0.1 Class design

In general the high level design of this browser has loosely followed the MVC pattern.

0.1.1 PageContent

The most important class to the underlaying model of this web browser is the **PageContent** class, this class contains references to the **PageHistory**, **History**, **Favourites**, and **BrowserResponse** classes. It abstracts away alot of the async behaviour by using an Event called **ContextChanged** which is triggered any time a HTTP get request returns a new **BrowserResponse** instance, this means the GUI elements can subscribe to this event and repaint the relevant elements when it gets triggered. This class also exposes all the navigation functionality to the GUI and the information such as the HTML code, web page title and status code returned via get requests.

0.1.2 BrowserResponse & HttpRequests

HTTP GET requests are handled by the *static Get(string url)* method of the **HttpRequests** class, this method returns an instance of the **BrowserResponse** class which is instantiated asynchronously and contains properties including the page title, HTML source code, URL, and status code.

0.1.3 History & Favourites

There is a significant degree of overlap in the behaviour of the **History** and **Favourites** classes so the common functionality has been implemented in a class called **EntryRecord**. The main distinction between them is the behaviour when there is a duplicated title in their respective lists, this is realised by overriding the *KeyExists()* method. **History** handles name collisions for new history entries by appending an integer to the end of the name,

Favourites on the other hand simply forces the user to enter a new unique custom name if they wish to favourite a page with a name already in its collection.

The **EntryRecord** class handles all the list operations

0.1.4 PageHistory

PageHistory is a session based history navigation class, it only exposes methods for moving back and forwards through the history for a given session, and adding new history items to its own list. It also has properties that provide information about the current node being pointed to in the list. This class is independent of the **History** class in terms of data, but the nodes representing a single page in the history do inherit from the same abstract class called **Entry**. It was a deliberate design descision to seperate the **PageHistory** and **History** classes, this behaviour is reflected in several prominent web browsers, for example Google Chrome and Mozilla Firefox.

0.2 Data structures

0.3 Gui Design

0.4 Advanced language constucts