P-1617

# I-SPY WEB ROBOT

A PROJECT REPORT

Submitted by


| | |
|---|---|
| GOWRISELVI.M | 71202104012 |
| KAVITHA.B | 71202104017 |
| VISALAKSHI.J | 71202104053 |


In partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE &ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE-06

ANNA UNIVERSITY::CHENNAI 600 025

MAY 2006

# ANNA UNIVERSITY: CHENNAI 600 025

## BONAFIDE CERTIFICATE

Certified that this project report **"I-SPY WEB ROBOT"** is the bonafide work of "M.GOWRISELVI, B.KAVITHA, and J.VISALAKSHI" who carried out the project work under my supervision.

**SIGNATURE**
(Dr.S.Thangasamy)
**HEAD OF THE DEPARTMENT**
HOD CSE
Computer Science and Engineering,
Kumaraguru College of Technology,
Coimbatore – 641006.

**SIGNATURE**
(Mrs.R.Kalaiselvi)
**SUPERVISOR**
Senior Lecturer
Computer Science and Engineering,
Kumaraguru College of Technology,
Coimbatore – 641006

The candidates with University Register Nos. 71202104012, 71202104017 and 71202104053 were examined by us in the project viva-voce examination held on 2/5/06

**INTERNAL EXAMINER**

02/05/06
**EXTERNAL EXAMINER**

# DECLARATION

We hereby declare that the project entitled "**I-SPY WEB ROBOT**" original work done by us and to the best of our knowledge, a similar work has not been submitted to Anna University or any institutions, for fulfillment of the requirement of course study.

The report is submitted in partial fulfillment of the requirements for the award of the Degree of Bachelor of Computer Science and Engineering of Anna University, Chennai.

**Place :**Coimbatore.

**Date :**

M. Gowriselvi

**(M.Gowriselvi)**

B. Kavitha

**(B.Kavitha)**

J. Visalakshi

**(J.Visalakshi)**

# ACKNOWLEDGEMENT

# ACKNOWLEDGEMENT

We wish to express our sincere thanks to **Dr.K.K.Padmanabhan,** B.Sc(engg.), M.Tech., Ph.D., Principal, Kumaraguru College of Technology for having given us a golden opportunity to carry out this project.

We are deeply obliged to **Dr.S.Thangasamy,** Ph.D., Professor, Head of the Department of Computer Science and Engineering for his valuable guidance and useful suggestions during the course of this project.

We extend our heartfelt thanks to our course coordinator **Mrs.P.Devaki,** M.S., Assistant Professor, Department of Computer Science and Engineering and valuable support given to us throughout this project.

We are extremely thankful to our project guide **Mrs.R.Kalaiselvi,** B.E, Senior Lecturer, Department of Computer Science and Engineering, for her guidance and support for completing this project.

We thank our class advisor **Mrs.Amutha Venkatesan,** M.E., Lecturer, Department of Computer Science and Engineering, for providing us the moral support for completing this project. We thank the **teaching and non-teaching staff** of our department for providing us the technical support in the duration of our project.

Eventually we thank our family and friends for their motivation. We thank each and everyone who were directly or indirectly associated with this project.

# ABSTRACT

# ABSTRACT

The aim of the I-Spy project is to try to keep the user informed of any changes that may appear on this Information Superhighway. It does this by validating the links in the World Wide Web (WWW) pages, providing accurate statistical information of the pages visited.

The primary function of this application is to check all links of web pages within a domain to verify that the URL links and all the other types of links are valid. The other types of links checked are FTP files, CGI scripts, anchors, email addresses, images, maps, Java scripts, and Java applets. For URL links which are valid ,the application gives a speed rating for the link and other statistical information. A hierarchical map is drawn of the pages found, which also displays summary information for each page. A detailed error windows displays all errors found organised by domain, then page, then error type.

The secondary function is to instruct ISWR to scan a wider range of web addresses and produce a map of the structure of the various domains .

# TABLE OF CONTENTS

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

# LIST OF ABBREVIATIONS

| S.No | ABBREVIATION | EXPANSION |
|---|---|---|
| 1 | URL | Universal Resource Locator |
| 2 | CGI | Common Gateway Interface |
| 3 | FTP | File Transfer Protocol |
| 4 | GUI | Graphical User Interface |
| 5 | ISWR | I-Spy Web Robot |

# INTRODUCTION

# 1 INTRODUCTION

## 1.1 GENERAL OVERVIEW

It will provide with extensive analysis of the web domains and become an indispensable tool for finding all types of erroneous links. The primary function of this product is to provide the user with a platform independent application which will check all the links of web pages. The information will be collated and summarised and used for maintenance of web pages. A map window graphically displays the structure of a domain's web pages or the structure of the links between multiple domains.

## 1.2 PROBLEM DEFINITION

In the www pages there are large number of erroneous links. The invalid links are unknown to the user. This project is to validate the links in the world wide web pages and to provide the required statistical information to the user.

## 1.3 OBJECTIVE

The aim of the I-Spy project is to try to keep the user informed of any changes that may appear on this Information Superhighway. It does this by validating the links in the World Wide Web (WWW) pages, providing accurate statistical information of the pages visited.

# PROGRAMMING ENVIRONMENT

# 2 PROGRAMMING ENVIRONMENT

## 2.1 HARDWARE REQUIREMENTS

The hardware requirements for I-Spy are identical to the requirements for the Java virtual machine for the chosen performance. The hardware recommended is a Pentium class computer with 16 Mega bytes of RAM running a 32 bit operating system.

## 2.2 SOFTWARE REQUIREMENTS

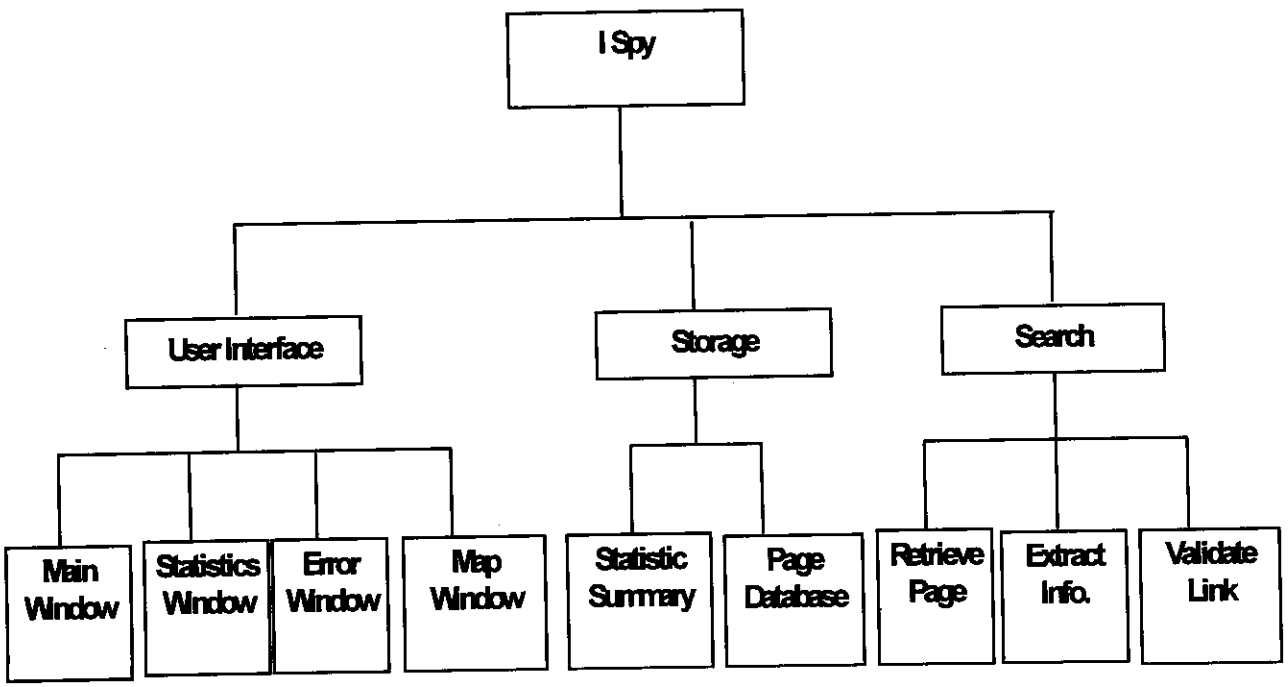
The I-Spy Web Robot requires a Java 1.5 runtime. Java 1.5 runtime environments are available for a wide variety of platforms including: Windows, OS/2, Macintosh, and UNIX.

# CODE STRUCTURE

# 3 CODE STRUCTURE

## 3.1 FUNCTIONAL MODULES

- **User Interface**: Graphical interface with the user, takes user inputs, and displays statistics, errors, and the map.
- **Search**: Retrieves an HTML page, finds and validates links, and collects information of the page.
- **Storage**: Interface to the database which also catches and summarises database information of the page.

