Search ③

Choose the keywords for the bug you looking for

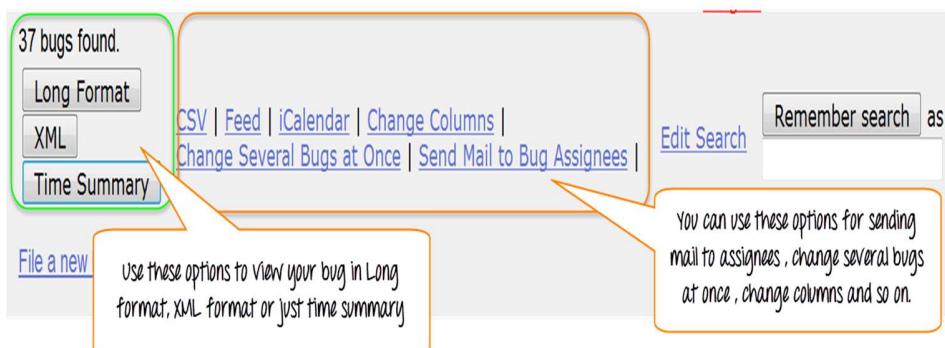
For closed status, it fetched 12 bugs.

12 bugs found.						
ID	Assignee	Resolution			Changed	
	Comp	Status	RESO	FIXE	Summary	
12412	Widget G sam.folkwilliams@gmail.com	RESO	FIXE	The widget stole my spacecraft!!	2011-08-27	
2998	WeatherC peter.rutherford@cnh.com	VERI	DUPL	Hardlinks not created and the world is thence seriously out of control	2010-11-23	
15235	WeatherC tara@bluemartini.com	RESO	DUPL	weather widget failed rain test	2011-06-14	
4297	Widget G justdave@syndicomm.com	RESO	FIXE	transation 01 - fails	2010-03-31	
6312	Widget G sam.folkwilliams@gmail.com	RESO	FIXE	lack of cogs in widget	2008-02-05	

Step 3) Likewise we have searched for Open status as well, and it has fetched 37 bugs related to our queries.

ID	Product	Comp	Assignee	Status
12431	Sam's Wi	Widget G	sam.folkwilliams@gmail.com	IN_P
12561	Sam's Wi	Widget G	sam.folkwilliams@gmail.com	CONF
7359	Sam's Wi	Widget G	raydevereaux@bc.com	IN_P
8907	Sam's Wi	Widget G	sam.folkwilliams@gmail.com	CONF

Also, at the bottom of the screen you have various options like how you want to see your bug - **an XML format, in Long format or just Time summary**. Apart from that you can also use other option like **send mail to bug assignee, change several bugs at once or change column of the screen**, etc.



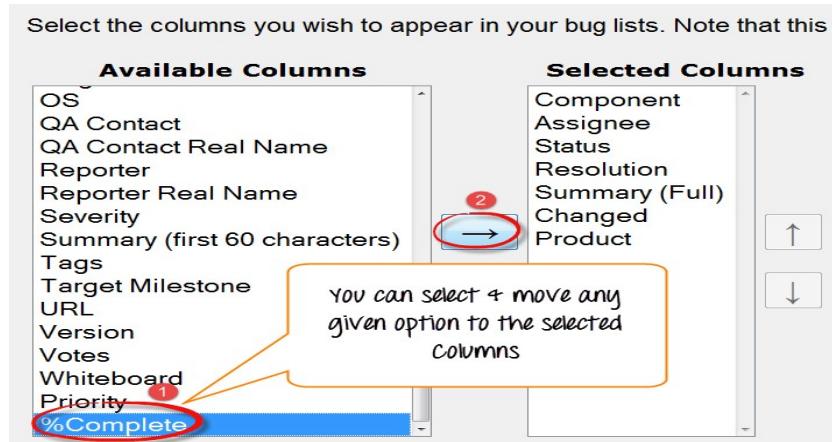
In next step, we will demonstrate one of this function **change column of the screen**, through which we will learn how to add or remove the column to the existing column.

How to add or remove a column to default search screen

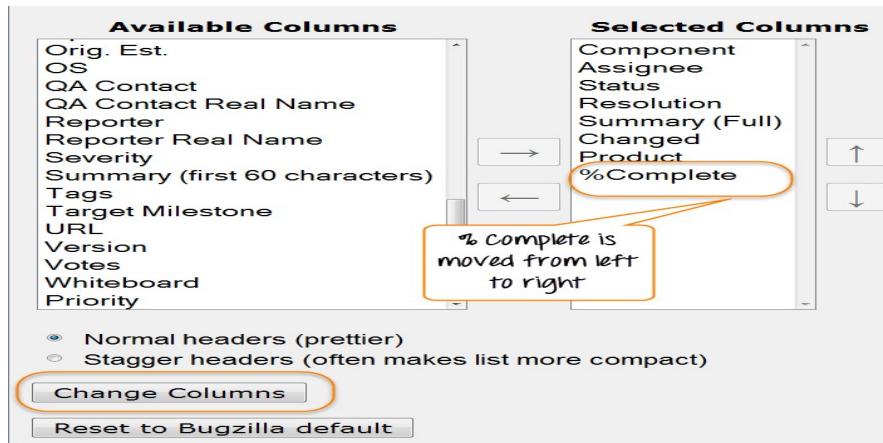
Step 1) Click on the **Change Column** as shown in above screen-shot. It will open a new window where you have to follow these steps.

- Select any given option from the column you want to appear in the main screen - here we have selected **% complete**
- Click on the **arrow button**, it will move % complete column from **Available Column** to the **Selected column**

These steps will move the selected column from left to right.



