

Web Browser in C#

Industrial Programming –Coursework 1

Jacob Cooper | F20SC | 31/10/19

# Introduction

This report will cover my implementation of a Web Browser in C#. I will show which requirements I have covered as well as a short comment on each. Then I go through my code and go over my design considerations especially any parts which make the code of higher quality e.g. higher performance, Modularity, readability etc. The next part is the User Guide, this is an in-depth guide on how to use my browser from the perspective of a User. After that the Developer Guide is there to go through my code in order to make a developer understand how I implemented this Web Browser. Then I show some Testing of my applications, any erroneous results, basic test to show my application works and some possible errors which I have handled. I assumed that all libraries were allowed to be used except the Web browser class.

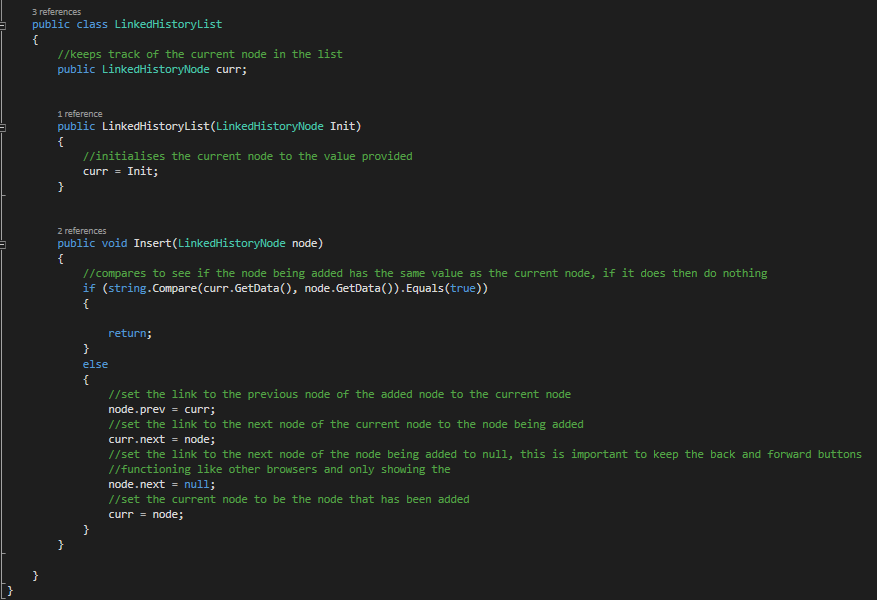
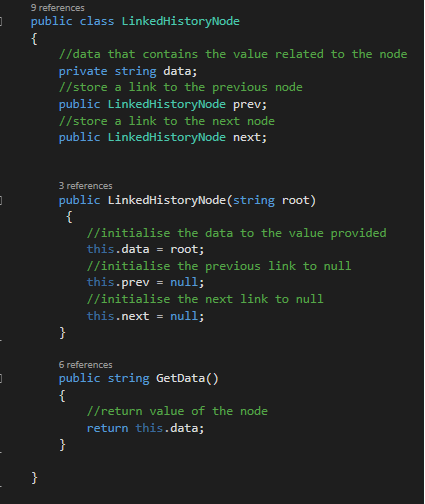
# Requirements’ Checklist

|  |  |  |
| --- | --- | --- |
| Requirement | Delivery | Comment |
| Sending Http requests | In Full | The user can type in a valid URL into the search bar and it will send a http request to the correct URL |
| Receiving HTTP response | In Full | Response is received if there is one and any Http response status is displayed clearly. The HTML content of the request is displayed clearly in the browsers main window |
| Display | In Full | Shows the Http status code, the Title of the web page at the top of the browsers main window and supports reloading of the current page. |
| Home Page | In Full | Users can change their home to anything they please and this homepage is saved between sessions and loaded on startup |
| Favorites | In Full | Users can add a favorite along with a name relating to it , Modification and deletion of these favorites is supported, the user can either click on the name of the favourite or the link to go to the associated web page. Favourites are saved between sessions and loaded up on start up. |
| History | In Full | History is a list of URLs that keep track of which web pages a user has been to and allows the user to go back to these and any point by clicking the relevant link in the history list. Users can also navigate forward and back to previous pages, this operates much like the forward and back buttons in google chrome |
| GUI | In Full | The GUI is laid out clearly so that all functionality is accommodated for and this includes use of menus and buttons to ease accessibility |

# Design Considerations

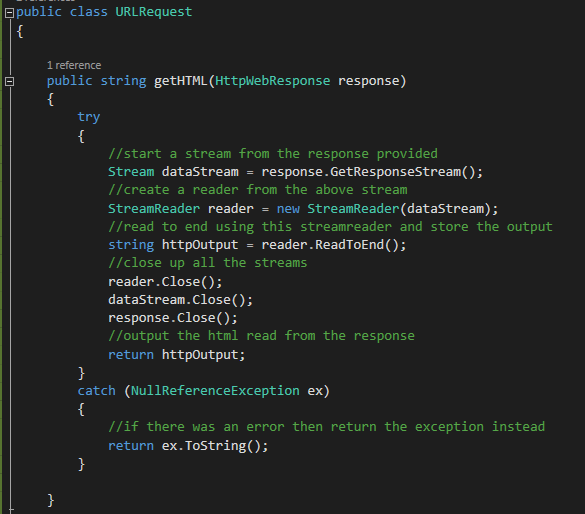
I decided to reduce my use of data structures to the minimum possible as in development I felt that this, along with having to display the results in a container, would slow down my application so wherever possible I used the container to not only display my results but store them for the session. I used this technique in storing the list History and Favourites.

One place where I implemented my own data structure was in the use of the back and forward buttons, here I implemented my own doubly linked list with two associated classes as seen below.



The reason I did this was because I could implement an efficient use of a Doubly-linked list that fit what my use for it was. Using a library would probably be less efficient and I couldn’t use my previous method (of using the as I used an implementation that is similar to Google Chrome where it only tracks the path from start to your current page and refuses to allow access to anything else .Therefore there is nothing to display and no container is used. This means the back and forward buttons are more up to the industrial standard seen today but it also mean I could not use the History list in my implementation of the back and forward buttons. I believe this to be the better approach, even though I have to use another data structure in order to implement it, because it follows the standard set by most browsers of today and makes it so the back and forward buttons intuitively make sense in what they do.

I decided to try and keep my methods as generic as possible in order to maintain modularity and reusable code so that implementations such as that of the Http request and handling could be used in another application without having to change much or anything.



I used “try and Catch” blocks in order to catch and handle errors so they do not crash my program. The handling of the error is mostly just making the user aware what error has happened such as displaying the http response status even if there is an error so that the user is informed of what has happened. (\*insert example of try and catch block here)

When designing the GUI I mostly kept a similar layout to most browsers but I with a few differences. I have the Status code and title both displayed in their own text box above the main window where the html is displayed. I chose to do this as I think it is clearer what each item is although I do think its space usage is a bit inefficient. Similarly I didn’t want to overcrowd the menu bar at the top so I decided to place my Favourites next to the main window as I think this makes it clear what it is a keeps it from cluttering the menu at the top of the screen

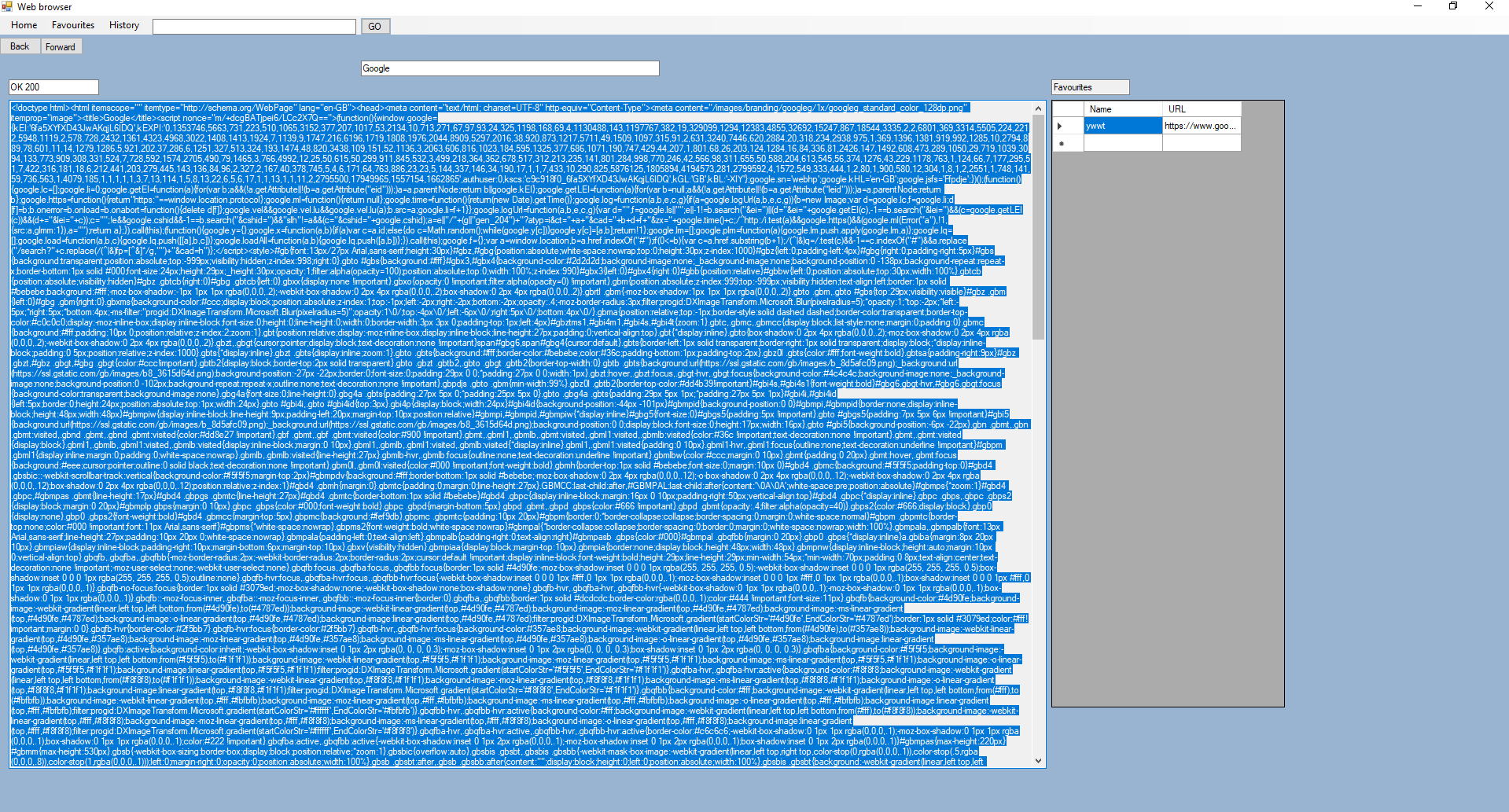
I used a total of 4 Classes : one for the form, 2 for the doubly linked list and one for handling web requests and. The form class handles all actions related to clicking the form or changing what is displayed. One class for the doubly linked list is the node classs

# User Guide

## On startup

When you start up the browser you should see a main window which is displaying the html of the homepage (if there is no homepage set then it defaults to <https://www.google.com>). Above this you should see two Labels which show the Status Code return by the web page and the Title of the web page respectively. To the side of the window you should see a table which is labelled at the top as favourites, this table is the favourites that have been saved (it will be empty if there are no saved favourites) and it should have a column for the Name of the favourite and a column for its associated URL.

At the very top of the screen you should see a menu strip containing Home, favourites ,History, a search bar and a button that is labelled ‘GO’. Just below this menu strip you should see two buttons labelled ‘forward’ and ‘back’.



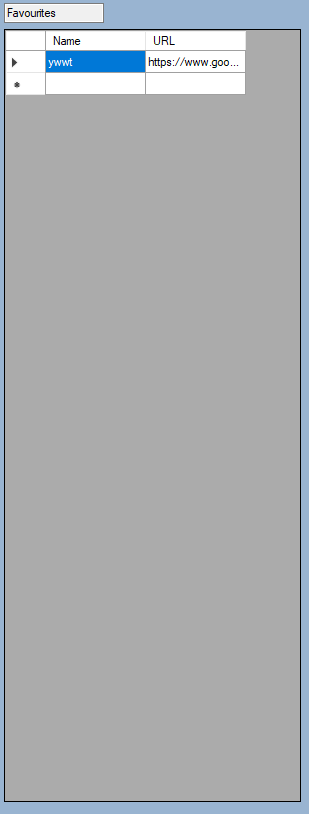
## The display

The main parts of the Web browser are the main window (which displays the HTML response from the webpage), the Status Text Box (which displays the Status code returned by the response from the webpage) and the Title Text Box (which displays the Title of the current webpage). These three parts Display everything needed from a webpage.

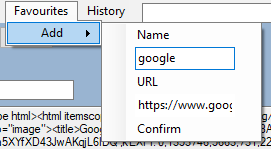


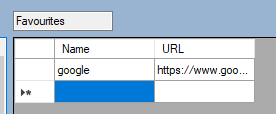
## Favourites

The favourites is displayed to the right of the main window and should show the name of the favourite as well as its link.



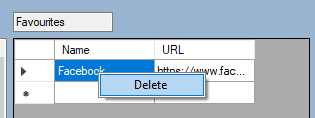
Adding favourites is done by clicking the ‘Favourites’ option at the top of the screen then clicking the ‘Add’ option that appears in the drop down menu. At this point another drop-down list should have appeared with a text box labelled ‘Name’ which should be empty , text box labeled ‘URL’ which should have the URL of the current page already loaded into it and an option at the bottom called ‘confirm’. If the current page is the one you want to favourite then just give it a name in the ‘Name’ text box and click confirm and it should add it to the table of favourites. If you want to change the URL then just input the URL you want in the ‘URL’ Text box along with a name in the ‘Name’ text box and click confirm.





If you wish to go to the favourite URL, you can either click on the name associated with the URL in the favourites data grid or the URL itself and it will take you to the web page associated with that URL.

To delete a favourite, first select it so that it is highlighted blue, right click it (A delete option should appear) and then click the delete option.

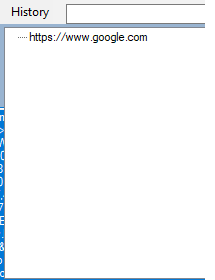


You can modify the name or URL by simply clicking on the name or URL you want to chnage

Favourites will be stored between sessions meaning next time the browser is started up all the favourites will still be there.

## History

History can be viewed by clicking the History option of the menu strip at the top. The History List should now be visible. This is a list of all web pages you have visited previously (including in previous sessions).

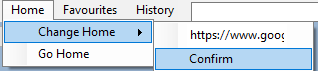


You can go to any link in the History List by simply clicking it.

History is saved between sessions so you will see history from previous sessions in the History List.

## Home

The home page by default is <https://www.google.com>. You can change this by: clicking the ‘Home’ option in the menu strip at the top of the page, then clicking the ‘Chang Home’ option which appears. Now a new drop down menu with a text box and a confirm option should appear. The text box should automatically have the URL of the current page in it, if this is the homepage you want to change to then all you need to do is click confirm. If it is not the URL you want to change your homepage to then just put the URL you want in the text box and click confirm.



You can return to your homepage at any point by clicking the ‘Home’ option in the Menu strip and then clicking the ‘Go Home’ option that appears.

## searching

You can go to any URL by it into the search bar at the Top and clicking the ‘GO’ button. The HTML response should load into the main window, the status in the status text box and the title in the title text box.

## Back and Forward Buttons

The back button will take you to the last page you were one (if there isn’t one then nothing happens) and the forward button will take you to the next page you were on (if there isn’t one then nothing happens). This works like Google Chromes back and forward button in that it will only keep a track of the page path you took to get to the page you are on currently.

# Developer Guide

## Display

## Favourites

## History

## Home

## searching

## Back and Forward Buttons

# Testing

# Conclusions

You might like the photo on the cover page as much as we do, but if it’s not ideal for your report, it’s easy to replace it with your own.

Just delete the placeholder picture. Then, on the Insert tab, click Picture to select one from your files.