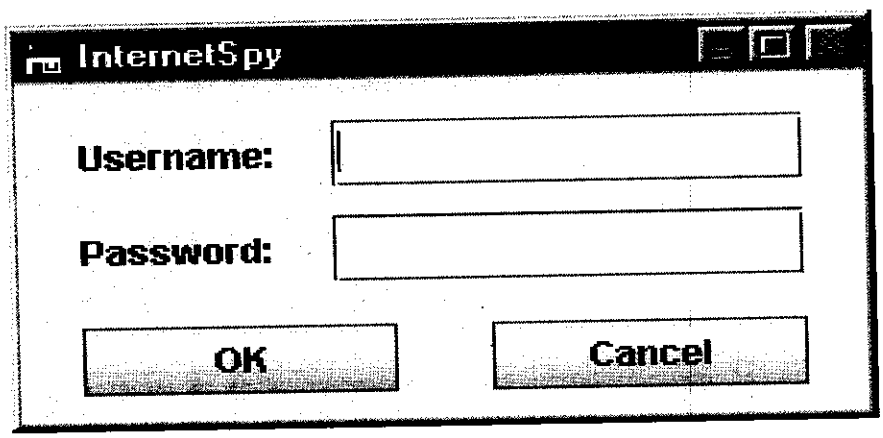



**3.1 General overview of I SPY**

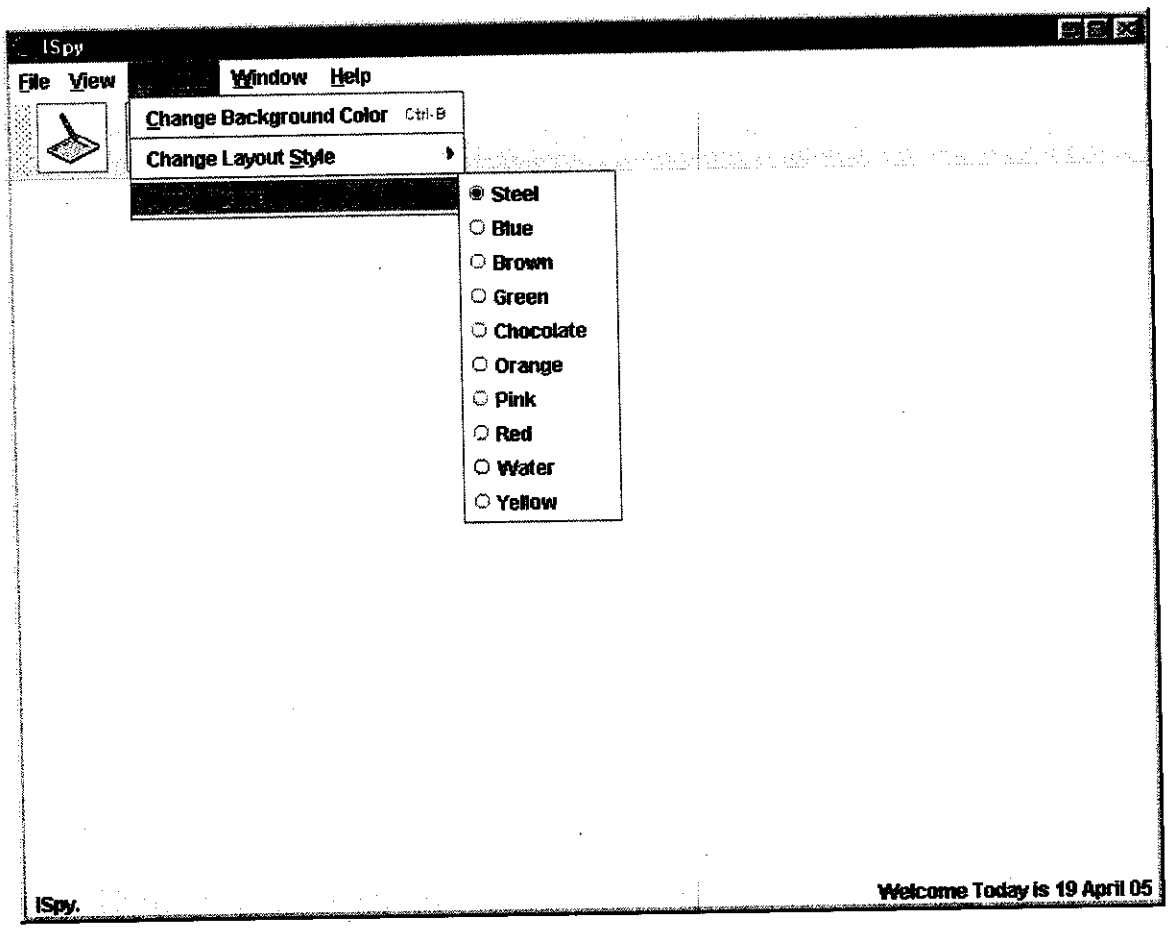
## 3.2 USER INTERFACE MODULE

### 3.2.1 MAIN WINDOW

It is the Main program control (user enters search information). The user input search parameters and other options and then start, stop and pause the search. Enter a search domain and accompanying text box, and Limit the search to these domains and accompanying list box are removed from the window. The user then enters a URL in the appropriate text box. The Type of links to test check boxes can then be unchecked or checked. Then the Start button may then be pressed to start the search.




**3.2 screenshot of Logon Window**

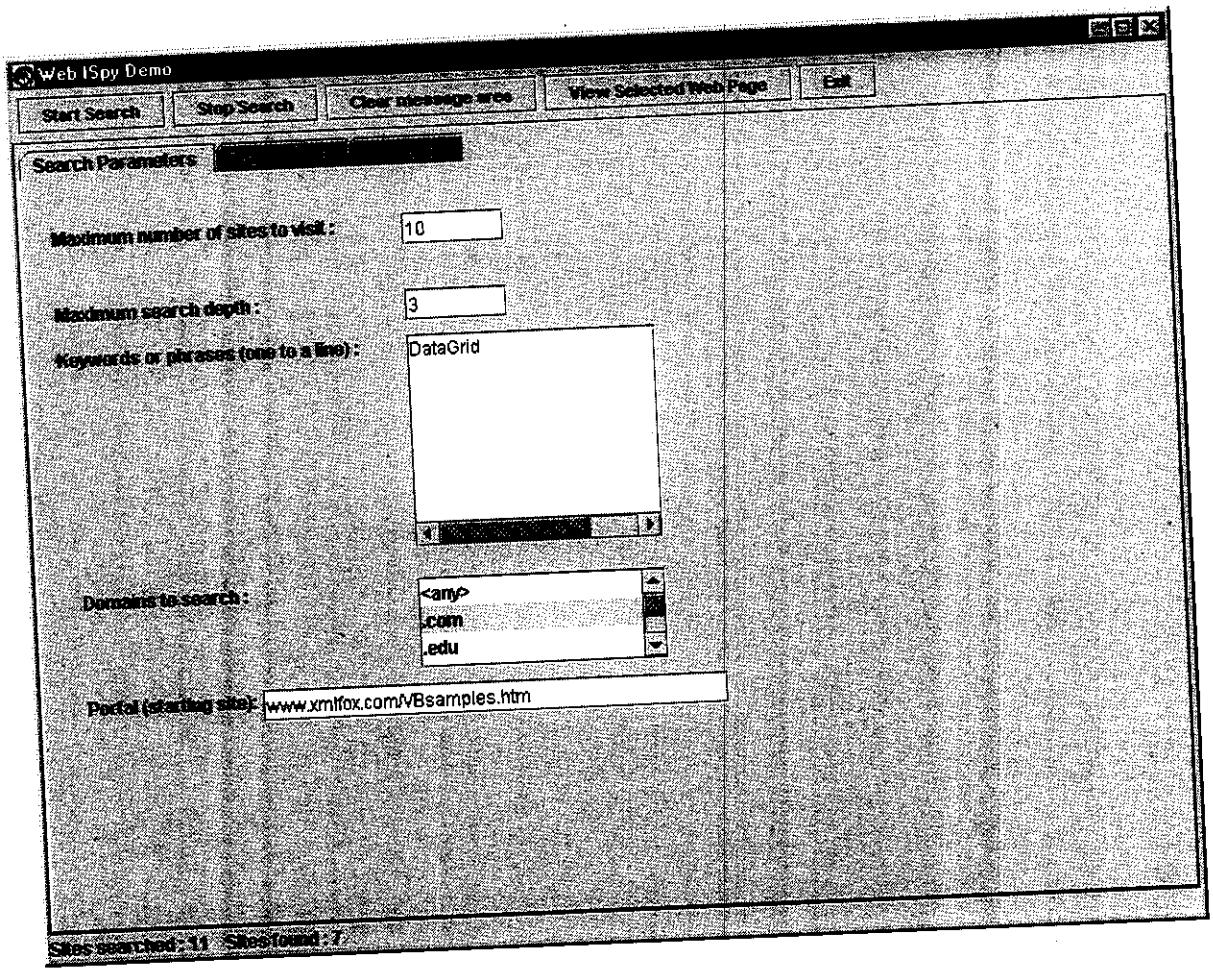



**3.3 Screenshot of main window**

## 3.3 SEARCH MODULE

Co-ordinates the URL searching - receives search parameters and calls the various sub classes until search is complete.

- Retrieve page: Retrieves an HTML page from the server.
- Extract Information: Scan the page and strips necessary data for the database.
- Validate: Check if the link is valid.