The % complete is moved from left to right as shown below, and once we click on **change column** it will appear in the main screen



Before- Search result screen before using "Change Column" option-

- There is no % complete column appears in search screen result as shown below

ID	Product	Comp	Assignee	Status	Resolution	Summary	Changed
12431	Sam's Wi	Widget G	sam.folkwilliams@gmail.com	IN_P	---	Widget Gears cannot start	2014-11-02
12561	Sam's Wi	Widget G	sam.folkwilliams@gmail.com	CONF	---	Gear is broke	2014-12-08
7359	Sam's Wi	Widget G	raydevereaux@bc.com	IN_P	---	funny	2011-05-01
8907	Sam's Wi	Widget G	sam.folkwilliams@gmail.com	CONF	---	Monkey Wren gears	2009-07-23

After- Search result screen after using "Change Column" option

- You can see **% complete** column added to the extreme right in the existing column in the main screen, which was not their previously.

37 bugs found.							
ID	Comp	Assignee	Status	Resolution	Summary	Changed	%Complete
12431	Widget G	sam.folkwilliams@gmail.com	IN_P	---	Widget Gears cannot start	2014-11-02	0 %
12561	Widget G	sam.folkwilliams@gmail.com	CONF	---	Gear is b	2014-12-08	0 %
7359	Widget G	raydevereaux@bc.com	IN_P	---	funny	2011-05-01	0 %
8907	Widget G	sam.folkwilliams@gmail.com	CONF	---	Monkey gears	2009-07-23	0 %

NOTE: Likewise you can remove or add any column you want.

How to use Advanced Search in Bugzilla

Step 1) After a Simple search we will look into the **Advanced Search** option for that you have to follow the following steps.

1. Click on the advanced search option
2. Select option for a summary, how you want to search
3. Enter the keyword for your bug- for example, **Widget gears twisted**
4. Select the category of your Bug under classification, here we selected Widget
5. Choose your product under which your Bug was created- Sam's Widget
6. Component- Widget gears
7. Status- Confirmed
8. Resolution

Simple Search

Advanced Search

contains all of the strings
contains any of the strings
contains the string
contains the string (exact case)
contains all of the words

② Summary:

Classification:
Unclassified
④ Widgets
Mercury

Product:
FoodReplicator
MyOwnBadSelf
⑤ Sam's Widget
WorldControl
Spider Sécretions

Component:
SpiceDispenser
Venom
VoiceInterface
WeatherControl
Web
⑥ Widget Gears

Status:
UNCONFIRMED
⑦ CONFIRMED
IN PROGRESS

Resolution:
⑧ FIXED
INVALID

Step 2) Once you select all the option, click on search button. It will detect the bug you created

The screenshot shows the Bugzilla search interface. At the top, there is a search bar with the text "Summary: contains all of the strings" followed by "widget gears twisted". A red circle highlights the "Search" button. Below the search bar, there are four filter sections: "Classification" (set to "Unclassified Widgets Mercury"), "Product" (set to "Sam's Widget"), "Component" (set to "Widget Gears"), "Status" (set to "UNCONFIRMED CONFIRMED"), and "Resolution" (set to "--- FIXED").

The advanced search will find your bug, and it will appear on the screen like this

The screenshot shows the search results for the query "widget gears twisted". The results are displayed in a table with columns: ID, Assignee, Resolution, Status, Summary, Changed, and %Complete. One result is shown:

ID	Assignee	Resolution	Status	Summary	Changed	%Complete
26320	Widget G sam.folkwilliams@gmail.com	CONF --- Gears for sams widget twisted	CONF ---	Gears for sams widget twisted	Wed 03:10	0 %

Below the table, it says "One bug found." and provides links for "Long Format", "XML", and "Time Summary". There is also a link to "CSV | Feed | iCalendar | Change Columns | Edit Search". An orange callout box points to the "Summary" column of the result table with the text "your bug appears on screen".

How to use preferences in BugZilla

Preferences in Bugzilla is used to customize the default setting made by Bugzilla as per our requirement. There are mainly five preferences available

- General Preferences
- E-mail Preferences
- Saved Searches
- Account Information
- Permissions

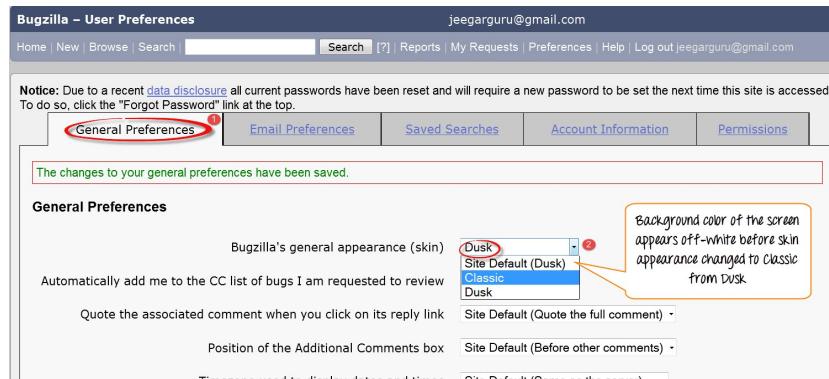
General Preferences

For **general preferences**, you have a various option like **changing Bugzilla general appearance, a position of the additional comment box, automatically add me to cc, etc.** Here we will see how to change the general appearance of the Bugzilla.

