**F20SC** : Industrial Programming

**Coursework 1** : Developing a Simple Web Browser

**Date of submission** : November 2, 2022

**Name** : Irfanuddin Syed

**Heriot-Watt ID** : H00389591

**Degree Programme** : BSc. Computer Science (Artificial Intelligence)

**Campus of Study** : Dubai

Table of Contents

[1. Introduction 1](#_Toc149562533)

[2. Requirements Checklist 2](#_Toc149562534)

[3. Design Considerations 2](#_Toc149562535)

[3.1. Class Design 2](#_Toc149562536)

[3.2. Data Structures 3](#_Toc149562537)

[3.3. GUI Design 3](#_Toc149562538)

[3.4. Advanced Language Constructs 5](#_Toc149562539)

[4. User Guide 5](#_Toc149562540)

[5. Developers Guide 5](#_Toc149562541)

[6. Testing 5](#_Toc149562542)

[7. Conclusions 5](#_Toc149562543)

# Introduction

This report is a documentation for the design, development, and functionality of a simple web browser I implemented in C#. The document outlines the functional requirements fulfilled and explains my design choices. I have also included comprehensive user’s guide and developer’s guide to help both groups understand and interact with the system.

# Requirements Checklist

* ***Sending*** HTTP request messages for URLs typed by the user.
* ***Receiving*** HTTP response messages and display the contents of the messages on the interface.
* ***Displaying*** the page title and response code at the top of the browser.
* Allowing the user to ***refresh*** the page by sending another HTTP request for the current web page.
* The user can create and edit a ***home page*** URL. The Home page URL is loaded on the browser’s start up, and is initialised with the Heriot-Watt website (<https://hw.ac.uk>)
* The user can add a URL for a web page requested to a list of ***favourite*** web pages. The user can associate a name with each favourite URL. Support for favourite items modification and deletion is provided.
  + The user can request a favourite web page by clicking its name on the Favourites list. On the browser’s start up, the favourites list is loaded to the browser.
* The browser maintains ***history***, i.e., a list of URLs, corresponding to the web pages requested by the user.
  + The user can navigate to ***previous and next pages***, and jump to a page by clicking on the links in the History list. On the browser’s start up, the history list is loaded to the browser.
* The application provides a ***bulk download*** facility. Upon entering a file path, for each URL in the file, the response code, byte size, and URL are saved to a downloads file and displayed on the screen.
* A simple ***GUI*** has been provided to perform the operations discussed above.

# Design Considerations

## Class Design

The web browser is separated into 7 different classes:

* Url.cs : This class defines a URL object which fetches and stores the contents, size, response code and title of the string URL provided and provides access using getter/setter functions using the HTTP class.
* Favorites.cs – This class takes a name and a URL object which represents a favorite webpage. The class is used to organize and manage user-defined favorite webpages within an application, also using the previously defined URL class for handling URL data.
* ReadWrite.cs – This class is designed to handle reading and writing operations from the specified file path for the bulk download facility. It has methods to read from a file, returning its content as an array of strings, and to write a string to a specified file. Exception handling is employed to manage potential file IO errors, outputting error messages to the console when encountered.
* SetNewHomepage.cs – The class displays a dialogue box where a user can specify a new homepage URL for a browser. When prompted with a URL object representing the current homepage, it opens the dialogue box with an input field (by default set to the current homepage URL) with submit and cancel buttons. Users can either submit a new URL, which updates the URL object and closes the dialog, or cancel to close the dialog without making any changes.
* AddToFavorites.cs – This class makes a dialogue box for adding a new favorite or editing an existing favorite in a favorites list. It takes 2 inputs: a URL (used to create a URL object) and a name for the favorite. If add is clicked, the new favorite is added and if edit is clicked, the existing favorite is updated with the new values.
* MainWindow.cs – This class connects all my classes and functionality for all my buttons. It handles back/next controls, URL input, history, favorites, and bulk download features, along with UI updates and event handling for user interactions within the browser.
* Program.cs – Main Method

## Data Structures

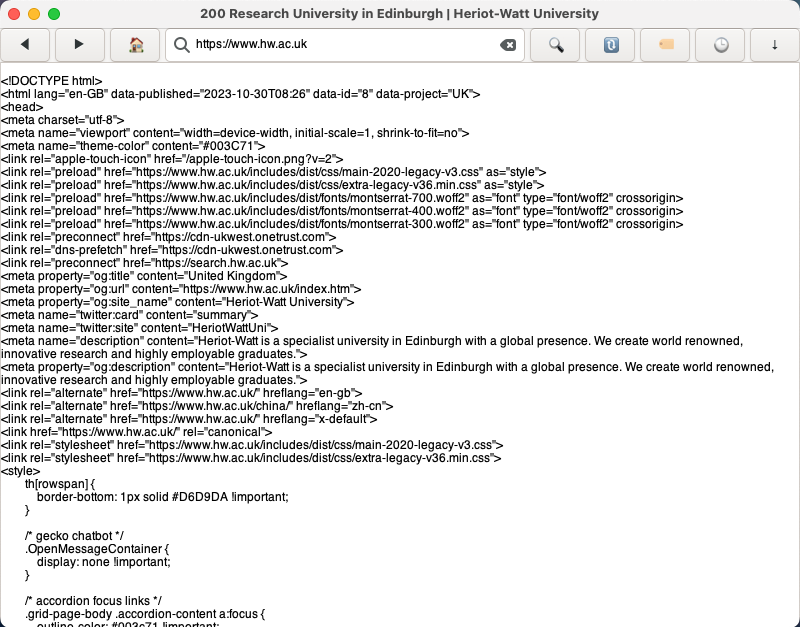
Throughout my coursework I employed different data structures to achieve my functionalities and make my code modular.

I used **classes** to set objects with their parameters (URL and Favorites) and for specific behavior handling (Read/Write, adding to favorites, setting homepage and the main browser window)

**Lists** are used in MainWindow and AddToFavorites for managing history, favorites, and bulk download URLs to provide a dynamic method to store the favorites objects and URLs.

**Arrays** are used in readwrite and MainWindow classes for reading file lines and managing bulk download URLs, wherever there is a fixed size collection of elements.

## GUI Design



The UI consists of the functional buttons all in the top bar for easy accessibility. The components consist of:

* The label at the top gives the status code and title of the current page
* Back and next buttons, to allow the user to move between recent pages,
* home button, to allow the user to go to the home page. Double clicking the home button opens a dialogue box to allow the user to set their own homepage.
* Search bar for the user to enter the URL they wish to visit.
* The user can either press the go button next to the search bar to load the URL contents or achieve the same results by pressing on the enter key.
* The refresh button allows the user to reload the HTML content on the screen.
* The bookmark button is used to store favorites, clicking on the button opens a popover of -existing favorites (with edit and delete button) and an add favorite button which in turn opens a dialogue box to enter the name and URL. We can click the names set to visit the favorited pages.
* The history button allows us to view the previously visited URLs. We can click on the page titles to visit them or click the delete button next to them to delete them individually. There is also a button to clear the entire history.
* The download button opens a popover which takes as input the file path that we want to download.
* The screen below displays the contents of the pages we visit, or the bulk download file.
* Keyboard shortcuts can be used to : refresh (ctrl + r) and go to homepage (ctrl + h)

# User Guide

1. **Web browser Home Screen:** this is the default home screen for the browser set to university page. The user can enter desired URL in the search bar and either click enter, or click the search button to the right of the search bar. After submitting, the HTML contents of the URL will be displayed on the display below the top bar.

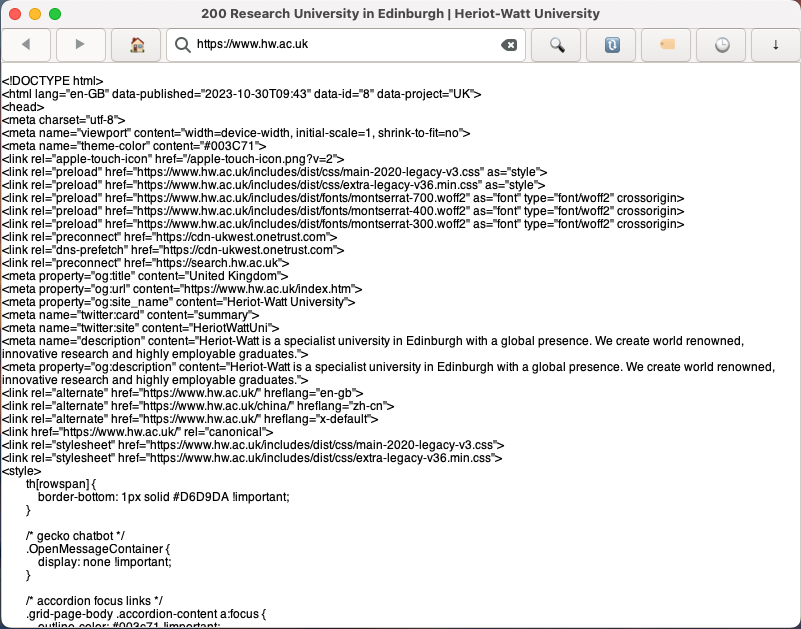


Figure 1: home screen

1. **Back / Next Button:** Click on the back or next button to go to that page in history. The buttons aren’t clickable when you are at the first page in history (back button disabled) and when you are at the end of history (next button disabled)



Figure 2: Back and Next button



Figure 3: Back disabled



Figure 4: Next disabled

**Home button:** The user can click on the home button to be sent to the set homepage. If the user double clicks on the home button, a dialogue box will appear which will take the new homepage as input, and that will be the page they are sent to when they click on the home button.

A screenshot of a phone

Description automatically generated

Figure 5: home button which reroutes to home URL

A screenshot of a computer

Description automatically generated

Figure 6: double clicking home gives this screen to submit new URL

1. **Search Bar and Button:** Enter URL in the input field and either hit enter or click the button to go to the entered page.

A white rectangular frame with a white border

Description automatically generated

Figure 7: input field and search button

1. **Refresh Button:** Click the refresh button to reload the HTML content by sending another HTTP request.

A screenshot of a phone

Description automatically generated

Figure 8: Refresh Button

1. **Favorites Buttons:** Click on the favorites button to view all favorites set by the user. You can visit the webpage by clicking on the name of the Favorite.

A white rectangular button with a yellow square

Description automatically generated

Figure 9: Favorites Button

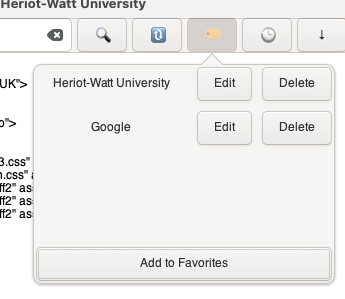


Figure 10: View/Visit favorites

You can also delete an existing favorite by clicking on the delete button next to their name. You can add a new favorite by clicking on the Add to Favorites Button, which will open another dialogue box to set a new favorite (default field values are the current page the user is on)

A screenshot of a computer

Description automatically generated

Figure 11: Set new Favorite

You can also edit the URL and name of existing favorites by clicking on the edit button next to their name.

A screenshot of a computer

Description automatically generated

Figure 12: Edit existing favorite

1. **History Buttons**: Click on the history button to view all the pages the user has visited. (most recent page at the top) (note: refreshing, or reentering the same URL as the one currently on the screen will not update history)

A white clock on a white background

Description automatically generated

Figure 13: History Button

Upon viewing the history, you can either visit the URL by clicking on the title, delete individual pages from the history by clicking on the delete button next to them, or clear the entire history by clicking the Clear History button below.

A screenshot of a computer

Description automatically generated

Figure 14: View, Visit, Delete and clear history

1. **Bulk Download Button:** Click on the download button to get a popover that takes an input file path that you wish to download the contents of. (by default bulk.txt)

A close up of a keyboard

Description automatically generated

Figure 15: Download Button

A screenshot of a computer

Description automatically generated

Figure 16: input field

Upon clicking enter, the contents of the file are downloaded and for each URL, displayed on the main screen as : response code bytes URL.

(currently bulk.txt contains <https://www.google.com>, <https://www.hw.ac.uk>, <https://www.youtube.com>, <https://www.instagram.com> )

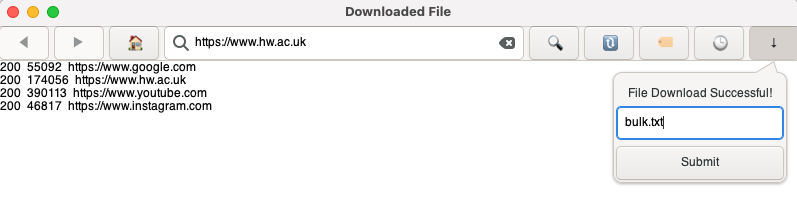


Figure 17: result when submitting bulk.txt for download

# Developers Guide

# Testing

# Conclusions