

WEB APPLICATION TESTING

A PROJECT REPORT

P- 2174

Submitted by

MADHUMIDHA.S.N

71204205019

PRIYA NOEL.N

71204205030

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

KUMARAGURU COLLEGE OF TECHNOLOGY, COIMBATORE

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2008



P- 2174

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report “WEB APPLICATION TESTING” is the bonafide work of “MADHUMIDHA.S.N and PRIYA NOEL.N” who carried out the project work under my supervision.



SIGNATURE

Dr.S.Thangasamy, BE(HONS), Phd
HEAD OF THE DEPARTMENT



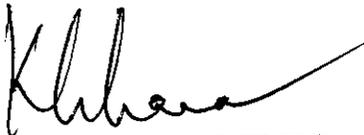
SIGNATURE

Mr.E.A.Vimal, M.E, Lecturer
SUPERVISOR

Dept of Information Technology,
Kumaraguru College of Technology,
Coimbatore – 641 006

Dept of Information Technology,
Kumaraguru College of Technology,
Coimbatore – 641 006.

The candidates with University Register No 71204205019 and 71204205030 were examined by us in the project viva-voice examination held on.....



INTERNAL EXAMINER



EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We express our sincere thanks to our chairman **Padmabhushan Arutselvar Dr. N. Mahalingam B.Sc, F.I.E** and Vice Chairman **Prof. K. Arumugam B.E., M.S., M.I.E.**, for all their support and ray of strengthening hope extended. We are immensely grateful to our principal **Dr. Joseph V. Thanikal M.E., Ph.D., PDF., CEPIT.**, for his invaluable support to the outcome of this project.

We are deeply obliged to **Dr. S. Thangasamy, B.E, Ph.D.**, Dean, Department of Information Technology for his valuable advice and useful suggestions throughout this project.

We also extend our heartfelt thanks to our project guide **Mr. E.A. Vimal M.E.**, Lecturer, Department of Information Technology who rendered his valuable guidance and support to perform our project work extremely well.

We thank the teaching and non-teaching staffs of our Department for providing us the technical support in the duration of our project.

We express our humble gratitude and thanks to our beloved parents and family members who have supported and helped us to complete the project and our friends, for lending us valuable tips, support and co-operation throughout the project work.

DECLARATION

We,

MADHUMIDHA.S.N

71204205019

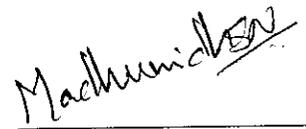
PRIYA NOEL.N

71204205030

hereby declare that the project entitled “**WEB APPLICATION TESTING**”, submitted in partial fulfillment to Anna University as the project work of Bachelor of Technology (Information Technology) Degree, is a record of original work done by us under the supervision and guidance of Department of Information Technology, Kumaraguru college of Technology, Coimbatore.

Place: Coimbatore

Date: 21-04-08



[Madhumidha.S.N]



[Priya Noel.N]

Project Guided by



[Mr.E.A.Vimal]

ABSTRACT

WEB applications are one of the fastest growing classes of software systems in which the usage pattern can change rapidly and maintenance consists of many small incremental change. Hence the search for a generalizable and practical approach for the testing of Web applications is more complicated.

We have proposed a more generalized testing tool using java to measure the performance characteristics like link checking, accuracy of image retrieved, and amount of resource utilized for any web application. The script of the web page for link checking is generated using the bean scripting framework and various views of the internet resource is generated to point to link where the error occurs. The actual image is captured to compare with the expected image and its accuracy is found. The resource utilized, its assembly coding, by any executable file can be obtained by giving the path of the executable file.

Our system includes various features like easy-to-use user interface, tests are XML documents, cookies, browser history, multiple views of Internet resources (browser, formatted source code, DOM tree), scriptable through Bean Scripting Framework and platform independent.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF TABLES	viii
	LIST OF FIGURES	ix
	LIST OF ABBREVIATIONS	x
1.	INTRODUCTION	
	1.1 GENERAL	1
	1.2 PROBLEM DEFINITION	2
	1.3 OBJECTIVE OF PROJECT	3
2.	LITERATURE REVIEW	
	2.1 FEASIBILITY STUDY	4
	2.1.1 CURRENT STATUS OF THE PROBLEM	4
	2.1.2 PROPOSED SYSTEM AND ITS ADVANTAGE	4
	2.2 HARDWARE REQUIREMENTS	5
	2.3 SOFTWARE REQUIREMENTS	5
	2.4 SOFTWARE OVERVIEW	5
3.	DETAILS OF METHODOLOGY EMPLOYED	9
4.	PERFORMANCE EVALUATION	
	4.1 UNIT TESTING	13
	4.2 INTEGRATION TESTING	14
	4.3 SYSTEM TESTING	14
	4.4 VALIDAION TESTING	14

5.	CONCLUSION	15
6.	FUTURE ENHANCEMENTS	16
	APPENDICES	
	APPENDIX I- SOURCE CODE	17
	APPENDIX II- SCREENSHOT	48
	REFERENCES	55

LIST OF TABLES

Table No	TITLE	Page No
1.	LIST OF METHODS USED	11

LIST OF FIGURES

Figure No	TITLE	Page No
1.	SEQUENCE DIAGRAM	9

LIST OF ABBREVIATIONS

URL	: Uniform Resource Locator
HTML	: HyperText Markup Language
XML	: Extended Markup Language
DOM	: Dynamic Object Model
BSF	: Bean Scripting Framework
API	: Application Programming Interface

1. INTRODUCTION

1.1 GENERAL:

The web application testing tool is developed for testing of web applications developed by different companies before its deployment. This is a more generalized tool for testing any web application. We can check if all the links in the particular web application is working. The testing of link is independent of the scripting code used and hence it is not affected by the small incremental changes made in the application. The same testlets saved can be easily played back to check if there is any problem due to the small incremental changes made in the application.

This testing tool can serve different purposes. It can check that in the particular web page all the links are working properly and expected strings are present at the expected location. It can perform image comparison to check if the image retrieved during the web browsing is same as the expected image.

Any web application using any scripting language can be retrieved and tested for its performance. To get a more generalised web testing tool we use BSF(Bean Scripting Framework). It uses a set of java classes with architecture for incorporating any scripting language into java application.

1.2 PROBLEM DEFINITION:

This is a more generalized testing tool using java to measure the performance characteristics of a web page before deployment. The scripting code for a web page is obtained using bean scripting framework. Various views for the web page is generated for testing.

Web application testing is done to check what characteristics of the application are unacceptable and in what way the amount of resources used can be reduced and system performance can be improved. This testing tool can be used to make sure that the following links are working as expected and images captured are same as expected. The customers request can be kept as a baseline to evaluate the effectiveness of the actual system.

Web applications are one of the fastest growing classes of software systems in which the usage pattern can change rapidly and maintenance consists of many small incremental changes. Hence the search for a generalizable and practical approach for the testing of Web applications is more complicated.

Our system includes various features like easy-to-use user interface, tests are XML documents, cookies, browser history, multiple views of Internet resources (browser, formatted source code, DOM tree), scriptable through Bean Scripting Framework and platform independent.

1.3 OBJECTIVE OF THE PROJECT:

The objective of our project is to develop a more generalized software-testing tool for testing web-based applications. It can demonstrate that the system meets performance criteria. It can perform image comparison to check if the image retrieved during the web browsing is same as the expected image.

This is a generalized testing tool using java to measure the performance characteristics like link checking, accuracy of image retrieved in a web application and amount of resource utilized by any executable file. The script of the web page for link checking is generated using the bean scripting framework and various views (browser view, script view, DOM tree view) of the internet resource is generated to point to link where the error occurs. The actual image is retrieved to compare with the expected image and its accuracy is found. The image is saved in JPEG format. Image is saved as numerical value and pixel-by-pixel comparison is done. The resource utilized, its assembly coding, by any executable file can be obtained by giving the path of the executable file.

This tool has an easy-to-use user interface which can be used even by inexperienced web application testers to check for the proper working of link. We need not click each link to ensure its proper working. All the links associated with a particular URL is automatically verified.

2. LITERATURE REVIEW

2.1 FEASIBILITY STUDY

2.1.1 CURRENT STATUS OF THE PROBLEM

While tools exist already to test the functionality of web sites in a repeatable manner, they have several drawbacks like:

- They are proprietary
- They are expensive
- They work only on Windows
- They are difficult to use
- They are non-intuitive

Performance tests when overestimate the performance and scalability of the web applications it leads to embarrassing and costly crashes, when underestimated capacity and scalability it leads to unnecessary spending on hardware and infrastructure.

2.1.2 PROPOSED SYSTEM AND ITS ADVANTAGE

Our system overcomes the disadvantages of the existing system by providing simple, generalized, platform independent testing tool for testing web applications before its deployment. This system tests for the performance characteristics like

- Link availability
- Availability of appropriate strings in expected locations
- Image comparison

It also includes various features like:

- Easy-to-use user interface
- Tests are XML documents
- Cookies
- Multiple views of Internet resources (browser, formatted source code, DOM tree)
- Scriptable through Bean Scripting Framework
- Platform independent

The images retrieved for comparison occupies less memory space and is stored in jpeg format.

2.2 HARDWARE REQUIREMENTS(MINIMUM)

Processor	:	Intel Pentium IV 2.4 GHZ
Hard Disk Drive	:	20 GB.
RAM	:	256 MB RAM minimum.