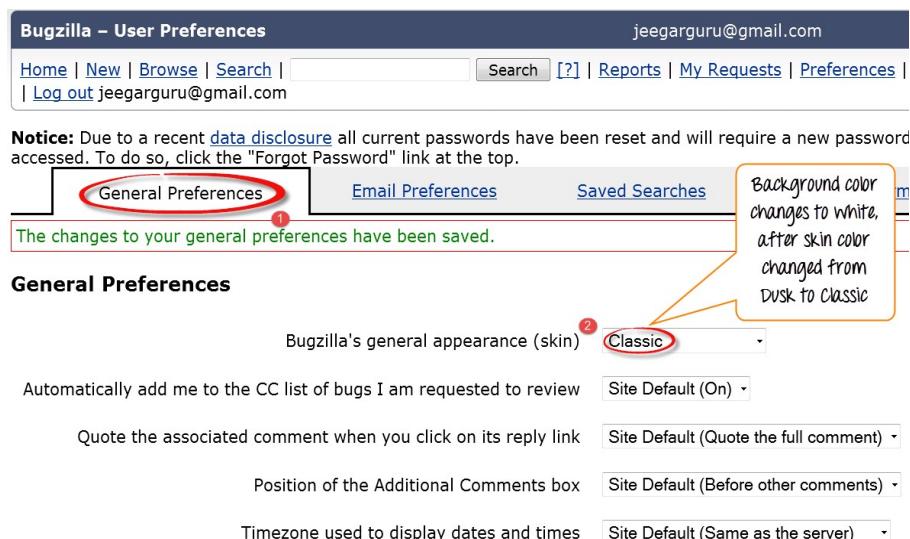
There are many changes you can do which are self-explanatory, and you can choose the option as per your requirement.

Step 1)

- To set the background Skin of Bugzilla
- Go to Bugzilla general preference (Skin)
- Select the option you want to see as a change and submit the change (Dusk □ Classic)
- A message will appear on the window saying changes have been saved, as soon as you submit the changes



After the skin preference is changed to Classic from Dusk, the background color of the screen appears white



Likewise, for other default settings changes can be done.

E-mail preferences

E-mail preferences enable you to decide how to receive the message and from whom to receive the messages.

Step 1) To set the e-mail preferences

1. Click on e-mail services

2. Enable or disable the mail to avoid receiving notification about changes to a bug
3. Receiving a mail when someone asks to set a flag or when someone sets a flag you asked for
4. When and from whom you want to receive mail and under which condition. After marking your option at the end, submit the changes.

Saved Searches Preference

Saved searches preference gives you the freedom to decide whether to share your bug or not to share.

Step 1) Click on saved searches, it will open window with the option like **edit bugs, don't share, can confirm, etc.** Choose the option as per your need.

Step 2) We can run our bug from "Saved Searches".

- Go to Saved Searches under preference
- Click on the "**Run**" button

General Preferences Email Preferences Saved Searches

Saved Searches

Your saved searches are as follows:

Search	Run	Edit	Footer	Share With a Group
My Bugs	Run	Edit	—	canconfirm ▾
Widget gears	Run	Edit	Forget	<input checked="" type="checkbox"/>

As soon as you run your search from Saved Searches it opens your bug as shown below

Resolution: --- Assignee: jeegarguru@gmail.com Reporter: jeegarguru@gmail.com

ID	Product	Comp	Assignee	Status	Resolution	Summary	Changed
26330	Sam's Wi	Widget G	sam.folkwilliams@gmail.com	CONF	---	Widget gears is not responsive	Wed 22:06
26320	Sam's Wi	Widget G	sam.folkwilliams@gmail.com	CONF	---	Gears for sams widget twisted	Wed 03:10

2 bugs found.

Long Format CSV | Feed | iCalendar | Change Columns | Change Several Bugs at Once | Time Summary

Edit Search Remember search as

Step 3) In the same window we can also choose specific users with whom we want to share the search by marking or unmarking the checkbox against the users

Search	Shared By	Shared To	Run	Edit	Show in Footer
123	Amrita <osmosys.dici@gmail.com>	editbugs	Run	Edit	X <input type="checkbox"/>
All Mercury	Ben Schultz (SoftTechnics) <ben.schultz@softtechnics.com>	editbugs	Run	Edit	✓ <input checked="" type="checkbox"/>
ami	Ami <ami_nahmani@walla.com>	editbugs	Run	Edit	✓ <input checked="" type="checkbox"/>
Bug1	Aruna <aruna_trimurti@yahoo.com>	editbugs	Run	Edit	X <input type="checkbox"/>

Home | New | Browse | Search | Both these users can edit our bug

My Bugs | Widget gears | All Mercury | ami | 26320 | sams widget

Observation	25	
Record	10	
Total	35	
Signature		

RESULT

Thus the Bug report in Bugzilla is created and Graphical reports are prepared.

Ex No. 7
Date:

Create a Simple TestNG Project

Aim:

To create a simple TestNG project

Algorithm:

Step 1: Right-click the src folder and select New > Other..

Step 2: Select TestNG class and then click Next:

Step 3: Click Browse... to select a source folder, select MyFirstTestNGProject > src, and click OK.

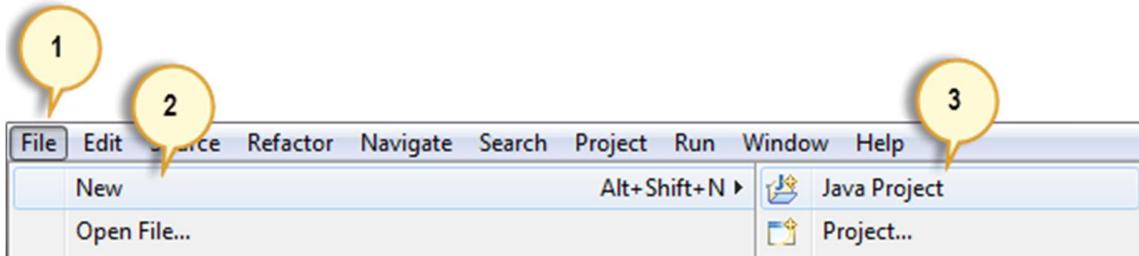
Step 4: Enter the following information: Package name: myfirsttestngpackage.

Step 5: Click Finish.

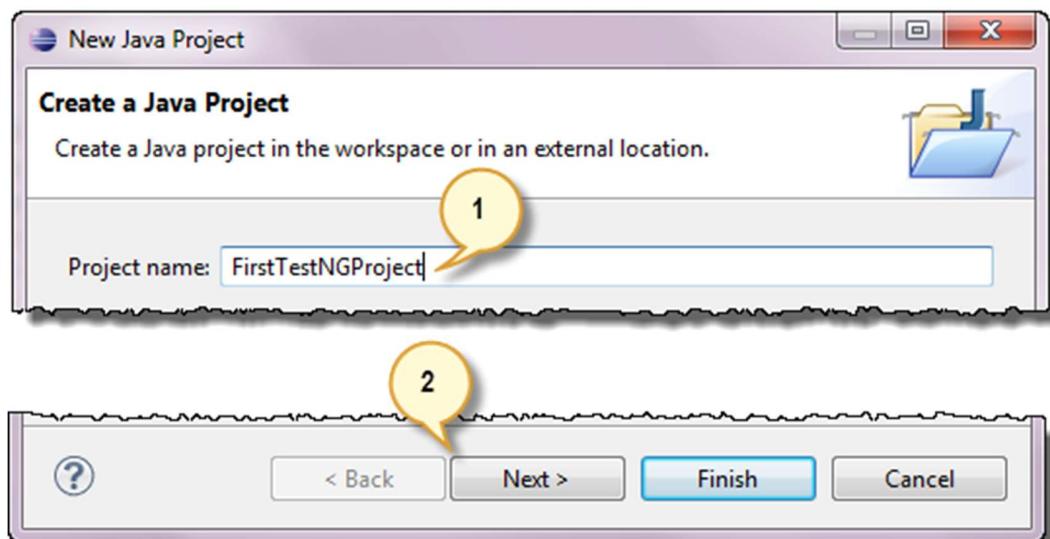
Procedure:

Setting up a new TestNG Project

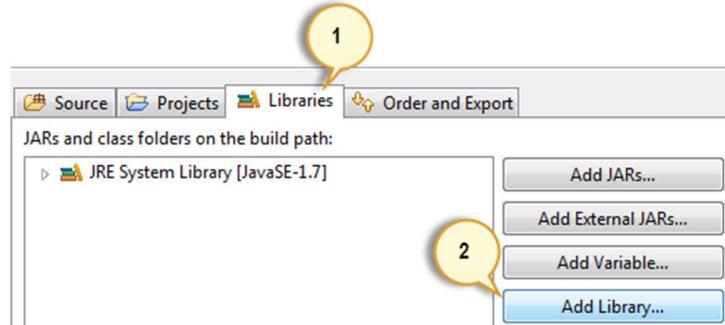
Step 1: Click File > New > Java Project



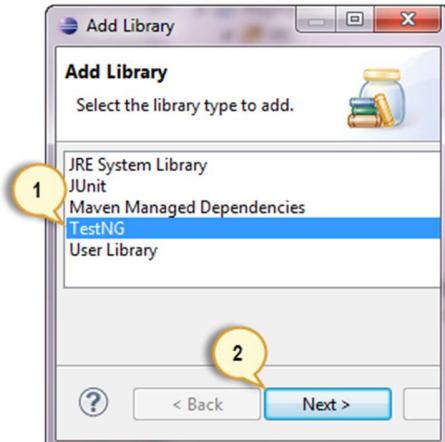
Step 2: Type "FirstTestNGProject" as the Project Name then click Next.



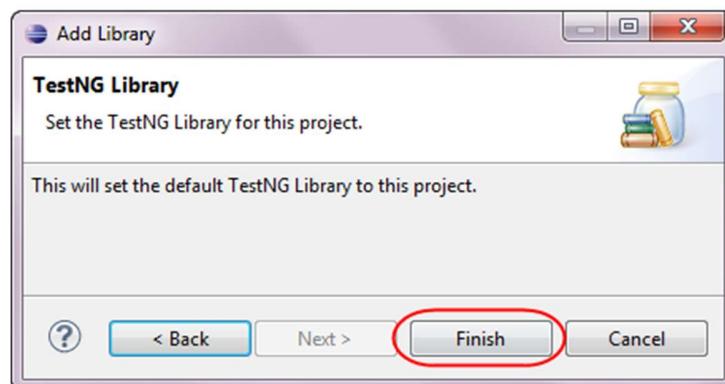
Step 3: We will now start to import the TestNG Libraries onto our project. Click on the "Libraries" tab, and then "Add Library..."



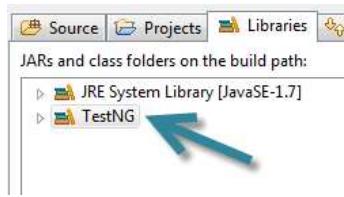
Step 4: On the Add Library dialog, choose "TestNG" and click Next.



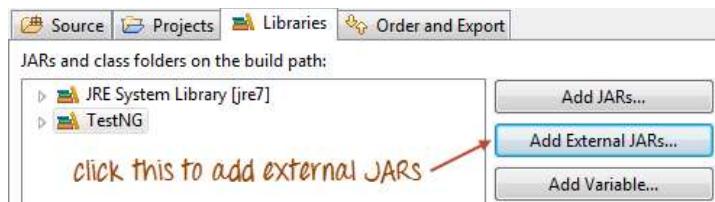
Step 5: Click Finish.



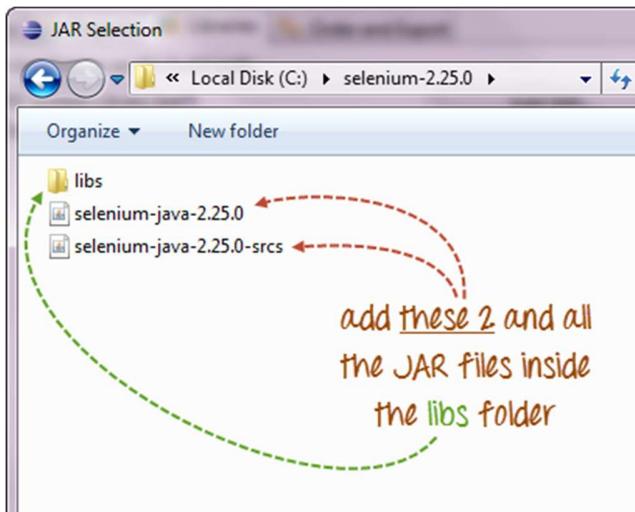
You should notice that TestNG is included on the Libraries list.



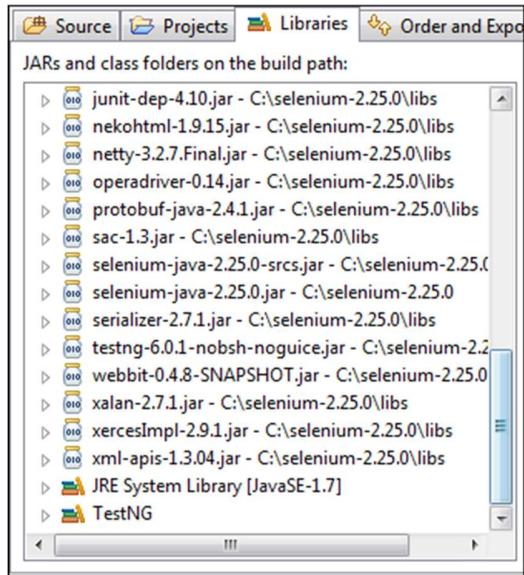
Step 6: We will now add the JAR files that contain the Selenium API. These files are found in the Java client driver that we downloaded from <http://docs.seleniumhq.org/download/> when we were installing Selenium and Eclipse in the previous chapters.



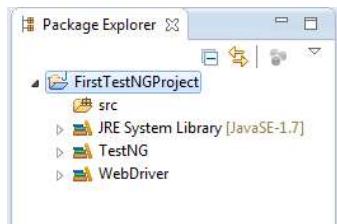
Then, navigate to where you have placed the Selenium JAR files.



After adding the external JARs, your screen should look like this.



Step 7: Click Finish and verify that our FirstTestNGProject is visible on Eclipse's Package Explorer window.



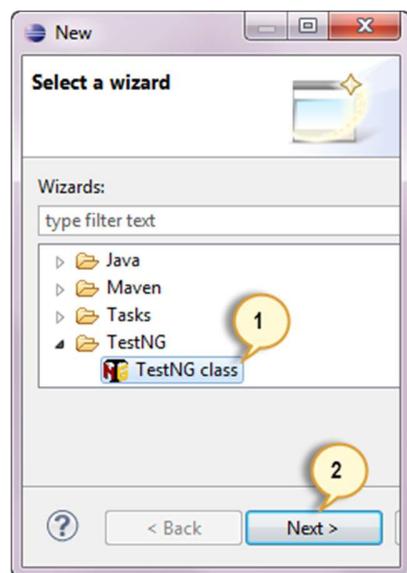
Creating a New TestNG Test File

Now that we are done setting up our project, let us create a new TestNG file.

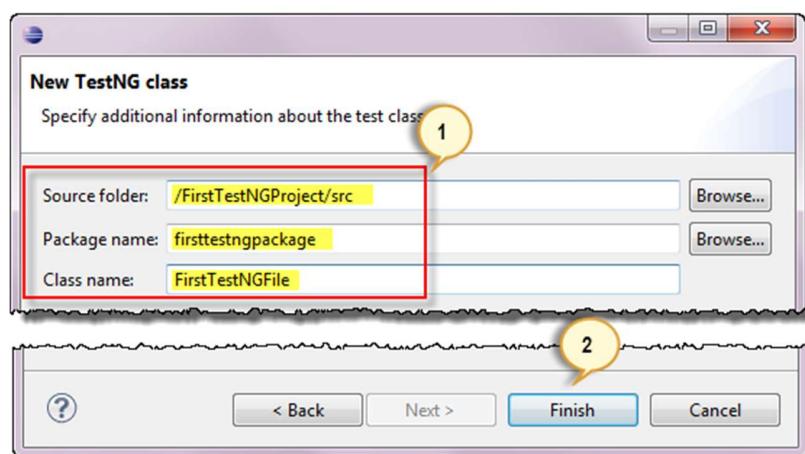
Step 1: Right-click on the "src" package folder then choose New >Other...



Step 2: Click on the TestNG folder and select the "TestNG class" option. Click Next.



Step 3: Type the values indicated below on the appropriate input boxes and click Finish. Notice that we have named our Java file as "FirstTestNGFile".



Eclipse should automatically create the template for our TestNG file shown below.

```
FirstTestNGFile.java
FirstTestNGProject src firsttestng
1 package firsttestngpackage;
2
3 import org.testng.annotations.Test;
4
5 public class FirstTestNGFile {
6     @Test
7     public void f() {
8     }
9 }
10
```


Coding Our First Test Case

Let us now create our first [Test Case](#) that will check if Mercury Tours' homepage is correct. Type your code as shown below.

```
package firsttestngpackage;
import org.openqa.selenium.*;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.Assert;
import org.testng.annotations.*;

public class firsttestngfile {
    public String baseUrl = "http://demo.guru99.com/test/newtours/";
    String driverPath = "C:\\geckodriver.exe";
    public WebDriver driver ;

    @Test
    public void verifyHomepageTitle() {

        System.out.println("launching firefox browser");
        System.setProperty("webdriver.firefox.marionette", driverPath);
        driver = new FirefoxDriver();
        driver.get(baseUrl);
        String expectedTitle = "Welcome: Mercury Tours";
        String actualTitle = driver.getTitle();
        Assert.assertEquals(actualTitle, expectedTitle);
        driver.close();
    }
}
```

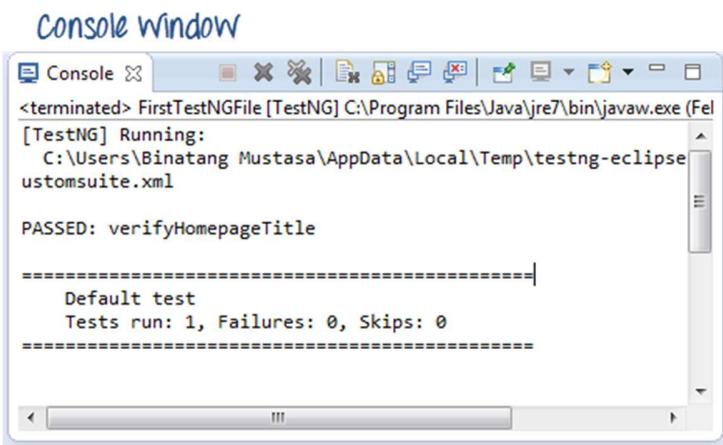
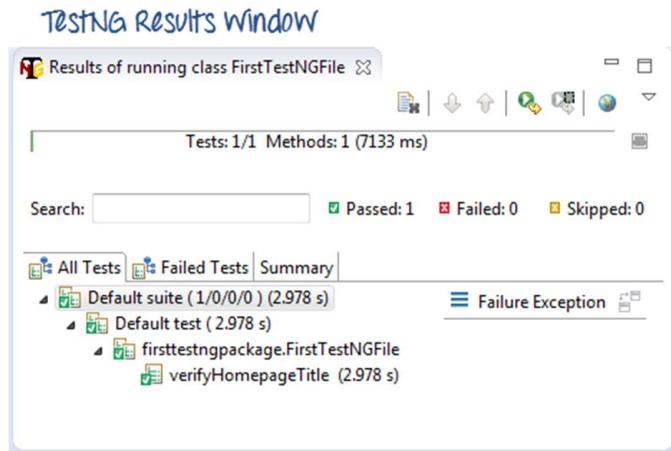
Notice the following.

- TestNG does not require you to have a main() method.
- Methods need not be static.
- We used the @Test annotation. **@Test is used to tell that the method under it is a test case.** In this case, we have set the verifyHomepageTitle() method to be our test case, so we placed an '@Test' annotation above it.
- Since we use annotations in TestNG, we needed to import the package org.testng.annotations.*.
- We used the Assert class. **The Assert class is used to conduct verification operations in TestNG.** To use it, we need to import the org.testng.Assert package.

You may have multiple test cases (therefore, multiple @Test annotations) in a single TestNG file. This will be tackled in more detail later in the section "Annotations used in TestNG."

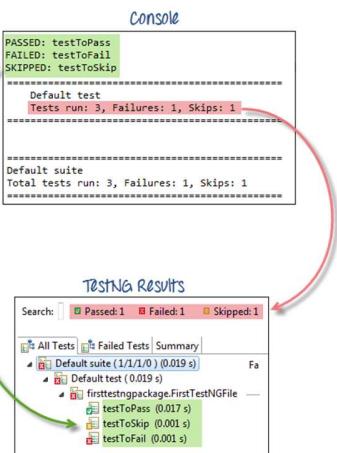
Running the Test

To run the test, simply run the file in Eclipse as you normally do. Eclipse will provide two outputs – one in the Console window and the other on the TestNG Results window.



Checking reports created by TestNG

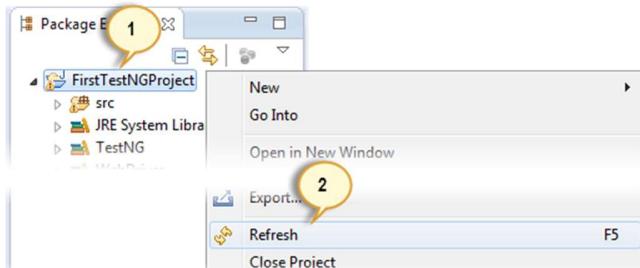
The Console window in Eclipse gives a text-based report of our test case results while the TestNG Results window gives us a graphical one.



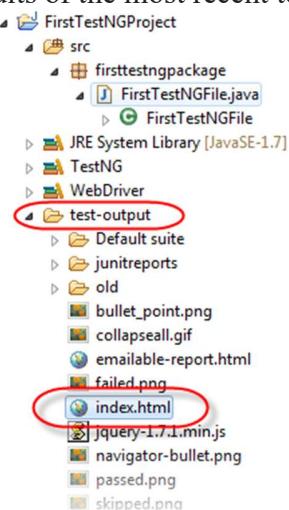
Generating HTML Reports

TestNG has the ability to generate reports in HTML format.

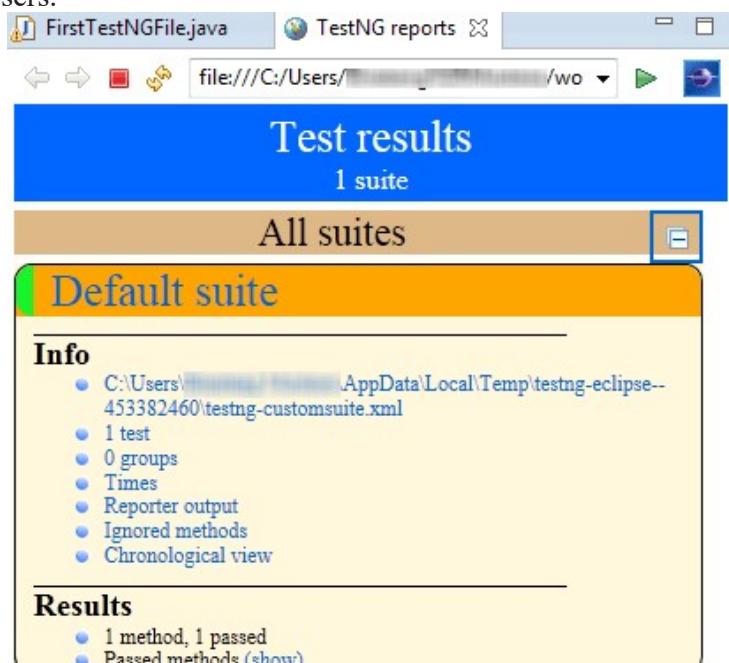
Step 1: After running our FirstTestNGFile that we created in the previous section, right-click the project name (FirstTestNGProject) in the Project Explorer window then click on the "Refresh" option.



Step 2: Notice that a "test-output" folder was created. Expand it and look for an index.html file. This HTML file is a report of the results of the most recent test run.



Step 3: Double-click on that index.html file to open it within Eclipse's built-in web browser. You can refresh this page any time after you rerun your test by simply pressing F5 just like in ordinary web browsers.



Annotations used in TestNG

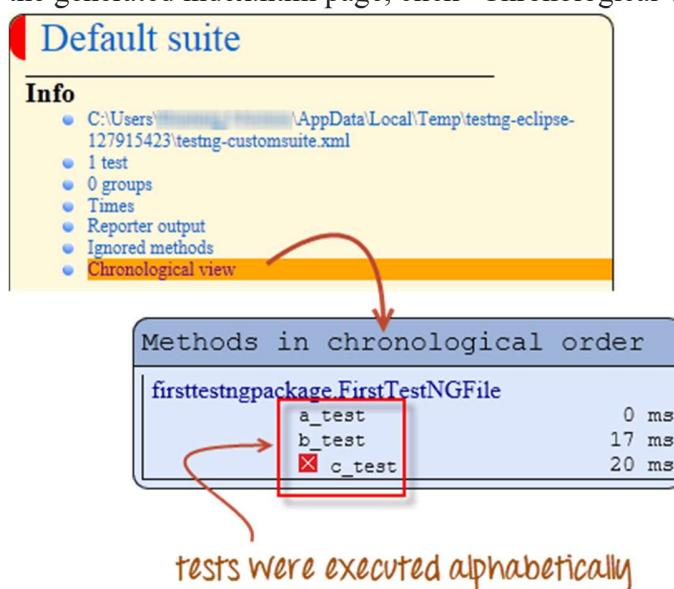
In the previous section, you have been introduced to the @Test annotation. Now, we shall be studying more advanced annotations and their usages.

Multiple Test Cases

We can use multiple @Test annotations in a single TestNG file. By default, methods annotated by @Test are executed alphabetically. See the code below. Though the methods c_test, a_test, and b_test are not arranged alphabetically in the code, they will be executed as such.

```
public class FirstTestNGfile {  
    @Test  
    public void c_test() {  
        Assert.fail();  
    }  
  
    @Test  
    public void a_test() {  
        Assert.assertTrue(true);  
    }  
  
    @Test  
    public void b_test() {  
        throw new SkipException("Skipping b_test...");  
    }  
}
```

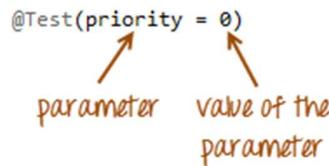
Run this code and on the generated index.html page, click "Chronological view."



Parameters

If you want the methods to be executed in a different order, use the parameter "priority". **Parameters are keywords that modify the annotation's function.**

- Parameters require you to assign a value to them. You do this by placing a "=" next to them, and then followed by the value.
- Parameters are enclosed in a pair of parentheses which are placed right after the annotation like the code snippet shown below.



TestNG will execute the @Test annotation with the lowest priority value up to the largest. There is no need for your priority values to be consecutive.

```
public class FirstTestNGfile {
    @Test(priority = 3) ← the 2nd least priority value so
    public void c_test() { this will be executed 2nd
        Assert.fail();
    }

    @Test(priority = 0) ← this has the lowest priority value
    public void a_test() { so this will be executed first
        Assert.assertTrue(true);
    }

    @Test(priority = 7) ← largest priority value so this will
    public void b_test() { be executed last
        throw new SkipException("Skipping b_test...");
    }
}
```

The TestNG HTML report will confirm that the methods were executed based on the ascending value of priority.

Methods in chronological order		
firsttestngpackage.FirstTestNGfile		
a_test		0 ms
✗ c_test		18 ms
b_test		23 ms

Multiple Parameters

Aside from "priority," @Test has another parameter called "alwaysRun" which can only be set to either "true" or "false." **To use two or more parameters in a single annotation, separate them with a comma** such as the one shown below.

```
@Test(priority = 0, alwaysRun = true)
```



Consider the code below.

```
packagefirsttestngpackage;
importorg.openqa.selenium.*;
importorg.openqa.selenium.firefox.FirefoxDriver;
```



```

import org.testng.Assert;
import org.testng.annotations.*;
public class firsttestngfile {
    public String baseUrl = "http://demo.guru99.com/test/newtours/";
    String driverPath = "C:\\geckodriver.exe";
    public WebDriver driver;

    @BeforeTest
    public void launchBrowser() {
        System.out.println("launching firefox browser");
        System.setProperty("webdriver.firefox.marionette", driverPath);
        driver = new FirefoxDriver();
        driver.get(baseUrl);
    }
    @Test
    public void verifyHomepageTitle() {
        String expectedTitle = "Welcome: Mercury Tours";
        String actualTitle = driver.getTitle();
        Assert.assertEquals(actualTitle, expectedTitle);
    }
    @AfterTest
    public void terminateBrowser(){
        driver.close();
    }
}

```

Applying the logic presented by the table and the code above, we can predict that the sequence by which methods will be executed is:

- 1st - launchBrowser()
- 2nd - verifyHomepageTitle()
- 3rd - terminateBrowser()

The placement of the annotation blocks can be interchanged without affecting the chronological order by which they will be executed. For example, try to rearrange the annotation blocks such that your code would look similar to the one below.

```

package firsttestngpackage;
import org.openqa.selenium.*;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.Assert;
import org.testng.annotations.*;
public class firsttestngfile {
    public String baseUrl = "http://demo.guru99.com/test/newtours/";
    String driverPath = "C:\\geckodriver.exe";
}

```



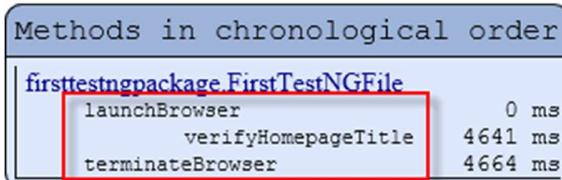
```

public WebDriver driver ;
@AfterTest           //Jumbled
public void terminateBrowser(){
driver.close();
}
@BeforeTest           //Jumbled
public void launchBrowser() {
System.out.println("launching firefox browser");
System.setProperty("webdriver.firefox.marionette", driverPath);
driver = new FirefoxDriver();
driver.get(baseUrl);
}
@Test                //Jumbled
public void verifyHomepageTitle() {
String expectedTitle = "Welcome: Mercury Tours";
String actualTitle = driver.getTitle();
Assert.assertEquals(actualTitle, expectedTitle);
}

}

```

Run the code above and notice that



@BeforeMethod and @AfterMethod

@BeforeMethod methods under this annotation will be executed **prior to each method in each test case**

@AfterMethod methods under this annotation will be executed **after each method in each test case**.

In Mercury Tours, suppose we like to verify the titles of the target pages of the two links below.



The flow of our test would be:

- Go to the homepage and verify its title.
- Click REGISTER and verify the title of its target page.
- Go back to the homepage and verify if it still has the correct title.
- Click SUPPORT and verify the title of its target page.

- Go back to the homepage and verify if it still has the correct title.

The code below illustrates how `@BeforeMethod` and `@AfterMethod` are used to efficiently execute the scenario mentioned above.

```
package firsttestngpackage;
import org.openqa.selenium.*;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.Assert;
import org.testng.annotations.*;
@Test
public class firsttestngfile {
    public String baseUrl = "http://demo.guru99.com/test/newtours/";
    String driverPath = "C:\\geckodriver.exe";
    public WebDriver driver;
    public String expected = null;
    public String actual = null;
    @BeforeTest
    public void launchBrowser() {
        System.out.println("launching firefox browser");
        System.setProperty("webdriver.firefox.marionette", driverPath);
        driver= new FirefoxDriver();
        driver.get(baseUrl);
    }
    @BeforeMethod
    public void verifyHomepageTitle() {
        String expectedTitle = "Welcome: Mercury Tours";
        String actualTitle = driver.getTitle();
        Assert.assertEquals(actualTitle, expectedTitle);
    }
    @Test(priority = 0)
    public void register(){
        driver.findElement(By.linkText("REGISTER")).click() ;
        expected = "Register: Mercury Tours";
        actual = driver.getTitle();
        Assert.assertEquals(actual, expected);
    }
    @Test(priority = 1)
    public void support() {
        driver.findElement(By.linkText("SUPPORT")).click() ;
        expected = "Under Construction: Mercury Tours";
        actual = driver.getTitle();
    }
}
```



```

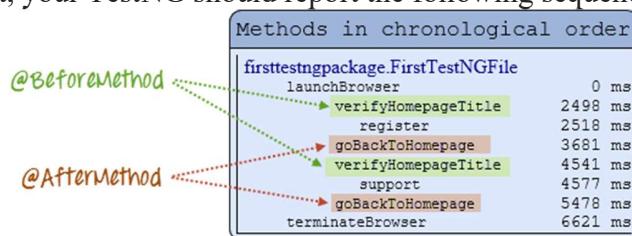
        Assert.assertEquals(actual, expected);
    }

    @AfterMethod
    public void goBackToHomepage() {
        driver.findElement(By.linkText("Home")).click();
    }

    @AfterTest
    public void terminateBrowser() {
        driver.close();
    }
}

```

After executing this test, your TestNG should report the following sequence.



Simply put, **@BeforeMethod** should contain methods that you need to run **before** each test case while **@AfterMethod** should contain methods that you need to run **after** each test case.

Summary of TestNG Annotations

@BeforeSuite: The annotated method will be run before all tests in this suite have run.

@AfterSuite: The annotated method will be run after all tests in this suite have run.

@BeforeTest: The annotated method will be run before any test method belonging to the classes inside the tag is run.

@AfterTest: The annotated method will be run after all the test methods belonging to the classes inside the tag have run.

@BeforeGroups: The list of groups that this configuration method will run before. This method is guaranteed to run shortly before the first test method that belongs to any of these groups is invoked.

@AfterGroups: The list of groups that this configuration method will run after. This method is guaranteed to run shortly after the last test method that belongs to any of these groups is invoked.

@BeforeClass: The annotated method will be run before the first test method in the current class is invoked.

@AfterClass: The annotated method will be run after all the test methods in the current class have been run.

@BeforeMethod: The annotated method will be run before each test method.

@AfterMethod: The annotated method will be run after each test method.

@Test: The annotated method is a part of a test case

Conclusion

- TestNG is a testing framework that is capable of making Selenium tests easier to understand and of generating reports that are easy to understand.
- The main advantages of TestNG over JUnit are the following.
 - Annotations are easier to use and understand.

- Test cases can be grouped more easily.
 - TestNG allows us to create parallel tests.
- The Console window in Eclipse generates a text-based result while the TestNG window is more useful because it gives us a graphical output of the test result plus other meaningful details such as:
 - Runtimes of each method.
 - The chronological order by which methods were executed
- TestNG is capable of generating HTML-based reports.
- Annotations can use parameters just like the usual Java meth

Observation	25	
Record	10	
Total	35	
Signature		

Result:

Thus create a simple TestNG project was created successfully.