**3.5 screenshot of search parameters setup**

## 3.3.2 VIEWING THE PROGRESS OF THE SEARCH

During a search the searching window appears as figure 3.7. The window displays:

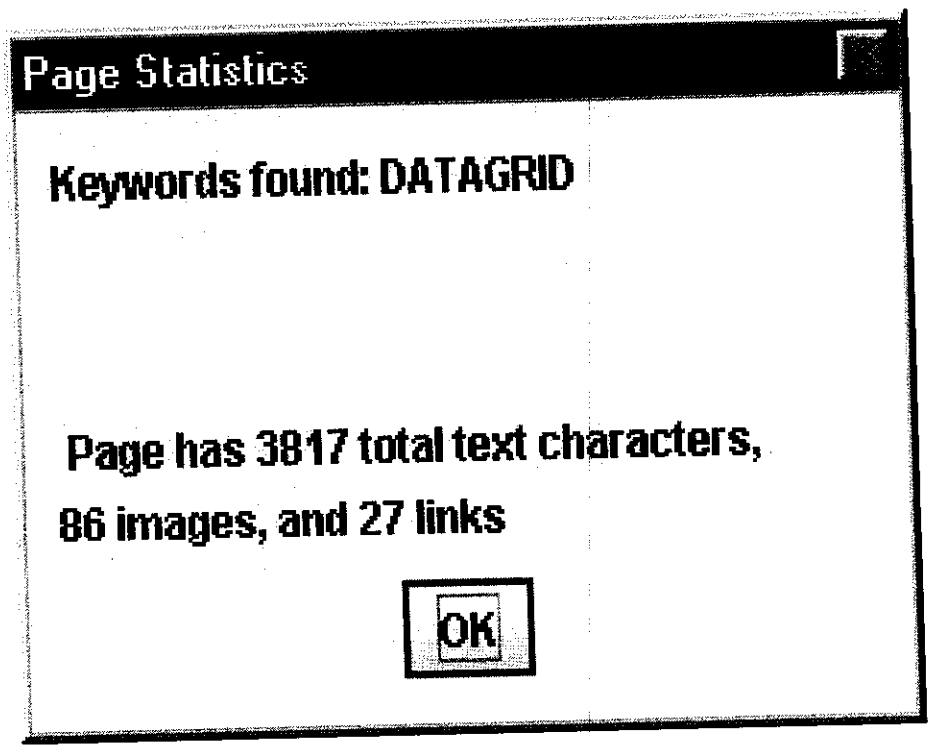
- The total number of sites found during search.
- The total number of levels visited during search.
- Page statistics

- Separate options such as stop search, view selected page, clear message area, save message, message and search tree.
- The links searched and the differentiation of the links for the sites having inputed keywords.

### 3.3.3 VIEWING RESULTS

#### 3.3.3.1 STATISTICS WINDOW

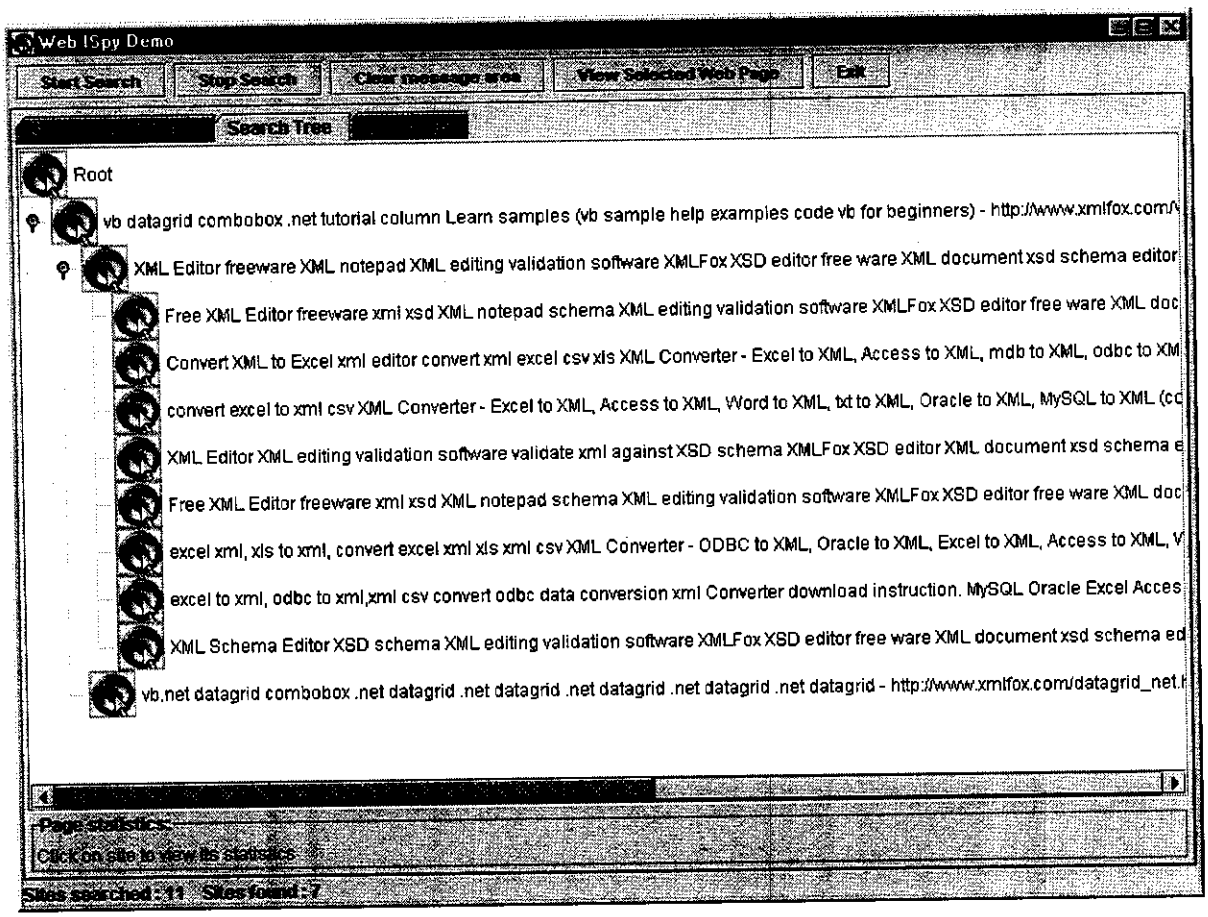
Displays a single page statistics and totals and average information for the entire search. The information includes the total number of images found, sites searched, levels searched, etc. To save the statistics information to disk press the Save button and you will be prompted with a standard save dialog box.




**3.6 screenshot of statistics window**

### 3.3.3.2 MAP WINDOW

Displays the pages found in a tree type view. If a page has any inputted keyword then the page will be in different colour. The required page can also be viewed from this page.

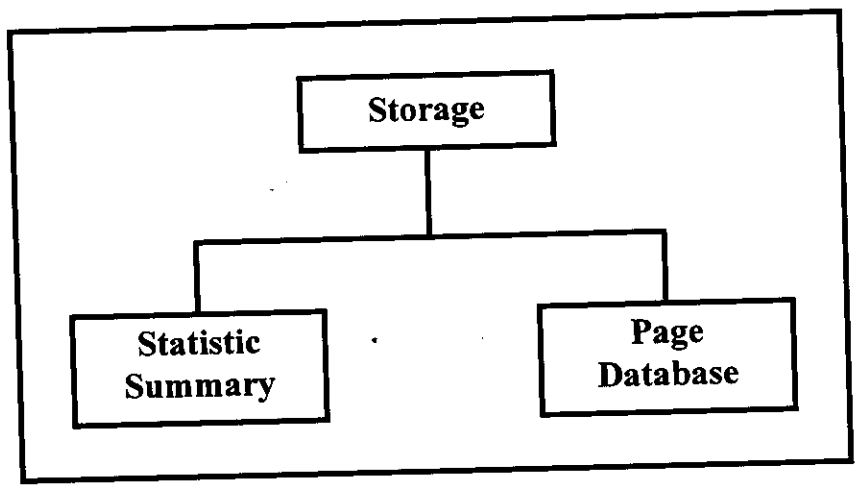



**3.7 screenshot of map window**

## 3.4 STORAGE MODULE

Receives search information and stores in Page database and statistics summary. Receives queries and returns the relevant data from the database.

- Statistics Summary: Average and totals information of the visited web pages: total number pages, links, size, and averages of links per page, errors per page and page size.
- Page database: Main database to store all data collected from search.




**3.8 Overview of storage module**

### 3.4.1 SAVING INFORMATION

When the **Save Statistics** or **Save Errors** menu item are selected a standard save dialog box for the operating system is displayed for the user to select the filename and directory to save the information.

### 3.4.2 STATISTICS SUMMARY

Average and totals information of the visited web pages: total number pages, links, size, and averages of links per page, errors per page and page size.

