



# **National University Of Computer and Emerging Sciences**



## **FAST Bug CSS Editor for Chrome Browser**

Umair Habib Peracha  
Pawail Qaisar

**Supervisor**  
Dr. Fakhar-ul-Islam Lodhi

B.S. Computer Science  
Final Year Project: December 2010

**Department of Computer Science  
FAST-NU, Lahore, Pakistan**

# Intellectual Property Right Declaration

This is to declare that the work under the title FASTBug  
carried out in partial fulfillment of the requirements of:

BS FYP     MS Research Survey     MS Thesis     Course Project

is the sole property of the National University of Computer and Emerging Science, and is protected under the intellectual property right laws and conventions. It can only be considered / used for purposes like extension for further enhancement, product development, adoption for commercial/organizational usage, etc., with the permission of the University.

This above statement applies to all students and faculty.

Date: 31<sup>st</sup> December, 2010

Student 1

Name: Umair Habib Peracha

Signature: \_\_\_\_\_

Student 2

Name: Pawail Qaisar

Signature: \_\_\_\_\_

Supervisor (Faculty)

Name: Dr. Fakhar-ul-Islam Lodhi

Signature: \_\_\_\_\_

# Anti-Plagiarism Declaration

This is to declare that the above publication produced under the title:

**FAST Bug**

---

is the sole contribution of the author(s) and no part hereof has been reproduced on **as it is** basis (cut and paste) which can be considered as **Plagiarism**. All referenced parts have been used to argue the idea and have been cited properly. I/We will be responsible and liable for any consequence if violation of this declaration is determined.

Date: 31<sup>st</sup> December, 2010

Student 1

Name: Umair Habib Peracha

Signature: \_\_\_\_\_

Student 2

Name: Pawail Qaisar

Signature: \_\_\_\_\_

## Table of Contents

1. Introduction.....	1
2. Goals and objectives.....	3
3. Scope of the project.....	4
4. Development Lifecycle and Project Summary.....	5
5. Functional Requirements.....	6
6. Non-Functional Requirements.....	7
7. Actors.....	8
8. Use-Cases.....	9
9. Graphical User Interfaces.....	14
10. Requirements Traceability Matrix.....	19
11. Assumptions.....	20
12. System Architecture.....	21
13. Class Design.....	23
14. Implementation.....	24
15. Conclusion.....	29
16. References.....	30
17. Glossary.....	31
18. Appendix - 1 .....	32

## **Abstract**

FAST Bug is a Google Chrome Extension for changing the look and feel of a web page and creating custom style sheets. It allows users to identify a piece of content on the page, hit a key to bring up a simple widget that tweaks the look of that content such as font, background color, foreground color, font-style, etc. It also allows users to take the computed style of any element from a web page. This enables a way to interactively and incrementally build up a CSS style sheet that matches user's needs. A user can adjust the look and feel of any page to enhance usability. The changed styles are stored and are applied every time the page is loaded.

## 1. Introduction

The report describes, in detail, the implementation of FAST Bug – a CSS Editor within Chrome Browser. The report deliberates upon the project, its requirements, its design and implementation.

The world of computing is moving from simple stand alone devices to an array of connected devices with people working on parts of the same project and collaborating on the same documents from different parts of the world. This brought a great need of a software system that would allow people to create and publish websites from within a web browser. There are a lot of tools available in the market such as Adobe ® Dreamweaver<sup>[1]</sup> and Microsoft® FrontPage<sup>[2]</sup> that allow users to create a website in WYSIWYG tools but they all follow the philosophy of browser within an HTML editor.

All such software applications contain a browser within the editor to test the page being developed. This is not sufficient enough for the kind of market we have today. In a world of collaboration and invention the need of the time is to embed an WYSIWYG tool within a web browser so that people from different parts of the world can see, share, and work on same website from different places while it is online and accessible by general public. This extension for Google Chrome will reverse that philosophy by introducing an editor within a browser which would allow even the most inexperienced users to make changes to their sites.

In addition to collaboration and communication, the internet has also changed the entire computing arena with growing popularity and lower power consumption of NetBooks and the demand of small laptops with Internet access. Today, a Personal Computing machine is not suppose to contain the entire application infect most of the computation and processing is transferred to the Cloud<sup>[3]</sup> and the client system has been made light. With the advent of Chromium OS, the need of light weight, portable, and robust applications that can run in a Browser Environment is on the high. This software runs in Google Chrome web browser and is portable, and lightweight.

While Content Management Systems such as Wordpress<sup>[4]</sup> and Drupal<sup>[5]</sup> enable people to work on the content of same site while the site is online; they offer very little to work on design of a site. This software would allow people to design and develop websites online from within a browser and publish it directly from browser to their respective webhosts. The website can be worked on simultaneously by people from different parts of the world. The website will be online during the development process.

Moreover, CSS, from its inception, was intended as a way to allow users to have the final say over how they want to see content on the web. But because creating user style sheets generally requires knowledge of CSS language, end users have not always been able to easily use this functionality. There was a need for a tool that could finally empower end user and give them full control of CSS to decide on how web content is displayed.

Cascading Style Sheets (CSS) are a part of every web designer's vocabulary when styling websites. Web designers, often impressed with designs found on the web, invest a lot of time copying the design or part of design for their own projects. A tool was needed that could give web developers a new and unique tool that would allow them to find any style on the web and adapt it for their own projects.

FAST Bug (named after Foundation for Advancement of Science and Technology) aims to simplify customizing the web, both for casual web surfers and web developers, by making it more accessible and adaptable. It puts the users in control of the web's presentation, allowing them to quickly change the appearance of any page.

FAST Bug is great tool for the design enthusiasts as well as for end users with specific needs and wants for their browsing experience. It allows users to select various elements on a web page and adjust how they are displayed in Chrome. For example, the extension makes web pages with small fonts more accessible by allowing users to increase the font size, and it can make browsing the web less commercial by removing ads. The web designers can easily and quickly copy the styles of any web page and use it in their own projects.