2.3 SOFTWARE REQUIREMENTS

Operating System:	Windows 2000/XP
Software	: Net Beans 6.0, jdk1.5

2.4 SOFTWARE OVERVIEW

2.4.1. JAVA:

Java is an object oriented programming language. Java compiler compiles the java source code to obtain an object code termed byte code in java terminology. Java is a platform independent, means the byte code which is the output of the java compiler will run on any processor and

operating system. Java programs are of two types, application and applets. A java application is a stand-alone program that can be invoked from the command line .a java applet program is a small program embedded in a web page and it will be run when that page is browsed using web browser. The Java Development Kit (JDK) from sun Microsystems consists of an extensive library of java classes called packages. It also contains java compiler (javac.exe) and the java interpreter (java.exe). It also provides a very useful utility called applet viewer (applet viewer.exe). The various advantages in using java language is, its an interpreted language, architectural neutral and portable.

There are six features in java that makes java as a powerful tool for internet applications.

1. Security
2. The core API
3. Open Standard
4. Distributed and Dynamic
5. Object Oriented
6. Multithread

2.4.2 JAVA SWINGS

The Swing package is part of Java Foundation Classes (JFC) in the Java platform. The JFC encompasses a group of features to help people build GUIs; Swing provides all the components from buttons to split panes and tables. The Swing package was first available as an add-on to JDK 1.1. Swing components is a “lightweight” component.

Swing provides many standard GUI components such as buttons, lists, menus and textareas, which we combine to create our program's GUI. It also includes containers such as windows and tool bars. Containers use layout managers to determine the size and position of the components they contain.

Swing programs perform all their painting and event handling in the event-dispatching thread. If we do something to a visible component that might depend on or affect its state, then we need to do it from the event-dispatching thread.

These are some of the major concepts we need to know to build Swing GUIs - the containment hierarchy, layout management, event handling, painting, and threads. These are some of the other important Swing features. They are

- Features that JComponent provides
- Icons
- Actions
- Pluggable Look and Feel support
- Support for assistive technologies
- Separate data and state models

2.4.3 NET BEANS

The NetBeans APIs are the public interfaces and classes which are available to module writers. They are divided into specific APIs for dealing with different types of functionality. The contents and behavior of the Java source packages and its sub packages, as specified in the NetBeans API List, are the NetBeans APIs.

After you register the NetBeans API sources and Javadoc, you can refer to them in the Source Editor, while developing NetBeans modules.

A rich-client application is a complete, functioning, standalone Swing application, built on top of the NetBeans Platform. When creating Java applications you typically set up a single main project, which contains both the project main class, and any required projects. Whenever you create separate standard projects for each of a project's source roots, you have to set up the classpath dependencies between the main project and the required projects. A required project is a project that has been added to another project's classpath.

For standard projects that have a main class is specified, the IDE automatically copies any JAR files on the project's classpath to the dist/lib folder when you build the project. The IDE also adds each of the JAR files to the Class-Path element in the application JAR's manifest.mf file. This simplifies running the application outside the IDE.

Whenever you clean and build a project, the IDE also cleans and builds its required projects. The required project's attached Javadoc and source code are also made available to the receiving project.

3. DETAILS OF METHODOLOGY EMPLOYED

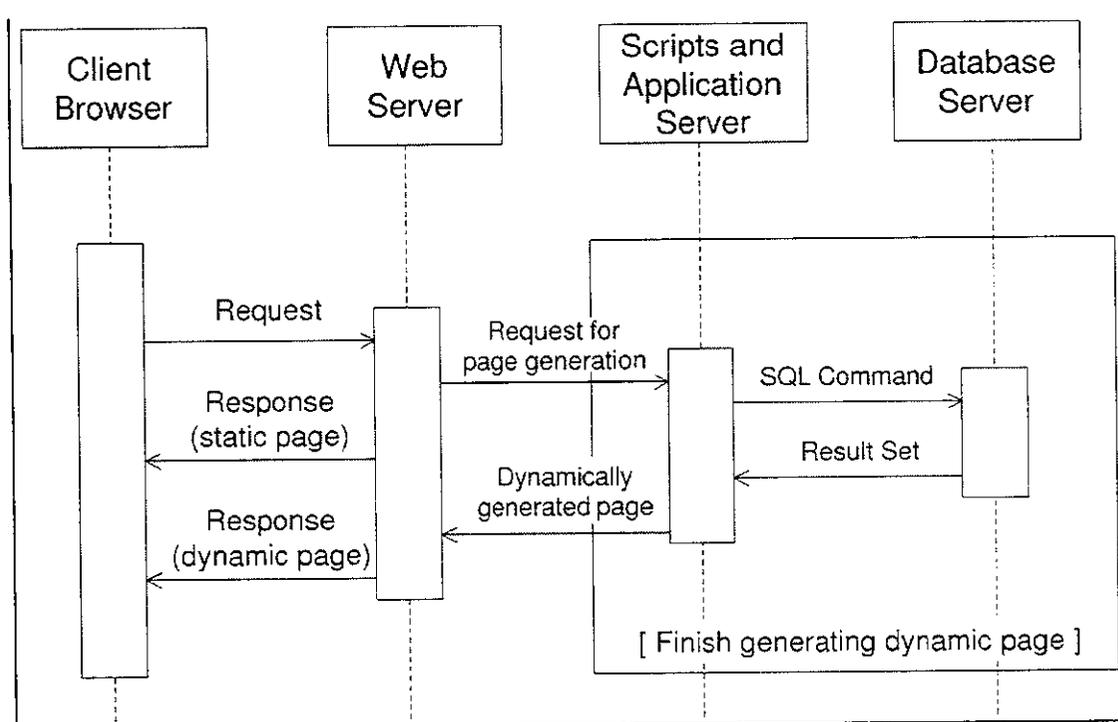


Fig. 1. Sequence diagram of a Web application.

Fig. 1 shows how a simple Web application operates. A user (client) sends a request through a Web browser. The Web server responds by delivering content to the client. This content generally takes the form of some markup language (e.g., HTML) that is later interpreted by the browser to render a Web page at the user site. Each web page is linked to a number of other sites. The link is clicked to access the corresponding pages.

Once a web page is created it is very difficult to check manually if each link is working or not. This link checking can be done in a more

simplified way by just giving the URL of the web page alone. This is done by using our web testing tool. Once the URL is added and the test is run the all the link connected to it is automatically checked. The testlets can be added for each URL and verified. If a link fails then the error position in the link is noted.

The web page can be retrieved in three different views namely browser view, script view and DOM (Dynamic Object Model) tree view BSF. The BSF framework is used to incorporate any scripting language into java code. The teslets can be added and saved. These teslets can be opened and later played back to check if its working for small incremental changes made in the design/coding.

These are the various methods are included in the WebTest:

Method	Description
	The WebTest application object.
getTest	Get the current org.webtest.Test object.
getBrowser	Get the browser component.
getConfiguration	Get the current Configuration.
newTest	Create a new test. If the current test is unsaved then a dialog will appear giving the option to save the current test.
open	Display a dialog to open a test.

save	Save the current test.
saveAs	Save the current test, displaying a file dialog first.
revert	Revert to the last saved version of the test.
play	Play the current test.
stop	Stop the current test.
viewSource	View the current page's source.
viewHeaders	View the current page's headers.
viewCookies	View the current cookies in memory.
back	Go back in the browser's history.
forward	Go forward in the browser's history.
properties	Show the properties dialog.

Table1: List of methods used

URL availability is verified by checking automatically all the links associated with a particular URL is automatically verified. This is done by finding the anchor links in the script of the web page. These links are verified by opening the URL connection. When some exceptions occurs an error message is displayed.

Testing is done to verify if the particular information is available at the expected location or in the expected path in the script retrieved. This is done by obtaining the information and path as string and using the Iterator class the particular path is tracked and the availability of the string is verified.



Image comparison is done to check if the image retrieved during the web browsing is same as the expected image. This is verified by capturing and comparing the images.

Robot class is used to capture screen contents. Screen width and height is kept as a rectangle to support Robot's createScreenCapture() method. To allow the display of images that cannot be fully displayed without scrolling, the ImageArea component is placed into a JScrollPane. A GUI for capturing the image is created. From the screenshot captured, the required image can be cropped and saved in JPEG format to save memory utilization.

The image stored in jpeg format is retrieved and a numeric value for each pixel of the image is found and stored in an array format. The absolute value of the image is retrieved and compared with expected image to check if it is same or not.

The file BitEditPlus is linked to the project to find the various resources utilized by any executable file. It gives various informations like the resources, target address, logs etc.

4. PERFORMANCE EVALUATION

Testing is a critical element of software quality and assurance and represents the ultimate review of specification design and coding. It is a vital activity that has to be enforced in the development of any system. This could be done in parallel during all the phases of system development. The feedback received from these tests can be used for further enhancement of the system under consideration. The testing phase conducts test using the Software Requirement Specification as a reference and with the goal to see whether system satisfies the specified requirements.

Standard procedures have been followed in testing our system. Test cases are generated for each screen. These test cases will cover every possibility which could result in both positive and negative results. These test plans are maintained for any further testing done on the system.

The main types of tests carried out are:

- Unit Testing.
- Integration Testing.
- System Testing.
- Validation Testing

4.1 Unit Testing

Each and every module is tested separately to check if its intended functionality is met. Some unit testing performed are:

1. Validating the User
2. Loading Database and applications

4.2 Integration Testing

It is the testing performed to detect errors on interconnection between modules. The applications should connect to respective databases. The application should connect to respective databases. The application events should be backed up in the log file for future recovery.

4.3 System Testing

The system is tested against the system requirements to see if all the requirements are met and if the system performs as per the specified requirements. The system is tested as a whole to check for its functionality.

4.4 Validation Testing

This test is done to check for the validity of the entered input. The user inputs to the corresponding application input fields are verified before updating in the database.