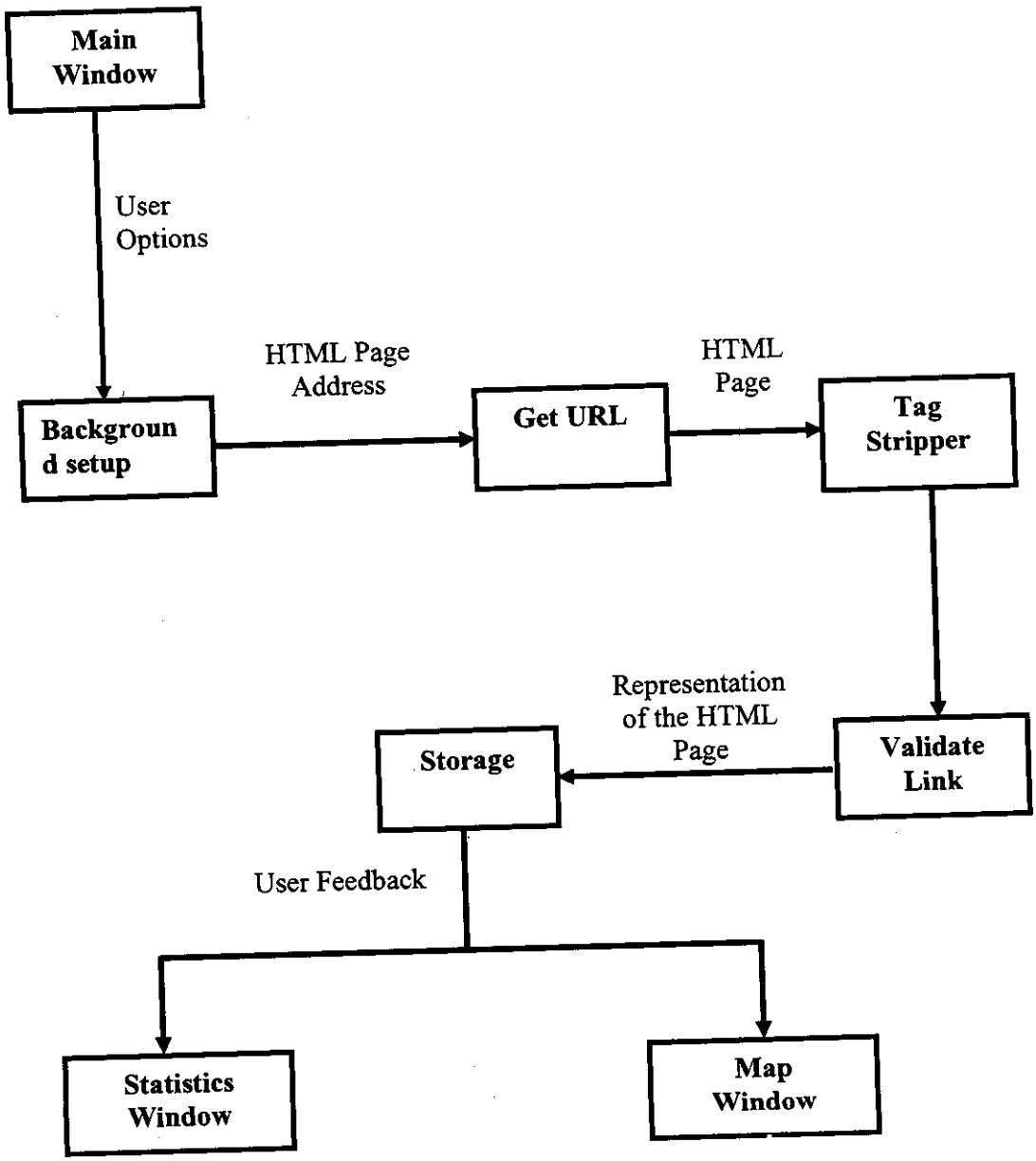
### 3.4.3 PAGE DATABASE

Main database to store all data collected from search.

# SYSTEM DESIGN

# 4 SYSTEM DESIGN

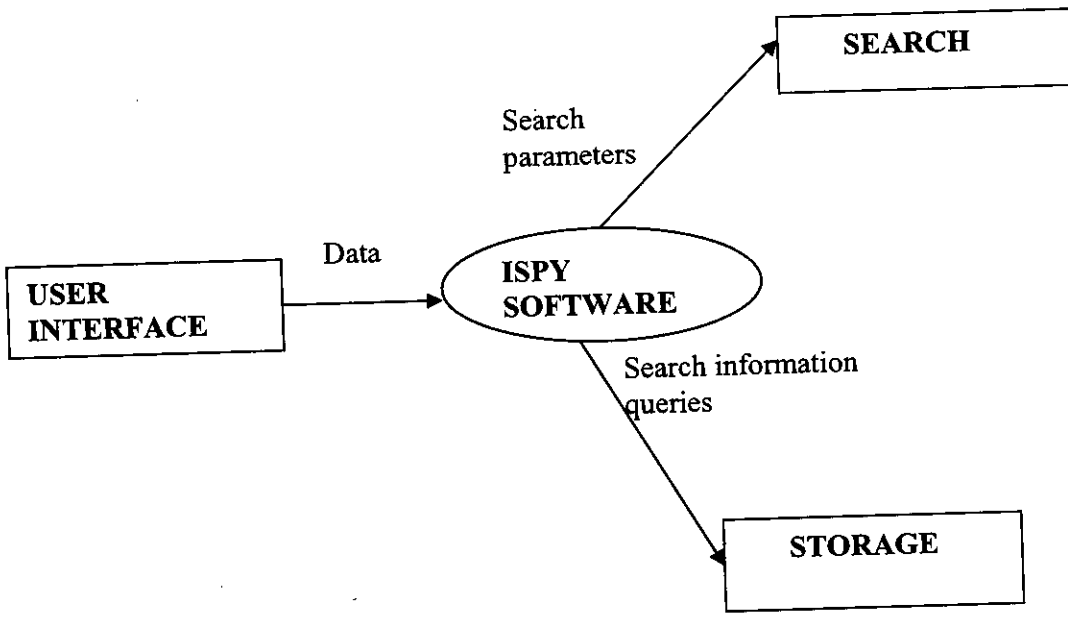
## 4.1 GENERAL FLOW DIAGRAM




4.1 General flow diagram

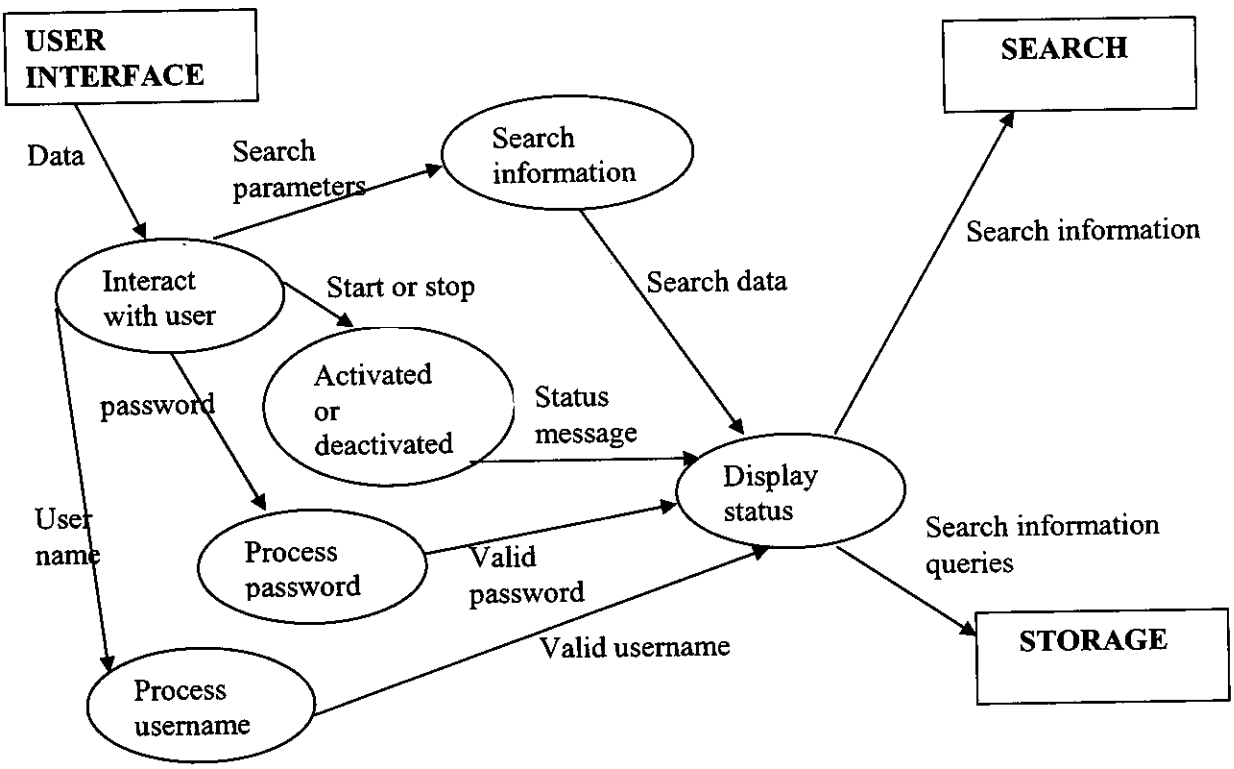
## 4.2 DATAFLOW DIAGRAM

### 4.2.1 LEVEL 0




4.2 LEVEL 0 DFD

### 4.2.2 LEVEL 1




**4.3 Level 1 DFD**

**SPECIFIC REQUIREMENTS**
**TESTING**
**CONCLUSION**

# 5 SPECIFIC REQUIREMENTS

## 5.1 INPUT DATA

URL, Search Domain, Types of links to search for and other user defined options.

## 5.2 OUTPUT DATA

| | |
|---|---|
| Current Page | : Current URL, Page Title, Size of web page, word count, Date last modified, page Speed, Number of links for each link type. |
| Totals | : Pages visited, Page Size, Links Found, Error Links. |
| Averages | : Page Size, Links per page, Error links per Page, speed of links. |

# 6 TESTING

Individual module testing which will test the integrity of the module. Integration testing,System testing which will test for over all functionality.

# 7 CONCLUSION

The primary function of this application is to check all links of web pages within a domain to verify that the URL links and all the other types of links are valid .The application may continue to search indefinitely if the program keeps finding new web pages. A search can be stopped either automatically or manually. To stop the search manually simply press the stop button. To stop the search automatically the application can be given a maximum level to recurse to, a maximum number of pages to find, or a maximum number of errors to find.

# FUTURE ENHANCEMENTS

# APPENDICES

# 8 FUTURE ENHANCEMENTS

- The ability to check a wider range of links
- Increasing the speed of the verification
- Alternative maps styles
- Sounds
- Improved splash screen

# 9 APPENDICES

## 9.1 SAMPLE CODE

### USER INTERFACE

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.util.*;
import java.text.*;

public class Logon extends JFrame implements ActionListener
{
private Dimension d = Toolkit.getDefaultToolkit().getScreenSize();
private JPanel pLog = new JPanel();
private JLabel lbUser, lbPass;
private JTextField txtUser;
private JPasswordField txtPass;
private JButton btnOk, btnCancel;
public boolean verify;
public String user;
public String pass;

 public Logon () {

    super ("ispy");
```

```
setIconImage (getToolkit().getImage ("Images/Home.gif"));
setSize (275, 150);
setResizable (false);
addWindowListener (new WindowAdapter () {
                public void windowClosing (WindowEvent we) {
                        setVisible (false);
                        dispose();
System.exit (0);                                        }
        }
        );

setLocation (d.width / 2 - getWidth() / 2, d.height / 2 -    getHeight() / 2);



        pLog.setLayout (null);



        lbUser = new JLabel ("Username:");
        lbUser.setForeground (Color.black);
        lbUser.setBounds (20, 15, 75, 25);
        lbPass = new JLabel ("Password:");
        lbPass.setForeground (Color.black);
        lbPass.setBounds (20, 50, 75, 25);

        txtUser = new JTextField ();
        txtUser.setBounds (100, 15, 150, 25);
        txtPass = new JPasswordField ();
        txtPass.setBounds (100, 50, 150, 25);

        btnOk = new JButton ("OK");
        btnOk.setBounds (20, 90, 100, 25);
        btnOk.addActionListener (this);
        btnCancel = new JButton ("Cancel");
        btnCancel.setBounds (150, 90, 100, 25);
        btnCancel.addActionListener (this);
```

```
            pLog.add (lbUser);
            pLog.add (lbPass);
            pLog.add (txtUser);
            pLog.add (txtPass);
            pLog.add (btnOk);
            pLog.add (btnCancel);


            getContentPane().add (pLog);


            setVisible (true);

}

public void actionPerformed (ActionEvent ae) {

Object obj = ae.getSource();

 if (obj == btnOk)
  {

    String password = new String (txtPass.getPassword());
    if (txtUser.getText().equals ("")) {
    JOptionPane.showMessageDialog (this, "Provide Username to Logon.");
    txtUser.requestFocus();
                    }
                    else if (password.equals (""))
   {
                        txtPass.requestFocus();
                        JOptionPane.showMessageDialog (null,"Provide
 Password to Logon.");
                    }
    else
    {

     verify = false;
     user = "user";
```

```
pass = "ispy";
 }

 if (txtUser.getText().equals ("user") && password.equals ("ispy"))
  {                                        verify = true;
                                           setVisible (false);
                                           dispose();
  }

          if (verify == false)
           {
                                           JOptionPane.showMessageDialog (this,
"Incorrect Information Provided.");
                                           txtUser.setText ("");
                                           txtPass.setText ("");
                                           txtUser.requestFocus ();
           }
                    }


     else if (obj == btnCancel)
     {

           setVisible (false);
           dispose();
           System.exit (0);

              }

        }

 public static void main(String args[])
 {
 Logon spy=new Logon();
 }
 }
```

**SEARCH**

```
import java.awt.*;
import javax.swing.*;
import javax.swing.tree.*;
import java.util.*;
import java.net.*;
public class ISpyControl extends javax.swing.JFrame implements
VerifierListener{

      public ISpyControl() {
      initComponents();
      setSize(800,600);
      setTitle("Web ISpy Demo");

     Dimension oursize = getSize();
     Dimension screensize = Toolkit.getDefaultToolkit().getScreenSize();

      int x = (screensize.width - oursize.width)/2;
      int y = (screensize.height- oursize.height)/2;
      x = Math.max(0,x);
      y = Math.max(0,y);        setLocation(x,y);

     DefaultMutableTreeNode root = new DefaultMutableTreeNode("Empty");
     DefaultTreeModel treeModel = new DefaultTreeModel(root);
     searchTree.setModel(treeModel);

      URL iconurl = getClass().getResource("ISpy.gif");      if(iconurl != null)
      {
       ImageIcon ic = new ImageIcon(iconurl);
       setIconImage(ic.getImage());          }

   }


      private void initComponents()
      toolBar = new javax.swing.JPanel();
      startButton = new javax.swing.JButton();
      stopButton = new javax.swing.JButton();
      clearMessageButton = new javax.swing.JButton();
```

```
viewButton = new javax.swing.JButton();
exitButton = new javax.swing.JButton();
centerPane = new javax.swing.JTabbedPane();
formTab = new javax.swing.JPanel();
siteLabel = new javax.swing.JLabel();
siteField = new javax.swing.JTextField();
depthLabel = new javax.swing.JLabel();
depthField = new javax.swing.JTextField();
keywordLabel = new javax.swing.JLabel();
keywordPane = new javax.swing.JScrollPane();
keywordArea = new javax.swing.JTextArea();
domainLabel = new javax.swing.JLabel();
domainPane = new javax.swing.JScrollPane();
domainList = new javax.swing.JList();
startingLabel = new javax.swing.JLabel();
errorLabel = new javax.swing.JLabel();
jTextArea1 = new javax.swing.JTextArea();
startSiteField = new javax.swing.JTextField();
treeTab = new javax.swing.JPanel();
searchTreePane = new javax.swing.JScrollPane();
searchTree = new javax.swing.JTree();
pageStatistics = new javax.swing.JTextArea();
messageTab = new javax.swing.JScrollPane();
messageArea = new javax.swing.JTextArea();
statusLabel = new javax.swing.JLabel();

setBackground(new java.awt.Color(153, 153, 255));
addWindowListener(new java.awt.event.WindowAdapter() {
   public void windowClosing(java.awt.event.WindowEvent evt) {
      exitForm(evt);
   }
});

toolBar.setLayout(new java.awt.FlowLayout(java.awt.FlowLayout.LEFT));

toolBar.setBackground(new java.awt.Color(204, 204, 204));
startButton.setFont(new java.awt.Font("Arial", 1, 11));
startButton.setText("Start Search");
startButton.setToolTipText("Start the search");
startButton.addActionListener(new java.awt.event.ActionListener() {
```

```
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            startButtonActionPerformed(evt);
        }
    });

    toolBar.add(startButton);

    stopButton.setFont(new java.awt.Font("Arial", 1, 11));
    stopButton.setText("Stop Search");
    stopButton.setToolTipText("Stop the search that is in progress");
    stopButton.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            stopButtonActionPerformed(evt);
        }
    });

    toolBar.add(stopButton);

    clearMessageButton.setFont(new java.awt.Font("Arial", 1, 11));
    clearMessageButton.setText("Clear message area");
    clearMessageButton.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            clearMessageButtonActionPerformed(evt);
        }
    });

    toolBar.add(clearMessageButton);

    viewButton.setFont(new java.awt.Font("Arial", 1, 11));
    viewButton.setText("View Selected Web Page");
    viewButton.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            viewButtonActionPerformed(evt);
        }
    });

    toolBar.add(viewButton);

    exitButton.setFont(new java.awt.Font("Arial", 1, 11));
```

```
exitButton.setText("Exit");
exitButton.addActionListener(new java.awt.event.ActionListener() {
   public void actionPerformed(java.awt.event.ActionEvent evt) {
      exitButtonActionPerformed(evt);
   }
});

toolBar.add(exitButton);

getContentPane().add(toolBar, java.awt.BorderLayout.NORTH);

centerPane.setBorder(new javax.swing.border.EtchedBorder());
formTab.setLayout(null);

formTab.setBackground(new java.awt.Color(204, 204, 204));
siteLabel.setText("Maximum number of sites to visit : ");
formTab.add(siteLabel);
siteLabel.setBounds(20, 30, 250, 15);

siteField.setColumns(8);
siteField.setText("100");
siteField.setInputVerifier(new IntegerVerifier(this,false,1,10000)
);
formTab.add(siteField);
siteField.setBounds(260, 30, 70, 21);

depthLabel.setText("Maximum search depth : ");
formTab.add(depthLabel);
depthLabel.setBounds(20, 80, 230, 15);

depthField.setColumns(8);
depthField.setText("10");
depthField.setInputVerifier(new IntegerVerifier(this,false,1,10000)
);
formTab.add(depthField);
depthField.setBounds(260, 80, 70, 21);

keywordLabel.setText("Keywords or phrases (one to a line) :");
formTab.add(keywordLabel);
keywordLabel.setBounds(20, 110, 220, 15);
```

```
        keywordArea.setColumns(20);
        keywordPane.setViewportView(keywordArea);

        formTab.add(keywordPane);
        keywordPane.setBounds(260, 110, 170, 140);

        domainLabel.setText("Domains to search : ");
        formTab.add(domainLabel);
        domainLabel.setBounds(30, 270, 120, 15);

        domainList.setModel(new javax.swing.AbstractListModel() {
            String[] strings = { "<any>", ".com", ".edu", ".gov", ".int", ".mil", ".net",
".org", ".us", ".ca" };
            public int getSize() { return strings.length; }
            public Object getElementAt(int i) { return strings[i]; }
        });
        domainList.setSelectedIndex(0);
        domainPane.setViewportView(domainList);

        formTab.add(domainPane);
        domainPane.setBounds(260, 270, 170, 60);

        startingLabel.setText("Portal (starting site): ");
        formTab.add(startingLabel);
        startingLabel.setBounds(30, 340, 120, 15);

        errorLabel.setForeground(new java.awt.Color(255, 51, 51));

    errorLabel.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        errorLabel.setText("  ");
        formTab.add(errorLabel);
        errorLabel.setBounds(30, 395, 400, 20);

        formTab.add(jTextArea1);
        jTextArea1.setBounds(240, 360, 0, 17);

        startSiteField.setColumns(80);
        formTab.add(startSiteField);
        startSiteField.setBounds(150, 340, 320, 21);
```

```
        centerPane.addTab("Search Parameters", formTab);

        treeTab.setLayout(new java.awt.BorderLayout());

        searchTreePane.setBackground(new java.awt.Color(204, 204, 204));
        searchTreePane.setBorder(new javax.swing.border.EtchedBorder());
        searchTree.addTreeSelectionListener(new
javax.swing.event.TreeSelectionListener() {
            public void valueChanged(javax.swing.event.TreeSelectionEvent evt) {
                searchTreeSelectionChange(evt);
            }
        });

        searchTreePane.setViewportView(searchTree);

        treeTab.add(searchTreePane, java.awt.BorderLayout.CENTER);

        pageStatistics.setBackground(new java.awt.Color(204, 204, 204));
        pageStatistics.setColumns(80);
        pageStatistics.setRows(2);
        pageStatistics.setText("Select item to display its statistics");
        pageStatistics.setBorder(new javax.swing.border.TitledBorder("Page
statistics:"));
        treeTab.add(pageStatistics, java.awt.BorderLayout.SOUTH);

        centerPane.addTab("Search Tree", treeTab);

        messageTab.setBackground(new java.awt.Color(204, 204, 204));
        messageTab.setBorder(new javax.swing.border.EtchedBorder());

messageTab.setVerticalScrollBarPolicy(javax.swing.JScrollPane.VERTICAL_S
CROLLBAR_ALWAYS);
        messageArea.setColumns(100);
        messageArea.setRows(5);
        messageTab.setViewportView(messageArea);

        centerPane.addTab("Messages", messageTab);

        getContentPane().add(centerPane, java.awt.BorderLayout.CENTER);
```

```
    statusLabel.setText("Inactive");
    getContentPane().add(statusLabel, java.awt.BorderLayout.SOUTH);

    pack();
  }
  private void searchTreeSelectionChange(javax.swing.event.TreeSelectionEvent
evt) {
      TreePath path = searchTree.getSelectionPath();
      if(path == null)
        return;
      DefaultMutableTreeNode node =
 (DefaultMutableTreeNode)path.getLastPathComponent();
      UrlTreeNode data = (UrlTreeNode)node.getUserObject();
      if(data != null && data instanceof UrlTreeNode)
      {
       String kstr = data.getKeywords();
       pageStatistics.setText("Keywords found : "+kstr+"\n");
       pageStatistics.append(data.getNodeStats());
       }
       else
        pageStatistics.setText("");
  }

       private void viewButtonActionPerformed(java.awt.event.ActionEvent evt) {
      try{
       TreePath path = searchTree.getSelectionPath();
       if(path == null)
          return;
       DefaultMutableTreeNode node =
   (DefaultMutableTreeNode)path.getLastPathComponent();
       UrlTreeNode data = (UrlTreeNode)node.getUserObject();
        if(data instanceof UrlTreeNode)
        {
        String urlstr = data.getUrlString();
        Runtime.getRuntime().exec("C:\\program files\\Internet
    Explorer\\iexplore.exe "+urlstr);
        }
        }
        catch(Exception e)
```

```
    {
    JOptionPane.showMessageDialog(this,"Could not launch Internet
Explorer","Error",JOptionPane.ERROR_MESSAGE);
    }
}
    private void
clearMessageButtonActionPerformed(java.awt.event.ActionEvent evt) {
    messageArea.setText("");
}
    private void stopButtonActionPerformed(java.awt.event.ActionEvent evt) {
    if(spidey != null)
      spidey.stopSearch();
  }
  private void startButtonActionPerformed(java.awt.event.ActionEvent evt)
 {
   int sitelimit =100,  depthlimit=100;

     try{
      sitelimit = Integer.parseInt(siteField.getText().trim()) ;
      depthlimit = Integer.parseInt(depthField.getText().trim()) ;
     }
     catch(NumberFormatException e)
     {
       errorLabel.setText("Invalid input for site limit or depth limit");
       return;
     }

       Object selected[] = domainList.getSelectedValues();
       String[] domains = new String[selected.length];
       for(int i = 0; i < domains.length; i++)
        domains[i]= selected[i].toString();

       StringTokenizer keywordtokens = new
  StringTokenizer(keywordArea.getText(), "\n");
       String keywords[] = new String[keywordtokens.countTokens()];
       int i = 0;
       while(keywordtokens.hasMoreTokens())
        keywords[i++] = keywordtokens.nextToken();

       String startsite = startSiteField.getText();
```

```
        if(startsite.length() <= 0)
        {
         errorLabel.setText("Starting web site cannot be blank!");
         return;
        }

        errorLabel.setText("");
        pageStatistics.setText("Click on site to view its statistics");
        centerPane.setSelectedIndex(1);
        spidey = new ISpy(searchTree, messageArea, statusLabel, startsite,
keywords, domains, sitelimit,depthlimit);
        spidey.start();

    }

    private void exitButtonActionPerformed(java.awt.event.ActionEvent evt) {
        System.exit(0);
    }
        private void exitForm(java.awt.event.WindowEvent evt) {
 System.exit(0);
    }

    public static void main(String args[]) {
        new ISpyControl().show();
    }

    public void invalidData(String message, JComponent jComponent) {
        errorLabel.setText(message);
        errorLabel.setForeground(Color.red);
        startButton.setEnabled(false);
        getToolkit().beep();
    }


    public void validData(JComponent jComponent) {
        errorLabel.setText("  ");
        startButton.setEnabled(true);  }

    private javax.swing.JTabbedPane centerPane;
    private javax.swing.JButton clearMessageButton;
```

```
private javax.swing.JTextField depthField;
private javax.swing.JLabel depthLabel;
private javax.swing.JLabel domainLabel;
private javax.swing.JList domainList;
private javax.swing.JScrollPane domainPane;
private javax.swing.JLabel errorLabel;
private javax.swing.JButton exitButton;
private javax.swing.JPanel formTab;
private javax.swing.JTextArea jTextArea1;
private javax.swing.JTextArea keywordArea;
private javax.swing.JLabel keywordLabel;
private javax.swing.JScrollPane keywordPane;
private javax.swing.JTextArea messageArea;
private javax.swing.JScrollPane messageTab;
private javax.swing.JTextArea pageStatistics;
private javax.swing.JTree searchTree;
private javax.swing.JScrollPane searchTreePane;
private javax.swing.JTextField siteField;
private javax.swing.JLabel siteLabel;
private javax.swing.JButton startButton;
private javax.swing.JTextField startSiteField;
private javax.swing.JLabel startingLabel;
private javax.swing.JLabel statusLabel;
private javax.swing.JButton stopButton;
private javax.swing.JPanel toolBar;
private javax.swing.JPanel treeTab;
private javax.swing.JButton viewButton;

ISpy spidey = null;
}



import java.util.*;
import java.io.*;
import java.net.*;
import javax.swing.*;
import javax.swing.tree.*;
import javax.swing.text.html.parser.*;
```

```
import javax.swing.text.html.HTMLEditorKit.*;
import javax.swing.text.html.*;
import javax.swing.text.*;

public class ISpy extends Thread{

private int siteLimit = 100;
private int depthLimit = 100;
private String keywordList[];
private String ipDomainList[];
private JTree searchTree = null;
private JTextArea messageArea;
private JLabel statsLabel;
private int sitesSearched = 0;
private int sitesFound = 0;

private String startSite;

private boolean stopSearch = false;
ublic ISpy(JTree atree, JTextArea amessagearea,JLabel astatlabel, String
astartsite, String[] akeywordlist, String[] aipdomainlist, int asitelimit, int
adepthlimit) {
    searchTree = atree;
    messageArea = amessagearea;
 statsLabel = astatlabel;
    startSite = fixHref(astartsite);

    keywordList = new String[akeywordlist.length];
    for(int i = 0; i< akeywordlist.length; i++)
      keywordList[i] = akeywordlist[i].toUpperCase();
    ipDomainList = new String[aipdomainlist.length];
    for(int i = 0; i< aipdomainlist.length; i++)
      ipDomainList[i] = aipdomainlist[i].toUpperCase();
    siteLimit = asitelimit;
    depthLimit = adepthlimit; DefaultMutableTreeNode root = new
 DefaultMutableTreeNode(new UrlTreeNode("Root"));
    DefaultTreeModel treeModel = new DefaultTreeModel(root);
 searchTree.setModel(treeModel);
    searchTree.setCellRenderer(new UrlNodeRenderer()); }
```

```
    public void run()
    {
     DefaultTreeModel treeModel = (DefaultTreeModel)searchTree.getModel();
DefaultMutableTreeNode root = (DefaultMutableTreeNode)treeModel.getRoot();
     String urllc = startSite.toLowerCase();
     if(!urllc.startsWith("http://") && !urllc.startsWith("ftp://") &&
        !urllc.startsWith("www."))
      {
       startSite = "file:///"+startSite       }
      else        if(urllc.startsWith("www."))
       {
        startSite = "http://"+startSite;
       }

      startSite = startSite.replace('\\', '/');
      sitesFound = 0;
      sitesSearched = 0;
      updateStats();
      searchWeb(root,startSite);
      messageArea.append("Done!\n\n");
     }
  public boolean urlHasBeenVisited(String urlstring)
  {

    String teststring = fixHref(urlstring);
    DefaultTreeModel treeModel = (DefaultTreeModel)searchTree.getModel();
    DefaultMutableTreeNode root =
  (DefaultMutableTreeNode)treeModel.getRoot();
    Enumeration etree = root.breadthFirstEnumeration();
    while(etree.hasMoreElements())
    {
     UrlTreeNode node =
  (UrlTreeNode)(((DefaultMutableTreeNode)etree.nextElement()).getUserObject()
  );
      if(node instanceof UrlTreeNode && node.equals(teststring))
       return true;
    }
    return false;
  }
    public boolean depthLimitExceeded(DefaultMutableTreeNode node)
```

```
{

 if(node.getLevel() >= depthLimit)
    return true;
 else
    return false;

}

private DefaultMutableTreeNode addNode(DefaultMutableTreeNode
parentnode, UrlTreeNode newnode)
{
     DefaultMutableTreeNode node = new DefaultMutableTreeNode(newnode);
     DefaultTreeModel treeModel = (DefaultTreeModel)searchTree.getModel();
int index = treeModel.getChildCount(parentnode);
treeModel.insertNodeInto(node, parentnode,index);
     TreePath tp = new TreePath(parentnode.getPath());
     searchTree.expandPath(tp);
     return node;

 }

 private boolean isDomainOk(URL url)
 {
  if(url.getProtocol().equals("file"))
     return true;
   String host = url.getHost();
   int lastdot = host.lastIndexOf(".");
   if(lastdot <= 0)
     return true;

   String domain = host.substring(lastdot);
   if(ipDomainList.length == 0)
     return true;

   for(int i=0; i < ipDomainList.length; i++)
   {
    if(ipDomainList[i].equalsIgnoreCase("<any>"))
       return true;
    if(ipDomainList[i].equalsIgnoreCase(domain))
```

```
        return true;
    }
    return false;

}
private void updateStats()
{
    statsLabel.setText("Sites searched : "+sitesSearched+"    Sites found :
"+sitesFound);

}

    public static String fixHref(String href)
    {
     String newhref = href.replace('\\', '/');
     int lastdot = newhref.lastIndexOf('.');
     int lastslash = newhref.lastIndexOf('/');
     if(lastslash > lastdot)
     {
      if(newhref.charAt(newhref.length()-1) != '/')
        newhref = newhref+"/";
     }

     return newhref;

     }

    public void searchWeb(DefaultMutableTrecNode parentnode, String urlstr)
   {

     if(urlHasBeenVisited(urlstr))
       return;
     if(depthLimitExceeded(parentnode))
        return;

     if(sitesSearched > siteLimit)
        return;

     yield();
     if(stopSearch)
```

```
 return;

 messageArea.append("Searching :"+urlstr+" \n");
 sitesSearched++;
 updateStats();

 try{
   URL url = new URL(urlstr);
   String protocol = url.getProtocol();    if(!protocol.equalsIgnoreCase("http")
&& !protocol.equalsIgnoreCase("file"))
   {
     messageArea.append("    Skipping : "+urlstr+" not a http site\n\n");
     return;
   }

   String path = url.getPath();
   int lastdot = path.lastIndexOf(".");    if(lastdot > 0)
   {
    String extension = path.substring(lastdot);
if(!extension.equalsIgnoreCase(".html") &&
!extension.equalsIgnoreCase(".htm"))
      return;
   }

   if(!isDomainOk(url))
   {
     messageArea.append("    Skipping : "+urlstr+" not in domain list\n\n");
     return;
   }


    UrlTreeNode newnode = new UrlTreeNode(url);    InputStream in =
 url.openStream();    InputStreamReader isr = new InputStreamReader(in);
    DefaultMutableTreeNode treenode = addNode(parentnode, newnode);
    ISpyParserCallback cb = new ISpyParserCallback(treenode);
    ParserDelegator pd = new ParserDelegator();    pd.parse(isr,cb,true);
    isr.close();

    catch(MalformedURLException ex)
    {
```

```
    messageArea.append("    Bad URL encountered : "+urlstr+"\n\n");
    }
    catch(IOException e)
    {
     messageArea.append("    IOException, could not access site :
"+e.getMessage()+"\n\n");
    }
    yield();
    return;

  }


  public void stopSearch()
  {
    stopSearch = true;
  }


    public class ISpyParserCallback extends HTMLEditorKit.ParserCallback {
          private UrlTreeNode node;
          private DefaultMutableTreeNode treenode;

        private String lastText = "";

        public ISpyParserCallback(DefaultMutableTreeNode atreenode) {
            treenode = atreenode;
            node = (UrlTreeNode)treenode.getUserObject();
        }

        public void handleSimpleTag(HTML.Tag t,
                        MutableAttributeSet a,
                        int pos)

        {
         if(t.equals(HTML.Tag.IMG))
         {
          node.addImages(1);
          return;
         }
```

```
   if(t.equals(HTML.Tag.BASE))
   {
    Object value = a.getAttribute(HTML.Attribute.HREF);
    if(value != null)
     node.setBase(fixHref(value.toString()));
   }

  }

  public void handleStartTag(HTML.Tag t,
                    MutableAttributeSet a,
                    int pos)
  {
    if(t.equals(HTML.Tag.TITLE))
    {
     lastText="";
     return;
    }
    if(t.equals(HTML.Tag.A))
    {
     Object value = a.getAttribute(HTML.Attribute.HREF);
     if(value != null)
     {
     node.addLinks(1);
     String href = value.toString();
     href = fixHref(href);
     try{
      URL referencedURL = new URL(node.getBase(),href);
      searchWeb(treenode,
referencedURL.getProtocol()+"://"+referencedURL.getHost()+referencedURL.ge
tPath());
     }
     catch (MalformedURLException e)
     {
      messageArea.append("    Bad URL encountered : "+href+"\n\n");
      return;
     }
     }
    }
```

```
}

 public void handleEndTag(HTML.Tag t,
                          int pos)
{
 if(t.equals(HTML.Tag.TITLE) && lastText != null)
 {
  node.setTitle(lastText.trim());
  DefaultTreeModel tm = (DefaultTreeModel)searchTree.getModel();
  tm.nodeChanged(treenode);
  }

 }

 public void handleText(char[] data, int pos)
 {
  lastText = new String(data);
  node.addChars(lastText.length());
  String text = lastText.toUpperCase();
  for(int i = 0; i < keywordList.length; i++)
  {
   if(text.indexOf(keywordList[i]) >= 0)
   {
    if(!node.isMatch())
    {
     sitesFound++;
     updateStats();
    }
    node.setMatch(keywordList[i]);
    return;
   }
  }
 }
 }

}
import javax.swing.*;
import java.awt.*;
public class IntegerVerifier extends javax.swing.InputVerifier {
```

```
  private VerifierListener listener = null;
  private boolean blankOk = false;
  int minValue = Integer.MIN_VALUE;
  int maxValue = Integer.MAX_VALUE;
  public IntegerVerifier(VerifierListener alistener, boolean blankok, int min, int
max) {
      listener = alistener;
      blankOk = blankok;
      minValue = min;
      maxValue = max;
   }

      public boolean verify(javax.swing.JComponent jComponent) {
    JTextField thefield = (JTextField)jComponent;
    String input = thefield.getText();
    int number;
    input = input.trim();     if(input.length() == 0 && blankOk)
    {
     thefield.setForeground(Color.black);
     if(listener != null)
       listener.validData(jComponent);
     return true;     }
    else
    if(input.length() == 0 && !blankOk)
    {
     reportError(thefield,"Field cannot be blank!");
     return false;
}

    try{
      number = Integer.parseInt(input);
    }
    catch (NumberFormatException e)
    {
     reportError(thefield,"You must enter a valid number");
     return false;
    }

    if(number < minValue || number > maxValue)
    {
```

```
    reportError(thefield,"You must enter a number between "+minValue+" and
"+maxValue);
    return false;
    }

    thefield.setForeground(Color.black);
    thefield.setText(""+number);
    if(listener != null)
      listener.validData(jComponent);
    return true;
    }

    private void reportError(JTextField thefield, String message)
    {
     thefield.setForeground(Color.red);      if(listener != null)
      listener.invalidData(message,thefield);
    }

}
```

**STORAGE**

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;
import java.util.*;

public class AddDetails extends JInternalFrame implements ActionListener,
FocusListener {

        private JPanel pSearch = new JPanel ();
        private JLabel lbSearchId, lbSearchSite, lbPageStatistics, lbMessages,
 lbKeywords;
        private JTextField txtSearchId, txtSearchSite, txtPageStatistics,
 txtMessages;
        private JComboBox cboKeywords;
        private String Keywords[] = {<any>", ".com", ".edu", ".gov", ".int", ".mil",
 ".net", ".org", ".us", ".ca"};       //ComboBox Items.
```

```
private JButton btnOk, btnCancel;

private Statement st;
private long id = 0;

public AddDetails (Connection con) {

                super ("Add New Serch Detail", false, true, false, true);
        setSize (325, 250);



        lbSearchId = new JLabel ("Search ID:");
        lbSearchId.setForeground (Color.black);
        lbSearchId.setBounds (15, 15, 100, 20);
        lbSearchSite = new JLabel ("Searchsite Name:");
        lbSearchSite.setForeground (Color.black);
        lbSearchSite.setBounds (15, 45, 100, 20);
        lbPageStatistics = new JLabel ("Page Statistics:");
        lbPageStatistics.setForeground (Color.black);
        lbPageStatistics.setBounds (15, 75, 100, 20);
        lbMessages = new JLabel ("Messages:");
        lbMessages.setForeground (Color.black);
        lbMessages.setBounds (15, 105, 100, 20);
        lbKeywords = new JLabel ("Keywords:");
        lbKeywords.setForeground (Color.black);
        lbKeywords.setBounds (15, 135, 100, 20);

                txtSearchId = new JTextField ();
        txtSearchId.setHorizontalAlignment (JTextField.RIGHT);
        txtSearchId.addFocusListener (this);
        txtSearchId.setBounds (120, 15, 175, 25);
        txtSearchSite = new JTextField ();
        txtSearchSite.setBounds (120, 45, 175, 25);
        txtSearchSite = new JTextField ();
        txtSearchSite.setBounds (120, 75, 175, 25);
        txtPageStatistics = new JTextField ();
        txtPageStatistics.setHorizontalAlignment (JTextField.RIGHT);
        txtPageStatistics.setBounds (120, 105, 175, 25);
```

```
                    cboKeywords = new JComboBox (category);
            cboKeywords.setBounds (120, 135, 175, 25);

                    btnOk = new JButton ("OK");
            btnOk.setBounds (50, 175, 100, 25);
            btnOk.addActionListener (this);
            btnCancel = new JButton ("Cancel");
            btnCancel.setBounds (170, 175, 100, 25);
            btnCancel.addActionListener (this);

            //Registering the KeyListener to Restrict user to type only Numeric
in Numeric Boxes.

            txtSearchId.addKeyListener (new KeyAdapter () {
                public void keyTyped (KeyEvent ke) {
                    char c = ke.getKeyChar ();
                    if (! ((Character.isDigit (c)) || (c ==
KeyEvent.VK_BACK_SPACE))) {
                        getToolkit().beep ();
                        ke.consume ();
                    }
                }
            }
            );
            pSearch.setLayout (null);
            pSearch.add (lbSearchId);
            pSearch.add (lbSearchSite);
            pSearch.add (lbPageStatistics);
            pSearch.add (lbMessages);
            pSearch.add (lbKeywords);
            pSearch.add (txtSearchId);
            pSearch.add (txtSearchSite);
            pSearch.add (txtPageStatistics);
            pSearch.add (txtMessages);
            pSearch.add (cboKeywords);
            pSearch.add (btnOk);
            pSearch.add (btnCancel);


            getContentPane().add (pProduct, BorderLayout.CENTER);
```

```
            try {
                    st = con.createStatement ();                    }
            catch (SQLException sqlex) {
      JOptionPane.showMessageDialog (null, "A Problem Occurs While
Loading Form.");
                    dispose ();                                     }

            setVisible (true);

      }

            public void actionPerformed (ActionEvent ae) {

            Object obj = ae.getSource();

            if (obj == btnOk) {                                                 if
(txtSearchId.getText().equals ("")) {
                        JOptionPane.showMessageDialog (this, "Search Id not
Provided.");
                        txtProductId.requestFocus ();
                    }
                    else if (txtSearchSite.getText().equals ("")) {
                        JOptionPane.showMessageDialog (this, "Search Site
Name not Provided.");
                        txtSearchSite.requestFocus ();
                    }
                    else if (txtPageStatistics.getText().equals ("")) {
                        JOptionPane.showMessageDialog (this, "Page Statistics
not Provided.");
                        txtPageStatistics.requestFocus ();
                    }
                    else if (txtMessages.getText().equals ("")) {
                        JOptionPane.showMessageDialog (this, "Text
messages not Provided.");
                        txtMessages.requestFocus ();
                    }
                    else { //If All Information Provided then.

                        int id = Integer.parseInt (txtSearchId.getText ());
```

```
                    try {                                   String q =
"INSERT INTO SearchDetails (SearchId, SearchSite, PageStatistics, Messages,
Keywords) " +
                                "VALUES (" + id + ", '" +
txtSearchSite.getText() + "', '" + txtPageStatistics.getText() + ", '" +
txtMessages.getText() + "', '" +
                                 + cboKeywords.getSelectedItem() + "')";
                                int result = st.executeUpdate (q);
                            if (result == 1) {
                                    JOptionPane.showMessageDialog (this,
"Record has been Saved.");
                                    txtClear ();
                        }
                                else {
                            JOptionPane.showMessageDialog (this, "Problem
while Saving the Record.");
                                }
                            }
                            catch (SQLException sqlex) { }
                        }

                }

                if (obj == btnCancel) {
                        setVisible (false);
                        dispose();

                }

        }

                public void focusGained (FocusEvent fe) { }

        public void focusLost (FocusEvent fe) {

                if (txtSearchId.getText().equals ("")) {                }
                else {
                        id = Integer.parseInt (txtSearchId.getText ());
                long SearchNo;
                        boolean found = false;
```

```
                try {                                   String q = "SELECT * FROM
SearchDetails WHERE SearchId = " + id + "";
                        ResultSet rs = st.executeQuery (q);
                        rs.next ();
      SearchNo = rs.getLong ("SearchId");
                        if (SearchNo == id) {
                found = true;
                                txtClear ();
                                JOptionPane.showMessageDialog (this, id + " is
already assigned.");
                        }
                        else {
                                found = false;
                        }
                }
                catch (SQLException sqlex) { }
        }

    }

        private void txtClear () {

        txtSearchId.setText ("");
        txtSearchSite.setText ("");
        txtPageStatistics.setText ("");
        txtMessage.setText ("");
        txtSearchId.requestFocus ();

    }

}
```

# REFERENCES

# 10 REFERENCES

## BOOKS


Deie,H.M. & Deitel,P.J , *Java, How To Program*, Prentice Hall Inc 1997

Flanagan D, *Java in a Nutshell*, O'Rielly & Associates INC, 1996

Koosis D.J, Koosis D, *Java Programming for Dummies*, IDG Books Worldwide, Inc., 1996

Lalani S. and Jamsa K, *Java Programmer's Library*, Las Vegas, 1996

Lemay L. & Perkins C. L, *Teach Yourself Java 1.1 in 21 Days*, 2nd edition Indianapolis, Sams.net Publishing, 1996

Lemay L. & Perkins C. L, *Teach Yourself Java in 21 Days*, Indianapolis, 1996

Morrison M., et al., *Java 1.1 Unleashed*, Sams.net Publishing, 1997

Overland B, *Java In Plain English*, New York, 1996

Sommerville I, *Software Engineering*, fifth edition Addison-wesley Publishing, 1995

Vanderburg G., et al., *Maximum Java 1.1*, Sams.net Publishing 1997


## WEB SITES


http://g.oswego.edu/dl/html/javaCodingStd.html

Gamelan
http://www.gamelan.com/frame/Gamelan.util.html

Thomas Riechmann's Java Index
http://www4.informatik.uni-erlangen.de/~riechman/java.html

Daeron's Java
http://www.geom.umn.edu/~daeron/apps/

LensApp
http://sunflight.osc.cuny.edu/~steven/demo/pub/Lenze/Lenze.html

The NetRexx Language
http://www2.hursley.ibm.com/netrexx/

Kona Technologies Site
http://kona.lotus.com/demos/kona3-1/demo/index.html

Directory of /pub/java/docs
ftp://ncc.hursley.ibm.com/pub/java/docs/

IBM Network Computing: IBM & Java Integrated Solutions
http://www.ibm.com/java/platform_solutions.html

JavaSoft Home Page
http://java.sun.com/