There are somewhat similar solutions in the market which cover part of the functionality provided by FAST Bug. Firebug<sup>[6]</sup> for chrome is a product that lets users debug and maintain CSS and JavaScript of their sites. It is a complex piece of software that is designed primarily for web developers. StyleBot<sup>[7]</sup>, on the other hand, is another such tools available for Google Chrome web browser as an extension. It lets user configure the design of a site but does not allow users to use the designs on the page that already exists. Lastly, there are a number of Bookmarklets<sup>[8]</sup> available on the internet but they do not allow the functionality of loading customized designs on page load.

This document is organized into several sections. Section 1 presents an introduction. Section 2 discusses the overall goals and objectives of the project while section 3 limits the scope for this implementation. Section 4 illustrates activities carried out during the project lifecycle. Section 5 and 6 are composed of functional and non functional requirements, respectively. Section 7 briefly defines the actors or users of this software. Section 8 goes through use cases while section 9 covers the User Interfaces. Section 10 shows the tractability matrix to conveniently trace UI's and their respective use cases and functional requirements. Section 11 covers assumption whereas section 12 is composed of the architecture of the software. Section 13 shows the complete Class diagram and section 14 defines implementation of the software with the help of sequence diagrams. Finally section 15 concludes the project report along with references, glossary, and appendix in section 16, 17, and 18 respectively.

## 2. Goals and objectives

FAST Bug is a Google Chrome extension that would enable people to generate websites and personalize web content. The application's main goals are:

- An editor for writing XHTML code for website content markup.
- Drag and Drop web page development feature.
- Web Page styling using CSS code or tools bar.
- CSS and XHTML validation
- Code highlighting for easy coding
- FTP capability to synch the work done on any web host (without a need of any web server based software)

The above mentioned business requirement are too broad and time consuming. Therefore, the scope of the project has been redefined to include only Web page styling using CSS code and tools.

### 3. Scope of the project

The scope of FAST Bug has been reduced to include only Web page styling using CSS code and tools. Current implementation is a lightweight extension for Google Chrome Web Browser which allows users to modify look and feel of a web page. The user who knows nothing about CSS can change the way websites look, e.g. changing background color and font size for easy readability, improving accessibility by hiding ads, and so on.

FAST Bug's UI allows users to tweak things such as font size, colors etc. with simple mouse clicks. The tool will also enable the advance users to save the settings as CSS rule to be used at any other place. It can also be used by web designers to incrementally build up CSS for their web pages. The tool allows users to extract actual style of any element or the entire web page.

Below are the features of FAST Bug:

- Select any part of a web page using mouse.
- Change Appearance of a web page (or some part of the page).
- Generate CSS of styles of any visible element on a web page.
- Extract CSS of the original styles on page.
- Change Styles conveniently using buttons and drop down list boxes.
- Edit CSS using Textbox (for advance users)

## **4. Development Lifecycle and Project Summary**

Initially the project was to be developed by making additions to Chromium code in Visual C++. Later, it was realized that making changes in Google Chrome code is difficult as well as time consuming. Moreover, launching a version of Chromium browser would render the application useless. Therefore, it was decided to develop the software as an extension to Chromium Web Browser.

The project was carried out in 2 iterations. The first iteration took place during the summers; the second iteration took place throughout the Fall semester. Below is the activity breakdown of each iteration:

### **4.1 Iteration 1 – Summer Activity**

1. Project Finalizing and Planning
2. Requirement Engineering
  - a. Analysis of current solutions
  - b. Defining Requirements
  - c. Documentation
3. Research & Development
  - a. Setting Platform Requirements and Building / Compiling Chromium code
  - b. Chromium Code and Architecture Review
4. Finalizing Iteration 1

### **4.2 Iteration 2 – Fall Activity**

1. Requirement Engineering
  - a. Limiting scope
  - b. Redefining requirements
2. Learning Chrome Extension Development
3. Designing and Architecture
4. Coding / Implementing
5. Finalizing Code and Testing
6. Reporting

## **5. Functional Requirements**

### **5.1 Select Element**

1. The system shall provide controls to select any element on web page.
  - a. The control shall compose of toggle button that shall enable / disable selection
2. The system shall highlight the hovered elements on web page.
3. The system shall un-highlight the element upon selecting the element.
4. The system shall map Active CSS (Computed Style) of the selected element on appropriate controls (appendix 1).
  - a. For definition of Active CSS see glossary

### **5.2 Change Styles**

1. The system shall allow user to change style of selected element using the controls (see appendix 1).
2. The system shall reflect the changes on web page.

### **5.3 Store and Apply Styles**

1. The system shall store the changed styles.
2. The system shall load stored styles when pages loads
3. System shall allow the user to undo all the changes made by him and restore original styles of a web page.

### **5.4 Grab Styles**

1. The system shall allow users to grab styles of any element of the web page.
2. The system shall allow users to copy all the styles applied on an element, in form of CSS.

### **5.5 Drag able Interface**

1. The System shall allow user to drag the User Interface on the screen.
2. The System UI shall not cover any part of screen permanently.

## **6. Non-Functional Requirements**

### **6.1 Technology**

The following technology guideline must apply:

1. The system shall be built as an extension in Google Chrome Web Browser
2. The system shall be coded using HTML, CSS, and JavaScript.

### **6.2 Coding Style**

The following coding guide lines must apply

1. JavaScript code shall use Object Literal notations for Singleton classes
2. Object Oriented coding shall be done using prototype.

## **7. Actors**

### **7.1 Web Page Developers / Designers**

Developers are the creators of web site. They use CSS to style their pages. Developers can take styles of any web page using this tool.

### **7.2 Web Surfers**

These people are the normal users of Google Chrome web browsers who use Chrome to surf the web. They open different sites for day to day tasks. Normal users can use this tool to change styles of a web page according to their taste and needs.

## **8. Use-Cases**

### **8.1 Select Element**

#### **8.1.1 Brief Description**

The user selects an Element on the page. It uses the toggle button for selection on / off.

#### **8.1.2 Pre-condition(s)**

The FAST Bug extension is on and visible on page.

#### **8.1.3 Main Flow**

1. The User clicks Select button
2. The System shows highlighted element on the web page.
3. The User hovers over the page and selects desired element.
4. The System highlights the selected element and populates controls based on the styles of selected element.

#### **8.1.4 Alternative Flows**

1. The User clicks Select button again to deselect Element.
2. The System un-highlights the selected element if any element is selected on page.

#### **8.1.5 Post-condition(s).**

The Element is selected and the controls on the extension are populated.

## **8.2 Change Style**

### **8.2.1 Brief Description**

The user changes a style of selected element.

### **8.2.2 Pre-condition(s)**

An element is selected on the web page

### **8.2.3 Main Flow**

1. The User selects a style using any of the controls on FAST Bug.
  - a. The control are the same as defined in Appendix 1
2. The System changes style on page.

### **8.2.4 Alternative Flows**

None

### **8.2.5 Post-condition(s).**

The style is changed on the page and stored in local storage.

## **8.3 Grab Style**

### **8.3.1 Brief Description**

The User grabs the style of selected element.

### **8.3.2 Pre-condition(s)**

An element is selected on page

### **8.3.3 Main Flow**

1. The User clicks Advance Tab on UI.
2. The System shows computed styles of element in form of CSS.
3. The User copies the CSS to clipboard.

### **8.3.4 Alternative Flows**

None

### **8.3.5 Post-condition(s).**

The CSS is copied to clipboard.

## **8.4 Move UI**

### **8.4.1 Brief Description**

The User changes position of UI.

### **8.4.2 Pre-condition(s)**

The FAST Bug extension is on and visible on page.

### **8.4.3 Main Flow**

1. The User clicks and holds down mouse on UI.
2. The User moves mouse on the page.
3. The System drags the UI on the page.

### **8.4.4 Alternative Flows**

None

### **8.4.5 Post-condition(s).**

The UI is dragged and repositioned on screen.

## **8.5 CSS for Entire Page**

### **8.5.1 Brief Description**

The User Selects CSS for entire page.

### **8.5.2 Pre-condition(s)**

The FAST Bug extension is on and visible on page.

### **8.5.3 Main Flow**

1. The User clicks on Page CSS.
2. The System generates CSS for entire page.
3. The System shows CSS in a Text Box.
4. The User copies the CSS in clipboard.

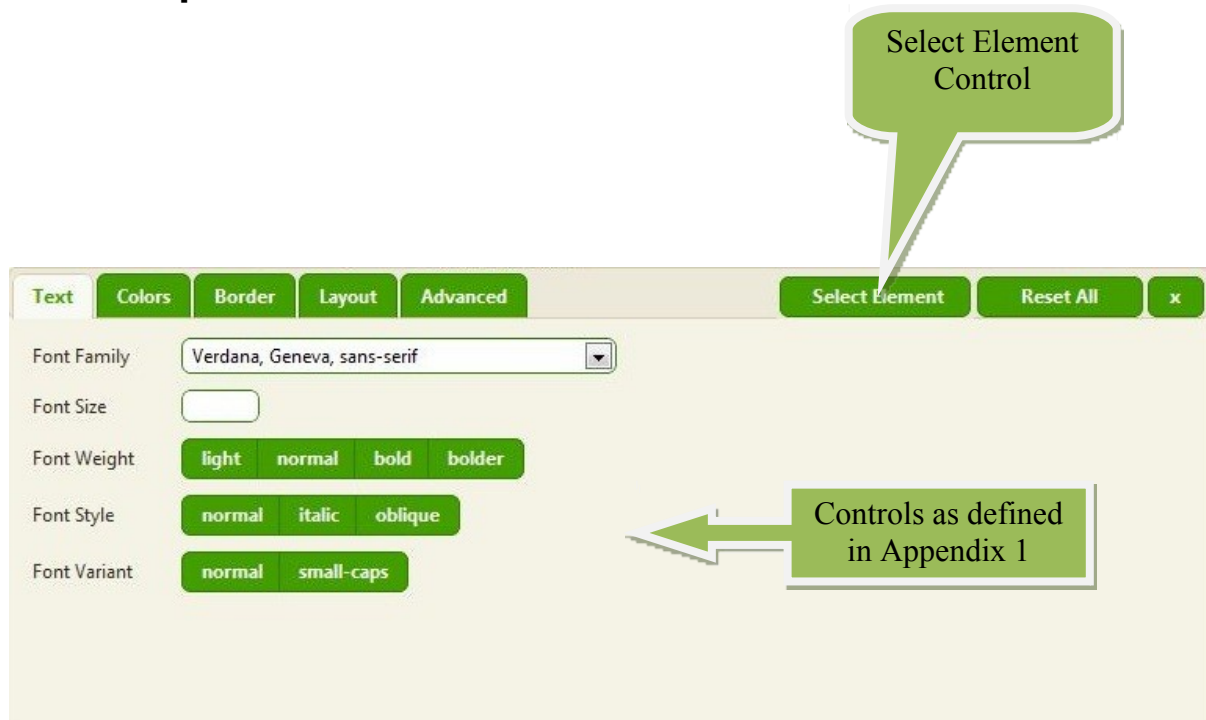
### **8.5.4 Alternative Flows**

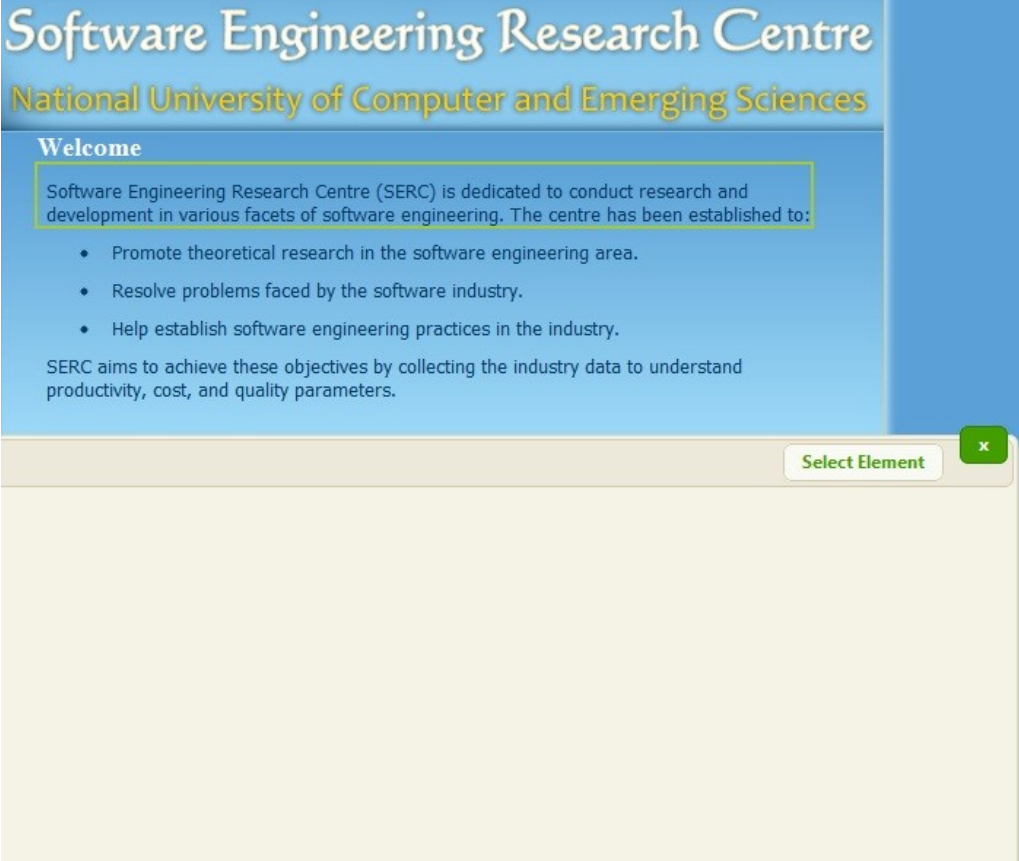
None

### **8.5.5 Post-condition(s).**

The CSS is copied to clipboard.

## 9. Graphical User Interfaces



<b>01: Select Element</b>	
<b>Interface Id.</b>	001
<b>Use case Reference</b>	Use Case 01
<b>Snapshot</b>	
	
<b>Description:</b> The select button is pressed and first paragraph is selected (highlighted)	

02: Change Style	
Interface Id.	002
Use case Reference	Use Case 02
Snapshot	
	
<b>Description:</b> Font size, Font Variant, and Font Style is changed and is reflected on the page.	

<b>03: Grab Style</b>	
<b>Interface Id.</b>	003
<b>Use case Reference</b>	Use Case 03
<b>Snapshot</b>	
<div style="border: 1px solid #ccc; padding: 5px;"><div style="display: flex; justify-content: space-between; align-items: center; border-bottom: 1px solid #ccc;"><div style="display: flex; gap: 5px;"><span style="background-color: #4CAF50; color: white; padding: 2px 5px; border-radius: 3px;">Text</span><span style="background-color: #4CAF50; color: white; padding: 2px 5px; border-radius: 3px;">Colors</span><span style="background-color: #4CAF50; color: white; padding: 2px 5px; border-radius: 3px;">Border</span><span style="background-color: #4CAF50; color: white; padding: 2px 5px; border-radius: 3px;">Layout</span><span style="background-color: #4CAF50; color: white; padding: 2px 5px; border-radius: 3px;">Advanced</span></div><span style="font-size: 0.9em;">Selected Element: <b>#menu ul</b></span></div><div style="padding: 5px; font-family: monospace; font-size: 0.8em; color: #4CAF50;"><pre>#menu ul{ font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 12px; font-weight: normal; font-style: normal; color: rgb(0, 58, 101); text-transform: i 0px; line-height: normal; text-align: left; vertical-align: baseline; direction: ltr; background-color: rgba(0, 0, 0, 0); background-image: none; bac background-attachment: scroll; opacity: 1; width: 220px; height: 144px; top: auto; right: auto; bottom: auto; left: auto; margin-top: 0px; margin 30px; padding-right: 0px; padding-bottom: 0px; padding-left: 0px; border-top-width: 0px; border-right-width: 0px; border-bottom-width: 0px; l right-color: rgb(0, 58, 101); border-bottom-color: rgb(0, 58, 101); border-left-color: rgb(0, 58, 101); border-top-style: none; border-right-style: n fixed; display: block; visibility: visible; z-index: auto; overflow-x: visible; overflow-y: visible; white-space: normal; clip: auto; float: none; clear: nc list-style-type: disc; marker-offset: null;}</pre></div></div>	
<b>Description:</b> The style of the selected element is presented in form of CSS which can be copied by the user and can be used elsewhere.	

<b>04: Drag UI</b>	
<b>Interface Id.</b>	003
<b>Use case Reference</b>	Use Case 04
<b>Snapshot</b>	
<b>Description:</b>	
The UI is dragged upwards and bottom part of the web page is visible	

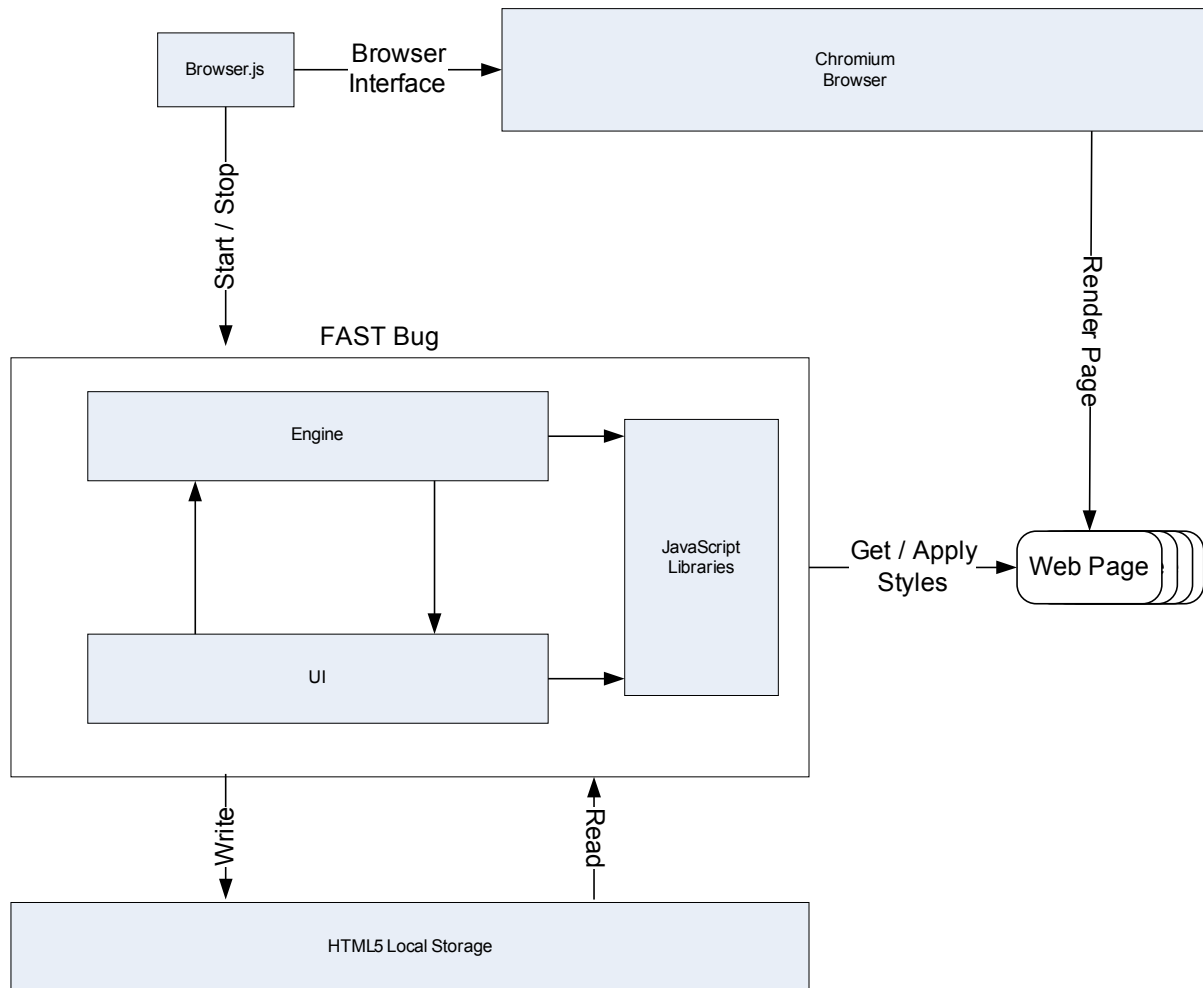
## 10. Requirements Traceability Matrix

Use Case	Actor	Functional Requirements	Non-Functional Requirements	GUI
1	1, 2	1	N/A	1
2	1, 2	2	N/A	2
3	1	3	N/A	3
4	1, 2	4	N/A	4

## **11. Assumptions**

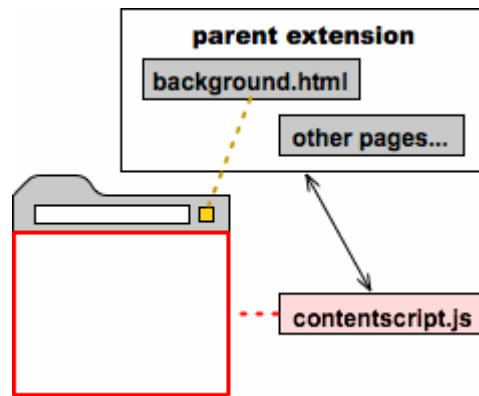
1. HTML 5 Local Storage is enabled.
2. This extension works on Chrome version 6 and above
3. The words Google Chrome and Chromium are used interchangeably in this document.

## 12. System Architecture



FAST Bug comprises of an Engine which is a collection of JS files containing the main logic of the system, UI which is drawn on the page, and JavaScript Libraries including jQuery.js and selectionBox.js. The entire FAST Bug code is injected on the page when it is loaded and the visibility of its UI is controlled using Browser.js file which communicated with the FAST Bug code on the page.

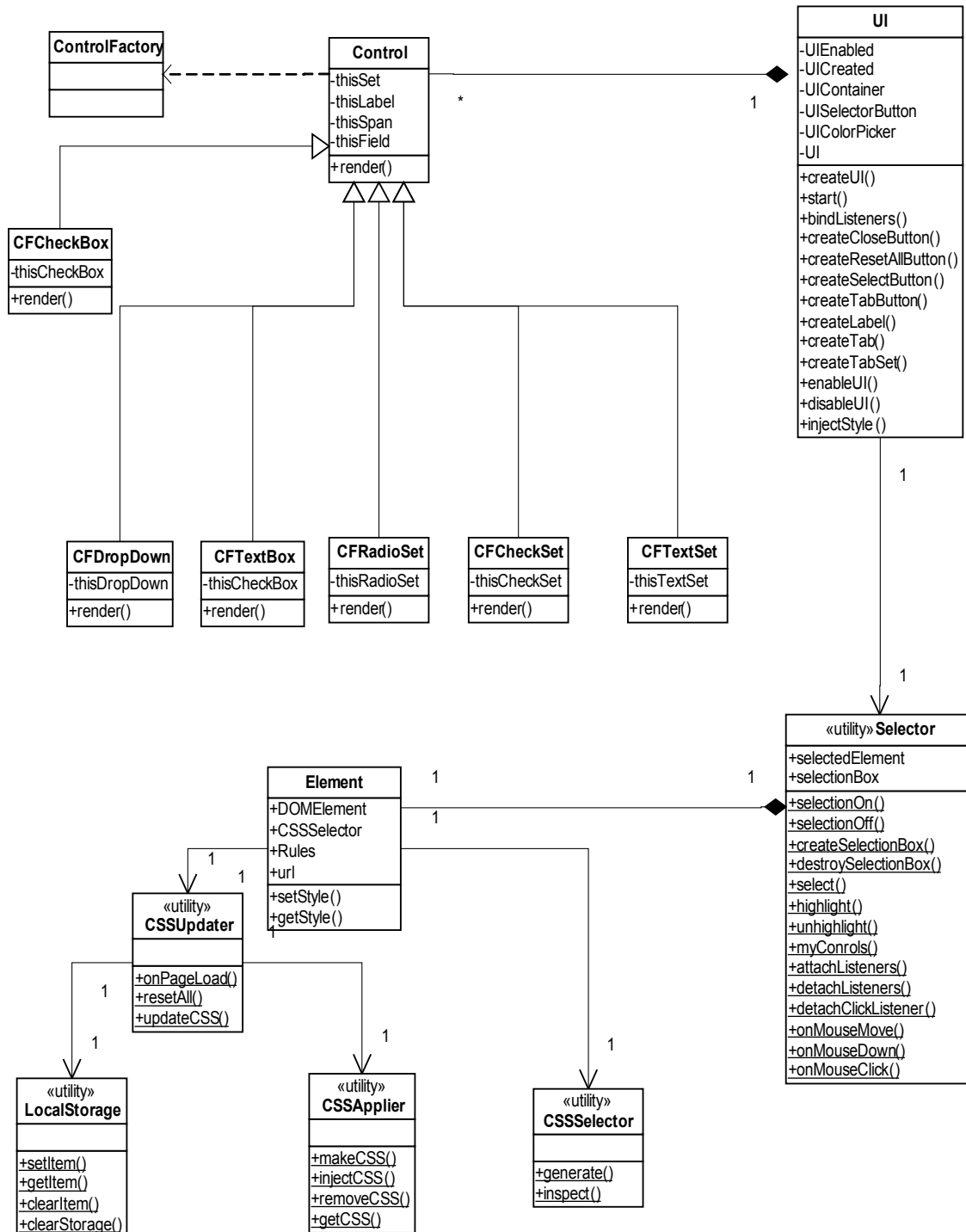
The system follows standard chromium extension architecture. This extension has a background page, background.html, an invisible page that holds the main logic of the extension. It has JavaScript code that controls the behavior of the browser action. When the extension needs to interact with web pages, then it needs a content script. The background.html passes message to FAST Bug Engine and starts it.



A content script is some JavaScript that executes in the context of a page that's been loaded into the browser. The content script becomes a part of that loaded page, not as part of the extension.

Content scripts can read details of the web pages, and they can make changes to the pages. The content script can read and modify the DOM for the displayed web page. The entire FAST Bug engine is injected in the loaded page using content scrip. The FAST Bug UI is created using Content Script.

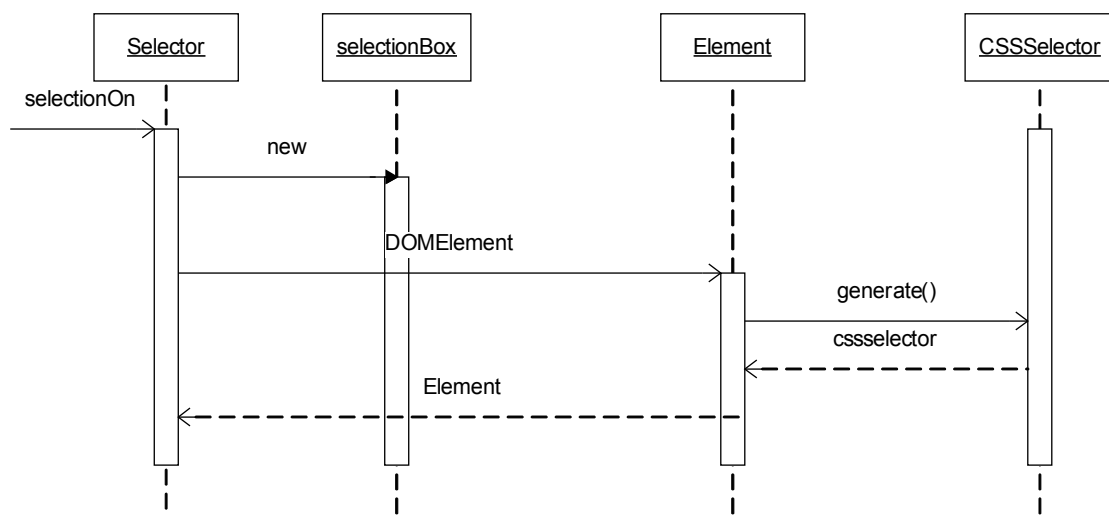
### 13. Class Design



## 14. Implementation

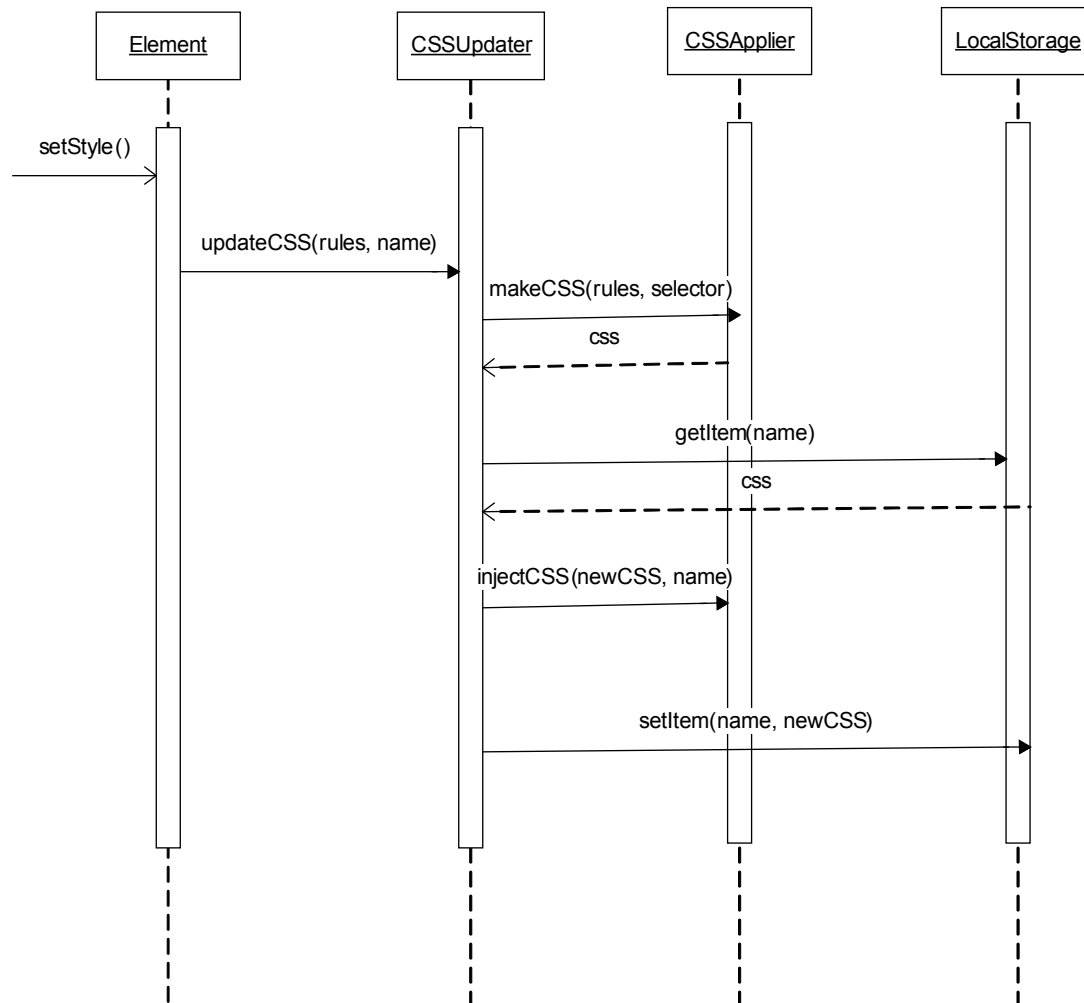
### 14.1 Select Element

The user when click “select” button, a call (selectionOn) is sent to Selector class that generates a selectionBox on page. With the help of Mouse, the user selects any part of page which returns a DOM element to selector. This DOM element is then sent to element class that generates it CSS Selector using CSSSelector.generate() function and returns an Element object to the Selector class.



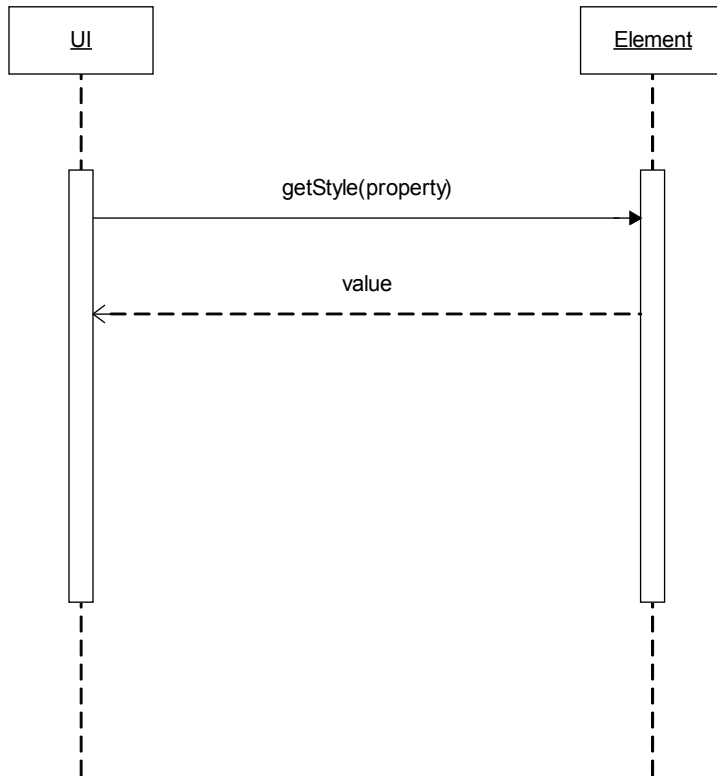
## 14.2 Change Style

The user changes style using any of the UI controls (appendix 1). The UI control calls `setStyle()` function of the selected element. It called `CSSUpdater.updateCSS()` to update CSS on the page, which uses `CSSApplier` to generate new CSS. The `CSSUpdater` also updates the new style in `LocalStorage` class by calling `LocalStorage.setItem()`.



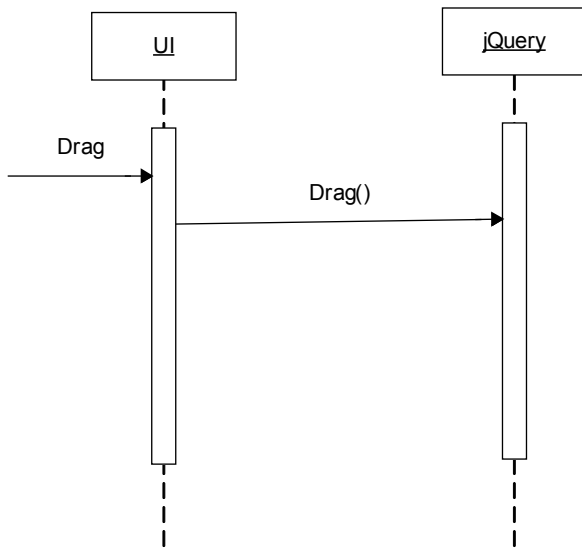
### 14.3 Grab Style

To populate UI controls and generate a CSS for user to copy, the UI class calls `getStyle()` function of the selected element that return the style using `window.getComputedStyle()` function provided by WebKit.



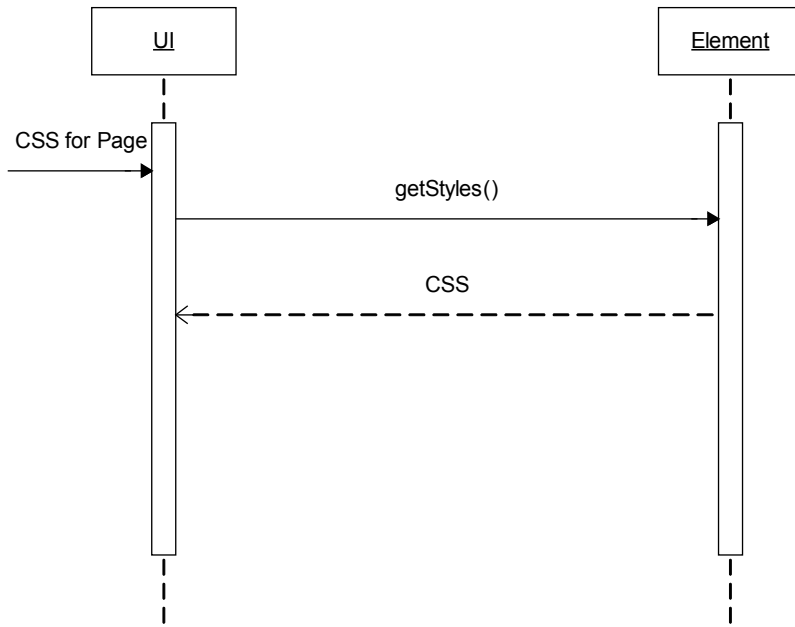
## 14.4 Move UI

The user drags the extension on the page to unhide part of the page covered by the UI. This process is done using jQuery library.



## 14.5 CSS for Entire Page

CSS for entire page is generated by first generating DOM tree of the complete page and then parsing each DOM element into Element object. For all the Element objects `getStyles()` function is called. The function return CSS for the selected element using `window.getComputedStyle()`.



## 15. Conclusion

Initially, FAST Bug was to be developed as an inbuilt feature of Chromium browser. It was suppose to be developed in Visual C++ because Chromium web browser is written in visual C++ using Microsoft Visual Studio 2005. The idea was to develop on the existing code and create a new web browser. The problem with this approach was that in order to use FAST Bug, one has to download and install our implementation of the browser. It was realized that this approach would render our work useless. Furthermore, the chromium code is not complete and changes are being made to the code regularly to add new features and fix bugs. In addition to that, the task at hand was very difficult and given the limited time it was not possible for us to complete the software.

Fortunately, Google Chrome comes with a feature that allows one to develop applications to extend the functionality of the browser. This feature is called Extensions. Extensions are small software programs that can modify and enhance the functionality of the Chrome browser. Extensions are written using web technologies such as HTML, JavaScript, and CSS. Extensions bundle all their files into a single file that the user downloads and installs. The rational for choosing extension over a new browser was to allow every Google Chrome and Chromium user to be able to use FAST Bug.

Current implementation covers only a portion of FAST Bug, namely the styling feature. Another interesting feature that could not be completed functional requirement 5. The only approach to develop this functionality into FAST Bug has complexity  $O(n^2)$  which is not reasonable, therefore this functionality is not yet introduced into FAST Bug.

During the development of FAST Bug, we gained a great insight in the web development tools making. For example, a new functionality was found that could further enhance the product. This functionality would allow people to see the inherited CSS properties of an object with respect to its parent element. This would greatly benefit web developers in designing web applications.

Chrome web browser also gives NPAPI which could be utilized to enhance functionality of the product by giving user ability to select Fonts which are already present in his system, for his site. This would also allow web surfers to make use of system fonts to further personalize the web.

A great level of understanding was developed regarding open source community in general and Chromium project in particular. The project can further be developed in accordance to the high level requirements stated in section 2 of this report. Moreover, the local storage can be better utilized by using JSON data structures. This would enable us to save and use permanent objects in our code.

## 16. References

1. Adobe® Dreamweaver is a world renowned Website development tool.  
<http://www.adobe.com/products/dreamweaver/>
2. Microsoft® FrontPage is a web site development tool by Microsoft.  
<http://office.microsoft.com>
3. Cloud computing is Internet-based computing, whereby shared servers provide resources, software, and data to computers and other devices on demand.  
[http://en.wikipedia.org/wiki/Cloud\\_computing](http://en.wikipedia.org/wiki/Cloud_computing)
4. Wordpress is an open source content Publishing tool. <http://wordpress.org/>
5. Drupal is an open source content management system for web sites. <http://drupal.org/>
6. Firebug is an extension for Mozilla Firefox web browser. It is developed to aid web developers in developing and debugging web sites. <http://getfirebug.com/>
7. Stylebot is an extension for Google Chrome web browser specifically designed to customize web pages for personal use. <http://stylebot.me/>

## **17. Glossary**

### **17.1 Active CSS**

The browser cascades all the style sheets together and form final styles based on rules of CSS hierarchy. These final rules are then applied to an element when the page loads. Active CSS refers to this final style that is made by the browser and what is shown to the user on screen.

### **17.2 WYSIWYG**

WYSIWYG stands for What You See Is What You Get. These are the tools that allow people to develop web content by drag and drop.

## **18. Appendix - 1**

Controls:

1. HTML Textboxes for value input such as pixels, color code
2. HTML Dropdown menus for Font styles
3. Toggle Buttons for properties that require enable / disable such as font-weight etc
4. Open source Color Picker library for color values.