TECHNICAL SPECIFICATION

Multiple Browser Configuration Manipulation System

By

Apoorva K R

(1225907)

Under the guidance of

Prof. Shoney Sebastian

September -2014

1. INTRODUCTION

1.1   OBJECTIVE

To overcome the problems faced in manipulating settings for each browser separately, a common interface is proposed to facilitate the process of manipulation of browser configurations. The system is used for detecting browsers installed in the user's computer and to manage the detected browsers efficiently.

Several settings information can be changed, customized, analyzed and managed according to the requirements. This system first detects the number of browsers existing in the system and it can access all the history details of these browsers and make changes to the settings.

Several system utilities are introduced in a user friendly manner such that even a layman can work on the system without much technical skills.

A **web browser** (commonly referred to as a **browser**) is a software application for retrieving, presenting and traversing information resources on the World Wide Web. An information resource is identified by a Uniform Resource Identifier (URI/URL) and may be a web page, image, video or other piece of content. Hyperlinks present in resources enable users easily to navigate their browsers to related resources.

Although browsers are primarily intended to use the World Wide Web, they can also be used to access information provided by web servers in private networks or files in file systems. The major web browsers are Firefox, Internet Explorer, Google Chrome, Opera, and Safari.

Managing multiple browsers in a system can be a time consuming task . Several configurations and settings to be managed together is a tedious task. A software which is designed to manage several such settings and configuration details at one go on a system is the probable solution.

Along with the settings other functionalities like detecting the number of browsers, IP masking, web masking, determining the time of response can be viewed and managed effectively.

1.2 PROBLEM STATEMENT

Our mission is to enable organizations to cost-effectively manage and secure web browsers and web applications at a global scale. We deliver on this mission by developing and delivering browser management solutions for the world's largest enterprises. Our browser management platform provides IT with granular control of browser environments, to optimize web application compatibility and security. This

enables a quick and cost-effective response to changes in web technology, without compromising legacy or modern business applications. The proposed system is designed to overcome all the disadvantages that are faced in the existing system. The security issue demands major concern in order to structure a reliable and easy solution for the current existing system.

Managing multiple browsers in a system can be a time consuming task . Several configurations and settings to be managed together is a tedious task. A software which is designed to manage several such settings and configuration details at one go on a system is the probable solution.

Along with the settings other functionalities like detecting the number of browsers, IP masking, web masking, determining the time of response can be viewed and managed effectively. To overcome the problems faced in manipulating settings for each browser separately, a common interface is proposed to facilitate the process of manipulation of browser configurations. The system is used for detecting browsers installed in the user's computer and to manage the detected browsers efficiently.

The product reduces costs and improves compatibility and security, freeing IT from the burdens of browser management so they can focus their energy on new, business-critical technology initiatives. In today's computing environments, browser choice has broadened for the enterprise and end-users. IT organizations are challenged with managing multi-browser environments or face security risks and an uptick in helpdesk calls. Browseal Catalyst is a multi-browser web traffic manager for the enterprise, enabling IT to pair all web applications with the most compatible and secure browser. Catalyst works with Internet Explorer, Chrome, and Firefox on all PCs in the enterprise.

In all usage scenarios and offers seamless browser management which ultimately gives the end users browser flexibility and a more secure computing environment. IT organizations maximize browser compatibility and strengthen security with this easy to manage utility.

Browseal is aimed to reduce technical and user education-related headaches for these use cases. This new tool gives IT managers a console through which they can control what browsers open which websites for every computer on their network.

The best part of software is probably how thorough it is. If the user set the google.com site to open only in Google Chrome, it will open in Chrome no matter what. The software will seamlessly open up the appropriate browser without interrupting the user or throwing any pop-up windows in their face. If , further the user wishes to block certain websites or IP masking feature is also available. The proposed system has a separate login facility for the admin in order to manage and control the settings effectively. Apart from the features mentioned above, some utility functions are also available which will make it more user friendly. An attractive design increases the easy understanding of the user. There is a very important feature in the proposed system where the history of the browsers can be deleted easily which saves time. The proposed system also has feature of determining the response time of the site that the user wishes to find out.

This software can enable IT-enforced browser restrictions that define so many corporate computing environments. With this software, IT can mandate that Internet Explorer be used for internal sites, but allow users to use their browser of choice for the rest of the Web, something that should help to alleviate headaches on both sides.

Additional features of this software includes IP masking and web masking.

A **mask** is used to divide an **IP** address into two parts. One part identifies the host (computer), the other part identifies the network to which it belongs. Another feature is web masking which helps the user to mask a web site from viewing. The IT firms can also know the web response time of a particular site. A user friendly navigator is embedded in the software which will give the provision to the user to use the software as a browser itself where one can navigate and clear history as well. If a user us using multiple browsers, this software helps to clear the history of all the browsers at one go. Security area is also taken care off where a special login option for the admin is provided.

2. REQUIREMENT ANALYSIS

2.1 SCOPE OF MULTIPLE BROWSERS

The aim of this project is to manage multiple browsers in a system in an effective and quick manner. To allow the IT organizations to cost- effectively manage browsers at global scale. Managing multiple browsers in a system can be a time consuming task. Several configurations and settings to be managed together is a tedious task. A software which is designed to manage several such settings and configuration details at one go on a system is the probable solution.

Along with the settings other functionalities like detecting the number of browsers, IP masking, web masking, determining the time of response can be viewed and managed effectively. Browseal is aimed to reduce technical and user education-related headaches for these use cases.

The best part of software is probably how thorough it is. If the user set the google.com site to open only in Google Chrome, it will open in Chrome no matter what. The software will seamlessly open up the appropriate browser without interrupting the user or throwing any pop-up windows in their face. If , further the user wishes to block certain websites or IP masking feature is also available. The proposed system has a separate login facility for the admin in order to manage and control the settings effectively.

The key objectives are:

- Manage several browser related settings.
    - Detect the number of browsers in the system.
    - Delete history details of all browsers.
    - Check web response of any site.
    - Enhance security through IP masking and web blocking.
- Implement VB.Net interface as front end for attractive design.
- Admin login available.
- Easy navigation and user friendly design facilities.
- Cost –effective and time saving methodology.
- Extra utilities related to the browser is available to manage its configuration.

2.2 EXISTING SYSTEM

Internet Explorer

Internet Explorer is one of the most widely used web browsers, attaining a peak of about 95% usage share during 2002 and 2003. Its usage share has since declined with the launch of Firefox (2004) and Google Chrome (2008), and with the growing popularity of operating systems such as OS X, Linux and Android that do not run Internet Explorer. Estimates for Internet Explorer's overall market share range from 27.4% to 54.13%, as of October 2012 (browser market share is notoriously difficult to calculate). Microsoft spent over US$100 million per year on Internet Explorer in the late 1990s, with over 1000 people working on it by 1999.

General Settings of Internet Explorer are:

- Change Internet Explorer General settings

  Change your home page, delete the browsing history, change search settings, change tab settings, and customize the appearance of Internet Explorer.

- Change Internet Explorer Security settings

  Set default and custom security levels for the Internet, intranet, and specific websites.

- Change Internet Explorer Privacy settings

  Change cookie and Pop-up Blocker settings.

- Change Internet Explorer Content settings

  Turn on Content Advisor or change its settings, view and manage security certificates, change AutoComplete settings, or change feed (RSS) settings.

- Internet Explorer Connections settings: recommended links

  Set up an Internet connection, add or change dial-up and virtual private network (VPN) settings, and change local area network (LAN) settings.

- Change Internet Explorer Programs settings

  Change your default web browser, e-mail program, HTML editor, Newsgroup reader, or Internet telephone, and manage web browser add-ons.

- Change Internet Explorer Advanced settings

  Change advanced settings for accessibility, browsing, HTTP protocol handling, International domain names, Java Virtual Machine use, Multimedia, Printing, Search, and security. Reset all Internet Explorer settings to the default settings.

Google Chrome

Google Chrome is a freeware web browser developed by Google. It used the WebKit layout engine until version 27 and, with the exception of its iOS releases, from version 28 and beyond uses the WebKit fork Blink. It was first released as a beta version for Microsoft Windows on September 2, 2008, and as a stable public release on December 11, 2008.

As of July 2014, Stat Counter estimates that Google Chrome has a 45% worldwide usage share of web browsers so this estimate indicates it is the most widely used web browser in the world.

In September 2008, Google released the majority of Chrome's source code as an open-source project called Chromium, on which Chrome releases are still based. A notable component that is not open source is the built-in Flash player.

Google Chrome aims to be secure, fast, simple and stable. There are extensive differences from its peers in Chrome's minimalistic user interface, which is atypical of modern web browsers.
For example, Chrome does not render RSS feeds.
 One of Chrome's strengths is its application performance and JavaScript processing speed, both of which were independently verified by multiple websites to be the swiftest among the major browsers of its time. Many of Chrome's unique features had been previously announced by other browser developers, but Google was the first to implement and publicly release them. For example, a prominent graphical user interface (GUI) innovation, the merging of the address bar and search bar (the Omnibox), was first announced by Mozilla in May 2008 as a planned feature for Firefox. Both Internet Explorer 9 and Safari (version 6) have since merged the search and address bar.
General Settings for Google Chrome are:

- **Reset browser settings**

  Google Chrome gives you the option to reset your browser settings in one easy click. In some cases, programs that you install can change your Chrome settings without your knowledge. You may see additional extensions and toolbars or a different search engine. Resetting your browser settings will

reset the unwanted changes caused by installing other programs. However, your saved bookmarks and passwords will not be cleared or changed.

- **Make webpages load faster (prerendering)**

  You can make webpages load faster by telling Google Chrome to prerender (preload) links. Google Chrome does this by predicting what links you might click, preparing them to load instantly for you.

- **Cookies and site data**

  Cookies are files created by websites you've visited to store browsing information, such as your site preferences or profile information. They're allowed by default. It's important to be aware of your cookie settings because cookies can allow sites to track your navigation during visits to sites.

- **SSL warning messages**

  You might see a message that says "Your connection is not private" when Chrome detects that your Internet connection or your computer is stopping Chrome from loading the page securely. If you see this message, you can click Advanced to learn more about the issue.

- **Troubleshooting Steps**

  You can use the troubleshooting steps below for the SSL warning message that you're experiencing. General:

SAFARI

Safari is a web browser developed by Apple Inc. and included with the OS X and iOS operating systems. First released as a public beta on January 7, 2003, on the company's OS X operating system, it became Apple's default browser beginning with Mac OS X v10.3 "Panther". Safari is also the native browser for iOS.

