

Ex.No.1 CAUSE & EFFECT TESTING FOR A TRIANGLE PROGRAM

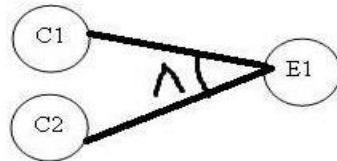
Aim:

To perform cause-effect graph testing to find a set of test cases for triangle program.

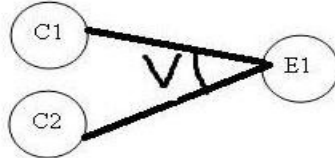
Cause-Effect Graph Description

The cause effect graph testing is a method of black box testing which logically test only the inputs and outputs without knowing the internal structure (code). Cause and effect graph is a dynamic test case writing technique. Here causes are the input conditions and effects are the results of those input conditions.

AND – For effect E1 to be true, both the causes C1 and C2 should be true

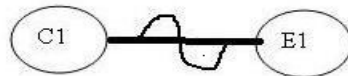


OR – For effect E1 to be true, either of causes C1 OR C2 should be true

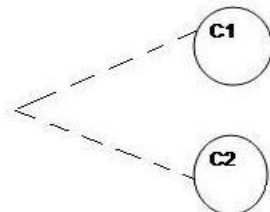


NOT – For Effect E1 to be True, Cause C1 should be false

or ||



MUTUALLY EXCLUSIVE – When only one of the causes will hold true.



Cause-Effect Specification:

To write a program that takes three positive integers a, b, c as input. To determine if the integers represent the three sides of triangle and to display the type of triangle.

Cause-Effect Condition:

The input condition for the cause-effect testing graph will have equal cause and effects. The type of triangle to be displayed and flag should be set it.

- i) Set it to 1 = Scalene triangle
- ii) Set it to 2 = Isosceles triangle
- iii) Set it to 5 = Equilateral triangle
- iv) Set it to 4 = Not a triangle
- v) **Test cases for the above specification**

Every type of triangle has specific conditions to be satisfied. The different conditions existed for the triangle type follows,

- i) Triangle $(a < (b+c)) \ \&\& \ (b < (a+c)) \ \&\& \ (c < (a+b))$
- ii) Not a triangle $(a==b) \wedge (a==c) \wedge (b==c) \ \&\& \ !((a==b) \ \&\& \ (a==c))$
- iii) Equilateral triangle $(a==b) \ \&\& \ (b==c)$
- iv) Scalene triangle $(a!=b) \ \&\& \ (a!=c) \ \&\& \ (b!=c)$

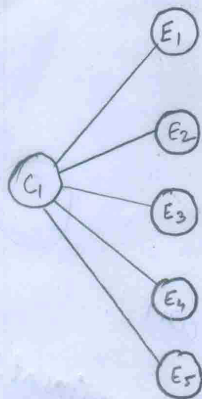
Test case table:

Test id	Test inputs			Expected output	Flag
	a	b	c		
1	3	2	3	Is a triangle	
	3	3	4		
	2	4	2		
2	7	2	3	Not a triangle	1
	4	6	1		
	2	2	5		
3	4	4	2	Isosceles triangle	2
4	7	7	7	Equilateral triangle	3
5	2	7	9	Scalene triangle	4

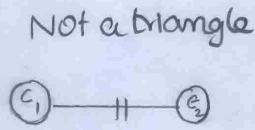
Decision Table:

Cause/effect	E1	E2	E3	E4	E5
C1	1	0	1	1	1
C2	1	0	1	0	0
C3	1	0	0	1	0
C4	1	0	0	0	1

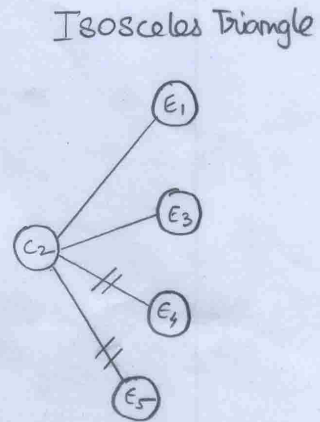
CAUSE AND EFFECT GRAPH



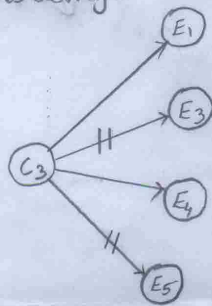
Isosceles triangle



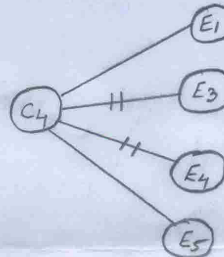
Not a triangle



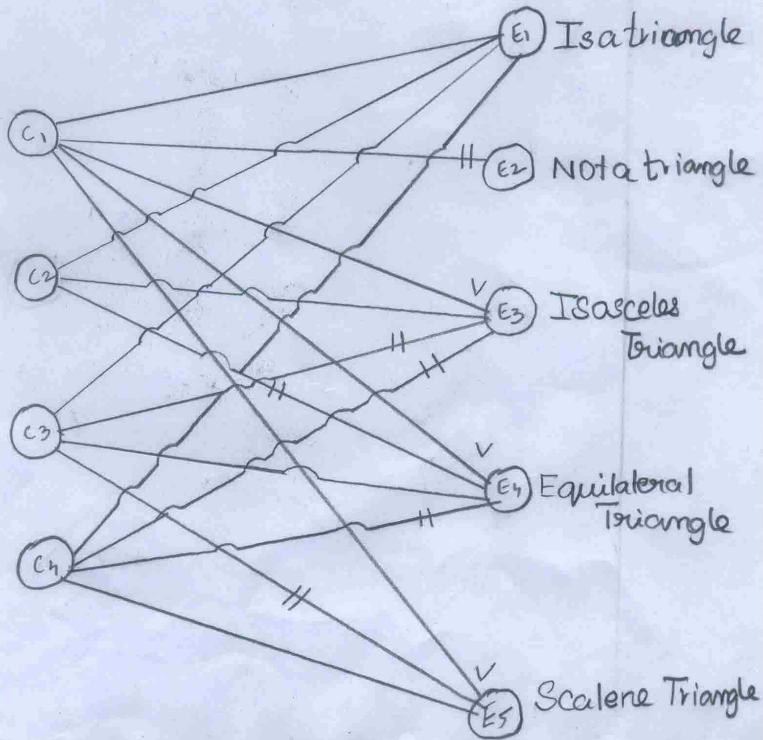
Isosceles Triangle



Equilateral triangle



Scalene triangle



Isosceles triangle

Not a triangle

Isosceles Triangle

Equilateral Triangle

Scalene Triangle

Program:

```
import java.io.*;
import java.util.*;
public class triangle {
public static void main(String args[])
{
    int a,b,c;
    boolean isatriangle;
    int flag=0;
    Scanner scan=new Scanner(System.in);
    System.out.println("TRIANGLE CLASSIFICATION PROBLEM");
    System.out.println("Enter a, b, c:");
    a= scan.nextInt();
    b= scan.nextInt();
    c= scan.nextInt();
    if((a<(b+c))&&(b<(c+a))&&(c<(b+a)))
    {
        isatriangle=true;
    }
    else
    {
        isatriangle=false;
    }
    if(isatriangle)
    {
        {
            if((a==b)^(a==c)^(b==c)&&!((a==b)&&(a==c)))
            {
                flag=1;
                //System.out.println("ISOSCELES");
            }
        }
        if((a==b)&&(a==c))
        {
            flag=2;
            //System.out.println("EQUILATERAL");
        }
        if((a!=b)&&(a!=c)&&(b!=c))
        {
            flag=3;
            //System.out.println("SCALENE");
        }
    }
    else
    {
        flag=4;
        //System.out.println("NOT A TRIANGLE");
    }
}
```

```
    }  
  
    if(flag==1)  
    {  
        System.out.println("iso");  
    }  
    if(flag==2)  
    {  
        System.out.println("equilateral");  
    }  
    if(flag==3)  
    {  
        System.out.println("scalene");  
    }  
    if(flag==4)  
    {  
        System.out.println("not tri");  
    }  
    }  
}
```

Output:

Enter the value: 2 2 2

Equilateral triangle

Enter the value: 2 2 3

Isosceles triangle

Enter the value: 3 2 4

Scalene triangle

Enter the value: 7 4 1

Not a triangle

Result:

EX: NO: 2

BOUNDARY VALUE ANALYSES FOR A SOFTWARE UNIT

AIM

To perform boundary value analysis for average of 25 floating point numbers

EQUIVALANCE CLASS PARTITIONING DESCRIPTION

- Equivalence partitioning (EP) is a specification-based or black-box technique.
- **Equivalence partitioning** is a software testing technique that divides the input data of a software unit into partitions of equivalent data from which test cases can be derived
- Equivalence classes are identified based on set of valid or invalid state for each i/p conditions
 - Specific numeric value
 - A range of values
 - Boolean conditions(must be)
 - Set of related values

The input conditions for the software unit:

- Range of values: (0 to 5000, <0, >5000)
- Number of values: (25, >25, <25)
- Must be condition: (UB>LB, LB<UB)

BOUNDARY VALUE ANALYSIS DESCRIPTION

- **Boundary Value Analysis** is a black box test design technique where test case are designed by using boundary values
- Boundary value analysis (BVA) is based on testing at the boundaries between partitions.
- It have both valid boundaries (in the valid partitions) and invalid boundaries (in the invalid partitions).
- A boundary value is any input or output value on the edge of an equivalence partition

SPECIFICATION

The software unit computes the average of 25 floating point numbers that lie on or between values which are positive values from 1.0 to 5000.0.the boundary values and the number are input to unit

CONDITIONS

- The input condition for the boundary value analysis for the software unit are,
- The boundary values and numbers to average are input
- Upper bound must be generated than lower bound
- If invalid set of values are input for boundaries error message appears and users is reported
- The average sum are output of unit value lie within the boundary

TEST INPUTS

1. RANGE CONDITIONS

- EC1->1.0-5000(valid)
- EC2->0-5000(invalid)
- EC3->1-5001(invalid)

2. NUMBER OF VALUES

- EC4->25(valid)
- EC5-> <24(invalid)
- EC6-> >25(invalid)

3. MUST BE

- EC7-> UL>LL(valid)
- EC8-> LL>UL(invalid)

TABLE FOR EQUIVALENCE CLASS PARTITIONING AND BOUNDARY VALUE ANALYSIS

TEST	TEST INPUT	VALID EC	INVALID EC	OUTPUT
TC1	Total inputs:25 1.0-5000.0 LB:1.0 UB:5000.0 2.3,3.4,50.7,45.8,67.6,160.4,182.3, 190.01,503.4,600.02,456.8,1034.31 1212.4,1313.6,1454.7,2990.7,3500 3956.7,4212.3,4354.6,4578.6,4679 4764.7,4845.3,4868.4	EC1(UB), EC7(LB) EC4	–	SUM :121709.98 AVG:4868.399
TC2	Total inputs:25 0.0-5000.0 LB:0.0 UB:5000.0	–	EC2(BUB)	Error Message Stating Lower Bound can't be zero
TC3	Total inputs:25 LB:1.0 UB:5000.0 1-5001	–	EC3(AUB)	Error Message Stating Upper Bound can't be zero
TC4	Total inputs:20 2-5000	EC1(AUB)	EC5	Error Message Stating Enter 25 values
TC5	Total inputs:26	EC1(BUB)	EC6	Error Message Stating Enter 25 values
TC6	Total inputs:25 UL:100;LL:5000	EC1	EC8	Error Message Stating that Upper Bound to be greater than Lower Bound

PROGRAM

```
package com.BoundaryValue;
import java.util.Scanner;
public class BoundaryValue
{
    public static void main(String[] args)
    {
        float Average=0;
        int Sum=0;
        Scanner Scan=new Scanner(System.in);
        System.out.println("Enter The No of Values to be added");
        int sizeval=Scan.nextInt();
        if(sizeval!=25)
        { System.out.println("The range is not 25...Please give only 25 numbers");}
        else
        {
            int[] base=new int[sizeval];
            System.out.println("Enter The Upper Limit for the Range");
            int upper=Scan.nextInt();
            if (upper>5000)
            {
                System.out.println("Please Enter Correct Value... Upper limnit maximum value is '5000.0' ");
            }
            System.out.println("Enter The Lower Limit for the Range");
            int lower=Scan.nextInt();

            if((lower>upper)||lower==0)
            {
                System.out.print("Please Enter Correct Value ... Upper value must be greater than lower value
                                and value must not be zero");
            }
            else
            {
                System.out.print("Enter The Values");
                {
                    for(int limit=0;limit<sizeval;limit++)
                    {
                        int value = Scan.nextInt();
                        if(value >=lower && value<=upper)
                        {
                            for(int k=0;k<sizeval;k++)
                            {
                                base[k]=value;
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```
else
{
System.out.println("Bad Number value Out Of Range...Please Try Again");}
}
for(int k=0;k<sizeval;k++)
{
Sum = Sum + base[k];
}
System.out.println("Sum of 25 numbers is "+Sum);
Average=Sum/sizeval;
System.out.println("The Average of 25 numbers Is "+Average);
}
}
}
}
}
```

OUTPUT

```
Enter the no of values:25
Enter the upper limit range:2000
Enter the values:1001,1002,1003,1004,1005,
                1007,1008,1009,1010,1011,
                1012,1013,1014,1015,1016,
                1017,1018,1019,1020,1021,
                1022,1023,1024,1025
Sum of 25 numbers is 25625.0
The average of 25 numbers is 1025.0
```

RESULT

Ex.No. 3 CYCLOMATIC COMPLEXITY FOR BINARY SEARCH

AIM:

To draw Control flow graph and calculate its cyclomatic complexity

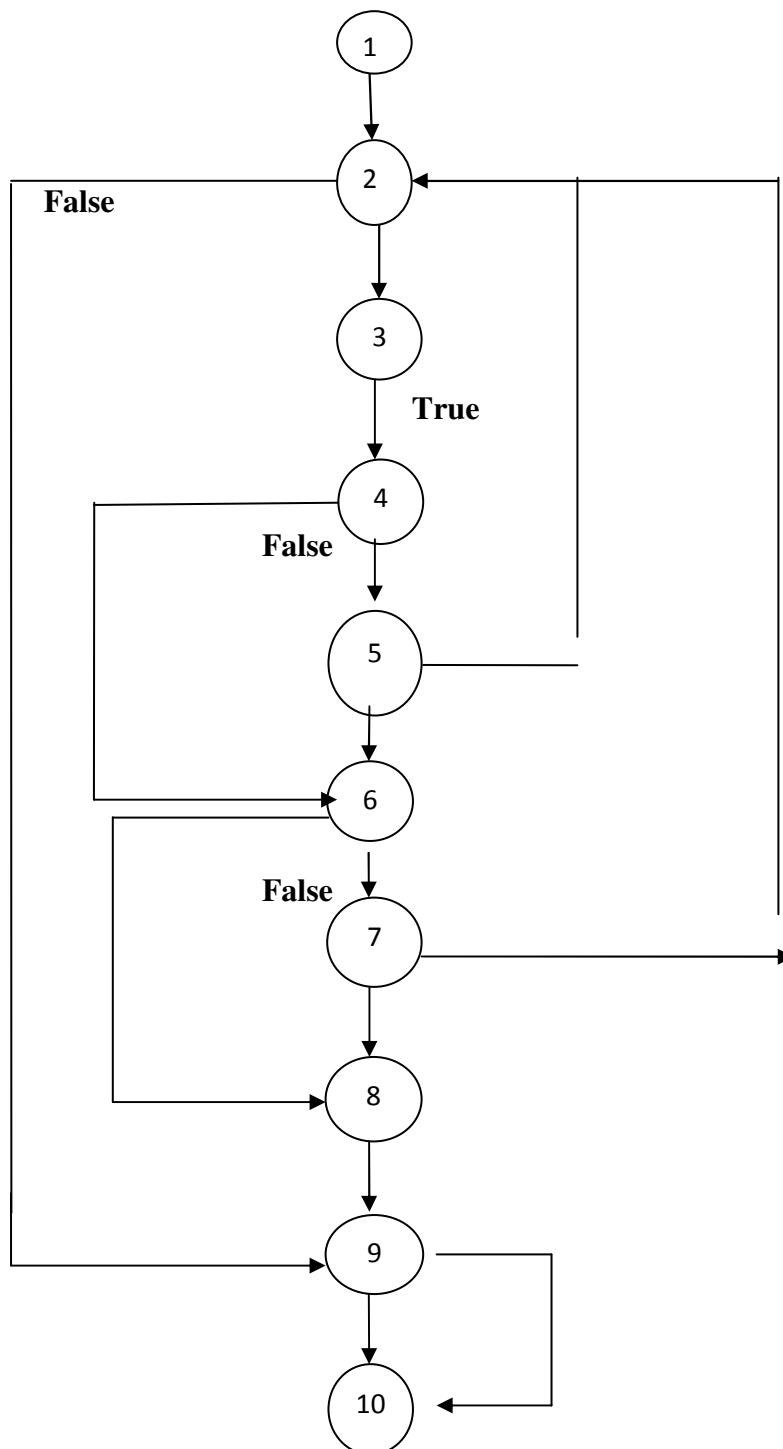
PSEUDO CODE:

```
int binsearch (int x, int v[], int n)
{
1.  int low, high, mid;
    low = 0;
    high = n-1;
2.  while (low <= high) {
3.    mid = (low+high)/2
4.    if (x < v[mid]
5.      high = mid-1;
6.    else if (x > v[mid])
7.      low = mid+1;
      else /* found match */
8.    return mid;
    }
9.  return -1; /* no match */
```

STEPS:

1. Draw the control flow graph
2. Calculate McCabe's cyclomatic Complexity
3. Identify the Independent paths
4. Calculate Region

Control flow representation:



$$V(G)=E-N+2=12-10+2=4$$

Region =4

No of Predicates+1=3+4

Independent Paths

P1=1,2,9,10

P2=1,2,3,4,5,2,9,10

P3=1,2,3,4,6,7,2,9,10

P4=1,2,3,4,6,8,10

Result:

EX NO:4**DATA FLOW TESTING FOR GREGORIAN CALENDAR**

AIM: To write a program to check whether the given year in Gregorian calendar is leap year or not.

DATA FLOW TESTING DESCRIPTION:

The testing method based on selecting paths through the program's control flow in order to explore sequences of events related to the status of variables or data objects. It focuses on points at which variables receive values and the points at which these values are used.

P (Predicate)- use: It is described for a variable that indicates its role in a predicate.

C (Computational)-use: It is used to indicate the variable's role as a part of computation

HINT TO DEFINE P-use and C-use:

- Consider a variable defined before the loop, then check how the variable is used inside the loop and outside the loop.
- Consider a variable defined inside the loop, then check the use of variable inside or outside the loop.

DATA FLOW GREGORIAN SPECIFICATION:

- i. $C1(T)=(T=\{1,101,1001,10001\})$
- ii. $C2(T)=(T=\{t|1995 \leq t \leq 2005\})$
- iii. $C3(T)=(T=\{t|1895 \leq t \leq 1905\})$
- iv. $C4(T)=(T=\{t\}^{\wedge t \in \{400,800,1200,1600,2000,2400\}})$
- v. $C5(T)=(T=\{t,t+1,t+2,t+3,t+4\}^{\wedge t \in \{100,200,300,400,500\}})$
- vi. $C6(T)=(T=\{t,t+1,t+2,\dots,t+399\}^{\wedge t \in D})$
- vii. $C7(T)=(T=\{t1,t2,t3\}^{\wedge t1,t2,t3 \in D})$

viii.

PROGRAM:

```
1    import java.util.Scanner;
2    public class lecture {
3    public static void main(String [] args) {
4    boolean loop=true;                (loop, def)
5    Scanner console =new Scanner (System. in);    (console, def)
6    while(loop) {                    (loop, p-use)
7    System.out.print("Enter the year");
8    int year=console.nextInt();        (year, def) (console,
c-use)
9    System.out.println( "The year is a leap year: "+ leapYear(year) ); (year, c-use)
10   System.out.print( "again?: " );
11   int again = console.nextInt();        (again, def)
(console, def)
12   if (again == 2) {                (again, p-use)
13   loop=false;                    (loop, def)
14   System.out.println("End of program"); }
15   public static boolean leapYear ( int year){
16   boolean leaped = false;        (leaped, def)
17   if (year%4==0){                (year, def)
18   leaped = true;                (leaped, def)
19   if(year>1582){                (year, p-use)
20   if (year%100==0&&year%400!=0){    (year, p-use)
21   leaped=false;                (leaped, def)
    }}}
```

```

22    return leaped;

    }}

```

(leaped, c-use)

OUTPUT:

Enter the year: 1000

The year is a leap year: false

again? :1

Enter the year:1996

The year is a leap year: true

again?:2

End of the program.

DATA FLOW TABLE:

TABLE FOR LEAPED

PAIR ID	DEF	USE
1	16	18
2	16	21
3	16	22

TABLE FOR LOOP

PAIR ID	DEF	USE
1	4	6
2	4	13

TABLE FOR AGAIN

PAIR ID	DEF	USE
1	11	12

TABLE FOR YEAR

PAIR ID	DEF	USE
1	8	17
2	8	19
3	8	20
4	22	30

TABLE FOR CONSOLE

PAIR ID	DEF	USE
1	5	8
2	5	11

TEST CASES:

- **CASE 1:** $C1(T) = (T = \{1, 101, 1001, 10001\})$

YEAR= 1, 101, 1001, 10001. (*All are not leap year*).

PAIR ID	VARIABLE
1	Year
1,2	Loop
1	Console
1	Again
1,3	Leaped

- **CASE 2:** $C2(T) = (T = \{t \mid t \in \{400, 800, 1200, 1600, 2000, 2400\}\})$

YEAR= 400, 800, 1200, 1600, 2000, 2400 (*All are Leap year*)

PAIR ID	VARIABLE
1,2,3	Year
1,2	Loop
1,2	Console
1	Again
1,2,3	Leaped

- **CASE 3:** $C3(T) = (T = \{t, t+1, t+2, t+3, t+4\} \mid \{100, 200, 300, 400, 500\})$

YEAR=
 { 100, 101, 102, 103, 104, 200, 201, 202, 203, 204, 300, 301, 302, 303, 304, 400, 401, 402, 403, 404
 500, 501, 502, 503, 504 }.

(*All the “t” and “t+4” are leap years and rest is not leap year*)

RESULT:

AIM:

To perform state based testing for an assembler.

State based Testing Description:

To analyze the different abstract states that a class can take. The state of an object is generally defined as a constraint on the values of its attributes. According to the state of the object, calls to certain methods may or may not be valid, or the method's behavior may change.

the process of using state-based testing techniques is as follows:

1. Define the steps.
2. Define the transitions between states.
3. Define test scenarios.
4. Define test values for each state.

State based Testing for an Assembler specification:

Suppose you were developing a simple assembler whose syntax can be described as follows :

<statement_ :: = <label field><op code><address>

<label field> :: = ‘none’ | <identifier> :

<op code> :: = MOVE | JUMP

<address> :: = <identifier> | <unsigned integer>

A stream of tokens is input to the assembler. The possible states for such an assembler are:

S1, prelabel; S2, label; S3, valid op code; S4, valid address; S5, valid numeric address. Start, Error, and Done. A table that describes the inputs and actions for the assembler is as follows:

Inputs Actions

no more tokens A1: Put the label in the symbol table.

Identifier A2: Look up the op code and store its binary value in op code field.

MOVE, JUMP A3: Look up symbol in symbol table and store its value in address field.

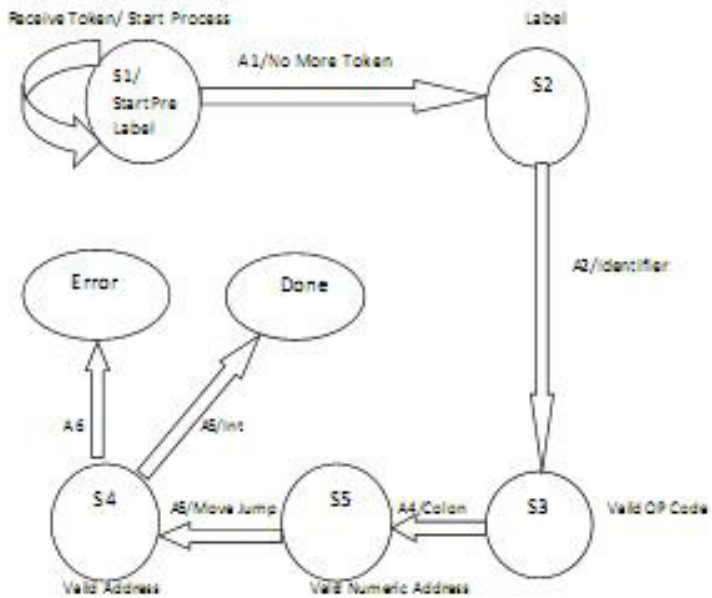
colon A4 : Convert number to binary, and store that value in address field.

Integer A5: Place instruction in the object module, and print a line in the listing.

A6: Print error message and put all zeroes in the instruction.

Using this information and any assumptions you need to make, develop a state transition diagram for the assembler. From the state transition diagram develop a set of test cases that will cover all of the state transitions. Be sure to describe the exact sequence of inputs as well as the expected Sequence of state changes and actions.

State Based diagram:



State Based Table:

Input	S1	S2	S3	S4	S5
Receive Token	Start the process				
No more Token	S2(A1)				
Identifier		S3(A2)			
Colon			S5(A4)		
Move, Jump					S4(A3)
Int				Done(A5)	
Int				Error(A6)	

Ex.No.5 STRESS TESTING OF A MAP AIDED VEHICLE TRACKING AND SCHEDULING SYSTEM

AIM:

To develop a system for stress testing a vehicle tracking and scheduling system and Test it using LoadRunner.

PROBLEM STATEMENT:

The Vehicle Tracking and Scheduling System is a sample application. This system helps to check the stress handled by the application by giving the username and quantity for the truck by number of users.

1. PROBLEM ANALYSIS:

1.1 Purpose:

The main objective of the project is to illustrate the stress handled by the vehicle tracking and scheduling application when number of user accessing it.

1.2 Document Conventions:

In the documentation,

- The heading font size is 14 and text should be in the Times New Roman, bold.
- The subheading should be in the Times New Roman, the font size is 12, bold.
- For the other font size is 12, Times New Roman and regular font.

1.3 Intended Audience and Reading Suggestions:

We are developing our project vehicle tracking and scheduling system for any trading organization. This SRS is mainly developed for the project development team. In this team there are the project manager, developer, coder, tester and documentation writer and the user of the project also.

1.4 Project Scope:

The Management processes facilitates the management of the programs, and policies. Specially, the management processes support the following activities among others,

- Analyzing the number of users.
- Analyzing the quantity of the truck.
- Analyzing the tracking of the vehicle.

2. OVERALL DESCRIPTION:

2.1 Product perspective:

The vehicle tracking and scheduling system is used to measure the stress involved by making number of users to access the application. This will help to find the amount of stress handled by the application.

2.2 Product Features:

There are two different users who will be using this product,

- The administrator who monitors the trucks.
- The truck which comes to measure the quantity.

2.3 User classes and characteristics:

The users of the system are trucks and the administrators who maintain the system. The administrator are assumed to have the basic knowledge of the computer and the internet browsing.

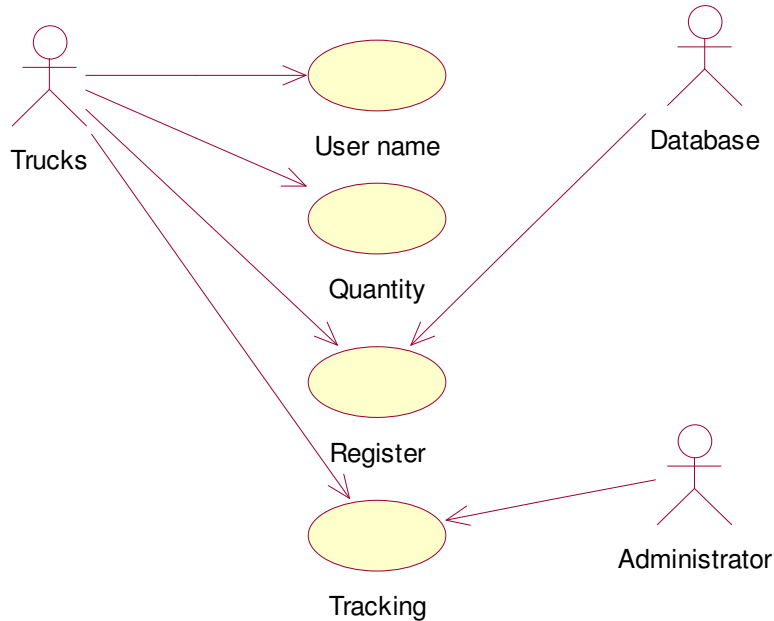
2.4 Software requirement:

The tools using to implement the Stress Testing are

- Visual Studio 2010
- SQL Server 2005

2.5 Design:

Use Case diagram:



3. FUNCTIONAL REQUIREMENTS:

- In this system there are lot of functioning.
- The administrator can be able to view the details about the truck.
- The details will be recorded in the database.

4. EXTERNAL INTERFACE SYSTEM:

4.1 User Interfaces:

- The design or the layout of every form will be very clear and very interactive to the user.
- When the user open the software the welcome window will appear.
- In the login window the user can easily entered the desired username.
- Then it will give the successful login message.

4.2 Hardware Interfaces:

The existing LAN will be used for collecting the data from the users and also for updating in the database.

4.3 Software Interface:

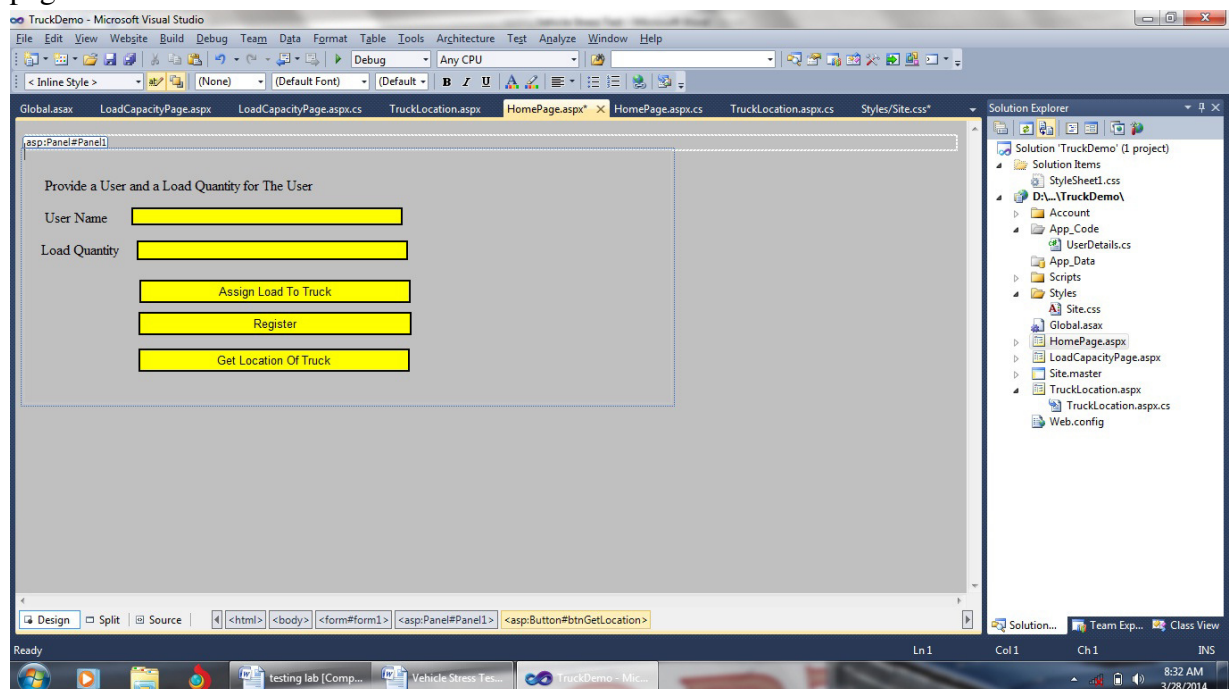
A firewall will be used with the server to prevent the unauthorized access to the system.

4.4 Communication Interface:

Vehicle tracking mechanism will be connected to the World Wide Web.

Steps:

1. Create a new project in Visual Studio 2010 by selecting **File > New ASP.Net Website**
2. In the Web Location dialog box, name the project as **TruckDemo** and click OK to create the project.
3. In the solution explorer right click the newly created TruckDemo solution and select Add New Item Option -> web form
4. In the Name dialog box, name the home page as **HomePage.aspx** and click add.
5. Create the following pages with the items as mentioned in the screenshot in the Home page screen.



HomePage.aspx

7. In the HomePage.aspx.cs page create the following code

```
public partial class HomePage : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
}
```

```

//Assigning Trucks

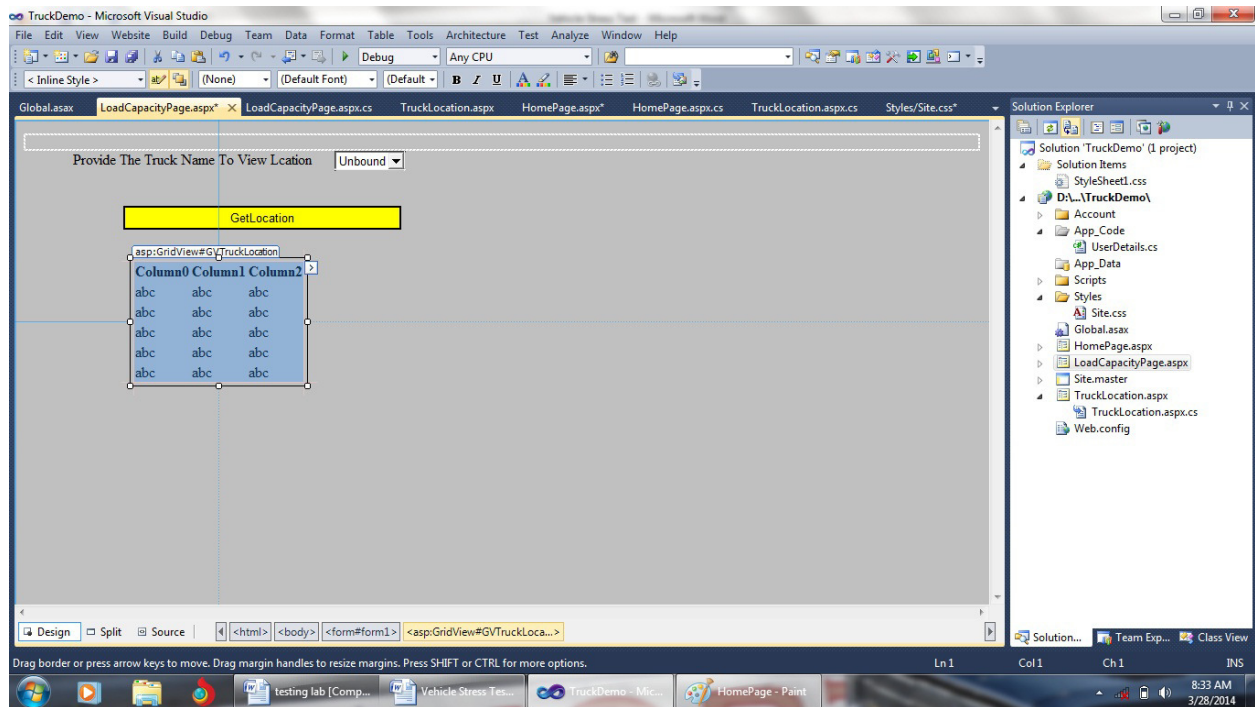
protected void btnGetValue_Click1(object sender, EventArgs e)
{
    string connectionString =
ConfigurationManager.ConnectionStrings["SQLDbConnection"].ToString();
    SqlConnection connection = new SqlConnection(connectionString);
    connection.Open();
    string UserName = txtUserName.Text;
    int LoadQuantity = Convert.ToInt32(txtLoadQuantity.Text);

    SqlCommand command = new SqlCommand("insert into UserDetails
values(@UserName,@LoadQuantity)", connection);
    {
        command.Parameters.Add(new SqlParameter("UserName", UserName));
        command.Parameters.Add(new SqlParameter("LoadQuantity", LoadQuantity));
        command.ExecuteNonQuery();
    }
    SqlDataAdapter adpt = new SqlDataAdapter(command);
}
// Go to LoadCapacityPage
protected void btnAssignLoad_Click(object sender, EventArgs e)
{
    Response.Redirect("LoadCapacityPage.aspx");
}

//Go To TruckLocation.aspx
protected void Button1_Click(object sender, EventArgs e)
{
    Response.Redirect("TruckLocation.aspx");
}
}

```

8. Similarly following steps 4 to 6 create the LoadCapacity page with the following elements



9. In the LoadCapacity.aspx.cs page create the following code

```
public partial class LoadCapacityPage : System.Web.UI.Page
{
    /// <summary>
    /// Call The BindTruckName() function on PageLoad
    /// </summary>
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            BindTruckName();
        }
    }

    /// <summary>
    /// Call the BindGridToTruck(); function on button click
    /// </summary>
    protected void btnGetLocation_Click(object sender, EventArgs e)
    {
        BindGridToTruck();
    }

    /// <summary>
    /// Bind The Truck Names From Database To DropDownList

```

```

/// </summary>

public void BindTruckName()
{
    string connectionString =
ConfigurationManager.ConnectionStrings["SQLDbConnection"].ToString();
    SqlConnection connection = new SqlConnection(connectionString);
    connection.Open();
    SqlCommand command = new SqlCommand("select truck_name from TruckInfo",
connection);
    command.ExecuteNonQuery();
    SqlDataReader dr = command.ExecuteReader();

    DropDownList1.DataSource = dr;
    DropDownList1.Items.Add("--Please Select Truck--");
    DropDownList1.DataTextField = "truck_name";
    DropDownList1.DataValueField = "truck_name";
    DropDownList1.DataBind();
    connection.Close();
    dr.Close();
}
/// <summary>
//Bind The Truck Location values from DB to GridView
//Selection Based on DropDownList Value
/// </summary>
public void BindGridToTruck()
{
    string connectionString =
ConfigurationManager.ConnectionStrings["SQLDbConnection"].ToString();
    SqlConnection connection = new SqlConnection(connectionString);
    connection.Open();
    SqlCommand command = new SqlCommand("select truck_location from TruckInfo where
truck_name='" + DropDownList1.SelectedValue + "'", connection);
    command.ExecuteNonQuery();
    SqlDataAdapter adpt = new SqlDataAdapter(command);
    DataTable dt1 = new DataTable();
    adpt.Fill(dt1);
    GVTruckLocation.DataSource = dt1;
    GVTruckLocation.DataBind();
    connection.Close();
}

protected void GVTruckLocation_SelectedIndexChanged(object sender, EventArgs e)
{
}

```

```

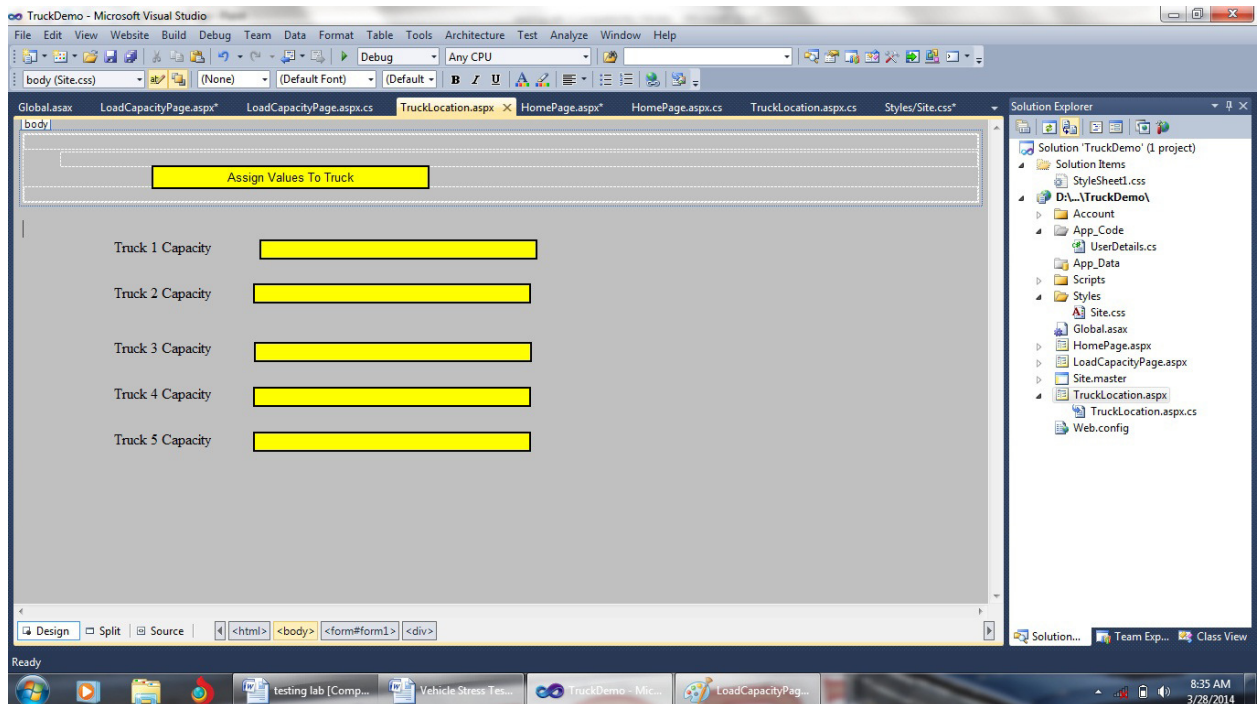
protected void DropDownList1_SelectedIndexChanged1(object sender, EventArgs e)
{

}

}

```

9. Similarly Following Steps 4 to 6 Create the TruckLocation Page with the following elements



10. In the TruckLocation.aspx.cs page create the following code

```

public partial class LoadCapacityPage : System.Web.UI.Page
{
    /// <summary>
    //Call The BindTruckName() function on PageLoad
    /// </summary>
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            BindTruckName();
        }
    }
}

```

```

/// <summary>
/// Call the BindGridToTruck(); function on button click
/// </summary>
protected void btnGetLocation_Click(object sender, EventArgs e)
{
    BindGridToTruck();
}

/// <summary>
///Bind The Truck Names From Database To DropDownList
/// </summary>

public void BindTruckName()
{
    string connectionString =
ConfigurationManager.ConnectionStrings["SQLDbConnection"].ToString();
    SqlConnection connection = new SqlConnection(connectionString);
    connection.Open();
    SqlCommand command = new SqlCommand("select truck_name from TruckInfo",
connection);
    command.ExecuteNonQuery();
    SqlDataReader dr = command.ExecuteReader();

    DropDownList1.DataSource = dr;
    DropDownList1.Items.Add("--Please Select Truck--");
    DropDownList1.DataTextField = "truck_name";
    DropDownList1.DataValueField = "truck_name";
    DropDownList1.DataBind();
    connection.Close();
    dr.Close();
}
/// <summary>
///Bind The Truck Location values from DB to GridView
///Selection Based on DropDownList Value
/// </summary>
public void BindGridToTruck()
{
    string connectionString =
ConfigurationManager.ConnectionStrings["SQLDbConnection"].ToString();
    SqlConnection connection = new SqlConnection(connectionString);
    connection.Open();
    SqlCommand command = new SqlCommand("select truck_location from TruckInfo where
truck_name='" + DropDownList1.SelectedValue + "'", connection);
    command.ExecuteNonQuery();
    SqlDataAdapter adpt = new SqlDataAdapter(command);

```

```
DataTable dt1 = new DataTable();  
adpt.Fill(dt1);  
GVTruckLocation.DataSource = dt1;  
GVTruckLocation.DataBind();  
connection.Close();  
}
```

```
protected void GVTruckLocation_SelectedIndexChanged(object sender, EventArgs e)  
{  
  
}  
protected void DropDownList1_SelectedIndexChanged1(object sender, EventArgs e)  
{  
  
}  
}
```

11.Save the project

12.Run the project.

LOAD RUNNER:

Load Runner is divided up into 3 smaller applications:

The Virtual User Generator allows us to determine what actions we would like our Vusers, or virtual users, to perform within the application. We create scripts that generate a series of actions, such as logging on, navigating through the application, and exiting the program.

The Controller takes the scripts that we have made and runs them through a schedule that we set up. We tell the Controller how many Vusers to activate, when to activate them, and how to group the Vusers and keep track of them.

The Results and Analysis program gives us all the results of the load test in various forms. It allows us to see summaries of data, as well as the details of the load test for pinpointing.

VIRTUAL USER (VUSER):

Load Runner emulates the environment in which thousands of users work with a client server system concurrently. It replaces human users with a VUser.

VIRTUAL SCRIPT:

Actions performed by the human users are recorded in the form of script.

ACTIONS:

Set of user transaction performed in an application to accomplish business tasks.

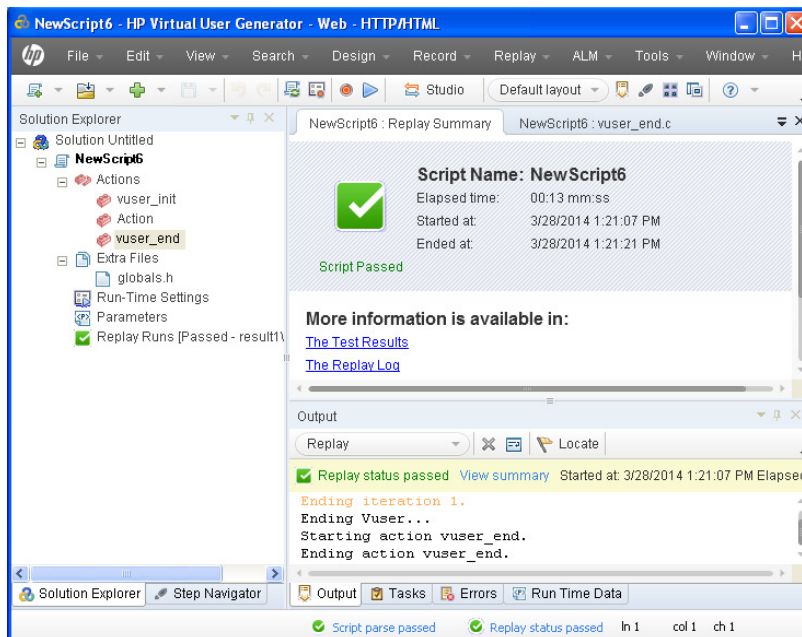
VUser-init(used for logon to the application).

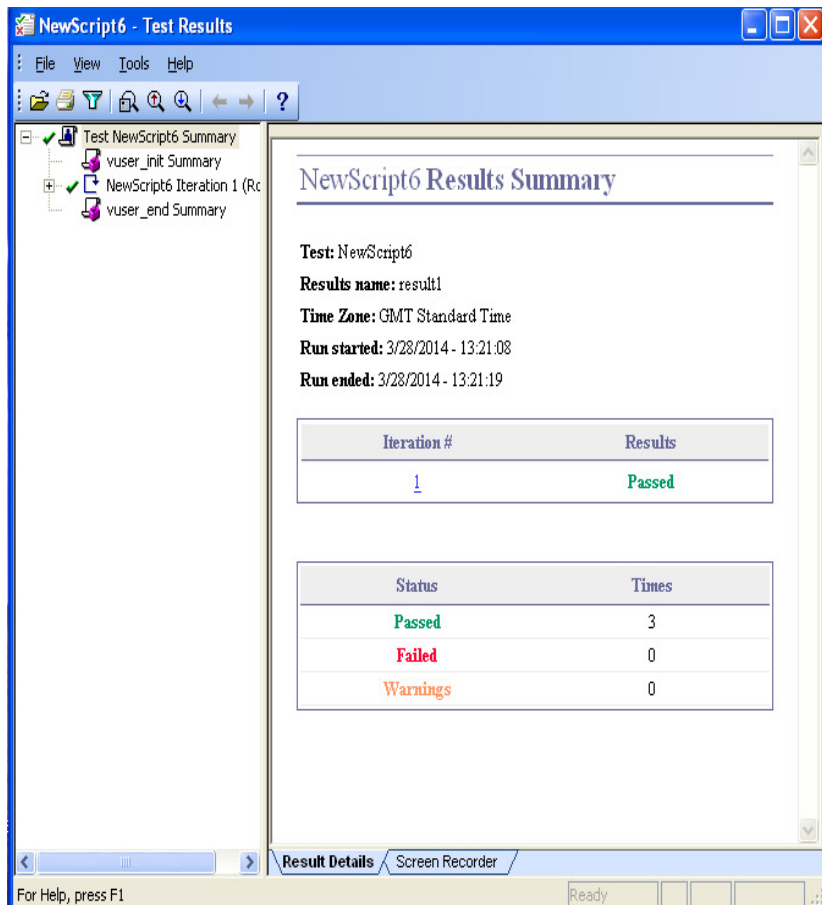
Action(used for business action that needs to be recorded).

VUser-end(used for log off from the application)

VUGen(VIRTUAL USER GENERATOR):

Records only the communication between client and the server.





Result:

Ex.No7

MODEL BASED TESTING

AIM:



To design and develop a scientific calculator program using various GUI components & events and to build the test model & to compare the actual output write the expected output.

PROCEDURE:

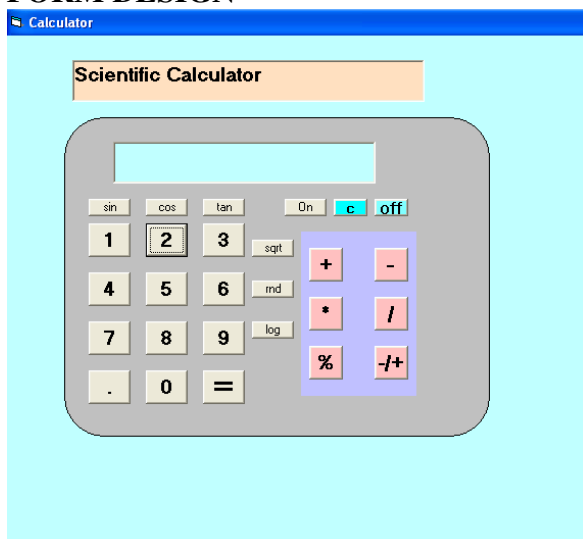
CREATING A COMPONENT:

1. start menu->Microsoft Visual Studio 6.0->Microsoft Visual Basic 6.0
2. select **Activex Control** then click **OK**
3. A new user control is provided,design a **template** for ur calculator in it containing necessary options with the help of text box, cmd buttons etc present in the tool box.
4. Double click every control and add appropriate code for it.
5. Save the user control with **.ctl** extension, thus an activex control is created.

TESTING THE COMPONENT :

1. Next, open a new standard .exe application
File -> New Project -> Standard EXE then click **OK**
2. Now,to import the component created earlier, select **Project -> Add User Control**, in the dialog appearing click on **Existing tab** and select the **user control** just created then click **OK**
3. In the bottom of the tool box, the **user control icon**  appears, click and load it in the form.
4. click on  **play** button standard tool bar.

FORM DESIGN



PROGRAM:

ActivexDLL

File-> new-> activexdll ->write code->r.c on class 1->save class1 as ->ex2.cls->save project->ex2.vbp->project1 properties ->ex2->make dll .

```
Public Function add(val1 As Integer, val2 As Integer) As Integer
add = val1 + val2
End Function
Public Function dif(val1 As Integer, val2 As Integer) As Integer
dif = val1 - val2
End Function
Public Function mul(val1 As Integer, val2 As Integer) As Integer
mul = val1 * val2
End Function
Public Function div(val1 As Integer, val2 As Integer) As Integer
div = val1 / val2
End Function
```

Project->references ->browse->find the path and select the project name ->ok

Project ->properties ->change project type as std exe

Project ->add form->type the code

```
Dim op As String
Option Explicit
```

```
Dim i As Integer
Dim exp1 As Double
Dim exp2 As Double
Dim Result As Double
Dim count1 As Integer
Dim scitype As String
Private X As Double
Dim mem As Variant
Private Sub Command1_Click(Index As Integer)
If count1 = 0 Then
Text1.Text = " "
MsgBox ("Calculator is not on")
End If
If count1 = 1 Then
Text1.Text = " "
count1 = count1 + 1
End If
If count1 > 1 Then
Text1.Text = Text1.Text & command1(Index).Caption
End If
End Sub
Private Sub Command2_Click()
Text1.Text = -Val(Text1.Text)
```

```

End Sub
Private Sub Command3_Click()
If count1 > 0 Then
exp2 = Val(Text1.Text)
Select Case (op)
Case "+"
    Result = exp1 + exp2
    Text1.Text = Result
    count1 = 0
Case "-"
    Result = exp1 - exp2
    Text1.Text = Result
    count1 = 0
Case "*"
    Result = exp1 * exp2
    Text1.Text = Result
    count1 = 0
Case "/"
    Result = exp1 / exp2
    Text1.Text = Result
    count1 = 0
Case "%"
    Result = (exp1 / 100) * exp2
    Text1.Text = Result
    count1 = 0
End Select
End If
End Sub
Private Sub Command4_Click(Index As Integer)
Result = exp1
exp1 = Result + Val(Text1.Text)
Text1.Text = " "
op = Command4(Index).Caption
End Sub
Private Sub Command5_Click()
Result = 0
exp1 = 0
exp2 = 0
Text1.Text = " "
count1 = 1
End Sub
Private Sub Command6_Click()
count1 = 0
Text1.Text = ""
End Sub
Private Sub Command7_Click()
Result = 0

```

```

exp1 = 0
exp2 = 0
count1 = 1
Text1.Text = "0"
End Sub

Private Sub UpdateLog()
    Trim (Form1.Text1.Text)

End Sub
Private Sub Command8_Click(Index As Integer)
scitype = Command8(Index).Caption
Select Case (scitype)
Case "sin"
    Text1.Text = (Text1.Text * 3.14) / 180
    Text1.Text = Math.Sin(Val(Text1.Text))
    count1 = 0
Case "cos"
    Text1.Text = (Text1.Text * 3.14) / 180
    Text1.Text = Math.Cos(Val(Text1.Text))
    count1 = 0
Case "tan"
    Text1.Text = (Text1.Text * 3.14) / 180
    Text1.Text = Math.Tan(Val(Text1.Text))
    count1 = 0
Case "sqrt"
    Text1.Text = Math.Sqrt(Val(Text1.Text))
    count1 = 0

Case "rnd"
    Text1.Text = Math.Rnd(Val(Text1.Text))
    count1 = 0
Case "log"
    Text1.Text = Math.Log(Val(Text1.Text))
    count1 = 0

End Select
End Sub

```

TEST APPLICATION USING WINRUNNER TOOL:

WinRunner tool is a functionality testing tool.

RUNNING PROCEDURE:

After running the Application open WinRunner

Start-> all programs->WinRunner application->click WinRunner

->click continue-> ok

Then click New Test-> click Record-> perform some operation in Application

-> then Stop -> then click From Top(for running the WinRunner)-> click ok

-> then Application is running-> then display the output

WINRUNNER SCRIPT:

Calci

```
win_mouse_click ("Calci", 74, 122);
set_window ("Calci", 1);
button_press ("1");
button_press ("+");
button_press ("2");
button_press ("=");
button_press ("2");
```

Calci_2

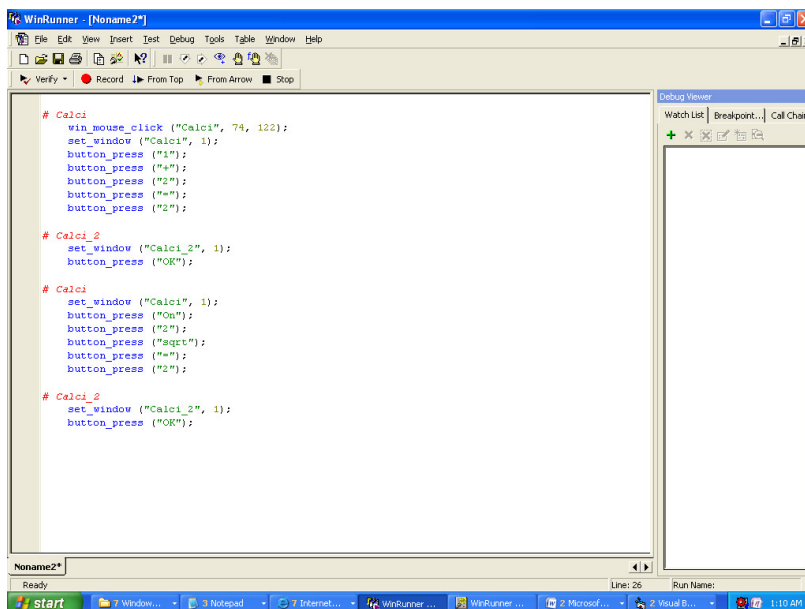
```
set_window ("Calci_2", 1);
button_press ("OK");
```

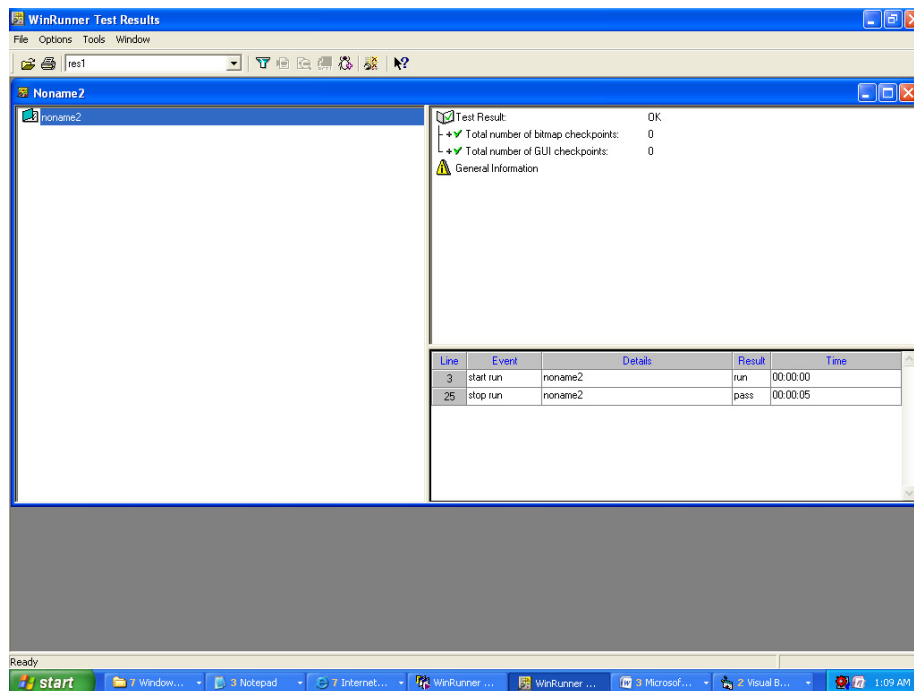
Calci

```
set_window ("Calci", 1);
button_press ("On");
button_press ("2");
button_press ("sqrt");
button_press ("=");
button_press ("2");
```

Calci_2

```
set_window ("Calci_2", 1);
button_press ("OK");
```





RESULT:

Aim:

To develop a web based Application for Student Grade System and to test its performance using v-test.

1. Problem Statement:

All student assignment work is marked and recorded by individual module tutors using a spreadsheet, or similar, of their own design. In the computing department these results are input into a master spreadsheet to track a student's overall progress throughout their programme of study. This is then made available to students through the web portal used in college. Perform web application testing for this scenario

2.Purpose

The main objective of this document is to illustrate the student grade system. The purpose is to store the student details with their grades.

2.1Intended Audience and Reading Suggestions

We are developing our project student mark grade for the college or any Institute. But this project can be deployed in any organization. This SRS is mainly developed for the project development team. In this team there are the project manager, developer, coder, tester and documentation writer and the user of the project also.

2.1.1 Student(user)

This document is intended to user to make them ensure that this document is well meeting the need of the users.

2.1.2 Project Manager

This SRS document is also very important for the project manager to ensure that can estimate the cost easily by referring to the SRS document and that it contains all the information require planning the project.

2.1.3 Project Developer

The project developer will refer to the SRS document to make sure that they developed exactly that the customer require

2.1.4 Tester

The tester will read this SRS document and they will ensure that the requirements are understandable from functionality point of view so that he can test the software and validate its working.

2.1.5 Document Writer

The document writer is reading the SRS document is to ensure that they understand the document well enough to be able to write the users manuals. The developers are those who develop the coding for this project, staffs, the users are students.

3. Project Scope

The Management Processes facilitates the management of the programs, and policies.

Specifically, the Management Processes support the following activities, among others:

- Analyzing the number of students present
- Analyzing branch and their department.
- Analyzing the appropriate grades to the student.
- Analyzing staff productivity and workflow.
- Tracking and verifying financial transactions.

4. Overall Description

4.1 Product Perspective

The student mark grade System is used to improve the efficiency of students and the staffs. The student mark grade System used to be developed benefits for the student details and the admin of institute. The system provides information to staffs about the students. The administrator can keep the student details updated all the time so that the members get the updated information all the time.

4.2 Product Features

There are two different users who will be using this product:

- The administration who maintain the student details and can add new students or staff details
- Staffs enter the grades of students.
- Students who can view their details.

4.3 User Classes and Characteristics

The users of the system are students, staff of the institute and the administrators who maintain the system. The staffs and the administrator are assumed to have basic knowledge of the computers and Internet browsing.

4.4. Operating Environment

4.4.1 Software Requirement

Database -: MY SQL

Language -: Eclipse EE (jdk 1.7)

Web Browser -: Apache Tomcat Server (any versions)

SQL GUI -: SqlYog (any versions)

5. Design and Implementation Constraints

The Product is developed using javajsp. The backend database used for this product is Microsoft Access. The product is accomplished with login facility so that specific function is available to specific student.

5.1.1 Modules:

Student Login:

The Student can enter into the web page and login and view their results.

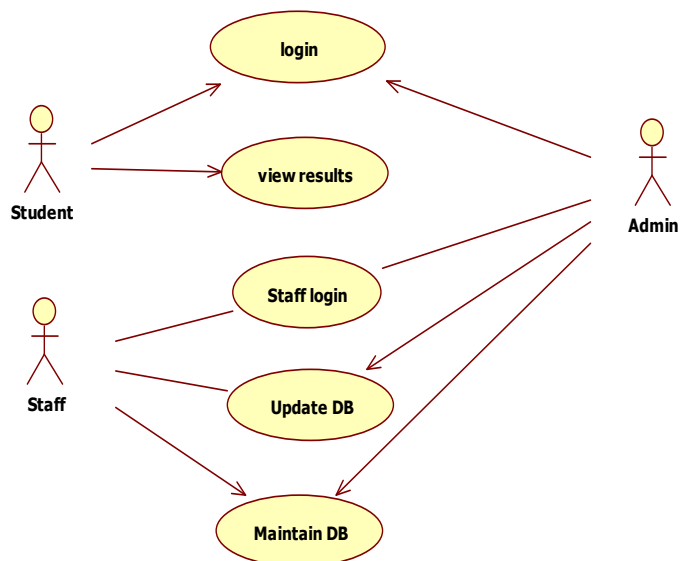
Staff Login:

Responsible for maintaining the student data.

Admin Login:

Controls the entire Operations.

5.1. use case diagram



6. User Documentation

The nature of these systems is unique to application development as they combine aspects of programming (hyperlinks, etc) with aspects of technical writing (organization, presentation). Online help is provided for each and every feature provided by the system. The User Manual describes the use of the system to staffs and the students. The user manual should be available as a hard copy.

7. Assumptions and Dependencies

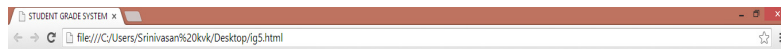
The users have sufficient knowledge of computers. The institute computers should have Internet connection and Internet server capabilities. The users know the English language, as the user interface will be provided in English. The product can access institute student database.

8. Steps for create the web application of a student grade system and to test it using V-Test:

1. Open Eclipse EE and create new Dynamic web application new->project->dynamic web application.
2. Create sufficient JSP pages which is relevant to the project and link with it corresponding Servlets.
3. once finishing the codings run the project in the local web browser use sql Server as a default server.
4. Mark the performance of the application in v test software.

9. SCREEN SHOTS:

9.1 Index Page for Student Portal Management System



WEB APPLICATION TESTING FOR STUDENT GRADE SYSTEM

StudentLogin	
ENTER USERNAME	<input type="text"/>
ENTER PASSWORD	<input type="password"/>
<input type="button" value="submit"/> <input type="button" value="reset"/>	

[click here for staff login](#)

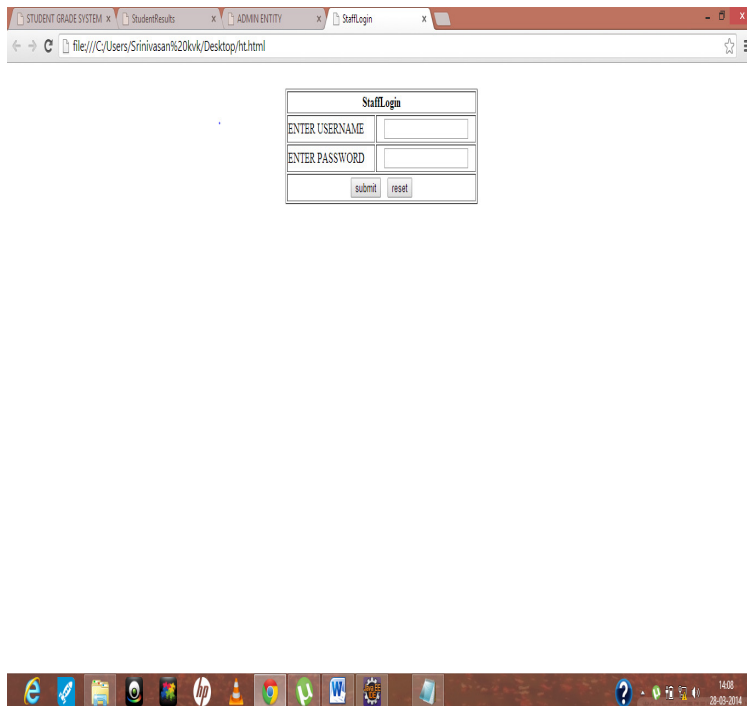
[click here for Administrator login](#)

9.2 STUDENT MASTER DATA SET

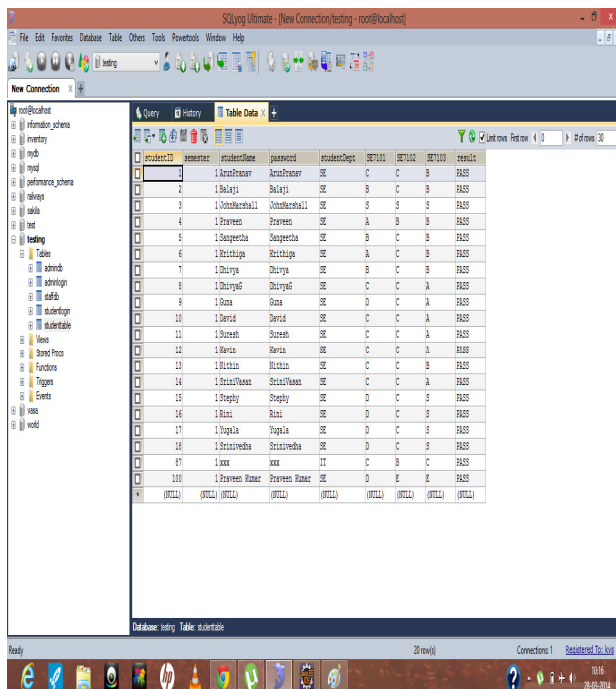


ALTER STUDENT DATABASE	
ENTER STUDENT ID	<input type="text"/>
ENTER SEMESTER	<input type="text"/>
ENTER STUDENT NAME	<input type="text"/>
ALLOCATE PASSWORD FOR STD	<input type="text"/>
ENTER STUDENT DEPT	<input type="text"/>
ENTER THE GRADE FOR SE7101	<input type="text"/>
ENTER THE GRADE FOR SE7102	<input type="text"/>
ENTER THE GRADE FOR SE7103	<input type="text"/>
ENTER THE RESULT	<input type="text"/>
<input type="button" value="submit"/> <input type="button" value="Reset"/>	

9.3 Staff Login Page



9.4 MASTER DATABASE



10. V-Test:

Functional Testing Tool :

V test is a functional testing tool for web apps. It enables you to comprehensively test your web applications in a large variety of settings. You can improve efficiency by easily producing automated functional testing scripts, running both automated and custom test scripts, producing test reports and locating software defects early in the development lifecycle.

V test improves the efficiency of software testing teams through its ability to produce automated testing scripts. These scripts can be executed unattended. They can also be executed based on a set schedule. This typically results in large productivity improvements and enables software testing teams to produce significantly more work

Benefits:

Automate regression and functional testing and hence accelerate product development

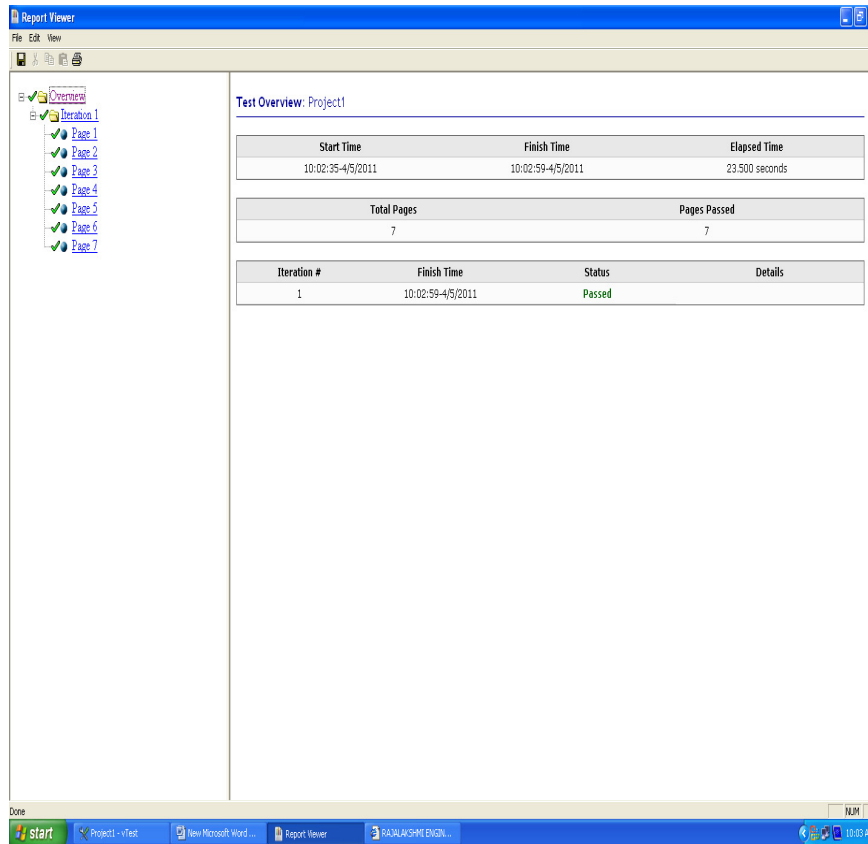
Use of an automated functional testing tool enhances the software testing engineer productivity by enabling them to spend less time on repetitive manual testing and more time on other tasks

Provide flexibility by offering the ability to run functional testing scripts on demand or according to a set schedule. A development background is not required. It does however fully support JavaScript and test engineers with a development background can use JavaScript for scripting.

10.1 Steps for Running V-Test:

1. open verisium v-test and click record button
2. enter the URL you want to track the performance
3. V –test Tracks each and every web pages and finally produces the report.

V-Test Report:



TEST CASES:

Require ment Number	Name of the test case	Requiremen t	Description	Input	Expected Output	Actual Output	Test Status Pass/ Fail
R1	Validating the process of URL.	Enter an URL.	To check whether the URL is correct or not.	URL address	URL to open its home page	URL to open its home page	Pass

R2	Validating the link.	Click on the link in the home page	To check whether the link is working or not.	Click needed on link.	Redirecti on of the link from a page to another page.	Link has been redirecte d.	Pass
R3	Validating all the available links	Click on all the links available	To check all the links are working correctly or not	All links to be clicked.	Redirected to required link.	Redirect ed to required link	Pass
R4	Validating the response time.	Click on the link.	To check based upon the response time of the server.	Click to be needed on link	Response time for the requested link has to be displayed .	Respons e time for the requeste d link has to be displaye d	Pass

Result: