# Defining and Understanding the Problem

## Problem Statement:

There are many very popular web browsers, such as Chrome, Microsoft Edge and Firefox, currently in the market. Chrome in particular has dominated the market, holding over 60% of the market. Further, the internet has become an integral part of our society with more and more users, it has become an essential product in our society. Each different web browser possesses its own unique functions and features to the benefit of the user, particularly in their quality of life, in addition to the primary purpose of providing the user access to the internet and web applications. I have used many web browsers, such as Chrome, Microsoft Edge, Internet Explorer, Firefox, Opera, Opera GX and TOR browser, and have found many desirable functions among all of them. Therefore, I sought to try and build my own web browser with many of these functions put together, most notably a browser that allows for easier use of multiple tabs or apps concurrently.

The needs of the product would be the primary function of a web browser in rendering and accessing the web pages and applications on the internet. Extra functionality should be added such that the maximised subset of functions are added for the convenience of the user, given the time constraints on the project. The product will need to be compatible for primarily Windows devices of substantial specifications, specifically as I seek to use the web browser it would need to run smoothly, that is without freezing or lagging more than 10 times in a 1 hour period, on a windows device with AMD Ryzen 7 7735HS with Radeon Graphics at 3.20 GHz with 32 GB of RAM and a 64bit operating system without using up more than 8 GB of RAM and 10% of the CPU processing power. The main boundaries of the project include the inability to control the speed of the internet that the user is using and thereby the speed in which pages are loaded cannot be controlled. Furthermore the operating system and specifications of the device that is using the web browser cannot be controlled and therefore the functioning of the software is dependent on the specifications and operating system of the user which is beyond the control of the developer. User inputs are also boundaries that cannot be controlled by the system.

## Design Specification:

### User specifications

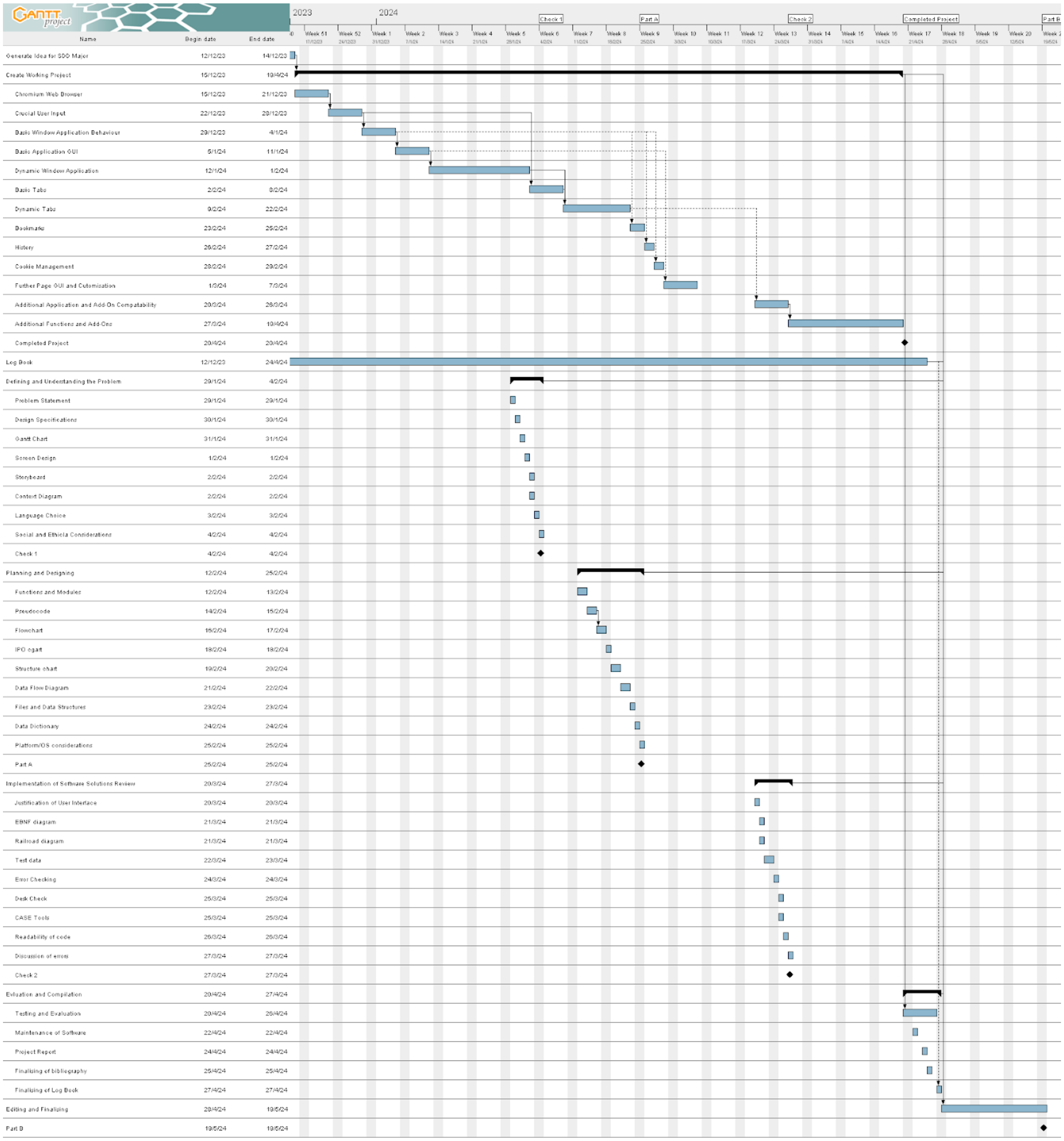
- Ergonomic and intuitive controls that are mostly consistent with existing products. This involves having designs that are consistent with other web browsers to ensure that users moving from one web browser to mine can do so easily with most of the controls being the same.
- Simplistic interface. This entails that the interface is not over complicated for users and maintains a reasonable amount of elements, no more than 10, on the screen at any point in time to ensure that newer users are not bombarded with numerous functions.
- A standardised algorithm for rendering html, css and javascript to ensure that websites are rendered correctly.
- Convenient functionality to help assist with web browsing without interfering with the user.

- Lightweight functions to reduce lag and ensure a smooth experience for the user in the web browser.
- Colour with enough contrast to allow for easy viewing and reading.
- Ensures privacy and security of personal data, especially entered passwords.
- Compatibility with multiple screens of different sizes.
- Appropriate error messages. Particularly for the rendering of websites that have incorrect URLs.
- Appropriate and universal icons to ensure that the user understands it. For instance a lock icon to demonstrate that the site they are using uses an https protocol.
- Data formatting in displays for ease of reading.
- Ensure inclusivity of different people and their different levels of digital literacy. This would involve having simple intuitive functions that are easy to understand.
- Effective and convenient sorting of tabs or web pages that the user can use to assist with their work.

## Developer Specifications

- Developed with an end-user approach as I, the developer, will be the primary user of the web browser.
- Efficient and logical data structures to ensure that there is a reduced lag so that the web browser can smoothly render web pages and sites.
- Infrequent and inconsequential errors or bugs so that the web browser can run smoothly, as many of the bugs will be a result of functionality added to the project, therefore care should be taken to ensure that the functionality is not a detriment to the project.
- Efficient algorithms with minimal threading, CPU usage and memory usage. This is important because web browsers are often not run on their own, as a game might be, often being needed to multitask with other software as a means to search for information..
- Well document progress on the project with modelling of the system. This will ensure that social and ethical issues of plagiarism and copyright can be observed by the details in the progress as well as ensure that deadlines are met.
- Appropriately named variable for ease of understanding. This will assist with other users who seek to use the code and myself as the primary programmer to be able to understand my code upon a second or later reading.
- Appropriate control structures.
- Appropriate data types.
- Quality assurance. This would entail that the program is checked to ensure that prerequisites are met and that the product is of a satisfactory quality.

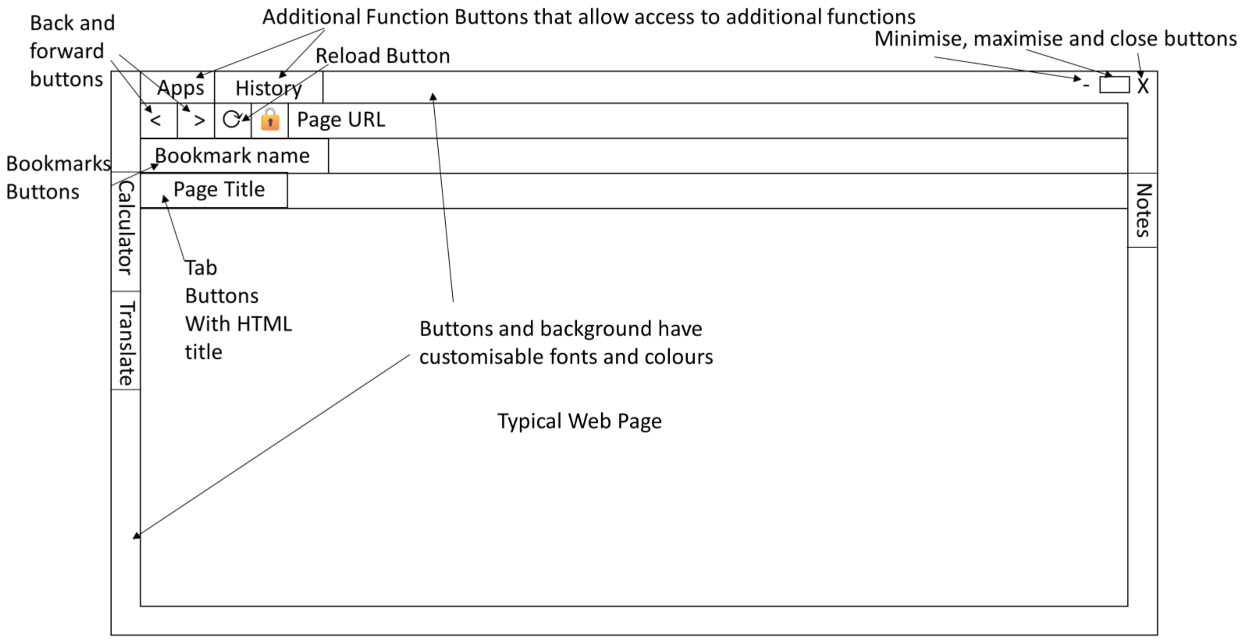
# Gantt Chart:



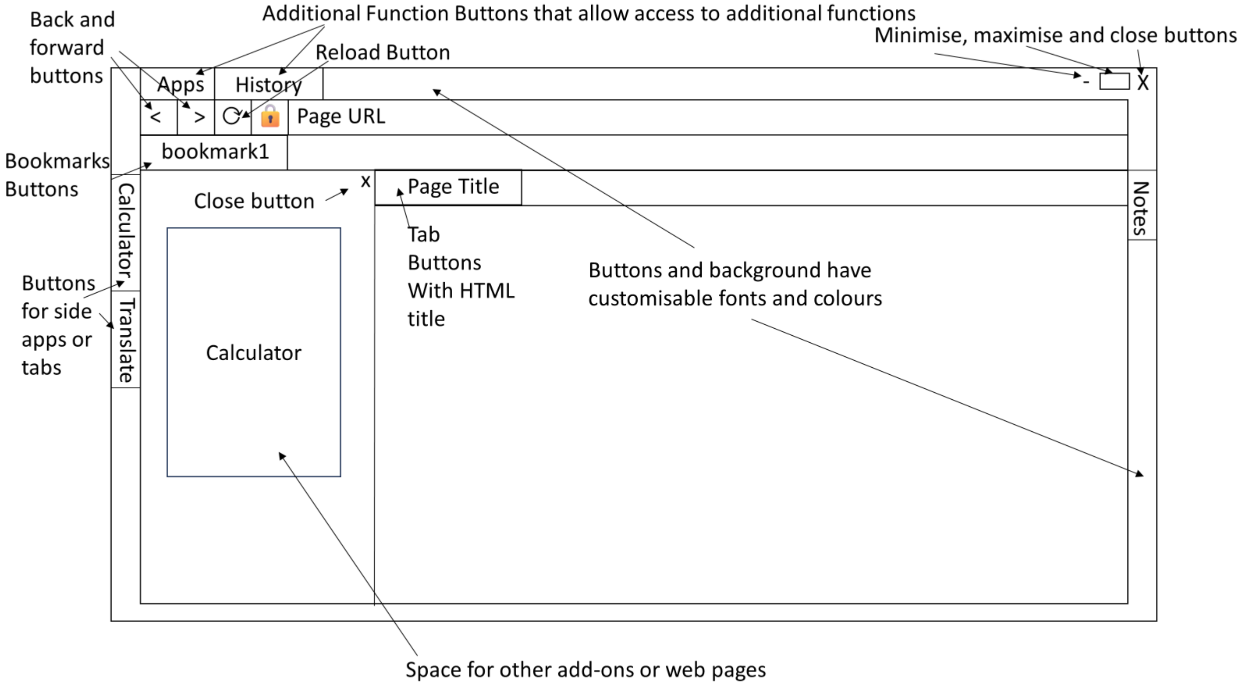
| Name | Begin date | End date |
|---|---|---|
| Generate Idea for SDD Major | 12/12/23 | 14/12/23 |
| Create Working Project | 15/12/23 | 19/4/24 |
| Chromium Web Browser | 15/12/23 | 21/12/23 |
| Crucial User Input | 22/12/23 | 28/12/23 |
| Basic Window Application Behaviour | 29/12/23 | 4/1/24 |
| Basic Application GUI | 5/1/24 | 11/1/24 |
| Dynamic Window Application | 12/1/24 | 1/2/24 |
| Basic Tabs | 2/2/24 | 8/2/24 |
| Dynamic Tabs | 9/2/24 | 22/2/24 |
| Bookmarks | 23/2/24 | 25/2/24 |
| History | 26/2/24 | 27/2/24 |
| Cookie Management | 28/2/24 | 29/2/24 |
| Further Page GUI and Cutomisation | 1/3/24 | 7/3/24 |
| Additional Application and Add-On Compatability | 20/3/24 | 26/3/24 |
| Additional Functions and Add-Ons | 27/3/24 | 19/4/24 |
| Completed Project | 20/4/24 | 20/4/24 |
| Log Book | 12/12/23 | 24/4/24 |
| Defining and Understanding the Problem | 29/1/24 | 4/2/24 |
| Problem Statement | 29/1/24 | 29/1/24 |
| Design Specifications | 30/1/24 | 30/1/24 |
| Gantt Chart | 31/1/24 | 31/1/24 |
| Screen Design | 1/2/24 | 1/2/24 |
| Storyboard | 2/2/24 | 2/2/24 |
| Context Diagram | 2/2/24 | 2/2/24 |
| Language Choice | 3/2/24 | 3/2/24 |
| Social and Ethicla Considerations | 4/2/24 | 4/2/24 |
| Check 1 | 4/2/24 | 4/2/24 |
| Planning and Designing | 12/2/24 | 25/2/24 |
| Functions and Modules | 12/2/24 | 13/2/24 |
| Pseudocode | 14/2/24 | 15/2/24 |
| Flowchart | 16/2/24 | 17/2/24 |
| IPO cgart | 18/2/24 | 18/2/24 |
| Structure chart | 19/2/24 | 20/2/24 |
| Data Flow Diagram | 21/2/24 | 22/2/24 |
| Files and Data Structures | 23/2/24 | 23/2/24 |
| Data Dictionary | 24/2/24 | 24/2/24 |
| Platform/OS considerations | 25/2/24 | 25/2/24 |
| Part A | 25/2/24 | 25/2/24 |
| Implementation of Software Solutions Review | 20/3/24 | 27/3/24 |
| Justification of User Interface | 20/3/24 | 20/3/24 |
| EBNF diagram | 21/3/24 | 21/3/24 |
| Railroad diagram | 21/3/24 | 21/3/24 |
| Test data | 22/3/24 | 23/3/24 |
| Error Checking | 24/3/24 | 24/3/24 |
| Desk Check | 25/3/24 | 25/3/24 |
| CASE Tools | 25/3/24 | 25/3/24 |
| Readability of code | 26/3/24 | 26/3/24 |
| Discussion of errors | 27/3/24 | 27/3/24 |
| Check 2 | 27/3/24 | 27/3/24 |
| Evluation and Compilation | 20/4/24 | 27/4/24 |
| Testing and Evaluation | 20/4/24 | 26/4/24 |
| Maintenance of Software | 22/4/24 | 22/4/24 |
| Project Report | 24/4/24 | 24/4/24 |
| Finalising of bibliography | 25/4/24 | 25/4/24 |
| Finalising of Log Book | 27/4/24 | 27/4/24 |
| Editing and Finalising | 28/4/24 | 19/5/24 |
| Part B | 19/5/24 | 19/5/24 |

