

NAME : **MASOOD SAID**

ID : **13723**

SEMESTER : **6th**

PAPER : **SOFTWARE VERIFICATION AND VALIDATION**

INSTRUCTOR NAME : **ZAIN SHAUKAT**

SILK TEST

a)pros :-

The major thing it has helped with is to reduce the worked on testing activities.

The statistics that are available are very good.

The ability to develop scripts in visual studio , visual studio, is the most valuable feature.

The scalability of the solution is quite good.you can easily expand the product if you need to.

The feature I like most is the ease of reporting.

b)Cons:-

The support for automation with iOS applications can be better.

Everything is very manual.It's up to us to find out exactly what the issues are.

They should extend some of the functions that are a bit clunky and improve the integration .

The solution has a lack of compatibility with newer technologies.

The pricting is an issue,the program is very expensive. That is something that can improve .

b)Functionality:-

Silk Test identifies all windows and controls of the application under test as objects and defines all of the properties and attributes of each window. Thus it supports an object-based implementation. Silk Test supports both recording and descriptive programming methods to identify and verify the elements in the AUT.

c)supporting languages:-

Silk test workbench allow automation testing on a visual level (similar to former TestPartner) as well as using VB. Net as scripting language

Silk Test Classic uses the domain specific 4Test language for automation scripting . It is an object oriented language similar to c++. It uses the concepts of classes , objects, and inheritance .

Silk4J allow automation in Eclipse using Java as scripting language

Silk4Net allows the same in Visual studio using VB or c#

d) Supporting Tests:-

Silk Test Client is an IDE for creating, maintaining and executing tests.

- **Silk Test Agent:** translates the script commands into GUI commands (User actions). These commands can be executed on the same machine as the host or on a remote machine.

Silk Test identifies all windows and controls of the application under test as objects and defines all of the properties and attributes of each window. Thus it supports an object-based implementation.

Silk Test supports both recording and descriptive programming methods to identify and verify the elements in the AUT. It also identifies any mouse movements and keystrokes, enabling testing against custom objects in the AUT.

Silk Test supports testing of different technologies: Mobile (iOS, Android), .NET (WinForms, WPF), Java (Swing, SWT), DOM, IE, Firefox, Chrome, Edge, Safari, SAP Windows GUI.

e) Write a short(faulty) code, Test using this tool, and show the bugs in the code.

```
Public class calculator
```

```
    Public shared function Add (left As Integer, right As Integer) As Integer
```

```
        Return left + right
```

```
    End Function
```

```
    Public Shared Function Subtract (left As Integer, right As Integer) As Integer
```

```
        Return left - right
```

```
    End Function
```

```
    Public Shared Function Multiply (left As Integer , right As Integer)
```

```
        Return left * right
```

```
    End Function
```

```
    Public Shared Function Divide (left As Integer, right As Integer) As Integer
```

```
        If right = 0 Then
```

```
            Throw New ArgumentException("can't divide by zero,")
```

```
        End If
```

```
        Return left / right
```

```
    End Function
```

```
End Class
```

```
Public Module Main
```

```
    Dim _desktop As Desktop = Agent .Desktop
```

```
    Public sub Main ( )
```

```
        Dim A As Integer = 1
```

```
        Dim B As Integer = 2
```

```
        Dim result As Integer
```

```
        result = calculator .Add(A,B)
```

```
        MsgBox (result)
```

```
    End sub
```

```
End Module
```

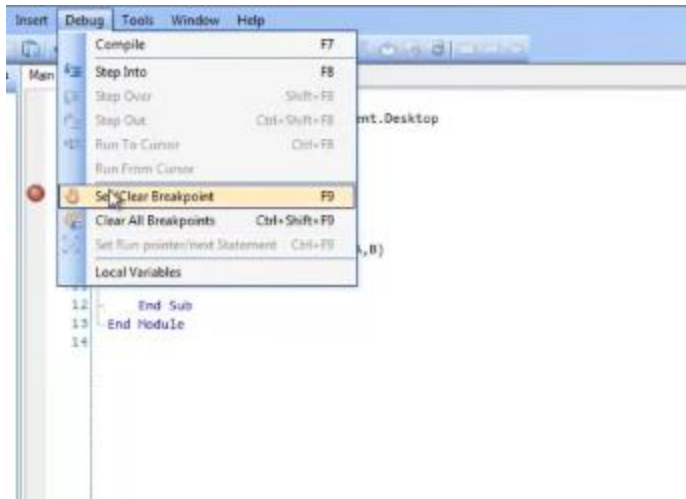
The main part :-

```
1 Public Class Calculator
2
3 Public Shared Function Add (left As Integer, right As Integer) As Integer
4     Return left + right
5 End Function
6
7 Public Shared Function Subtract (left As Integer, right As Integer) As Integer
8     Return left - right
9 End Function
10
11 Public Shared Function Multiply (left As Integer, right As Integer) As Integer
12     Return left * right
13 End Function
14
15 Public Shared Function Divide (left As Integer, right As Integer) As Integer
16     If right = 0 Then
17         Throw New ArgumentException("Can't divide by zero.")
18     End If
19     Return left / right
20 End Function
21
22 End Class
```

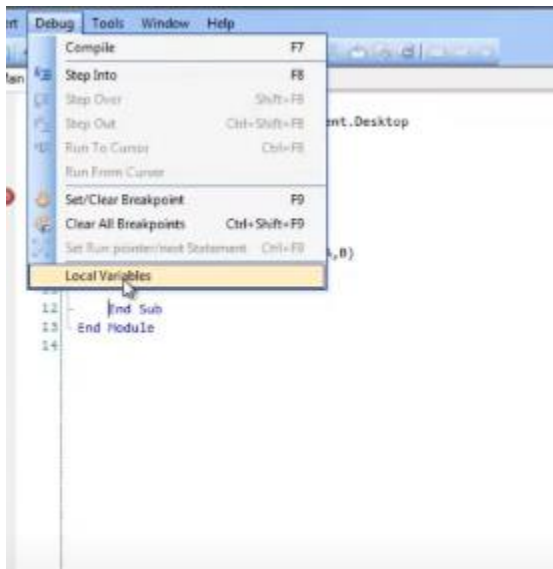
Calculator functions :-

```
1 Public Class Calculator
2
3     Public Shared Function Add (left As Integer, right As Integer) As Integer
4         Return left + right
5     End Function
6
7     Public Shared Function Subtract (left As Integer, right As Integer) As Integer
8         Return left - right
9     End Function
10
11    Public Shared Function Multiply (left As Integer, right As Integer) As Integer
12        Return left * right
13    End Function
14
15    Public Shared Function Divide (left As Integer, right As Integer) As Integer
16        If right = 0 Then
17            Throw New ArgumentException("Can't divide by zero.")
18        End If
19        Return left / right
20    End Function
21
22 End Class
```

DEBUGGING:-



The variable is missing then I debug then it will show



Testing complete this is the screen shot :-

The screenshot displays the Visual Studio IDE. The main window shows a code file named 'Calculator' with the following VB.NET code:

```
1 Public Module Main
2     Dim _desktop As Desktop = Agent.Desktop
3
4     Public Sub Main()
5         Dim A As Integer = 1
6         Dim B As Integer = 2
7         Dim result As Integer
8
9         result = Calculator.Add(A,B)
10        MsgBox(result)
11
12    End Sub
13 End Module
14
```

The 'Local Variables (Demo)' window at the bottom shows the state of local variables and parameters:

Variable Name	Value	Type
A	1	Integer
B	2	Integer
result	0	Integer