**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 stores the contents, size, response code and title of the string URL provided and provides access using getter/setter functions.
* Favorites.cs – This class defines a Favorite object which stores the URL and given name of the favorited webpage and provides access using getter/setter functions.
* ReadWrite.cs – This class is designed to handle reading and writing operations from the specified file path for the bulk download facility.
* SetNewHomepage.cs – This class creates a dialogue box and takes the new homepage to be set as input
* AddToFavorites.cs – This class creates a dialogue box and takes the parameters of the favorite to be set (name and URL) as input and adds it to the list
* MainWindow.cs – This class connects all my classes and functionality for all my buttons.
* Program.cs – Main Method

## Data Structures

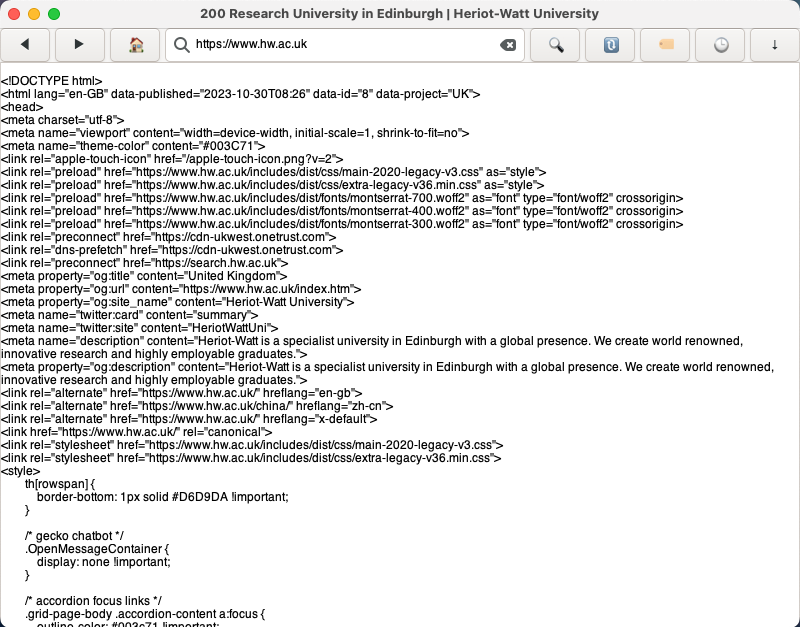
Throughout my coursework I employed different data structures to achieve my functionalities and make my code modular.

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.

# User Guide

# Developers Guide

# Testing

# Conclusions