A version of Safari for the Microsoft Windows operating system was first released on June 11, 2007,and supported Windows XP Service Pack 2, or later, but it has been discontinued. Safari 5.1.7, released on May 9, 2012, is the last version available for Windows.

According to Net Applications, Safari accounted for 46.07% of mobile web browsing traffic and 5.28% of desktop traffic in June 2014, giving a combined market share of 12.32%.

On Mac OS X, Safari is a Cocoa application. It uses Apple's WebKit for rendering web pages and running JavaScript. WebKit consists of WebCore and JavaScriptCore .Like KHTML and KJS, WebCore and JavaScriptCore are free software and are released under the terms of the GNU Lesser General Public License. Some Apple improvements to the KHTML code are merged back into the Konqueror project. Apple also releases additional code under an open source 2-clause BSD-like license.

Until Safari 6.0, it included a built-in web feed aggregator that supported the RSS and Atom standards. Current features include Private Browsing (a mode in which no record of information about the user's web activity is retained by the browser), a "Ask websites not to track me" privacy setting, the ability to archive web content in Web Archive format, the ability to e-mail complete web pages directly from a browser menu, the ability to search bookmarks, and the ability to share tabs between all Macs and iOS devices running appropriate versions of software via an iCloud account.

- **Block pop-up windows**

  Older versions of Safari had an option to block pop-up windows in the program menu. Apple has removed this option from the menu, but the feature is still available. Now you need to enable it by checking the box in the Security pane of Safari's preferences. With it enabled, Web sites will be prevented from launching new browser windows, which on some spam sites can result in many windows opening at once and cluttering your display.

- **Set homepage options**

  The first options are those you will see whenever you create new Safari window or tab, and these are located in the General section of the Safari preferences. In here, you can choose to open new windows and tabs with an empty page, Safari's Top Sites view, the same page that was loaded on the last active tab or window, or a custom home page you specify in the Homepage field. In addition, you can choose to load an organized folder of tabs when you open a new window, which can be set up in Safari's bookmarks organization interface. If you use the Top Sites feature, you can also set the number of previews to show in the same preferences pane. By default it shows 12, but you can select either six or 24, depending on your needs.

- **Clear site-specific data**

  Web sites you visit often store site-specific settings in cookies and caches. If this data is corrupted, then sites may not load properly, or show odd behavior like not accepting log-in credentials. Often when Web sites have such problems you can try clearing cookies and other site-specific data, but using the Reset Safari option will clear data from all of your Web sites, which may not be preferable. Instead, go to the Privacy section of Safari's preferences and click "Details," under the option to remove all Web site data. In the panel that appears, you can search for a site you've visited to remove data for that site only.

- **Manage site-specific plug-in options**

  Similar to the management of site-specific data, the plug-ins that Web sites use to display content can be managed on a per-site basis. While Safari has an option in its Security preferences to allow or block plug-ins, next to this option is a "Manage Website Settings" button that will allow you tospecify not only how Safari handles each plug-in globally, but also how Safari will do so for specific Web sites.

  To do this, after clicking the button, you can select a plug-in and choose the restriction level to use for other Web sites. This will bring up the global settings for the plug-in, as well as individual options for sites listed.

OPERA

Opera is a web browser developed by Opera Software. The latest version currently runs on Microsoft Windows and OS X operating systems and uses the Blink layout engine. An earlier version using the Presto layout engine is still supported, and additionally runs on Linux and FreeBSD systems. As of August 2014, a Blink-based Linux version is available on the developer stream. Editions of Opera are available for devices running the Android, iOS, Symbian, Maemo, Bada, BlackBerry and Windows Mobile operating systems, and for Java ME capable devices.

According to Opera Software, the browser had more than 350 million users worldwide (more than 270 million users with mobile versions) in December 2013.

Opera has been noted for originating many features later adopted by other web browsers. A prominent example is Speed Dial.

Opera includes built-in tabbed browsing, a bookmarks bar, add-ons, and a download manager. Opera has "Speed Dial", which allows the user to add an unlimited number of pages shown in thumbnail form in a page displayed when a new tab is opened. Speed Dial allows the user to more easily navigate to the selected web pages.

- **Content blocking**

  Some webpages contain additional content that you may find distracting, such as advertisements. You can control what is displayed by selectively blocking undesirable images or plug-in content. To do this: Right-click the webpage, and select "Block Content". In content blocking mode, all content that you are able to block is highlighted. Content on a website is either contained in a folder for the page or displayed as a specific file. You can block either the whole folder, so that anything that is loaded into that folder is automatically blocked, or the specific file. To block the folder content, select the content that you want to block, then click "Done". To block a specific file, hold Shift while you select the content to block. To check what you have blocked, select "Details" while in content-blocking mode. A list of folders or specific files that you have blocked displays.

- **Site-specific preferences**

  Not all WebPages are the same. Sometimes, you may want to apply a preference setting to most sites, but apply a different preference to a specific site. For example, you may want to block unrequested pop-ups by default, but enable them for a specific site that relies on using them. To solve this dilemma, you can set permanent preferences for individual sites, which override global settings. To do this, right-click the webpage and select "Edit Site Preferences".

  To manage or remove site preferences, select Settings > Preferences > Advanced > Content > Manage site preferences.

AVANT

Avant Browser is a freeware web browser from a Chinese programmer named Anderson Che, which unites the Trident layout engine built into Windows (see Internet Explorer shell) with an interface intended to be more feature-rich, flexible and ergonomic than Microsoft's Internet Explorer (IE). It runs on Windows 2000 and above, including Windows 8 and Windows 8.1. Internet Explorer versions 6 through 9 are supported.

Version 2012 (released October 2011) was separated in two editions: Ultimate edition, which added the Gecko layout engine (used by Mozilla Firefox), allowing the user choose between both layout engines, and Lite edition which contains only the Trident layout engine. As of November 2008, total downloads surpassed 22.5 million. Avant Browser is currently available in 41 languages.

Features:

- Interface
- Tabbing
- User-customizable tabbed interface
- Progress bar on each tab (optional)
- Tab locking, duplication, and drop-and-drag arrangement
- Automatically save and reopen any number of pages
- Can restore open pages after system crash
- Undo menu for last 25 closed tabs
- All pages open in single process, reducing resource demands
- Multiple home pages
- Tools
- Compatible with many Internet Explorer toolbars and add-ins, including Google toolbar.
- Highlight marker to find keywords on any page (the words also used by the search engine)
- Facility for user to add custom-made function buttons
- RSS feed reader built in
- URL alias facility to create keyboard shortcuts to favorite sites

- Page zoom slider - 25% to 500%

- Full desktop mode - hides toolbars

- Full screen mode - hides toolbars and taskbar

- Compact mode - saves interface space

- Boss button - quickly hides browser

- Tray icon - keeps Avant running in system tray for fast access (optional)

- Mouse gestures (customizable)

- Autoscroll facility for hands-free reading

- Context-activated floating toolbars to save and edit images, text and Flash

- Customizable right-click menu entries

- Quick translation menu

- 15 interface languages available so far

- Automatic page refresh at custom intervals

Sidebar access to

- Advanced search

- Blocked URL record

- History

- Bookmarks

- Feeds

Appearance

- Fully skinnable:

- Range of bundled skin options (21 total)

- Accessory application allows user to create own skins

- Highly customizable:

- Toolbars can be rearranged freely

- Most functions may be accessed from either toolbar buttons or menus

- All Buttons and menus are optional

Security and privacy

- Push-button quick blocking of unwanted items (Popups, Adverts, ActiveX, Java, Scripts, Flash, Video, Sound, Images)
- Customizable black/white lists for popups and ad servers
- Multiple proxy servers may be configured
- Browsing tracks cleaner - can clear traces automatically
- Windows XP Service Pack 2 security extensions can be integrated

Search

- Search bar with separate engines configurable for web, images, groups, directories and news
- Search within site, search within page, and target word highlight buttons
- Quick Search facility - user can configure any engine for searches from the address bar
- Similar websites toolbar - searches in background for pages that relate to the current

Bookmarks and online storage

- Online secure bookmark, feed and setting storage - allows access of same browser setup anywhere
- Portable flash drive version allows running of Avant without installation
- Bookmark search
- Bookmark validation
- Bookmark toolbar for fast link access
- Bookmarks can be arranged in task-orientated groups that can be opened together or individually
- Optionally uses Internet Explorer favorites
-

2.3 PROPOSED SYSTEM

The system is used for detecting browsers installed in the user's computer and to manage the detected browsers efficiently. Several settings information can be changed, customized, analyzed and managed according to the requirements. This system first detects the number of browsers existing in the system and it can access all the history details of these browsers and make changes to the settings.

Several system utilities are introduced in a user friendly manner such that even a layman can work on the system without much technical skills.

The proposed system is designed to overcome all the disadvantages that are faced in the existing system. The security issue demands major concern in order to structure a reliable and easy solution for the current existing system.

2.4 LITERATURE SURVEY

The ways that web browser makers fund their development costs has changed over time. The first web browser, Worldwide Web, was a research project. Netscape Navigator was originally sold commercially, as was Opera; Netscape no longer exists and has been replaced with the free Firefox, while Opera is now downloadable free of charge.

Internet Explorer, on the other hand, was from its first release always included with the Windows operating system (and furthermore was downloadable for no extra charge), and therefore it was funded partly by the sales of Windows to computer manufacturers and direct to users. Internet Explorer also used to be available for the Mac. It is likely that releasing IE for the Mac was part of Microsoft's overall strategy to fight threats to its quasi-monopoly platform dominance - threats such as web standards and Java - by making some web developers, or at least their managers, assume that there was "no need" to develop for anything other than Internet Explorer (an assumption that later proved to be badly mistaken, with the rise of Firefox and Chrome). In this respect, IE may have contributed to Windows and Microsoft applications sales in another way, through tricking organisations into inadvertent "lock-in" into the Microsoft platform.

The primary purpose of a web browser is to bring information resources to the user ("retrieval" or "fetching"), allowing them to view the information ("display", "rendering"), and then access other information ("navigation", "following links").

This process begins when the user inputs a <u>Uniform Resource Locator</u> (URL), for example http://en.wikipedia.org/, into the browser. The prefix of the URL, the Uniform Resource Identifier or <u>URI</u>, determines how the URL will be interpreted. The most commonly used kind of URI starts with *http:* and identifies a resource to be retrieved over the <u>Hypertext Transfer Protocol</u> (HTTP).