# Screen Designs:

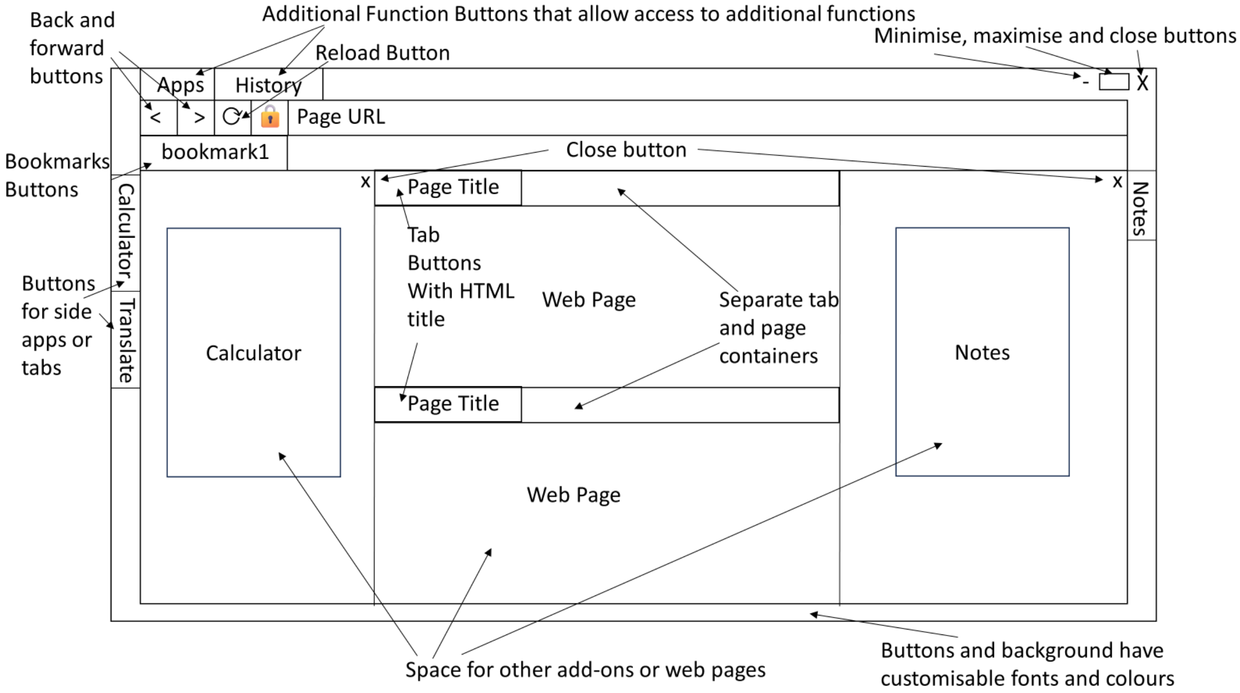
Default View.


Back and forward buttons
Additional Function Buttons that allow access to additional functions
Minimise, maximise and close buttons
Reload Button
Apps
History
- □ X
< >
Page URL
Bookmarks Buttons
Bookmark name
Page Title
Calculator
Translate
Notes
Tab Buttons With HTML title
Buttons and background have customisable fonts and colours
Typical Web Page

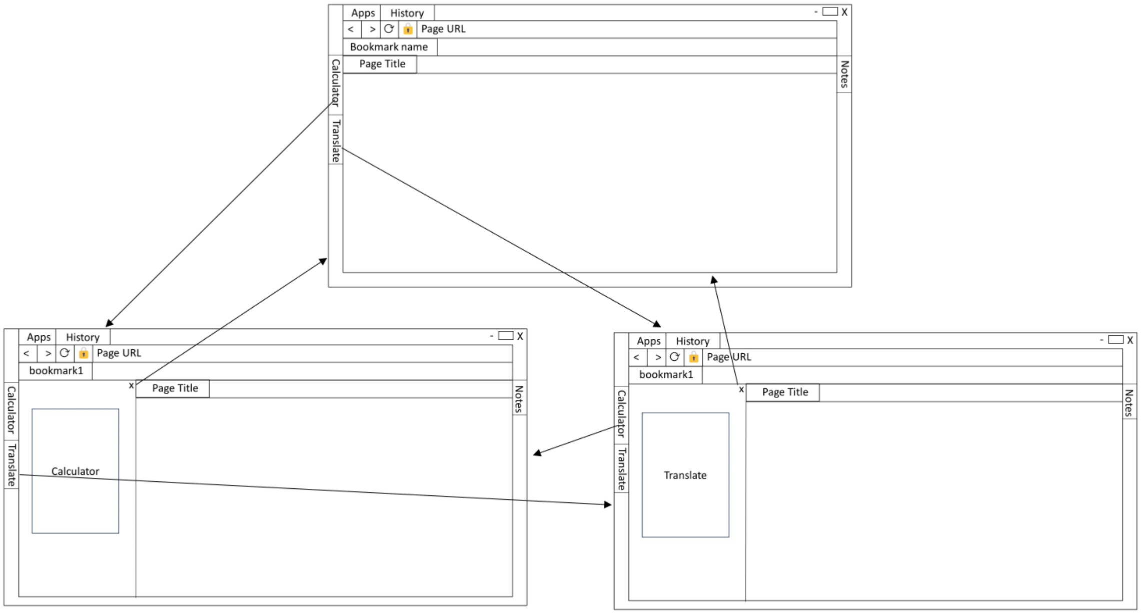
View with single calculator add-on open


Back and forward buttons
Additional Function Buttons that allow access to additional functions
Minimise, maximise and close buttons
Reload Button
Apps
History
- □ X
< >
Page URL
Bookmarks Buttons
bookmark1
Close button
x
Page Title
Calculator
Translate
Notes
Buttons for side apps or tabs
Tab Buttons With HTML title
Buttons and background have customisable fonts and colours
Calculator
Space for other add-ons or web pages