5. CONCLUSION

The complete design and development of the system is presented in the dissertation. The system has user-friendly features and easy-to-use user interface. Its platform independent and can be expanded in future. The system has been tested for various web pages and they provide satisfactory performance. This simple tool can be used to check for the performance of the web sites and report the developer.

6.FUTURE ENHANCEMENT

This project can be further enhanced to include various other performance related testing tools. This project is designed in such a way that further enhancements can be made in an easy manner and is quite flexible.

The project can be enhanced to correct the broken links that are found during the link checking. The error messages can be processed and displayed in meaningful way. This can be expanded to check if the attributes given within the tags in the script is as expected by the customer.

APPENDIX

SOURCE CODE

Welcome.java

```
package javaapplication23;
import java.security.Provider;
import java.security.Security;
import org.apache.log4j.*;
import org.logitest.*;
import org.logitest.io.*;
import org.logitest.ui.util.*;
import org.logitest.util.*;
import org.logitest.event.*;
import org.logitest.ui.LogiTest;
import java.io.*;
import java.awt.event.*;
import java.net.*;
import java.util.*;
import java.security.*;
import javax.swing.*;
import javax.swing.tree.*;
import javax.swing.undo.*;
import javax.swing.event.*;
import org.apache.log4j.*;
import org.logitest.*;
import org.logitest.io.*;
import org.logitest.ui.util.*;
import org.logitest.util.*;
import org.logitest.event.*;
import javaapplication23.org.logitest.ui.*;
public class Welcome extends javax.swing.JFrame {
    public static final String DEFAULT_BASE_PATH =
        "org/logitest/icons/";
    public Welcome() {
        initComponents();
    }

    private void initComponents() {
```

```

jLabel1 = new javax.swing.JLabel();
jScrollPane1 = new javax.swing.JScrollPane();
jEditorPane1 = new javax.swing.JEditorPane();
jPanel1 = new javax.swing.JPanel();
jButton1 = new javax.swing.JButton();
jButton3 = new javax.swing.JButton();
jPanel2 = new javax.swing.JPanel();
jButton2 = new javax.swing.JButton();
jButton4 = new javax.swing.JButton();
    jSeparator1 = new javax.swing.JSeparator();
    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_
        ON_CLOSE);
setTitle("Main");
jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18));
jLabel1.setText("Welcome to Web Testing");
jEditorPane1.setEditable(false);
jScrollPane1.setViewportView(jEditorPane1);
jPanel1.setBorder(javax.swing.BorderFactory.createEtchedBorder());
jButton1.setText("Start Service");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});
jButton3.setText("System Restore");
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});
org.jdesktop.layout.GroupLayout jPanel1Layout = new
org.jdesktop.layout.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
jPanel1Layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
.add(jPanel1Layout.createSequentialGroup()
    add(39, 39, 39)
    .add(jPanel1Layout.createParallelGroup(org.jdesktop.layout.GroupLayoutLa
yout.LEADING)
.add(org.jdesktop.layout.GroupLayout.TRAILING,
jButton3, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 109,
Short.MAX_VALUE)

```

```

        .add(org.jdesktop.layout.GroupLayout.TRAILING, jButton1,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 109,
Short.MAX_VALUE))
        .add(32, 32, 32));
jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup
        (org.jdesktop.layout.GroupLayout.LEADING)
        .add(jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .add(jButton1,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 23,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
        .add(18, 18, 18)
        .add(jButton3)
        .addContainerGap(25, Short.MAX_VALUE)));
jPanel2.setBorder(javax.swing.BorderFactory.createEtchedBorder());
jButton2.setText("Web Testing");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});
jButton4.setText("Property Capture");
jButton4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton4ActionPerformed(evt);
    }
});
org.jdesktop.layout.GroupLayout jPanel2Layout = new
org.jdesktop.layout.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
jPanel2Layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
        .add(jPanel2Layout.createSequentialGroup()
        .add(jPanel2Layout.createSequentialGroup()
        .add(24, 24, 24)
        .add(jPanel2Layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING, false)
        .add(org.jdesktop.layout.GroupLayout.TRAILING, jButton4,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

```

```

        .add(org.jdesktop.layout.GroupLayout.TRAILING, jButton2,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 116,
Short.MAX_VALUE))
        .addContainerGap(35, Short.MAX_VALUE));
jPanel2Layout.setVerticalGroup(
jPanel2Layout.createParallelGroup
(org.jdesktop.layout.GroupLayout.LEADING)
.add(jPanel2Layout.createSequentialGroup()
.addContainerGap()
.add(jButton2)
.add(18, 18, 18)
.add(jButton4)
.addContainerGap(25, Short.MAX_VALUE));
org.jdesktop.layout.GroupLayout layout = new
org.jdesktop.layout.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
.add(jSeparator1,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 400,
Short.MAX_VALUE)
.add(layout.createSequentialGroup()
.addContainerGap()
.add(jPanel1,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.add(jPanel2,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
.add(20, 20, 20))
.add(layout.createSequentialGroup()
.addContainerGap()
.add(jScrollPane1)
.addContainerGap())
.add(layout.createSequentialGroup()
.add(92, 92, 92)

```

```

        .add(jLabel1)
        .addContainerGap(84, Short.MAX_VALUE) );
    layout.setVerticalGroup(
layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
        .add(layout.createSequentialGroup()
            .addContainerGap()
            .add(jLabel1)
            .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
            .add(jScrollPane1,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 141,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(org.jdesktop.layout.LayoutStyle.UNRELATED)
                .add(jSeparator1,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 10,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                    .add(layout.createParallelGroup
(org.jdesktop.layout.GroupLayout.LEADING)
                        .add(jPanel1,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                            .add(jPanel2,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
                                .addContainerGap(org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE) );
        pack();
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            Runtime r=Runtime.getRuntime();
            r.exec("cmd /c start net start");
        } catch(Exception e) {
            jEditorPanel1.setText(String.valueOf(e));
        }
    }
    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

```

```

try {
    Runtime r=Runtime.getRuntime();
    r.exec("C:\\WINDOWS\\system32\\Restore\\rstrui");
} catch(Exception e) {
    jEditorPanel.setText(String.valueOf(e));
}
}
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
IconManager.setDefaultBasePath(DEFAULT_BASE_PATH);
    ResourceManager.setLocale(Locale.FRENCH);
    try{
        LogiTest app = new LogiTest();
        app.setTitle("Web Test");
        app.show();
    } catch(Exception e){ }
}
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    new Property_Capture().setVisible(true);
    }
    public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Welcome().setVisible(true);
        }
    });
}
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JEditorPane jEditorPanel1;
private javax.swing.JLabel jLabel1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JSeparator jSeparator1;
}

```

Main.Java

```
package javaapplication23;
import java.security.Provider;
import java.security.Security;
import org.apache.log4j.*;
import org.logitest.*;
import org.logitest.io.*;
import org.logitest.ui.util.*;
import org.logitest.util.*;
import org.logitest.event.*;
import org.logitest.ui.LogiTest;
import java.io.*;
import java.awt.event.*;
import java.net.*;
import java.util.*;
import java.security.*;
import javax.swing.*;
import javax.swing.tree.*;
import javax.swing.undo.*;
import javax.swing.event.*;
import org.apache.log4j.*;
import org.logitest.*;
import org.logitest.io.*;
import org.logitest.ui.util.*;
import org.logitest.util.*;
import org.logitest.event.*;
import javaapplication23.org.logitest.ui.*;
public class Main {
    public static final String DEFAULT_BASE_PATH = "org/logitest/icons/";

    public static void main(String[] args) {
        if(args.length > 0){
            PropertyConfigurator.configure(args[0]);
        } else {
            BasicConfigurator.configure();
        }
        IconManager.setDefaultBasePath(DEFAULT_BASE_PATH);
        try{
```

```

        Security.addProvider((Provider)Class.forName(
"com.sun.net.ssl.internal.ssl.Provider").newInstance());
        System.setProperty("java.protocol.handler.pkgs",
            "com.sun.net.ssl.internal.www.protocol");
    } catch(Exception e){
        System.out.println("Unable to registered SSL provider.");
    }
    HttpURLConnection.setFollowRedirects(true);
    try{
        Preferences.getInstance().load();
    } catch(Exception e){

    }
    try{
        LogiTest app = new LogiTest();
        app.show();
    } catch(Exception e){
    } // new LogiTest().show();
    // TODO code application logic here
}
}
}

```

ExpectTestlet.java

```

package javaapplication23.org.logitest.testlet;
import java.awt.Component;
import java.net.*;
import java.util.*;
import javax.swing.*;
import org.jdom.*;
import org.logitest.*;
public class LinkTestlet extends Testlet{
    public LinkTestlet(){ }
    public boolean test(Resource resource) throws Exception{
        Document document = resource.getDocument();
        URL context = new URL(resource.getURL());
        List passed = new ArrayList();
        List failed = new ArrayList();
        test(document.getRootElement(), context, passed, failed);
    }
}

```

```

        if(failed.size() == 0){
            setState(Test.PASSED);
            return true;
        } else {
            setState(Test.FAILED);
            return false;
        }
    }
}

```

```

protected void test(Element element, URL context, List passed, List
failed){

```

```

    Iterator children = element.getChildren().iterator();
    while(children.hasNext()){
        Element child = (Element)children.next();
        if(child.getName().equals("a")){
            if(checkLink(context, child)){
                passed.add(child);
            } else {
                failed.add(child);
            }
        }
    }

    test(child, context, passed, failed);
}
}

```

```

protected boolean checkLink(URL context, Element element){
    Attribute att = element.getAttribute("href");
    if(att != null){
        //System.out.println("Checking attribute");
        try{
            URL url = new URL(context, att.getValue());
            URLConnection conn = url.openConnection();
            conn.connect();
        } catch(Exception e){
            System.out.println("Error in link"+e);
            // JOptionPane.showMessageDialog(this, "link not found");
            return false;
        }
    }
}

```

```

        return true;
    }
    public Component getEditor(){
        return new JPanel();
    }
    protected Map getProperties(){
        if(properties == null){
            properties = new HashMap();
            properties.put(DISPLAY_NAME, "Link Testlet");
            properties.put(DESCRIPTION, "Test for link validity.");
            properties.put(AUTHOR, "Anthony Eden");
            properties.put(VERSION, "1.0");
        }
        return properties;
    }
    private Map properties;
}

```

LinkTestlet.java

```

package javaapplication23.org.logitest.testlet;
import java.awt.Component;
import java.net.*;
import java.util.*;
import javax.swing.*;
import org.jdom.*;
import org.logitest.*;
public class LinkTestlet extends Testlet{
    public LinkTestlet() {}
    public boolean test(Resource resource) throws Exception{
        Document document = resource.getDocument();
        URL context = new URL(resource.getURL());
        List passed = new ArrayList();
        List failed = new ArrayList();
        test(document.getRootElement(), context, passed, failed);
        if(failed.size() == 0){
            setState(Test.PASSED);
            return true;
        } else {
            setState(Test.FAILED);
        }
    }
}

```

```

        return false;
    }
}
protected void test(Element element, URL context, List passed, List
failed){
    Iterator children = element.getChildren().iterator();
    while(children.hasNext()){
        Element child = (Element)children.next();
        if(child.getName().equals("a")){
            if(checkLink(context, child)){
                passed.add(child);
            } else {
                failed.add(child);
            }
        }
    }
    test(child, context, passed, failed);
}

protected boolean checkLink(URL context, Element element){
    Attribute att = element.getAttribute("href");
    if(att != null){
        //System.out.println("Checking attribute");
        try{
            URL url = new URL(context, att.getValue());
            URLConnection conn = url.openConnection();
            conn.connect();
        } catch(Exception e){
            System.out.println("Error in link"+e);
            // JOptionPane.showMessageDialog(this, "link not found");
            return false;
        }
    }
    return true;
}

public Component getEditor(){
    return new JPanel();
}

protected Map getProperties(){
    if(properties == null){
        properties = new HashMap();
    }
}

```

```

        properties.put(DISPLAY_NAME, "Link Testlet");
        properties.put(DESCRIPTION, "Test for link validity.");
        properties.put(AUTHOR, "Anthony Eden");
        properties.put(VERSION, "1.0");
    }
    return properties;
}
private Map properties;
}

```

Capture.java

```

package javaapplication23;
import java.awt.*;
import java.awt.event.*;
import java.awt.image.*;
import java.io.*;
import java.util.*;
import javax.imageio.*;
import javax.imageio.stream.*;
import javax.swing.*;
public class Capture extends JFrame
{
    BufferedImage biScreen;
    Dimension dimScreenSize;
    ImageArea ia = new ImageArea ();
    Rectangle rectScreenSize;
    static Robot robot;
    JScrollPane jsp;
    public Capture (String title)
    {
        super (title);
        setDefaultCloseOperation (EXIT_ON_CLOSE);
        dimScreenSize = Toolkit.getDefaultToolkit ().getScreenSize ();
        rectScreenSize = new Rectangle (dimScreenSize);
        final JFileChooser fcSave = new JFileChooser ();
        fcSave.setCurrentDirectory (new File (System.getProperty ("user.dir")));
        fcSave.setAcceptAllFileFilterUsed (false);
        fcSave.setFileFilter (new ImageFileFilter ());
        // Create the application's menus.
    }
}

```

```

JMenuBar mb = new JMenuBar ();
JMenu menu = new JMenu ("File");
ActionListener al;
JMenuItem mi = new JMenuItem ("Save As...");
mi.setAccelerator (KeyStroke.getKeyStroke (KeyEvent.VK_S,
                                             InputEvent.ALT_MASK));
al = new ActionListener ()
{
    public void actionPerformed (ActionEvent e)
    {
        // Disallow image saving if there is no image to save.
        if (ia.getImage () == null)
        {
            showError ("No captured image.");
            return;
        }
        fcSave.setSelectedFile (null);
        if (fcSave.showSaveDialog (Capture.this) !=
            JFileChooser.APPROVE_OPTION)
            return;
        File file = fcSave.getSelectedFile ();
        String path = file.getAbsolutePath ().toLowerCase ();
        if (!path.endsWith (".jpg") && !path.endsWith (".jpeg"))
            file = new File (path += ".jpg");
        if (file.exists ())
        {
            int choice = JOptionPane.
                showConfirmDialog (null,
                                   "Overwrite file?",
                                   "Capture",
                                   JOptionPane.
                                   YES_NO_OPTION);
            if (choice == JOptionPane.NO_OPTION)
                return;
        }
        ImageWriter writer = null;
        ImageOutputStream ios = null;

        try
        {

```

```

Iterator iter;
iter = ImageIO.getImageWritersByFormatName ("jpeg");
if (!iter.hasNext ())
{
    showError ("Unable to save image to jpeg file type.");
    return;
}
writer = (ImageWriter) iter.next();
ios = ImageIO.createImageOutputStream (file);
writer.setOutput (ios);
ImageWriteParam iwp = writer.getDefaultWriteParam ();
iwp.setCompressionMode
(ImageWriteParam.MODE_EXPLICIT);
iwp.setCompressionQuality (0.95f);
writer.write (null,
              new IIOMemoryImage ((BufferedImage)
                                   ia.getImage (), null, null),iwp);
}
catch (IOException e2)
{
    showError (e2.getMessage ());
}
finally
{
    try
    {
        if (ios != null)
        {
            ios.flush ();
            ios.close ();
        }
        if (writer != null)
            writer.dispose ();
    }
    catch (IOException e2)
    {
        } } } };

```

```

mi.addActionListener (al);
menu.add (mi);

```

```

menu.addSeparator ();
mi = new JMenuItem ("Exit");
mi.setAccelerator (KeyStroke.getKeyStroke (KeyEvent.VK_X,
                                             InputEvent.ALT_MASK));
mi.addActionListener (new ActionListener ()
    {
        public void actionPerformed (ActionEvent e)
        {
            System.exit (0);
        }
    });
menu.add (mi);
mb.add (menu);
menu = new JMenu ("Capture");
mi = new JMenuItem ("Capture");
mi.setAccelerator (KeyStroke.getKeyStroke (KeyEvent.VK_C,
                                             InputEvent.ALT_MASK));
al = new ActionListener ()
    {
        public void actionPerformed (ActionEvent e)
        {
            setVisible (false);
            biScreen = robot.createScreenCapture (rectScreenSize);
            setVisible (true);
            ia.setImage (biScreen);
            jsp.getHorizontalScrollBar ().setValue (0);
            jsp.getVerticalScrollBar ().setValue (0);
        }
    };

mi.addActionListener (al);
menu.add (mi);
mb.add (menu);
mi = new JMenuItem ("Crop");
mi.setAccelerator (KeyStroke.getKeyStroke (KeyEvent.VK_K,
                                             InputEvent.ALT_MASK));
al = new ActionListener ()
    {
        public void actionPerformed (ActionEvent e)
        {

```

```

        if (ia.crop ())
        {
            jsp.getHorizontalScrollBar ().setValue (0);
            jsp.getVerticalScrollBar ().setValue (0);
        }
        else
            showError ("Out of bounds.");
    }
};

```

```

mi.addActionListener (al);
menu.add (mi);
mb.add (menu);
setJMenuBar (mb);
getContentPane ().add (jsp = new JScrollPane (ia));
setSize (dimScreenSize.width/2, dimScreenSize.height/2);
setLocation ((dimScreenSize.width-dimScreenSize.width/2)/2,
            (dimScreenSize.height-dimScreenSize.height/2)/2);
setVisible (true);
}

```

```

public static void showError (String message)

```

```

{
    JOptionPane.showMessageDialog (null, message, "Capture",
        JOptionPane.ERROR_MESSAGE);
}

```

```

public static void main (String [] args)

```

```

{
    try
    {
        robot = new Robot ();
    }
    catch (AWTException e)
    {
        showError (e.getMessage ());
        System.exit (0);
    }
    catch (SecurityException e)
    {
        showError ("Permission required to use Robot.");
        System.exit (0);
    }
}

```

```
    }  
    new Capture ("Capture");  
}}
```

ImageArea.java

```
package javaapplication23;  
import java.awt.*;  
import java.awt.event.*;  
import java.awt.image.*;  
import javax.swing.*;  
public class ImageArea extends JPanel  
{  
    private BasicStroke bs;  
    private GradientPaint gp;  
    private Image image;  
    private int srcx, srcy;  
    private int destx, desty;  
    private Rectangle rectSelection;  
    public ImageArea ()  
    {  
        rectSelection = new Rectangle ();  
        // Define the stroke for drawing selection rectangle outline.  
  
        bs = new BasicStroke (5, BasicStroke.CAP_ROUND,  
BasicStroke.JOIN_ROUND,  
0, new float [] { 12, 12 }, 0);  
        // Define the gradient paint for coloring selection rectangle outline.  
        gp = new GradientPaint (0.0f, 0.0f, Color.red, 1.0f, 1.0f, Color.white,  
true);  
        MouseListener ml;  
        ml = new MouseAdapter ()  
        {  
            public void mousePressed (MouseEvent e)  
            {  
                if (image == null)  
                    return;  
                destx = srcx = e.getX ();  
                desty = srcy = e.getY ();  
                repaint ();  
            }  
        }  
    }  
}
```

```

    }
};
addMouseListener (ml);
    MouseMotionListener mml;
mml = new MouseMotionAdapter ()
    {
        public void mouseDragged (MouseEvent e)
        {
            if (image == null)
                return;
            destx = e.getX ();
            desty = e.getY ();
            repaint ();
        }
    };
addMouseMotionListener (mml);
}

public boolean crop ()
{
    if (srcx == destx && srcy == desty)
        return true;
    boolean succeeded = true;
    int x1 = (srcx < destx) ? srcx : destx;
    int y1 = (srcy < desty) ? srcy : desty;

    int x2 = (srcx > destx) ? srcx : destx;
    int y2 = (srcy > desty) ? srcy : desty;
    // Compute width and height of selection rectangle.
    int width = (x2-x1)+1;
    int height = (y2-y1)+1;
    // Create a buffer to hold cropped image.
    BufferedImage biCrop = new BufferedImage (width, height,
        BufferedImage.TYPE_INT_RGB);
    Graphics2D g2d = biCrop.createGraphics ();
    // Perform the crop operation.
    try
    {
        BufferedImage bi = (BufferedImage) image;
        BufferedImage bi2 = bi.getSubimage (x1, y1, width, height);
    }
}

```

```

    g2d.drawImage (bi2, null, 0, 0);
}
catch (RasterFormatException e)
{
    succeeded = false;
}
g2d.dispose ();
if (succeeded)
    setImage (biCrop); // Implicitly remove selection rectangle.
else
{
    // Prepare to remove selection rectangle.
    srcx = destx;
    srcy = desty;
    // Explicitly remove selection rectangle.
    repaint ();
}
return succeeded;
}
public Image getImage ()
{
    return image;
}
public void paintComponent (Graphics g)
{
    super.paintComponent (g);
    if (image != null)
        g.drawImage (image, 0, 0, this);
    // Draw the selection rectangle if present.
    if (srcx != destx || srcy != desty)
    {
        // Compute upper-left and lower-right coordinates for selection
        // rectangle corners.
        int x1 = (srcx < destx) ? srcx : destx;
        int y1 = (srcy < desty) ? srcy : desty;
        int x2 = (srcx > destx) ? srcx : destx;
        int y2 = (srcy > desty) ? srcy : desty;

        // Establish selection rectangle origin.
        rectSelection.x = x1;

```

```

rectSelection.y = y1;

// Establish selection rectangle extents.
rectSelection.width = (x2-x1)+1;
rectSelection.height = (y2-y1)+1;

// Draw selection rectangle.
Graphics2D g2d = (Graphics2D) g;
g2d.setStroke (bs);
g2d.setPaint (gp);
g2d.draw (rectSelection);
}
}
public void setImage (Image image)
{
    this.image = image;
    setPreferredSize (new Dimension (image.getWidth (this),
                                     image.getHeight (this)));

    revalidate ();
    srcx = destx;
    srcy = desty;
    repaint ();
}
}
}

```

ImageFileFilter.java

```

package javaapplication23;
import java.io.*;
class ImageFileFilter extends javax.swing.filechooser.FileFilter
{
    public boolean accept (File f)
    {
        if (f.isDirectory ())
            return true;
        String s = f.getName ();
        int i = s.lastIndexOf ('.');
        if (i > 0 && i < s.length ()-1)

```

```

{
    String ext = s.substring (i+1).toLowerCase ();

    if (ext.equals ("jpg") || ext.equals ("jpeg"))
        return true;
    }
return false;
}
public String getDescription ()
{
    return "Accepts .jpg and .jpeg files";
}
}

```

Property_capture.java

```

package javaapplication23;
import java.io.IOException;
import java.util.logging.Level;
import java.util.logging.Logger;

public class Property_Capture extends javax.swing.JFrame {
    public Property_Capture() {
        initComponents();
    }
    private void initComponents() {
        jLabel1 = new javax.swing.JLabel();
        jPanel1 = new javax.swing.JPanel();
        jScrollPane1 = new javax.swing.JScrollPane();
        jEditorPane1 = new javax.swing.JEditorPane();
        jButton3 = new javax.swing.JButton();
        jButton1 = new javax.swing.JButton();
        jButton5 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_C
LOSE);
        setTitle("Property Capture");
    }
}

```

```

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 14));
jLabel1.setText("Bitmap Property Testing");
jPanel1.setBorder(javax.swing.BorderFactory.createEtchedBorder());
    jEditorPane1.setEditable(false);
jScrollPane1.setViewportView(jEditorPane1);
org.jdesktop.layout.GroupLayout jPanel1Layout = new
org.jdesktop.layout.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(
        jPanel1Layout.createParallelGroup(org.jdesktop.layout.
            GroupLayout.LEADING)
            .add(jPanel1Layout.createSequentialGroup()
                .add(jPanel1Layout.createParallelGroup()
                    .add(ContainerGap.class)
                    .add(jScrollPane1)
                    .add(ContainerGap.class));
            .add(jPanel1Layout.createSequentialGroup()
                .add(jPanel1Layout.createParallelGroup(org.jdesktop.layout.
                    GroupLayout.LEADING)
                    .add(jPanel1Layout.createSequentialGroup()
                        .add(ContainerGap.class)
                        .add(jScrollPane1,
                            org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 113,
                            org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                        .add(ContainerGap.class));
                .add(jPanel1Layout.createParallelGroup()
                    .add(ContainerGap.class)
                    .add(jButton3)
                    .add(ContainerGap.class));
            .add(jPanel1Layout.createSequentialGroup()
                .add(jButton3)
                .add(ContainerGap.class)
                .add(jButton1)
                .add(ContainerGap.class)
                .add(jButton5)
                .add(ContainerGap.class));
        );
jButton3.setText("Comparison");
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});

jButton1.setText("Bitmap Capture");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

jButton5.setText("Test Results");
jButton5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton5ActionPerformed(evt);
    }
});

```

```

jButton2.setText("Exe Capture");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

org.jdesktop.layout.GroupLayout layout = new
org.jdesktop.layout.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
    .add(org.jdesktop.layout.GroupLayout.TRAILING,
layout.createSequentialGroup()
    .add(ContainerGap(148, Short.MAX_VALUE)
    .add(jLabel1)
    .add(111, 111, 111))
    .add(layout.createSequentialGroup()
    .add(57, 57, 57)
    .add(jButton5)
    .add(PreferredGap(org.jdesktop.layout.LayoutStyle.RELATED,
118, Short.MAX_VALUE)
    .add(jButton2,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 106,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
    .add(56, 56, 56))
    .add(layout.createParallelGroup
        (org.jdesktop.layout.GroupLayout.LEADING)
    .add(layout.createSequentialGroup()
    .add(ContainerGap()
    .add(layout.createParallelGroup
        (org.jdesktop.layout.GroupLayout.LEADING)
    .add(jPanel1,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .add(org.jdesktop.layout.GroupLayout.TRAILING,
layout.createSequentialGroup()
    .add(47, 47, 47)

```

```

        .add(jButton3)
.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED, 117,
Short.MAX_VALUE)
        .add(jButton1,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 111,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
        .add(44, 44, 44)))
        .addContainerGap()))
);
layout.setVerticalGroup(
layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
        .add(layout.createSequentialGroup()
        .addContainerGap()
        .add(jLabel1)
        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED,
224, Short.MAX_VALUE)
.add(layout.createParallelGroup(org.jdesktop.
        layout.GroupLayout.BASELINE)
        .add(jButton5)
        .add(jButton2))
        .add(25, 25, 25))
.add(layout.createParallelGroup
        (org.jdesktop.layout.GroupLayout.LEADING)
        .add(layout.createSequentialGroup()
        .add(30, 30, 30)
        .add(jPanel1,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
        .add(26, 26, 26)
.add(layout.createParallelGroup
        (org.jdesktop.layout.GroupLayout.BASELINE)
        .add(jButton1)
        .add(jButton3))
        .addContainerGap(82, Short.MAX_VALUE)))));

pack();
} // </editor-fold>
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    new compare().setVisible(true); // TODO add your handling code here:

```

```

    }
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Runtime r = Runtime.getRuntime();
        r.exec("a.bat");
    } catch (IOException ex) {
        Logger.getLogger(Welcome.class.getName()).log(Level.SEVERE,
null, ex);
    }
}
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    a.setVisible(true);
}
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
try
{
    Runtime r=Runtime.getRuntime();
    r.exec("bep\\bep\\bep.exe");
}
catch(Exception e)
{
    jEditorPane1.setText(String.valueOf(e));
}
}
public static void main(String args[]) {
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new Property_Capture().setVisible(true);
    }
});
}
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton5;
private javax.swing.JEditorPane jEditorPane1;
private javax.swing.JLabel jLabel1;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
// End of variables declaration

```

```
    TestResults a=new TestResults();
}
```

Compare.java

```
package javaapplication23;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import java.util.logging.Level;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
public class compare extends javax.swing.JFrame {
    String files,files1;
    String s[],s1;
    File f1,f2;
    BufferedImage bi,bii;
    ImageIcon im,im1;
    BufferedImage img,img1;
    int r[]=new int[256],r1[]=new int[256];
    int h=0,w=0;
    int pixel[];
    int pixel1[];
    int pixels[][];
    int pixels1[][];
    int red1=0;
    int blue1=0;
    int green1=0;
    public compare() {
        initComponents();
    }
    private void initComponents() {
        jLabel1 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
    }
}
```

```

jScrollPane1 = new javax.swing.JScrollPane();
jLabel2 = new javax.swing.JLabel();
jScrollPane2 = new javax.swing.JScrollPane();
jLabel3 = new javax.swing.JLabel();
jButton3 = new javax.swing.JButton();
setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setTitle("Bitmap Comparison");
jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18));
jLabel1.setText("Bitmap Comparison");
jButton1.setText("Browse Expected :");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});
jButton2.setText("Browse Actual :");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});
jScrollPane1.setViewportView(jLabel2);
jScrollPane2.setViewportView(jLabel3);
jButton3.setText("Test");
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});
org.jdesktop.layout.GroupLayout layout = new
org.jdesktop.layout.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
    .add(layout.createSequentialGroup()
        .add(108, 108, 108)
        .add(jLabel1)
        .addContainerGap(121, Short.MAX_VALUE))
    .add(layout.createSequentialGroup()

```

```

.add(36, 36, 36)
.add(jButton1)

.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED, 71,
Short.MAX_VALUE)
.add(jButton2)
.add(67, 67, 67))
.add(layout.createSequentialGroup())
.addContainerGap()
.add(jScrollPane1,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 201,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
.add(jScrollPane2,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 177,
Short.MAX_VALUE)
.addContainerGap())
.add(org.jdesktop.layout.GroupLayout.TRAILING,
layout.createSequentialGroup())
.addContainerGap(169, Short.MAX_VALUE)
.add(jButton3,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 74,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
.add(161, 161, 161));
layout.setVerticalGroup(
layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
.add(layout.createSequentialGroup())
.addContainerGap()
.add(jLabel1)
.add(18, 18, 18)
.add(layout.createParallelGroup
(org.jdesktop.layout.GroupLayout.BASELINE)
.add(jButton2)
.add(jButton1))
.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
.add(layout.createParallelGroup
(org.jdesktop.layout.GroupLayout.LEADING)
.add(jScrollPane2,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 215,
Short.MAX_VALUE)

```

```

        .add(jScrollPane1,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 215,
Short.MAX_VALUE))
        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
        .add(jButton3)));
    pack();
}
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    javax.swing.JFileChooser filechooser =new JFileChooser(".");
    int returnVal = filechooser.showOpenDialog(this);
    if(returnVal == JFileChooser.APPROVE_OPTION) {
        fl= filechooser.getSelectedFile();
        files = fl.getPath();
        if(fl.isFile())
        {
            try {
                bi = ImageIO.read(fl);
                ImageIcon im = new ImageIcon(bi);
                jLabel2.setIcon(im);
                //img = ImageIO.read(new File(files));
                h = bi.getHeight();
                w = bi.getWidth();
                int[] rgbs = new int[w*h];
                int x = 0,k=0;
                bi.getRGB(0, 0,w, h, rgbs, 0, w );
                for (int i=0; i<10; i++)
                    for (int j=0;j<10;j++)
                        {
                            System.out.println("Pixel " + i +"," + j + "has " +
                                "RGB values of "+ rgbs[x]);
                            r[k]=rgbs[x];
                            x++; k++;
                        }
            } catch (IOException ex) {
                System.out.println(ex);
            }
        }
    }
    System.out.println(files);
}
}
}

```

```

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    javax.swing.JFileChooser filechooser =new JFileChooser(".");
    int returnVal = filechooser.showOpenDialog(this);
    if(returnVal == JFileChooser.APPROVE_OPTION) {
        f2= filechooser.getSelectedFile();
        files1 = f2.getPath();
        if(f2.isFile())
        {
            try {
                bii = ImageIO.read(f2);
                ImageIcon im1 = new ImageIcon(bii);
                jLabel3.setIcon(im1);
                h = bii.getHeight();
                w = bii.getWidth();
                int[] rgb = new int[w*h];
                int x1 = 0,k=0;
                bii.getRGB(0, 0,w, h, rgb, 0, w );
                for (int i=0; i<10; i++)
                    for (int j=0;j<10;j++)
                    {
                        System.out.println("Pixel " + i +"," + j + "has " + "RGB
                        values of "+ rgb[x1]);
                        r1[k]=rgb[x1];
                        x1++; k++;
                    }
                } catch (IOException ex) {
                    System.out.println(ex);
                }
            }
        }
        System.out.println(files1);
    }
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    for(int i=0; i<=r.length;i++)
    {
        if(Math.abs(r[i])==Math.abs(r1[i]))
        {
            JOptionPane.showMessageDialog(this, "Images are same");
            break;
        }
    }
}

```

```

        }
    else
    {
        JOptionPane.showMessageDialog(this, "Images are not same");
        break;
    }
}

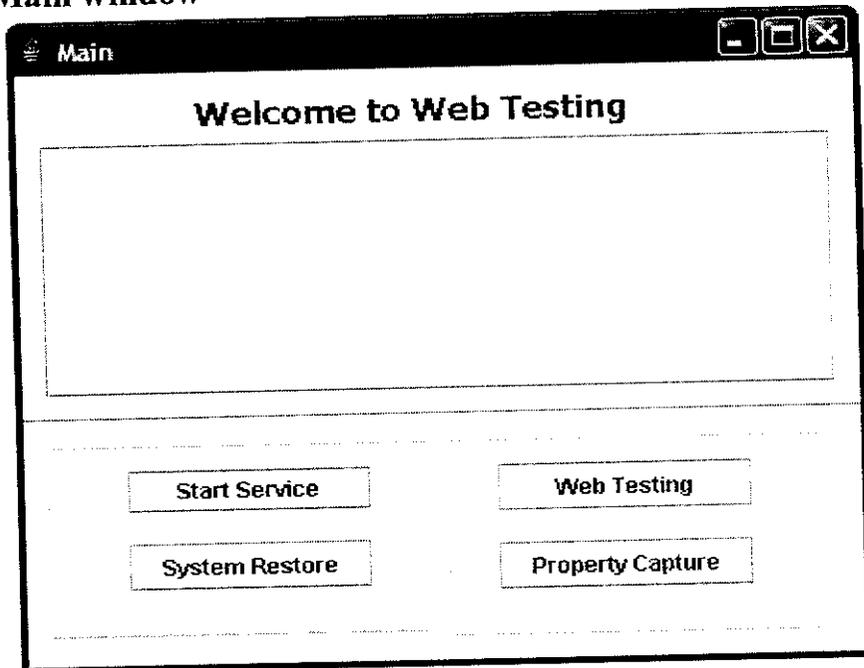
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new compare().setVisible(true);
        }
    });
}

private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
}

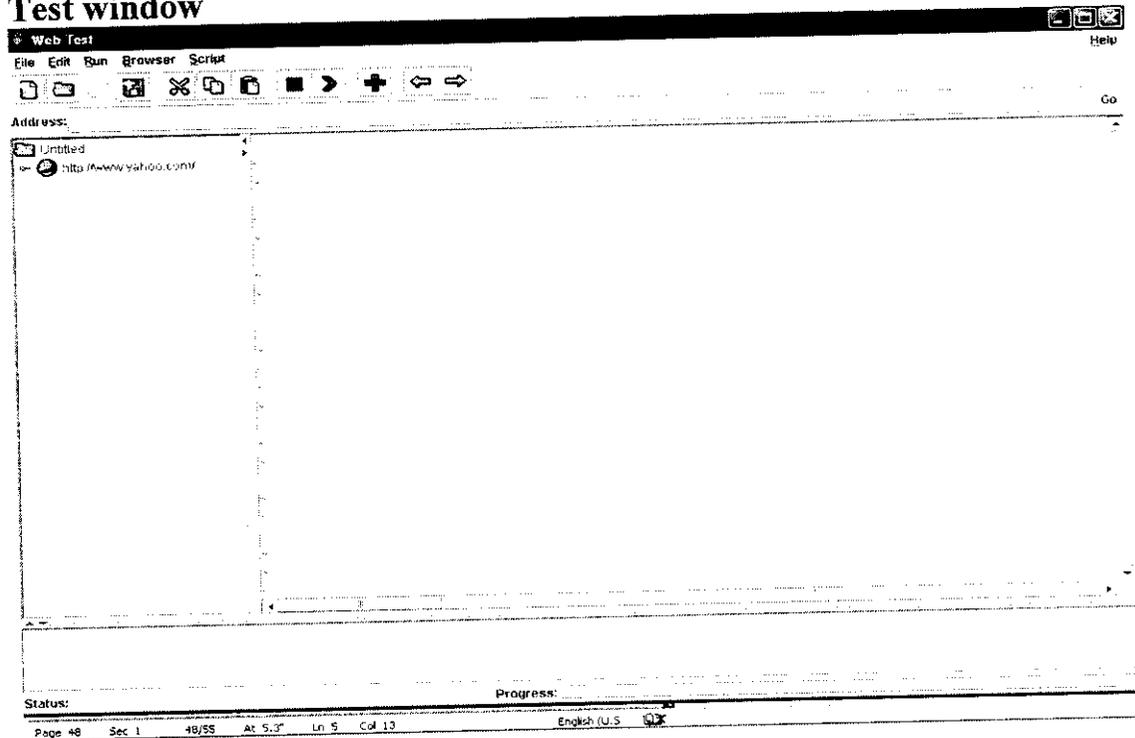
```

SCREENSHOT:

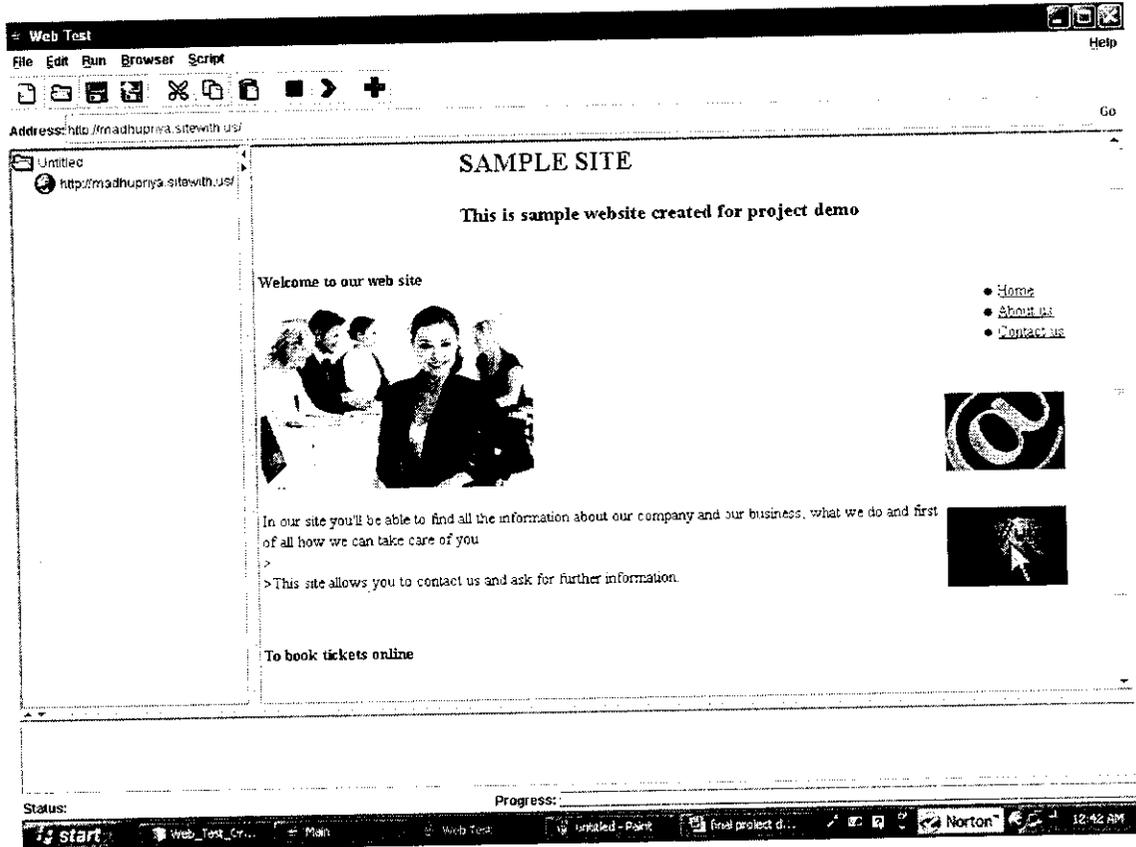
Main window



Test window



Browser view

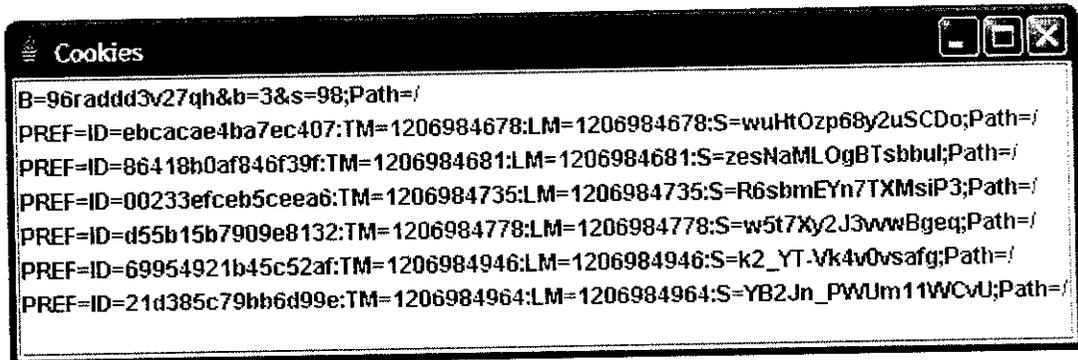


The screenshot shows a web browser window titled "Web Test". The address bar contains "http://madhupriya.sitewith.us/". The main content area displays a "SAMPLE SITE" with the following text and elements:

- SAMPLE SITE**
- This is sample website created for project demo**
- Welcome to our web site**
- A photograph of a group of people.
- A navigation menu with links: [Home](#), [About Us](#), and [Contact us](#).
- A circular logo with a stylized 'e'.
- A paragraph: "In our site you'll be able to find all the information about our company and our business, what we do and first of all how we can take care of you".
- A paragraph: "> This site allows you to contact us and ask for further information."
- A button: "To book tickets online".

The browser's status bar at the bottom shows "Status:" and "Progress:". The taskbar at the very bottom includes the Start button, several open applications (Web_Test_Cr..., Mail, Web Test, Untitled - Paint, final project d...), Norton, and the system clock showing 12:42 AM.

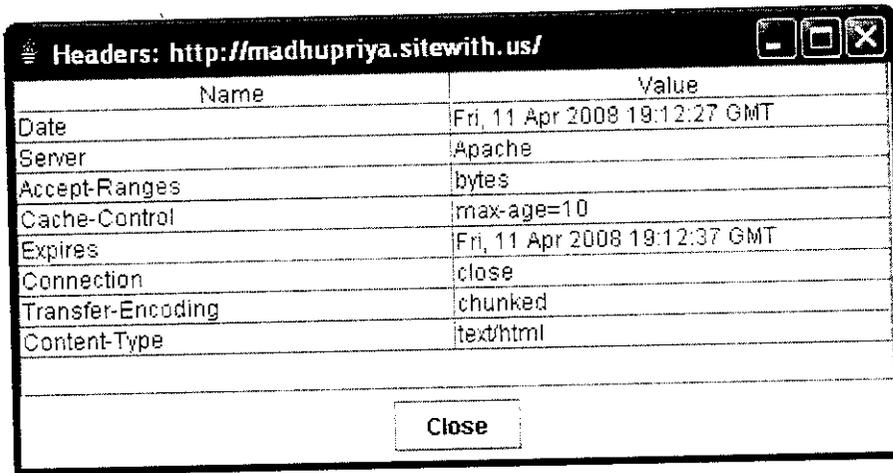
Cookies



The screenshot shows a "Cookies" dialog box with the following text:

```
B=96raddd3v27qh&b=3&s=98;Path=/
PREF=ID=ebcacae4ba7ec407:TM=1206984678:LM=1206984678:S=wuHtOzp68y2uSCDo;Path=/
PREF=ID=86418b0af846f39f:TM=1206984681:LM=1206984681:S=zesNaMLOgBTsbbul;Path=/
PREF=ID=00233efceb5ceeae6:TM=1206984735:LM=1206984735:S=R6sbmEYn7TXMsiP3;Path=/
PREF=ID=d55b15b7909e8132:TM=1206984778:LM=1206984778:S=w5t7Xy2J3wwBgeq;Path=/
PREF=ID=69954921b45c52af:TM=1206984946:LM=1206984946:S=k2_YT.Vk4v0vsafg;Path=/
PREF=ID=21d385c79bb6d99e:TM=1206984964:LM=1206984964:S=YB2Jn_PWUm11WCvU;Path=/'
```

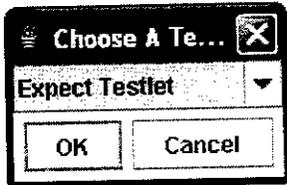
Header Information



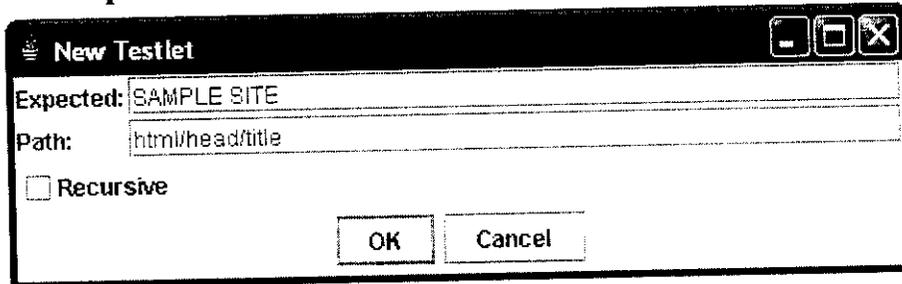
Name	Value
Date	Fri, 11 Apr 2008 19:12:27 GMT
Server	Apache
Accept-Ranges	bytes
Cache-Control	max-age=10
Expires	Fri, 11 Apr 2008 19:12:37 GMT
Connection	close
Transfer-Encoding	chunked
Content-Type	text/html

Close

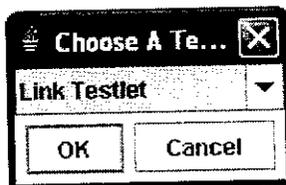
Expected testlet



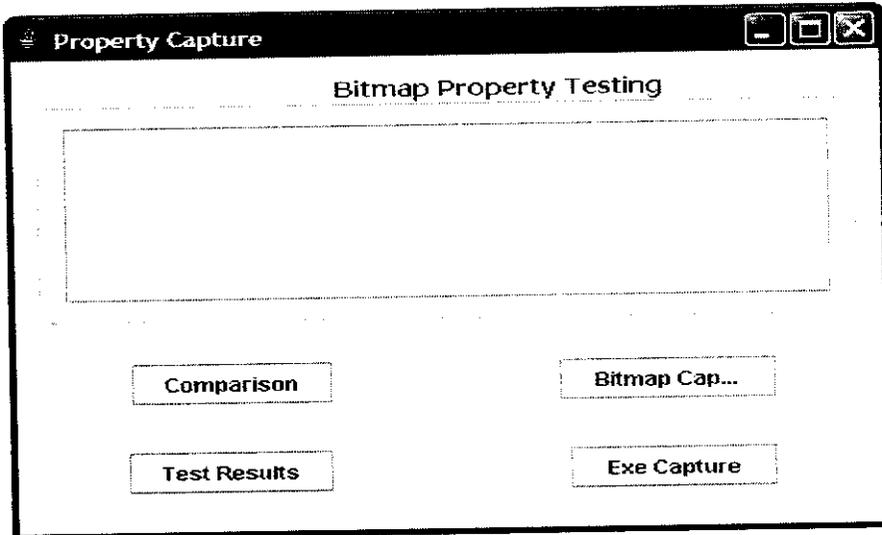
New Expected testlet



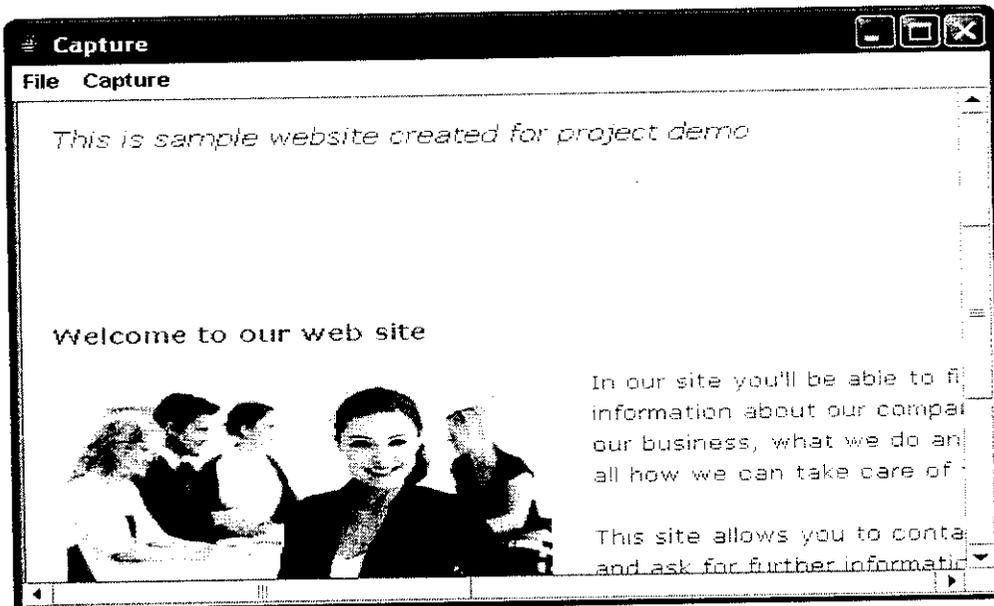
Link testlet



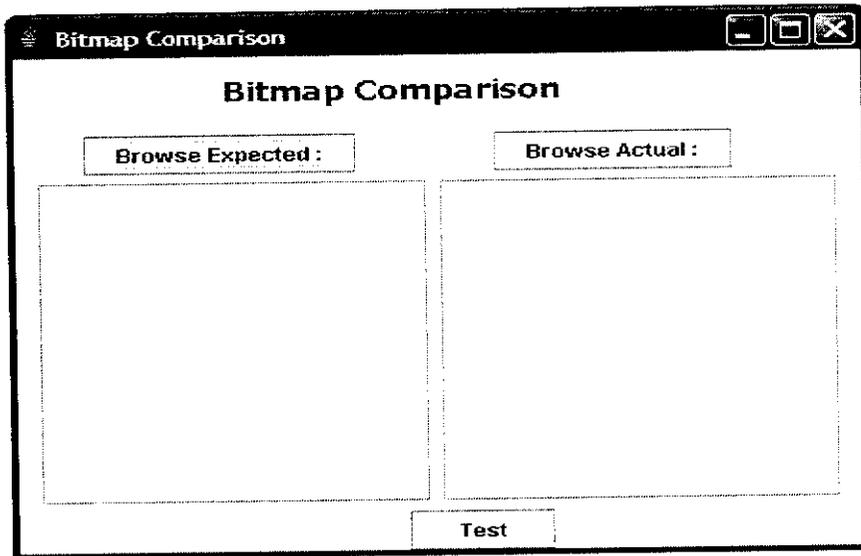
Bitmap property



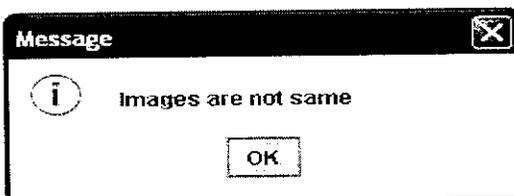
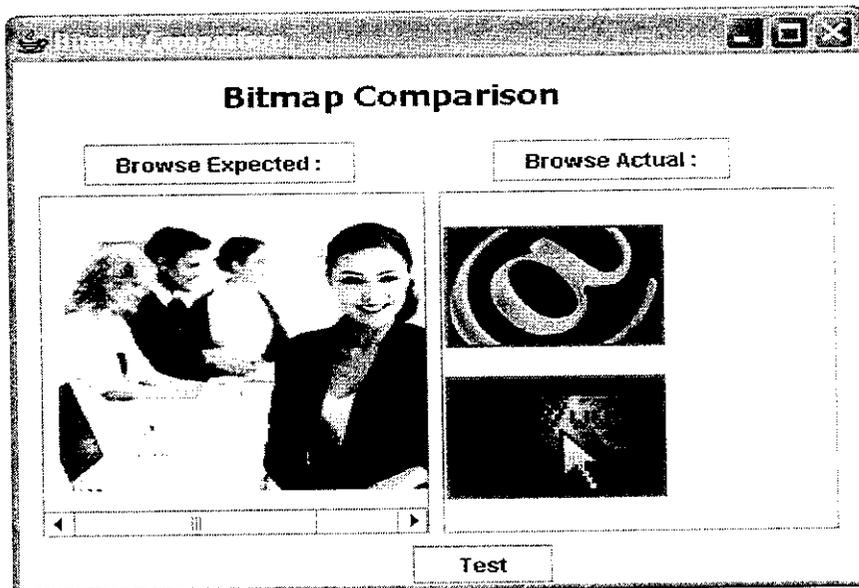
Capture window



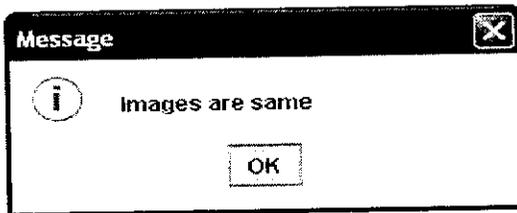
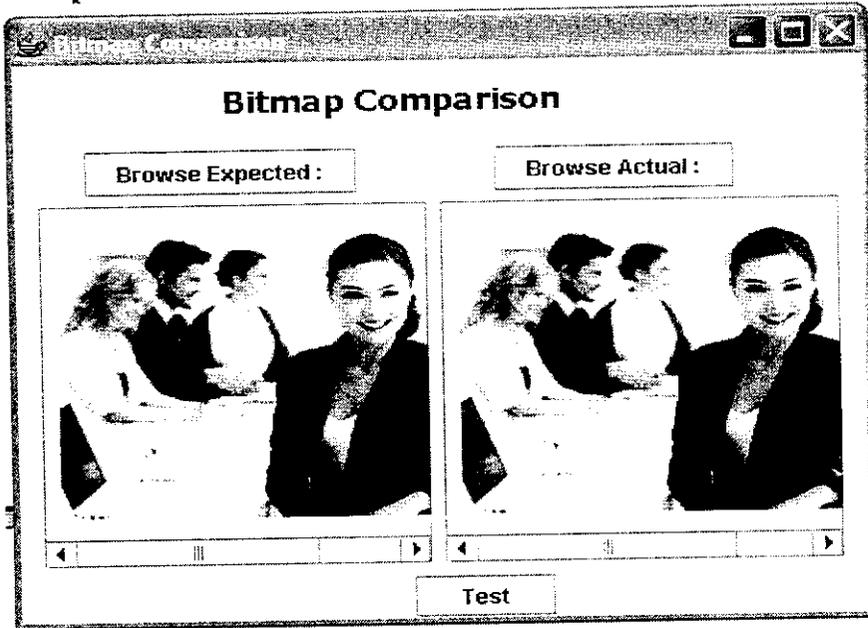
Bitmap comparison



Comparison – Example1



Comparison – Example 2



REFERENCES

1. D. Chays, S. Dan, P. Frankl, F. Vokolos, and E. Weyuker (Aug. 2005.), 'A Framework for Testing Database Applications' Proc. Int'l Symposium. Software Testing and Analysis, pp. 147-157.
2. J. Conallen (2003), 'Building Web Applications with UML', Addison-Wesley.
3. Dart, S. (2004), '*Configuration management: A missing link in Web engineering*', Norwood, MA: Artech House.
4. G. DiLucca, A. Fasolino, F. Faralli, and U. Carlini (Nov. 2002), "Testing Web Applications," Proc. Int'l Conf. Software Maintenance, pp. 310-319.
5. Gerrad, P. & Thompson, N. (2002). '*Risk-based e-business testing*', Artech Publishers.
6. E. Hieatt and R. Mee (Mar,2002), "Going Faster: Testing the Web Application,"IEEE Software, pp. 60-65.
7. Testing E-business Applications, Cutter IT Journal, September 2001.
8. <http://www.java.sun.com/>
9. <http://www.javascript-coder.com/>
10. <http://www.google.com/>
11. <http://www.testerbrian.com/>
12. <http://www.webrefernces.com/>