2.5 FUNCTIONAL REQUIREMENTS

Multiple browser management system is aimed to help the IT industry to manage browsers on a global scale. The software contains different modules which contribute the essence of the different functionalities involved in the browser .

Modules

- History deletion

  One of the most important functionality is the deletion of the browser history. The links that were opened by the user in the recent past is stored in a folder. That particular folder is recognized and all the history details can be deleted in a easy and time saving manner. If the user's system has more than two browsers also this particular feature is applicable.

- Masking

  IP masking and web blocking are the key modules of the software. A particular IP address can be masked or blocked from access . The particular IP address which the user wishes to block can be entered and the user will not be able to access it. Similarly web blocking is also possible. A website which the user wants to block in order to avoid illegal access for further security issues to be avoided can also be done in an efficient manner and saves a lot time on a global scale in any IT firm.

- Detection of browsers

  When the software is installed in the system, the first module to work is the detection of different browsers in the system. This module further gives various options like deleting browsers or uninstalling the browser from the system in an easy manner, deleting the history of the browsers detected can also be done.

- Utilities

  In addition to the different functionalities present in the software, there are certain utilities present .One of the utility includes recording the problem which will be converted to a web page form and can be mailed to the admin who can address the problem related to that particular browser and solve it . This makes the software cost effective and time saving tool.

2.6 NON FUNCTIONAL REQUIREMENTS

- Reliability

  This is one of the most important requirements of any project. This means that the application that is created can be relied upon. It must be a sound system and should always give consistent results. This is just in regard with the user , the networking issues are beyond the scope.

- Maintainability

  Maintainability would not pose much of a problem as the necessary documentation is carried out. It will also be easy to upgrade and modify the project to make it even better.

- Usability

  Usability refers to whether the system is actually serving the purpose for which it is intended to. If not then the wrong product has been built. It also portrays the ease of use of the application for any type of end users.

- Efficiency

  This refers to whether the project as a whole is efficient in terms of time and accuracy. In the proposed project the management of several browsers ensures for a more reliable as well as efficient system.

2.7 HARDWARE REQUIREMENTS

Processor   : Intel Pentium 4 and higher

Memory   : Minimum RAM required 512 MB

Free disk space : Minimum Hard disk required 40 GB

2.8 SOFTWARE REQUIREMENTS

Front end : VB.Net

3. DESIGN SPECIFICATIONS

3.1 BLOCK DIAGRAM
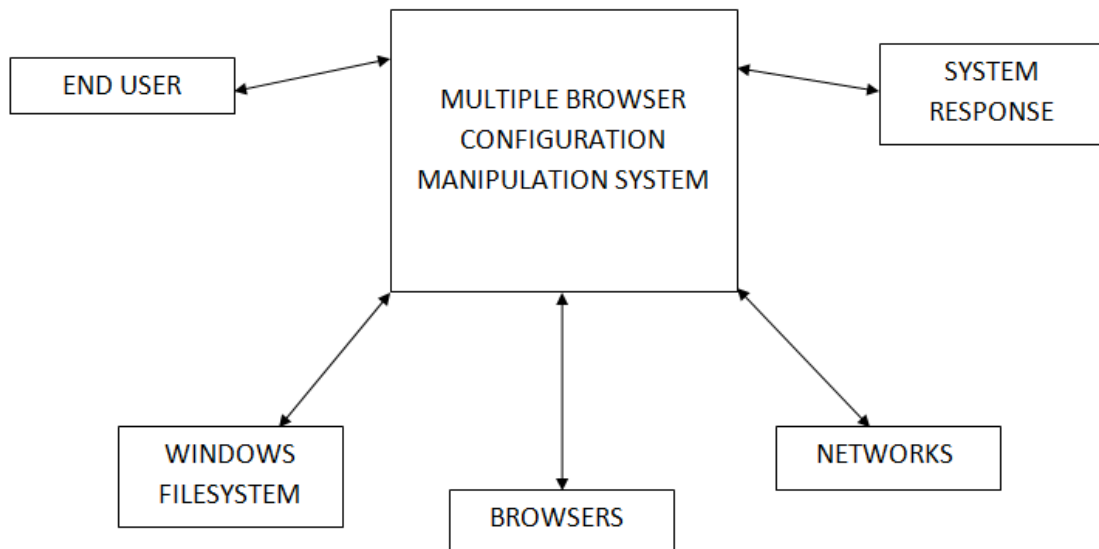
Fig: 3.1 Block diagram

The above Block diagram shows that,one of the most important functionality is the deletion of the browser history. The links that were opened by the user in the recent past is stored in a folder. That particular folder is recognized and all the history details can be deleted.

A particular IP address can be masked or blocked from access . The particular IP address which the user wishes to block can be entered and the user will not be able to access it. Similarly web blocking is also possible.

When the software is installed in the system, the first module to work is the detection of different browsers in the system. This module further gives various options like deleting browsers or uninstalling the browser from the system in an easy manner, deleting the history of the browsers detected can also be done.

3.2. SYSTEM SPECIFIC MODULES:

It has been modularized into following modules:

- History deletion

  One of the most important functionality is the deletion of the browser history. The links that were opened by the user in the recent past is stored in a folder. That particular folder is recognized and all the history details can be deleted in a easy and time saving manner. If the user's system has more than two browsers also this particular feature is applicable.

- Masking

  IP masking and web blocking are the key modules of the software. A particular IP address can be masked or blocked from access . The particular IP address which the user wishes to block can be entered and the user will not be able to access it. Similarly web blocking is also possible..

- Blocking

  A website which the user wants to block in order to avoid illegal access for further security issues to be avoided can also be done in an efficient manner and saves a lot time on a global scale in any IT firm.

- Detection of browsers

  When the software is installed in the system, the first module to work is the detection of different browsers in the system. This module further gives various options like deleting browsers or uninstalling the browser from the system in an easy manner, deleting the history of the browsers detected can also be done.

- Utilities

  In addition to the different functionalities present in the software, there are certain utilities present .One of the utility includes recording the problem which will be converted to a web page form and can be mailed to the admin who can address the problem related to that particular browser and solve it. This makes the software cost effective and time saving tool.

4. IMPLEMENTATION

4.1. SAMPLE CODE

```
Public Class Admin
Private Sub picClose_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles picClose.Click
Me.Close()
End Sub
Private Sub picClear_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles picClear.Click
txt_UID.Text = ""
txtPassword.Text = ""
End Sub
Private Sub picLogin_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles picLogin.Click 'Read from file and Match Uname and
Password.'Temp code
If (Trim(txt_UID.Text = "admin")) Then
If (Trim(txtPassword.Text = "password")) Then
MsgBox("Welcome Browseal Master", vbInformation)
Me.Close()
'setting page open here
Else
MsgBox("Password is incorrect. Please try again", vbCritical)
txtPassword.Text = ""
txtPassword.Focus()
End If
Else
MsgBox("User Name or Password is incorrect. Please try again", vbCritical)
txt_UID.Text = ""
txtPassword.Text = ""
txt_UID.Focus()
End If
End Sub
End Class
Public Class Browser
Private Sub but_navigate_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles but_navigate.Click
WebBrowser1.Navigate("http://" + txt_navigate.Text)
End Sub
Private Sub but_back_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles but_back.Click
WebBrowser1.GoBack()
End Sub
Private Sub but_forwd_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles but_forwd.Click
WebBrowser1.GoForward()
End Sub
```

```vb
Private Sub WebBrowser1_DocumentCompleted(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.WebBrowserDocumentCompletedEventArgs)
Handles WebBrowser1.DocumentCompleted
lab_url.Text = WebBrowser1.Document.Url.AbsoluteUri
' lab_urlTitle.Text = WebBrowser1.DocumentTitle
List_History.Items.Add(lab_url.Text)
End Sub
Private Sub Button1_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles but_openLink.Click
lab_url.Text = List_History.SelectedItems(0).SubItems(0).Text
txt_navigate.Text = lab_url.Text
WebBrowser1.Navigate(lab_url.Text)
End Sub
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
grp_webNavigate.Visible = True
'grp_result.Visible = False
End Sub
Private Sub Button1_Click_2(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
List_History.Clear()
End Sub
End Class
Public Class Detect
Dim intP1, intP2, intP3, intP4, intP5 As Integer
Private Sub Detect_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
txt_ListBroser.Text = ""
grp_result.Visible = True
'grp_webNavigate.Visible = False
'Enabe timer for progres
' Progress_Timer.Enabled = True
'Detect Browser from reg Level..
Dim readValue =
My.Computer.Registry.GetValue("HKEY_LOCAL_MACHINE\SOFTWARE\Client
s\StartMenuInternet\IEXPLORE.EXE","", Nothing)
'     MsgBox("The value is " & readValue)
Dim readValue1 =
My.Computer.Registry.GetValue("HKEY_LOCAL_MACHINE\SOFTWARE\Client
s\StartMenuInternet\Google Chrome",
"", Nothing)
'MsgBox("The value is " & readValue1)
Dim readValue2 =
My.Computer.Registry.GetValue("HKEY_LOCAL_MACHINE\SOFTWARE\Client
s\StartMenuInternet\Opera",
"", Nothing)
Dim readValue3 =
My.Computer.Registry.GetValue("HKEY_LOCAL_MACHINE\SOFTWARE\Client
s\StartMenuInternet\FIREFOX.EXE",
```

```
"", Nothing)
Dim readValue4 =
My.Computer.Registry.GetValue("HKEY_LOCAL_MACHINE\SOFTWARE\Client
s\StartMenuInternet\Avant.Browser",
"", Nothing)
If readValue <> "" Then
' MsgBox("I am here")
txt_ListBroser.Text = readValue.ToString
End If
If readValue1 <> "" Then
'MsgBox("I am here..1")
txt_ListBroser.Text = txt_ListBroser.Text + vbNewLine + readValue1.ToString
End If
If readValue2 <> "" Then
txt_ListBroser.Text = txt_ListBroser.Text + vbNewLine + readValue2.ToString
End If
If readValue3 <> "" Then
txt_ListBroser.Text = txt_ListBroser.Text + vbNewLine + readValue3.ToString
End If
If readValue4 <> "" Then
txt_ListBroser.Text = txt_ListBroser.Text + vbNewLine + readValue4.ToString
End If
Dim arr() As String
Dim i As Integer
arr = Split(txt_ListBroser.Text, vbNewLine)
For i = 0 To UBound(arr)
If CheckBox1.Text = "" Then
CheckBox1.Visible = True
CheckBox1.Text = arr(i)
setImage(arr(i), 1)
ElseIf CheckBox2.Text = "" Then
CheckBox2.Visible = True
CheckBox2.Text = arr(i)
setImage(arr(i), 2)
ElseIf CheckBox3.Text = "" Then
CheckBox3.Visible = True
CheckBox3.Text = arr(i)
setImage(arr(i), 3)
ElseIf CheckBox4.Text = "" Then
CheckBox4.Visible = True
CheckBox4.Text = arr(i)
setImage(arr(i), 4)
ElseIf CheckBox5.Text = "" Then
CheckBox5.Visible = True
CheckBox5.Text = arr(i)
setImage(arr(i), 5)
End If
Next
```

```
End Sub
Private Sub setImage(ByVal data As String, ByVal pos As Integer)
Try
If pos = 1 Then
picb1.Visible = True
If (data.Equals("Google Chrome")) Then
picb1.Image = My.Resources.chrome
intP1 = 1
ElseIf (data.Equals("Internet Explorer")) Then
picb1.Image = My.Resources.IE
intP1 = 2
ElseIf (data.Equals("Opera")) Then
picb1.Image = My.Resources.opera
intP1 = 3
ElseIf (data.Equals("Avant")) Then
picb1.Image = My.Resources.avant
intP1 = 4
ElseIf (data.Equals("Mozilla Firefox")) Then
picb1.Image = My.Resources.MF
intP1 = 5
End If
End If
If pos = 2 Then
picb2.Visible = True
If (data.Equals("Google Chrome")) Then
picb2.Image = My.Resources.chrome
intP2 = 1
ElseIf (data.Equals("Internet Explorer")) Then
picb2.Image = My.Resources.IE
intP2 = 2
ElseIf (data.Equals("Opera")) Then
picb2.Image = My.Resources.opera
intP2 = 3
ElseIf (data.Equals("Avant")) Then
picb2.Image = My.Resources.avant
intP2 = 4
ElseIf (data.Equals("Mozilla Firefox")) Then
picb2.Image = My.Resources.MF
intP2 = 5
End If
End If
If pos = 3 Then
picb3.Visible = True
If (data.Equals("Google Chrome")) Then
picb3.Image = My.Resources.chrome
intP3 = 1
ElseIf (data.Equals("Internet Explorer")) Then
picb3.Image = My.Resources.IE
```

```
intP3 = 2
ElseIf (data.Equals("Opera")) Then
picb3.Image = My.Resources.opera
intP3 = 3
ElseIf (data.Equals("Avant")) Then
picb3.Image = My.Resources.avant
intP3 = 4
ElseIf (data.Equals("Mozilla Firefox")) Then
picb3.Image = My.Resources.MF
intP3 = 5
End If
End If
If pos = 4 Then
picb4.Visible = True
If (data.Equals("Google Chrome")) Then
intP4 = 1
picb4.Image = My.Resources.chrome
ElseIf (data.Equals("Internet Explorer")) Then
intP4 = 2
picb4.Image = My.Resources.IE
ElseIf (data.Equals("Opera")) Then
intP4 = 3
picb4.Image = My.Resources.opera
ElseIf (data.Equals("Avant")) Then
intP4 = 4
picb4.Image = My.Resources.avant
ElseIf (data.Equals("Mozilla Firefox")) Then
intP4 = 5
picb4.Image = My.Resources.MF
End If
End If
If pos = 5 Then
picb5.Visible = True
If (data.Equals("Google Chrome")) Then
picb5.Image = My.Resources.chrome
intP5 = 2
ElseIf (data.Equals("Internet Explorer")) Then
picb5.Image = My.Resources.IE
intP5 = 2
ElseIf (data.Equals("Opera")) Then
intP5 = 3
picb5.Image = My.Resources.opera
ElseIf (data.Equals("Avant")) Then
intP5 = 4
picb5.Image = My.Resources.avant
ElseIf (data.Equals("Mozilla Firefox")) Then
intP5 = 5
picb5.Image = My.Resources.MF
```

```vbnet
End If
End If
Catch ex As Exception
End Try
End Sub
Private Sub runIE()
End Sub

Private Sub grp_result_Enter(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles grp_result.Enter
End Sub

Private Sub picb1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles picb1.Click
Try
If (intP1 = 1) Then
System.Diagnostics.Process.Start("C:\Program Files
(x86)\Google\Chrome\Application\chrome.exe")
End If
If (intP1 = 2) Then
System.Diagnostics.Process.Start("C:\Program Files (x86)\Internet
Explorer\iexplore.exe")
End If
Catch ex As Exception
End Try
End Sub
Private Sub picb2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles picb2.Click
Try
If (intP2 = 1) Then
System.Diagnostics.Process.Start("C:\Program Files
(x86)\Google\Chrome\Application\chrome.exe")
End If
If (intP2 = 2) Then
System.Diagnostics.Process.Start("C:\Program Files (x86)\Internet
Explorer\iexplore.exe")
End If
Catch ex As Exception
End Try
End Sub
Private Sub picClose_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles picClose.Click
End Sub

Private Sub picb5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles picb5.Click
End Sub
End Class
Public Class MasterForm
```

```vb
Dim Flag As Integer
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
MaskingEntry.Hide()
' txt_ListBroser.Visible = False
Progress_bar.Visible = False
'grp_result.Visible = False
'grp_webNavigate.Visible = False
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
Progress_bar.Visible = False
'txt_ListBroser.Visible = False
End Sub

Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
System.Diagnostics.Process.Start("C:\Windows\System32\psr.exe")
End Sub
Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
System.Diagnostics.Process.Start("C:\Windows\System32\mstsc.exe")
End Sub
Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
System.Diagnostics.Process.Start("C:\Windows\System32\resmon.exe")
End Sub

Private Sub PictureBox2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox2.Click
System.Diagnostics.Process.Start("C:\Windows\System32\psr.exe")
End Sub
Private Sub PictureBox6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox6.Click
System.Diagnostics.Process.Start("C:\Windows\System32\mstsc.exe")
End Sub

Private Sub PicResponse_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PicResponse.Click
ResponseManager.Show()

End Sub

Private Sub PicBrowse_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PicBrowse.Click
' grp_webNavigate.Visible = True
'grp_result.Visible = False
Flag = 2
```

```
Progress_Timer.Enabled = True
End Sub

Private Sub PicDetect_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PicDetect.Click
Flag = 1
Progress_Timer.Enabled = True
End Sub

Private Sub PicMask_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PicMask.Click
'Form with Text box to enter mask IP
Flag = 5
Progress_Timer.Enabled = True
'Add it to Host.config
End Sub

Private Sub PictureBox7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox7.Click
'Form with Text box to enter mask IP
Flag = 3
Progress_Timer.Enabled = True
'Add it to Host.config
End Sub

Private Sub PicHist_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PicHist.Click
'Delete history of Google Chrome
Flag = 4
Progress_Timer.Enabled = True
End Sub

Private Sub PictureBox8_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
Admin.Show()
Admin.Focus()
'ask for user name and password. !
End Sub

Private Sub PictureBox9_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox9.Click
MsgBox("Thank you for using BrowsealV1.0", vbInformation)
Me.Close()
End Sub

Private Sub Progress_Timer_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Progress_Timer.Tick
Progress_bar.Visible = True
```

```
labProcess.Visible = True
Progress_bar.Value = Progress_bar.Value + 1
If Progress_bar.Value = 100 Then
Progress_Timer.Enabled = False
labProcess.Visible = False
' txt_ListBroser.Visible = True
If Flag = 1 Then
Detect.Show()
Detect.Focus()
ElseIf Flag = 2 Then
Browser.Show()
Browser.Focus()
ElseIf Flag = 3 Then
MaskingEntry.Text = " Web Blocker"
MaskingEntry.Show()
MaskingEntry.Focus()
ElseIf Flag = 4 Then
HistoryManager.Show()
HistoryManager.Focus()
ElseIf Flag = 5 Then
MaskingEntry.Text = " Masking Entry"
MaskingEntry.Show()
MaskingEntry.Focus()
End If

Progress_bar.Visible = False
Progress_bar.Value = 0
End If
End Sub

Private Sub PictureBox8_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox8.Click
Admin.Show()
End Sub

Private Sub PictureBox4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox4.Click
Try
Dim parts() As String = Split(My.User.Name, "\")
Dim username As String = parts(1)
Dim pathChrome As String = "C:\Users\" & username & "\AppData\Local\Temp"

Catch ex As Exception
MsgBox(ex.Message)
End Try
End Sub
End Class
Public Class HistoryManager
```

```
Dim f1, f2, f3, f4, f5 As Integer
Private Sub HistoryManager_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
Try
CheckBox1.Visible = False
CheckBox2.Visible = False
CheckBox3.Visible = False
CheckBox4.Visible = False
CheckBox5.Visible = False
CheckBox1.Text = ""
CheckBox2.Text = ""
CheckBox3.Text = ""
CheckBox4.Text = ""
CheckBox5.Text = ""
'Detect Browser from reg Level..
Dim readValue =
My.Computer.Registry.GetValue("HKEY_LOCAL_MACHINE\SOFTWARE\Client
s\StartMenuInternet\IEXPLORE.EXE",
"", Nothing)
'      MsgBox("The value is " & readValue)
Dim readValue1 =
My.Computer.Registry.GetValue("HKEY_LOCAL_MACHINE\SOFTWARE\Client
s\StartMenuInternet\Google Chrome",
"", Nothing)
'MsgBox("The value is " & readValue1)
Dim readValue2 =
My.Computer.Registry.GetValue("HKEY_LOCAL_MACHINE\SOFTWARE\Client
s\StartMenuInternet\Opera",
"", Nothing)
Dim readValue3 =
My.Computer.Registry.GetValue("HKEY_LOCAL_MACHINE\SOFTWARE\Client
s\StartMenuInternet\FIREFOX.EXE",
"", Nothing)
Dim readValue4 =
My.Computer.Registry.GetValue("HKEY_LOCAL_MACHINE\SOFTWARE\Client
s\StartMenuInternet\Avant.Browser",
"", Nothing)
If readValue <> "" Then
' MsgBox("I am here")
txt_ListBroser.Text = readValue.ToString
End If

If readValue1 <> "" Then
'MsgBox("I am here..1")
txt_ListBroser.Text = txt_ListBroser.Text + vbNewLine + readValue1.ToString
End If
If readValue2 <> "" Then
txt_ListBroser.Text = txt_ListBroser.Text + vbNewLine + readValue2.ToString
End If
```

```vbnet
If readValue3 <> "" Then
txt_ListBroser.Text = txt_ListBroser.Text + vbNewLine + readValue3.ToString
End If
If readValue4 <> "" Then
txt_ListBroser.Text = txt_ListBroser.Text + vbNewLine + readValue4.ToString
End If
' Dim f1, f2, f3, f4, f5 As Integer
Dim arr() As String
Dim i As Integer
arr = Split(txt_ListBroser.Text, vbNewLine)
For i = 0 To UBound(arr)
If CheckBox1.Text = "" Then
CheckBox1.Visible = True
CheckBox1.Text = arr(i)
ElseIf CheckBox2.Text = "" Then
CheckBox2.Visible = True
CheckBox2.Text = arr(i)
ElseIf CheckBox3.Text = "" Then
CheckBox3.Visible = True
CheckBox3.Text = arr(i)
ElseIf CheckBox4.Text = "" Then
CheckBox4.Visible = True
CheckBox4.Text = arr(i)
ElseIf CheckBox5.Text = "" Then
CheckBox5.Visible = True
CheckBox5.Text = arr(i)
End If
Next
Catch ex As Exception
MsgBox("Not a Good time to delete History, Try Later")
End Try
End Sub


Private Sub picClose_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles picClose.Click
Me.Close()
End Sub


Private Sub PictureBox1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox1.Click
'Get the UserName
Dim parts() As String = Split(My.User.Name, "\")
Dim username As String = parts(1)
Dim pathChrome As String = "C:\Users\" & username &
"\AppData\Local\Google\Chrome\User Data\Default"
' Dim whatnAll As String
'MsgBox(pathChrome)
If CheckBox1.Visible = True And CheckBox1.Checked = True Then
```

```
If CheckBox1.Text = "Google Chrome" Then
GoogleChrome(pathChrome)
End If
If CheckBox1.Text = "Internet Explorer" Then
IEDelete()
End If
'add Opera and Firefox code
End If
If CheckBox2.Visible = True And CheckBox2.Checked = True Then
If CheckBox2.Text = "Google Chrome" Then
GoogleChrome(pathChrome)
End If
If CheckBox2.Text = "Internet Explorer" Then
GoogleChrome(pathChrome)
End If
'add Opera and Firefox code
End If
If CheckBox3.Visible = True And CheckBox3.Checked = True Then
If CheckBox3.Text = "Google Chrome" Then
GoogleChrome(pathChrome)
End If
If CheckBox3.Text = "Internet Explorer" Then
GoogleChrome(pathChrome)
End If
'add Opera and Firefox code
End If
End Sub

Private Sub GoogleChrome(ByVal pathChrome As String)
Dim flag As Integer = 1
Try
If checkHist.Checked = True Then
' MsgBox("User Name" & username)
If My.Computer.FileSystem.FileExists(pathChrome & "\History") Then
If (checSave.Checked = True) Then
'take back up
My.Computer.FileSystem.CopyFile(pathChrome & "\History", "\FileBackup")
End If
'delete the file
'whatnAll = "History"
My.Computer.FileSystem.DeleteFile(pathChrome & "\History")
flag = 1
Else
MsgBox("Hurray, You Dont have any history data at the moment..!!", vbInformation)
flag = 0
End If
End If
If checkCookies.Checked = True Then
```

```
If My.Computer.FileSystem.FileExists(pathChrome & "\Cookies") Then
My.Computer.FileSystem.DeleteFile(pathChrome & "\Cookies")
flag = 1
End If
End If
If checkData.Checked = True Then
If My.Computer.FileSystem.FileExists(pathChrome & "\Login Data") Then
My.Computer.FileSystem.DeleteFile(pathChrome & "\Login Data")
flag = 1
End If
End If
If checkCache.Checked = True Then
If My.Computer.FileSystem.FileExists(pathChrome & "\Bookmarks") Then
My.Computer.FileSystem.DeleteFile(pathChrome & "\Bookmarks")
flag = 1
End If
End If
Catch ex As Exception
flag = 0
MsgBox("Please Close Google Chrome Browser and Try Again", vbInformation)
End Try
If (flag = 1) Then
MsgBox("Operation Completed; Perfomance Boosted")
End If
End Sub


Private Sub IEDelete()
Try
If checkHist.Checked = True Then
Shell("RunDll32.exe InetCpl.cpl,ClearMyTracksByProcess 1")
MsgBox("IE Deletion Operation Completed", vbInformation)
End If
Catch ex As Exception
MsgBox("IE Operation Failed Please Try Again")
End Try
End Sub
End Class

Public Class MaskingEntry
Private Sub MaskingEntry_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
rtfHosts.Text =
System.IO.File.ReadAllText("C:\Windows\System32\drivers\etc\hosts")
rtfHosts.AppendText(vbLf)
End Sub

Private Sub PictureBox3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox3.Click
```

```vb
End Sub

Private Sub PictureBox4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox4.Click
MsgBox("Tips :  " & vbNewLine & "1. Please Seperate Each entry with Line" &
vbNewLine & "2. Comments must prefixed with # Symbol",
MsgBoxStyle.Information)
End Sub

Private Sub PictureBox2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox2.Click
MsgBox("Do you want to close the mask process Window ?", MsgBoxStyle.YesNo)
If (MsgBoxResult.Yes) Then
Me.Close()
End If
End Sub

Private Sub PictureBox1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox1.Click
Try
MsgBox("Are you sure to save the Mask File ?", MsgBoxStyle.OkCancel, "Save
File")
If (MsgBoxResult.Ok) Then
System.IO.File.WriteAllText("C:\Windows\System32\drivers\etc\hosts",
rtfHosts.Text)
MsgBox("Mask File Saved, Please open the browser verify the Mask Process",
vbInformation)
End If
Catch ex As Exception
MsgBox("Mask process Failed, Please try again", vbCritical)
End Try
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
rtfHosts.AppendText(vbLf)
End Sub
End Class

Public Class ResponseManager
Private Sub ResponseManager_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
End Sub
Private Sub Label1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label1.Click
End Sub

Private Sub PictureBox2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox2.Click
```

```vb
'Try
'   If My.Computer.Network.Ping(txtCommand.Text) Then
'       txtResult.Text = My.Computer.Network.Ping(txtCommand.Text)
'       MsgBox("Server pinged successfully.")
'   Else
'       MsgBox("Ping request timed out.")
'   End If
'Catch ex As Exception
'End Try


' Private Sub StartProcess()
txtResult.ForeColor = Color.Red
txtResult.Text = "~~~~~~~~~~~~~~~~~~~~~~~~BROWSEAL VERSION
1.0~~~~~~~~~~~~~~~~~~" & vbNewLine
'txtResult.ForeColor = Color.G
Dim Proc = New Process()
Proc.StartInfo.FileName = "ping"
Proc.StartInfo.Arguments = txtCommand.Text
Proc.StartInfo.RedirectStandardOutput = True
Proc.StartInfo.RedirectStandardError = True
Proc.EnableRaisingEvents = True
Application.DoEvents()
Proc.StartInfo.CreateNoWindow = True
Proc.StartInfo.UseShellExecute = False
AddHandler Proc.ErrorDataReceived, AddressOf proc_OutputDataReceived
AddHandler Proc.OutputDataReceived, AddressOf proc_OutputDataReceived
Proc.Start()
Proc.BeginErrorReadLine()
Proc.BeginOutputReadLine()
'Proc.WaitForExit()
End Sub
Delegate Sub UpdateTextBoxDelg(ByVal text As String)
Public myDelegate As UpdateTextBoxDelg = New UpdateTextBoxDelg(AddressOf
UpdateTextBox)
Public Sub UpdateTextBox(ByVal text As String)
Try
txtResult.AppendText(text & vbNewLine)
Catch ex As Exception

End Try
End Sub


Public Sub proc_OutputDataReceived(ByVal sender As Object, ByVal e As
DataReceivedEventArgs)
Try
'On Error Resume Next
If Me.InvokeRequired = True Then
Me.Invoke(myDelegate, e.Data)
```

```
Else
UpdateTextBox(e.Data)
End If
Catch ex As Exception
End Try
End Sub


Private Sub txtResult_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)
End Sub


Private Sub PictureBox1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox1.Click
txtResult.ForeColor = Color.Red
txtResult.Text = "~~~~~~~~~~~~~~~~~~~~~~~~BROWSEAL VERSION
1.0~~~~~~~~~~~~~~~~~~" & vbNewLine
'     txtResult.ForeColor = Color.Green
Dim Proc = New Process()
Proc.StartInfo.FileName = "tracert"
Proc.StartInfo.Arguments = txtCommand.Text
Proc.StartInfo.RedirectStandardOutput = True
Proc.StartInfo.RedirectStandardError = True
Proc.EnableRaisingEvents = True
Application.DoEvents()
Proc.StartInfo.CreateNoWindow = True
Proc.StartInfo.UseShellExecute = False
AddHandler Proc.ErrorDataReceived, AddressOf proc_OutputDataReceived
AddHandler Proc.OutputDataReceived, AddressOf proc_OutputDataReceived
Proc.Start()
Proc.BeginErrorReadLine()
Proc.BeginOutputReadLine()
'Proc.WaitForExit()
End Sub


Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
txtResult.Text = ""
txtCommand.Text = ""
End Sub


Private Sub picClose_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles picClose.Click
Me.Close()
End Sub
End Class
```

4.2 SCREENS SHOTS

This is the Home page of Browseal Software.
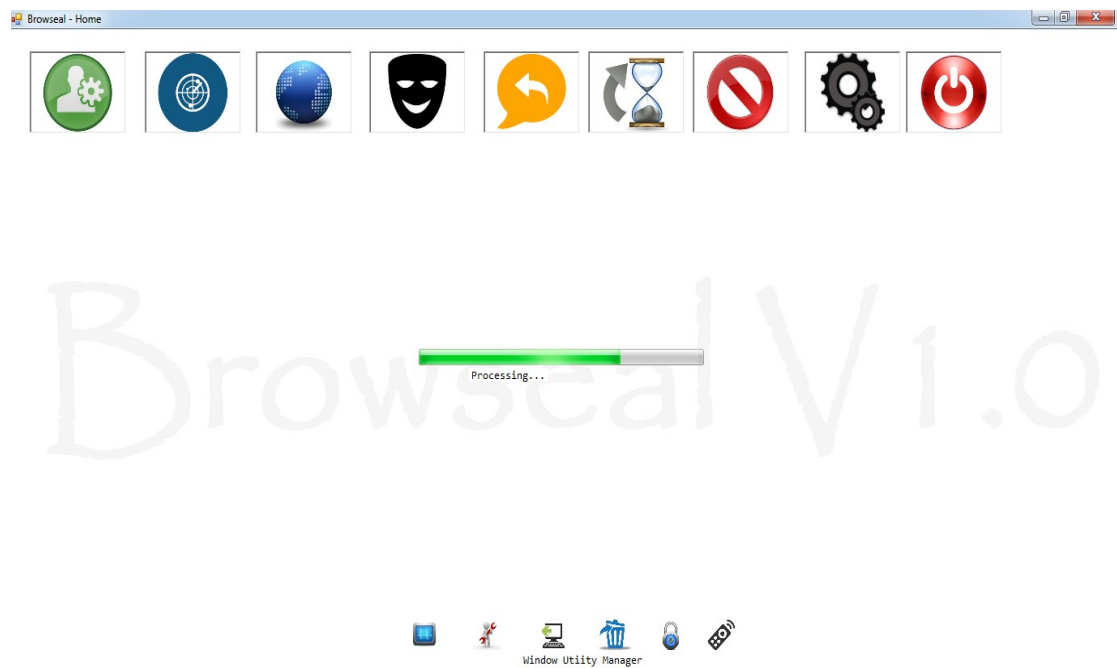
Fig4.1 Homepage

All the browsers available in the system are detected
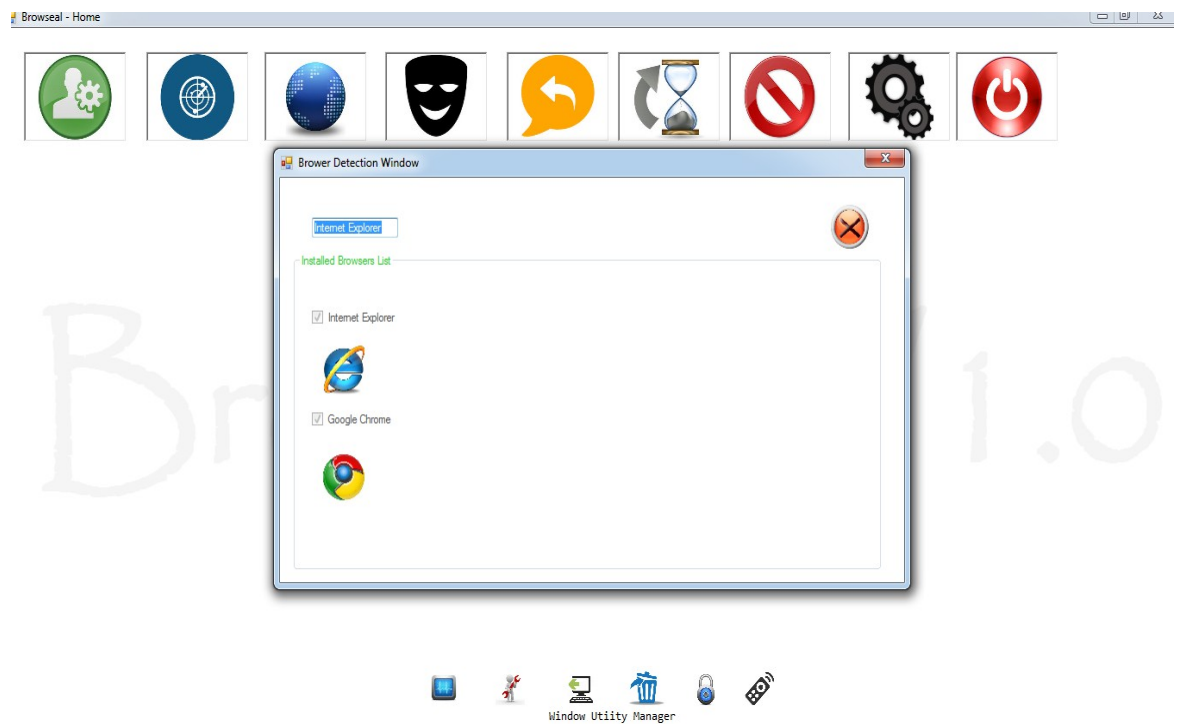
Fig 4.2 Detection of Browsers

This pages helps to navigate between different web pages
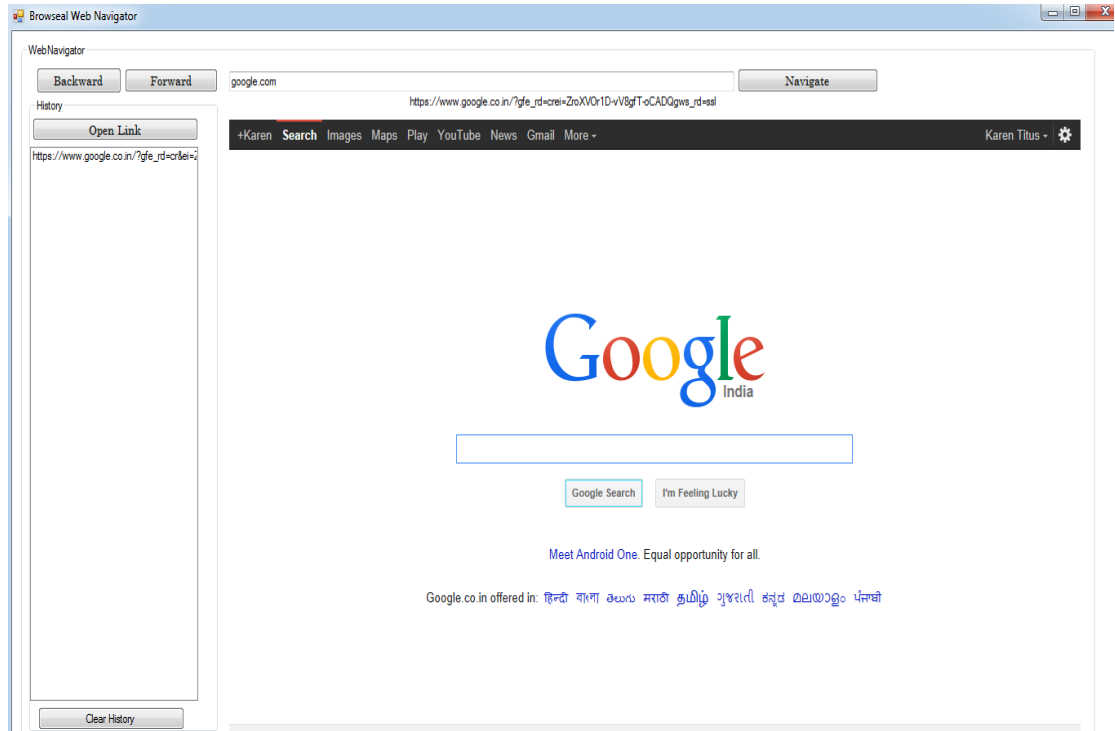
Fig 4.3 Web Navigate

This page masks one browser with another browser according to the input given

Fig 4.4 Masking

Keeps track of the response time of URL's
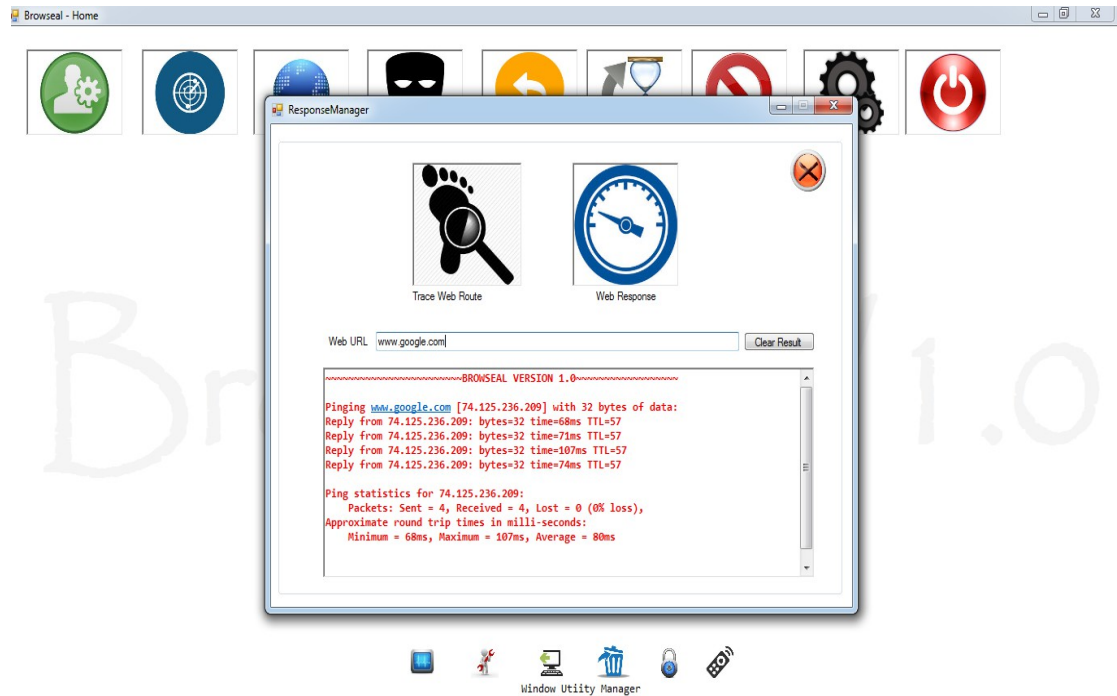
Fig 4.5 Response Manager1
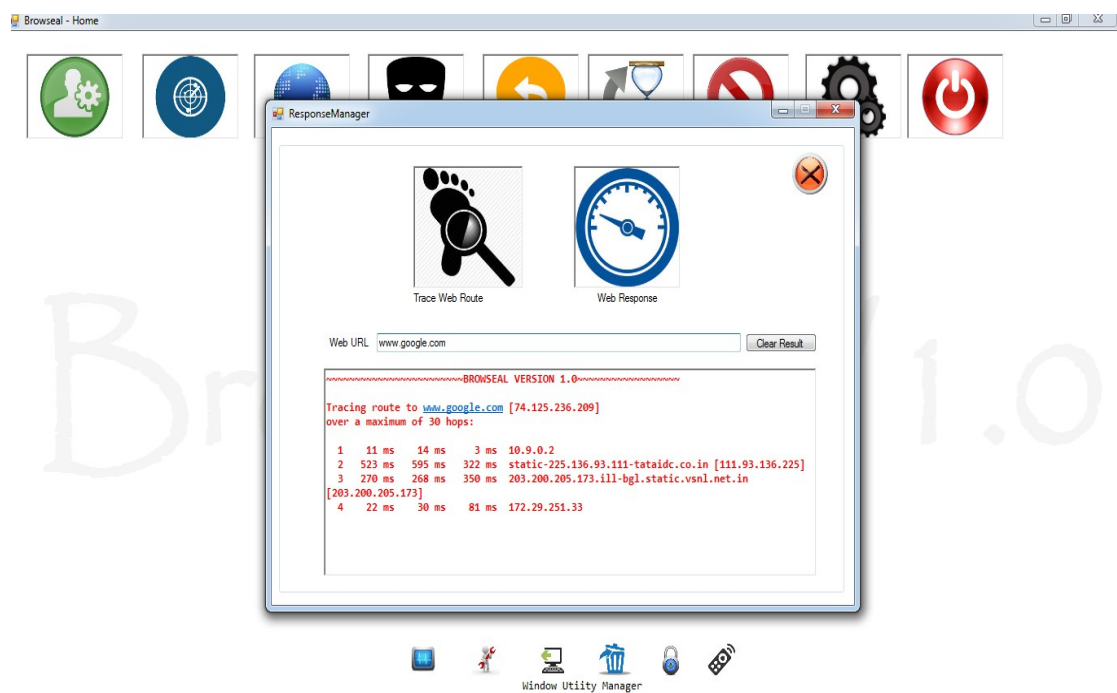
Fig 4.6 Response Manager2

History of multiple browsers can be deleted
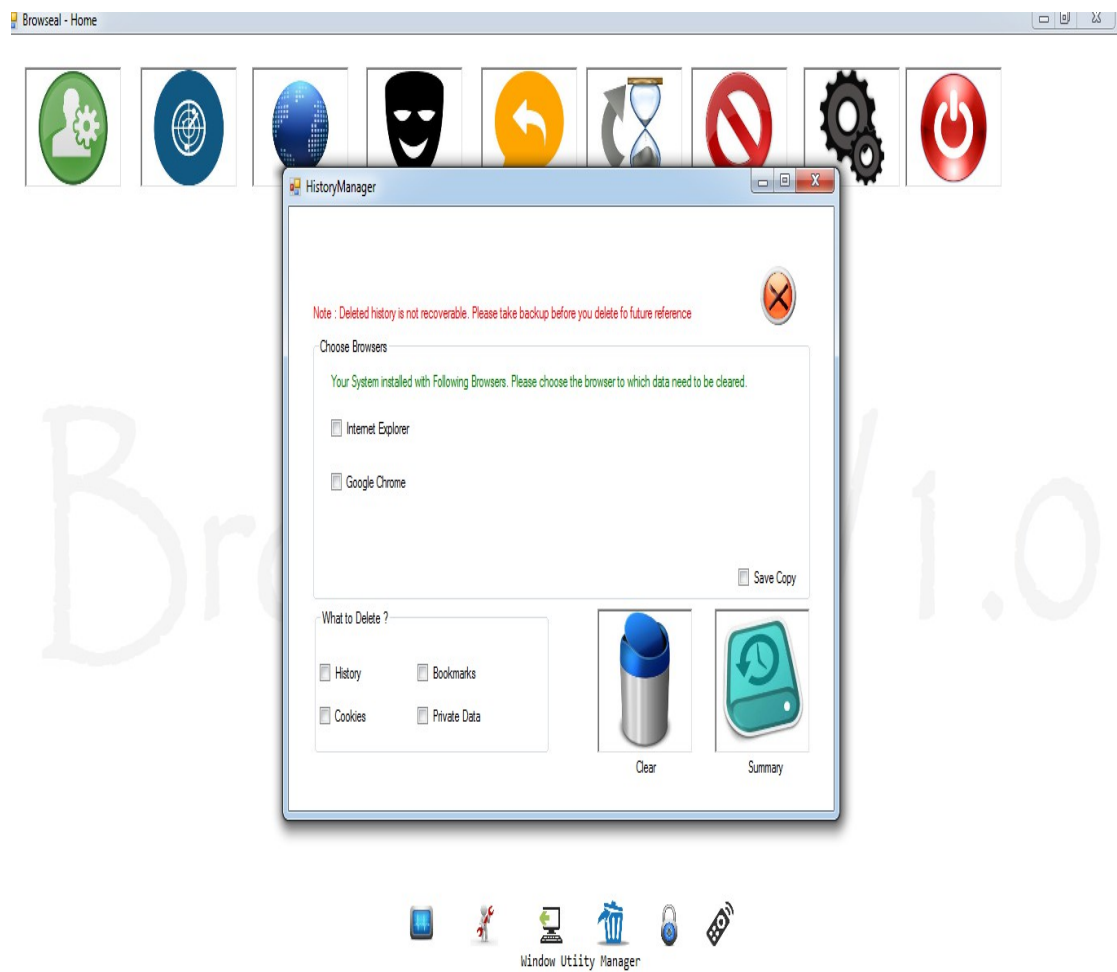
Fig 4.7 History Manager

The History Deletion process.
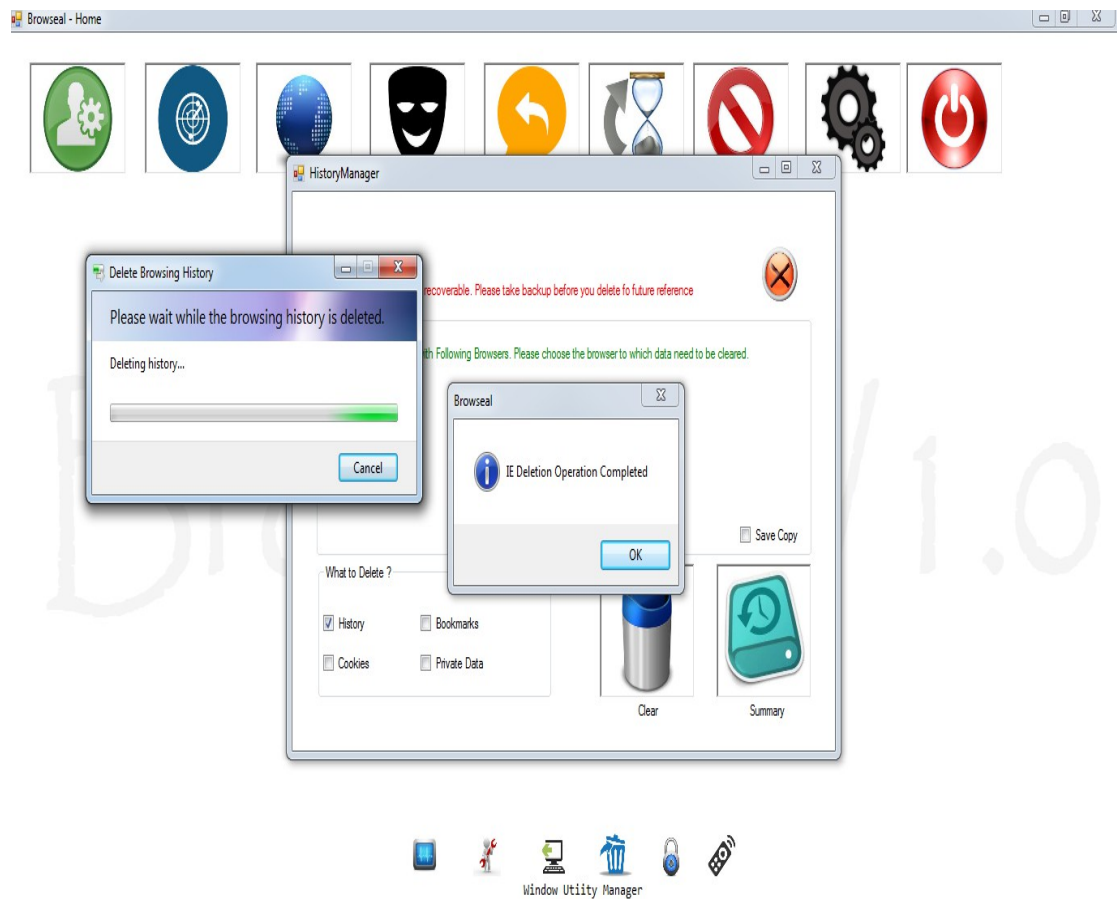
Fig 4.8 Deleting Process

5Here, the web pages can be blocked

Fig 4.8 Web Blocker

This is the administrator login form

Fig 4.9 Browseal Master

5. TESTING

Testing is a set of activities that can be planned in advance and conducted systematically. For this reason a template for software testing should be defined for the software process. Testing begins at the component level and works outward toward the integration of the entire computer based system. Different testing techniques are appropriate at different points in time. It is conducted by the developer of the software and an independent test group.

A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against customer requirements. A strategy must provide guidance for the practitioner and set of milestones for the manager. Because the steps of the test strategy occur at a time when dead line pressure begins to rise, progress must be measurable and problems must surface as early as possible.

Unit Testing

Unit testing is a method by which individual units of source code. Each unit can be entire module but it is more commonly an individual function or procedure.

The Unit testing was used for testing the application. Each module was tested based on the source code. Boundary condition for each record which was fetched from the database was checked, that is, if the data exists in the database or not. The code was tested for array values, if the values were null or it consists of values. After integrating the all modules, the application was tested different components that are, passing information from one page to another page.

Integration Testing

Integration testing is a systematic technique for constructing the program structure at the same time conducting test to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. All components are combined in advance. The entire program is tested as a whole.

The program is constructed and tested in small increments, where errors are easier to isolate and correct; interfaces are more likely to be tested completely, and as systematic approach may be applied.

5.1 TEST CASES

Table5.1. TEST CASES

Test case-ID	Screen Name	Description	Expected Output
1	Login	Tried to login with the correct details	Logged in to the system
2	Login	Tried to login with incorrect login details	Error Message
3	Navigate	Tried to press browse button with writing the url	Please mention a url
4	Masking	Tried to run a url without masking it to any other url	Please mention a url to be masked

5.2TEST REPORTS

Incorrect User name or Password form
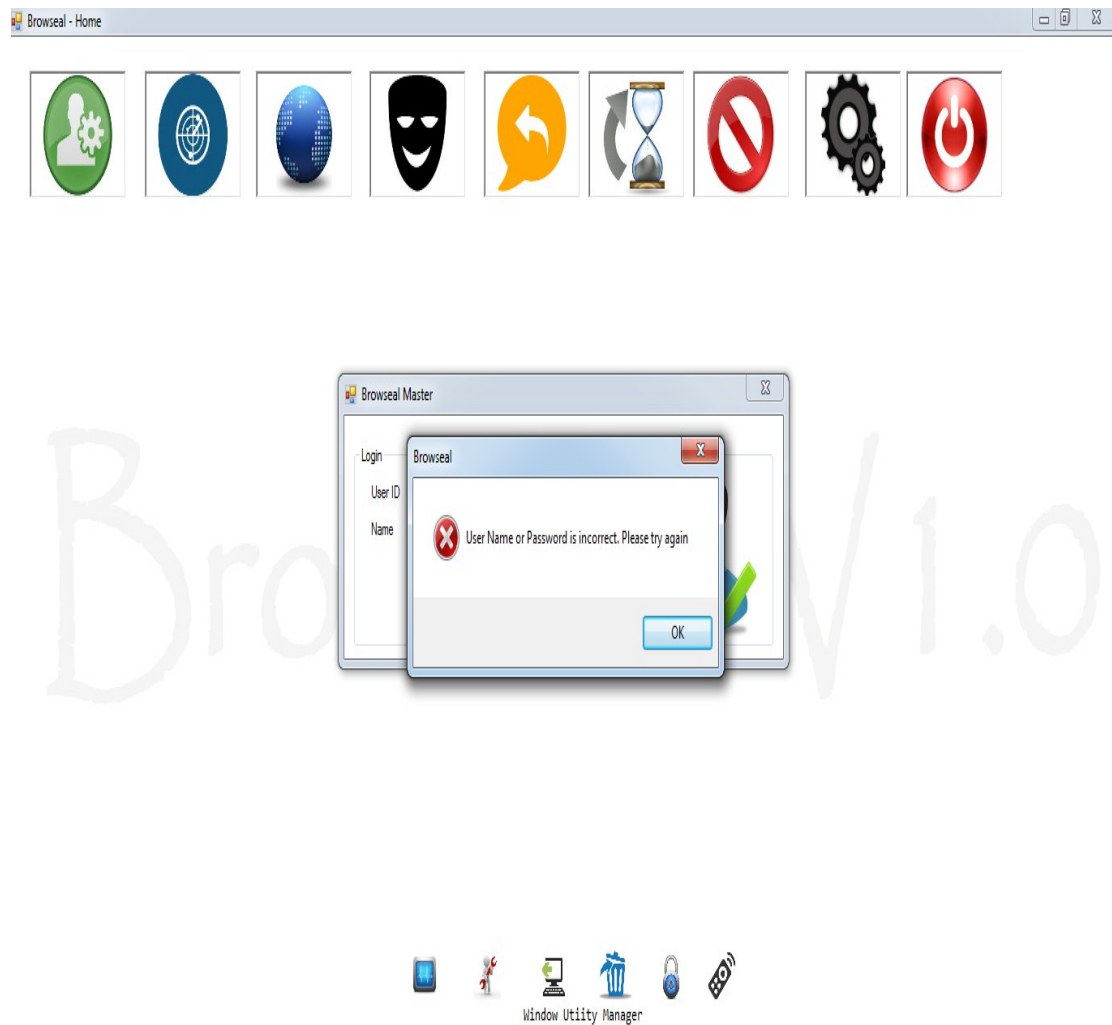
Fig 5.1 Admin Test form

Message box appears when the close button is pressed.
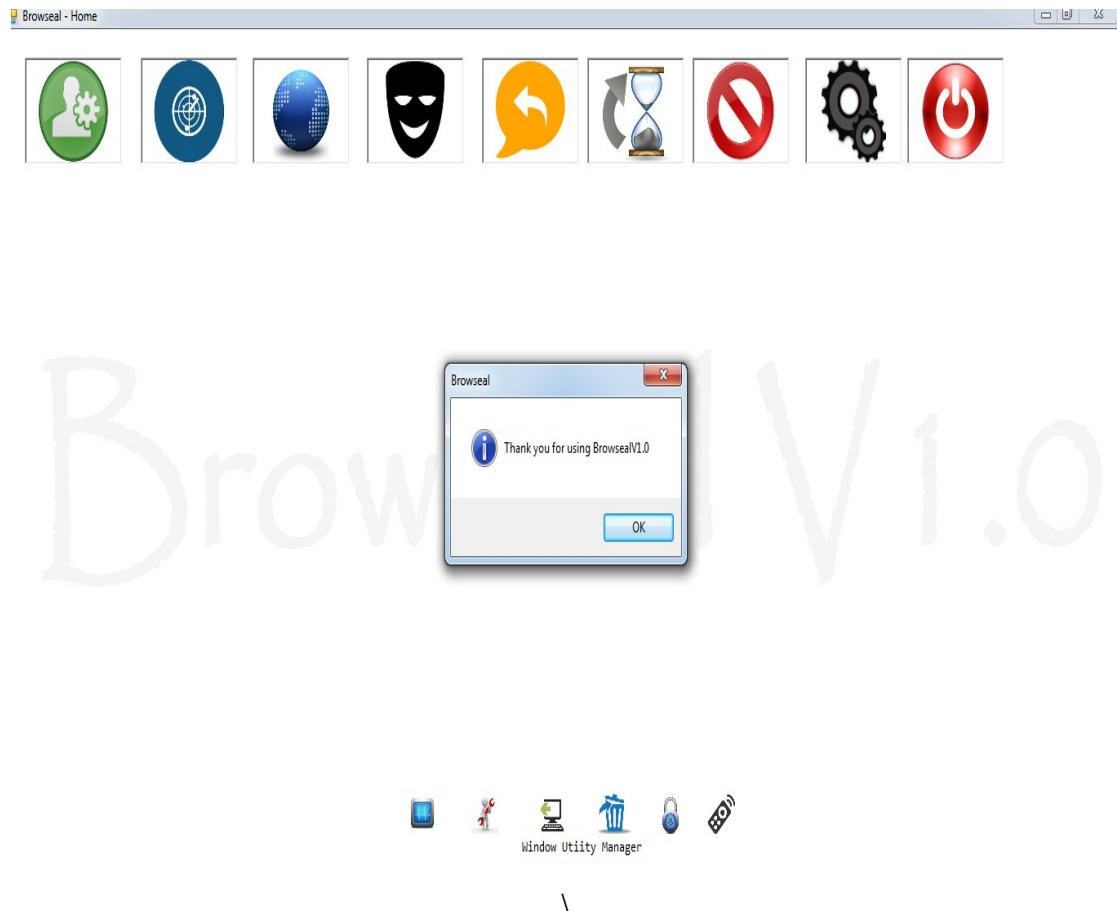
Fig 5.2 Admin Test form

Once the delete history button is pressed, all the selected browser's history is deleted.
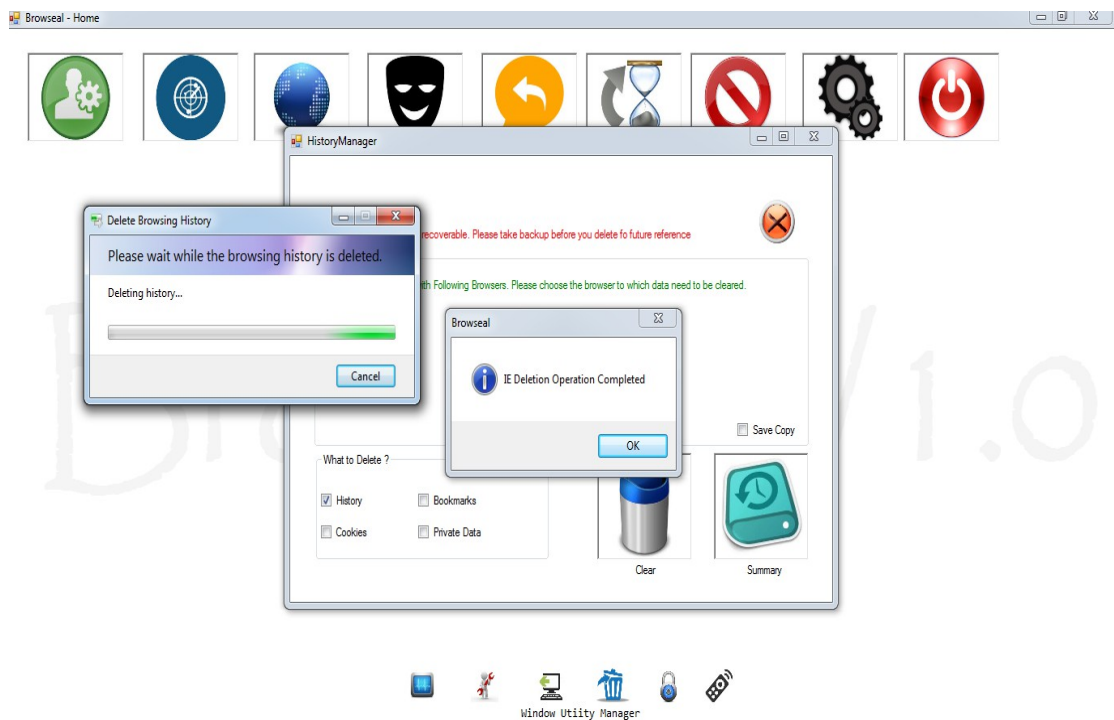
Fig 5.3 Deleting History Form

6. CONCLUSION

Multiple browser configuration system is designed to manage multiple browsers on a system in an IT organization with useful utilities and tools to cost-effectively manage and secure web browsers and web applications at a global scale. Our browser management platform provides IT with granular control of browser environments, to optimize web application compatibility and security.

6.1 LIMITATIONS.

- Limited applications on systems which do not use multiple browsers.

- The user needs to have an adequate amount of knowledge about the usage of the software.

6.2 ENHANCEMENTS

It is not possible to develop a system that makes all the requirements of the user. User requirements keep changing as the system is being used. Some of the future enhancements that can be done to this system are:

- As the technology emerges, it is possible to upgrade the system and can be adaptable to desired environment.
- Because it is based on an easy interface tool, any further changes can be easily adaptable.
- Based on the future security issues, security can be improved using emerging technologies.
- A Vpn connection can be established between systems and browsers in different systems.
- The application can be expanded through vpn connections and applied in cyber café for improved security and not just in IT organizations.

REFERENCES

[1] "Option Explicit and Option Strict in Visual Basic .NET and in Visual Basic". Support. Microsoft. 19 March 2008. Retrieved 22 August 2013.

[2] "Visual Basic Version of Hello, World". MSDN - Developer Center. Retrieved 20 January 2010.

[3] Mackenzie, Duncan (2006). "Navigate The .NET Framework And Your Projects With The My Namespace". MSDN Magazine Visual Studio 2005 Guided Tour 2006. Microsoft.

TABLE OF CONTENTS