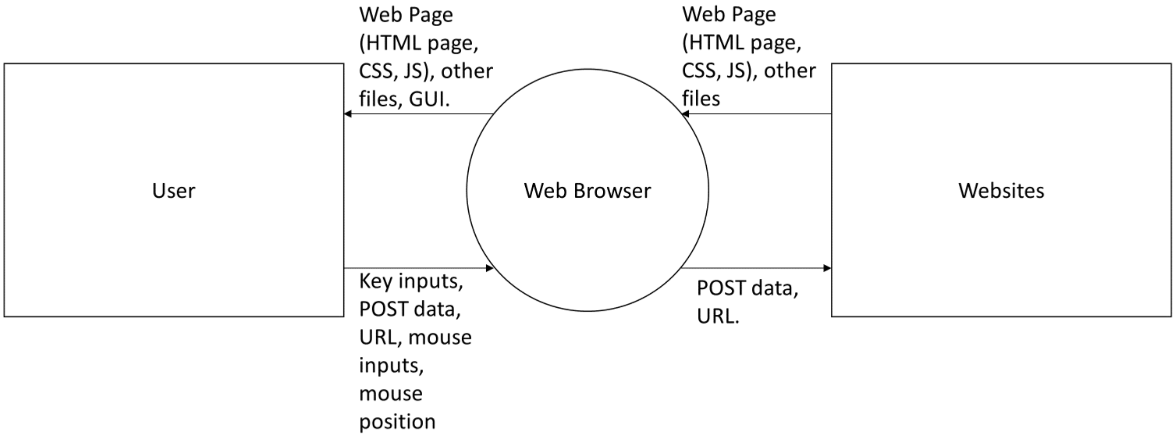
View with add-ons and extra web pages open.


Back and forward buttons
Additional Function Buttons that allow access to additional functions
Minimise, maximise and close buttons
Reload Button
Apps
History
- X
< >
Page URL
Bookmarks Buttons
bookmark1
Close button
x
Page Title
x
Calculator
Translate
Notes
Tab Buttons With HTML title
Web Page
Separate tab and page containers
Buttons for side apps or tabs
Calculator
Notes
Page Title
Web Page
Space for other add-ons or web pages
Buttons and background have customisable fonts and colours

# Storyboard:


Apps
History
- X
< >
Page URL
Bookmark name
Page Title
Calculator
Translate
Notes
Apps
History
Page URL
bookmark1
x
Page Title
Calculator
Translate
Notes
Calculator
Apps
History
Page URL
bookmark1
x
Page Title
Calculator
Translate
Notes
Translate

# Context Diagram:


Web Page (HTML page, CSS, JS), other files, GUI.
Web Page (HTML page, CSS, JS), other files
User
Web Browser
Websites
Key inputs, POST data, URL, mouse inputs, mouse position
POST data, URL.

# Selection of Language:

The chosen language for the project will be C#.

C# is a high-level general purpose programming language developed by Microsoft that runs on the .NET framework to compile the file into bytecode. C# has static typing, strong typing, object oriented and component oriented programming. This allows C# to be effective for this project as it allows compilation into bytecode which is preferable over interpreter programming language as it allows users who do not have an interpreter to run it, hides source code and has reduced overhead as the compilation is done prior to the use of the software. In addition, in built support for window applications in C# exists in Visual Studio reducing the need for an additional front end language and automatically completing the GUI, which is preferable in this situation due to time constraints. Furthermore, C# possesses the necessary packages to access the internet and create a chromium-based browser through CEF Sharp, making C# an effective language for this project. The project will need to be event-driven software with a GUI requirement which can easily be handled by C#'s module, Windows.Form. The programming logic will be primarily driven by the user who will decide which buttons, text input and functions need to be utilised and completed allowing them to have control over the web browser. This causes the program to run numerous threads to run these functions synchronously, although when memory needs to be read and written simultaneously, functions need to wait before executing which may impact the user depending on their hardware and operating system. Compatible computer systems is from the .NET framework translates the C# files into the executable files allowing cross-platform framework for Windows, macOS and Linux.

## Social and Ethical Considerations:

### Intellectual Property & Plagiarism

The project will need to ensure that it does not infringe on the contractual rights of the products, assets and materials that are used in the product. Any images, audio and code used must abide by copyright laws. In this project, images for icons and code for function should be checked to ensure that due credit is given and that licences are upheld. However as this product is not going to be sold, many of the licence restrictions will be met. But even still, the permissions granted from software licences should be checked for all code libraries and algorithms used. Further, software libraries and code snippets should be acknowledged in the code to ensure academic integrity and avoid plagiarism, as well as acknowledgement of used assets within the project such as colour schemes, images and packages.

### Security

Significant security will be required for the handling of private information, particularly personal data of the users in the form of cookies, passwords and other sensitive data that may be entered through or stored by the browser. Therefore encryption for any stored information on the browser will be necessary to ensure the user's security remains protected. Furthermore, any data stored should not be sent across the internet but should be stored on the local device and any data wished to be deleted should be wholly deleted to ensure the security and safety of the user. This is particularly important as often many individuals will access private and confidential information through web browsers, such as emails or other messaging applications, therefore, security is crucial to prevent data breaches and hacks into people’s

### Inclusivity

The product should seek to ensure that a wide range of users can use the product easily. Therefore a simplistic design and intuitive user interface should be used to ensure that it can be accessed by a variety of users. In addition, flagging of potentially unsafe sites should be included to assist users who are not as technologically literate stay safe on the internet. Furthermore, a variety of colour schemes should be offered to allow users to customise the web browser and be better suited for different visual requirements.

### Privacy

Privacy is very important for a web browser to ensure only necessary information is stored. In particular, cookies are an important aspect of web browsing that can grant access to highly personal information and so should be encrypted and stored securely to prevent unauthorised access. Likewise this data should be deleted immediately if the user wishes for it to be deleted.

# Planning and Designing

## Functions

1. getChildControls

Given a control, this function finds all the child controls and grandchild controls, all the way down, to allow for all child controls to be accessed, returning a list of all child controls.

2. checkURL
   Given a string, this function returns whether or not the string follows a format of a valid URL using a series of regular expressions.
3. convertColor
   Given a Windows.UI.Color, this function converts the colour into a System.Drawing.Color which can then be used throughout the form and returns the usable colour class.
4. addTab
   Given a ChromiumWebBrowser, this function initialises a Tab class to contain the browser and other information around the tab and then returns the newly generated tab.
5. checkTabButtons
   Given a tab, this function checks to see if the tab's associated button matches the button that is being checked and returns whether or not the buttons are the same.

