



**REVA**  
**UNIVERSITY**

Bengaluru, India

SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY

Software Testing Lab

B20EJ0503

For

Fifth Semester

B. Tech in Computing and Information Technology

NAME	
SRN	
SEM	
SECTION	
BRANCH	
ACADEMIC YEAR	

**LAB QUESTIONS**

<b>SL NO</b>	<b>Lab Programs</b>	<b>PAGE NO</b>
1	Write programs in "C" language to demonstrate the working of the following: a. Constructs: i) do...while ii) while....do iii) if...else iv) switch v) for	
2	A program written in "C" language for Matrix Multiplication fails or Introspect the causes for its failure and write down the possible reasons for its failure.	
3	Take any system (e.g. ATM System) and study its system specifications and report the various bugs	
4	Take any system (e.g. ATM System) and study its system specifications and report the various bugs.	
5	Create a test plan document for any application (eg. Library Management System)	
6	Study of any testing tool (eg., Win runner)	
7	Study of any web testing tool(eg., Selenium)	
8	Study of any bug tracking tool (eg., Bugzilla, bugbit)	
9	Study of any test management tool (eg., Test Director)	
10	Study of any open source-testing tool(eg., Test Link)	

**School Vision:**

To produce excellent quality technologists and researchers of global standards in computing and Information technology who have potential to contribute to the development of the nation and the society with their expertise, skills, innovative problem-solving abilities, strong moral and ethical values.

**School Mission:**

1. To create state of the art computing labs infrastructure and research facilities in information technology.
2. To provide student-centric learning environment in Computing and Information technology through innovative pedagogy and education reforms.
3. To encourage research, innovation and entrepreneurship in computing and information technology through industry/academia collaborations and extension activities
4. Organize programs through club activities for knowledge enhancement in thrust areas of information technology.
5. To enhance leadership qualities among the youth and enrich personality traits, promote patriotism, moral and ethical values.

**6. Instruction Details**

Course Duration: 26-Practical sessions

**Introduction**

This course examines fundamental software testing and related program analysis techniques. In particular, the important phases of testing will be reviewed, emphasizing the significance of each phase when testing different types of software. The course will also include concepts such as test case generation, test coverage, regression testing, program analysis (e.g., program-flow and data-flow analysis), and test prioritization.

**Objectives:** The objectives of this course are to:

1. Discuss fundamental concepts in software testing.
2. Illustrate the use of different software testing methods.
3. Demonstrate the use of Selenium IDE to develop applications.
4. Discuss the use of locators in developing real world applications.

# **SOFTWARE TESTING**

## **EXPERIMENT: 1**

**NAME OF THE EXPERIMENT:** Write program in ‘C’ language to demonstrate the working of the following constructs

i.)**do..while:**

**Syntax:**

Iteration-statement:

do statement while ( expression ) ;

Example:

```
#include<stdio.h>
#include<conio.h>
#include <stdio.h>
```

```
main()
```

```
{
```

```
    nt i = 10;
```

```
    do{
        printf("Hello %d\n", i );
        i = i -1;
        if( i == 6 )
        {
            break;
        }
    }while ( i > 0 );
}
```

ii.) do..while

```
while(condition)
{
    Loop body
    Increment or decrement;
}
```

Example:

```
#include<stdio.h>

int main()

{
    int counter, howmuch;

    scanf("%d", &howmuch);
    counter = 0;
    while ( counter < howmuch)
    {
        counter++;
        printf("%d\n", counter);
    }
    return 0;
}
```

iii) if...else

syntax:

```
if( condition 1 )
    statement1;
else if( condition 2 )
    statement2;
else if( condition 3 )
    statement3;
else
    statement4;
```

example:

```

#include<stdio.h>
int main(){
    int x,y;
    printf("Enter value for x :");
    scanf("%d",&x);
    printf("Enter value for y :");
    scanf("%d",&y);
    if ( x > y ){
        printf("X is large number - %d\n",x);
    }
    else{
        printf("Y is large number - %d\n",y);
    }
    return 0;
}

```

#### iv)switch

syntax:

```

switch(int/char const)
{
    Case const 1:stmt1;
    Break;
    Case const 2:stmt2;
    Break;
}
default: stmt n;
Break;
}

```

Example:

```

#include <stdio.h>

int main() {
    int color = 1;
    printf("Please choose a color(1: red,2: green,3: blue):\n");
    scanf("%d", &color);

    switch (color)
    {

```

```

case 1:
    printf("you chose red color\n");
    break;
case 2:
    printf("you chose green color\n");
    break;
case 3:
    printf("you chose blue color\n");
    break;
default:
    printf("you did not choose any color\n");
}
return 0;
}

```

#### v) for      Syntax:

```

for (initialization; condition; increment/decrement)
{
    //body of the loop
}

```

Example:

```

#include <stdio.h>

int main()
{
    int x;
    /* The loop goes while x < 10, and x increases by one every loop*/ for ( x = 0; x
    < 10; x++ )
    {
        /* Keep in mind that the loop condition checks the
        conditional statement before it loops again. consequently,
        when x equals 10 the loop breaks. x is updated before the
        condition is checked. */

        printf( "%d\n", x );
    }
    getchar();
}

```

**Viva questions:**

1. How to find entered number is EVEN or ODD without using conditional statement (not using if.. else, while, do... while...., for....)
2. Write a function to swap any two numbers?
3. How can we find out prime numbers from 1 to 50?

## **EXPERIMENT: 2**

### **NAME OF THE EXPERIMENT:**

Write a C program that uses functions to perform the following:

- ii) Multiplication of Two Matrices\*/**

```
#include<stdio.h>
#include<conio.h>
Void main()
{
    int a[10][10], b[10][10], m[10][10], i, j, p, q, r, s, k; clrscr ();
    printf("Enter the size of A Matrix");
    scanf("%d %d", &p, &q);
    printf("Enter the size of B Matrix");
    scanf("%d %d", &r, &s);
    if(q==r)
    {
        printf("Enter the elements of matrix A:");
        for(i=0; i<p; i++)
        {
            for(j=0; j<q; j++)
            {
                scanf("%d", &a[i][j]);
            }
        }
        printf("Enter the elements of matrix b:");
        for(i=0; i<r; i++)
        {
            for(j=0; j<s; j++)
            {
                scanf("%d", &b[i][j]);
            }
        }
        for (i=0; i<p; i++)
        {
            for(j=0; j<s; j++)
            {
                m[i][j]=0;
                for(k=0; k<q; k++)
                {
                    m[i][j] =m[i][j] +a[i][j] * b[k][j];
                }
            }
        }
        printf("matrix multiplication is:");
        for(i=0; i<p; i++)
        {
            for(j=0; j<s; j++)
            {
                printf("%d\t", m[i][j]);
            }
        }
    }
}
```

```
    ` printf("\n");
}
}

else
printf("matrix multiplication is not possible");
getch();
}
```

### **FAILURE CASES:**

#### **output:**

1. Enter the size of a: 2 3

Enter the size of b: 2 3

Matrix multiplication is not possible.

**Reason to fail:** to do multiplication of matrices the number of columns in matrix —a[] should be equal to number of rows in matrix —bl.

2. Enter the size of a: p q

Enter the size of b: q s

Matrix multiplication is not possible.

**Reason to fail:** to do multiplication of matrices the number of columns in matrix —al should be equal to number of rows in matrix —bl, and rows & columns should be integer values.

3. Enter the size of a: 1.5 2

Enter the size of b: 2 3

Matrix multiplication is not possible.

**Reason to fail:** to do multiplication of matrices the number of columns in matrix —al should be equal to number of rows in matrix —bl, and rows & columns should be integer values.

4. Enter the size of a: 350 480 Enter

the size of b: 480 620

Matrix multiplication is not possible.

**Reason to fail:** size of buffer will be not be sufficient to handle this multiplication.

5. Enter the size of a: -1 -2

Enter the size of b: -2 3

Matrix multiplication is not possible.

**Reason to fail:** to do multiplication of matrices the number of columns in matrix —al should be equal to number of rows in matrix —bl, and rows & columns should be positive integer values.

#### **viva questions:**

1. syntax for multiplication
2. syntax for matrix multiplication
- 3.what the logic for matrix multiplication?

### **EXPERIMENT: 3**

**NAME OF THE EXPERIMENT:** ATM system specifications and report the various bugs

#### **Purpose:**

This document describes the software requirements and specification (SRS) for an automated teller machine (ATM) network. The document is intended for the customer and the developer (designers, testers, maintainers). The reader is assumed to have basic knowledge of banking accounts and account services. Knowledge and understanding of Unified Modeling Language (UML) diagrams is also required.

#### **Scope:**

The software supports a computerized banking network called Bank24. The network enables customers to complete simple bank account services via automated teller machines (ATMs) that may be located off premise and that need not be owned and operated by the customer's bank. The ATM identifies a customer by a cash card and password. It collects information about a simple account transaction (e.g., deposit, withdrawal, transfer, bill payment), communicates the transaction information to the customer's bank, and dispenses cash to the customer. The banks provide their own software for their own computers. The Bank24 software requires appropriate record keeping and security provisions. The software must handle concurrent accesses to the same account correctly.

#### **Intended Audience:**

The intended audience of this SRS consists of:

- Software designers
- Systems engineers
- Software developers
- Software testers
- Customers

The actors of the system are:

1. User
2. ATM Machine
3. Bank

#### **Product Perspective:**

An automated teller machine (ATM) is a computerized telecommunications device that provides the customers of a financial institution with access to financial transactions in a public space without the need for a human clerk or bank teller. On most modern ATMs, the customer is identified by inserting a plastic ATM card with a magnetic stripe or a plastic smartcard with a chip, that contains a unique card number and some security information, such as an expiration date or CVC (CVV). Security is provided by the customer entering a personal identification number (PIN).

#### **Product functions:**

Using an ATM, customers can access their bank accounts in order to make cash withdrawals (or credit card cash advances) and check their account balances.

The functions of the system are:

1. Login
2. Get Balance Information
3. Withdraw Cash
4. Transfer Funds

#### **Operating Environments:**

The hardware, software and technology used should have following specifications:

- Ability to read the ATM card.
- Ability to count the currency notes.
- Touch screen for convenience.
- Keypad(in case touchpad fails)
- Continuous power supply.

- Ability to connect to bank's network.
- Ability to validate user.

### **Design/implementation constraints:**

Login:

Validate Bank Card

- Validate for Card Expiration Date
- Validate that the card's expiration date is later than today's date
- If card is expired, prompt error message "Card is expired"

Validate for Stolen or Lost Card

- Validate that the card is not reported lost or stolen
- If card is lost, prompt error message, "Card has been reported lost"
- If card is stolen, prompt error message, "Card has been reported stolen"

Validate for Disabled Card

- Validate that the card is not disabled
- If card is disabled, prompt error message, "Card has been disabled as of <expiration date>"

Validate for Locked Account

- Validate that the account is not locked
- If account is locked, prompt error message "Account is locked"

Validate PIN

- Validate that the password is not blank
- If PIN is blank, prompt error message "Please provide PIN"
- Validate that the password entered matches the password on file
- If password does not match, prompt error message "Password is Incorrect"

Lock Account

- If number of consecutive unsuccessful logins exceeds three attempts, lock account

Maintain Consecutive Unsuccessful Login Counter

Increment Login Counter

For every consecutive Login attempt, increment logic counter by 1.

Reset login counter to 0 after login is successful.

Get Balance Information

Withdraw Cash

Transfer Funds

### **Assumptions and Dependencies:**

- Hardware never fails
- ATM casing is impenetrable
- Limited number of transactions per day (sufficient paper for receipts)
- Limited amount of money withdrawn per day (sufficient money)

### **External Interface Requirements**

#### **User interfaces**

The customer user interface should be intuitive, such that 99.9% of all new ATM users are able to

complete their banking transactions without any assistance.

### Hardware interfaces

The hardware should have following specifications:

- Ability to read the ATM card
- Ability to count the currency notes
- Touch screen for convenience
- Keypad (in case touchpad fails)
- Continuous power supply
- Ability to connect to bank's network
- Ability to take input from user
- Ability to validate user

### Software interfaces

The software interfaces are specific to the target banking software systems. At present, two known banking systems will participate in the ATM network.

- State Bank
- Indian Overseas Bank

### Safety requirements:

Must be safe kept in physical aspects, say in a cabin

- Must be bolted to floor to prevent any kind of theft
- Must have an emergency phone outside the cabin
- There must be an emergency phone just outside the cabin
- The cabin door must have an ATM card swipe slot
- The cabin door will always be locked, which will open only when user swipes his/her ATM card in the slot & is validated as genuine

### Security requirements:

- Users accessibility is censured in all the ways
- Users are advised to change their PIN on first use
- Users are advised not to tell their PIN to anyone
- The maximum number of attempts to enter PIN will be three

Some of the possible Bugs on ATM machine?

1. Successful insertion of ATM card
2. Unsuccessful operation due to insert card in wrong angle
3. Unsuccessful operation due to invalid account Ex: other bank card or time expired card
4. successful entry of PIN number
5. un successful operation due to enter wrong PIN number 3times
6. successful selection of language
7. successful selection of account type
8. unsuccessful operation due to invalid account type
9. successful selection of withdraw operation
10. successful selection of amount to be withdrawl
11. successful withdraw operation

12. unsuccessful withdraw operation due to wrong denominations
13. unsuccessful withdraw operation due to amount is greater than day limit
14. unsuccessful withdraw operation due to lack of money in ATM
  
15. unsuccessful withdraw operation due to amount is greater than possible balance
16. unsuccessful withdraw operation due to transactions is greater than day limit
17. unsuccessful withdraw operation due to click cancel after insert card
18. unsuccessful withdraw operation due to click cancel after insert card & pin number
19. unsuccessful withdraw operation due to click cancel after insert card , pin number & language
20. unsuccessful withdraw operation due to click cancel after insert card , pin number , language & account type
21. unsuccessful withdrawal operation due to click cancel after insert card , pin number , language ,account type & withdrawal operation
22. unsuccessful withdrawal operation due to click cancel after insert card , pin number , language ,account type ,withdrawal operation &amount to be withdraw

## **EXPERIMENT: 4**

### **NAME OF THE EXPERIMENT: Test cases for banking applications**

Banking applications are considered to be one of the most complex applications in today's software development and testing industry. **What makes Banking application so complex?** What approach should be followed in order to test the complex workflows involved? In this article we will be highlighting different stages and techniques involved in testing Banking applications. **The characteristics of a Banking application are as follows:**

- Multi tier functionality to support thousands of concurrent user sessions
- Large scale Integration , typically a banking application integrates with numerous other applications such as Bill Pay utility and Trading accounts
- Complex Business workflows
- Real Time and Batch processing
- High rate of Transactions per seconds
- Secure Transactions
- Robust Reporting section to keep track of day to day transactions
- Strong Auditing to troubleshoot customer issues
- Massive storage system
- Disaster Management.

The above listed ten points are the **most important characteristics of a Banking application.**

Banking applications have multiple tiers involved in performing an operation. For Example, a **banking application may have:**

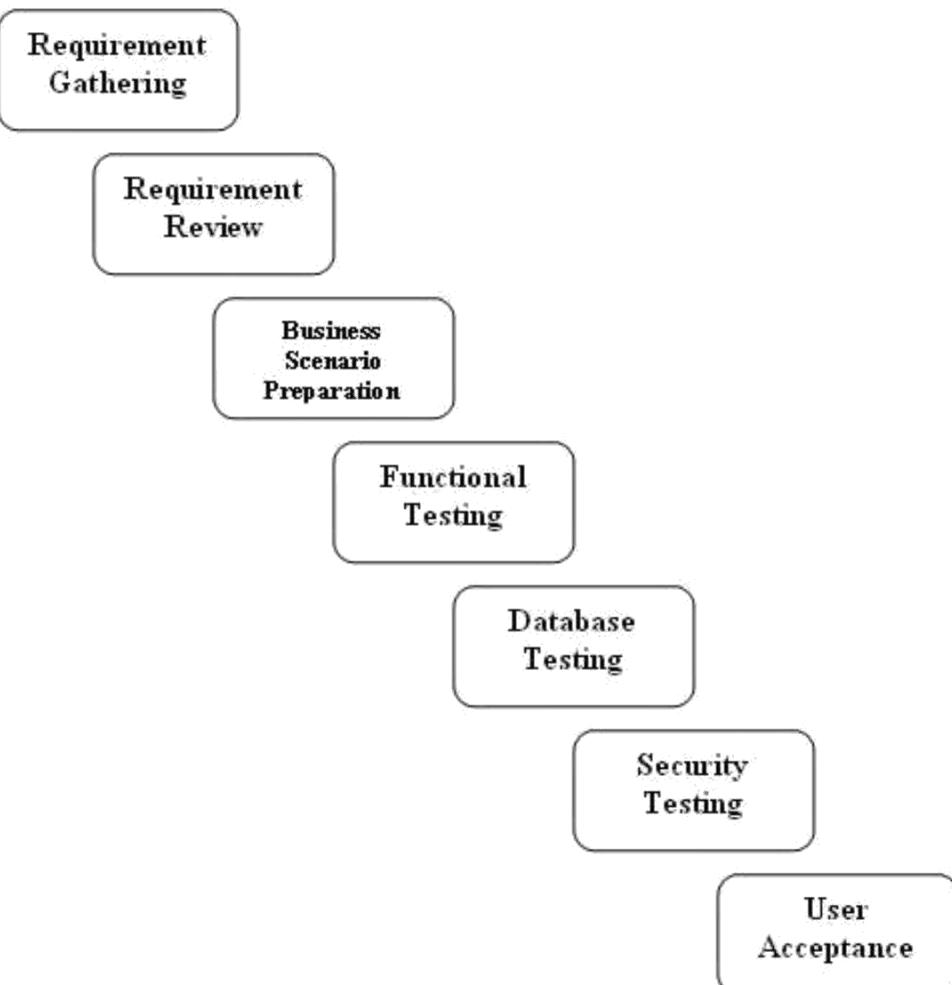
1. Web Server to interact with end users via Browser
2. Middle Tier to validate the input and output for web server
3. Data Base to store data and procedures
4. Transaction Processor which could be a large capacity Mainframe or any other Legacy system to carry out Trillions of transactions per second.

If we talk about testing banking applications it requires an **end to end testing methodology involving multiple software testing techniques to ensure:**

- Total coverage of all banking workflows and Business Requirements
- Functional aspect of the application
- Security aspect of the application
- Data Integrity
- Concurrency
- User Experience

**Typical stages involved in testing Banking Applications** are shown in below workflow which we will be discussing individually.

## Banking Application Testing Workflow



### 1) Requirement Gathering:

Requirement gathering phase involves documentation of requirements either as Functional Specifications or Use Cases. Requirements are gathered as per customer needs and documented by Banking Experts or Business Analyst. To write requirements on more than one subject experts are involved as banking itself has multiple sub domains and one full fledge banking application will be the integration of all. For Example: A banking application may have separate modules for Transfers, Credit Cards, Reports, Loan Accounts, Bill Payments, Trading Etc.

### 2) Requirement Review:

The deliverable of Requirement Gathering is reviewed by all the stakeholders such as QA Engineers, Development leads and Peer Business Analysts. They cross check that neither existing business workflows nor new workflows are violated.

### 3) Business Scenario Preparations:

In this stage QA Engineers derive Business Scenarios from the requirement documents (Functions Specs or Use Cases); Business Scenarios are derived in such a way that all Business Requirements are covered. Business Scenarios are high level scenarios without any detailed steps, further these Business Scenarios are reviewed by Business Analyst to ensure all of Business Requirements are met and its easier for BAs to review high level scenarios than reviewing low level detailed Test Cases.

#### 4) Functional Testing:

In this stage functional testing is performed and the usual software testing activities are performed such as:

##### **Test Case Preparation:**

In this stage Test Cases are derived from Business Scenarios, one Business Scenario leads to several positive test cases and negative test cases. Generally tools used during this stage are Microsoft Excel, Test Director or Quality Center.

##### **Test Case Review:**

Reviews by peer QA Engineers

##### **Test Case Execution:**

Test Case Execution could be either manual or automatic involving tools like QC, QTP or any other.

#### 5) Database Testing:

Banking Application involves complex transaction which are performed both at UI level and Database level, Therefore Database testing is as important as functional testing. Database in itself is an entirely separate layer hence it is carried out by database specialists and it uses techniques like

- Data loading
- Database Migration
- Testing DB Schema and Data types
- Rules Testing
- Testing Stored Procedures and Functions
- Testing Triggers
- Data Integrity

#### 6) Security Testing:

Security Testing is usually the last stage in the testing cycle as completing functional and non functional are entry criteria to commence Security testing. Security testing is one of the major stages in the entire Application testing cycle as this stage ensures that application complies with Federal and Industry standards. Security testing cycle makes sure the application does not have any web vulnerability which may expose sensitive data to an intruder or an attacker and complies with standards like OWASP.

In this stage the major task involves in the whole application scan which is carried out using tools like IBM Appscan or HP WebInspect (2 Most popular tools).

Once the Scan is complete the Scan Report is published out of which False Positives are filtered out and rest of the vulnerability are reported to Development team for fixing depending on the Severity.

Other **Manual tools for Security Testing** used are: Paros Proxy, Http Watch, Burp Suite, Fortify tools Etc.

Apart from the above stages there might be different stages involved like Integration Testing and Performance Testing.

In today's scenario **majority of Banking Projects are using**: Agile/Scrum, RUP and Continuous Integration methodologies, and Tools packages like Microsoft's VSTS and Rational Tools.

As we mentioned RUP above, RUP stands for Rational Unified Process, which is an iterative software development methodology introduced by IBM which comprises of four phases in which development and testing activities are carried out.

**Four phases are:**

- i) Inception
- ii) Collaboration
- iii) Construction and
- iv) Transition

RUP widely involves IBM Rational tools.

In this article we discussed **how complex a Banking application could be** and what are the **typical phases involved in testing the application**. Apart from that we also discussed current trends followed by IT industries including software development methodologies and tools.

Test cases for opening bank account

**1. Input parameters checking -**

Name

-Date of Birth -

Photo -Address

Proof -Identity

proof

-Introducers (if applicable) -

PAN card

-Initial deposit

-Whether checkbook / ATM card / Online banking facilities are needed or not -Customer signature

**Type of account -**

Savings account -Salary

account -Joint account -

Current account -

Secondary account -RD

account

-Account for a company

**Test cases**

-Checking mandatory input parameters -Checking optional input parameters -Check whether able to create account entity.

-Check whether you are able to deposit an amount in the newly created account (and thus updating the balance)

-Check whether you are able to withdraw an amount in the newly created account (after deposit) (and thus updating the balance)

-Check whether company name and its pan number and other details are provided in case of salary account

-Check whether primary account number is provided in case of secondary account

-Check whether company details are provided in cases of company's current account

-Check whether proofs for joint account is provided in case of joint account

-Check whether you are able deposit an account in the name of either of the person in an joint

account.

-Check whether you are able withdraw an account in the name of either of the person in an joint account.

-Check whether you are able to maintain zero balance in salary account

-Check whether you are not able to maintain zero balance (or mini balance) in non-salary account.

### **viva questions**

1. Can you explain boundary value analysis?
2. Can you explain equivalence partitioning?
3. Can you explain random/monkey testing?
4. What are semi-random test cases?
5. What is negative and positive testing?
6. How did you define severity ratings in your project?

## **EXPERIMENT: 5**

### **NAME OF THE EXPERIMENT: Test plan document for library application**

The Library Management System is an online application for assisting a librarian in managing a book library in a University. The system would provide basic set of features to add/update clients, add/update books, search for books, and manage check-in / checkout processes. Our test group tested the system based on the requirement specification.

## **INTRODUCTION**

This test report is the result for testing in the LMS. It mainly focuses on two problems: what we will test and how we will test.

### **Result**

#### **GUI test**

Pass criteria: librarians could use this GUI to interface with the backend library database without any difficulties

Result: pass

#### **Database test**

Pass criteria: Results of all basic and advanced operations are normal (refer to section 4)

Result: pass

#### **Basic function test**

##### **Add a student**

Pass criteria:

- Each customer/student should have following attributes: Student ID/SSN (unique), Name, Address and Phone number.  
Result: pass
- The retrieved customer information by viewing customer detail should contain the four attributes.  
Result: pass

##### **Update/delete student**

Pass criteria:

- The record would be selected using the student ID Result:  
pass
- Updates can be made on full. Items only: Name, Address, Phone number Result: pass
- The record can be deleted if there are no books issued by user.  
Result: Partially pass. When no books issued by user, he can be deleted. But when there are books Issued by this user, he was also deleted. It is wrong.
- The updated values would be reflected if the same customer's ID/SSN is called for.  
Result: pass
- If customer were deleted, it would not appear in further search queries.  
Result: pass

##### **Add a book**

Pass criteria:

- Each book shall have following attributes: Call Number, ISBN, Title, Author name.  
Result: pass
- The retrieved book information should contain the four attributes.  
Result: pass

##### **Update/delete book**

Pass criteria:

- The book item can be retrieved using the call number Result: did not pass. Can not retrive using the call number
- The data items which can be updated are: ISBN, Title, Author name Result: pass
- The book can be deleted only if no user has issued it.

Result: partially pass. When no user has issued it, pass. When there are user having issued it, did not pass

- The updated values would be reflected if the same call number is called for Result: pass
- If book were deleted, it would not appear in further search queries. Result: pass

### **Search for book**

Pass criteria:

- The product shall let Librarian query books' detail information by their ISBN number or Author or Title.

Result: pass

- The search results would produce a list of books, which match the search parameters with following Details: Call number, ISBN number, Title, Author Result: pass

- The display would also provide the number of copies which is available for issue Result: pass

- The display shall provide a means to select one or more rows to a user-list Result: pass

- A detailed view of each book should provide information about check-in/check out status, with the borrower's information.

Result: pass

- The search display will be restricted to 20 results per page and there would be means to navigate from sets of search results.

Result: pass

- The user can perform multiple searches before finally selecting a set of books for check in or checkout. These should be stored across searches.

Result: pass

- A book may have more than one copy. But every copy with the same ISBN number should have same detail information.

Result: pass

- The borrower's list should agree with the data in students' account

Result: pass

### **Check-in book**

Pass criteria:

- Librarians can check in a book using its call number

Result: pass

- The check-in can be initiated from a previous search operation where user has selected a set of books.

Result: pass

- The return date would automatically reflect the current system date. Result: did not pass.

- Any late fees would be computed as difference between due date and return date at rate of 10 cents a day.

Result: did not pass

- A book, which has been checked in once, should not be checked in again Result: pass

### **Check-out book**

Pass criteria:

- Librarians can check out a book using its call number Result: pass

- The checkout can be initiated from a previous search operation where user has selected a set of books.

Result: pass

- The student ID who is issuing the book would be entered

Result: pass

- The issue date would automatically reflect the current system date. Result: did not pass

- The due date would automatically be stamped as 5 days from current date. Result: did not pass
  
  
  
  
  
- A book, which has been checked out once, should not be checked out again Result: pass
- A student who has books due should not be allowed to check out any books Result: did not pass
- The max. No of books that can be issued to a customer would be 10. The system should not allow checkout of books beyond this limit.  
Result: pass

### **View book detail**

**Pass criteria:**

- This view would display details about a selected book from search operation Result: pass
- The details to be displayed are: Call number, ISBN, Title, Author, Issue status (In library or checked out), If book is checked out it would display, User ID & Name, Checkout date, Due date  
Result: for checkout date and due date, did not pass
- Books checked in should not display user summary  
Result: pass
- Books checked out should display correct user details. Result:  
pass

### **View student detail**

**Pass criteria:**

- Librarians can select a user record for detailed view Result:  
pass
- The detail view should show:
  - a. User name, ID, Address & Phone number Result:  
pass
  - b. The books issued by user with issue date, due date, call number, title Result: did not pass
  - c. Late fees & Fines summary and total  
Result: did not pass
- The display should match existing user profile Result:  
pass
- The books checked out should have their statuses marked Result: pass
- The book search query should show the user id correctly. Result: pass

### **Network test**

**Pass criteria:** Results of operations (ping, ftp and ODBC connectivity check) are normal

Result: did not test this item, because no enough machines and no available environment.

### **Viva questions**

1. How to create a test plan document for Library Management System?
2. What is object repository
3. How many test cases can you write                          1) File - open dialog box in notepad please write 5 if software failed in customer environment what we called a) error b) fault c) defect d) failure
4. What test plan should contain?
5. What is test strategy?
6. Define test Plan? What is the difference between Master Test Plan and Test Plan?

## **EXPERIMENT: 6**

**NAME OF THE EXPERIMENT:** Study of Any Testing Tool( WinRunner)

WinRunner is a program that is responsible for the automated testing of software. WinRunner is a Mercury Interactive's enterprise functional testing tool for Microsoft windows applications.

### **Testing an Application using WinRunner**

After installing the WinRunner on your computer, invoke the WinRunner application:

Start -> Programs ->WinRunner ->WinRunner

The opening screen of the WinRunner application is displayed, prompting you to select one of the three options:

New Test: To create a new test script  
Open Test: To open an existing test script  
Quick Preview: To view the quick preview of WinRunner



### **Recording Test Cases**

There are two modes of recording:

Context Sensitive mode: This mode of recording is used when the location of the GUI controls (i.e. X and Y coordinates) or the mouse positions are not necessary.  
Analog mode: This mode of recording is used when the mouse positions, the location of the controls in the application, also play an important role in testing the application. This mode of recording has to be used to validate bitmaps, testing the signature etc.

The procedure for recording a test case is as follows:

Step 1: Open a new document: File -> New (or) Select "New Test" from the WinRunner's Welcome screen.

Step 2: Open (run) the application to be tested.

Step 3: Start recording a test case. Create ->Record - Context Sensitive (or) click on the toolbar's "Record" button once, to record in Context Sensitive mode.

Step 4: Select the application to be tested by clicking on the application's title bar.

Step 5: Perform all the actions to be recorded.

Step 6: Once all required actions are recorded, stop the recording. Create -> Stop (or) Click on the toolbar's "Stop" button to stop the recording. WinRunner generates the script for the recorded actions.

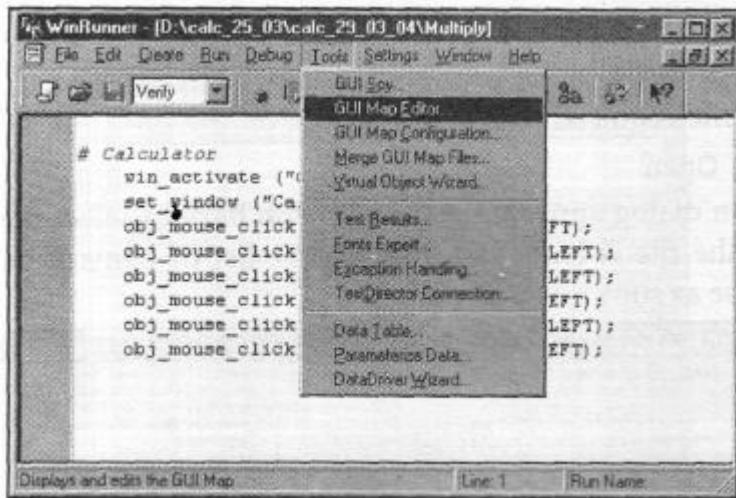
There are two modes for generating the test cases: "Global GUI map file mode" and "GUI map file per test mode". By default, it is in "Global GUI map file mode".

In Global GUI map file mode, you have to explicitly save the information learnt by WinRunner. WinRunner saves it in a file with extension "gui". When you have to run a test, you need to load the corresponding GUI map file; otherwise it will not be able to recognize the objects in the test case and displays an error message

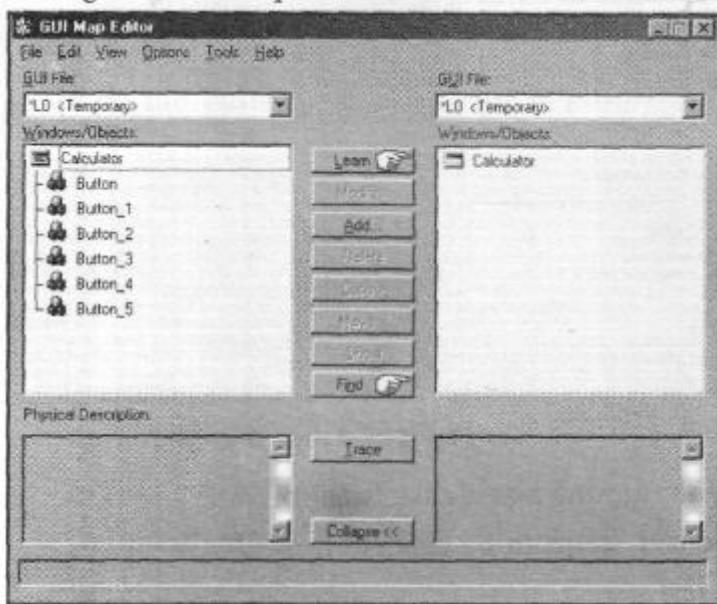
The procedure for saving the GUI map file in Global GUI map file mode is as follows:

Step 1: Record a test case by following the preceding procedure.

Step 2: Open the GUI Map Editor window as shown in Fig. Tools -> GUI Map Editor



Step 3: On selecting the GUI Map Editor. The screen as shown in figure is displayed



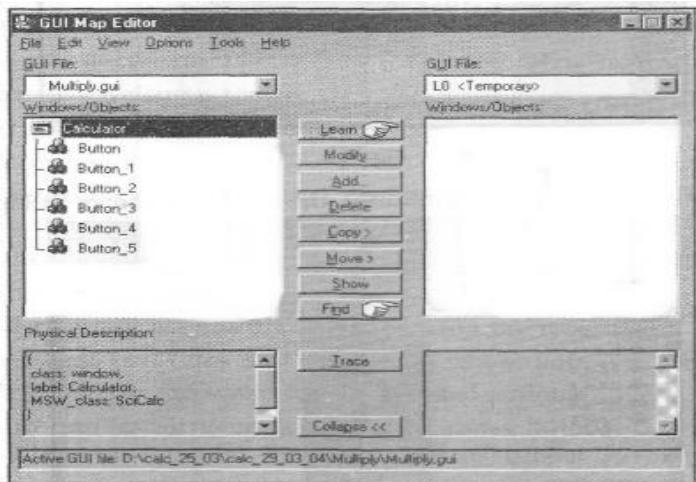
Step 4: Save the GUI Map file. File -> Save As A File dialog appears and you need to enter the filename.

Step 5: Close the GUI Map Editor window.

The procedure for loading the GUI map file is as follows:

Step 1: Open the GUI Map Editor. Tools -> GUI Map Editor

Step 2: Close all the opened GUI Map files File -> Close all.

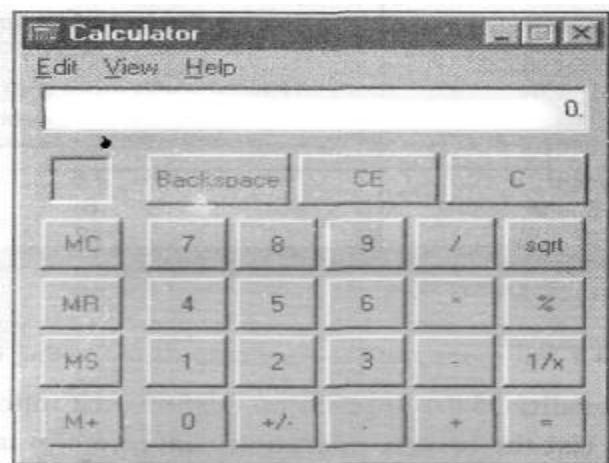


The procedure for running a test case is as follows:

Step 1: Open the test script to be executed.

Step 2: Run the test Run -> Run from top (or) press F5 to run the test.

WinRunner executes the generated script and displays the results in the Test Results window. We will now illustrate using WinRunner to test the "Standard Calculator" application available on your Windows system. You can invoke the calculator application from the desktop Start -> Programs -> Accessories -> Calculator. The GUI of the "Calculator" application is shown in Fig.



The symbols on the buttons of Calculator application represent the following functions:

+	:	To perform addition
-	:	To perform subtraction
*	:	To perform multiplication
/	:	To perform division
.	:	Decimal point
sqrt	:	To find square root of a number
%	:	To find percent
1/x	:	To find inverse of a number
MC	:	To clear the memory
MR	:	To recall from memory
MS	:	To save in the memory

M+	:	To add to the memory
C	:	To clear the current calculation
CE	:	To clear the displayed number
+/-	:	To give sign to a number (positive or negative)
Backspace:		To remove left most digit

To test the complete functionality of the application, we need to generate test cases in such a way that all the buttons are made use of. We need to generate some test cases which will give correct output and also some test cases which will give error messages. Table gives such test cases and the expected output for each test case.

### Test Cases and the Expected Output for Testing the Calculator

Test Case	Expected Output
4 1/x	0.25
- 6 sqrt	Err: "Invalid input for function"
4 C	Clears the Display
1.2 * 3	3.6
5 / 2.0	2.5
7 + 8 - 9	6
600 * 2 %	12
2, MS, C, MR	2
MC, 2, M+, 3, M+, C, MR	5

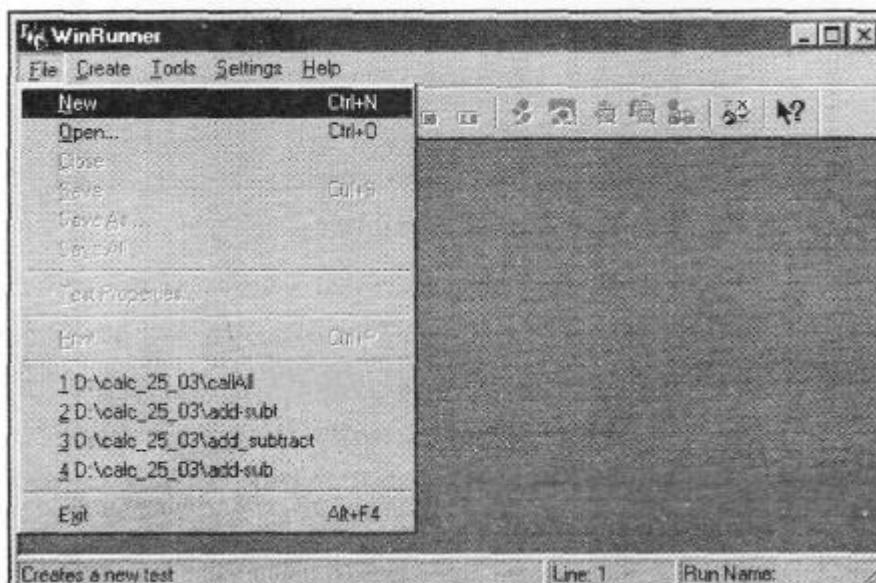
To test the functionality of the application perform the following steps:

**Test Case #1:** To test the Inverse operation (inverse of 4 using 1/x button)

Step 1: Open WinRunner application.

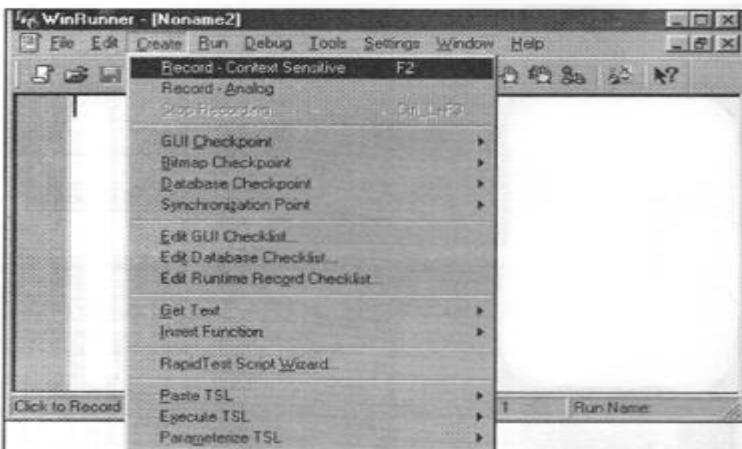
Step 2: Open Calculator application.

Step 3: Create a new document as shown in Figure. File -> New or Click Q (New) on tool bar or press Ctrl+N



Step 4: Start recording Create -> Record-Context Sensitive (or) press F2 (or) Click # on the toolbar

Click the (Record-Context Sensitive) button on the toolbar of WinRunner as shown in Figure or Select "Record - Context sensitive" option from the "Create" menu as shown in Figure.

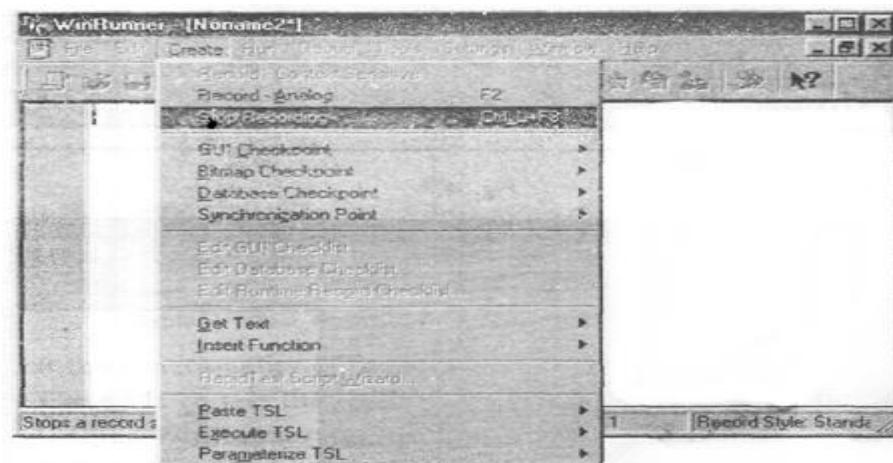


Step 5: Select the Calculator application and start recording the actions. a Click "4" on the Calculator

Click the "1/x" button on the Calculator to find the inverse of 4. The result, 0.25 will be displayed on the Calculator.

Step 6: Stop the Recording process.

Create -> Stop Recording (or) Click (Stop) on toolbar Click (Stop Recording) button on the toolbar of WinRunner as shown in Figure or Select the "Stop Recording" option from the "Create" menu as shown in Figure.

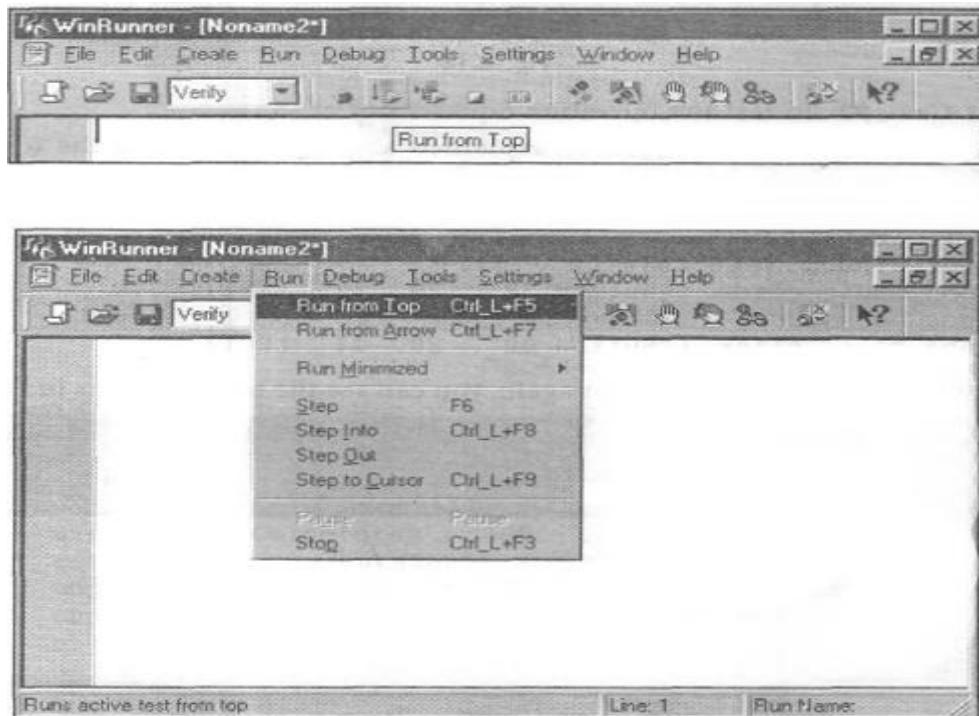


Step 8: Save the file as "inverse" in the selected folder.

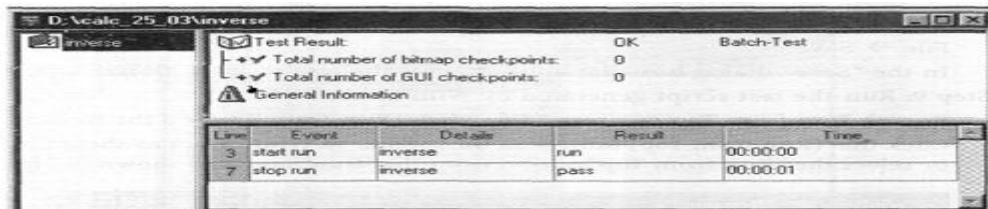
File -> Save In the "Save" dialog box that appears, save the test script with name "inverse".

Step 9: Run the test script generated by WinRunner. Run -> Run from Top or press F5 or Click (Run

from Top) on the toolbar Click the (Run from Top) button on the toolbar of WinRunner as shown in Figure or select the "Run from Top" option from the "Run" menu as shown in Figure.



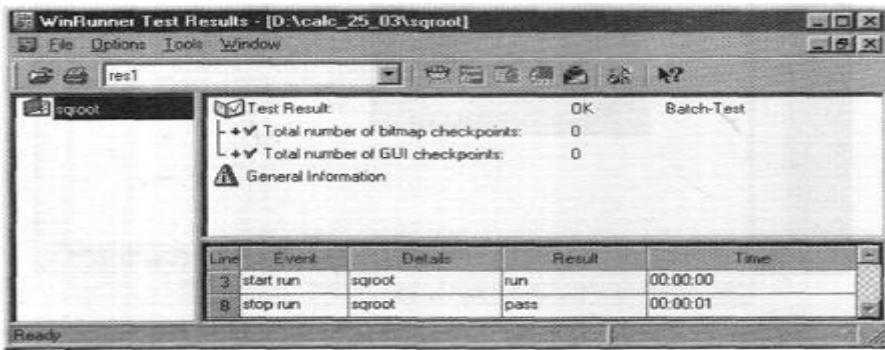
Step 10: After executing the TSL statements, WinRunner generates test results as shown in Figure. The Results column indicates whether the test has "Passed" or "Failed". The test results also give useful information such as the name of the test case, the line numbers in the test script and the time taken for executing the test case.



You can use the same procedure explained above for recording the test case. The following test script will be generated:

```
#Calculator  
winactivate("Calculator");  
set_window("Calculator",1);  
obj_mouse_click ("Button_38",20,12,LEFT);  
objmousedrag ("Button_35",10,15,11,14,LEFT);  
obj_mouse_click ("Button_60",20,11,LEFT);
```

When you run the test script again, you can see the test results, as in Figure.



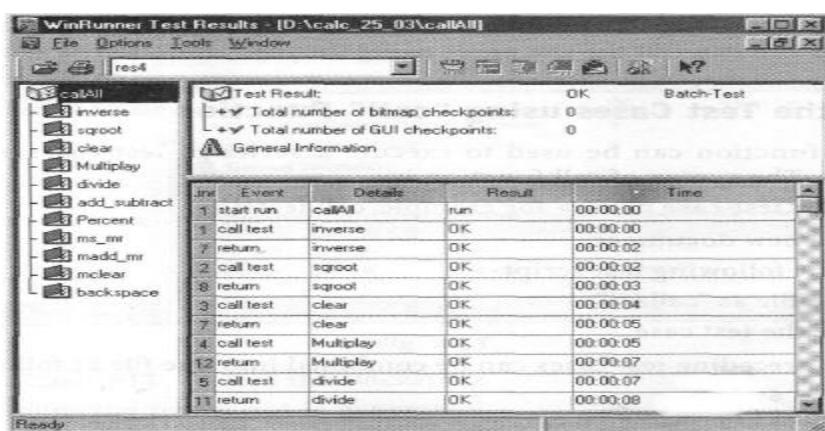
### Calling the Test Cases using "call" Function

The "call" function can be used to execute a series of test cases without any user interaction. The syntax of call function is: call for example, call test1();

Create a new document Write the following test script Save the file as "callAll" Execute the test case All the preceding test cases can be combined into one file as follows:

```
call inverseO; call sqrootO;
call clearO;
call MultiplayO;
call divideO;
call add_subtract();
call PercentO;
call msjnrO;
call maddmrO;
call mclearO;
call backspaceO;
```

When you execute this test script, all the earlier test cases are executed in one shot. The test results screen will be as shown in Figure. As you can see from the table, the "Details" column gives the various test cases executed. The "Result" column shows whether the test has passed or failed. The "Time" column gives the time taken to execute the test case.



When you have to retest the application using the same test cases, you can run the script in unattended mode. You can save the script in a file and run the script at specified time.

This feature of WinRunner is extremely useful for regression testing. When you are developing the software, you need to run the same set of test cases many times. So, you can run the application once,

generate the test script and then keep doing the regression testing. Obviously, the productivity of the test engineers will be very high when this tool is used.

#### **Viva questions**

1. Explain WinRunner testing process?
2. What is contained in the GUI map?
3. How does WinRunner evaluate test results?
4. What is the use of GUI map?
5. What happens when GUI map file get loaded?

## **EXPERIMENT: 7**

**NAME OF THE EXPERIMENT:** Study of any web testing tool (e.g. Selenium)

### **What is Selenium?**

JavaScript framework that runs in your web browser Works anywhere JavaScript is supported Hooks for many other languages Java, Ruby, Python Can simulate a user navigating through pages and then assert for specific marks on the pages All you need to really know is HTML to start using it right away

### **Selenium IDE**

Selenium Integrated Development Environment (IDE) is a Firefox plugin that lets testers to record their actions as they follow the workflow that they need to test.

It provides a Graphical User Interface for recording user actions using Firefox which is used to learn and use Selenium, but it can only be used with Firefox browser as other browsers are not supported.

However, the recorded scripts can be converted into various programming languages supported by Selenium and the scripts can be executed on other browsers as well.

#### **Selenium - IDE Download**

**Step 1** – Launch Firefox and navigate to the following URL - <http://seleniumhq.org/download/>.

Under the Selenium IDE section, click on the link that shows the current version number as shown below.



The screenshot shows the SeleniumHQ website with the following details:

- Header:** SeleniumHQ Browser Automation, edit this page, search selenium: Go, Projects, Download, Documentation, Support, About.
- Left Sidebar:** Selenium Downloads, Latest Releases, Previous Releases, Source Code, Maven Information, Donate to Selenium (with PayPal), and a Donate button.
- Downloads Section:** Downloads, Selenium IDE. It says: "Below is where you can find the latest releases of all the Selenium components. You can also find a list of previous releases, source code, and additional information for Maven users (Maven is a popular Java build tool)."
- Selenium IDE Sub-section:** Describes Selenium IDE as a Firefox plugin for recording and playing back user interactions. It links to the latest release (version 2.5.0) and release notes.
- Bottom:** A note about an unreleased version under development.

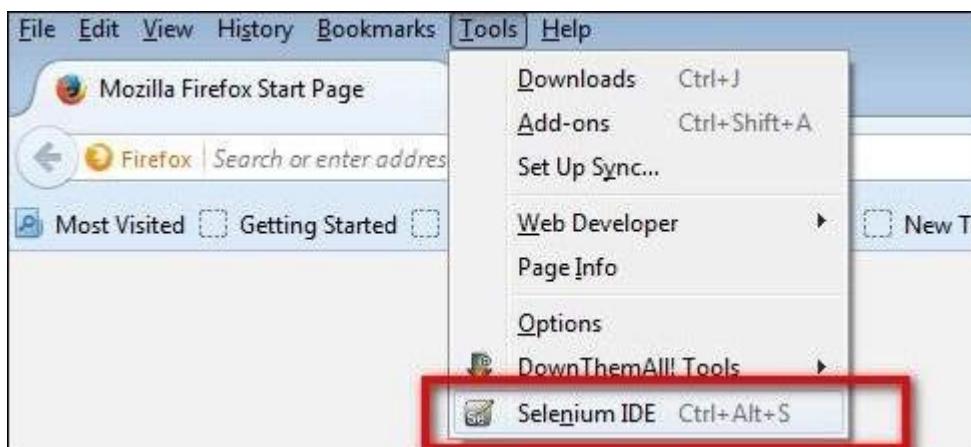
**Step 2** – Firefox add-ons notifier pops up with allow and disallow options. User has to allow the installation.



**Step 3** – The add-ons installer warns the user about untrusted add-ons. Click 'Install Now'.



**Step 4** – The Selenium IDE can now be accessed by navigating to Tools >> Selenium IDE.



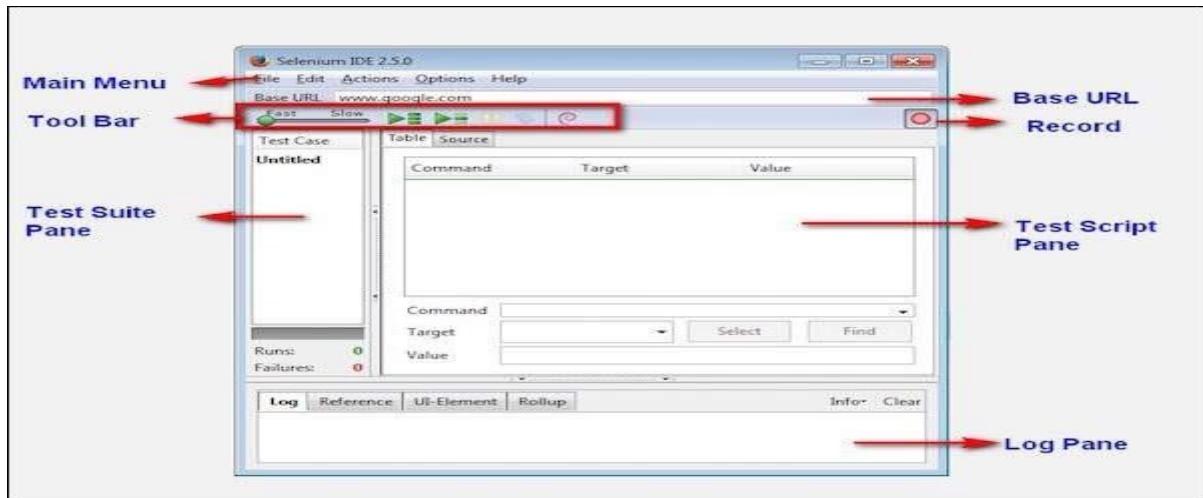
**Step 5** – The Selenium IDE can also be accessed directly from the quick access menu bar as shown below.



### Selenium IDE Features

This section deals with the features available in Selenium IDE.

The following image shows the features of Selenium IDE with the help of a simple tool-tip.



The features of the record tool bar are explained below.

Control	Control Name	Description
	Speed Control	This helps in controlling the speed of the test case runs.
	Run All	Executes the entire test suite that contains multiple test cases.
	Run	Executes the currently selected test.
	Pause/Resume	Allows user to pause or resume the script execution. Enabled only during the execution.
	Step	Helps user to debug the test by executing only one step of a test case at a time.
	Test Runner Mode	Allows user to execute the test case in a browser loaded with the selenium Core. It is an obsolete functionality that likely to be deprecated.
	Apply Rollup Rules	This features allows repetitive sequences of selenium commands to be grouped into a single action.
	Record	This features helps user to Records the user's browser actions.

## Creating Selenium IDE Tests

This section deals with how to create IDE tests using recording feature.

The following steps are involved in creating Selenium tests using IDE –

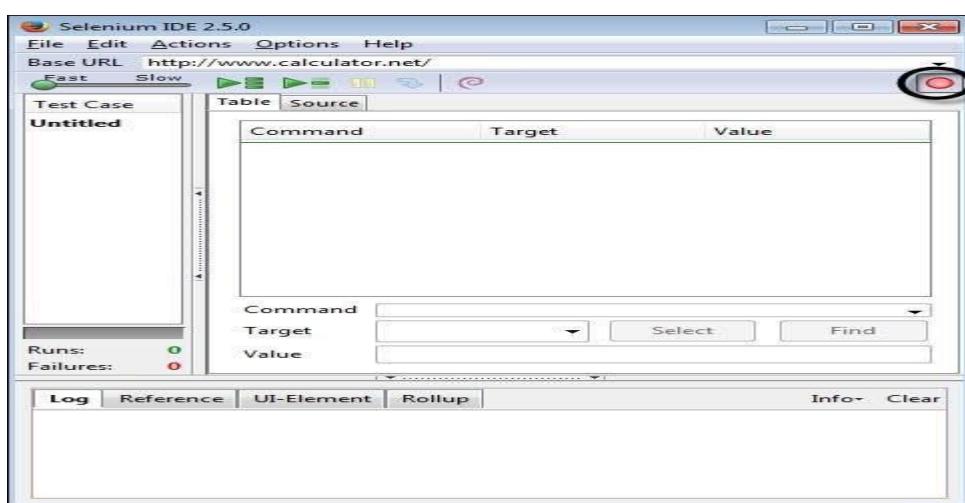
- Recording and adding commands in a test
- Saving the recorded test
- Saving the test suite
- Executing the recorded test

### Recording and Adding Commands in a Test

We will use [www.ncalculators.com](http://www.ncalculators.com) to demonstrate the features of Selenium.

**Step 1** – Launch the Firefox browser and navigate to the website - <https://www.ncalculators.com/>

**Step 2** – Open Selenium IDE from the Tools menu and press the record button that is on the top-right corner.



**Step 3** – Navigate to "Math Calculator" >> "Percent Calculator" >> enter "10" as number1 and 50 as number2 and click "calculate".



**Step 4** – User can then insert a checkpoint by right clicking on the webelement and select "Show all available commands" >> select "assert text css=b 5"



**Step 5** – The recorded script is generated and the script is displayed as shown below.

A screenshot of the Selenium IDE 2.5.0 interface. The left sidebar shows a 'Test Case' section with 'Untitled \*'. The main area has tabs 'Table' and 'Source'. A table is displayed with the following rows:

Command	Target	Value
open	/	
clickAndWait	xpath=//a[contains(text(),'Math')])[2]	
clickAndWait	link=Percent Calculator	
type	id=cpar1	10
type	id=cpar2	50
clickAndWait	css=input[type="image"]	
assertText	css=b	5

A red box highlights the last row of the table, which contains the 'assertText' command.

### Saving the Recorded Test

**Step 1** – Save the Test Case by navigating to "File" >> "Save Test" and save the file in the location of your choice. The file is saved as .HTML as default.

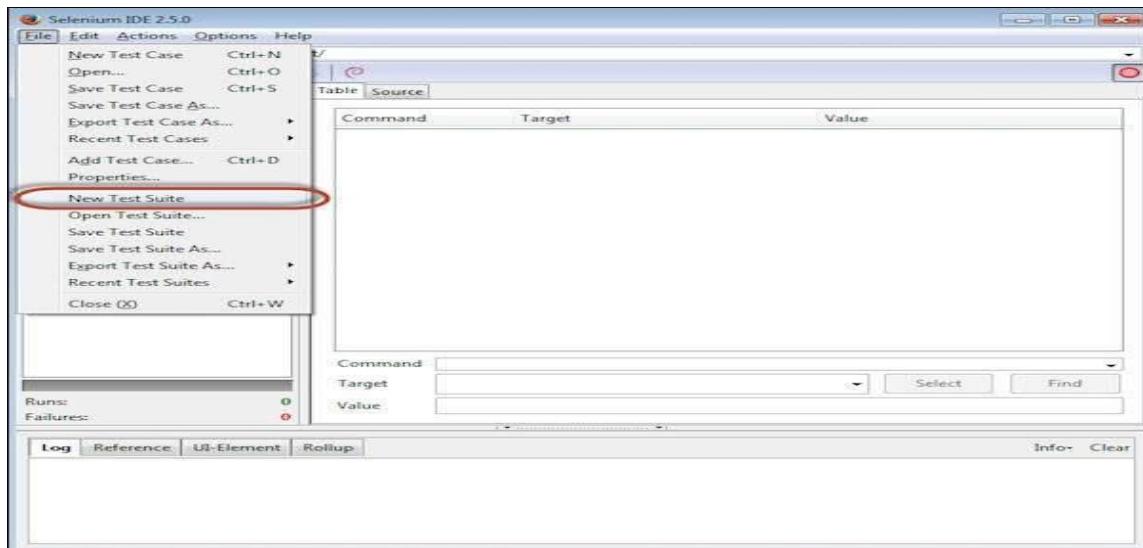
The test can also be saved with an extension htm, shtml, and xhtml.

Command	Target	Value
open	/	
clickAndWait	xpath=//a[contains(text(),'Math')][2]	
clickAndWait	link=Percent Calculator	
type	id=cpar1	10
type	id=cpar2	50
clickAndWait	css=input[type="image"]	
assertText	css=b	5

### Saving the Test Suite

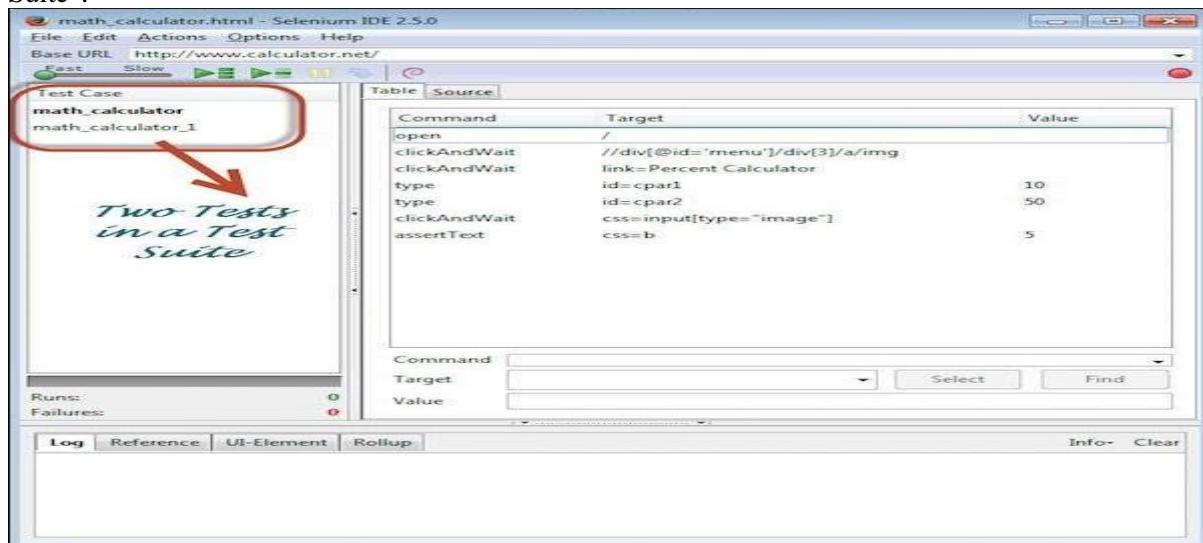
A test suite is a collection of tests that can be executed as a single entity.

**Step 1** – Create a test suite by navigating to "File" >> "New Test Suite" as shown below.



**Step 2** – The tests can be recorded one by one by choosing the option "New Test Case" from the "File" Menu.

**Step 3 –** The individual tests are saved with a name along with saving a "Test Suite".



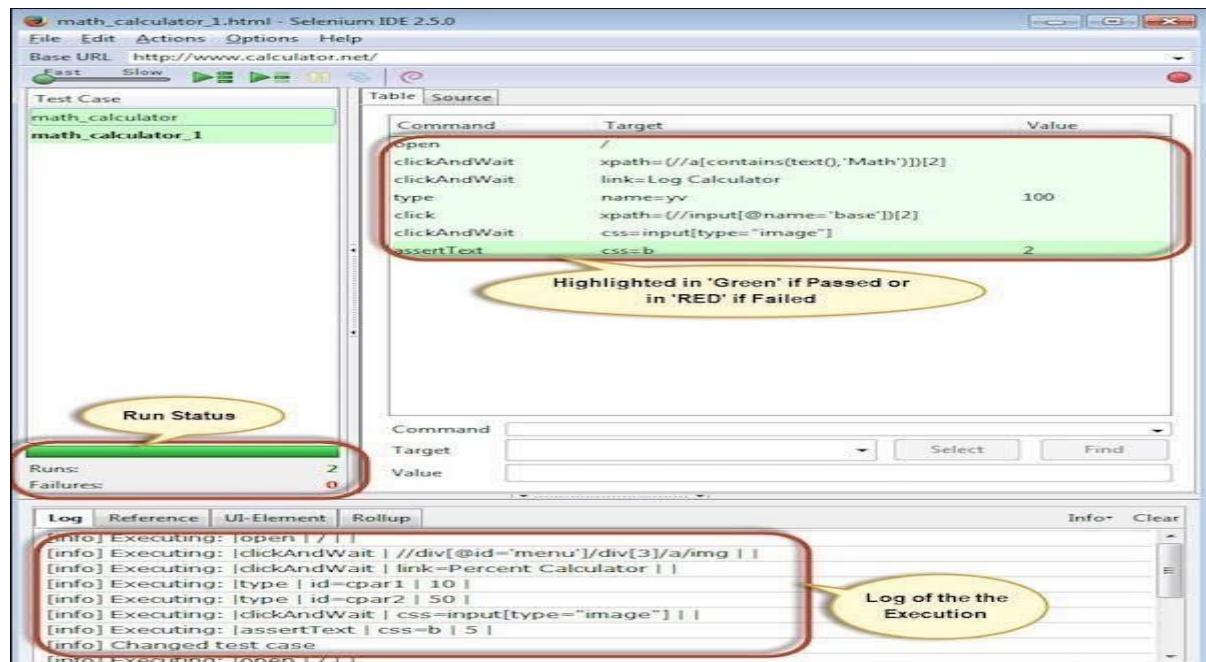
### Executing the Recorded Test

The recorded scripts can then be executed either by clicking "Play entire suite" or "Play current test" button in the toolbar.

**Step 1 –** The Run status can be seen in the status pane that displays the number of tests passed and failed.

**Step 2 –** Once a step is executed, the user can see the result in the "Log" Pane.

**Step 3 –** After executing each step, the background of the test step turns "Green" if passed and "Red" if failed as shown below.



### Selenium IDE Script Debugging

This section deals with debugging the Selenium IDE script.

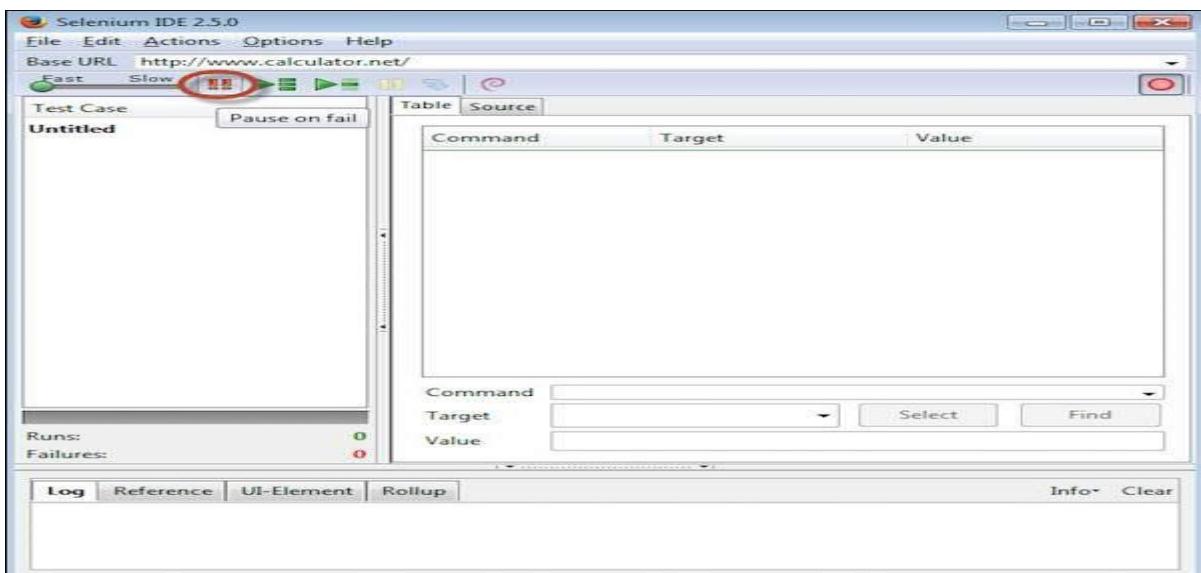
Debugging is the process of finding and fixing errors in the test script. It is a common step in any script development. To make the process more robust, we can make use of a plugin "Power Debugger".

for Selenium IDE.

**Step 1** – To install Power Debugger for Selenium IDE, navigate to <https://addons.mozilla.org/en-US/firefox/addon/power-debugger-selenium-ide/> and click "Add to Firefox" as shown below.



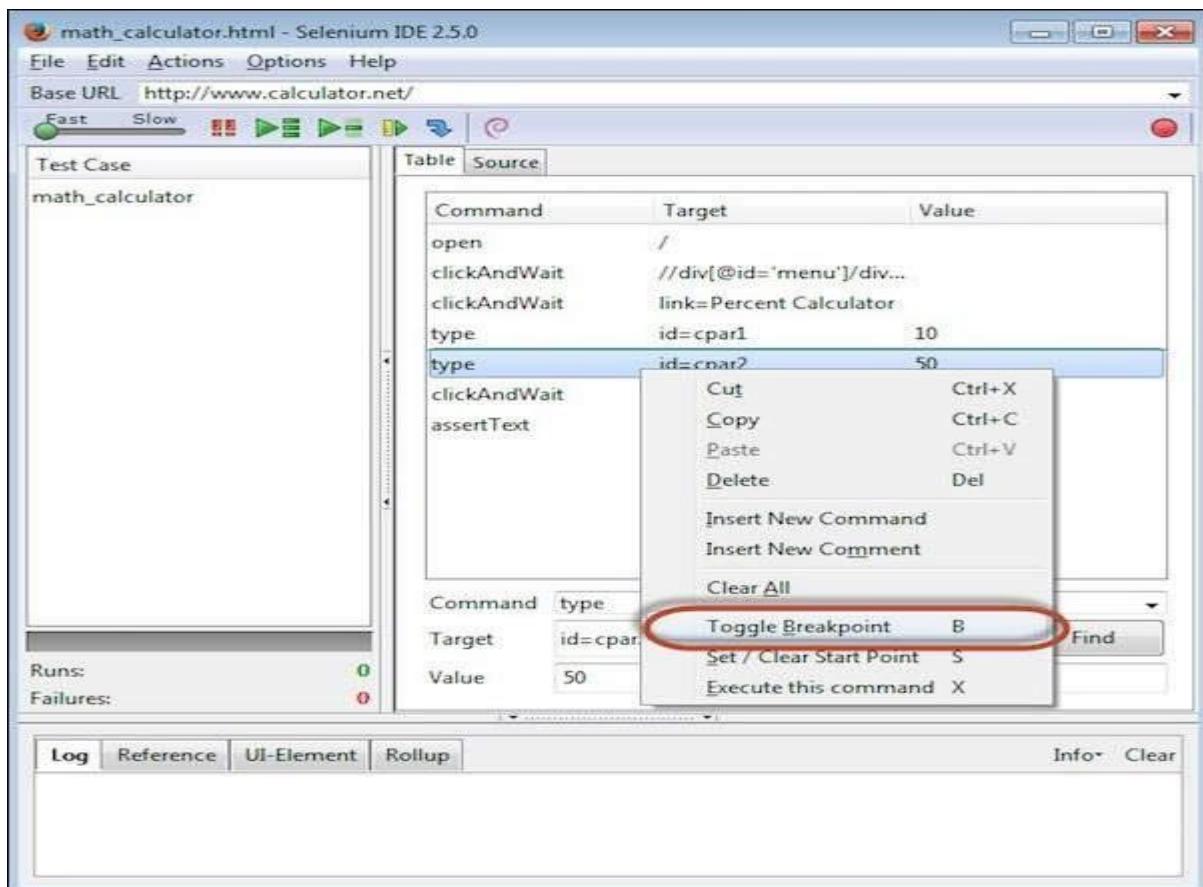
**Step 2** – Now launch 'Selenium IDE' and you will notice a new icon, "Pause on Fail" on recording toolbar as shown below. Click it to turn it ON. Upon clicking again, it would be turned "OFF".



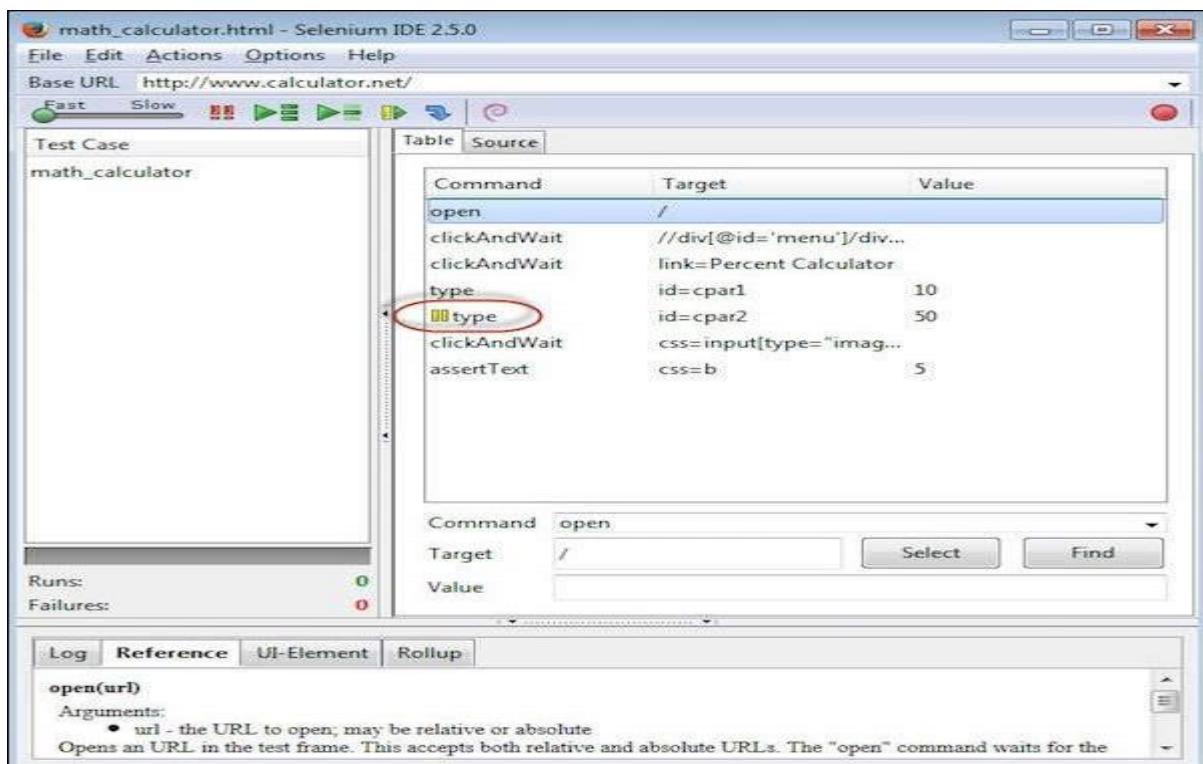
**Step 3** – Users can turn "pause on fail" on or off any time even when the test is running.

**Step 4** – Once the test case pauses due to a failed step, you can use the resume/step buttons to continue the test execution. The execution will **NOT** be paused if the failure is on the last command of any test case.

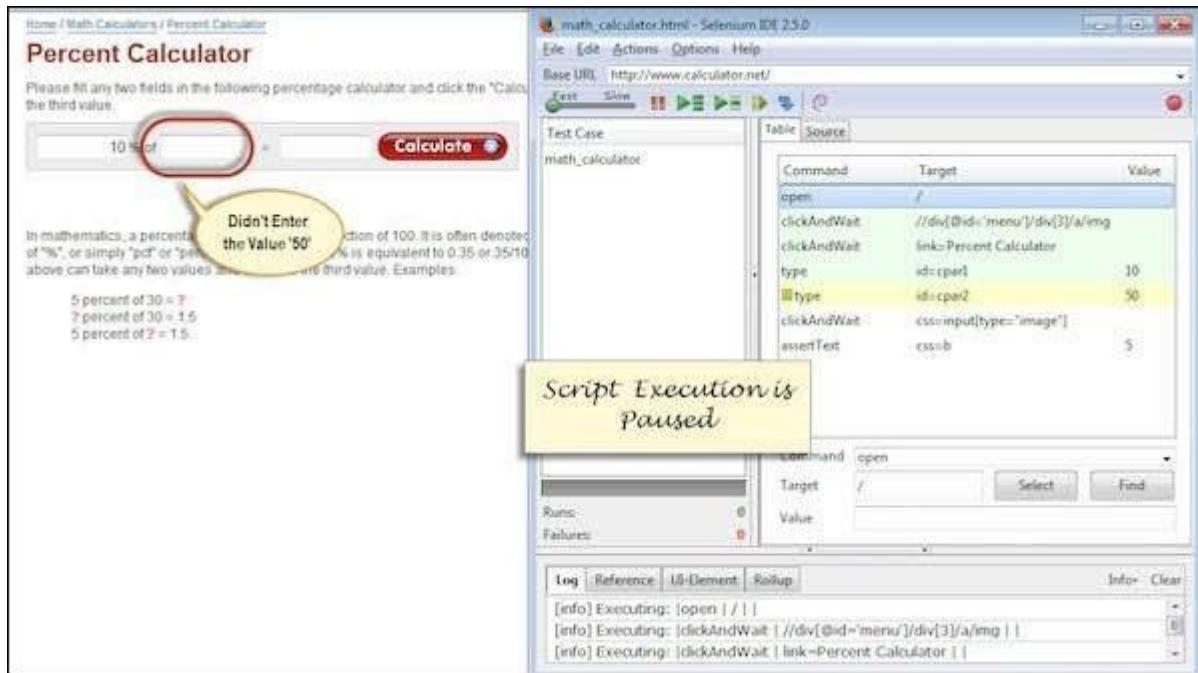
**Step 5** – We can also use breakpoints to understand what exactly happens during the step. To insert a breakpoint on a particular step, "Right Click" and select "Toggle Breakpoint" from the context-sensitive menu.



**Step 6** – Upon inserting the breakpoint, the particular step is displayed with a pause icon as shown below.



**Step 7** – When we execute the script, the script execution is paused where the breakpoint is inserted. This will help the user to evaluate the value/presence of an element when the execution is in progress.

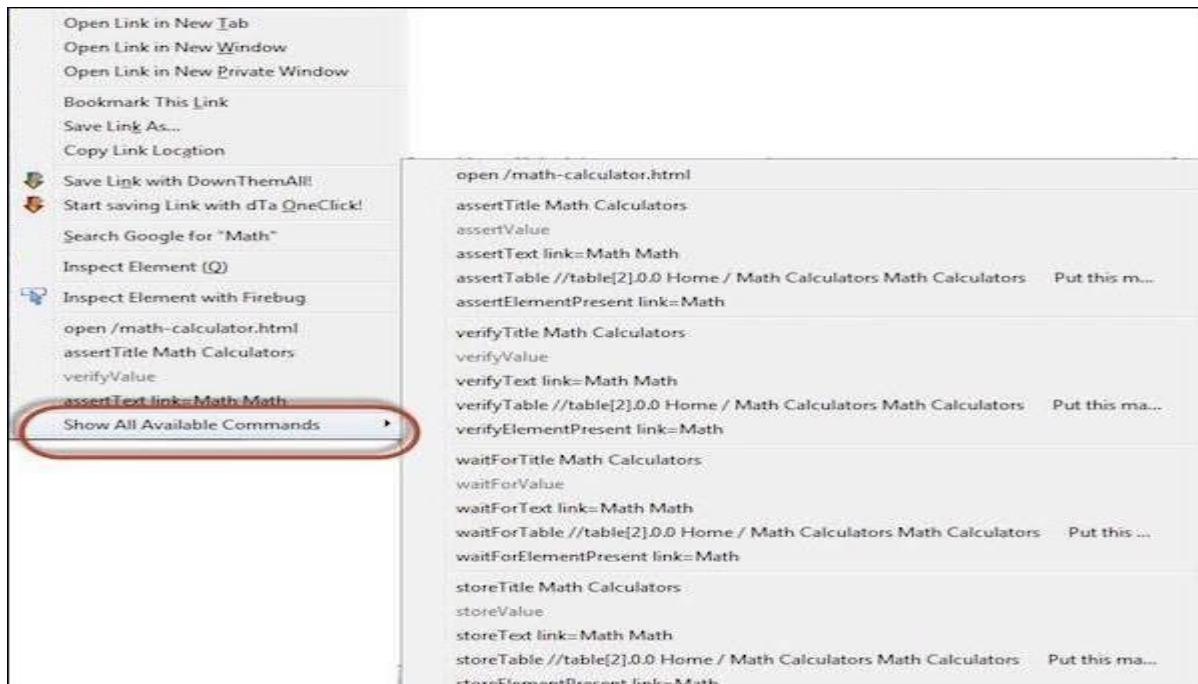


## Inserting Verification Points

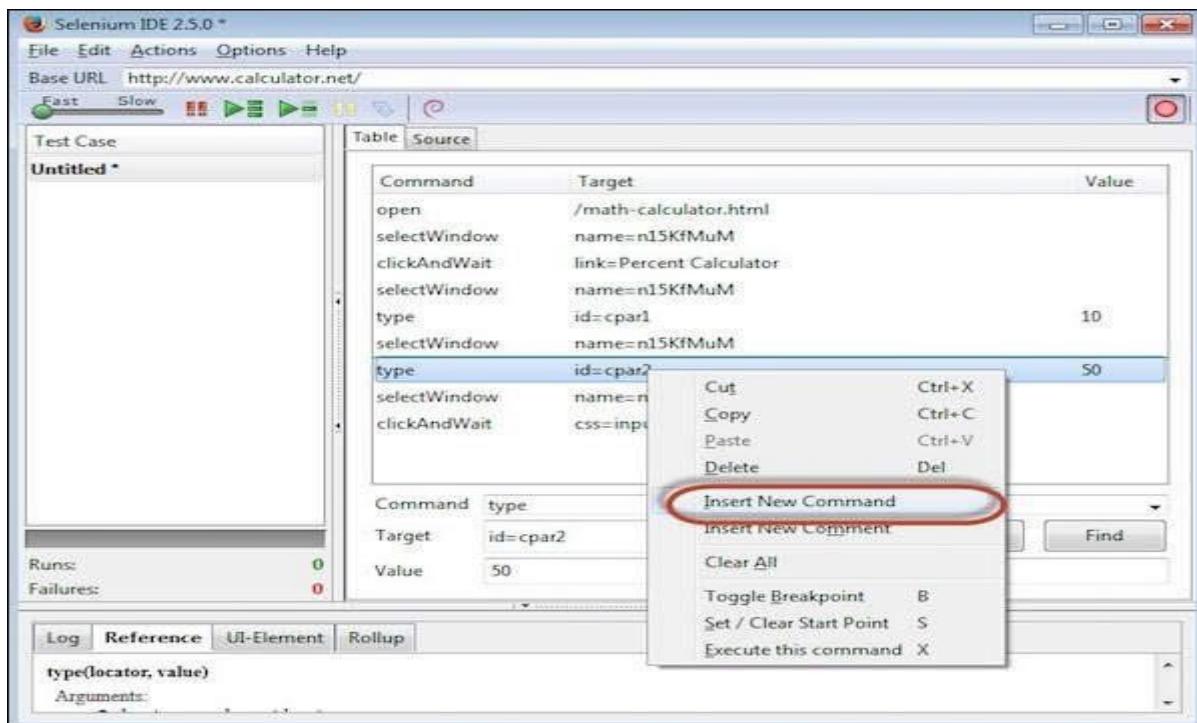
This section describes how to insert verification points in Selenium IDE.

The test cases that we develop also need to check the properties of a web page. It requires assert and verify commands. There are two ways to insert verification points into the script.

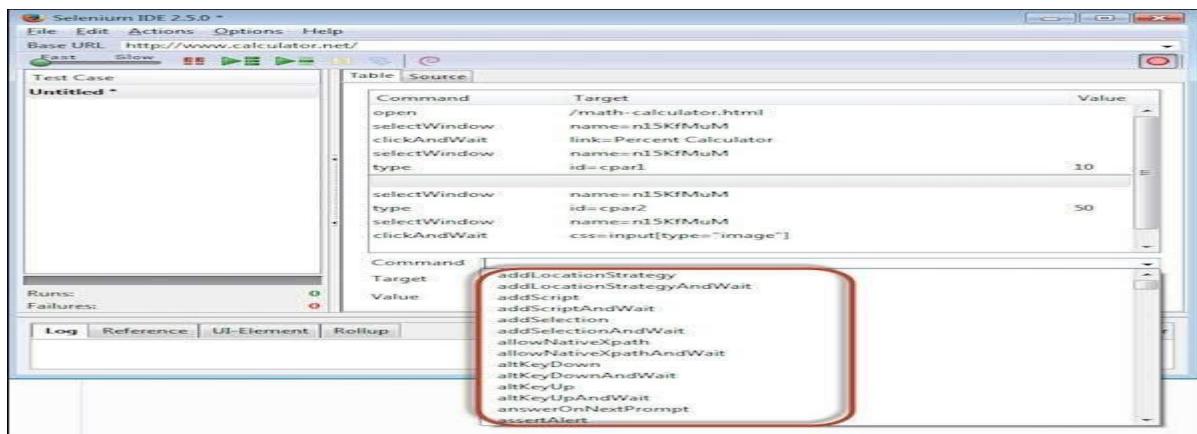
To insert a verification point in recording mode, "Right click" on the element and choose "Show all Available Commands" as shown below.



We can also insert a command by performing a "Right-Click" and choosing "Insert New Command".



After inserting a new command, click 'Command' dropdown and select appropriate verification point from the available list of commands as shown below.



Given below are the mostly used verification commands that help us check if a particular step has passed or failed.

- verifyElementPresent
- assertElementPresent
- verifyElementNotPresent
- assertElementNotPresent
- verifyText
- assertText
- verifyAttribute
- assertAttribute
- verifyChecked
- assertChecked

- verifyAlert
- assertAlert
- verifyTitle
- assertTitle

#### Synchronization Points

During script execution, the application might respond based on server load, hence it is required for the application and script to be in sync. Given below are few commands that we can use to ensure that the script and application are in sync.

- waitForAlertNotPresent
- waitForAlertPresent
- waitForElementPresent
- waitForElementNotPresent
- waitForTextPresent
- waitForTextNotPresent
- waitForPageToLoad
- waitForFrameToLoad

#### Selenium Pattern Matching

- This section deals with how to work with regular expressions using IDE.

Like locators, patterns are a type of parameter frequently used by Selenium. It allows users to describe patterns with the help of special characters. Many a time, the text that we would like to verify are dynamic; in that case, pattern matching is very useful.

Pattern matching is used with all the verification point commands - verifyTextPresent, verifyTitle, verifyAlert, assertConfirmation, verifyText, and verifyPrompt.

There are three ways to define a pattern –

- globbing
- regular expressions, and
- exact patterns.

#### Globbing

Most techies who have used file matching patterns in Linux or Windows while searching for a certain file type like \*.doc or \*.jpg. would be familiar with term "globbing"

Globbing in Selenium supports only three special characters: \*, ?, and [ ].

- \* – matches any number of characters.
- ? – matches a single character.

- [ ] – called a character class, lets you match any single character found within the brackets.  
[0-9] matches any digit.

To specify a glob in a Selenium command, prefix the pattern with the keyword 'glob:'. For example, if you would like to search for the texts "tax year 2013" or "tax year 2014", then you can use the glob "tax year \*" as shown below.

However the usage of "glob:" is optional while specifying a text pattern because globbing patterns are the default in Selenium.

Command	Target	Value
clickAndWait	link = search	
verifyTextPresent	glob: tax year *	

#### Exact Patterns

Patterns with the prefix 'exact:' will match the given text as it is. Let us say, the user wants an exact match with the value string, i.e., without the glob operator doing its work, one can use the 'exact' pattern as shown below. In this example the operator '\*' will work as a normal character rather than a pattern-matching wildcard character.

Command	Target	Value
clickAndWait	link = search	
verifyValue	exact: *.doc	

#### Regular Expressions

Regular expressions are the most useful among the pattern matching techniques available. Selenium supports the complete set of regular expression patterns that Javascript supports. Hence the users are no longer limited by \*, ? and [] globbing patterns.

To use RegEx patterns, we need to prefix with either "regexp:" or "regexpi:". The prefix "regexpi" is case-insensitive. The glob: and the exact: patterns are the subsets of the Regular Expression patterns. Everything that is done with glob: or exact: can be accomplished with the help of RegExp.

#### Example

For example, the following will test if an input field with the id 'name' contains the string 'tax year', 'Tax Year', or 'tax Year'.

Command	Target	Value
clickAndWait	link = search	
verifyValue	id = name	regexp:[Tt]ax ([Yy]ear)

## Selenium User Extensions

The Java script that allows users to customize or add new functionality.

It is easy to extend Selenium IDE by adding customized actions, assertions, and locator-strategies. It is done with the help of JavaScript by adding methods to the Selenium object prototype. On startup, Selenium will automatically look through the methods on these prototypes, using name patterns to recognize which ones are actions, assertions, and locators.

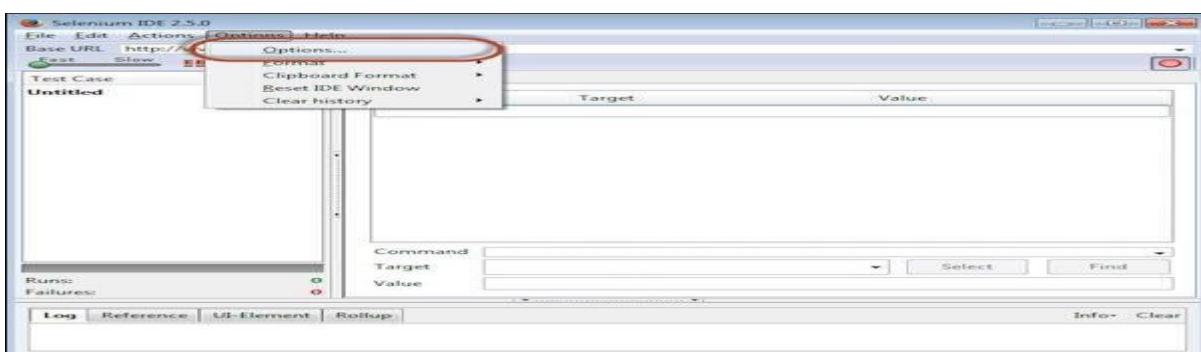
Let us add a 'while' Loop in Selenium IDE with the help of JavaScript.

**Step 1** – To add the js file, first navigate

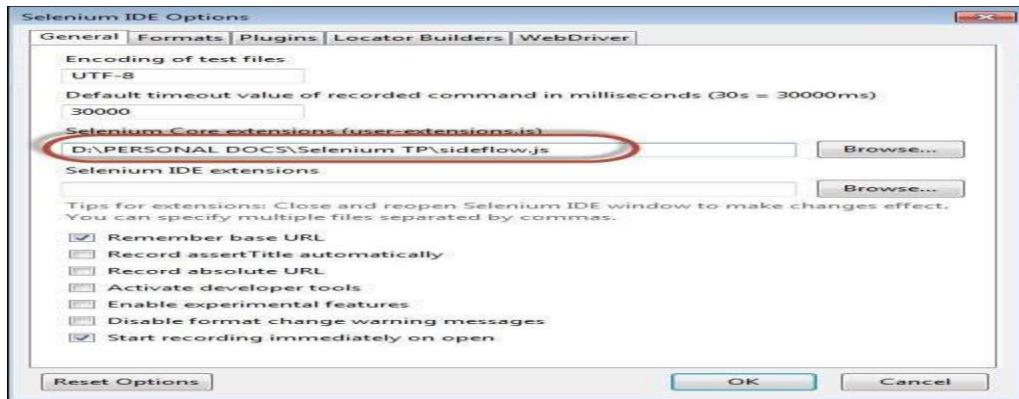
to <https://github.com/darrenderidder/sideflow/blob/master/sideflow.js> and copy the script and place save it as 'sideflow.js' in your local folder as shown below.



**Step 2** – Now launch 'Selenium IDE' and navigate to "Options" >> "Options" as shown below.



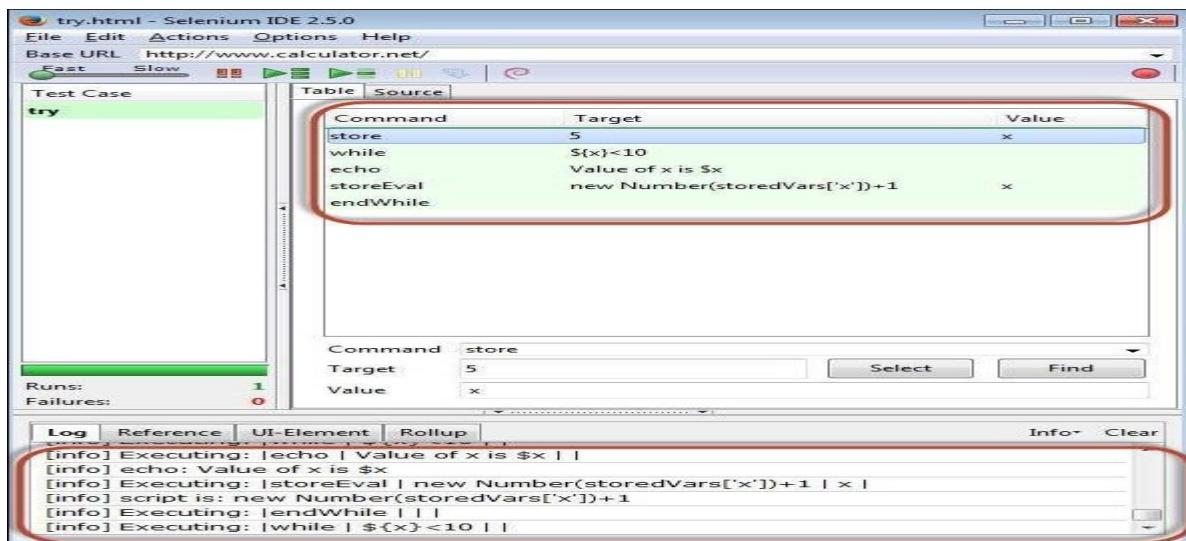
**Step 3** – Click the 'Browse' button under 'Selenium Core Extensions' area and point to the js file that we have saved in Step 1.



**Step 4** – Restart Selenium IDE.

**Step 5** – Now you will have access to a few more commands such as "Label", "While" etc.

**Step 6** – Now we will be able to create a While loop within Selenium IDE and it will execute as shown below.



#### Viva questions:

1. What is Selenium?
2. What are the different Selenium components?
3. What are the testing types that can be supported by Selenium?
4. What are the limitations of Selenium?
5. What is Selenese?

## **EXPERIMENT: 8**

**NAME OF THE EXPERIMENT:** Study of Any Bug Tracking Tool (Bugzilla)

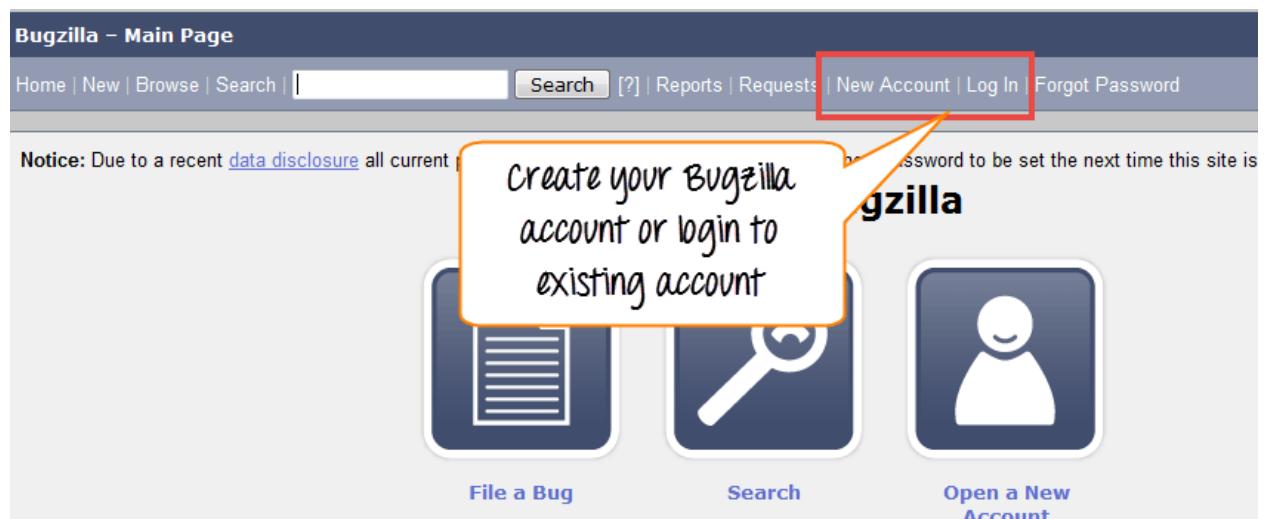
### **8.1 Introduction**

Bugzilla is a popular bug-tracking system that allows teammates and developers to track outstanding bug, Assign, track, and resolve error fixes and conflict resolutions using the Bugzilla database Communicate with teammates, Submit and review patches and Manage quality assurance (QA).

Bugzilla is one example of a class of programs called "Defect Tracking Systems", or, more commonly, "Bug-Tracking Systems". Defect Tracking Systems allow individual or groups of developers to keep track of outstanding bugs in their product effectively.

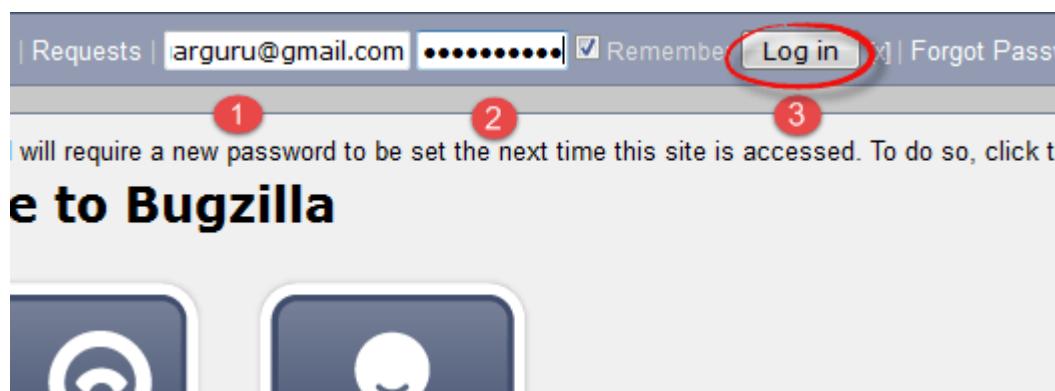
#### **How to log-in to Bugzilla**

**Step 1)** Use the following [link](#) for your handons. To create an account in Bugzilla 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

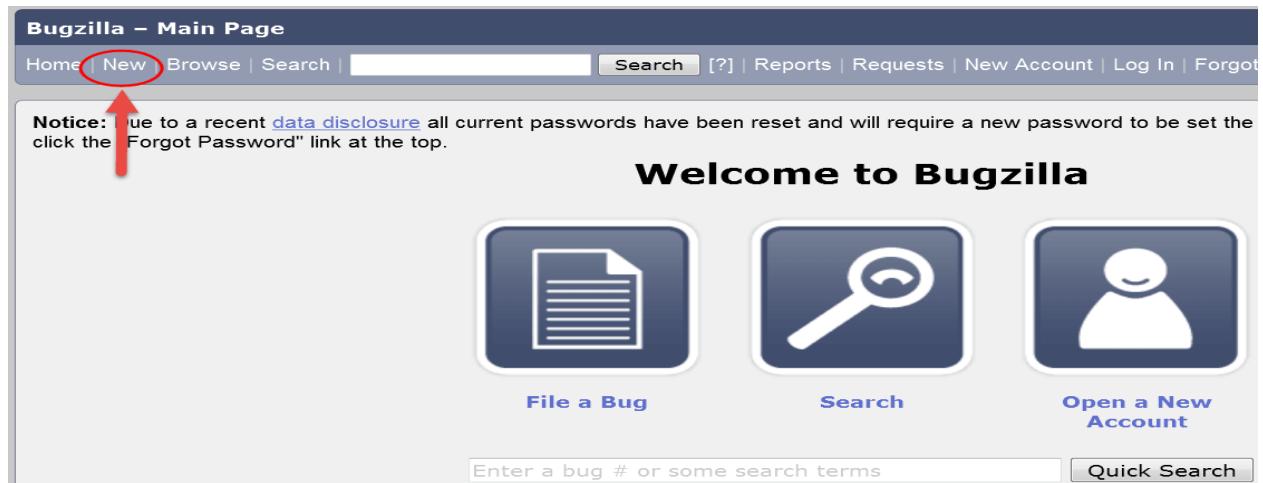
Reports | My Requests | Preferences | Log out jeegarguru@gmail.com

eset and will require a new password to be set the next time this site is acce

# Welcome to Bugzilla

## 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

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

**Notice:** Due to a recent [data disclosure](#) all current passwords have been reset and will require a new password to be set. Before reporting a bug, please read the [bug writing guidelines](#), please look at the list of [most frequently reported bugs](#).

Show Advanced Fields (\* = Required Field)

① * <b>Product:</b> Sam's Widget	<b>Reporter:</b> jeegarguru@gmail.com	
② <b>Component:</b> Widget Gears	Component Description Gears for Sam's widgets	
④ <b>Version:</b> unspecified	⑤ <b>Severity:</b> normal	
	⑥ <b>Hardware:</b> PC	
	⑦ <b>OS:</b> Windows NT	
We've made a guess at your operating system and platform. Check them and make any corrections if necessary.		
⑧ * <b>Summary:</b>	<input type="text"/>	
⑨ <b>Description:</b>	<input type="text"/>	
⑩ <b>Attachment:</b>	<input type="button" value="Add an attachment"/>	
<input type="button" value="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

① \* **Summary:** Gears for sams widget twisted

You must enter a Summary for this bug.

Possible Duplicates:	Bug ID	Summary	Status	
	<a href="#">7776</a>	when using the Widget Gears, the mV signal unexpectedly goes to 0	CONFIRMED	<input type="button" value="Add Me to the CC List"/>
	<a href="#">7777</a>	Widget Gears causes wrong mV signal to appear	CONFIRMED	<input type="button" value="Add Me to the CC List"/>
	<a href="#">12431</a>	Widget Gears cannot start	IN_PROGRESS	<input type="button" value="Add Me to the CC List"/>
	<a href="#">12480</a>	The Gear of sams widgets failed its validation	CONFIRMED	<input type="button" value="Add Me to the CC List"/>
	<a href="#">15407</a>	Sams Widget came pipe	CONFIRMED	<input type="button" value="Add Me to the CC List"/>
	<a href="#">21019</a>	Gears are bound up	CONFIRMED	<input type="button" value="Add Me to the CC List"/>
	<a href="#">21841</a>	Widget gears are stuck	CONFIRMED	<input type="button" value="Add Me to the CC List"/>

② **Description:** The widget gears are twisted at the end and not showing correct signal

**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)	Reported: 2015-01-07 02:50 PST by James														
Product: Sam's Widget	Modified: 2015-01-07 03:10 PST (History)														
Component: Widget Gears	CC List: 1 user including you (edit)														
Version: unspecified															
Hardware: PC Windows NT															
Importance: P2 normal															
et.Milestone: ---															
Assigned To: sam_folk-williams (edit) (take)															
QA Contact: (edit) (take)															
② URL:															
③ Whiteboard:															
④ Keywords:															
⑤ Tags:															
⑥ Depends on:															
⑦ Blocks:															
Show dependency tree / graph															
<table border="1"> <tr> <th>Orig. Est.:</th> <th>Current Est.:</th> <th>Hours Worked:</th> <th>Hours Left:</th> <th>%Complete:</th> <th>Gain:</th> <th>Deadline:</th> </tr> <tr> <td>0.0</td> <td>0.0</td> <td>0.0 + 0</td> <td>0.0</td> <td>0</td> <td>0.0</td> <td>2015-01-09</td> </tr> </table> <p>Summarize time (including time for bugs blocking this bug)</p>		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
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									
<b>Attachments</b> <a href="#">Add an attachment</a> (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.**

Orig. Est.:	Current Est.:	Hours Worked:	Hours Left:	%Complete:	Gain:	Deadline:
0.0	0.0	0.0 + 0	0.0	0	0.0	<input type="button" value="Calendar"/>

Summarize time (including time for bugs)

**Attachments**  
[Add an attachment](#) (proposed patch, testcase, etc.)

**Additional Comments:**

**Status:** CONFIRMED (edit)  
 CONFIRMED  
 IN\_PROGRESS  
 RESOLVED

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

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

**Description [reply] [-]**

**Save Changes**

Collapsible All Comments  
[Expand All Comments](#)

You can select bug status over here

Select deadline for your bug-report

## 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 report. It will generate a report with crucial information.

**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.

① Vertical Axis: Severity

② Plot Data Sets:  Individually  Stacked

③ Horizontal Axis: Component

④ Multiple Images: <none>

⑤ Format:  Line Graph  Bar Chart  Pie Chart

⑥ Summary: contains all of the strings

⑦ Classification: Unclassified Widgets Mercury

⑧ Product: Sam's Widget

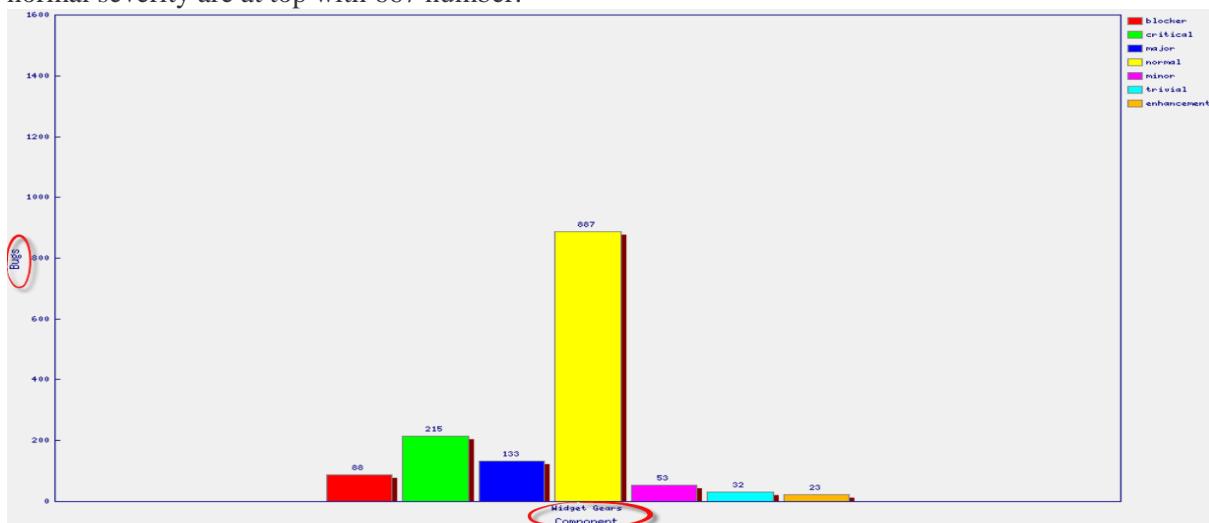
⑨ Component: Widget Gears

⑩ Status: UNCONFIRMED CONFIRMED IN\_PROGRESS RESOLVED VERIFIED

⑪ Resolution: FIXED INVALID WONTFIX LATER REMIND DUPLICATE

⑫ Generate Report

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 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 |   [?] Reports My Requests | Preferences | Help 1

**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.
- **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 report

**Bugzilla – Generate Graphical Report**

Home | New | Browse | 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 1

**Plot Data Sets:**  Individually  Stacked 5

**Horizontal Axis:** Deadline 2

**Multiple Images:** Severity 3

**Format:**  Line Graph  Bar Chart  Pie Chart 4

**Vertical labels:**

**Summary:** contains all of the strings 11

**Classification:** Unclassified, Widgets, Mercury 6

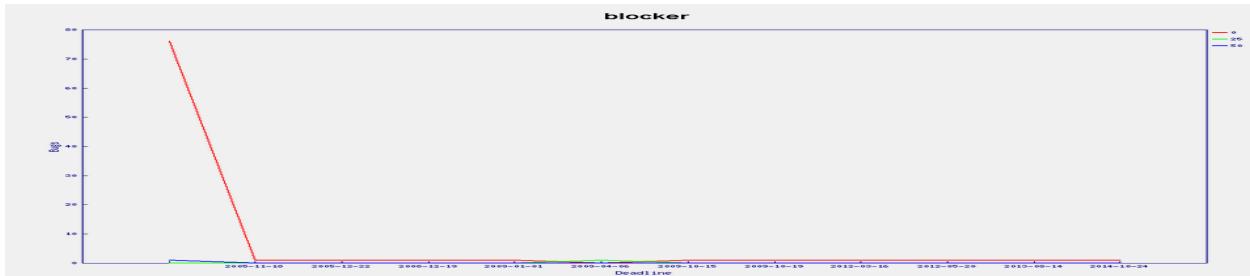
**Product:** Sam's Widget 7

**Component:** Widget Gears 8

**Status:** UNCONFIRMED, CONFIRMED, IN\_PROGRESS, RESOLVED, VERIFIED 9

**Resolution:** FIXED, INVALID, WON'T FIX, LATER, REMIND, DUPLICATE 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

**Step 3)** It opens another window, in this click on component "widget gears". Bugzilla Components are sub-sections of a product. For instance, here 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. Please click on the "Forgot Password" link at the top.

**Select a component that component:** It'll take you to the bugs in the Bug list created for that component.

**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

[Hide Search Description](#)

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	Changed
1256	Sam's Wi	Widget G	justdave@syndicomm.com	CONF	---	summary	2014-10-23
4219	Sam's Wi	Widget G	justdave@syndicomm.com	CONF	---	sdadasad	2012-06-13
4742	Sam's	Widget G	stdave@syndicomm.com	CONF	---	just a test	2009-01-26
5509	Sam's	Widget G	stdave@syndicomm.com	CONF	---	Test Bug	2014-05-04
2566	Sam's	Widget G	ndfill@gavinsharp.com	CONF	---	test	2009-11-04
6504	Sam's Wi	Widget G	mabst45@gmail.com	CONF	---	Won't run	2010-09-21
3010	Sam's Wi	Widget G	mickesnow@yahoo.com.mx	CONF	---	buhägs	2012-03-15
24741	Sam's Wi	Widget G	neha.malik028@gmail.com	CONF	---	cancel button not working	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](#)) [Save Changes](#)

<a href="#">Status:</a> CONFIRMED ( <a href="#">edit</a> )	<a href="#">Reported:</a> 2003-05-27 13:28 PDT by <a href="#">Jason McCallum</a>
<a href="#">Product:</a> Sam's Widget	<a href="#">Modified:</a> 2014-10-23 08:09 PDT ( <a href="#">History</a> )
<a href="#">Component:</a> Widget Gears	<a href="#">CC List:</a> <input checked="" type="checkbox"/> Add me to CC list 6 users ( <a href="#">edit</a> )
<a href="#">Version:</a> unspecified	<a href="#">See Also:</a> ( <a href="#">add</a> )
<a href="#">Hardware:</a> HP	<a href="#">Large text box:</a>
<a href="#">Importance:</a> P1	<a href="#">free text:</a>
<a href="#">Target Milestone:</a> ---	<a href="#">A multiple-select box:</a> Always Appears Also Always Appears Third Value, Always
<a href="#">Assigned To:</a> <a href="#">Dave Miller</a> ( <a href="#">edit</a> ) ( <a href="#">take</a> )	
<a href="#">QA Contact:</a> ( <a href="#">edit</a> ) ( <a href="#">take</a> )	
<a href="#">URL:</a>	
<a href="#">Whiteboard:</a>	

You can add users to CC list from here

You can change the assignee or QA contact from here itself

## How to use Simple search option in BugZilla

Bugzilla provides two ways of searching 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 **Unclassified** FoodReplicator Sp'ider Sécretions

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

**Open** ④

**Status:** All ⑤

**Product:** Sam's Widget

**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 secure SSL flash.

**Status:** Closed ①

**Product:** All

**Words:** widget gears

**Search** ②

*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	Summary	
<a href="#">12412</a> Widget G <a href="mailto:sam.folkwilliams@gmail.com">sam.folkwilliams@gmail.com</a>	RESO FIXE	<a href="#">The widget stole my spacecraft!!</a>	2011-08-27
<a href="#">2998</a> WeatherC <a href="mailto:peter.rutherford@cnh.com">peter.rutherford@cnh.com</a>	VERI FIXE	Hardlinks not created and the world is thence seriously out of control	2010-11-23
<a href="#">15235</a> WeatherC <a href="mailto:tara@bluemartini.com">tara@bluemartini.com</a>	RESO DUPL	<a href="#">weather widget failed rain test</a>	2011-06-14
<a href="#">4297</a> Widget G <a href="mailto:justdave@syndicomm.com">justdave@syndicomm.com</a>	RESO FIXE	<a href="#">transation 01 - fails</a>	2010-03-31
<a href="#">6312</a> Widget G <a href="mailto:sam.folkwilliams@gmail.com">sam.folkwilliams@gmail.com</a>	RESO FIXE	<a href="#">lack of cogs in widget</a>	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
<b>8907</b>	<b>Sam's Wi</b>	<b>Widget G</b>	<b>sam.folkwilliams@gmail.com</b>	<b>CONF</b>

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.

The screenshot shows a search results page with a message '37 bugs found.' and a table of bugs. Below the table are several links: CSV, Feed, iCalendar, Change Columns, Change Several Bugs at Once, Send Mail to Bug Assignees, Edit Search, and Remember search as. A callout box points to the 'Time Summary' link in the original screenshot, with the text: 'Use these options to view your bug in Long format, XML format or just time summary'. Another callout box points to the 'Send Mail to Bug Assignees' link, with the text: 'You can use these options for sending mail to assignees, change several bugs at once, change columns and so on.'

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 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.

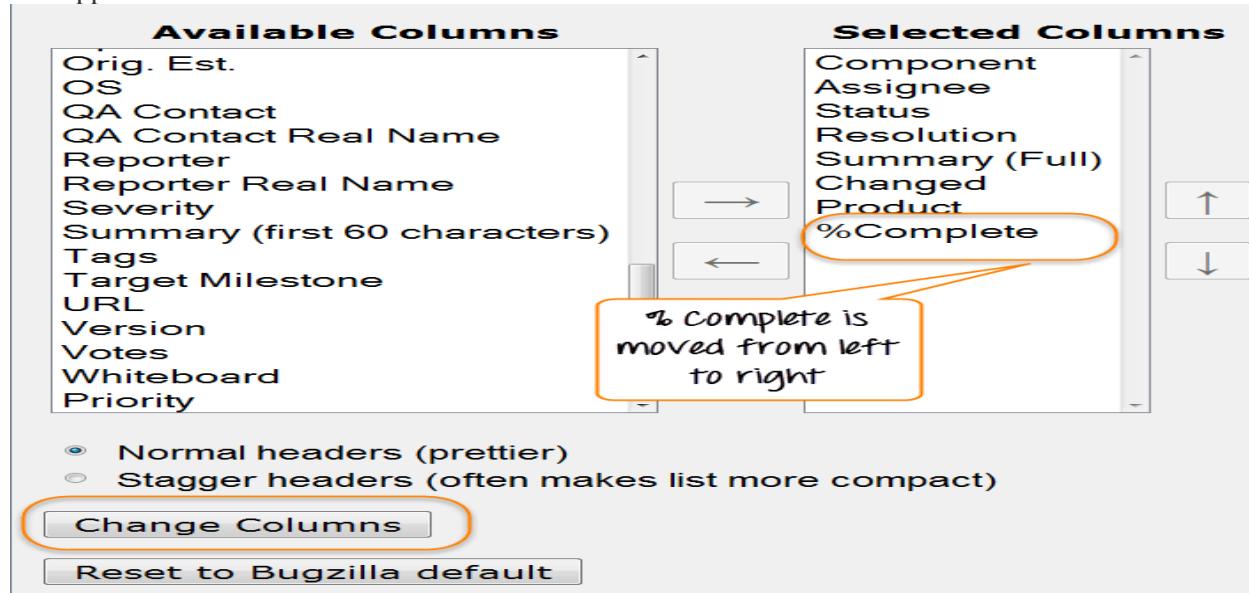
- Select any given option from the column you want to appear in the main screen - here we have selected **% complete**
- Click on the **arrow button**, it will move % complete column from  **Available Column** to the **Selected column**

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

The screenshot shows a dialog box titled 'Select the columns you wish to appear in your bug lists. Note that this'. It has two columns: 'Available Columns' and 'Selected Columns'. The 'Available Columns' list includes: OS, QA Contact, QA Contact Real Name, Reporter, Reporter Real Name, Severity, Summary (first 60 characters), Tags, Target Milestone, URL, Version, Votes, Whiteboard, Priority, and %Complete. The 'Selected Columns' list includes: Component, Assignee, Status, Resolution, Summary (Full), Changed, and Product. A red circle highlights the '%Complete' item in the available list. A blue arrow points from the available list to the selected list. A callout box points to the arrow with the text: 'You can select & move any given option to the selected columns'.

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	2014-12-02
12561	Sam's Wi	Widget G	sam.folkwilliams@gmail.com	CONF	---	Gear is broke	12-08
7359	Sam's Wi	Widget G	raydevereaux@bc.com	IN_P	---	funny	011-05-01
8907	Sam's Wi	Widget G	sam.folkwilliams@gmail.com	CONF	---	Monkey Wren gears	009-07-23

**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	2014-12-02	0 %
12561	Widget G	sam.folkwilliams@gmail.com	CONF	---	Gear is b	12-08	0 %
7359	Widget G	raydevereaux@bc.com	IN_P	---	funny	011-05-01	0 %
8907	Widget G	sam.folkwilliams@gmail.com	CONF	---	Monkey gears	009-07-23	0 %

### How to use Advance Search in BugZilla

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

- Click on advanced search option
- Select option for summary, how you want to search
- 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

**Summary:**

**Classification:**  
Unclassified  
**Widgets**  
Mercury

**Product:**  
FoodReplicator  
MyOwnBadSelf  
**Sam's Widget**  
WorldControl  
Spider Sécrétions

**Component:**  
SpiceDispenser  
Venom  
VoiceInterface  
WeatherControl  
Web  
**Widget Gears**

**Status:**  
**UNCONFIRMED**  
**CONFIRMED IN PROGRESS**

**Resolution:**  
**FIXED**  
INVALID

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

Summary: contains all of the strings widget gears twisted

Search

**Classification:**  
Unclassified  
**Widgets**  
Mercury

**Product:**  
Sam's Widget

**Component:**  
Widget Gears

**Status:**  
**UNCONFIRMED**  
**CONFIRMED**

**Resolution:**  
---  
**FIXED**

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

Wed Jan 7 2015 23:43:39 PST

XD 壴威

[Hide Search Description](#)

**Summary:** widget gears twisted    **Resolution:** ---    **Classification:** Widgets    **Status:** CONFIRMED    **Component:** Widget Gears    **Product:** Sam's Widget

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

Totals 0 %

One bug found.

[Long Format](#)

[XML](#)

[CSV](#) | [Feed](#) | [iCalendar](#) | [Change Columns](#) | [Edit Search](#)

[Time Summary](#)

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 various option like **changing Bugzilla general appearance, 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

The changes to your general preferences have been saved.

Bugzilla's general appearance (skin)

Dusk  
Site Default (Dusk)  
**Classic**  
Dusk

Background color of the screen appears off-white before skin appearance changed to Classic from Dusk

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

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

The changes to your general preferences have been saved.

General Preferences

Bugzilla's general appearance (skin) 2 Classic

Automatically add me to the CC list of bugs I am requested to review	Site Default (On)
Quote the associated comment when you click on its reply link	Site Default (Quote the full comment)
Position of the Additional Comments box	Site Default (Before other comments)
Timezone used to display dates and times	Site Default (Same as the server)

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 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.

Email Preferences

If you want to change your e-mail address [jeegarguru@gmail.com](mailto:jeegarguru@gmail.com) to which Bugzilla sends notifications, click the "Account Information" tab above.

If you don't like getting a notification for "trivial" changes to bugs, you can use the settings below to filter some or all notifications.

Enable All Mail  Disable All Mail 2

**Global options:** 3

- Email me when someone asks me to set a flag
- Email me when someone sets a flag I asked for

**Field/recipient specific options:** 4

When my relationship to this bug is:				I want to receive mail when:
Assignee	Reporter	CCed	QA Contact	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	I'm added to or removed from this capacity
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	A new bug is created

## 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 **editbugs, don't share, canconfirm, etc.** Choose the option as per your need.

The changes to your saved searches have been saved.

### Saved Searches

Your saved searches are as follows:

Search	Run	Edit	Forget	Show in Footer	Share With a Group
My Bugs	<a href="#">Run</a>	<a href="#">Edit</a>	<a href="#">Forget</a>	<input checked="" type="checkbox"/>	—
Widget gears	<a href="#">Run</a>	<a href="#">Edit</a>	<a href="#">Forget</a>	<input checked="" type="checkbox"/>	editbugs

You may use these searches saved and shared by others:

**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** ①

### Saved Searches

Your saved searches are as follows:

Search	Run	Edit	Show in Footer	Share With a Group	
My Bugs	<a href="#">Run</a> ②	<a href="#">Edit</a>	<input checked="" type="checkbox"/>	—	
Widget gears	<a href="#">Run</a>	<a href="#">Edit</a>	<a href="#">Forget</a>	<input checked="" type="checkbox"/>	canconfirm

We will run our search or queries from here

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 | XML | 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	<a href="#">Run</a>	<a href="#">Edit</a>	<input type="checkbox"/>
All Mercury	Ben Schultz (SofTechnics) <ben.schultz@softechnics.com>	editbugs	<a href="#">Run</a>	<a href="#">Edit</a>	<input checked="" type="checkbox"/>
ami	Ami <ami_nahmani@walla.com>	editbugs	<a href="#">Run</a>	<a href="#">Edit</a>	<input checked="" type="checkbox"/>
Bug1	Aruna <aruna_trimurti@yahoo.com>	editbugs	<a href="#">Run</a>	<a href="#">Edit</a>	<input type="checkbox"/>

Home | New | Browse | Search | [Search](#)

My Bugs | Widget gears  
[All Mercury](#) | [ami](#)  
 | 26320 | sams widget

Both these users can edit our bug

### viva questions

1. What is Bugzilla?
2. What are Bugzilla's features?
3. What are Bugzilla Components?
4. How do I change my user name in Bugzilla?
5. What is about the Bug List page?
6. How to Write a Useful Bug Report with Bugzilla

## **EXPERIMENT: 9**

**NAME OF THE EXPERIMENT:** Study of Any Test Management Tool ( Test Director)

Test Director is a global test management solution which provides communication, organization, documentation and structure to the testing project.

### **Test Director**

Software Automated Tool TestDirector simplifies test management by helping you organize and manage all phases of the software testing process, including planning, creating tests, executing tests, and tracking defects.

#### **PLANNING TESTS:**

Divide your application into test subjects and build a project.

##### **1. Define your testing goals.**

Examine your application, system environment, and testing resources to determine what and how you want to test.

##### **2. Define test subjects.**

Define test subjects by dividing your application into modules or functions to be tested. Build a test plan tree that represents the hierarchical relationship of the subjects.

##### **3. Define tests.**

Determine the tests you want to create and add a description of each test to the test plan tree.

##### **4. Design test steps.**

Break down each test into steps describing the operations to be performed and the points you want to check. Define the expected outcome of each step.



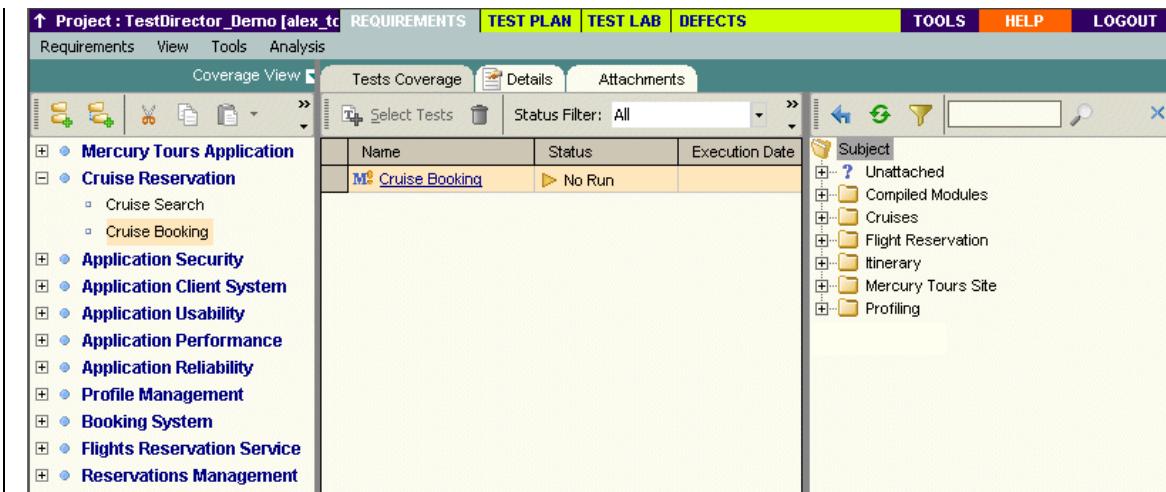
Test Plan Tree			
Step Name	Description	Expected Result	Actions
Display the Cruise Special page.	Click the Cruises button.	The Cruise Special page opens.	
Display the Cruise Reservation page.	Click the Now Accepting Reservations button.	The Cruise Reservation page opens.	
Book the cruise.	Enter passenger name, credit card information, and address. Click OK.	The Cruise Confirmation page opens.	
Print cruise confirmation.	Click the Print button.	A confirmation page is printed.	
Log off.	Click the Sign-Off button.	Returns to the Sign-On page.	

##### **5. Automate tests.**

Decide whether to perform each test manually or to automate it. If you choose to perform a test manually, the test is ready for execution as soon as you define the test steps. If you choose to automate a test, use WinRunner to create automated test scripts in Mercury Interactive's Test Script Language (TSL).

## 6. Analyze the test plan.

Generate reports and graphs to help you analyze your test plan. Determine whether the tests in the project will enable you to successfully meet your goals.



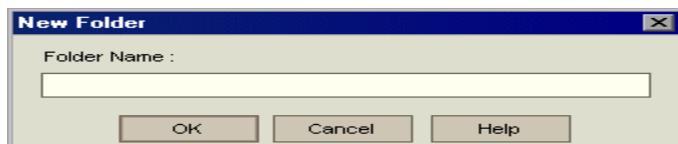
## RUNNING TESTS:

Create test sets and perform test runs.

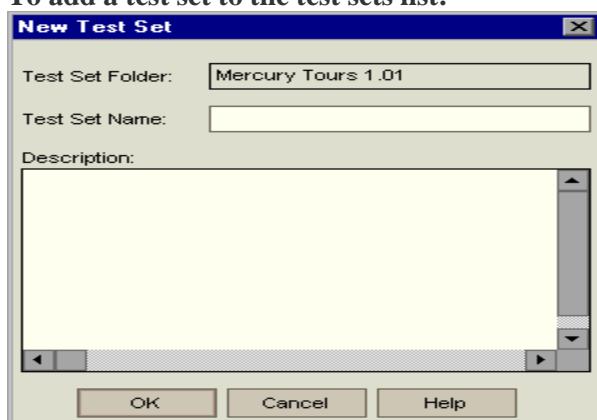
### 1. Create test sets.

Create test sets by selecting tests from the project. A test set is a group of tests you execute to meet a specific testing goal.

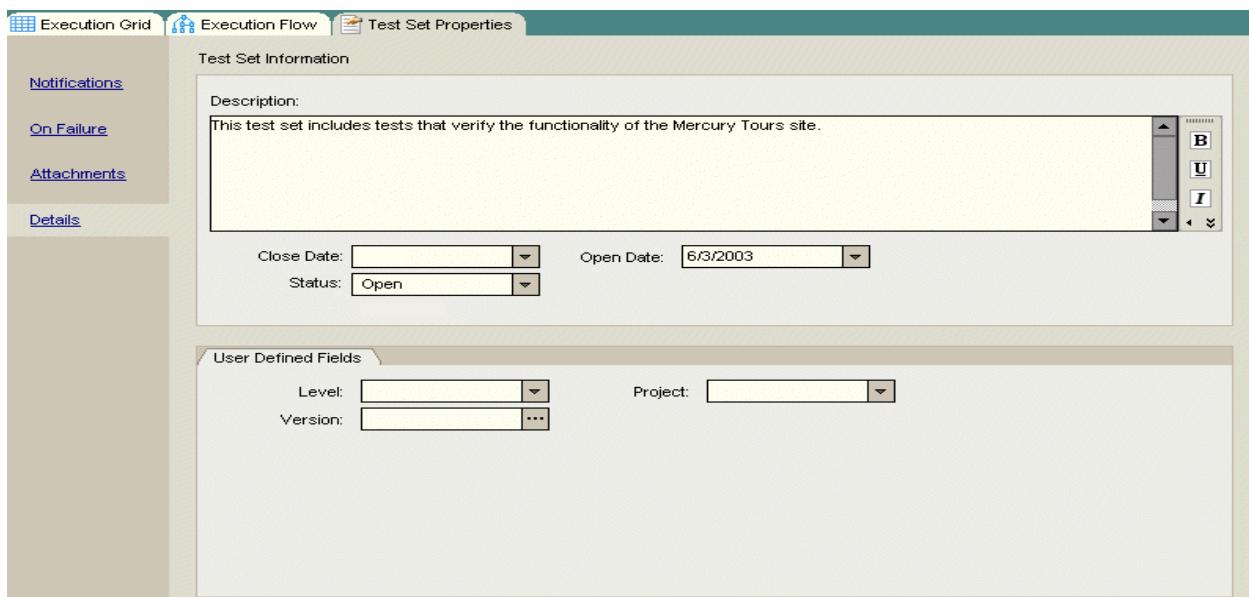
#### To define a test set:



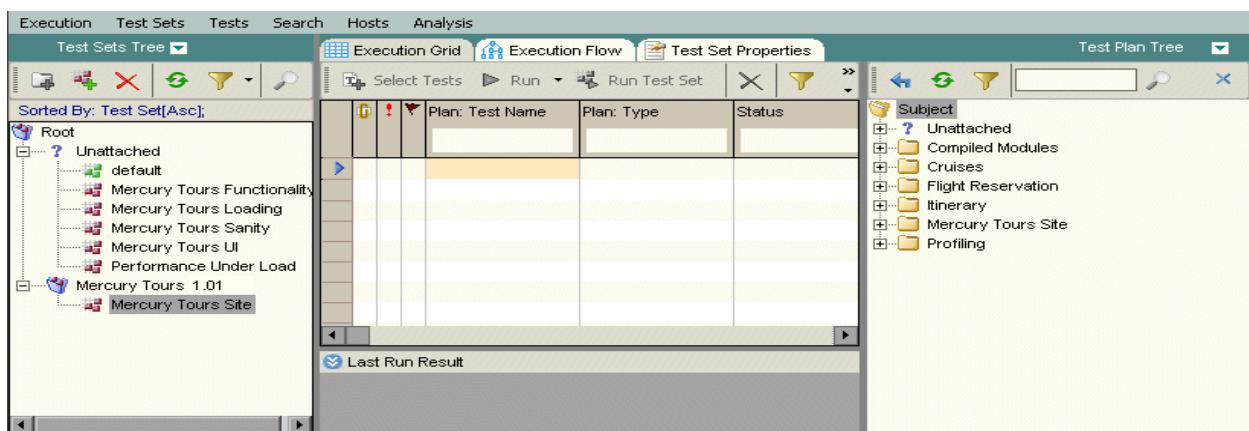
#### To add a test set to the test sets list:



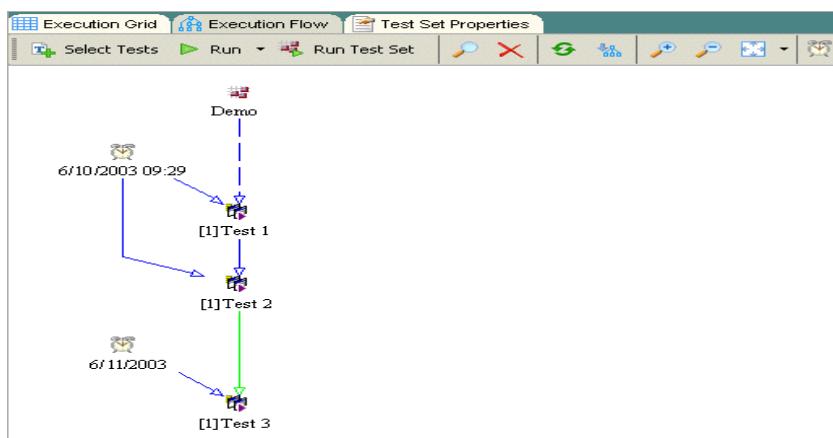
## Define the test set details:



## Adding tests to a test set:

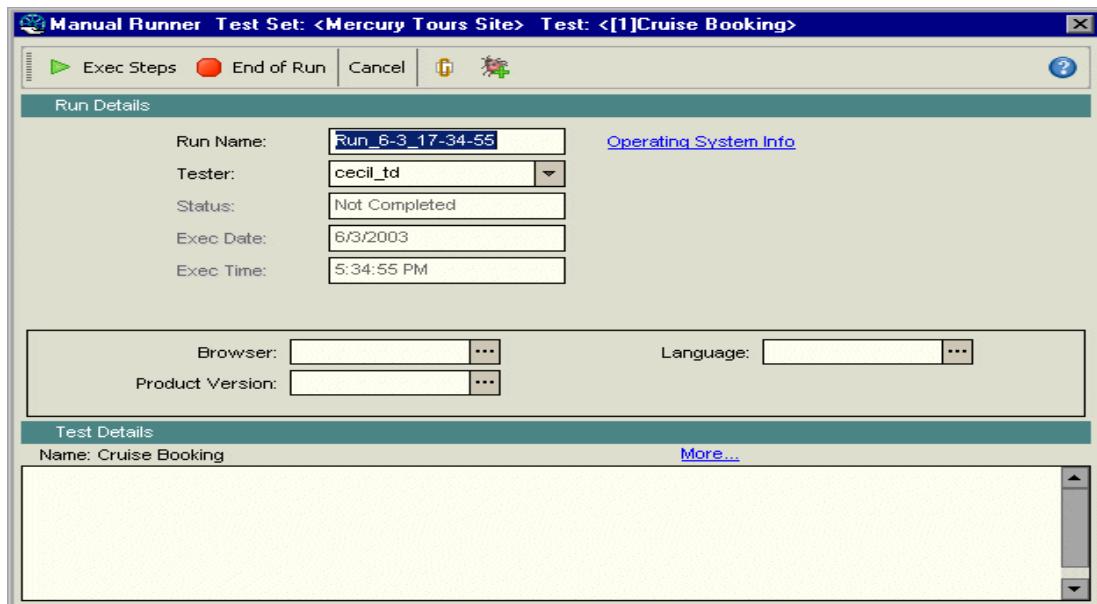


## Scheduling Test Runs:

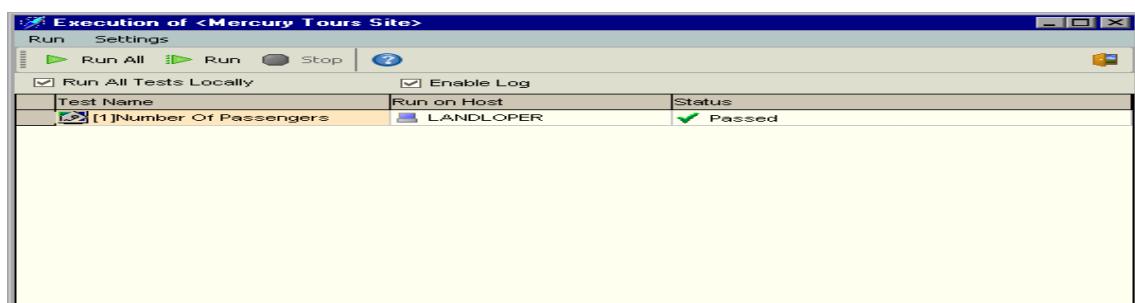


Schedule test execution and assign tasks to testers. Run the manual and/or automated tests in the test sets.

## Running Tests Manually:



Run the tests:



View a summary of test results in the execution grid:

The screenshot shows the 'Execution Grid' window. It features a grid of test results and a detailed 'Last Run Result' section:

Plan: Test Name	Plan: Type	Status	Planned Host Name	Responsible Tester
[1]Cancel All Reservations	VWR-AUTOMATED	No Run		
[1]Airline Preference	VWR-AUTOMATED	No Run		
[1]Number Of Passengers	QUICKTEST_TEST	Passed		
M [1]Cruise Booking	MANUAL	Failed		
[1]Cruise Search	VWR-AUTOMATED	No Run		

**Last Run Result:**

Step Name	Status	Description
Start Test		Description: Start Test Number Of Passengers
Start Global Iteration		
Start Action		
Replay Error	Passed	Expected: Actual:

## 4. Analyze the testing progress.

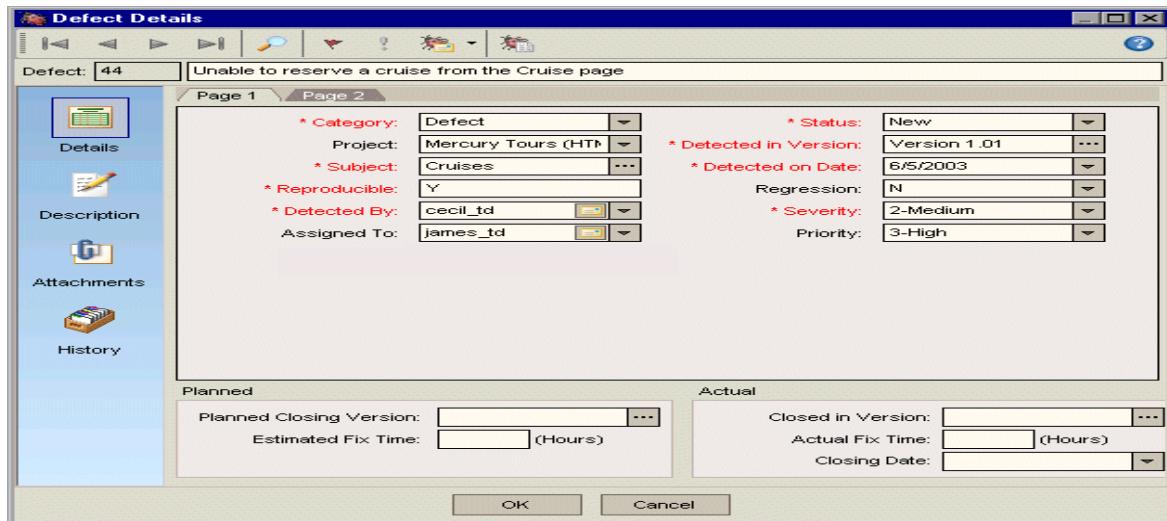
Generate reports and graphs to help you determine the progress of test execution

## **TRACKING DEFECTS:**

1. Report defects detected in your application and track how repairs are progressing.
2. Report defects detected in the software. Each new defect is added to the defect database.
3. Track defects.

Review all new defects reported to the database and decide which ones should be repaired. Test a new version of the application after the defects are corrected.

Updating Defects:



4. Analyze defect tracking.

Generate reports and graphs to help you analyze the progress of defect repairs, and to help you determine when to release the application.

- a) End a defect report to the TestDirector database.
- b) Review the defect and assign it to a member of the development team.
- c) Repair the open defect.
- d) Test a new build of the application after the defect is corrected. If the defect does not reoccur, change the status of the defect.
- e) Generate reports and graphs to help you analyze the progress of the defects in your TestDirector project.
- f) Reporting a New Defect

You can report a new defect at any stage of the testing process by adding a defect record to the project database. Each defect is tracked through four stages: New, Open, Fixed, and Closed. When you initially report a defect to the project database, you assign it the status New.

### **Viva questions**

1. What is test director?
2. What are the modules of test director?
3. What is the use of test director?
4. What are the various reports in test director?

## **EXPERIMENT: 10**

**NAME OF THE EXPERIMENT:** Study of any open source testing tool (TestLink)

Testlink is an open source test management tool. It enables creation and organization of test cases and helps manage into test plan.

### **Login to TestLink**

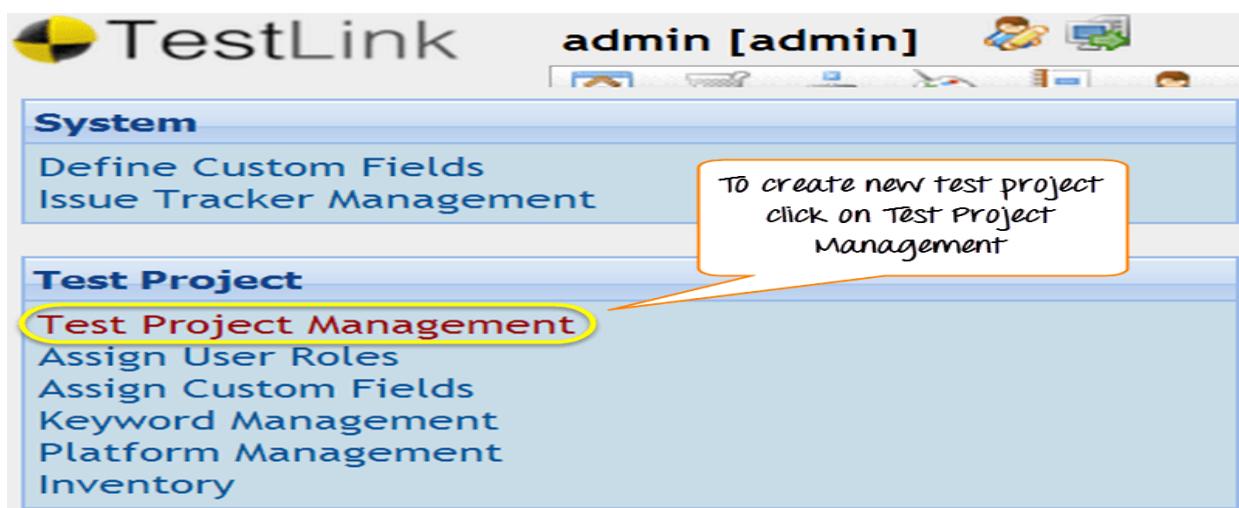
**Step 1 :** Open the Testlink home-page and enter the login details

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



### **Creating a Test Project**

**Step 1:** In the main window click on Test Project Management, it will open another window



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

The screenshot shows a table of test projects with columns for Name and Description. A callout box with the text "Click on tab 'Create' for new test project" points to the "Create" button in the top right corner of the header.

Name	Description
mre	This project is to test MRE login functionality.
HR44Transcoding	Creating a Test Project: HR44Transcoding
MTERP	MTERP
projecttime	
JIRA-TEST	

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

The screenshot shows a dialog box for creating a new test project. It includes fields for Name (Test\_demo\_Project), Prefix (Guru 99), Description (The project is to check Guru 99 home-page login functionality), and checkboxes for Requirements, Testing Priority, Test Automation (API keys), Inventory, Active Issue Tracker, and Active Public. A callout box with the text "This will create a project for GURU 99 successfully" points to the "Create" button at the bottom left.

This will create your project "Guru99" successfully.

The screenshot shows the same Test Project Management interface after creating a new project. The newly created project "Guru 99" is listed in the table. A callout box with the text "Guru 99 project is created successfully" points to the "Guru 99" row in the list.

Name	Description	Prefix	Issue Tracker	Requirement Feature	Active	Public	delete
Guru 99	The project is to check Guru 99 home-page login functionality	GU					X

## Creating a Test Plan

Test plan holds the complete information like scope of Software testing, milestone, test suites and test cases. Once you have created a Test Project, next step is to create Test plan.

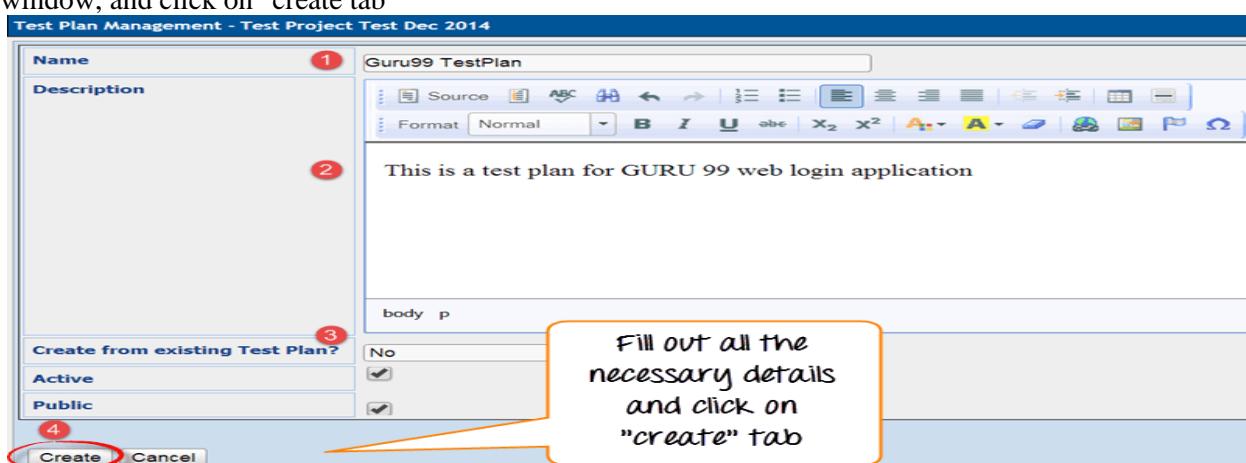
**Step 1:** From the home-page, click on Test Plan Management from home-page



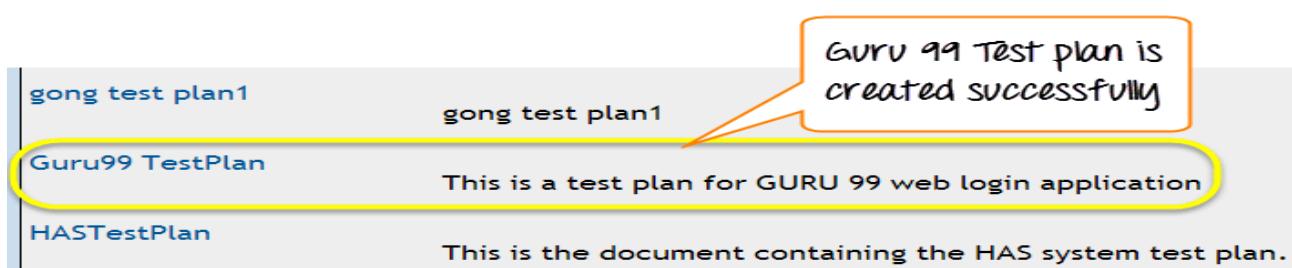
**Step 2:** It will open another page, at the bottom of the page click on a tab "Create"



**Step 3:** Fill out all the necessary information like name, description, create from existing test plan, etc. in the open window, and click on "create tab"



**Step 4:** Guru 99 Test Plan is created successfully



## Build Creation

Build is a specific release of software

**Step 1:** Click on Builds/Releases under Test Plan from the home page

## Test Plan

Test Plan Management  
Builds / Releases  
Milestones

From home-page select Build/Release for creating specific release for your software

**Step 2:** In the next window, fill all necessary details for software release and click on create to save your release

1. Enter the title name
2. Enter the description about the software release
3. Mark the check-box for status- Active
4. Mark the check-box for status- Open
5. Choose the date of release
6. Click on create button

**Create a new Build**

Title	Guru 99- sample build
Description	<p>This is a sample build for Guru 99 web application</p>
Active	<input checked="" type="checkbox"/>
Open	<input checked="" type="checkbox"/>
Release date	15/01/2015
Copy tester assignments	<input type="checkbox"/>
<b>Create</b> <b>Cancel</b>	

Once you have a release the software it, will appear like this

Title	Description
Guru 99- sample build	<p>This is a sample build for Guru 99 web application</p>

**Sample build for Guru 99 is created successfully**

## Creating Test suite

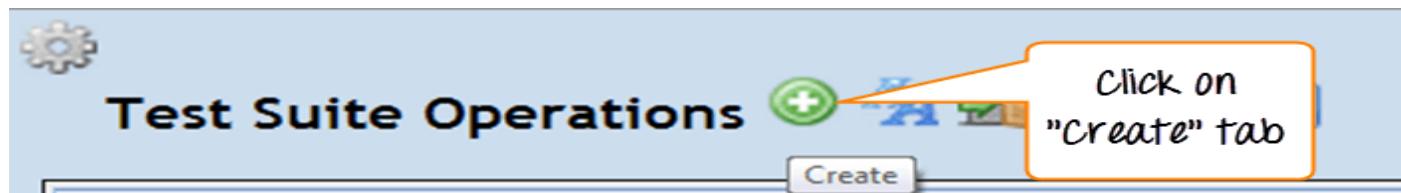
Test suite is a collection of test cases which may be testing or validating the same component. Following steps will explain how to create test suite for your project.

**Step 1:** Click on test specification option from the home page.



**Step 2:** On the right-hand side of the panel, click on the setting icon It will display a series of test operation.

**Step 3:** Click on the "create" tab for the test suite

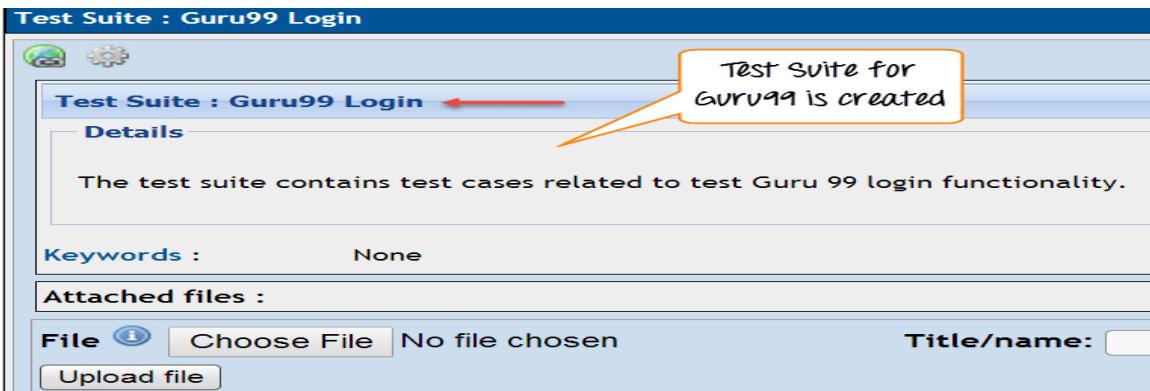


**Step 4:** Fill-up all the details for test-suite and click on save it tab.

1. Enter the test suite name
2. Enter the details about your test suite
3. Click on save button to save the details of test-suite

The screenshot shows a dialog box for creating a test suite. At the top are "Save" and "Cancel" buttons. The "Test Suite Name" field contains "Guru99 Login". The "Details" section has a rich text editor with a toolbar. A message in the editor says: "The test suite contains test cases related to test Guru 99 login functionality." The "Keywords" section lists several items: "2G Removal", "2G-Only RAT Support", "3G RAT Support", "Components", "DOM", "DOM", "HSDPA". At the bottom are "Save" and "Cancel" buttons. An orange callout bubble points to the "Save" button with the text: "Fill up all the required details for test-suite and save it".

You can see test suite for Guru 99 is created

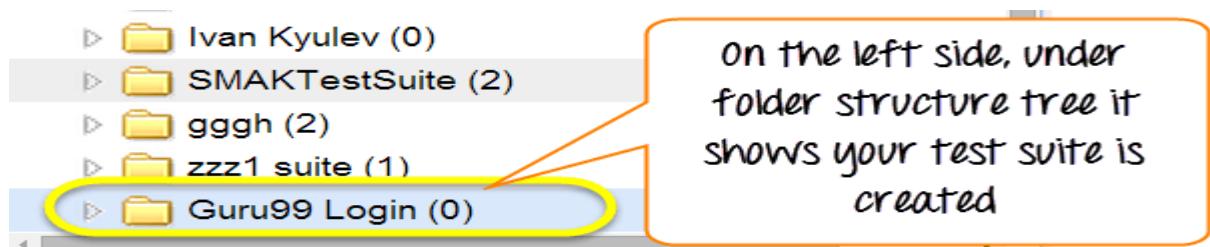


Your test suite appears on the left side of the panel under folder structure tree

### Creating a Testcase

Testcase holds a sequence of test steps to test a specific scenario with 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 folder tree structure



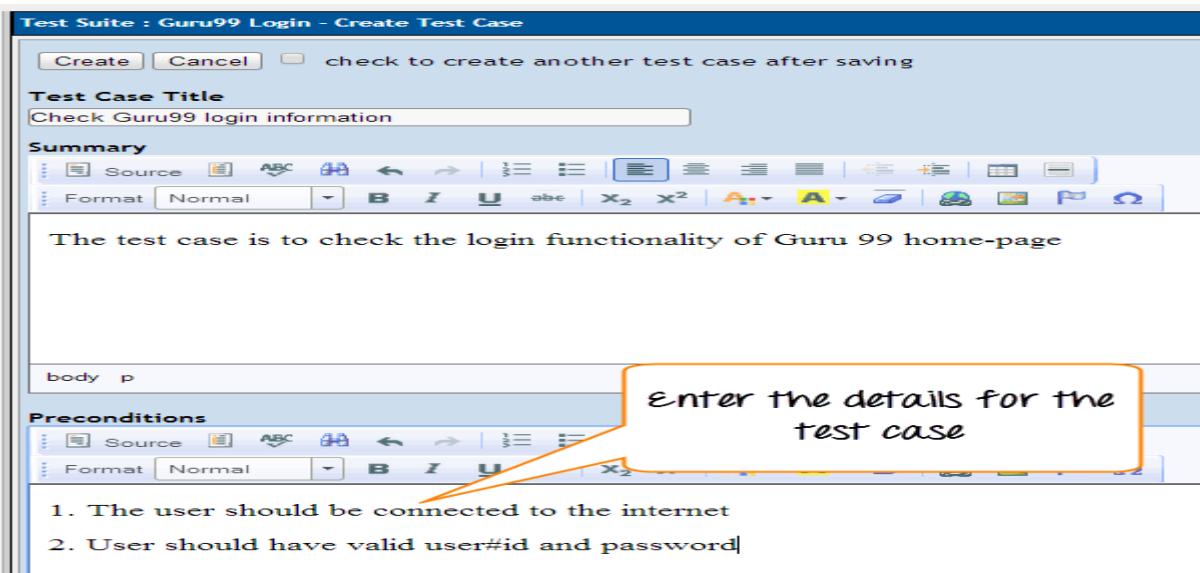
**Step 2:** Click on the setting icon in the right side panel. List of test case operations will be displayed on the right side panel



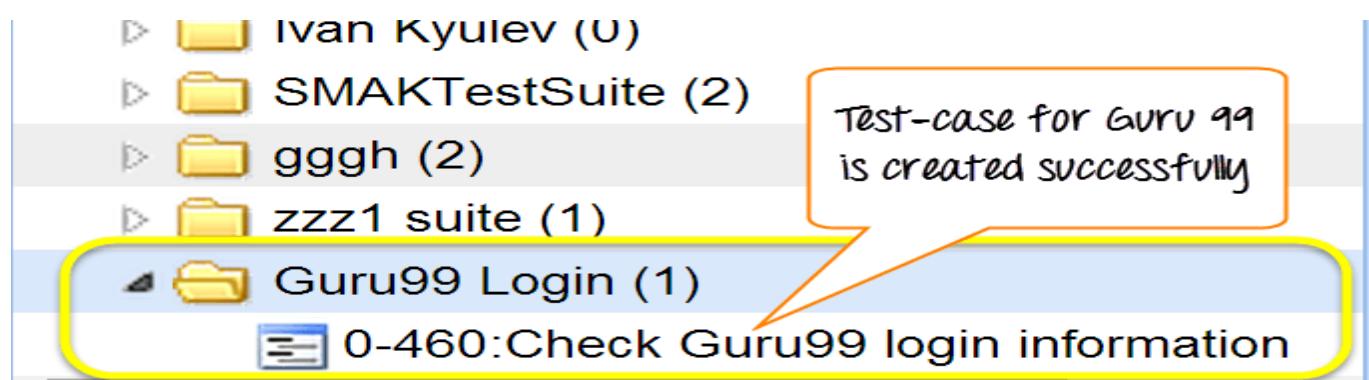
**Step 3:** New window will open, to create test cases click on create button in test-case operations



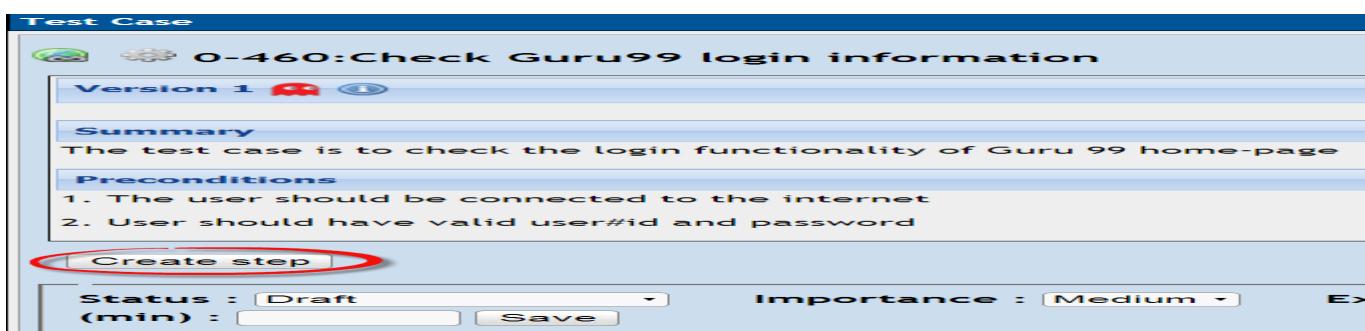
**Step 4:** Enter the details in the test case specification page



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

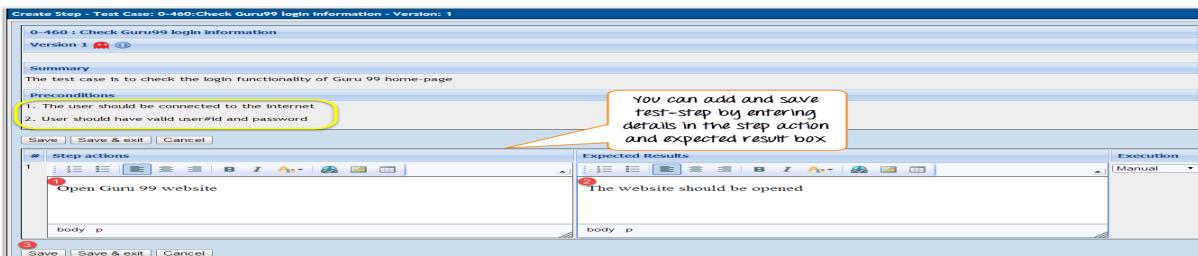


**Step 6:** Click on test-case from the folder as shown above, it will open a window. Click on "create steps" button in test case. It will open a test case step editor



**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



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

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

## Assigning test case to test plan

For test case to get execute, it should be assign 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**

New window will open, search your project "Guru99"

**Step 3)**

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

Mark the check box and then click on Add button

This will add your test case to your Test Plan.

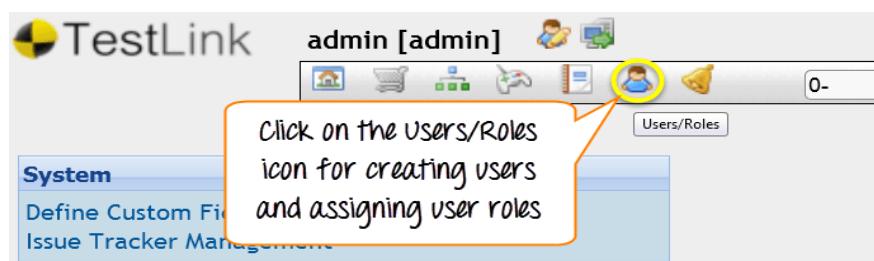
## Creating Users and Assigning Roles in TestLink

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

**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																		
<b>User details</b> <table border="1"> <tr> <td>Login</td> <td>Jamesguru</td> </tr> <tr> <td>First Name</td> <td>Jamesguru</td> </tr> <tr> <td>Last Name</td> <td>Bryan</td> </tr> <tr> <td>Password</td> <td>*****</td> </tr> <tr> <td>Email</td> <td>jamesguru@gmail.com</td> </tr> <tr> <td>Role</td> <td>Senior Test Engineer</td> </tr> <tr> <td>Locale</td> <td>English (wide/UK)</td> </tr> <tr> <td>Authentication method</td> <td>Default(DB)</td> </tr> <tr> <td>Active</td> <td><input checked="" type="checkbox"/></td> </tr> </table>					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	<input checked="" type="checkbox"/>
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	<input checked="" type="checkbox"/>																					
<input type="button" value="Save"/> <input type="button" value="Cancel"/> <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;">           Fill out all the details about the User and click on save button         </div>																						

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

jafgc	João	joaofgc@gmail.com	leader
james	james	abc@163.om	leader
Jamesguru	Jamesguru	jamesguru@gmail.com	Senior Test Engineer
jan	jan	sivaprakasam	leader

#### Step 4: Allotting test project role to the user,

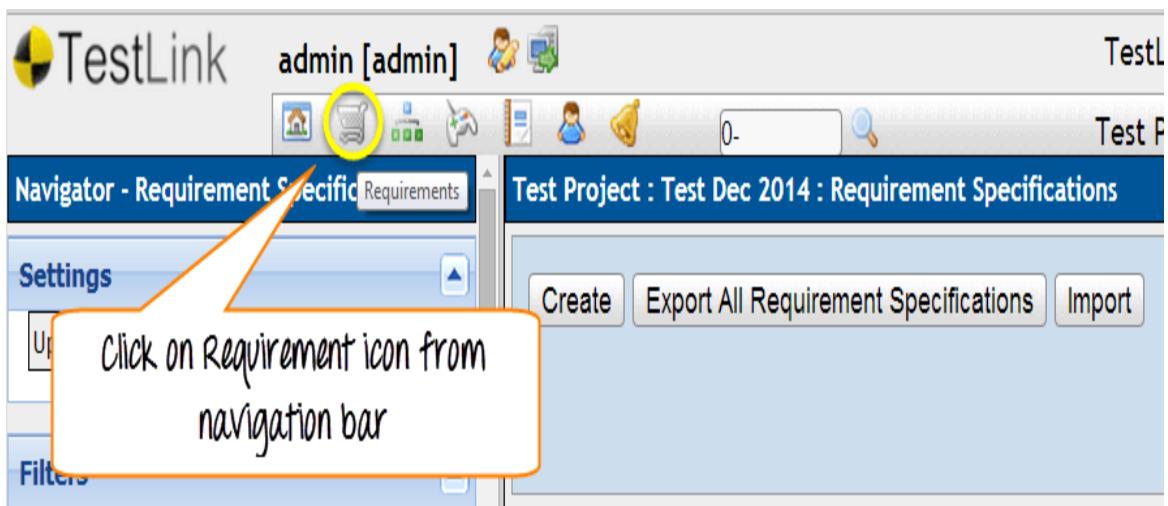
1. Click on "Assign Test Project Roles" tab
2. Choose the project name
3. Select the users role from the drop down

User Management - Assign roles

New User	View Users	View roles	1 Assign Test Project roles	Assign Test Plan roles				
Test Project 12 Guru 99 - <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;">             We assigned our test project "Guru 99" to senior test engineer-             Jamesguru           </div>								
Set roles to 3 Senior Test Engineer <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;">             We assigned our test project "Guru 99" to senior test engineer-             Jamesguru           </div>								
Show 25 entries <table border="1"> <thead> <tr> <th>User</th> <th>Test Project Role (Guru 99 -)</th> </tr> </thead> <tbody> <tr> <td>Jamesguru (Jamesguru Bryan)</td> <td>&lt;inherited&gt; Senior Test Engineer</td> </tr> </tbody> </table>					User	Test Project Role (Guru 99 -)	Jamesguru (Jamesguru Bryan)	<inherited> Senior Test Engineer
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.



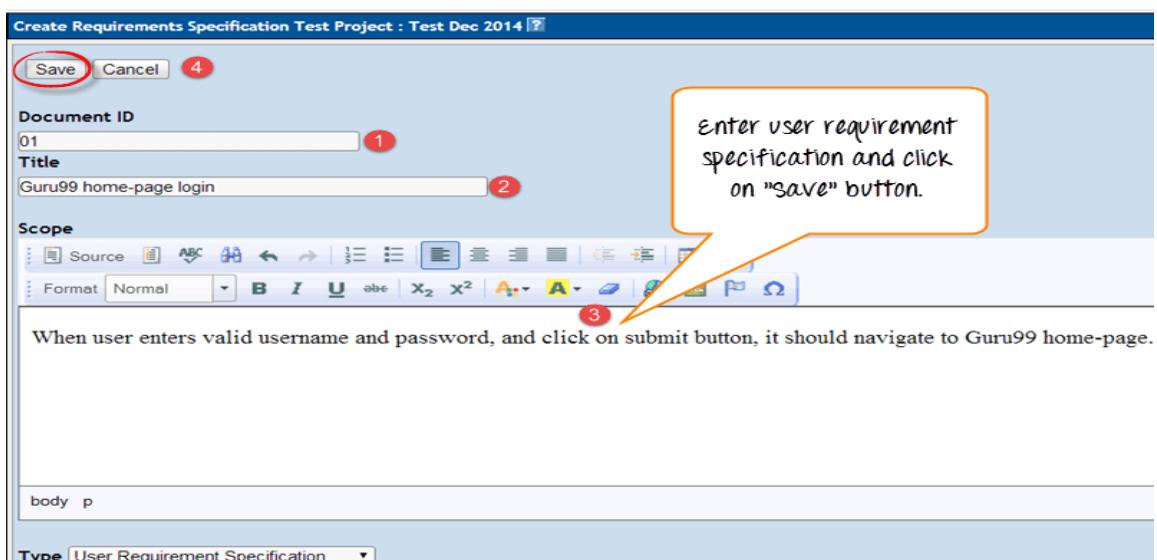
**Step 2:** From the requirement page, on the right side of the panel click on "create" button



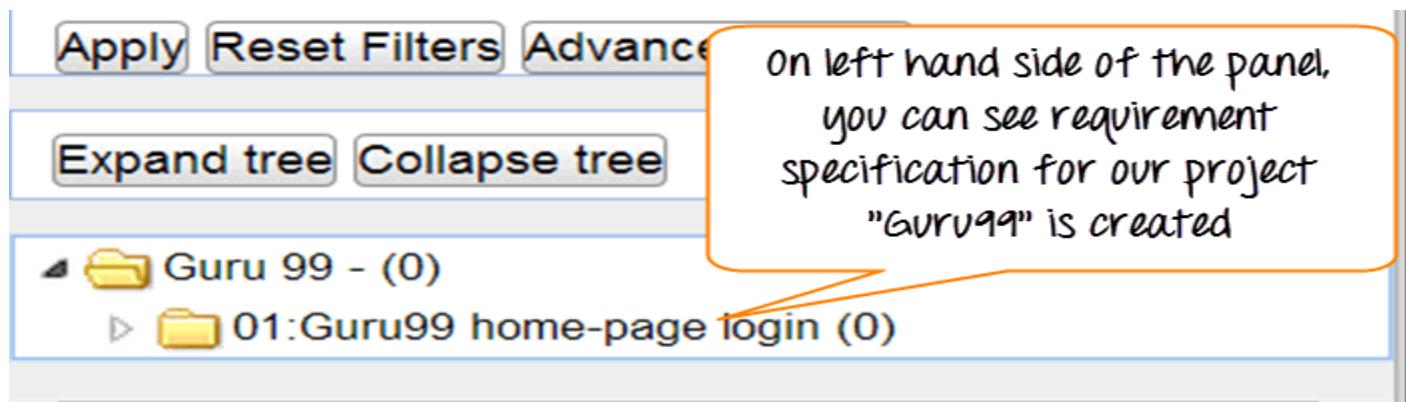
**Step 3:** A new window will open, enter all the details like

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

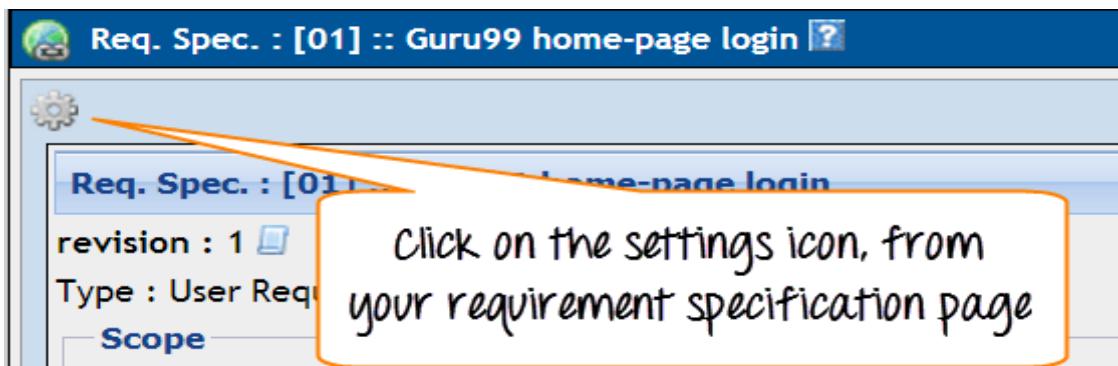
For the type, you can choose the option from the drop-down- here we chose "**User Requirement Specification**"



**Step 4:** It should create Requirement specification and displayed on the left side panel under 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

Save Cancel ⑦  
 check to create another requirement after saving

**Document ID**  
02 ①

**Title**  
Guru99 home-page login ②

**Scope**

Format Normal ③  
 Source ABC

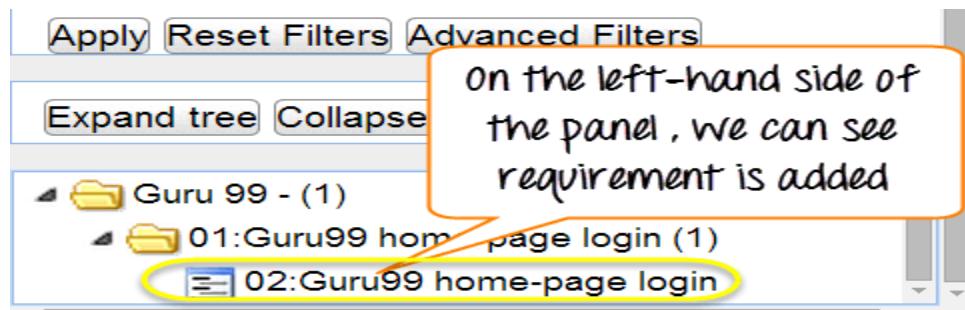
**Status** Valid ④

**Type** Use Case ⑤

**Number of test cases needed** 1 ⑥

**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 requirement is added.



### Assigning requirement to test-cases

In Testlink, Requirement can be connected to test cases. It is very crucial feature in order to track test coverage based on requirements. In test reports, you can verify which requirements **are not covered and act on them to append** in test suites for maximum test coverage

**Step 1:** From test specification section open any single test case and click on requirement icon

Test Case

GU-1:Check Guru99 login information

Version 1

Summary

Preconditions

- The user should be connected to the internet
- User should have valid user#id and password

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

Create step Resequence Steps

Status : Draft Import

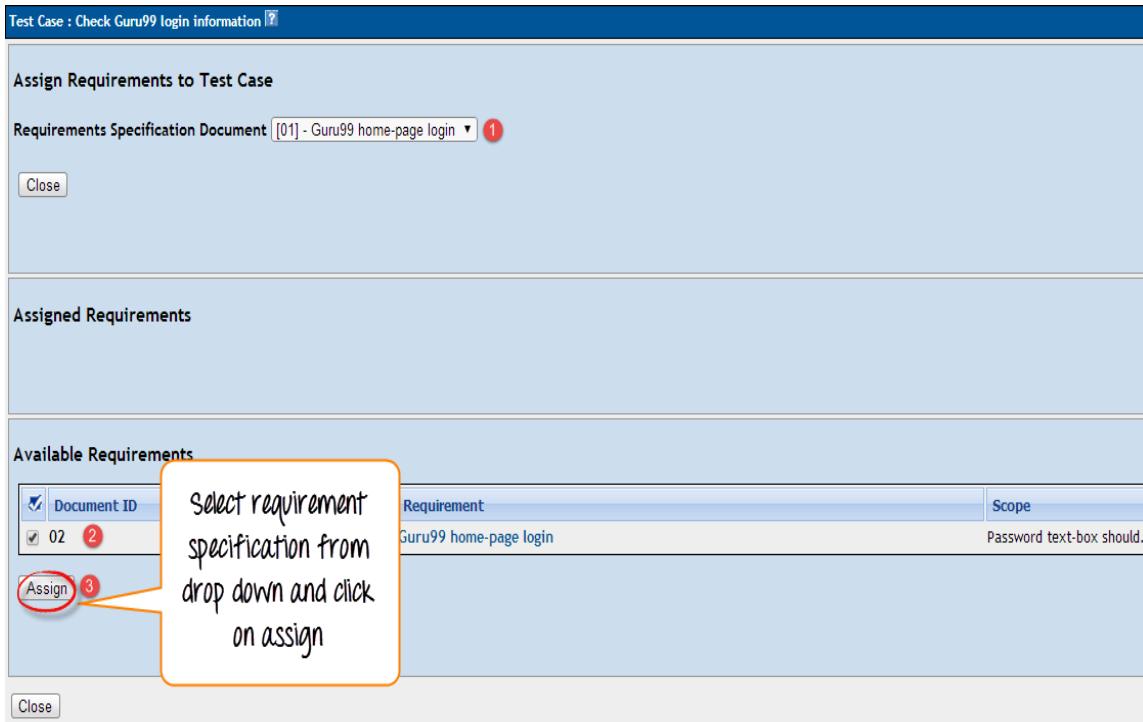
Keywords: None

Requirements: None

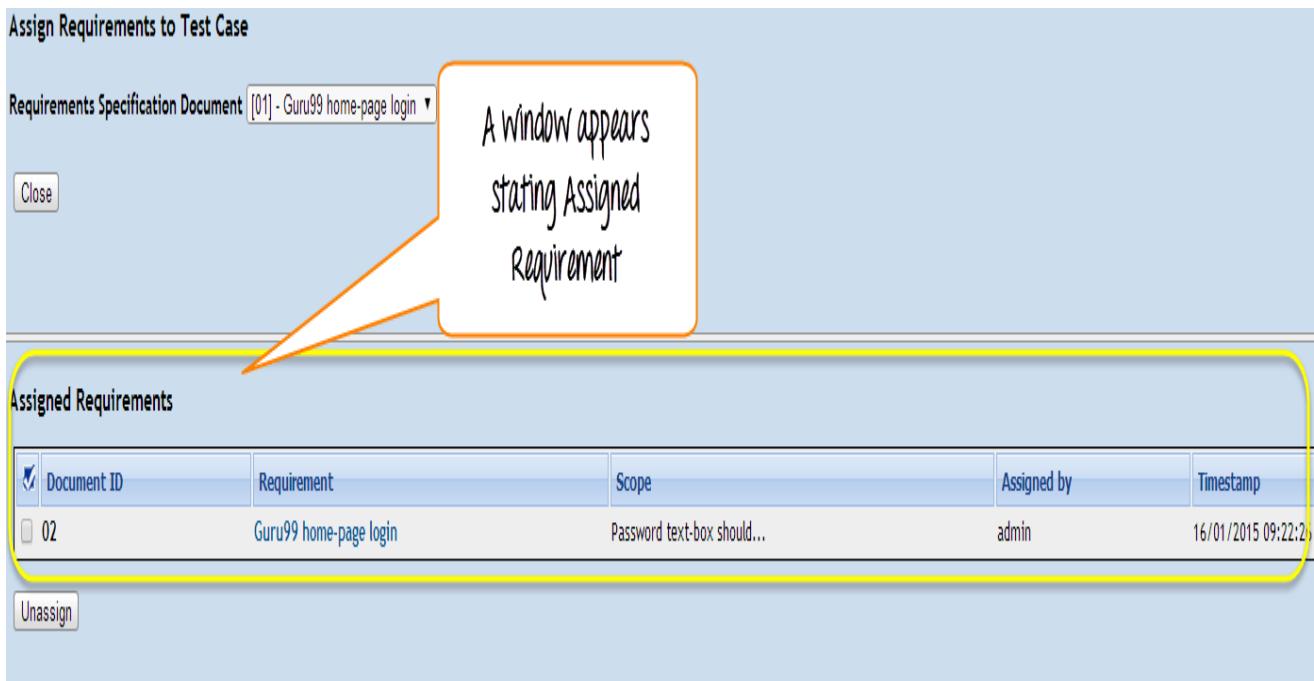
Estimated exec. (min) :  Save

**Step 2:** To assign requirements specification to test case you have to follow the following steps

1. Scroll the drop down box to select the requirements specification
2. Mark the requirement check box
3. Click on "assign" tab



After clicking on "assign" tab, a window will appear stating "Assigned Requirement."



### Executing a test case

In TestLink, we can run a test case and change execution status of a test case. Status of a test-case **can be set** to "blocked" "Passed", or "failed". Initially, **it will be in "not run"** status but once you **have** updated it, it cannot be altered to "not run" status again.

**Step 1:** From the navigation bar click on the "Test Execution" link. It will direct you to the Test Execution Panel.

To execute test case, click on test execution tab

Test Execution

admin [admin]

Exec. Context: Test Plan Guru99 TestPlan

Test Plan Guru99 TestPlan

This is a test plan for GURU 99

Build Guru 99- sample build

This is a sample build for Guru

**Step 2:** Pick the Test case you want to run from the left side panel

Apply Reset Filters Advanced Filters

Expand tree Collapse tree

Select the test case you want to execute

Guru 99 - / Guru99 TestPlan (1)(1, 0, 0, 0)

Guru99 Login (1)(1, 0, 0, 0)

GU-1:Check Guru99 login information

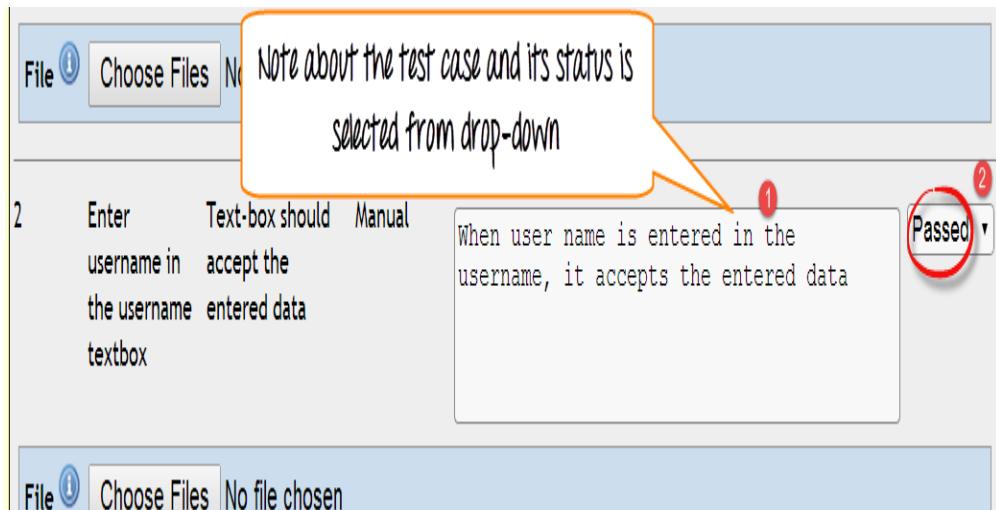
**Step 3:** Once you have selected the test cases, it will open a window.

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

#	Step actions	Expected Results	Execution	Execution notes	Result
1	Open Guru 99 website	The website should be opened	Manual	You can enter the notes related to execution and select the status from the drop down for the test case	
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		

**Step 4:** Follow the following steps

1. Enter the notes related to test case executed
2. Select its status



**Step 5:** On the same page, you have to fill similar detail about the execution of test-case. Fill the details, select the status and then click on "save execution".

The screenshot shows the "Test Results on Build Guru 99- sample build" page. It includes a navigation bar with links like "Print", "Show complete execution history", and "Import Results (XML)". The main content area displays a tree view of test plans, builds, and suites. A yellow box highlights the "Latest execution (any build)" section, which shows a green bar with the date "Date : 16/01/2015 10:35:06 - Tested by : admin - Build : Guru 99- sample build - Status : Passed". Below this, another section titled "Latest execution (current build) - Build : Guru 99- sample build" shows a table with columns: Date, Build, Tested by, Status, Exec (min), Version, and Run mode. The table contains one row with the same information as the green bar. A yellow box also highlights the "Notes" section, which contains the text "Test executed successfully in Firefox 12, in windows 8 platform". A callout bubble points to this notes section with the text "A window will appear showing latest execution of your test case".

## Generating Test Reports

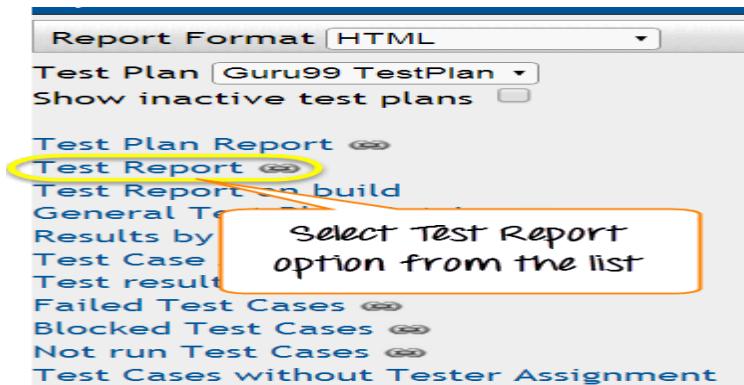
Test link supports various test report formats like

- HTML
- MS Word
- MS excel
- OpenOffice Writer
- OpenOffice calc

**Step 1:** From the navigation bar, click on Test Reports option

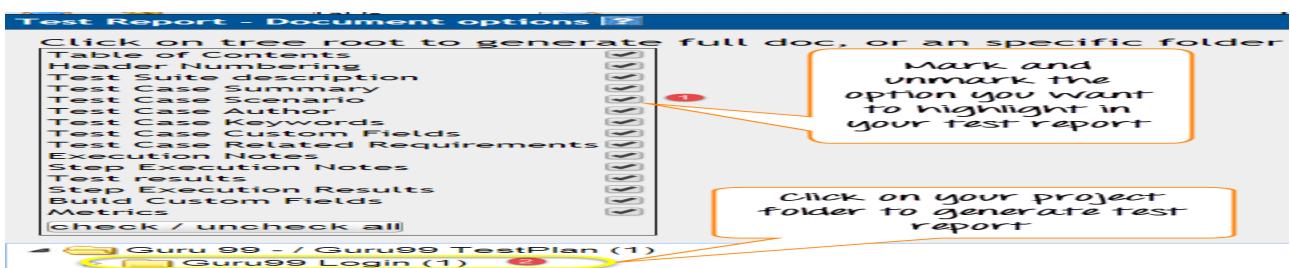


**Step 2:** From the left side panel, select "Test Report" link



**Step 3:** To generate a report follow the following steps

1. Mark and unmark the option you want to highlight in your test report
2. click on your project folder



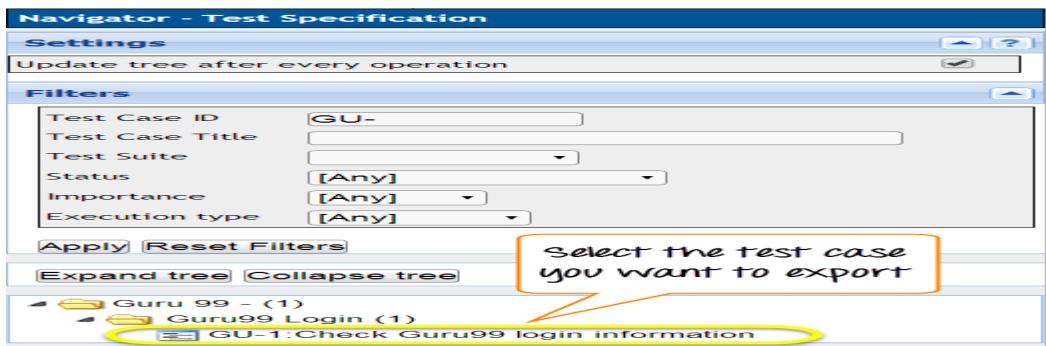
The test report will look like this

1.1. Test Suite : Guru99 Login					
The test suite contains test cases related to test Guru 99 login functionality					
<b>Test Case GU-1: Check Guru99 login information</b>					
Author: admin					
Preconditions:					
1. The user should be connected to the internet					
2. User should have valid user/id and password					
#	Step actions:	Expected Results:	Execution notes:	Execution Status:	
1	Open Guru 99 website	This website should be opened			
2	Enter user name in the username textbox	Text-box should accept the entered data			
3	Enter password in the password-text box	Text-box should accept the entered data			
4	Click on "sign in" button	Login should success and navigate to the home page			
Execution type: Manual					

### Export Test case/ Test Suite

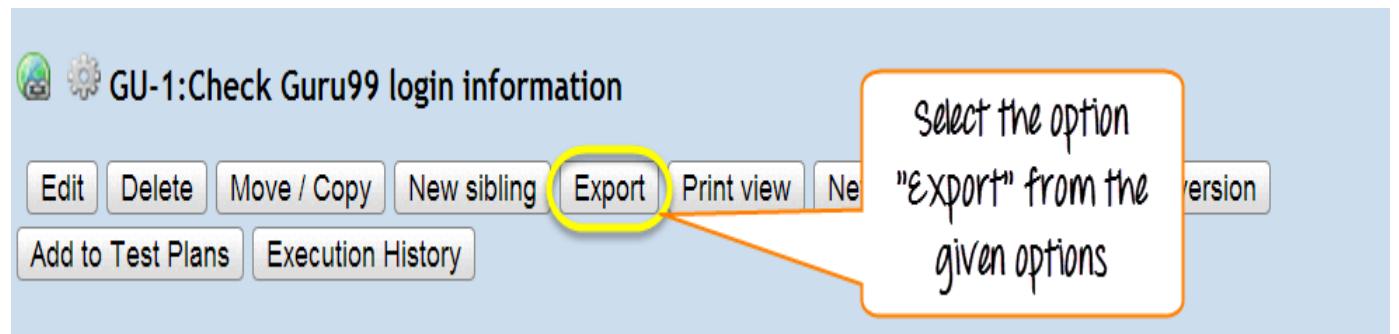
Testlink provides the features to export test projects/test suites in your Testlink and then you can import them into another Testlink project on different server or system. In order to do that you have to follow the following step

**Step 1:** Choose the test case you want to export in the Test specification page

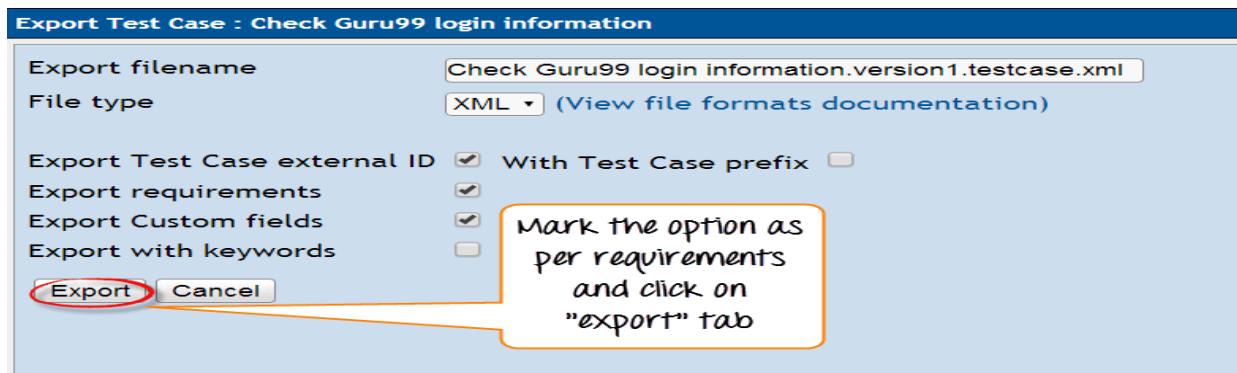


**Step 2:** Now on the right-hand side of the panel click on the setting icon, it will display all the operations that can be performed on the test case.

**Step 3:** Click the "export" button



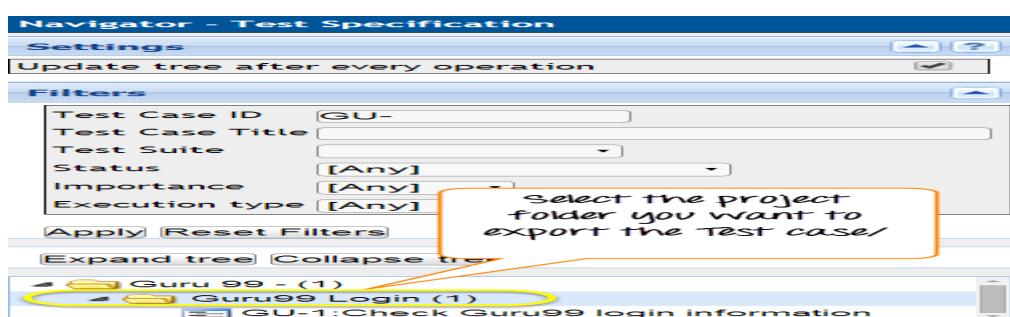
**Step 4:** It will open another window, mark the option as per requirement and click on the export tab



Following XML is generated

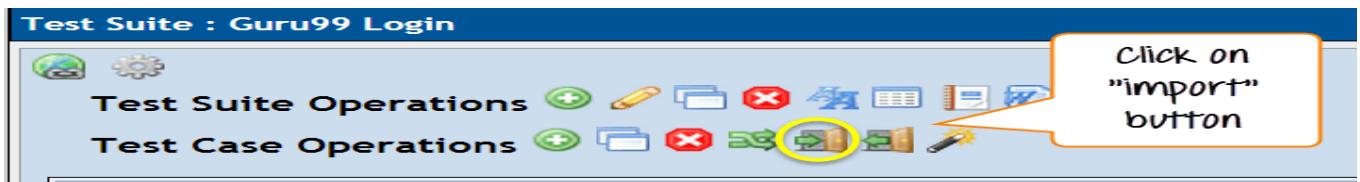
### Importing Test case/ Test suite

**Step 1:** Select the Test suite folder inside which you want to import the test case



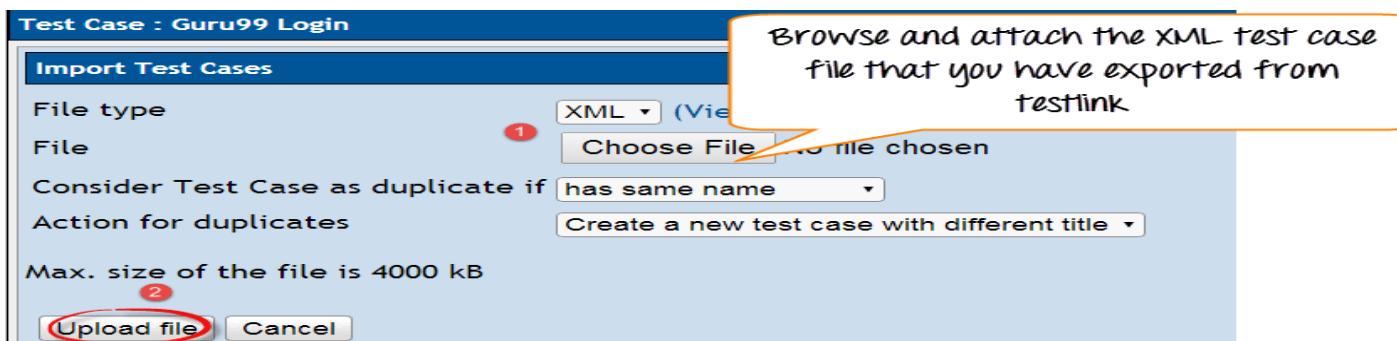
**Step 2:** Click on the setting icon  on the right hand-side of the panel, it will display all the operations that can be executed on the test suite/test case

**Step 3:** Click on the import button in the test case operations list as

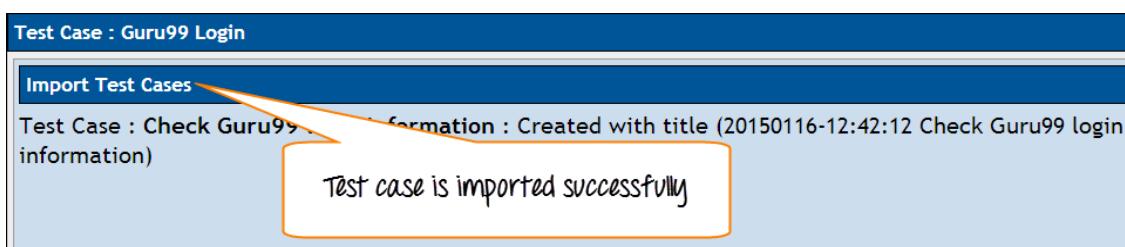


**Step 4:** Browse and attach the xml test case file that you have exported from test link and click on upload button.

1. Use the browse option to attach the XML test case file that you have exported from testlink
2. Click on upload file



When you upload a file, it will open window stating import test cases



**Step 5:** Test case will be uploaded and displayed on the right-hand side of the panel