## Modules

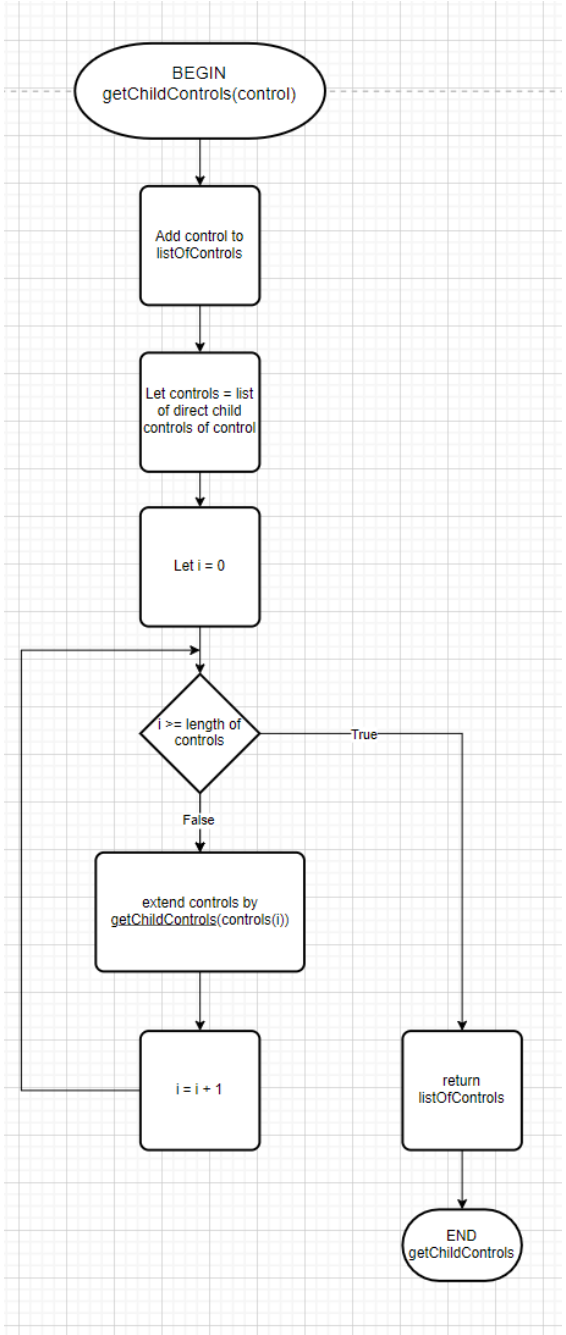
1. CefSharp
   This module contains classes and structures that are used for accessing the internet and websites, handling the reception, delivery and displaying of information through the web. It allows for a way to embed a fully-functional standards-compliant web browser into my C# application.
2. RegularExpressions
   This module allows for the creation and usage of regular expressions which allows the program to recognise certain string patterns which is used to detect whether an inputted URL is valid and so should be redirected to or whether it should be queried through a search engine.
3. Windows.UI.ViewManagement
   This module enables support for the handling and management of certain view settings associated with the active Universal Windows Platform (UWP) app. This allows my application to access the preferences and accessibility settings to better adapt the program to the default settings of the user.
4. System.Windows.Forms
   This module enables the use of windows forms which handle the generation of the GUI for my Web Browser, allowing the creation of controls, text boxes, buttons, events and other classes that enable the application to run smoothly.
5. SDDTabs
   This module contains a class that stores the information about each individual tab, such as the history, button and web browser associated with each tab. This allows for easy storage and mapping for each tab to easily access information necessary for changing tabs.

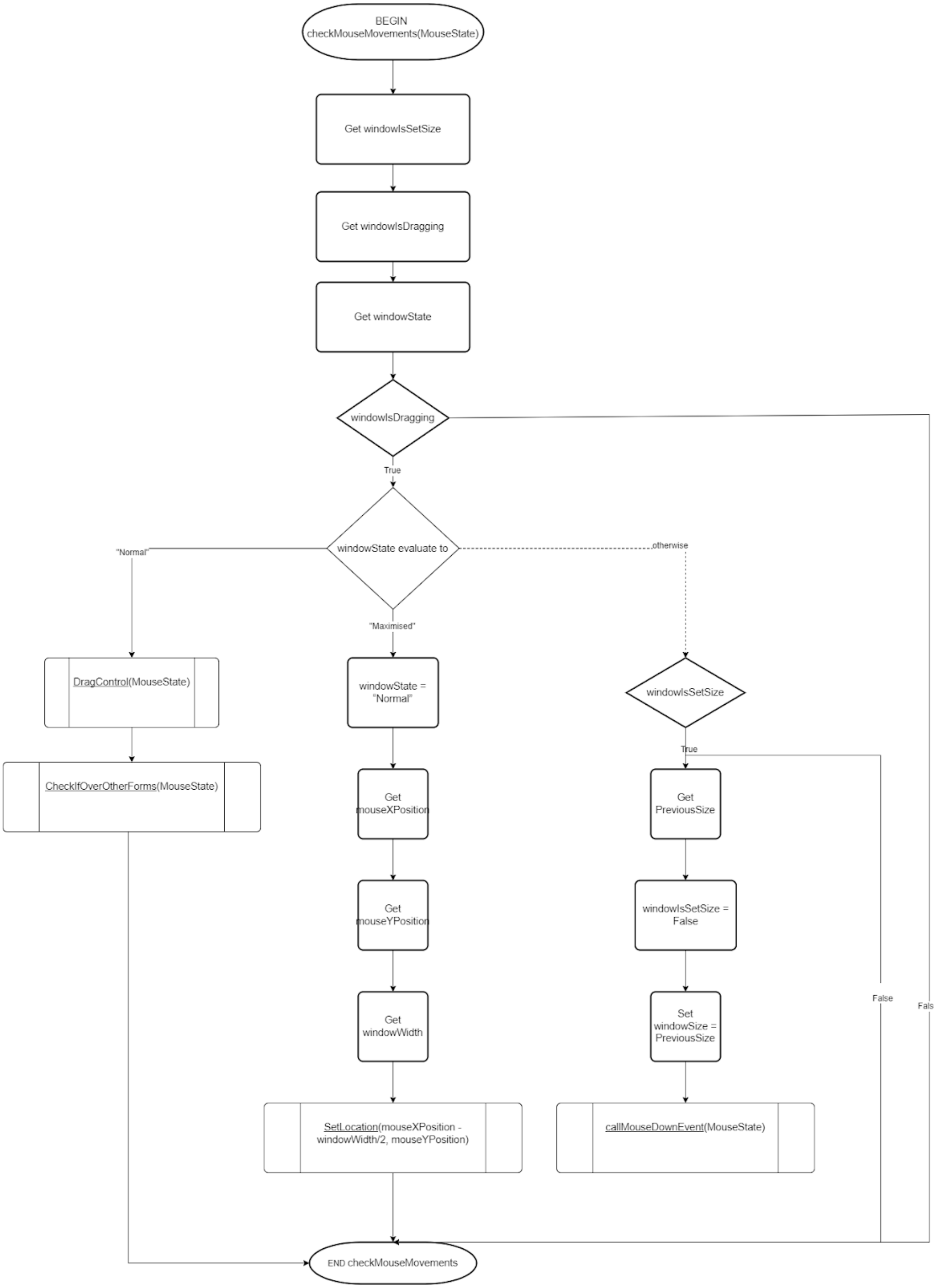
## Pseudocode

```
BEGIN getChildControls(control)
Add control to listOfControls
Let controls = list of direct child controls of control
FOR i=0 TO length of controls
Add getChildControls(controls(i)) to controls
NEXT i
```

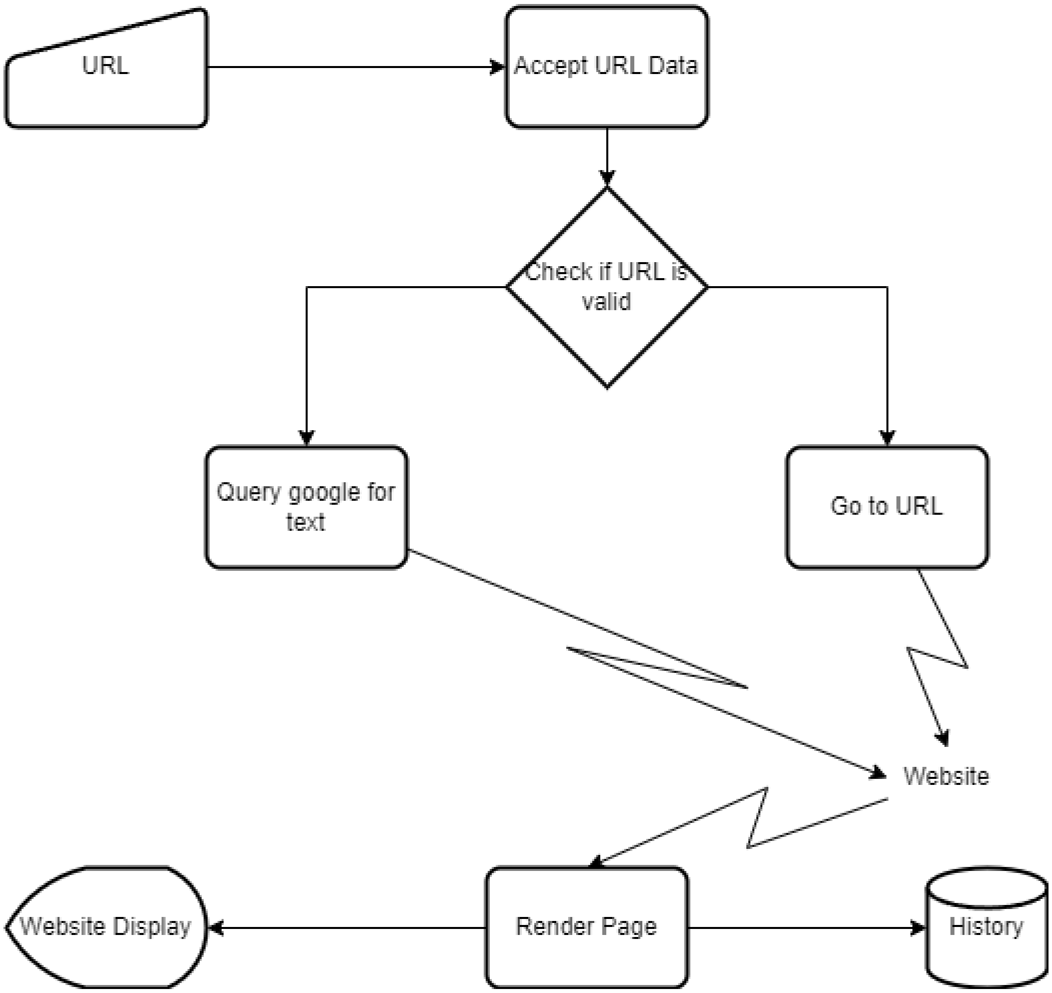
```
Return controls
END getChildControls


BEGIN checkMouseMovements(MouseState)
Get windowIsSetSize
Get windowIsDragging
Get windowState
IF windowIsDragging THEN
    CASEWHERE windowState evaluate to
    “Normal”:
        DragControl(MouseState)
        CheckIfOverOtherForms(MouseState)
    “Maximised”:
        windowState = “Normal”
        Get mouseXPosition
        Get mouseYPosition
        Get windowWidth
        SetLocation(mouseXPosition - windowWidth/2, mouseYPosition)
        OTHERWISE:
        IF windowIsSetSize THEN
            Get PreviousSize
            windowIsSetSize = False
            Set windowSize = PreviousSize
            callMouseDownEvent(MouseState)
    ENDIF
        ENDCASE
ENDIF
```

## Flowchart


BEGIN getChildControls(control)
Add control to listOfControls
Let controls = list of direct child controls of control
Let i = 0
i >= length of controls
True
False
extend controls by getChildControls(controls(i))
i = i + 1
return listOfControls
END getChildControls


BEGIN checkMouseMovements(MouseState)
Get windowIsSetSize
Get windowIsDragging
Get windowState
windowIsDragging
True
windowState evaluate to
"Normal"
"Maximised"
otherwise
DragControl(MouseState)
CheckIfOverOtherForms(MouseState)
windowState = "Normal"
Get mouseXPosition
Get mouseYPosition
Get windowWidth
SetLocation(mouseXPosition - windowWidth/2, mouseYPosition)
windowIsSetSize
True
Get PreviousSize
windowIsSetSize = False
Set windowSize = PreviousSize
callMouseDownEvent(MouseState)
False
False
END checkMouseMovements

## System Flowchart

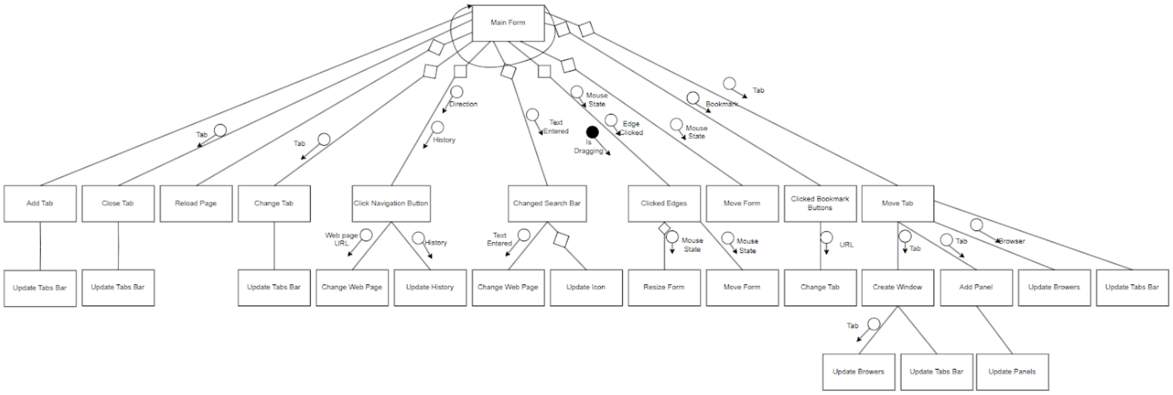


## IPO chart

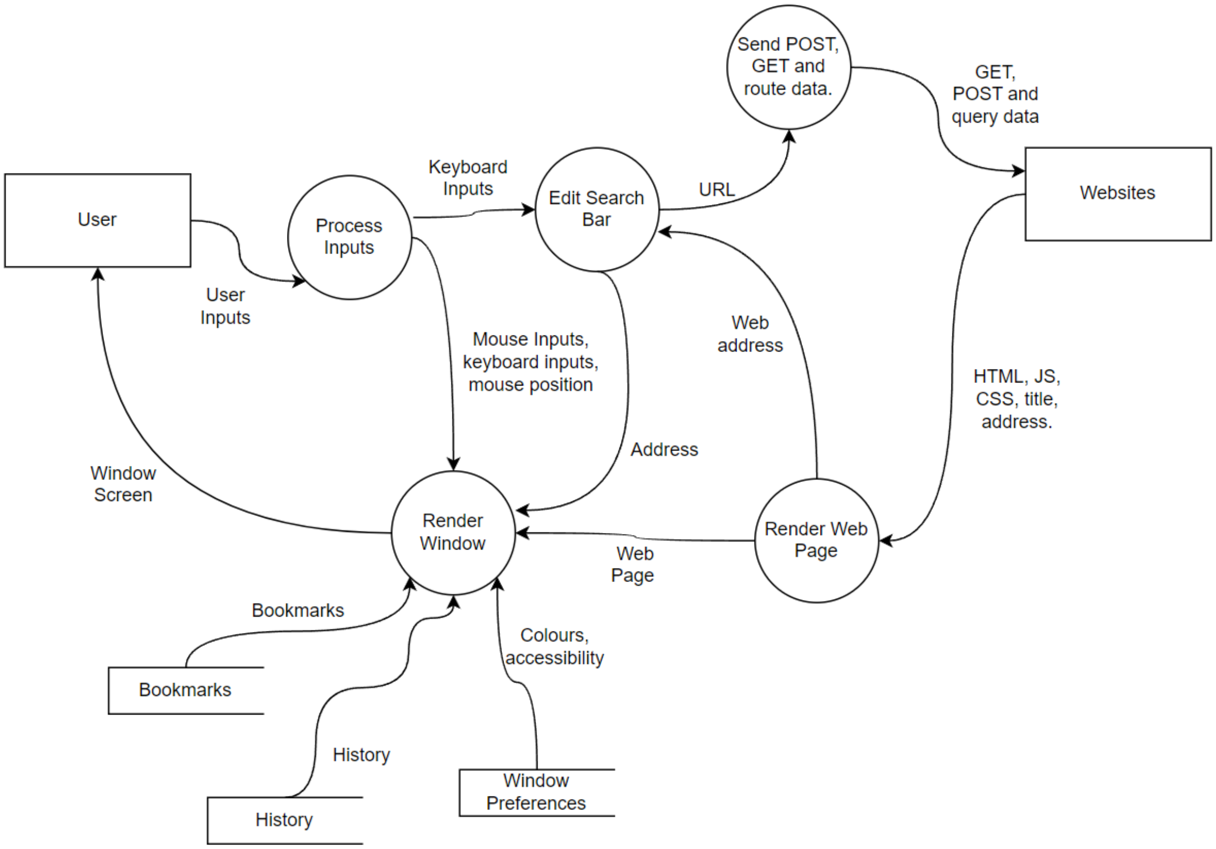
| Input | Process | Output |
|---|---|---|
| Keyboard inputs | Calculate which button is pressed<br>Check to see if the updated url is in the format of a url or a query<br>Check to see if the button is the enter key. | An icon to indicate whether the text will redirect to the website or query the text in a search engine.<br>The webpage of the inputted url or query. |

| | | |
|---|---|---|
| | Updates the web browser accordingly.<br>Web browser updates GUI. | |
| Mouse position, Mouse button pressed | Checks to see if the mouse has left clicked on a button<br>Changes web page and apps on the screens<br>Checks to see if the mouse has changed position.<br>Changes form position and size.<br>Checks to see if the mouse has left clicked on an edge<br>Changes the form position<br>Checks to see if the mouse has left clicked on the form.<br>Changes form position and size<br>Moves or changes the form respectively.<br>Checks to see if the mouse has left clicked on the tab.<br>Create a new window and tab.<br>Checks to see if tab is over another window<br>Adds tab to window. | New app GUI images and controls<br>New rescaled GUI<br>New window with tabs.<br>GUI of application that tab is over with new tabs included |

# Structure chart



# Data Flow Diagram



# Files and Data Structures

## Files

| **File Name** | **File Type** | **Data Structure** | **Purpose** |
|---|---|---|---|
| Bookmarks | .JSON | Array of Records | Stores all the bookmarks and the data required for them in a similar format to that of which other browsers have it so that there is cross platform compatibility, which other browsers possess.<br>The JSON format allows the data to be stored under each record, organising the different properties of each bookmark. |
| Bookmarks | .bak | Array of Records | Stores a backup for the bookmarks in case of corruption or deletion so that the user is far less likely to lose their data. The .bak file type is essentially a text file but labelled to show it is a backup. |
| CefSharp | .dll | Sequential Files | Helps to modularize the code by storing small programs that can be used by the program to reduce memory usage.<br>This file stores all the functions that are a part of the CefSharp library, mainly involving the rendering and handling of webpages. |
| Icon | .png | Sequential Files | Contains a bit-mapped image file for the icon of my web browser.<br>The .png or portable network graphic format is a raster image file type that uses lossless compression, maintaining the information of the image and since the image is loaded on the hardware, it is better than using lossy compression. |
| SDDBrowser | .exe | Sequential Files | Executable files store the binary code necessary for Windows Operating |

| | | | |
|---|---|---|---|
| | | | System to run the application and this file stores the binary for the entire program. |
| domain_names | .csv | Array | Contains a list of all domains for url detection in text format. Comma-separated values (csv) is a simple file type that allows the data to be saved in a table format, allowing modifications to be made easily. |
| icudtl | .dat | Sequential Files | This is a data file type, containing binary information, this one pertains to the unicode encoding. It contains pre-compiled data that is used by the web-browser to perform text-related operations such as the rendering of text characters. |
| | .pdf | Sequential Files | Contains input for opening a pdf file that can be rendered by the web browser.<br>Portable document format is effective for files that do not need to be modified but printed and shared. The chromium web browser and most other web browsers are capable of rendering pdfs and images. |
| | .html | Sequential Files | Contains input for opening an HTML file that can be rendered by the web browser. Most web browsers use html to render online files and so can render local html files. Hypertext markup language is a declarative language that is effective for front end design, mostly in websites and web pages. |

## Data Dictionary

| **Data Item** | **Data Type** | **Format** | **Bytes required for storage** | **Size for Display** | **Description** | **Example** | **Validation** |
|---|---|---|---|---|---|---|---|

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| URL | String | | 400 | 40 | Contains the URL of the web page being displayed. | https://www.google.com | Check if the text follows a url format |
| History stack | Array (String) | | 400* number of urls. | 0 | Contains a list of URLs in the history | https://www.google.com<br>https://www.google.com/search?client=opera-gx&q=what+does+type.dll+mean&sourceid=opera&ie=UTF-8&oe=UTF-8 | |
| Forecolour | Integer | NNNNNN | 6 | 6 | Contains the fore colour colour | FFFFFF | |
| Back Colour | Integer | NNNNNN | 6 | 6 | Contains the back colour colour | 000000 | |
| Width | Integer | | 4 | 4 | The width of the form | 1280 | |
| Height | Integer | | 4 | 4 | The height of the form | 720 | |
| Name | String | | 40 | 40 | The name of the form | Main form | |
| ChromiumWeb Browser | Record | | 4000 | 4000000 | The record which stores all the data from the browser | | |
| isDragging | Bool | X | 1 | 0 | Specifies if the mouse has been held down. | True | Check if the mouse has been clicked and if the mouse is released. |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| isSnapped | Bool | X | 1 | 0 | Specifies if the window is currently in fullscreen or half-screen mode | True | Check if the mouse was near an edge when released. |
| Domains | Array (String) | | 20 * number of domains | 20 * number of domains | Contains all the domains for url checking. | | |
| Bookmarks | Array (Record) | | 400 * number of bookmarks | 400 * number of bookmarks | Contains all the bookmarks for the array. | | Checks to see if the checksum of the bookmarks is correct to test for corruption. |
| | | | | | | | |

## Platform/OS considerations

A number of platform and operating system considerations should be taken into account for the product. The web browser is designed to be used on a laptop or computer platform and a window operating system. This will help to reduce the complexity in compilation and different device variables that are accessible. Further, due to constraints of not possessing other platforms to test the system on, being able to work on a windows operating system and laptop platform is the only testable device and still meets the requirements that are specified in the problem statement. The project is built in C# which, using the .net framework, compiles the project into bytecode for the windows operating system, using many libraries that are specifically designed for the windows operating system. Specifically, the web browser is being tested on the Windows 11 Home, version 22H2, but should work on all Windows 11 versions, with some libraries not being available to Windows 10 and other operating systems. The memory and processing power for the application should be relatively low in comparison to the powerful CPUs and expansive memory that modern computers possess, specifically most devices can reach 3.8GHz and 8GB of RAM which should completely exceed any requirements that the project needs. The web browser itself will not require wifi, however accessing web pages through the internet will, that is it can still render html pages and certain file types that are local to the computer. In terms of hard drive size, it should not take up more than 1GB of space which should make it light-weight enough for daily use synchronously with other programs, although many modern devices have at least 256GB of hard drive storage so this web browser should be almost negligible in taking up such space. However, the overhead of the project should still be managed through efficient functions so as to not hinder the multitasking capabilities of the user whilst having the browser active.

# Logbook

| **Date (of entry)** | **Description of Issue/Task**<br>**(What I've done)** | **Problems encountered /What I should have done**<br>**(Reflection)** | **What I need to do**<br>**(Forward planning)** | **Completion sign-off**<br>**(Issue resolved /Task completed)** |
|---|---|---|---|---|
| 15/12 | I created a basic web browser with reload, back, forward and one page. | There was some difficulty in the back and forward buttons, with the check for if the individual was at the end or beginning being delayed. | Next, I plan to allow the window to be more dynamic with resizing and movement like other applications. | |
| 19/12 | I fixed a few bugs with the important packages and threading issue, where some data is being used simultaneously. | Issues of thread-safe calls and the inability to utilise certain modules in packages. | As before, I plan to create a more dynamic window. | |
| 20/12 | I worked on allowing the screen to move and resize. | I ran into issues with resizing, with difficulties in detecting mouse's position near the edge for resizing. | I plan to get resizing working | |
| 22/12 | I tried to get cursors and resizing working, but I was unable to. | I ran into many issues when trying to use the mouse position to determine if it is on the border. | Again, I plan to get resizing working | |
| 26/12 | I managed to find a solution to allow both movement and resizing of the application as well as access window colours. | | Next, I plan to get GUI looking nice. | |
| 3/1 | I resized things and got all controls uniform and positioned. | I had an issue where resizing the window causes it to be smaller | Next, I plan to work on colours and snapping the size of the window | |

| | | | | |
|---|---|---|---|---|
| | | than the minimum size pushes the window. | | |
| 4/1 | I shifted the colours to be consistent with system colour preferences. | | Next, I plan on getting the snapping of the window working and multiple tabs. | |
| 13/1 | Changed the method for detecting valid URLs using the standardised package. | | | |
| 14/1 | Started working on getting tabs to work and began creating a new class to store tab details. | I had issues in getting the html title of webpages to use for tab titles. | | |
| 1/2 | I fixed a few bugs with the movement of the window from full screen | The main issue was that the mouse location was dependent on the window so when the window resizes the mouse position is not where it should be therefore the next mouse-move event must be used | As before, I plan on getting tabs and window snapping working. | |
| 3/2 | I further fixed some issues with the detection of the validity of the url, and movement of the form. I also began work on allowing the app to snap to the edges of the screen. | The main issue is that the windows Uri detection only looks mostly for the http or https protocols and so rejects links that would normally work | | |
| 4/2 | I further worked on getting tabs working, getting the generation of the buttons working. Further I worked on the ability to snap to the sides of the screen and fixing up the URL detection algorithm. | The issue with the buttons, as aforementioned was that the title could only be detected on an event, so a separate class, binding tabs and buttons together needed to be made. After that, it became relatively trivial to get the button generation working. The snapping to the sides had issues | I plan on getting the tabs to fully work in future and to add previews for edge snapping. | |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  | involving the detection of the sides and trying to get a preview working, which I was unable to attain due to the difficulty of painting outside the form. I plan to add this in future by generating new forms. The URL detection was found to have more issues, involving the Uri detection allowing most strings if they had the http or https protocol so I had to resolve back to the original algorithm, essentially using a series of regular expression formats for the URL to follow. |  |  |
| 5/2 | I worked most on the documentation, completing the “Defining and Understanding the Problem” section. I also worked on getting the app to return to its size for snapping to the edge when removed from that position and I got the project up onto GitHub. |  |  |  |
| 6/2 | I managed to get basic tab functions working, mainly opening, closing and changing tabs. | Some minor issues were encountered such as difficulties in ensuring data transfer between the tab class and the form class. | Next, I will begin making the tabs more dynamic and able to be displayed concurrently. |  |
| 12/2 | I worked further on the documentation, completing the modules and functions sections as well as cleaning up some of the code. |  |  |  |

| | | | | |
|---|---|---|---|---|
| 13/2 | I continued working on the documentation, completing the pseudocode and flowchart. | | | |
| 16/2 | I began working on making dynamic tabs. | I encountered numerous issues, due to the difficulties of events through multiple forms and many odd interactions. | I plan to continue working on the dynamic tabs and removing many glitches involved with them. | |
| 17/2 | I continued working on making dynamic tabs, but I have encountered many issues. | I have had many difficulties fixing numerous issues with the tabs. Mainly, that the new window has issues upon loading. Some of these issues include: the mouse slipping off the form and then failing to move it as a result, attempting to repeat the movement of tabs has caused numerous glitches although, I believe, that most of these are caused by the previous issues. As a result of these issues, the creation of dynamic tabs is likely to need to be extended past original projections. | | |
| 17/2 | I continued working on dynamic tabs and have successfully allowed the tabs to leave the main window to create their own. | The main issue currently is that by closing the original window all other windows close. | I plan on creating functionality to allow the tabs to return to other windows. | |
| 19/2 | I worked on the structure chart. | | | |
| 20/2 | I worked on the Data Flow Diagram. | | | |
| 24/2 | I wrote up the rest of the theory work on operating systems. | | | |

| | | | | |
|---|---|---|---|---|
| 2/3 | I added a system flowchart. | | | |
| 7/3 | I worked on updating the colour scheme when settings are updated. | | As before I plan to add the ability for tabs to join other windows. | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |