ANGLIA RUSKIN UNIVERSITY

Little Bits of Changes:

Comparing traditional web applications with applications built using the AJAX paradigm in the context of a module choice milieu within an academic community.

Dominic Myers

A Project in partial fulfilment of the requirements of Anglia Ruskin University for the degree of Master of Science in Computer Science

Submitted: May 2006

Acknowledgements

The help and support of my family and friends was without doubt the primary factor associated with the completion of this work. My sons Oz, Ash and Izaac were able to put up with my ramblings with patience and only the smallest amount of condescension, my parents helped with support and encouragement. Friends including Jacky Pettit, Amanda Donaghey, Jackie Barwise, Barnabas Baggs, Phil Done, Agneta Budd, Graham Haigh and Lynne Hester all helped in their individual ways and I make no comment on such assistance by the placement of their names.

My supervisor Dr. Jo Stanley showed enthusiasm and a keen appreciation for what I have been trying to do and I can't thank her enough.

Paul Canning, the Web Development Officer for Cambridge City Council helped immeasurably by directing me to alternative perceptions of what I thought was a whizz-bang idea and introduced me to the Accessify forum. The participants at Kiwi's forum also helped, if not always directly, then at least with a much needed break.

Katrina Winter was able to utilise remarkable skills in the application of active listening and it was some time before I realised that she wasn't listening at all, she was also a fabulous study partner and all round good-egg... I don't think I'd be here without her.

Table of Contents

Tables

Appendices

ANGLIA RUSKIN UNIVERSITY

ABSTRACT

Faculty of Science and Technology

MASTER OF SCIENCE

Little Bits of Changes:

Comparing traditional web applications with applications built using the AJAX paradigm in the context of a module choice milieu within an academic community.

By Dominic Myers

May 2006

An academic course is often difficult enough but anxieties can be compounded by students having to struggle against laborious administrative procedures associated with the time-tabling of the elements of the course. This work will examine approaches to such scheduling tasks using an incremental approach starting with a number of traditional approaches using an online two-page model and concluding with the development of a web-based application which relies upon actions being initiated by small portions of user interaction and causing little bits of changes. Such solutions are applicable to a number of settings but have heretofore primarily resided in the commercial domain.

The technology associated is extensively examined and leads to an appreciation of the increasing dynamism of the interaction available. Where users were once presented with static data they are now able to impact the page displayed almost immediately; leading Michael Mahemoff to talk about "little bits of changes" in his recent (May 2005) podcast: *The AJAX Web Architecture* (available at http://softwareas.com/ajax-podcast/).

A process of progressive development was utilised during the development of the application and this is fully explored along with the platforms upon which the units of work took place, testing was by necessity continual but was supplemented by limited user testing and critique upon completion of the application. A common criticism of the Ajax paradigm is that it makes no allowances for users of assistive technology and this was borne out; but along with this was a growing appreciation of the depth of the main technology available, namely JavaScript, and how it might be used in conjunction with traditional methods to allow for greater enrichment of user experience while acknowledging the breadth of users.

I. Introduction

A given academic institution[1] offers a choice of educational programs[2] which are made up of distinct units of education[3]. In some cases students are required to have attained a certain level of proficiency on commencement of the unit; otherwise the unit would be beyond their capabilities. An educational credit is gained on completion of each unit and overall success in the program depends upon the attainment of a certain number of credits. Only a given number of units can be undertaken in a given division of time[4].

Traditionally the contents of units are outlined in a prospectus and it is the responsibility for the student to choose their program using a form and with reference to that course catalogue.

Bill Gates, when being interviewed about public and private bodies embrace of the Internet, was quoted by Ian Burrell in The Independent[5] as saying, "In education, every country is really just at the start".

The *Literature Review* will show how academic institutions were at the forefront in the development of technologies which could make choosing an

---

1. In this instance the institution is Anglia Ruskin University in Cambridge, UK. It must be noted that the given modules are not necessarily current
2. A Bachelor of Science or Arts degree for instance. A BSc will form the focus of this work
3. In this context these units of education are called modules
4. In this context these are semesters
5. Burrell I (2005) Where do we go from here? The Independent Media Weekly supplement March 20th 2006 pp.4-7

academic program a much more user friendly experience. Such technologies have not been implemented by these institutions in practical ways though. It has been left to commercial concerns to enhance user experience.

One such company, Google, caused a re-evaluation of Asynchronous JavaScript and XML[6] techniques when they applied them in an innovative way in the form of Google Maps. The application of this to the business of module choice will be examined after more traditional avenues have been explored. This process will be documented in the *Design and Methodology* section.

A number of perspectives will be used in the *Testing and Analysis* section of this work, as will a critique of the testing processes associated with JavaScript.

The *Conclusion* will summarise the problems encountered during testing and explore possible solutions.

6. Henceforth to be known as Ajax

II. Literature Review

1.0 Introduction

No discussion of the provenance of Ajax techniques would be complete without some discussion of the Internet and World Wide Web and their development. The history of Internet browsers and associated technologies will also be examined as will the development of JavaScript. This will lead to an examination of the techniques that have come to be known as Ajax.

2.0 The Internet

JCR Licklider[7] is credited with being the “Father of the Internet”. Licklider discussed the concept of the “Galactic Network” in 1962 in a series of memos in which he envisioned a globally interconnected set of computers through which users might access data and programs from any machine[8].

Licklider was made the Head of the Information Processing Techniques Office (IPTO) at the United States Department of Advanced Research Projects Agency[9] in 1962. There he promoted the formation of an all encompassing computer network, when he left in 1964 he encouraged his successors to continue this funding strategy. ARPA had been created by President Eisenhower in 1958 as part of an Air Force Appropriations Bill, it had the remit of pursuing the research and development of technologies which might be of benefit to the military forces. A number of commentators have suggested that that its creation was a direct response to the launching of the Sputnik[10] satellite by the USSR.

---

7. “J.C.R. Licklider may well be one of the most influential people in the history of computer science. As Director of the Information Processing Techniques Office (IPTO), a division of the Pentagon's Advanced Research Projects Agency (ARPA), Licklider from 1963-64 put in place the funding priorities which would lead to the Internet, and the invention of the "mouse," "windows" and "hypertext." Together these elements comprise the foundation of our networked society, and it owes much of its existence to the man who held the purse-strings, and also created a management culture where graduate students were left to run a multi-million dollar research project.” Bennahum D (Unknown) J.C.R. Licklider (1915-1990) [WWW] http://memex.org/licklider.html
8. Leiner B & Cerf V *et al* (2003) A Brief History of the Internet Version 3.32 [WWW] http://www.isoc.org/internet/history/brief.shtml
9. Henceforth to be known as ARPA, In 1972 it was renamed as the Defense Advanced Research Projects Agency (DARPA)
10. “History changed on October 4, 1957, when the Soviet Union successfully launched Sputnik I. The world's first artificial satellite was about the size of a basketball, weighed only 183

Eisenhower later issued warnings about the rising economic, political and spiritual power of what he coined the "military-industrial complex". In one speech he gave before leaving office he listed technological issues as being one of the gravest threats facing the USA, on the one hand he warned about scholars being manipulated by the financial power of the "military-industrial complex", whereas he was also conscious of "the equal and opposite danger that public policy could itself become the captive of a scientific-technological elite"[11].

While Licklider provided the concept of the Internet three teams were working concurrently on the technologies which made it possible.

Leonard Kleinrock[12] worked towards establishing a mathematical theory of packet switching for his PhD in 1961[13], it was finally published in 1962 and was released as book in 1964[14]. At about the same time (1962-1965) the RAND

---

pounds, and took about 98 minutes to orbit the Earth on its elliptical path. That launch ushered in new political, military, technological, and scientific developments. While the Sputnik launch was a single event, it marked the start of the space age and the U.S.-U.S.S.R space race." Garber S (2003) Sputnik and The Dawn of the Space Age [WWW] http://www.hq.nasa.gov/office/pao/History/sputnik/

11. Eisenhower D Farewell Radio and Television Address to the American People IN Public Papers of the Presidents (1960) Washington: GPO Also: [WWW] http://millercenter.virginia.edu/scripps/diglibrary/prezspeeches/eisenhower/dde_1961_0116.html
12. "Leonard Kleinrock (born 1934) is a computer scientist, and a professor of computer science at UCLA, who made several extremely important contributions to the field of computer networking, in particular to the theoretical side of computer networking. He also played an important role in the development of the ARPANET at UCLA." Wikipedia (2005) Leonard Kleinrock [WWW] http://en.wikipedia.org/wiki/Leonard_Kleinrock
13. Kleinrock L (1961) Information Flow in Large Communication Nets PhD. Thesis Proposal, Massachusetts Institute of Technology
14. Kleinrock L (1964) Communication Nets: Stochastic Message Flow and Design Columbus: McGraw-Hill

Corporation[15], sponsored by the US Air Force, was funding Paul Baran[16] who came up with the concept of robust communications networks through a distributed network rather than the centralised or decentralised network models popular at the time. Primarily he was concerned with “a communication network which will allow several hundred major communications stations to talk with one another after an enemy attack”[17], alongside his work on network architecture he also worked on packet switching techniques as a way of utilising the network. Meanwhile (1964-1967) Donald Davies[18] of the United Kingdoms National Physical Laboratory was also working on packet switching.

Kleinrock's theories about packet switching were implemented by Lawrence Roberts[19] in 1965 when he connected 2 computers over a telephone

---

15. “The RAND Corporation is an American think tank first formed to offer research and analysis to the U.S. military. The organization has since expanded to working with other governments and commercial organizations. RAND has around 1600 employees based at six sites: Santa Monica (California), Arlington (Virginia), Pittsburgh (Pennsylvania), and in Europe: Leiden (The Netherlands), Berlin (Germany) and Cambridge (United Kingdom). Some consider the corporation's name to be a contraction of the phrase "Research ANd Development". (Gen. Curtis LeMay quipped that RAND meant "Research And No Development".)” Wikipedia (2005) RAND Corporation [WWW] http://en.wikipedia.org/wiki/RAND_Corporation
16. “Paul Baran (born 1926) was one of the developers of packet-switched networks along with Donald Davies and Leonard Kleinrock. He was born in Poland, but his family moved to Boston in 1928. He obtained his Masters degree in Engineering from UCLA in 1959 and began working for the RAND Corporation in the same year.” Wikipedia (2005) Paul Baran [WWW] http://en.wikipedia.org/wiki/Paul_Baran
17. Baran P (1964) Rand Memoranda on Distributed Communication [WWW] http://www.rand.org/publications/RM/RM3420/
18. “Donald Watts Davies CBE FRS (June 7, 1924 – May 28, 2000) was a British computer scientist who was a co-inventor of packet switching (and originator of the term), along with Paul Baran and Leonard Kleinrock in the US.” Wikipedia (2005) Donald Davies [WWW] http://en.wikipedia.org/wiki/Donald_Davies
19. “Larry Roberts is sometimes called the "father of the ARPANET." He earned this nickname by directing the team of engineers that created the ARPANET. Roberts was also the principal architect of the ARAPNET.” Griffin S (2000) Internet Pioneers [WWW] http://www.ibiblio.org/pioneers/index.html

line. In 1966 the then Head of IPTO at ARPA (Robert Taylor) recruited Roberts to link together the computers at ARPA-funded institutions, within a year he put together a plan for ARPANET[20]. When he presented his ideas to conference he met Davies who made him aware of the work of Baran and thus the theoretical work of all three teams were synthesised into a concrete specification. In 1969 the US Department of Defence commissioned the ARPANET for research into the networking of computers[21]. More pragmatic commentators[22] have suggested that ARPANET was originally conceived not as a test-bed for networking technologies but as a means for ARPA-funded bodies to share computer resources and results rather than provide them with their own computers, something Taylor felt was an expensive luxury.

More and more computers were added, and yet more networks were connected to ARPANET, making the first Internet[23]. The ARPANET was decommissioned in 1990 but not before many of the protocols which make up the modern Internet were tested on it. ARPANET was demonstrated to the public in 1972 by Bob Kahn[24] at which time work commenced on a second generation of network protocols based upon the ideas developed by the three

---

20. Roberts L (1967) Multiple computer networks and intercomputer communication Proceedings of the first ACM symposium on Operating System Principles
21. PBS Online (1998) Nerds 2.0.1. Timeline [WWW] http://www.pbs.org/opb/nerds2.0.1/timeline/index.html
22. Griffin S (2000) Internet Pioneers [WWW] http://www.ibiblio.org/pioneers/index.html
23. Or network of networks
24. “Robert E. Kahn, (born December 23, 1938), along with Vinton G. Cerf, invented the TCP/IP protocol, the technology used to transmit information on the modern Internet.” Wikipedia (2005) Bob Kahn http://en.wikipedia.org/wiki/Bob_Kahn

teams detailed above. By 1982 a family of new protocols had been developed. The Transmission Control Protocol – responsible for host communications, and the Internet Protocol – responsible for routing and addressing (more commonly known as TCP/IP) were by far the most prominent members of this family[25]. TCP/IP is operating system agnostic and though originally designed to work upon large computers it was eventually able to network smaller workstations as well[26].

We can see that the impetus for the creation of the Internet came from Licklider, that its architecture was primarily the responsibility of Baran, that the protocols used upon it were developed by Roberts, Davies and Baran and refined by Kahn and Cerf, but the final contributor towards today's Internet and World Wide Web is Berners-Lee.

---

25. Davidson J (1988) An Introduction to TCP/IP New York: Springer-Verlag

26. It is has been suggested that the development of TCP/IP, along with the personal computing revolution of the 80s and 90s, represent a significant step towards the wired world in which we now live. Asleson & Schutta (2005) Foundations of Ajax Berkeley: Apress

3.0 The World Wide Web

Berners-Lee has written that the concept of the Internet pre-dated the work of Licklider and notes that Vannevar Bush wrote of a theoretical machine called the Memex[27] in 1945[28] and that Ted Nelson used Bush's theories when he wrote of Literary Machines in 1965[29]. Nelson's machines, once networked, would allow users to share information as equals using a process he called hypertext[30]. Bush's concept of hypertext was also used as the the basis of a collaborative workspace called On Line System (NLS) developed in the 1960s by Doug Engelbart at Stanford. But their contributions, while visionary, didn't come to fruition in the way that Licklider's did. Berners-Lee is also conscious of the element of timing in the success of his work. He states, "I happened to come along with time, and the right interest and inclination, after hypertext and the Internet had come of age. The task left to me was to marry them together"[31].

Berners-Lee used the idea of hypertext as developed by Nelson and Engelbart to link documents together. In much the same way as an academic

---

27. “He described the device as mechanical desk linked to an extensive archive of microfilms and able to display books, texts or any document from the library, and further able to automatically follow references from any given page to the specific page referenced.” Wikipedia (2005) Hypertext [WWW] http://en.wikipedia.org/wiki/Hypertext
28. Bush V (1945) As We May Think Atlantic Monthly
29. Berners-Lee (1999) Weaving the Web The Past, Present and Future of the World Wide Web by its Inventor London: Orion Business Books
30. "What is hypertext? As the smallest common denominator it can be said that hypertext is text, distributed to a set of discrete sections, with referential links in between." Müller-Prove M (2005) 2 Hypertext [WWW] http://www.mprove.de/diplom/text/2_hypertext.html
31. Berners-Lee, 1999, p7

paper has references to other academic papers Berners-Lee thought that it might be appropriate for references in electronic resources to link to other electronic resources using an addressing scheme. This would allow the reader to directly access the referenced resource by clicking on a link in the document[32].

Berners-Lee wasn't alone in his enthusiasm for hypertext but he was initially alone in his belief in the possibilities of placing hypertext on the Internet. After looking at already available hypertext products he decided that a simplified subset of Standardised General Markup Language (SGML) which he called Hypertext Markup Language (HTML) would, combined with a protocol called Hypertext Transport Protocol[33], provide a mechanism by which scientists working on diverse hardware platforms would be able to share data in a universally accessible format over the infant Internet.

---

32. Initially these links were called URNs (Universal Resource Name), later they were known as URLs (Universal resource Locator), and more recently they are called URIs (Uniform Resource Identifier). More colloquially they are known as web addresses

33. Or HTTP. “HTTP is a request/response protocol between clients and servers. An HTTP client, such as a web browser, typically initiates a request by establishing a TCP connection to a particular port on a remote host (port 80 by default). An HTTP server listening on that port waits for the client to send a request string, such as "GET / HTTP/1.1" (which would request the default page of that web server), followed by an email-like MIME message which has a number of informational header strings that describe aspects of the request, followed by an optional body of arbitrary data. Some headers are optional, while others (such as Host) are required by the HTTP/1.1 protocol. Upon receiving the request, the server sends back a response string, such as "200 OK", and a message of its own, the body of which is perhaps the requested file, an error message, or some other information.” Wikipedia (2005) HyperTextTransfer Protocol [WWW] http://en.wikipedia.org/wiki/Http

3.1 HTML

Charles Goldfarb, the creator of SGML, saw its development as result of the work of a number of individuals active in the late 1960s whose main concern was the presentation of printed material[34]. A number of people proposed splitting the data contained in documents from the formatting of documents; the formatting would then be described in specific ways. Goldfarb took these ideas, and with his colleagues at IBM in 1969, created General Markup Language (GML) in order to allow text-editing, formatting and information retrieval subsystems to share documents in an integrated Law Office information system.

Goldfarb continued to work with markup languages[35] and designed SGML to have: Common data representation allowing different hardware/ software combinations to read and write the same document; flexibility to be able to work with any of the myriad different types of document; rules for the creation of a formal description of documents of the same type.

While Goldfarb started work on SGML almost immediately after his work on GML, it wasn't until 1974 that SGML was properly proven. This was when Goldfarb proved that software could check the validity of a document against its document type definition[36]. SGML was ratified as a standard in 1986, though it

---

34. Goldfarb C (1990) SGML HISTORY [WWW] http://xml.coverpages.org/sgmlhist0.html
35. Goldfarb C & Prescod P (2001) THE XML HANDBOOK 3rd Ed. London: Prentice Hall
36. The Document Type Definition, or DTD, is still in use today. The XHTML DTD is "<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">"

had been in use for some time prior to this in industry.

SGML was also used as the basis for XML[37]. HTML represents a subset of SGML but it shares only the first feature of SGML - in that it allows the same documents to be read on different systems. It is not formally extensible and it does not enforce specific rules. To some extent it might be said that SGML is the father of both HTML and XML but while HTML represents a subset of SGML, XML represents a simplified version of SGML. Tim Bray has said about the development of XML that, “The idea was to take SGML and throw away the 95 percent that never got used and retain the 5 percent that did.”[38].

37. eXtensible Markup Language

38. Gray J (2005) A Conversation with Tim Bray ACM Queue vol. 3, no. 1 [WWW] http://www.acmqueue.com/modules.php?name=Content&pa=showpage&pid=282&page=1

3.2 Browsers

Along with a language and a protocol Berners-Lee also created the WorldWideWeb reader/editor, after all there would have been little point in creating a language without a means of reading it. Interestingly this points out a dichotomy between SGML and HTML and it might be argued that if Goldfarb had not had to await a means of checking a document against its DTD it might have had more success. Though in terms of usability it is unlikely that SGML would have been successful as a base language for the Internet as it has a reputation of being excessively complicated. WorldWideWeb was only able to edit HTML files within the "file:" space rather than the "http:" space as the HTTP PUT method had not been implemented when it was created in 1990, that is to say that it was only possible to edit files on local computers rather than over the Internet[39]. WorldWideWeb was later renamed Nexus in order to save confusion with the abstract information space[40].

In the early 90s a number of people wrote graphical browsers but by far the most successful was Mosaic written by students at NCSA in 1993[41]. Though

---

39. Berners-Lee T (no date) The WorldWideWeb browser [WWW] http://www.w3.org/People/Berners-Lee/WorldWideWeb.html

40. This is a term borrowed from Berners-Lee, it is now typed as World Wide Web (with spaces). The W3 call it a “wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents”. W3 (1993) World Wide Web [WWW] http://www.w3.org/History/19921103-hypertext/hypertext/WWW/TheProject.html

41. Marc Andreessen and Eric Bina of the National Center for Supercomputing Applications at the University of Illinois-Urbana-Champaign. Asleson & Schutta (2005) Foundations of Ajax Berkeley: Apress

initially written for Unix computers it was quickly ported to the Macintosh and then other platforms. People have split the development of browsers into eras[42], Berners-Lee's WorldWideWeb and Mosaic represent the first era and are distinguishable primarily because Mosaic allowed for the display of images.

The second era (1994-1997) saw, though careful marketing[43], the rise of the Netscape for-profit browser. Microsoft also released early versions of Internet Explorer[44], even though it gave it away for free Netscape was dominant.

The third era (1997-1999) witnessed a battle for market share between Netscape and Microsoft. Microsoft won by bundling Internet Explorer with the Windows 95 operating system and though supporting standards better than the Netscape browsers[45]. Internet Explorer was also ported to the Macintosh and was again made available for free. There were other browsers, most notably Opera, that managed to gain some market share as a result Microsoft and Netscape's battle for market dominance.

In the forth era (1999-2003) Microsoft dominated the browser market. The Netscape browser (originally called Mozilla) was increasingly prone to bugs

---

42. Kock P (2004) A history of browsers [WWW] http://www.quirksmode.org/index.html?/browsers/history.html
43. “Its mother company, Netscape Communications Corporation, asked money for the use of its browser but gave it away for free to carefully selected target audiences, like students and teachers. This helped immeasurably to spread the word and to ensure Netscape dominance.” Kock P (2004) A history of browsers [WWW] http://www.quirksmode.org/index.html?/browsers/history.html
44. Based on licensed Mosaic technology
45. Particularly Internet Explorer 5 in March 1999

during the third era and it became clear that it required substantial rewriting if it were to become a contender to Internet Explorer so Netscape opened the source of the browser to the public in 1998, the Mozilla Foundation was created in 2003 in order “to ensure that the Mozilla project continues to exist beyond the participation of individual volunteers, to enable contributions of intellectual property and funds and to provide a vehicle for limiting legal exposure while participating in open-source software projects”[46].

The fifth era (2003-Present) has seen the Mozilla Foundation having some measure of success with its reworked Netscape browser and the market share of Firefox is gradually on the rise. Microsoft haven't upgraded their browser for some time and other developers have had a chance to not only catch up with Microsoft but, arguably, to surpass it[47]. Apple have used the Konquerer[48] code base to create their own browser called Safari and Opera has made the transition to being free rather than for-profit. We're left with four browsers which all share common abilities and facilities, particularly in regards to their handling of technologies such as the Document Object Model[49], Cascading style Sheets[50] and JavaScript.

---

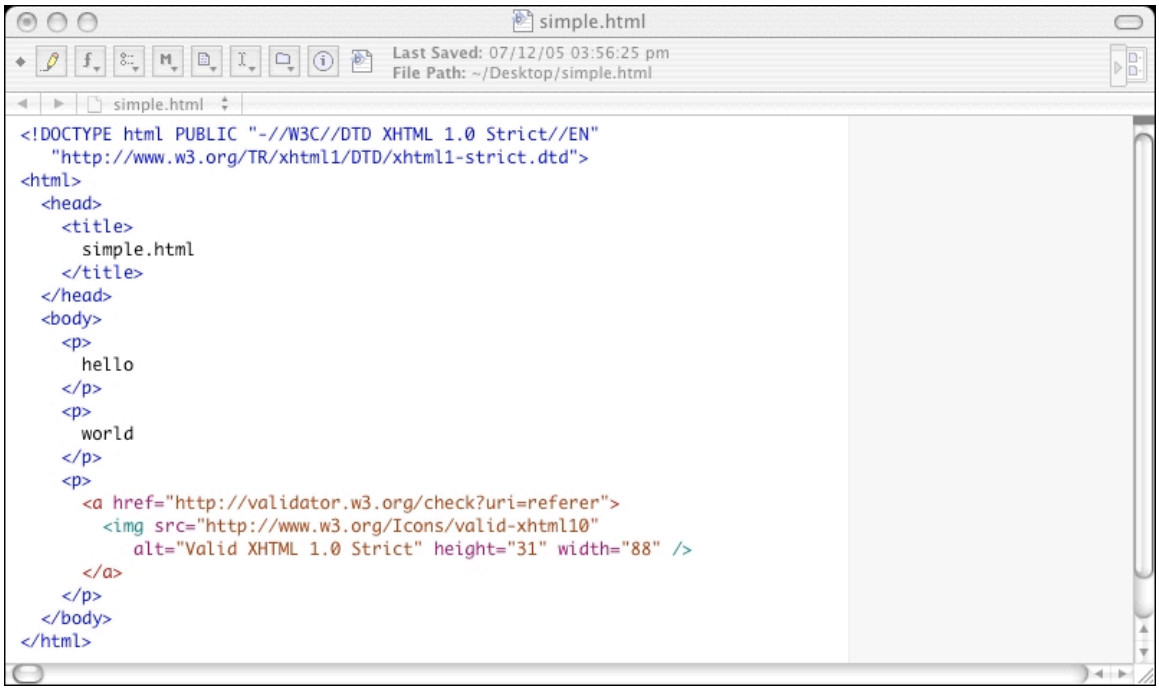
46. mozilla.org (2004) About the Mozilla Foundation [WWW] http://www.mozilla.org/foundation/
47. Though Internet Explorer 7 is currently in development
48. A browser for the Linux operating system. Named following a perceived progression in browsers, that is to say that first came the Navigator, then the Explorer and finally the Konqueror. The spelling is thus because it runs upon the K Desktop Environment (KDE)
49. Henceforth to be known as the DOM
50. Henceforth to be known as CSS

3.3 The Document Object Model

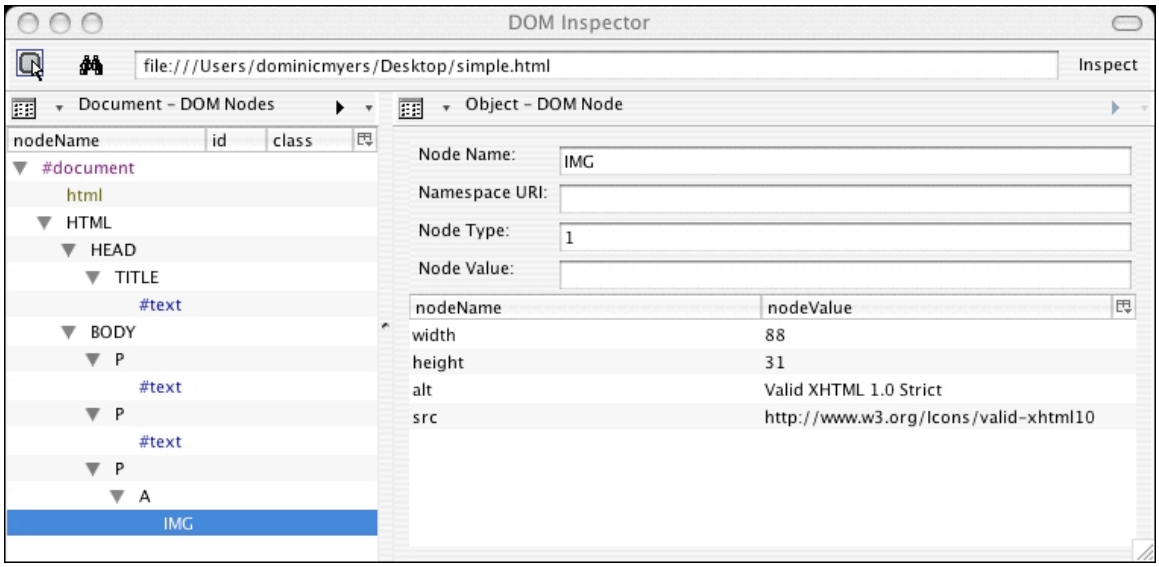
The Document Object Model[51] is an API[52] created by browser manufacturers to allow access to the underlying structure of HTML (and XML) documents. Behind the API a HTML document is represented with an object-orientated model, and sometimes as a tree model. An example of this is offered by the pictures entitled "Browser view", "Editor view" and "DOM Inspector view". These all show different aspects of the same HTML document which is named simple.html. The first, "Browser view", is a screen grab of simple.html as displayed by the Safari browser. The second, "Editor View", shows the HTML source of simple.html and the last, "DOM Inspector view", is how the DOM Inspector in the FireFox browser represents the document.

Picture 1: Browser view

51. Henceforth to be known as DOM
52. Application Programming Interface



Picture 2: Editor view



Picture 3: DOM Inspector view

Netscape released the first DOM[53] and other browsers followed, though their implementations were sometimes different they were based upon the work

of Netscape. Because they were different the World Wide Web Consortium[54] standardising the DOM and, though never recognised by the W3C, the DOM Level 0, as created by Netscape in their 2.0 release served as the basis for this standardisation. Presently all major browsers support the W3CDOM.

53. To allow JavaScript to interact with HTML

54. Henceforth to be known as the W3C. The W3C mission is: "To lead the World Wide Web to its full potential by developing protocols and guidelines that ensure long-term growth for the Web". Jacobs I (2005) About the World Wide Web Consortium (W3C) [WWW] http://www.w3.org/Consortium/

3.4 Cascading Style Sheets

While HTML splits the data and the formatting of documents there is very little control over how the formatting appears. Different browsers implement formatting in different ways. With the first release of the Netscape browser some additional HTML elements[55] were introduced and these have been added to but it became clear to the architects of HTML[56] that as a language HTML should never become a page specification language.

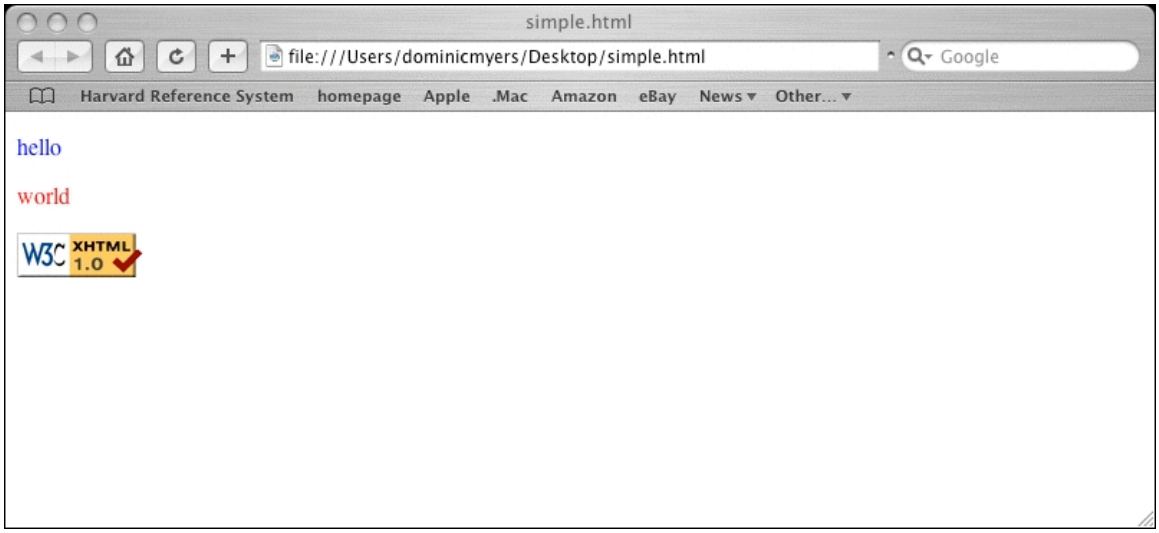
Berners-Lee had a mechanism in Nexus which allowed for the styling of pages but it has been suggested by Bersvendsen[57] that because Berners-Lee felt that the browser should dictate how pages should look he never released the specification.

While it may have been sufficient for academics to have minimal control over the final rendering of a page the web was becoming more popular and businesses and designers were far more conscious of image, preferring not to let software manufacturers dictate how their presence on the web be displayed. This led to a number of people creating languages which might specify the styling of the structural elements of a HTML page. These ideas were eventually synthesised by Lie[58] and Bos[59]. Together they decided that the styling of HTML

---

55. Also often called tags
56. Dave Raggett was the main architect of the HTML 3.0 specification
57. Bersvendson A (2005) Who created CSS? CSS Early History [WWW] http://virtuelvis.com/archives/2005/01/css-history
58. Håkon Wium Lie is presently Chief Technology Officer of Opera Software, where he has

should cascade, that styling information defaults to the browser default but that it could be defined in external CSS files, within the head of the HTML file or within the individual HTML element, these would cascade into one final rendering, generally in that order. If we again look at simple.html (using the picture entitled, "Styled browser view") we can see three methods of entering styling information onto a page, and some examples of how such information used to be entered.



Picture 4: Styled browser view

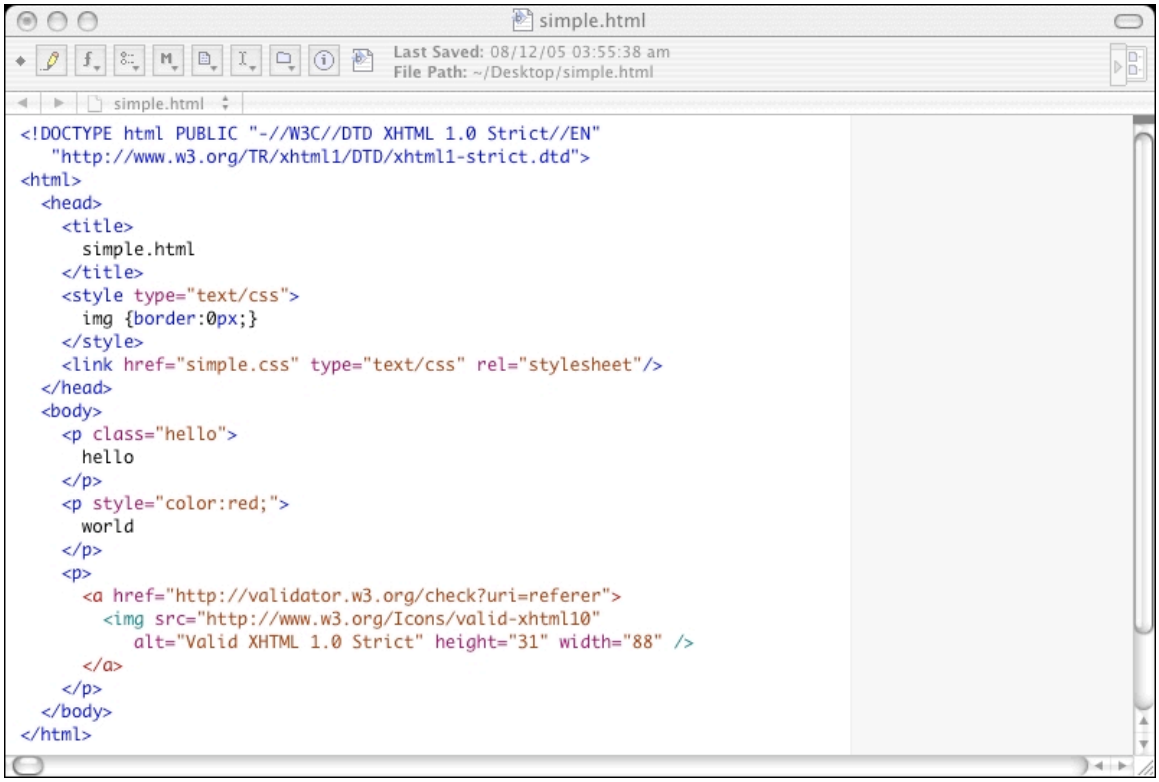
A screen capture of the 2 files (pictures entitled "Styled HTML view" and "CSS view") show how the external CSS file alters the colour[60] of the first

worked since 1999. Wikipedia (2005) Håkon Wium Lie [WWW] http://en.wikipedia.org/wiki/H%C3%A5kon_Wium_Lie

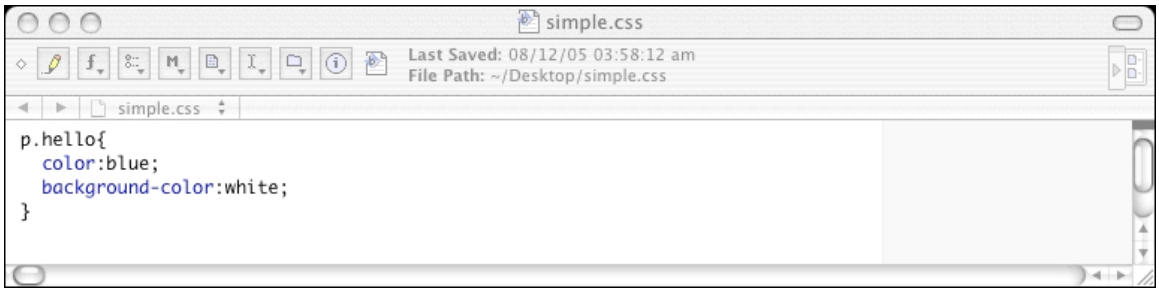
59. Bert Bos was creating a highly customisable browser called Argo when Lie, with the encouragement of Raggart, published the first draft of CSS. At the time it was called Cascading HTML Style Sheets but with the realisation that it could be used with other markup languages is was renamed to CSS

60. CSS uses American English

paragraph, the style information in the header makes the border around the image disappear[61], and the element styling on the last paragraph makes the text red.



Picture 5: Styled HTML view



Picture 6: CSS view

61. Images that are links default to having a blue border in HTML

Styling information can be entered can be entered directly into a HTML element as in the height and width attributes of the image.

3.5 JavaScript

Initially created by Brendan Eich under the auspices of Netscape in 1995 JavaScript has been "extended to contexts that range far beyond the initial intent of its designers"[62], it has also suffered from a lack of development tools such as an IDE[63], debuggers and meaningful error messages. These deficits have not stopped it being by far the most popular language on the web.

While it was being developed it was called Mocha and then LiveScript but when it was released in Netscape 2.0 it was called JavaScript. Its naming is somewhat controversial with Champeon writing that it is because it was designed to aid designers in integrating Java applets[64] into web pages and that its name was changed with Sun and Netscape asserting that it was a complement to HTML and Java. Nicholas Zakas[65] states that it was an attempt to cash in on the latest buzzword of the time which caused Netscape to change the name. Champeon notes that the name change plagued web developers for years to come as the names were confused on mailing lists and Usenet discussions. This situation isn't likely to change as some now call it

---

62. Champeon S (2001) JavaScript: How Did We Get Here? [WWW] http://www.oreillynet.com/pub/a/javascript/2001/04/06/js_history.html
63. Integrated Development Environment
64. "A Java applet is an applet written in the Java programming language. Java applets can run in a web browser using a Java virtual machine (JVM), or in Sun's AppletViewer, a stand alone tool to test applets". Wikipedia (2005) Java applet [WWW] http://en.wikipedia.org/wiki/Java_applet
65. Zahas NC (2005) Professional JavaScript for Web Developers Indiana: Wrox

ECMAScript.

Microsoft countered Netscape's development of JavaScript by first developing its own scripting language called VBScript and then cloning JavaScript and calling it JScript but it lagged behind Netscape in its development of scripting languages.

Stephen Chapman[66] notes that after its conception Microsoft's JScript mimicked JavaScript more and more and suggests that this was because Netscape was at the time the dominant browser. With two different languages offering essentially identical functionality within the two browsers the industry felt that the languages should be standardised and JavaScript 1.1 was submitted to the ECMA[67] in 1996 with it being adopted in 1997 (ECMA-262), it was subsequently adopted by the ISO/IEC[68] in 1998 (ISO/IEC-16262). The Technical Committee[69] included programmers from Netscape, Sun, Microsoft, Borland and others.

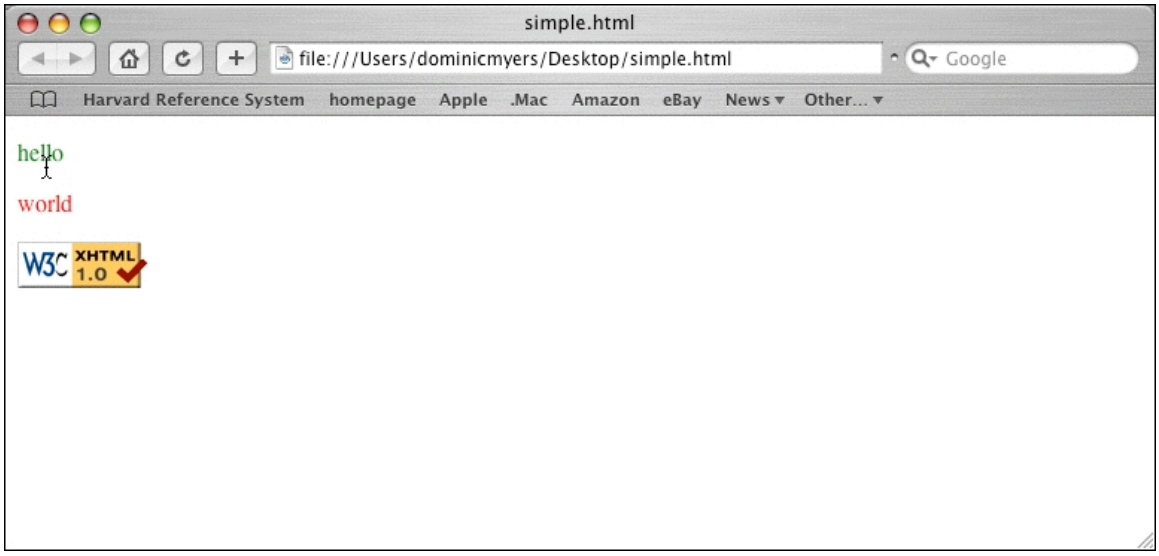
There have been 3 editions of the ECMAScript standard with the first essentially being the same as Netscape's JavaScript 1.1, the second was an update to bring both the ECMA and ISO/IEC standards into strict agreement.

---

66. Chapman S (no date) A Brief History of Javascript [WWW] http://javascript.about.com/od/reference/a/history.htm
67. European Computer Manufacturers Association, an international standards body
68. Both the International Organization for Standardization and the International Electrotechnical Commission are international standards body
69. ECMA TC39 was assigned to "standardise the syntax and semantics of a general purpose, cross-platform, vendor-neutral scripting language". ECMA (no date) TC39 - Programming and Scripting languages [WWW] http://www.ecma-international.org/memento/TC39.htm

The third edition represented the first real update to the standard and Zakas notes that to many, "this marked the arrival of ECMAScript as a true programming language"[70]. Recent releases of Internet Explorer, FireFox, Opera and Safari all now have ECMAScript compliance.

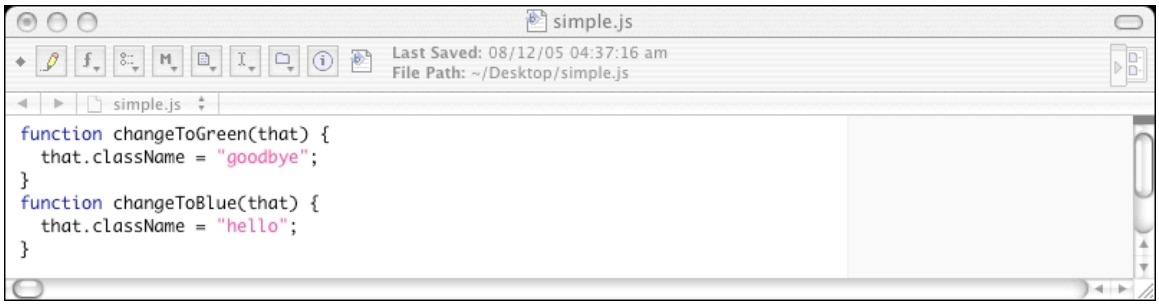
With JavaScript and its access to the DOM we're able to do the effect shown in the picture entitled "Scripted browser view".



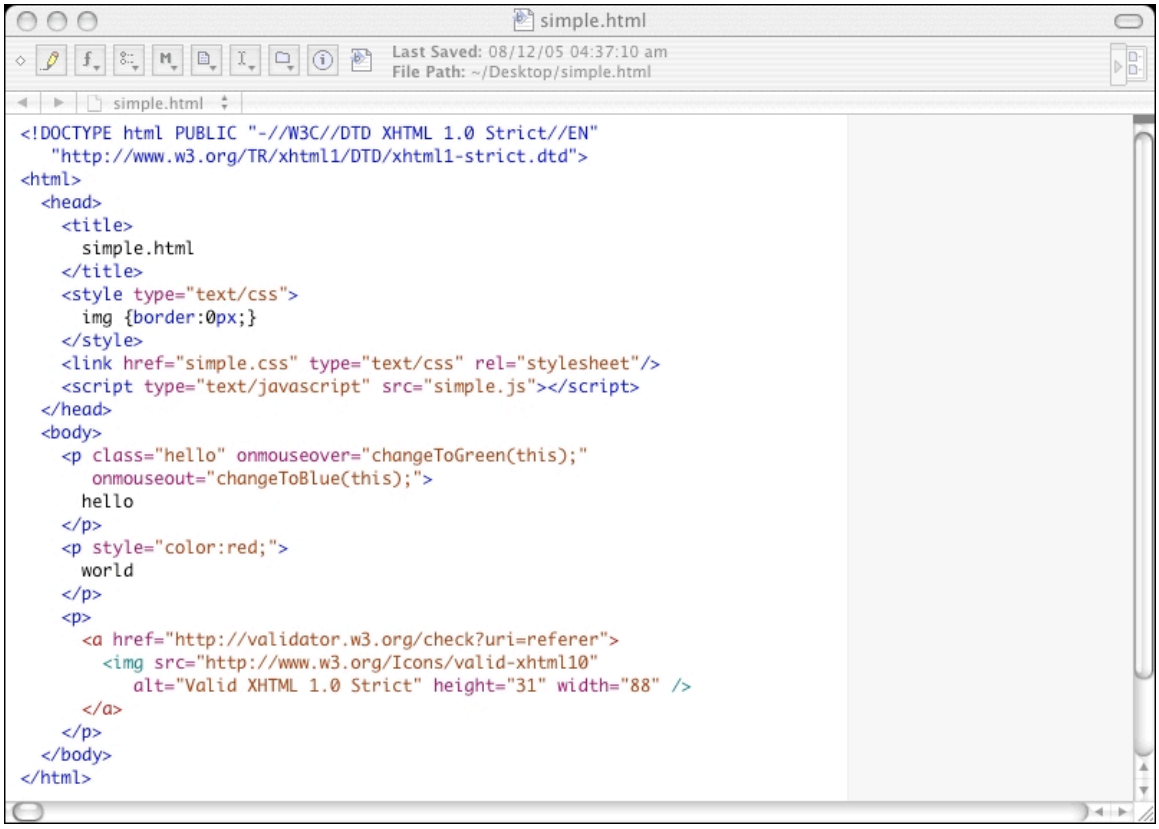
Picture 7: Scripted browser view

The above effect is subtle but the colour of the text in the first paragraph changed due to the action of JavaScript file (which is shown in the picture entitled "JavaScript view") and by adapting the simple.html ("Scripted and styled HTML view") and its associated CSS ("Adapted CSS view") file a little.

70. Zakas NC p4

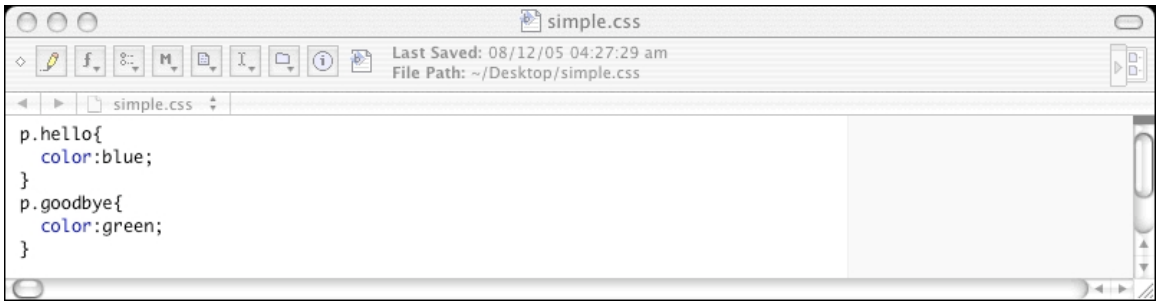


Picture 8: JavaScript view



Picture 9: Scripted and styled HTML view

Now, when the user moves their mouse over the first paragraph its class and thus its style changes. When the mouse moves out it reverts to its original class and style.



Picture 10: Adapted CSS view

3.6 Summary of the Web

We have now looked at browsers and the technologies used to display information to readers over the Internet. HTML, CSS and JavaScript are all ASCII text[71] based technologies and thus able to be created on the simplest of text editors and, once stored on a Internet server or on a local file-system, are accessible through browsers. Once stored on an Internet server it is possible to write server-side scripts[72] which dictate the HTML, CSS or JavaScript sent in response to a request.

Writing in the Financial Times Richard Waters quotes Eich as saying that JavaScript suffered from "premature standardisation"[73]. He also says that:

> *Our original intent with JavaScript was to enable client-side applications that did not have to reload pages from servers to respond to user input, and that offered a rich graphical toolkit to script authors.*

It was not until 1999 that Microsoft created a technology which is now

---

71. "American Standard Code for Information Interchange (ASCII) is the basis of character sets used in almost all present day computers. ASCII text(pronounced ask-ee text), can be read by every computer system, including Windows, DOS, Unix, Macintosh or any other. ASCII, can be created in such applications as Notepad or SimpleText. Unlike Word documents, ASCII documents cannot be formatted, not can they render foreign characters (such as French accented vowels)." bytown internet (no date) glossary [WWW] http://www.bytowninternet.com/glossary

72. "Server-side scripting is a web server technology in which a user's request is fulfilled by running a script directly on the web server to generate dynamic HTML pages. It is usually used to provide interactive web sites that interface to databases or other data stores. This is different from client-side scripting where scripts are run by the viewing web browser, usually in JavaScript. The primary advantage to server-side scripting is the ability to highly customize the response based on the user's requirements, access rights, or queries into data stores." Wikipedia (2005) Server-side scripting [WWW] http://en.wikipedia.org/wiki/Server-side_scripting

73. Waters R (2005) Byline The Financial Times, June 3rd 2005 p14

being used to fulfill Eich's plans for JavaScript.

The major browsers (Internet Explorer, FireFox, Opera and Safari) now offer essentially identical facilities in terms of HTML, CSS, JavaScript and the DOM and display pages served to them from servers via HTTP in essentially identical ways[74].

74. Though the W3C is now in control of the HTML (and more latterly the XHTML) standard each company that produces a browser has their own ideas on the best way to display the HTML within their browser

4.0 Ajax

A browser connects to the Internet and requests a HTML document via the HTTP protocol. A user might input data into the HTML document through some means and send the data back to the server, depending on the data returned another HTML document is sent to the browser. This is the model of the Web as it is most commonly envisaged and is illustrated in the figure entitled "Traditional Web Usage".

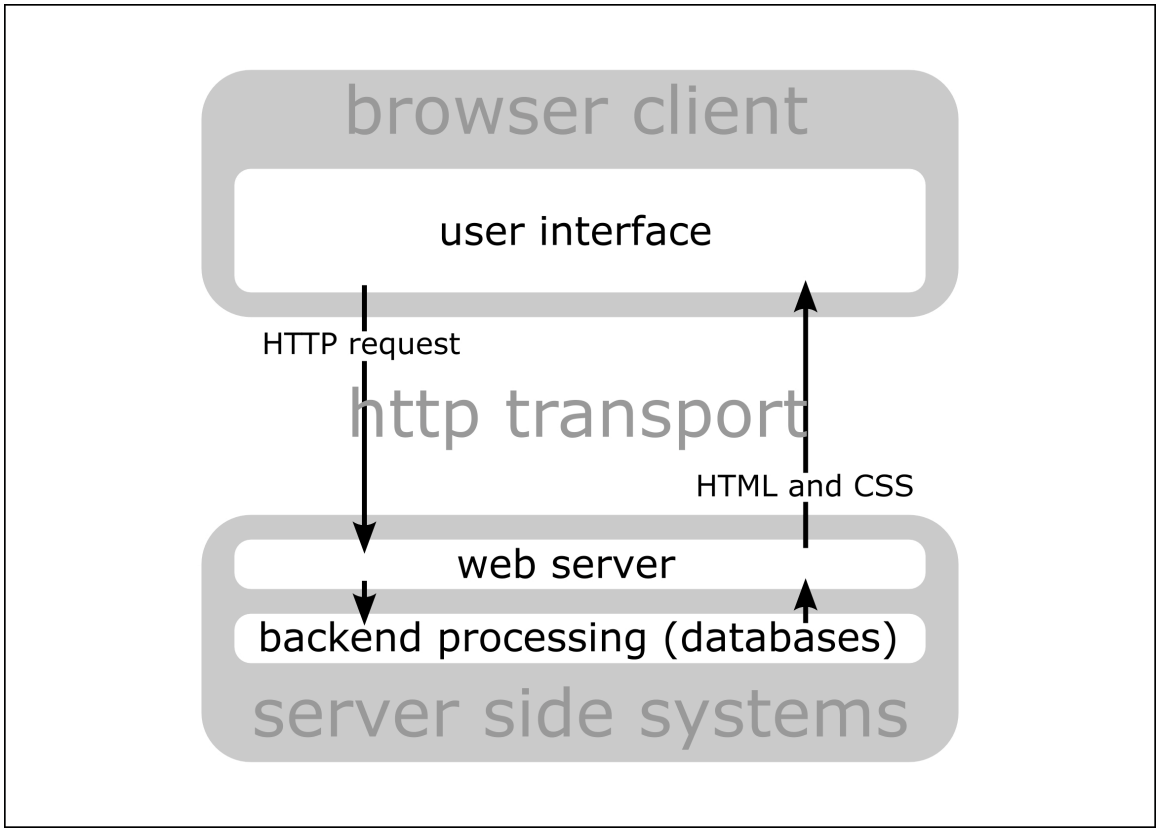


Figure 1: Traditional Web Usage

This model restricts the web as a platform for the development of

applications as the user needs to wait for the HTTP transport to the server and back to the browser. Nor does it take into account the continual improvement in power available on clients machines as it treats the browser as a dumb terminal suitable only for displaying content. Such content, depending upon the site being accessed by the user, might change very little between interaction. That is to say that the page being returned from the server may be structurally exact to that which is being refreshed; further adding to the load placed on the HTTP transport layer.

Jesse James Garrett[75], who wrote the seminal article which popularised the term "Ajax", argues that the delay associated with repeated HTTP requests is the primary reason for using Ajax though he comes at the issue from the direction of user interaction. He uses the model as illustrated in the figure entitled "Ajax Web Usage" to describe a web based application which uses Ajax techniques, saying:

> *An Ajax application eliminates the start-stop-start-stop nature of interaction on the Web by introducing an intermediary — an Ajax engine — between the user and the server. It seems like adding a layer to the application would make it less responsive, but the opposite is true.*
>
> *Instead of loading a webpage, at the start of the session, the browser loads an Ajax engine — written in JavaScript and usually tucked away in a hidden frame. This engine is responsible for both rendering the interface the user sees and communicating with the server on the user's behalf. The Ajax engine allows the user's interaction with the application to happen asynchronously — independent of*

75. "Jesse James Garrett is an information architect and founder of Adaptive Path, an information architecture and user experience firm." Wikipedia (2005) Jesse James Garrett [WWW] http://en.wikipedia.org/wiki/Jesse_James_Garrett

communication with the server. So the user is never staring at a blank browser window and an hourglass icon, waiting around for the server to do something.[76]

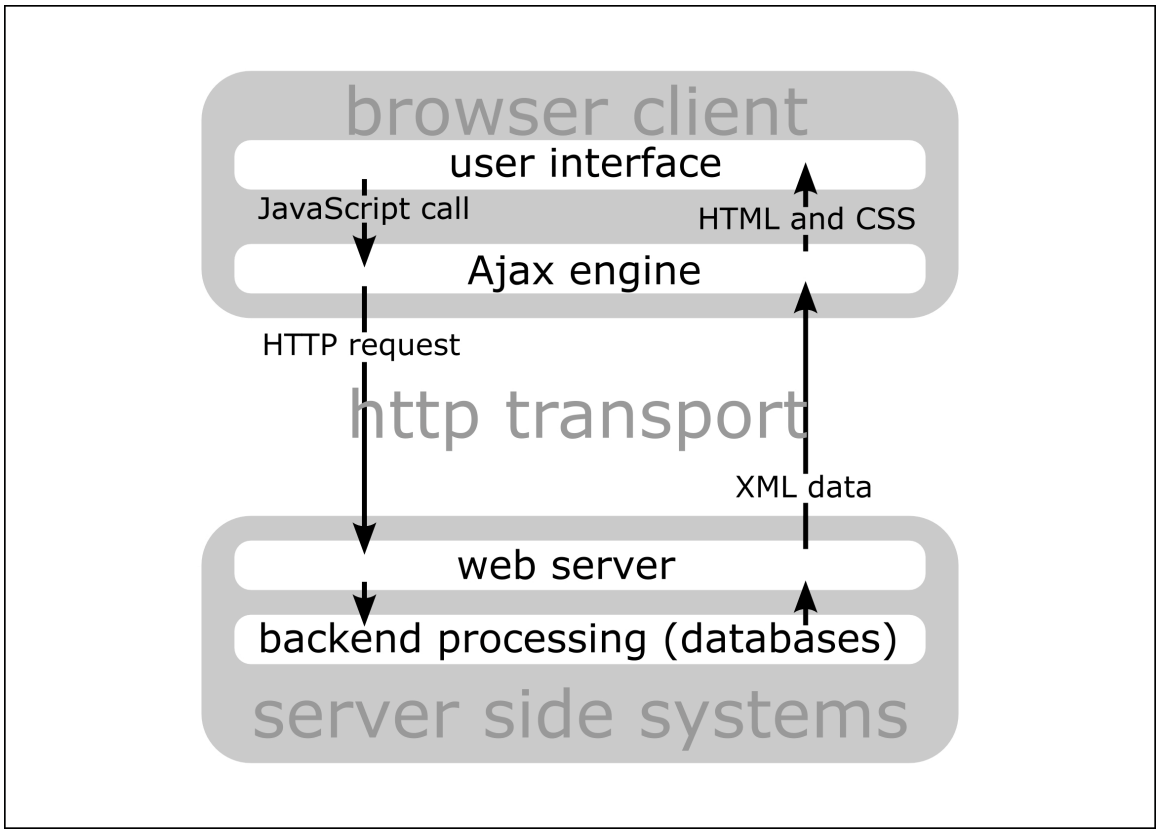


Figure 2: Ajax Web Usage

Ryan Asleson and Nathaniel T Schutta[77] also approach Ajax from the direction of user interaction and suggest Ajax is a technique which removes the necessity of developing software applications for diverse hardware settings, with the associated problems that that entails. They suggest instead that it might be far easier to develop an application which can run on a server and be accessible

76. Garrett JJ (2005) Ajax: A New Approach to Web Applications [WWW] http://www.adaptivepath.com/publications/essays/archives/000385.php

77. Asleson R & Schutta NT (2005) Foundations of Ajax Berkeley: Apress

to any number of users via their browsers. This seems like an eminently sensible approach but was only really possible after the wide-spread adoption of the XMLHttpRequest[78] object.

The XHR object was first introduced in 1999 with Microsoft's Internet Explorer 5 as an ActiveX[79] object and was available to Microsoft's JScript and VBScript. It was not widely used due to the proprietary nature of the technology but Mozilla's engineers implemented a compatible solution for version 1.0[80] of their browser, Apple did the same for Safari 1.2 and Opera has had similar functionality since version 8.0. Sayre[81] has suggested that the primary rationale for Microsoft's development of the object was Outlook[82] Web Access, allowing users of Outlook to have an experience similar to using a desktop based application but through their browser. Despite its name it is not limited to sending or returning XML.

The functionality of Microsoft's ActiveX XHR object is similar to that offered by the other browsers but does require that calls to the object need to

---

78. Henceforth to be known as XHR
79. "A loosely defined set of technologies developed by Microsoft for sharing information among different applications. ActiveX is an outgrowth of two other Microsoft technologies called OLE (Object Linking and Embedding) and COM (Component Object Model). As a moniker, ActiveX can be very confusing because it applies to a whole set of COM-based technologies. Most people, however, think only of ActiveX controls, which represent a specific way of implementing ActiveX technologies." Webopedia (no date) ActiveX [WWW] http://www.webopedia.com/TERM/A/ActiveX.html
80. Compatible to Netscape 7.0
81. Sayre R (2005) Ever Wonder Why It's Called "XMLHTTPRequest"? [WWW] http://www.franklinmint.fm/blog/archives/000294.html
82. Microsoft Outlook is a personal information manager from Microsoft, and is part of the Microsoft Office

employ branching logic in order to differentiate between the different implementations, thanks to the recent interest in Ajax techniques this logic has now been encapsulated into any number of Ajax libraries.

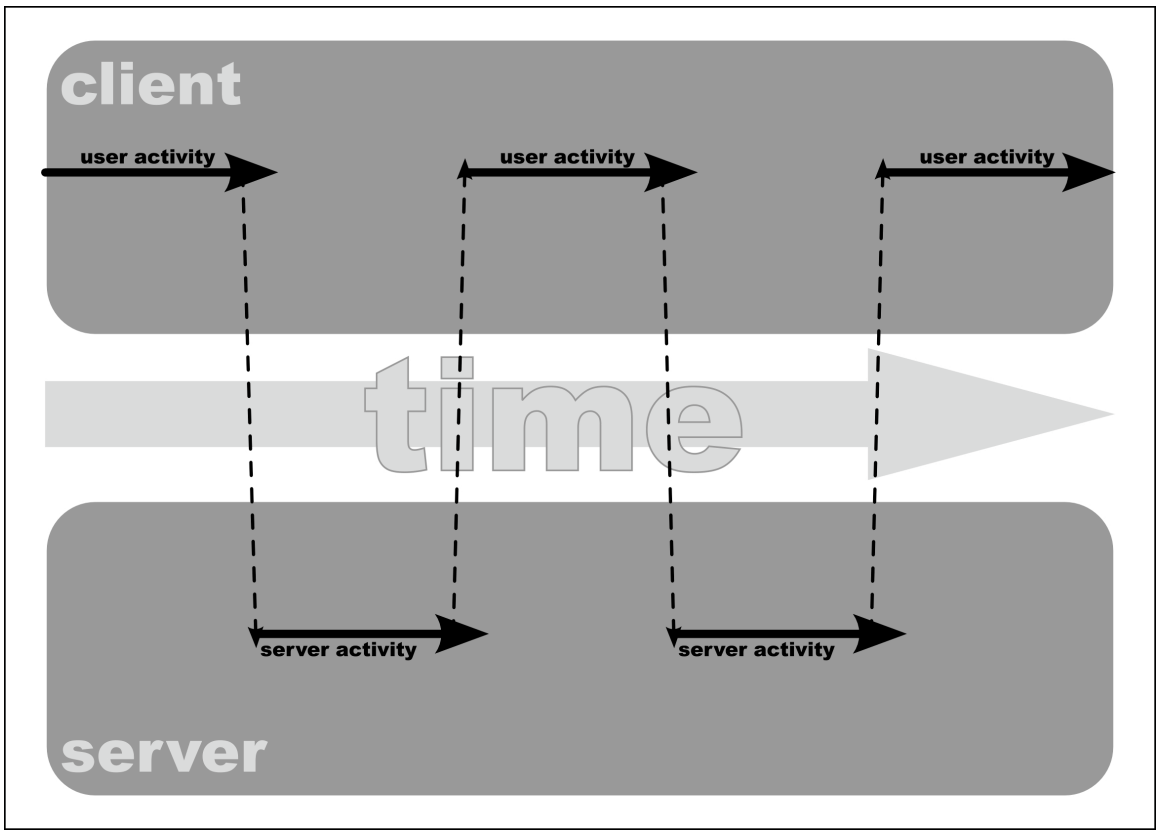


Figure 3: Classic web model (synchronous)

At its simplest Ajax techniques allow for "little bits of change"[83] to be incorporated into a web page without refreshing the underlying HTML, that is to say that a page is served to the browser which has the ability to further query the server depending on demand. In terms of an illustration we could use the

83. Mahemoff M (2005) AJAX Web Architecture podcast radio program [WWW] AJAX Web Architecture

illustration entitled "Classic web model (synchronous)" to show traditional web based application usage with the illustration entitled "Ajax web model (asynchronous)" showing an Ajax application.

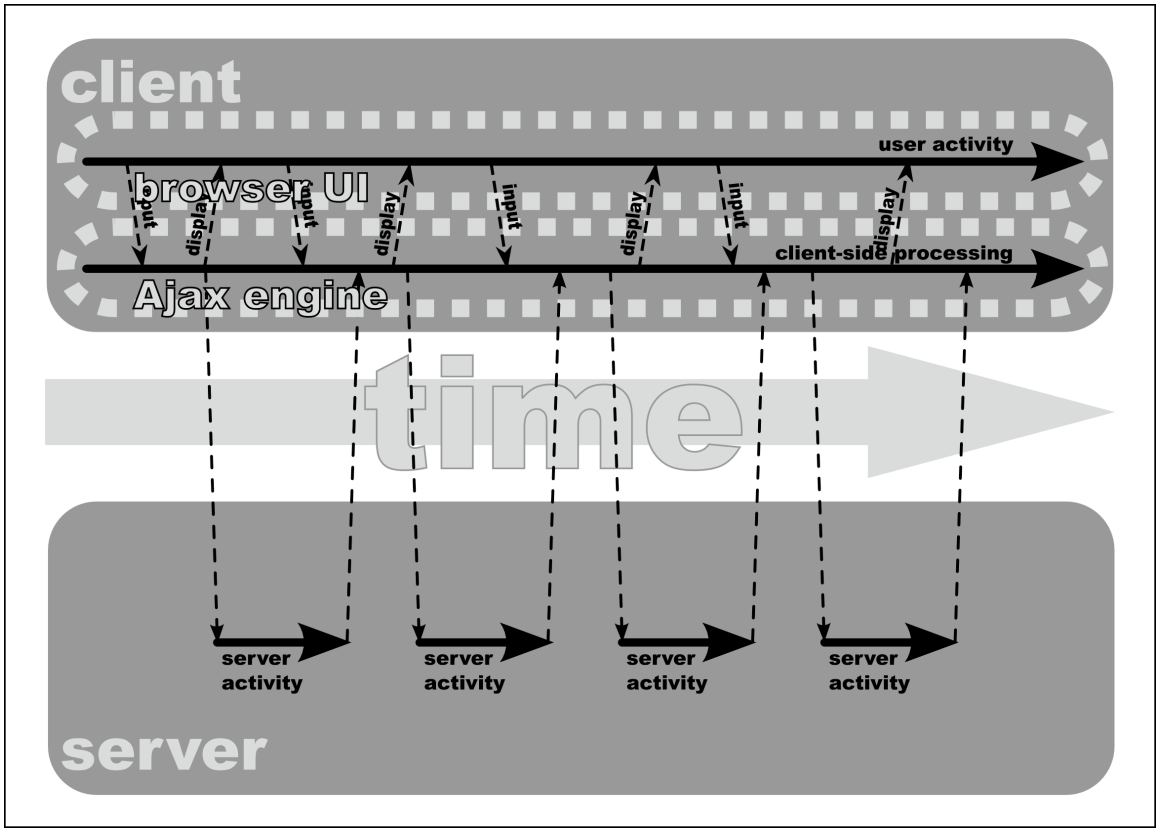
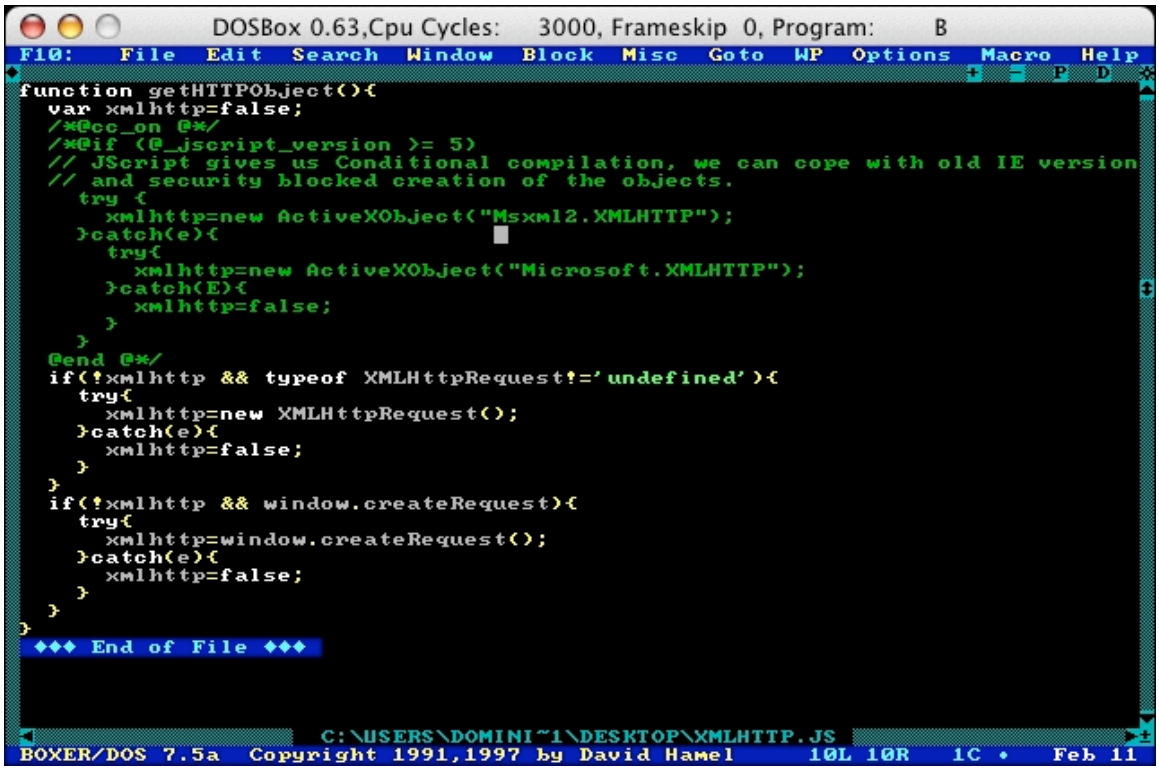


Figure 4: Ajax web model (asynchronous)

4.1 The getHTTPObject function



Picture 11: The getHTTPObject code view

As noted above the XHR object has been incorporated into a number of different code libraries. For the purposes of this study the getHTTPObject from Jim Ley[84] will be used as it has been actively developed over a number of years and works with modern versions of IE, FireFox, Safari and Opera. At it's basic is looks like the picture entitled "The getHTTPObject code view". Analysis of the code sees that the function is passed no variables from outside and that xmlhttp

84. Ley J (2006) Using the XML HTTP Request object [WWW]http://jibbering.com/2002/4/httprequest.html

is initialised at the beginning of the code block. The code highlighted by green in the screen grab is an example of JScript conditional compilation, conditional compilation is described by the Microsoft Developers Network[85] thus[86]:

> "*Conditional compilation enables JScript to use new language features without sacrificing compatibility with older versions that do not support the features. Some typical uses for conditional compilation include using new features in JScript, embedding debugging support into a script, and tracing code execution.*"

The green highlighted code will create the XHR object if the browser is Internet Explorer, the most popular browser on the Internet, with the code that follows creating the XHR object for browsers with native support for XMLHttpRequest (such as Firefox and Opera). The final block of code creates the XHR object in the ICEbrowser[87] using its window.createRequest() method, this is probably superfluous for our purposes but is left in the maintain compatibility with as many browsers as possible.

Microsoft developer Sunava Dutta recently announced on the Internet Explorer Blog that the forthcoming IE7 will "support a scriptable native version of XMLHTTP"[88]. This means that the forking and browser detection implemented

---

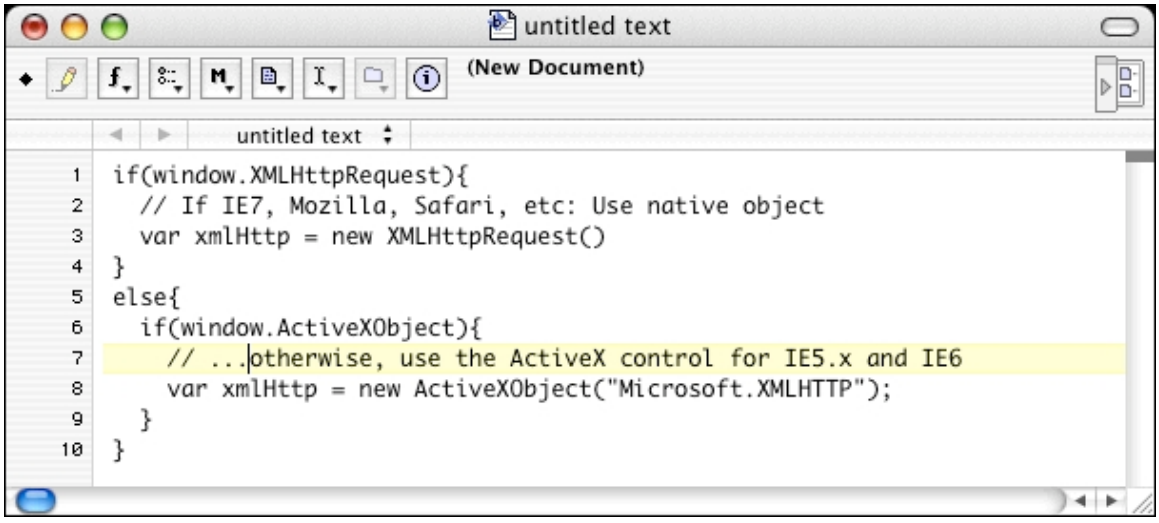
85. The Microsoft Developers Network (http://msdn.microsoft.com/), is described as "the portion of Microsoft responsible for managing the firm's relationship with developers." Wikipedia (2006) Microsoft Developer Network [WWW] http://en.wikipedia.org/wiki/Msdn

86. MSDN (2006) JScript .NET Conditional Compilation [WWW] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/jscript7/html/jsConditionalCompilation.asp

87. The ICEbrowser has been described by Jim Ley thus: "IceBrowser is quite a nice browser, it's Java designed for embedding in other situations, I don't really know what their business model is or if it's successful, but the actual browser really is pretty good, especially for something written in Java." Ley J (2006) IceBrowser and XMLHTTP Request [WWW] http://jibbering.com/blog/?p=189

88. Dutta S (2006) Native XMLHTTPRequest object [WWW] http://blogs.msdn.com/ie/archive/2006/01/23/516393.aspx

in the getHTTPObject above will no longer required. The code as shown in the picture entitled "Future getHTTPObject code view" should suffice in future:



Picture 12: Future getHTTPObject code view

This would make the development of Ajax based application significantly easier though it must be noted that it would still be possible for organisations to disable the new native XHR object using Windows operating systems Group Policy or IE Options. Having said that it is also possible for the native XHR object to be disabled in other browsers by the user opting to disabling JavaScript altogether.

III. Design and Methodology

1.0 Introduction

The development of the application went though three distinct stages but each stage utilised a number of common elements from the web server stack commonly known as XAMP[89]. In this instance the application stack used changed numerous times and these changes will be explored fully in the following sections. The underlying data, that of the modules which make up the degree, also changed as more up-to-date information became available via the Internet[90]. This will also be detailed before a more thorough exploration of the techniques used in the final application is made.

There are alternatives to XAMP, the primary one being based on proprietary, and thus not free, Microsoft products: A Windows based OS[91] with IIS[92] provides similar functionality to Apache running on a Linux machine; ASP[93] provides a scripting interface to Microsoft's SQLServer which is comparable to

---

89. Traditionally this was LAMP (Linux, Apache, MySQL and PHP), though recently the Perl and Python scripting languages have been added as has support for other operating systems, thus the "X"
90. The address though subject to change is at the time of writing: http://www.anglia.ac.uk/ruskin/en/home/prospectus/undergrad2006/computing/Computer_Science_BSc__Hons_.html
91. Operating System
92. Internet Information Services. "It is the world's second most used web server in terms of overall websites but is perhaps the most widely used web server for corporate websites." Wikipedia (2006) Internet Information Services [WWW] http://en.wikipedia.org/wiki/Internet_Information_Services
93. Active Server Pages

PHP/Perl/Python interfacing with the MySQL relational database management system[94].

94. Henceforth to be known as RDBMS

2.0 The Data

The data upon which the application rests was abstracted from notes supplied to students by the college. This data went through two distinct revisions as the notes were changed. The initial set of notes had a number of specifications concerning requirements for the modules. These specifications were analysed to produce the concept of requisites and then further analysed to produce sub-categories of requisites. One such note said, "Applicants must have completed [module X] in order to do this module", which lead to the creation of the pre-requisite requisite. Another said, "Applicants must do [module X] along with this module", leading to the creation of the co-requisite requisite.

A further two types of requisite were abstracted. These are non-requisite (which suggests a comment such as, "Applicants who do this module must not do [module X]".) and post-requisite (suggesting a comment such as, "Applicants who do this module must later do [module X]".). In order to simplify matters it was deemed that the notes such as those associated with the post-requirements could be reversed to make just two types of requisite. After further analysis the non-requirement was not used but it would be possible to implement such a check in the final application.

The scope of this application dictated that the full data set available for the academic institution need not be parsed but that the data available for a

single degree program should suffice. It would be a trivial, though time consuming, process to extrapolate the application to include the full data set. A further barrier was that access to the full data was not available, nor was it sought as the scope of the application was already limited.

There are forty-seven modules in the most recent data set. These have between zero and three requisites each so the data needed to be normalised before placing it in tables, as the original data would produce a table with 10 columns, thus:

Code	Title	Credits	Semester	Requisite Type	Requisite Code	Requisite Type	Requisite Code	Requisite Type	Requisite Code

Table 1: Un-normalised

The "Requisite Type" and "Requisite Code" columns are repeated three times with the majority of modules leaving one or more of these column empty. A table can be said to be in first normal form if it contains no repeating columns so it would be possible to replace the table with headings as above with one with six headings, thus:

Code	Title	Credits	Semester	Requisite Type	Requisite Code

Table 2: First Normal Form

But in order to fill such a table the data in the first four columns would have to be repeated a number of times and there would be times when there would be no entry under the "Requisite Type" and "Requisite Code" columns.

Rather it was deemed better to split the table in two and create the following two tables:

Code	Title	Credits	Semester

Table 3: Second Normal From (1st table)

Code	Requisite Type	Requisite Code

Table 4: Second Normal Form (2nd table)

Thus it is quite possible to recreate the initial table. During the development these two tables became known as "Module" and "Requisites" respectively.

3.0 The First Iterations

The first iterations of the application were developed on both Windows and Linux platforms utilising the Apache web server and PHP version 4. PHP initially gathered data from the TxtDB API but this was replaced by MySQL. This is illustrated in the figure entitled "The first iterations".

The Apache HTTP server is a free and open source[95] application which runs under a number of operating systems. Ostensibly developed and maintained by open source developers under the auspices of the Apache Software Foundation it is credited as being one of the best known examples of open source software[96]. It is also the most popular server on the Internet today, a position it has held since April 1996[97], with the February 2006 Netcraft Web Server Survey showing that it had 68.01% of the server market[98]. It is also claimed to be the one of the key reasons for the initial growth of the Internet.

PHP is a scripted programming language which is primarily used for the creation of (X)HTML pages, though it can be used from the command line in order to fulfil other purposes. It is licensed under the PHP License, a BSD-style

---

95. The licence under which Apache is released is interesting in that it allows for open- and closed-source derivations of its source code to be released, whereas traditional open-source software generally only allows open-source derivatives to be released
96. Välimäki M (2005) The Rise of Open Source Licensing: A Challenge to the Use of Intellectual Property in the Software Industry Helsinki: Turre Publishing
97. The Apache Software foundation (2005) Welcome! - The Apache HTTP Server Project http://httpd.apache.org/
98. Netcraft (2006) Netcraft: Web Server Survey Archives [WWW] http://news.netcraft.com/archives/web_server_survey.html

license, which is considered to have fewer restrictions than other open-source or even traditional copyright licences.

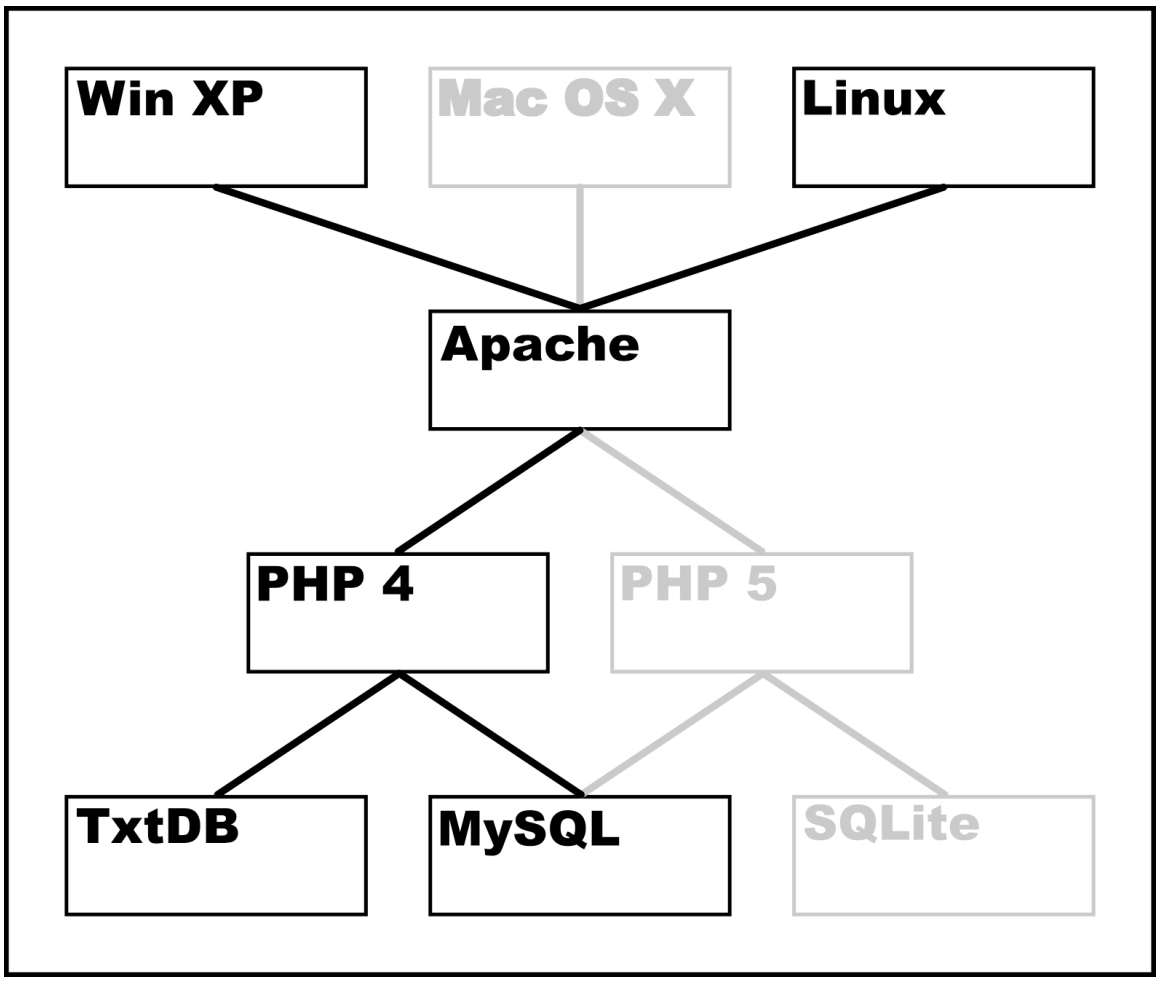


Figure 5: The first iterations

3.1 secondpage

Though developed on a platform which had MySQL[99] the server available didn't offer MySQL for free so an alternative approach was required. After considerable research the Text-DB API from c-worker[100] was utalized as it offered the ability to use properly formatted SQL (thus making future transition to a proper RDBMS less problematic) with the benefits of having the underlying data being contained in ASCII text files easily editable with a suitable editor. From the Php Text Database API Manual[101]:

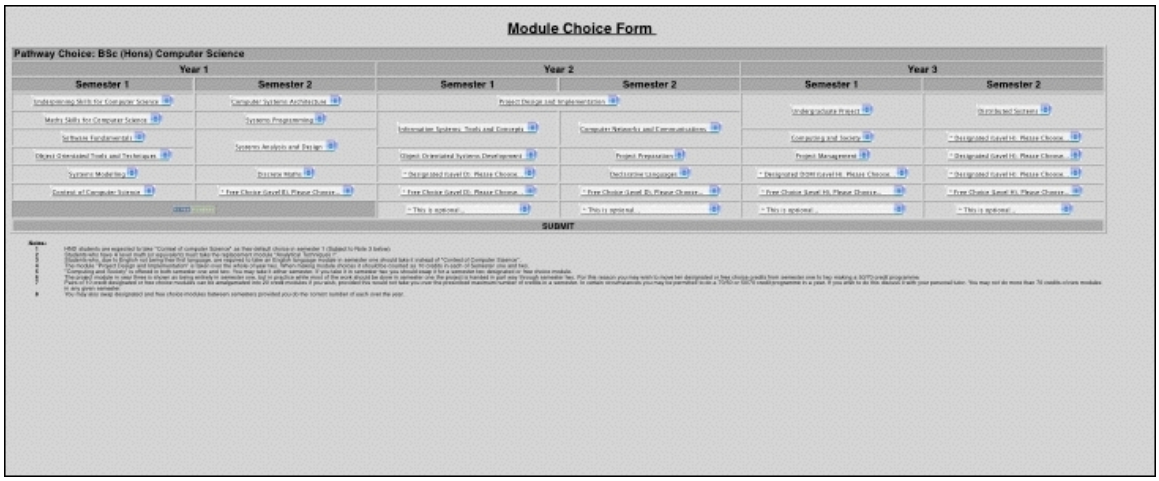
> *"If you wanted to make a dynamic PHP-Website you had two choices:*
>
> *1. You could save your Data in Text-Files in a self defined format. This has one big disadvantage: You'll have none of the advantages a SQL Database could provide, such as performing common SQL commands like SELECT, INSERT, or UPDATE, as well as filtering or setting the order of a result.*
>
> *2. You could host your website on a webserver that supports a database software, but websites hosted on such servers regularly cost extra money.*
>
> *That's where the "Txt-Db-API" could help. With the "Txt-Db-API" you can access normal Text Files like a SQL Database. You don't need any special Database Software on the Webserver, yet you can still use SQL! You can use the API to create, query, change, or delete tables, and you don't have to bother with the Text-Files themselves. Almost everything can be done with the included API calls."*

---

99. Initially this was PHP Triad (http://sourceforge.net/projects/phptriad/), then a commercial WAMP distribution when PHP Triad became too unstable

100. http://www.c-worker.ch/

101. c-worker (2005) Manual [WWW] http://www.c-worker.ch/txtdbapi/manual_html_eng/intro.html

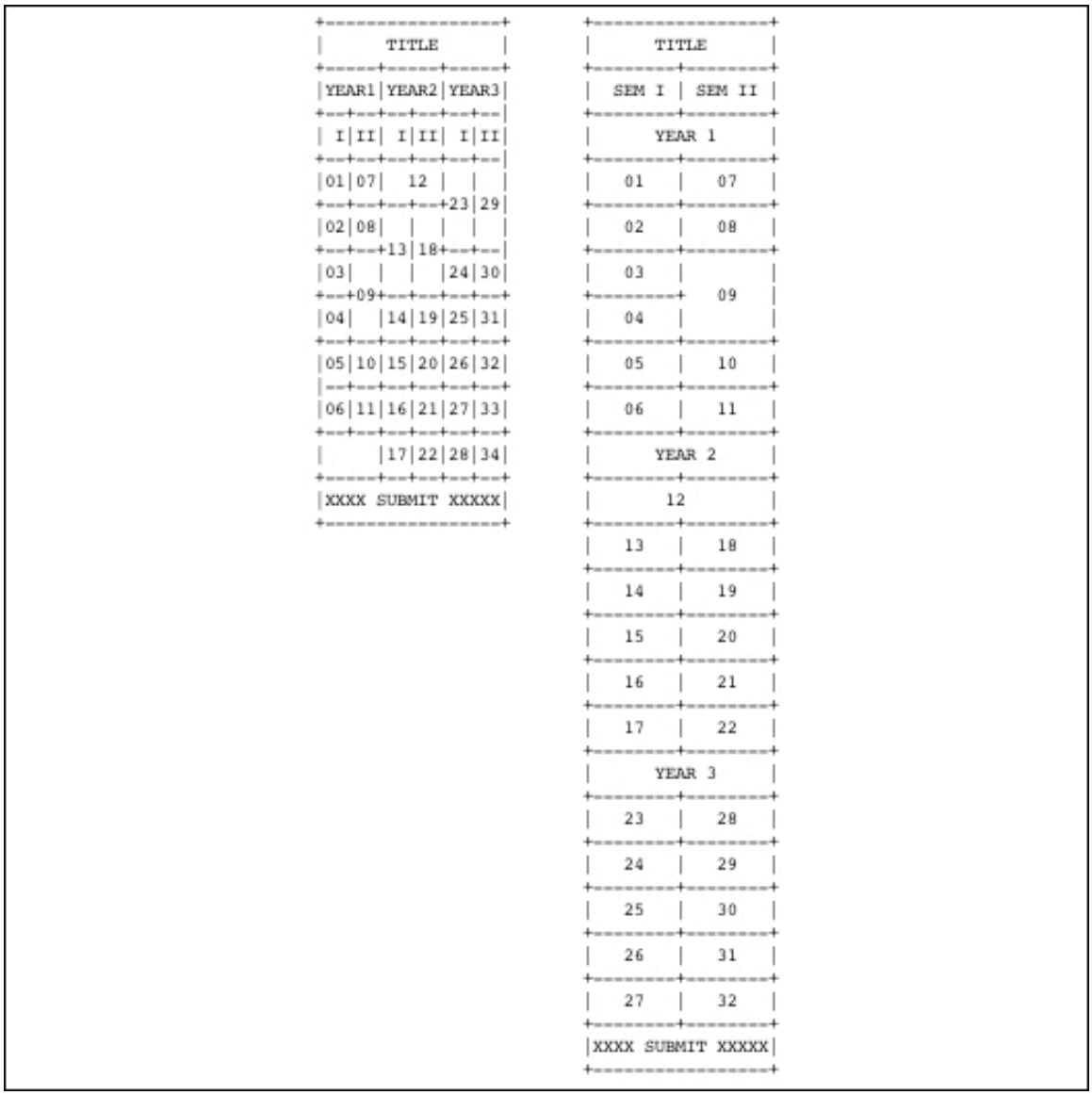


Picture 13: secondpage (50% zoom)

Unfortunately the API is very slow and in order to get the required data from the database PHP needed to be configured to run the script far longer than it is defaulted to allow a script to run; also the server needed to wait longer before sending an error message to the calling page. The resulting user experience was constrained because of this delay. Because this was the initial iteration the format was not overly attractive or even user friendly[102] as the years of the degree were spread horizontally across the page, requiring significant scrolling even on the largest displays. The Opera browser has the ability to zoom the display of pages and the picture titled "secondpage (50% zoom)" is a representation of this iteration at a 50% zoom on a small display.

102. This issue will be discussed at greater length later

3.2 secondpagebest



Picture 14: ASCII table view

The design of secondpagebest was altered so that the years of the degree are arranged vertically over the page rather than horizontally; otherwise the page is functionally identical. Primarily this change was prompted by

aesthetic concerns and is detailed in the comments on the accompanying functions.php file (Appendix 1.4) which are repeated in the picture titled "ASCII table view".

Apart from concerns about the appearance of the page from an aesthetic standpoint there was also a realisation that people were much more used to scrolling vertically rather than horizontally when browsing the Internet. This is borne out by an appreciation of the scroll metaphor in the context of electronic media as discussed by Agarwal-Hollands and Andrews (2001)[103]. However this metaphor may be seen as contra-intuitive when we note that the browser window itself is a horizontal space, especially when a browser's controls are placed in relation to the page being displayed. Williams and Tollett, in their book The Non-Designers Web Book[104], say that an often neglected aspect of interface design is that of page orientation and go on to say that the ideal size for a page is 640 x 460 pixels, allowing the greatest percentage of users to see the page as it was intended.

This shift away from the paradigm of the printed page doesn't seem to hold true when they go on to castigate sites which dictate sideways scrolling in order to navigate. Nielsen[105] has noted a change over time as to whether or not

---

103. Agarwal-Hollands U & Andrews R (2001) From Scroll... to Codex... and Back Again *Education, Communication & Information* **1**(1), pp.59-73

104. Williams R & Tollett J (2000) The Non-Designers Web Book 2nd ed. Berkeley: Peachpit Press

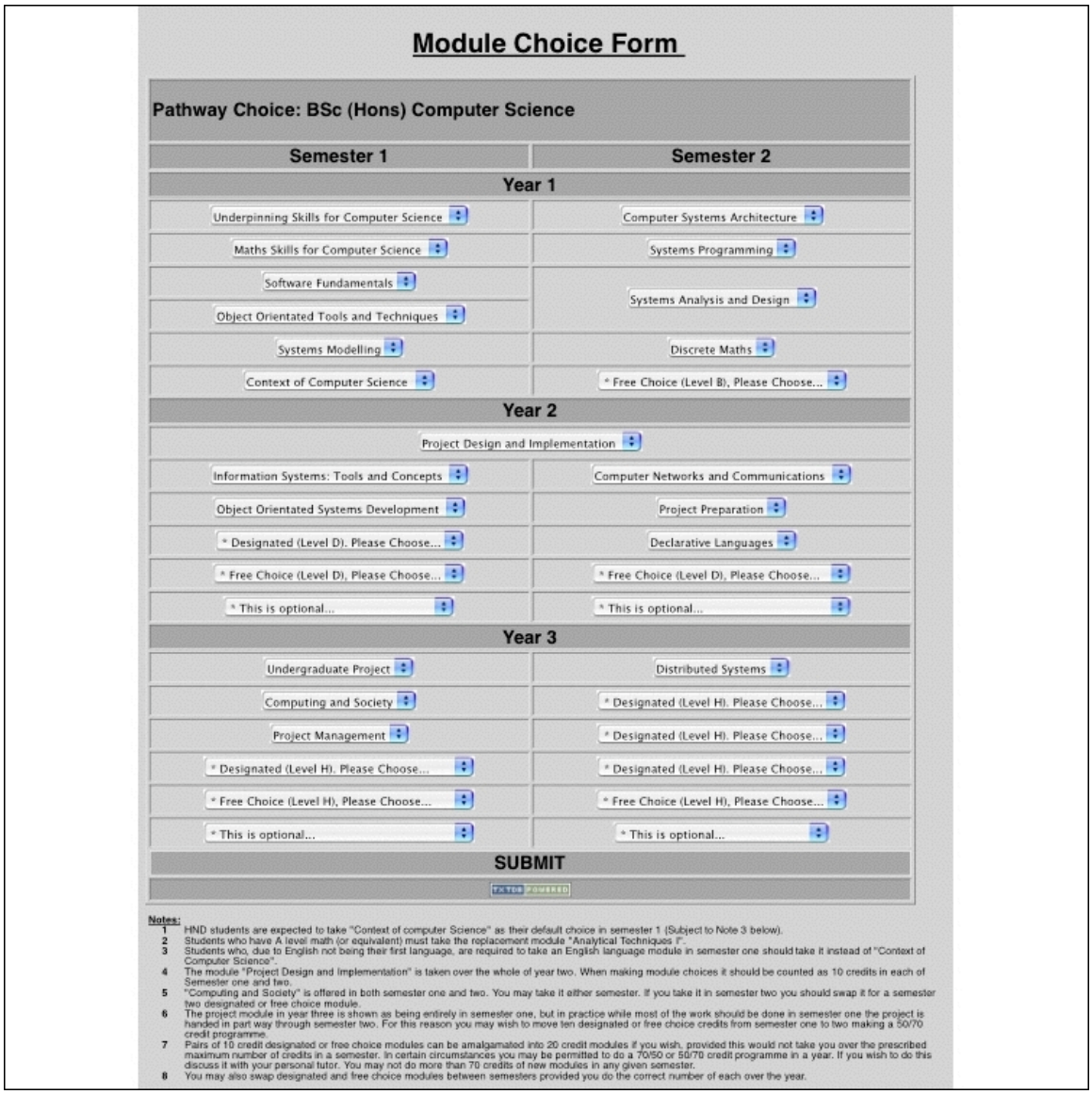
105. Nielson J (2000) Designing Web Usability: The Practice of Simplicity Indianapolis: New Riders

people are willing to scroll within a page with only 10% of users in 1994 scrolling, the exception being those users "who had arrived at a destination page with an article that they found interesting or important to their work" (p112). He goes on to say that that percentage has increased but blames this on the acceptance of bad design.

Swann (2004)[106] further suggests a number of guidelines associated with credible websites. Although he focuses on online news sites it is arguable that such gravitas is just as important on academic sites. The perceived audience of an academic site is not limited to prospective students but also includes current students, academic staff and research sponsors[107]. Thus it might be said that the effort associated with making sites conform to Swann's guidelines might be of benefit not only to the user but also to the credibility of the institution itself.

106. Swann C (2004) WEB DESIGN'S EFFECT ON PERCEIVED CREDIBILITY OF ONLINE NEWS STORIES MASTER OF ARTS IN MASS COMMUNICATION UNIVERSITY OF FLORIDA

107. Olsina L, Lafuente GJ, Godoy D & Rossi G (1999) Assessing the Quality of Academic Websites: a Case Study *New Review of Hypermedia and Multimedia Journal* **5** pp.81-103



Module Choice Form

Pathway Choice: BSc (Hons) Computer Science

Semester 1	Semester 2
Year 1	
Underpinning Skills for Computer Science	Computer Systems Architecture
Maths Skills for Computer Science	Systems Programming
Software Fundamentals	Systems Analysis and Design
Object Orientated Tools and Techniques	
Systems Modelling	Discrete Maths
Context of Computer Science	* Free Choice (Level B), Please Choose...
Year 2	
Project Design and Implementation	
Information Systems: Tools and Concepts	Computer Networks and Communications
Object Orientated Systems Development	Project Preparation
* Designated (Level D), Please Choose...	Declarative Languages
* Free Choice (Level D), Please Choose...	* Free Choice (Level D), Please Choose...
* This is optional...	* This is optional...
Year 3	
Undergraduate Project	Distributed Systems
Computing and Society	* Designated (Level H), Please Choose...
Project Management	* Designated (Level H), Please Choose...
* Designated (Level H), Please Choose...	* Designated (Level H), Please Choose...
* Free Choice (Level H), Please Choose...	* Free Choice (Level H), Please Choose...
* This is optional...	* This is optional...
SUBMIT	

Notes:

1. HND students are expected to take "Context of computer Science" as their default choice in semester 1 (Subject to Note 3 below).
2. Students who have A level math (or equivalent) must take the replacement module "Analytical Techniques I".
3. Students who, due to English not being their first language, are required to take an English language module in semester one should take it instead of "Context of Computer Science".
4. The module "Project Design and Implementation" is taken over the whole of year two. When making module choices it should be counted as 10 credits in each of Semester one and two.
5. "Computing and Society" is offered in both semester one and two. You may take it either semester. If you take it in semester two you should swap it for a semester two designated or free choice module.
6. The project module in year three is shown as being entirely in semester one, but in practice while most of the work should be done in semester one the project is handed in part way through semester two. For this reason you may wish to move ten designated or free choice credits from semester one to two making a 50/70 credit programme.
7. Pairs of 10 credit designated or free choice modules can be amalgamated into 20 credit modules if you wish, provided this would not take you over the prescribed maximum number of credits in a semester. In certain circumstances you may be permitted to do a 70/50 or 50/70 credit programme in a year. If you wish to do this discuss it with your personal tutor. You may not do more than 70 credits of new modules in any given semester.
8. You may also swap designated and free choice modules between semesters provided you do the correct number of each over the year.

Picture 15: secondpagebest

In the realm of presenting data which doesn't fit comfortably on a single page Swann suggests that designers avoid creating pages which require horizontal scrolling and that when vertical scrolling is required then the content should not be greater than two page heights. Callahan (2005)[108] has suggested

108. Callahan E (2005) Cultural similarities and differences in the design of university websites

that such guidelines might be culturally affected. In the present climate of academic institutions endeavouring to attract revenue from international students it might be more appropriate for the layout of the page to be determined by the user according to their needs rather than conform to culturally specific guidelines.

secondpagebest is just over two pages heights[109] and still suffers from the slow loading of its predecessor but it is a significant improvement in that no zooming was required in order to obtain the picture entitled "secondpagebest".

Journal of Computer-Mediated Communication **11**(1) http://jcmc.indiana.edu/vol11/issue1/callahan.html

109. This is when displayed on a relatively small 800 x 600 pixel display. On the smallest standard display of 640 x 480 the page is nearly 4 page heights and the user needs to scroll the page a little to the right in order to select some options

3.3 secondpagebestfast

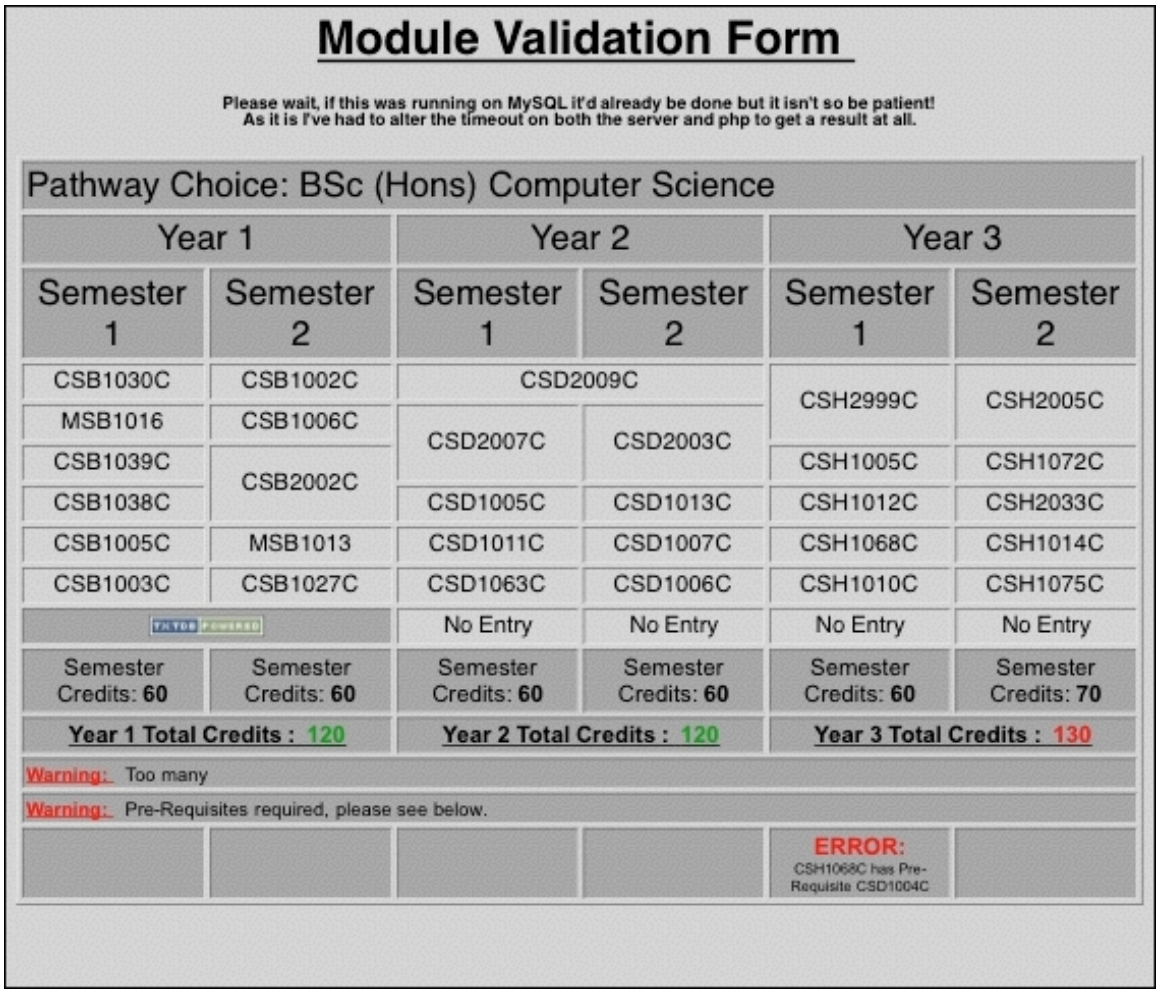
While developing the previous two versions the commercial server which was available began to offer the use of a single MySQL database for free and so the functions within the functions.php file were adapted to use MySQL rather than the Text-DB API. This contributed to significant speed improvements and this is reflected in the title of the page. All of these iterations (secondpage, secondpagebest and secondpagebestfast) were initially loaded from a front page which was called firstpage and offered a number of degree paths[110]. This rather simplistic approach was continued on the page which checked the choices, as it was called thirdpage. The secondpage iterations greatly reduce the module options by dictating a number of initial choices which the user has to go out of their way to alter. This seemed eminently sensible at the time as a significant number of modules were dictated by the degree program notes.

secondpagebestfast is virtually identical to its predecessor and merely represents an improvement of page loading speed as it uses MySQL rather than the processor intensive Text-DB API.

110. Though these other links were "dummy" and had no pages associated with them

3.4 thirdpage

As noted above the secondpage iterations all submit to this page which again uses the Text-DB API to validate the choices made. Initially a number of variables are declared and these are used throughout the page to check different aspects of the degree program.



Module Validation Form

Please wait, if this was running on MySQL it'd already be done but it isn't so be patient!
As it is I've had to alter the timeout on both the server and php to get a result at all.

Pathway Choice: BSc (Hons) Computer Science

Year 1		Year 2		Year 3	
Semester 1	Semester 2	Semester 1	Semester 2	Semester 1	Semester 2
CSB1030C	CSB1002C	CSD2009C		CSH2999C	CSH2005C
MSB1016	CSB1006C	CSD2007C	CSD2003C		
CSB1039C	CSB2002C			CSH1005C	CSH1072C
CSB1038C		CSD1005C	CSD1013C	CSH1012C	CSH2033C
CSB1005C	MSB1013	CSD1011C	CSD1007C	CSH1068C	CSH1014C
CSB1003C	CSB1027C	CSD1063C	CSD1006C	CSH1010C	CSH1075C
		No Entry	No Entry	No Entry	No Entry
Semester Credits: **60**	Semester Credits: **60**	Semester Credits: **60**	Semester Credits: **60**	Semester Credits: **60**	Semester Credits: **70**
Year 1 Total Credits : 120		**Year 2 Total Credits : 120**		**Year 3 Total Credits : 130**	

Warning: Too many

Warning: Pre-Requisites required, please see below.

ERROR: CSH1068C has Pre-Requisite CSD1004C

Picture 16: thirdpage

At the most basic the page accesses the variables passed to it and

checked to see if they have already been selected - this would indicate that the same module had been chosen twice. If it has not then the credits associated with that module are added to the total credits for that semester and the module code is passed into a large array holding all the modules selected. This large array is then processed towards the bottom of the page to ensure that all co- and pre-requisites are met for a given module. This initial process is illustrated in the figure entitled "thirdpage simple flow-chart". Please note that the convention of No being to the left and Yes being to the right is followed in this flow-chart.

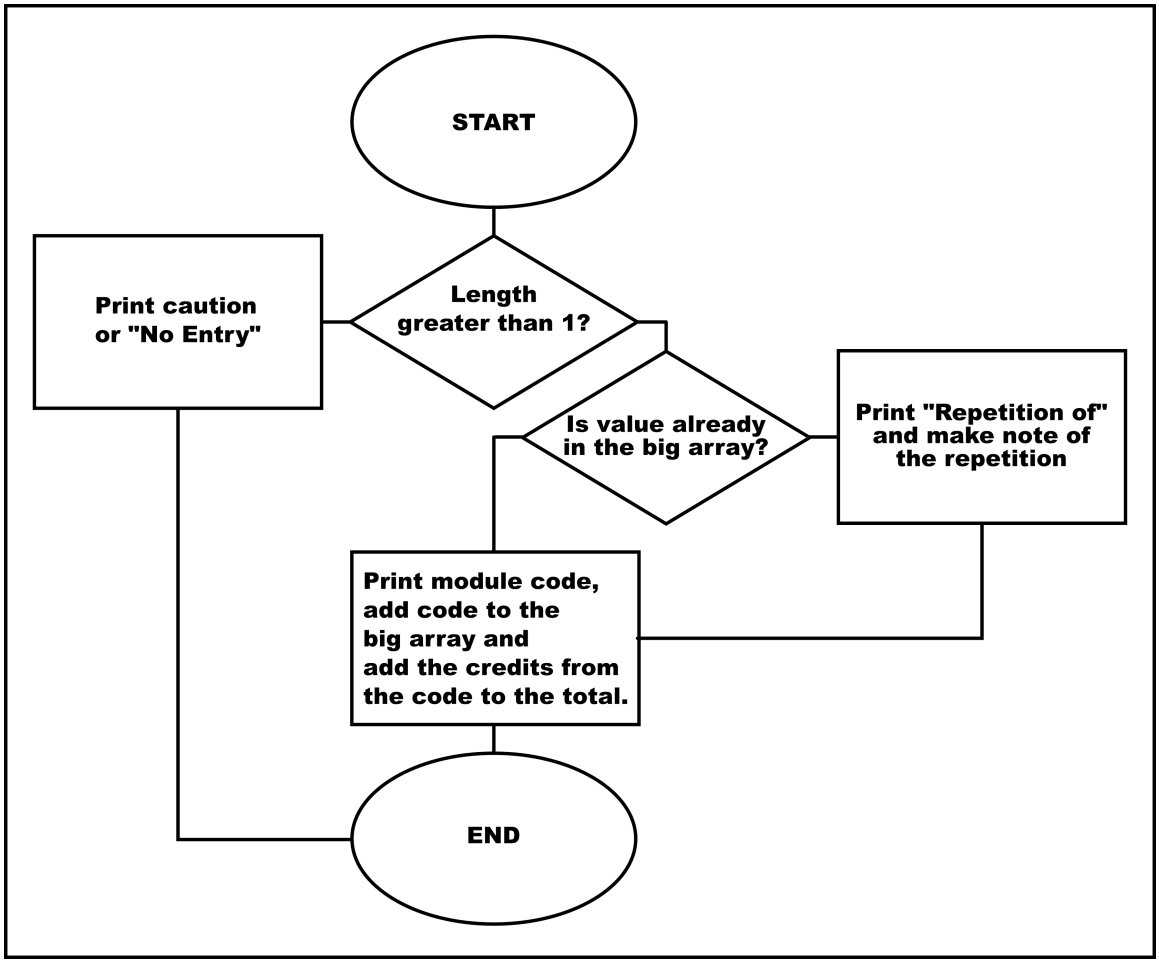


Figure 6: thirdpage simple flow-chart

thirdpage follows the initial format of secondpage in that the representation of the structure of the degree is biased to the horizontal rather than the vertical. At this juncture other aesthetic elements of the application were developed that would last through the iterations. These included the general colour scheme and the highlighting of errors with appropriate colours as can be seen in the picture entitled "thirdpage". At this time it became obvious that the general structure of the application required significant re-engineering, hence the second iterations.

4.0 The Second Iterations

The second set of iterations of the application were developed on Windows XP, Mac OS X and implemented on a Linux server. PHP was upgraded to version 5 on the various servers during the development. The application stack used is illustrated in the figure entitled "The second iterations".

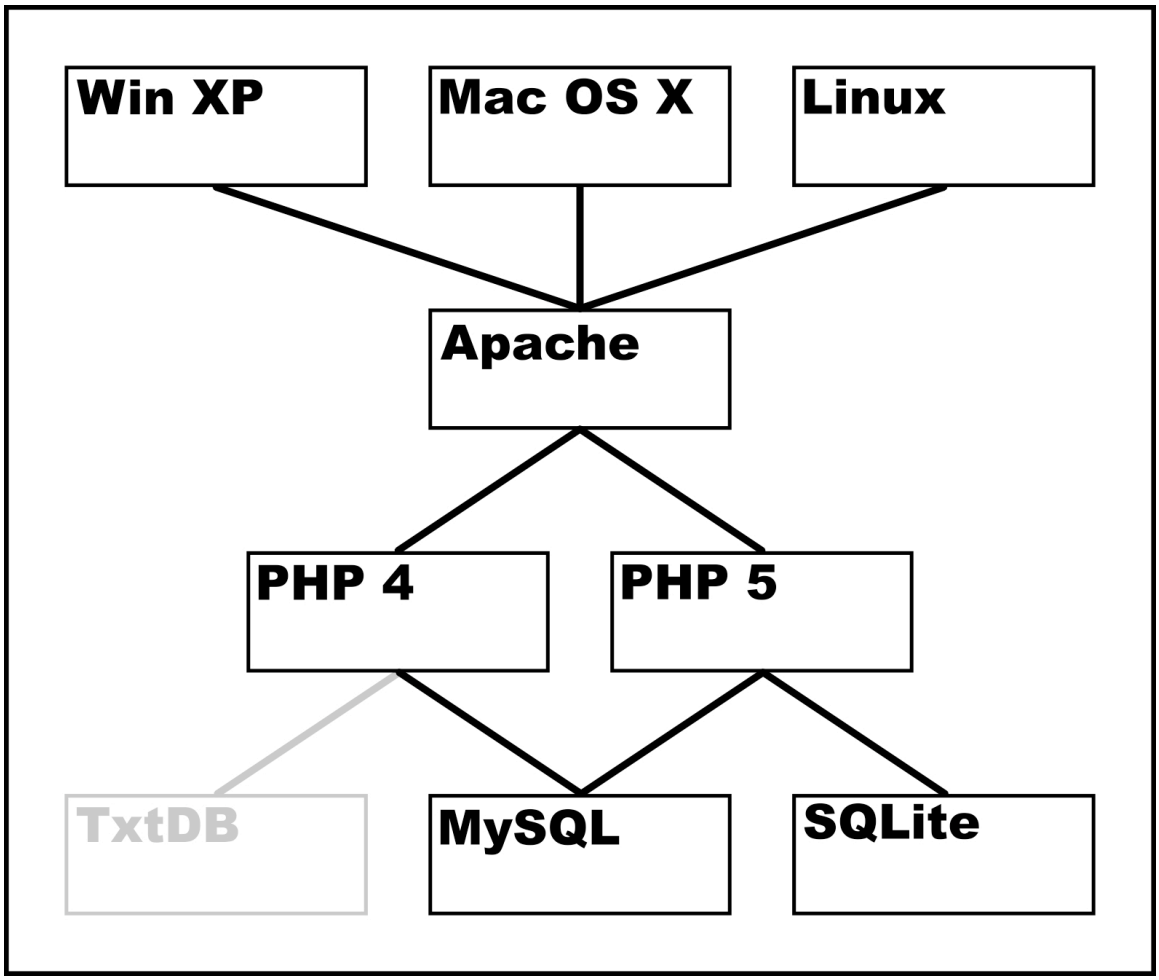
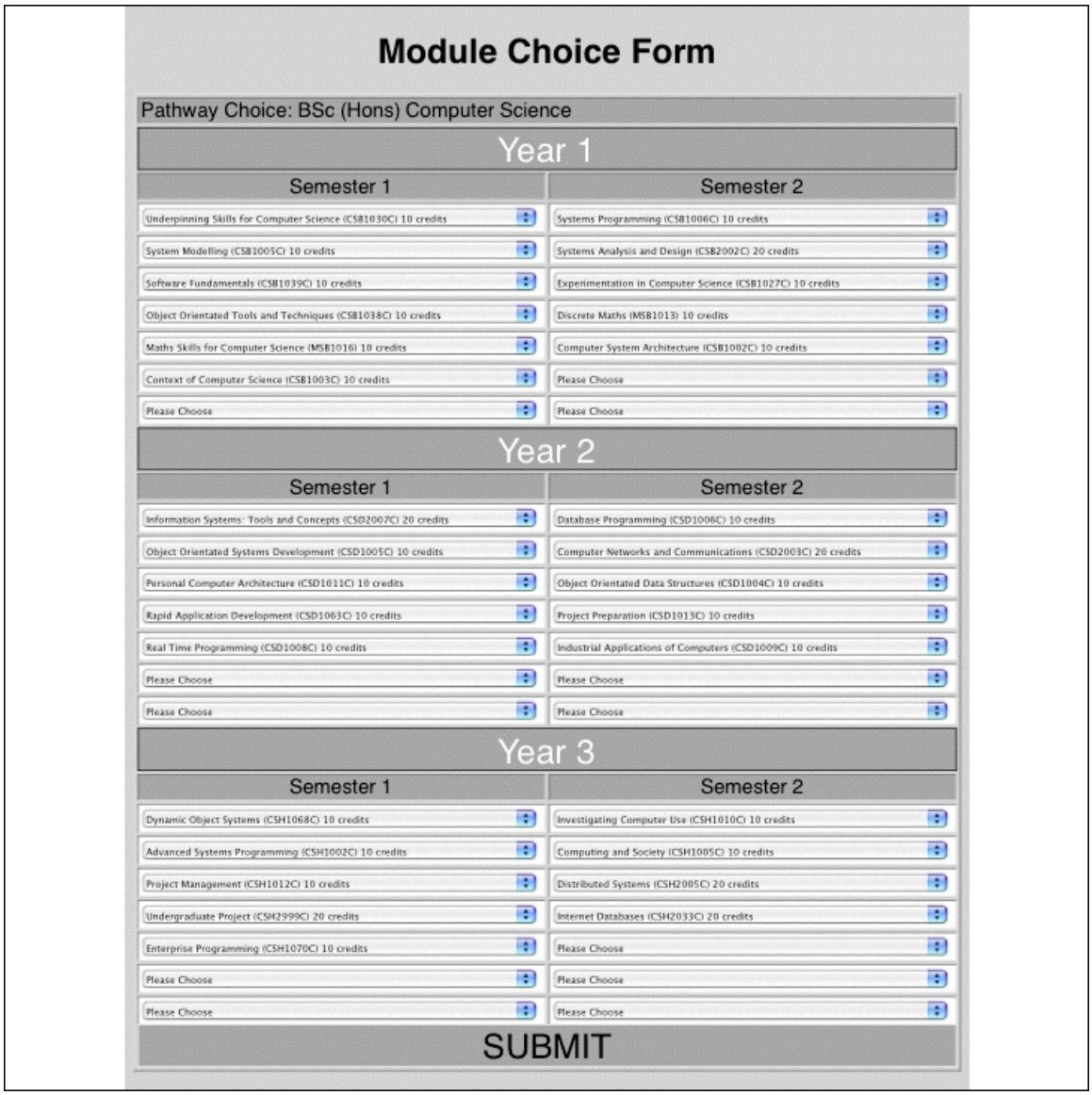


Figure 7: The second iterations

There were two versions developed during this iteration reflecting the

differing services available on the server during the course of the development.

4.1 choice



Picture 17: choice

The first version benefited from a significantly improved understanding of CSS[111] as well as the availability of the MySQL RDBMS. The PHP code is much

111. The dropdown boxes are all now a uniform width for example

clearer as a number of things were abstracted out of the underlying structure. The PHP code is written for the 4th version of that language though it is forward compatible with the 5th version except in relation to the blasé approach to MySQL. MySQL isn't automatically configured in PHP 5 but rather requires specific configuration.

The abstraction allowed for a number of loops which reduced the code significantly, the page is only seventy-two lines long yet produces mark-up which is significantly longer when the source code is cleaned using a tool such as that available for the Firefox browser[112]. It is much cleaner in appearance as can be seen in the picture titled "choice".

The mechanism where a number of modules are pre-selected for the user was removed in these iterations primarily because the code was easier to implement with this mechanism removed.

Without the pre-selection the PHP script simply requires access to the underlying data and then loops through that data with reference to the preferences for that degree contained in a separate file (in this instance this was prefs.php which can be seen in Appendix 2.8). This not only offered significant improvements in terms of code efficiency but required far less intervention on the part of others, that is to say that should the application be implemented a significant stumbling block associated with the adoption of new technology, that

112. The HTML Validator Mozilla extension

of user hostility, might be avoided. Lipson and Fisher[113] have said (p.258):

> "*Successful organisational structures tend to be stable; they resist change even when change is crucial to their ability to survive.*"

Suggesting that the very attributes which make institutions successful are those that stop those institutions from embracing change. By making the application as simple as possible to implement the change is so slight that it is possible for it to be more accepted, for it to slip under the radar if you will.

It might be argued that Lipson and Fisher were concerned with the teaching staff when they discussed barriers to change whereas the application under investigation comes primarily under the auspices of an organisations administrative staff. In 1997 Clegg *et al*[114] found that of the IT investments which do not meet their performance objectives the reason is rarely purely technical, they go on to say (p.851):

> *The context of technical change, the ways in which IT is developed and implemented, a range of human and organisational factors, and the roles of managers and end-users, are identified as critical areas affecting performance.*

The application merely requires access to the database of degrees and modules which administrators are already tasked to produce. No increase or alteration of their skill set is required and no extra work entailed.

Thus it may be argued that the move away from pre-selection added to

---

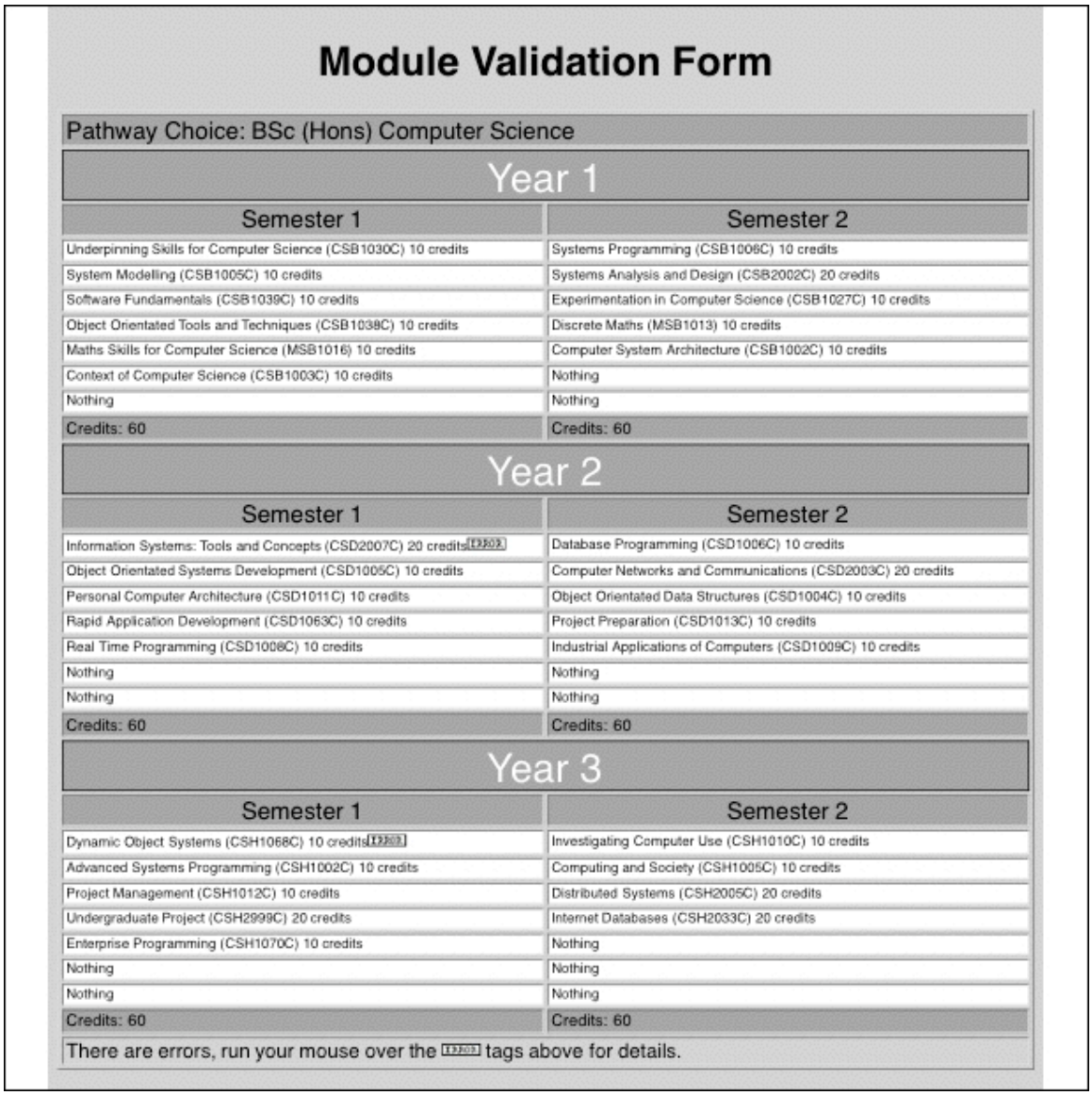
113. Lipson JT & Fisher KM (1983). Technology and the classroom: Promise or threat? Theory into practice **22** pp.253-259

114. Clegg C, Axtell C, Damodaran L, Farbey B, Hull R, Lloyd-Jones R, Nicholls J, Sell R & Tomlinson C (1997) Information technology: a study of performance and the role of human and organizational factors Ergonomics 40(9) pp.851-871

the simplicity of the application and aided towards its integration into pre-existing work-flows. The underlying database need not be MySQL (or even the Text-DB API or SQLite) as PHP includes support for ODBC[115] which means that as long as the database used by administrators is compatible with that standard no further work will be required on their part.

115. Created by Microsoft the "Open Database Connectivity (ODBC) provides a standard software API method for using database management systems (DBMS). The designers of ODBC aimed to make it independent of programming language, database system and operating system." Wikipedia (2006) Open Database Connectivity [WWW] http://en.wikipedia.org/wiki/Open_Database_Connectivity

4.2 validation



Module Validation Form

Pathway Choice: BSc (Hons) Computer Science

Year 1

Semester 1	Semester 2
Underpinning Skills for Computer Science (CSB1030C) 10 credits	Systems Programming (CSB1006C) 10 credits
System Modelling (CSB1005C) 10 credits	Systems Analysis and Design (CSB2002C) 20 credits
Software Fundamentals (CSB1039C) 10 credits	Experimentation in Computer Science (CSB1027C) 10 credits
Object Orientated Tools and Techniques (CSB1038C) 10 credits	Discrete Maths (MSB1013) 10 credits
Maths Skills for Computer Science (MSB1016) 10 credits	Computer System Architecture (CSB1002C) 10 credits
Context of Computer Science (CSB1003C) 10 credits	Nothing
Nothing	Nothing
Credits: 60	Credits: 60

Year 2

Semester 1	Semester 2
Information Systems: Tools and Concepts (CSD2007C) 20 credits [13203]	Database Programming (CSD1006C) 10 credits
Object Orientated Systems Development (CSD1005C) 10 credits	Computer Networks and Communications (CSD2003C) 20 credits
Personal Computer Architecture (CSD1011C) 10 credits	Object Orientated Data Structures (CSD1004C) 10 credits
Rapid Application Development (CSD1063C) 10 credits	Project Preparation (CSD1013C) 10 credits
Real Time Programming (CSD1008C) 10 credits	Industrial Applications of Computers (CSD1009C) 10 credits
Nothing	Nothing
Nothing	Nothing
Credits: 60	Credits: 60

Year 3

Semester 1	Semester 2
Dynamic Object Systems (CSH1068C) 10 credits [13203]	Investigating Computer Use (CSH1010C) 10 credits
Advanced Systems Programming (CSH1002C) 10 credits	Computing and Society (CSH1005C) 10 credits
Project Management (CSH1012C) 10 credits	Distributed Systems (CSH2005C) 20 credits
Undergraduate Project (CSH2999C) 20 credits	Internet Databases (CSH2033C) 20 credits
Enterprise Programming (CSH1070C) 10 credits	Nothing
Nothing	Nothing
Nothing	Nothing
Credits: 60	Credits: 60

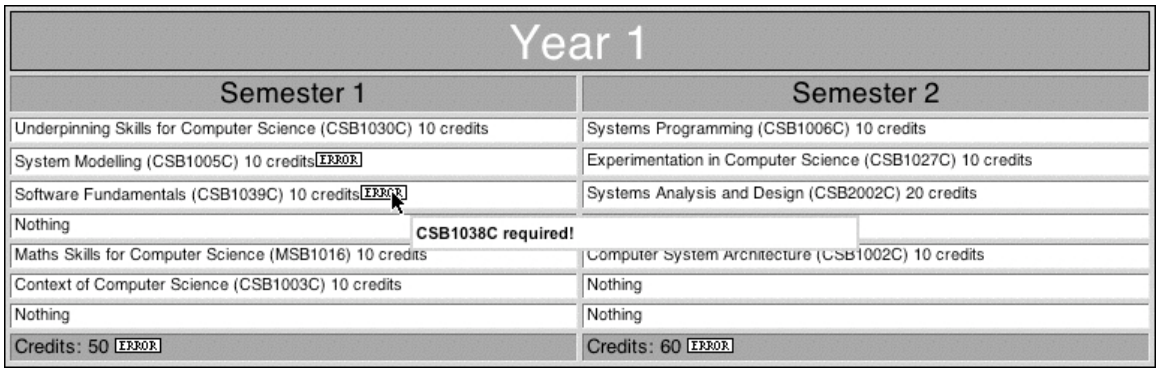
There are errors, run your mouse over the [13203] tags above for details.

Picture 18: validation

The naming scheme for the different pages of the application was rationalised within this iteration as the choice page submitted data to the validation page (picture "validation" shows the result of a validation of the data

submitted in picture "choice").

It is very similar to the choice form in appearance and reproduces the loops used in the choice page in order to fill itself once the data has been collated into a multi-dimensional array. The notes and comments at the bottom of thirdpage are removed and where problems are found there is a simple error image by the module title. Upon the user moving their mouse over the error image a tooltip appears detailing the nature of the problem. This approach seems far more intuitive as errors are directly linked to their cause rather than residing in a different area of the screen. This is illustrated in the picture titled "Highlighted error on validation", which shows that the module with the code CSB1038C[116] is required to to do the module Software Fundamentals (CSB1039C).



Picture 19: Highlighted error on validation

These tooltips are implemented using Walter Zorn's[117] cross-browser

116. Though it is not detailed on the page, that module is Object Orientated Tools and Techniques

DHTML Tooltips[118] which allows for the addition of JavaScript tooltips once the page is fully loaded within the browser. Stylistically the page still leaves a little to desired as the error images are placed directly next to the text and some mechanism whereby the error messages could be placed in a uniform location throughout the page was implemented in the next iteration. Also, the use of codes in the error messages meant that the user needed to hit the back button on their browser while remembering the code and then searching for the location of modules with that code, this was also altered in the next iteration.

117. http://www.walterzorn.com

118. http://www.walterzorn.com/tooltip/tooltip_e.htm

4.3 sqlite_choice_xhtml

The only differences in this iteration was that the code was made XHTML[119] compliant and that the underlying database was changed to SQLite.

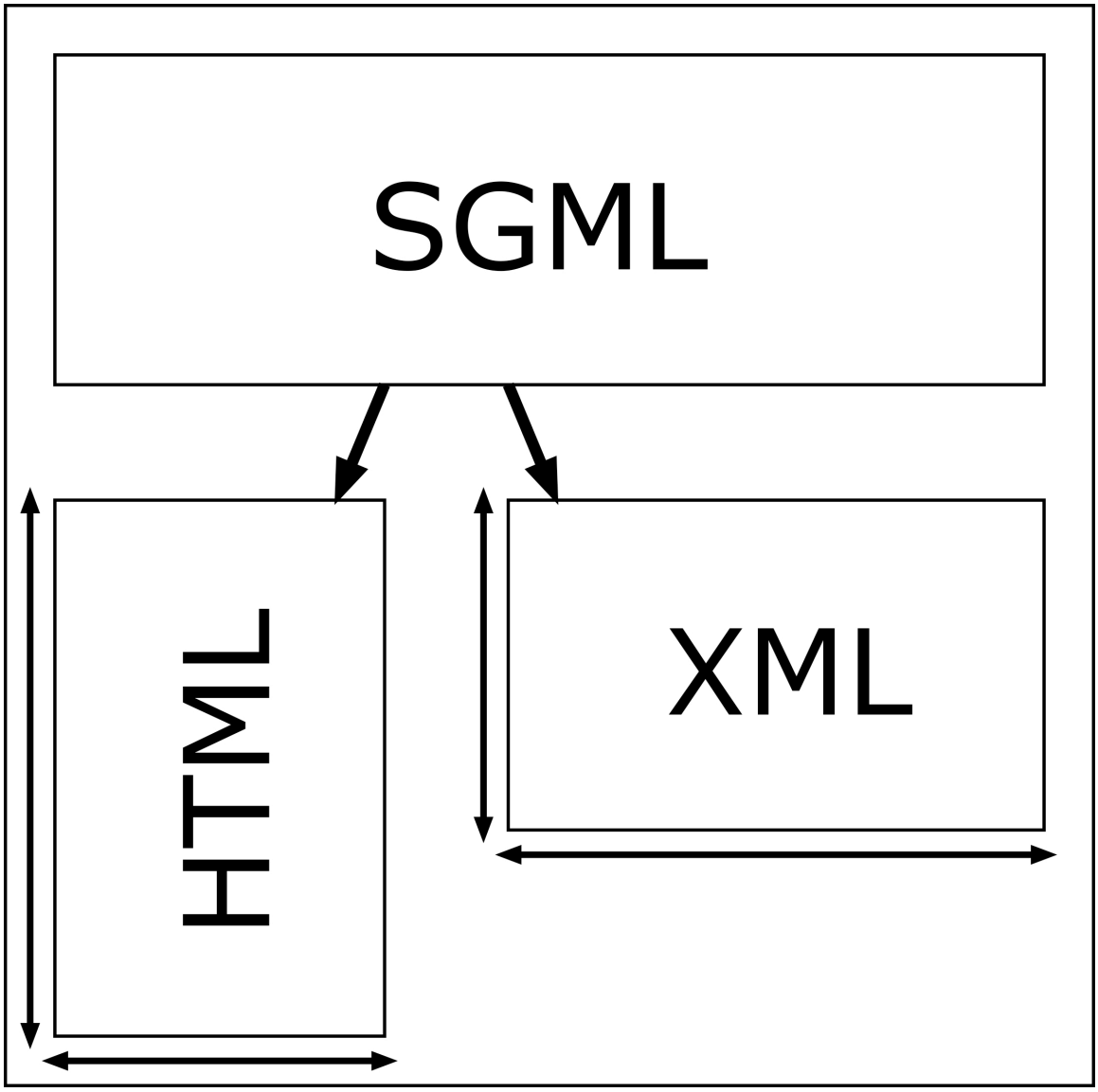


Figure 8: The relationship between HTML and XML

119. Extensible HyperText Markup Language

The relationship between HTML and XML has already been discussed in the literature review but the figure titled "The relationship between HTML and XML" illustrates how both HTML and XML represent re-factored aspects of SGML, XHTML is simply a more restrictive subset of SGML. There is much debate about whether or not using XHTML is of any great benefit in terms of serving web-pages.

Stuart Langridge[120] talks at some length about the benefits and deficits of using XHTML rather than HTML when using JavaScript. He makes an interesting distinction in terms of what is and isn't acceptable and notes that HTML and XHTML are both W3C standards and reside alongside each other, rather than XHTML being a replacement for HTML. This is debatable but it seems likely that both will be accepted for some time to come by all browsers. He says (p.xi):

> *In short, using XHTML right now provides very little in the way of benefits, but brings with it a fair few extra complications. HTML 4.01 Strict is just as valid as XHTML - XHTML did not replace HTML but sits alongside it. It's just as easy to validate an HTML 4.01 page as it is to validate XHTML page.*

Jeremy Keith[121] also suggests that the distinction between HTML and XHTML is a matter of choice, he says (p.149):

> *Personally, I like to use XHTML. It is stricter about the markup it allows, so it encourages me to write cleaner markup.*

---

120. Langridge S (2005) DHTML Utopia: Modern Web Design Using Javascript & DOM Victoria: sitepoint

121. Keith J (2005) DOM Scripting Web Design with JavaScript and the Document Object Model Berkeley: Apress

To some extents the conversion to XHTML in this iteration was simple an academic exercise in order assess whether or not it offered anything over HTML as a delivery platform, in any case the page validates when checked under HTML 4.01 and XHTML 1.0.

The move to SQLite was to a great extent prompted by the development environment used. Because the application was being jointly developed on a Windows XP machine, a Mac running OS X Version 10.3.9 and a Linux server it became arduous to keep the MySQL databases concurrent, especially when the machines did not have access to the Internet at some times; meaning a single database couldn't be accessed. The use of the SQLite database meant that the application could be transferred between machines as a simple flat file and made development much easier. SQLite also offered significant increases in speed because of the relatively small size of the data set.

The move towards SQLite by PHP is controversial yet it must be noted that PHP 5 still supports the MySQL RDBMS but that the support must be specified explicitly rather than the support being implicit as in previous versions[122]. SQLite uses a single file database model which means that the complete database must be locked when an update or insert is made. This can cause serious performance issues if the database is updated frequently[123]. In the

---

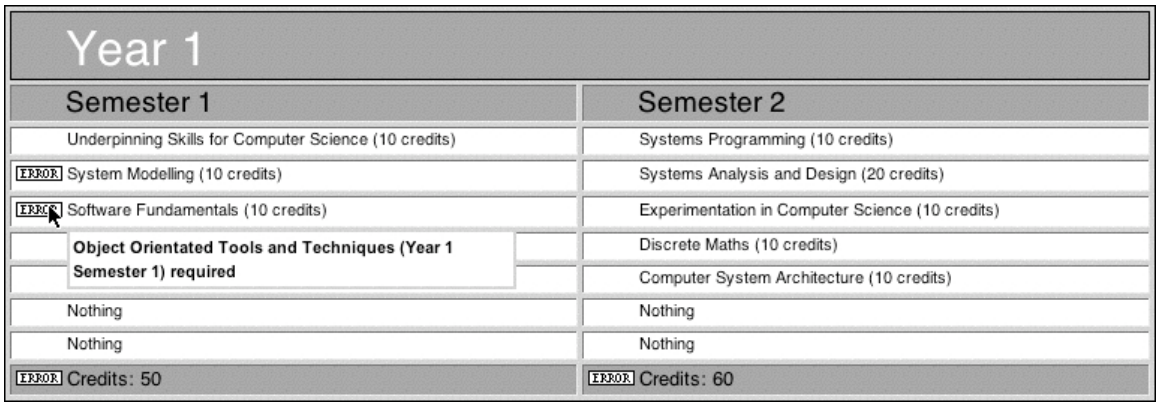
122. SQLite was available to PHP to some cersions of 4 but needed to be specifically activated

123. According to one of the developers of the PunBB discussion board software as noted here http://forum.textdrive.com/viewtopic.php?pid=59576 in 2005

context of the application development takes place prior to the module choices needing to be made, with usage of the application being generally low except for those periods when students are required to make module choices. It is even possible to automate the process of copying relevant data from the main administrative database to an SQLite file using a simple PHP script which would allow for greater abstraction - and thus security - of the data.

4.4 sqlite_validation_xhtml

This follows the validation iteration in that all the modules are first placed into an array, this is significant in that the processing of the page is simplified. In the secondpage iterations an extra tier of processing was required before the variables were placed in the array.

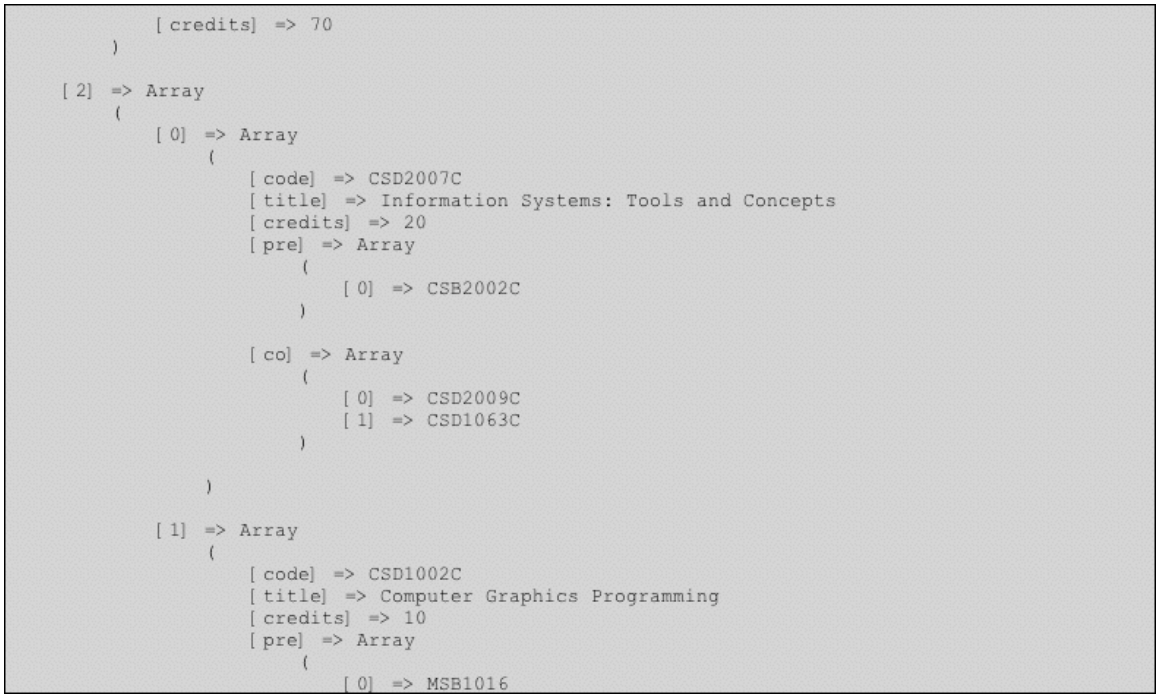


Picture 20: Highlighed error on sqlite_validation_xhtml

The stylistic issues noted in the validation iteration were addressed in this version with the error messages being placed to the left of the selected modules. This is illustrated in the picture titled "Highlighed error on sqlite_validation_xhtml", also displayed in this picture is the updated mechanism for directing the user in ways of correcting the error; instead of simply displaying the module code its title and location are displayed.

Both this iteration and its predecessor placed all the variables into an array and then, if the variable was a module code it was checked to see if it was repeated, if it is unique it discovers what co- and pre-requisites are required and

places these in the array as well as placing the credit score for that module in the array. It is possible to show this by using the PHP function print_r()[124] and the picture "print_r view" shows the array when the module "Information Systems: Tools and Concepts" has been selected in the third semester. Just to the top of the screen grab is the element "[credits] => 70" which indicates that all the modules from the second semester have been selected and that there are too many credits for that semester, this will flag an error on the page.



Picture 21: print_r view

The picture "print_r view" is from validation but is equally relevant to

124. "print_r() displays information about a variable in a way that's readable by humans. If given a string, integer or float, the value itself will be printed. If given an array, values will be presented in a format that shows keys and elements." The PHP Group (2006) print_r [WWW] http://uk2.php.net/print_r

sqlite_validation_xhtml.

Once the array is filled the PHP works through the array and checks to see if requisites are filled, if they are not an error message is posted onto the page, at the end of the semester the credits are displayed and if there are too many or too few a further error message is displayed.

5.0 The Final Iteration

This iteration was developed solely on PHP version 5 and uses the SQLite database as illustrated in the figure titled "The final iteration".

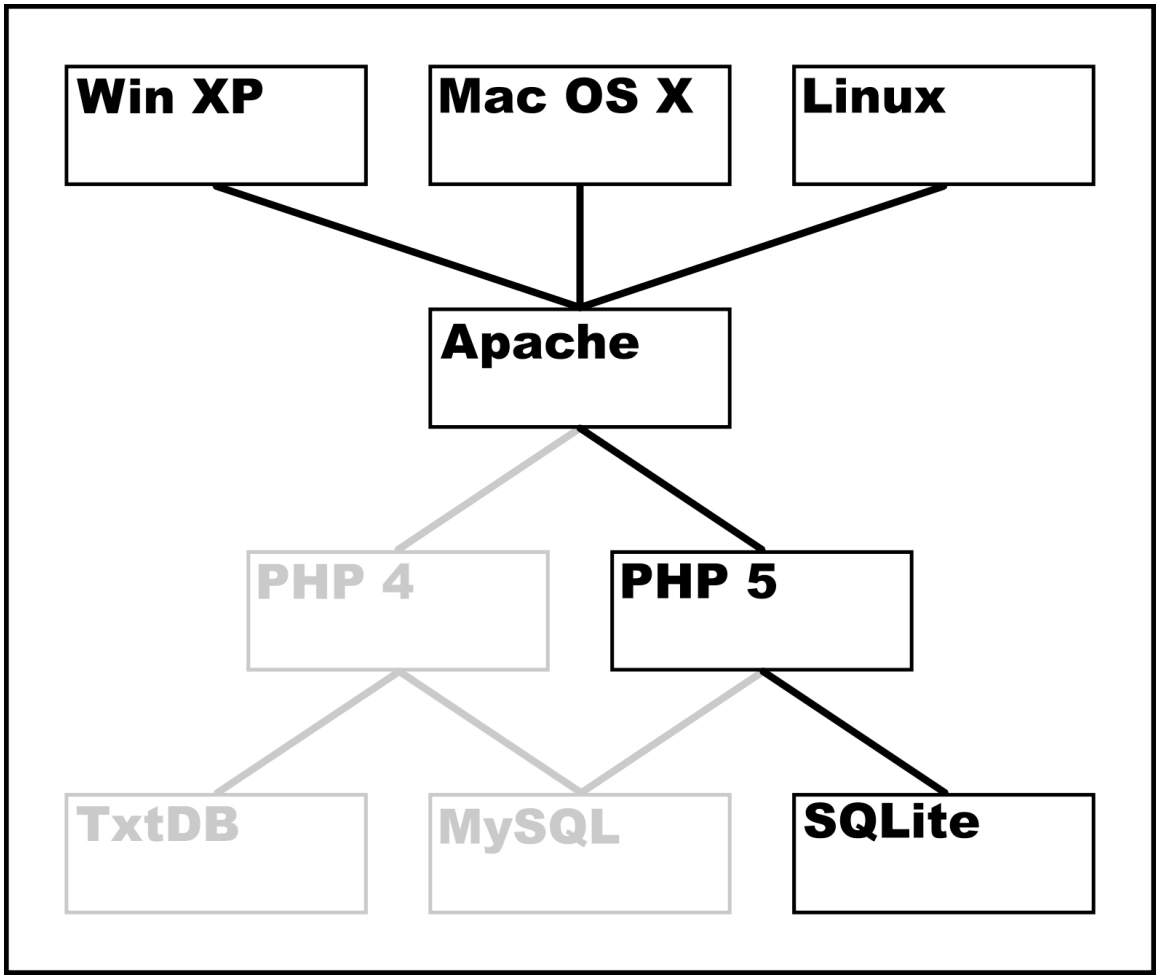


Figure 9: The final iteration

This iteration took the best design elements of the previous iterations and added to them. This focus on issues of appearance might at first seem extraneous but, for instance, whereas the developer of a traditional application

is able to check the results of their efforts from the command line the developer of web-based applications is constantly faced with a browser window when not working on mark-up or other code of one type or another.

Picture 22: Rounded corners

Steve Krug[125] has talked about how designers of web-sites have a

tendency to become disheartened when they come to understand that users rarely appreciate the time and effort invested in them, but it might be argued that designers strive to make sites aesthetically pleasing as they have to spend so much time and effort on them.

The repetitive process associated with the development of the application meant that the developer had to spend long periods of time interfacing with the screen checking each new element of functionality. Aesthetic concerns became particularly important in such straits. This is what led to the rounded corners of the final iteration as can be seen in the picture titled "Rounded corners". The application maintains its aspect ratio of being one screen width wide, even on the smallest of screens[126], while still not being quite three screen heights tall.

Rather than the two page model used in the previous iterations this page made use of the XHR object to check each and every selection made. There are three events in a select which can be used to trigger JavaScript and in this instance the onChange event was used rather than the onBlur or onFocus events[127]. onFocus would have been wholly inappropriate as the intention was to check the submission made by the user rather than the users location. Testing

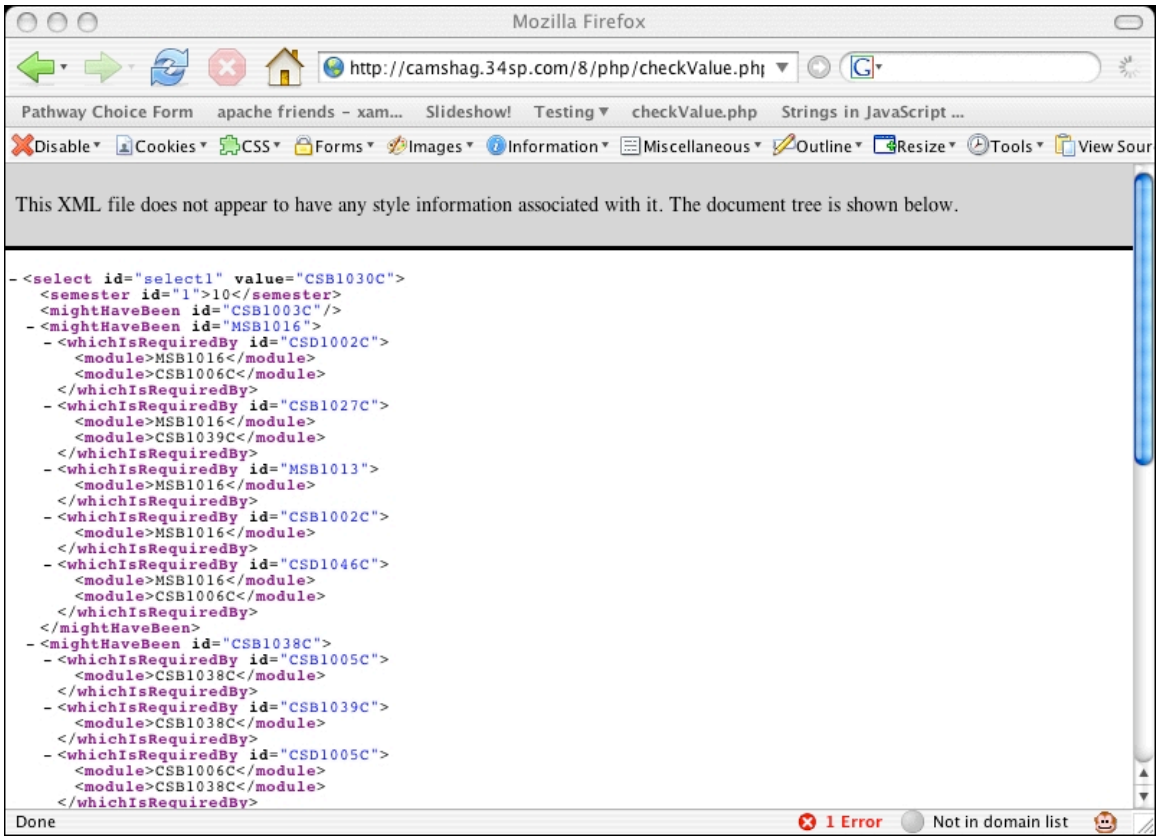
---

125. Krug S (2006) Don't Make Me Think! A common Sense Approach to Web Usability 2nd Ed. Berkeley: New Riders

126. This was accomplished by the use of tables, though tables are falling out of fashion generally they were particularly important in this application for reasons which will be explored more fully.

127. There is an excellent discussion of JavaScript event on The Computer Technology Documentation Project at http://www.comptechdoc.org/independent/web/cgi/javamanual/javaevents.html

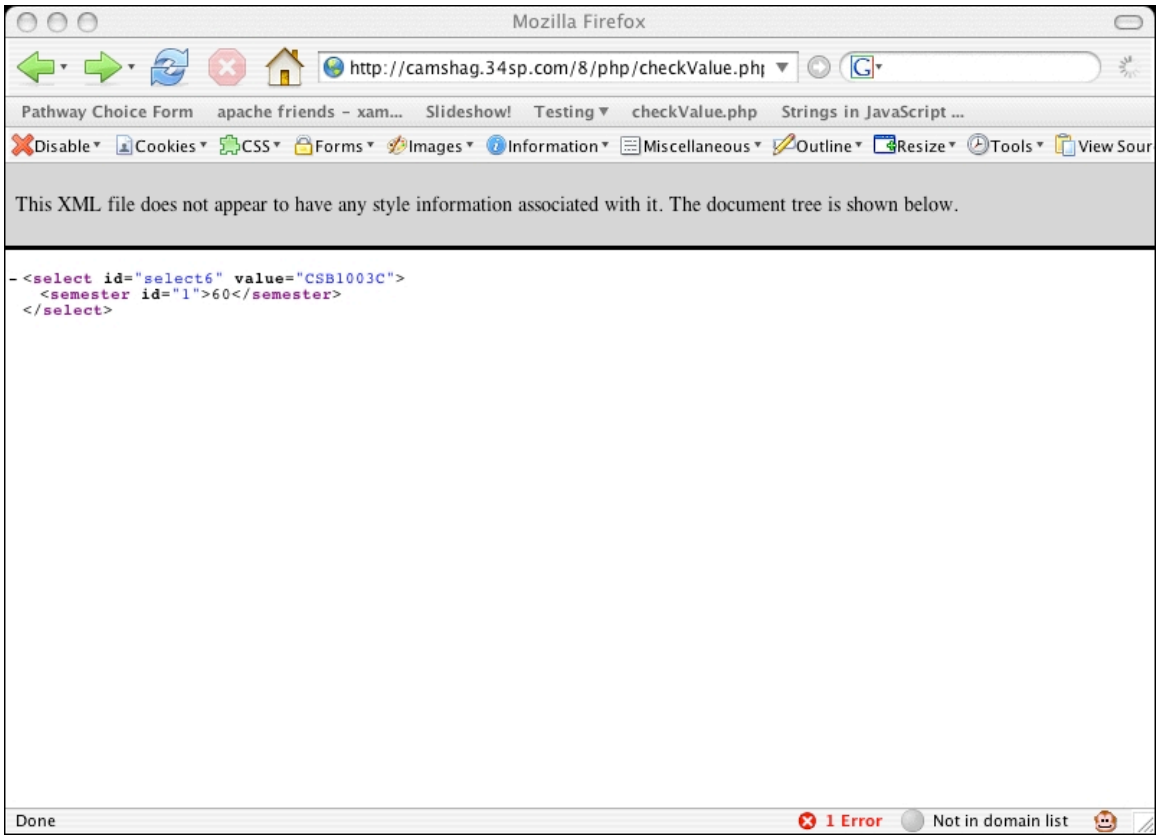
was carried out on the onBlur event but the selection was not checked until after the user had moved away whereas the onChange event allowed the user to remain at the same select and test for themselves the result of different selects. The sequence of events was, in the case of onBlur - click arrow beside select, choose option then move to next select if necessary. The option just selected was only checked after the move.



Picture 23: First Ajax request

As can be seen from the mark-up (Appendix 3.1) PHP was not used to create the page, this was due to time factors solely as it would have been possible to create the page in the same way as the second iterations (i.e. by

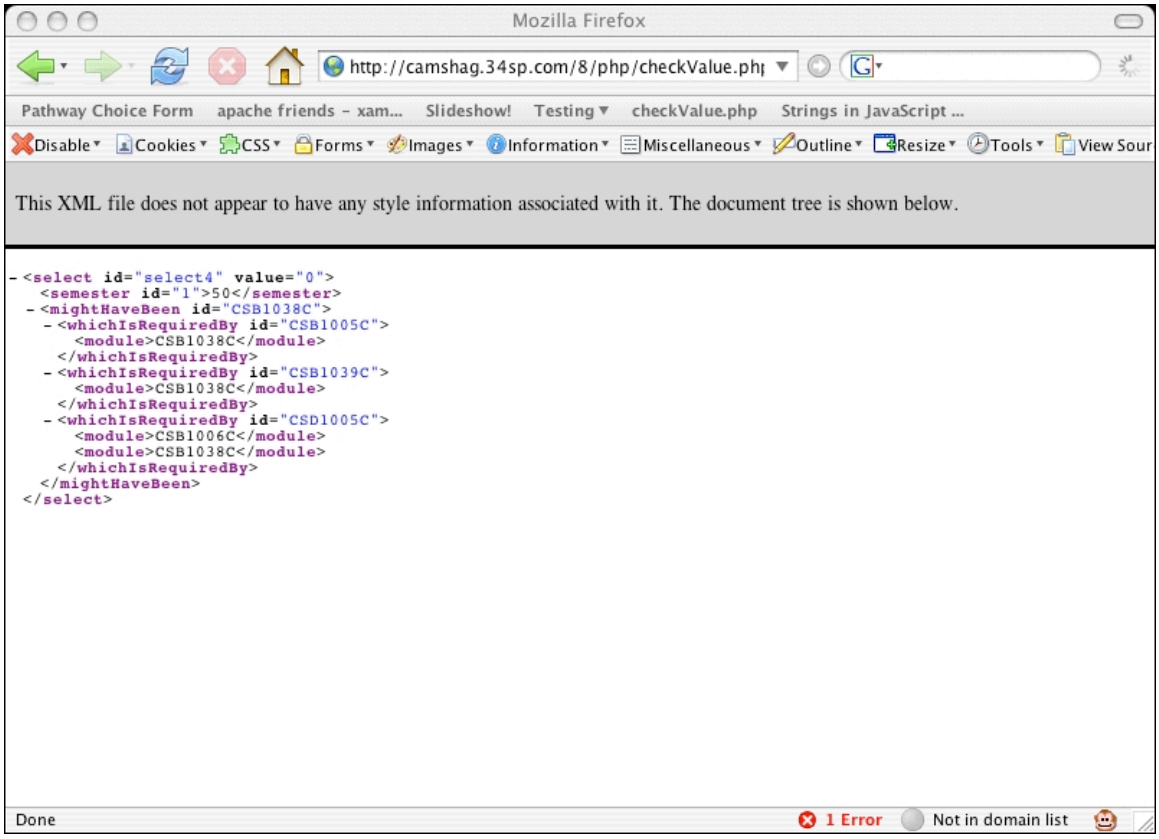
using PHP to create the mark-up by using loops) but because this application only deals with one possible degree course it was befitting to simply copy and paste the relevant sections of code repeatedly. This does unfortunately mean that the source code is 963 lines long.



Picture 24: Second Ajax request

Once a module is selected the JavaScript (Appendix 3.6) is sent the value selected and the ID of the select and then removes any errors from beside the select using the unFlagError function, it checks to see if it has been selected beforehand, then gathers the data required for the Ajax request.

The JavaScript collates the other selected values for the semester and the other possible options for that select and separates these values with colons. These values are then passed to the XHR object and sent to the server where PHP processes the values by way of accessing the SQLite database[128]. Should the first module in the first select of the first semester be chosen the Ajax request can be repeated from the address bar of the browser to produce the picture titled "First Ajax request".



Picture 25: Third Ajax request

128. Interestingly, the distinction between co- and pre-requisites is removed, it is arguable that the distinction was unnecessary in previous iterations as well but it did aid in the error messages in the secondpage iterations

This returned XML is perhaps a little too complex to analyse though, a more simplistic return is generated by selecting all the possible modules in order in the first semester (as can be seen the the picture titled "Second Ajax request"), this does not reveal enough information to warrant analysis; if all modules bar "Object Orientated Tools and Techniques" in the first semester are chosen the picture titled "Third Ajax request" is obtained and this is suitable for greater examination.

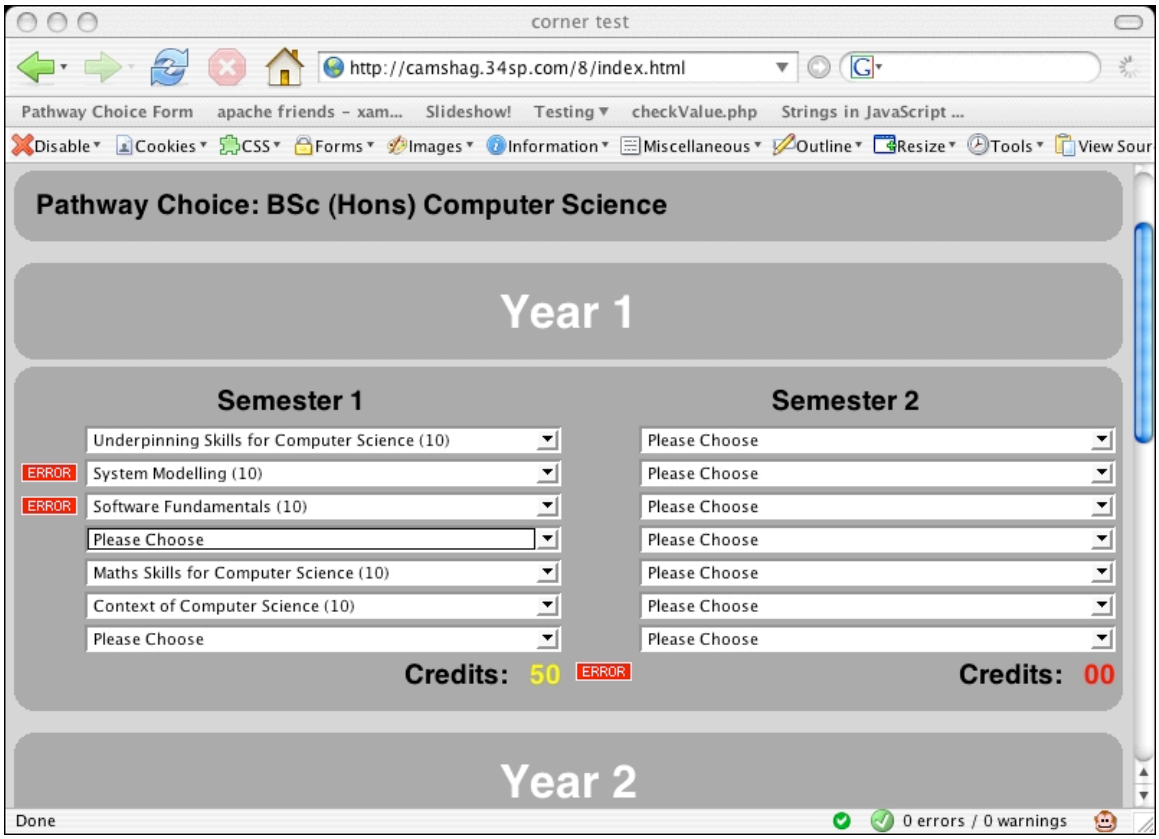
From it we can gather that the select with the ID of "select4" has been used and that the user chose the option[129] which has a value of "0", it can be seen that the select belongs to the semester with the ID of "1" and that the previous choices for that semester total "50" credits. The only other possible module code which could have been selected[130] is "CSB1038C" and that module is required by the modules[131] with the ID of "CSB1005C", "CSB1039C" and "CSD1005C"[132]. The first two require "CSB1038C" alone whereas "CSD1005C" also needs "CSB1006C".

The JavaScript parses the DOM of the returned XML and acts on the body of the page as a result of that reading. The source code is well documented[133] and use of the flagError function dictated the use of tables as a

---

129. "Please Choose" is this option, the default for all the selects upon loading
130. mightHaveBeen
131. whichIsRequiredBy
132. Please note the difference in the third letter

layout technique as the script needs to navigate the DOM in order to find the correct place for the error image. In the example analysed above it would be necessary to flag the modules "CSB1005C", "CSB1039C" and "CSD1005C" with an error if they were selected. This was indeed the case and the result is visible in the image titled "Third Ajax request result".



Picture 26: Third Ajax request result

The inclusion of the extra variables in the XHR object request allows for the checking of the total number of credits for a semester and also the flagging

133. If a little informally

of errors which might result from a value being changed. To clarify, the select has an error flagged if it has requisites which are not filled. Other selects did not have their errors flagged if a requisite is removed by the user selecting a different module until this functionality was added.

The error tooltips use a different mechanism as those which use the DHTML Tooltips are only set on the page as it is loaded and errors in the final application had to be added and removed over the course of its usage. dhtmltooltips.js (Appendix 3.7) offered this functionality and a similar appearance to the previous tooltip script.

IV. Testing and Analysis

Development was aided considerably by using the Firefox extension called FireBug[134] developed by Joe Hewitt[135], more specifically it's XMLHttpRequest Spy allowed for the viewing of Ajax requests and subsequent replies from the server. This tool became available during development, it was envisaged that the XMLHttpRequest Debugging User Script[136] for Greasemonkey[137] would be used but the benefits of FireBug led to its adoption instead.

The criticisms of JavaScript noted in the literature review[138] meant that testing was continuous during the development, should an area of functionality not be implemented in a sub-iteration then the cause was usually obvious and thus correctable. Testing on different browsers[139] also aided this process and led to the use of whitespace.js (Appendix 3.5) as the Firefox browser

---

134. https://addons.mozilla.org/extensions/moreinfo.php?application=firefox&id=1843

135. http://www.joehewitt.com/

136. Written by Julien Couvreur and available at http://blog.monstuff.com/archives/images/XMLHttpRequestDebugging.v1.0.user.js

137. "Greasemonkey is a Firefox extension that allows you to add bits of JavaScript, known as *user scripts*, to any Web page to change its behaviour. Greasemonkey is similar to other Firefox extentions that, for example, allow you to change a page's CSS rules. Greasemonkey is really just a proxy for the users scripts that are responsible for doing the real work; you can easily download and install it from its home page at greasemonkey.mozdev.org." Asleson R & Schutta NT (2006) Foundations of Ajax Berkely: Apress

138. i.e. The lack of an IDE

139. Because development was carried out on Mac OS X and Windows XP most browsers were available, including the Beta of IE7 and Safari

implements the DOM in a different way to other browsers[140].

In terms of the development of the final iteration areas of functionality were added to the functions.js file (Appendix 3.6) - and subsequently to the checkValue.php (Appendix 3.3) file: Initially the code selected and the ID of the select were transmitted, with the requisites for the code and the credits awarded being returned. Next the other modules for the semester were sent and the semester credit value returned along with the initial variables. These two areas allowed for the errors associated with requisites not being filled and insufficient or too many credits to be flagged. That only left errors caused by the user deselecting a module which is a requisite for another, selected, module to be factored. This required the other values that it would be possible to select to be added to the request. Continuous testing of this functionality on all browsers was thus difficult to formalise within a traditional testing regime.

The finished application was user tested by 4 people and the general feeling was that the final iteration was by far quicker to use than the two page model of the previous iterations[141], this is a small sample size so further user testing was deemed to be beneficial.

---

140. It is arguable that the Firefox implementation is the correct implementation but it is at variance and means that the usual method of navigating the DOM needed to be amended, thankfully the solution is cross-browser compatible and thus browser detection and code-forking is not required

141. Of the four user testers two timed their interaction with the application. The first spent 15 minutes working on the two page model to get a valid degree and only 5 minutes on the final iteration to reach the same degree of validity. The second user took sixteen and seven minutes respectively. Anecdotally, it is reported that such a process generally takes much longer without the application (i.e. Using pen and paper)

Testing the application though WebXACT[142] brings up eleven Priority 1 checkpoint warnings:

> *"1.1 If an image conveys important information beyond what is in its alternative text, provide an extended description. 33*
>
> *2.1 If you use color to convey information, make sure the information is also represented another way. 33*
>
> *4.1 Identify any changes in the document's language.*
>
> *5.1 If this is a data table (not used for layout only), identify headers for the table rows and columns. 7*
>
> *5.2 If a table has two or more rows or columns that serve as headers, use structural markup to identify their hierarchy and relationship. 7*
>
> *6.1 If style sheets are ignored or unsupported, ensure that pages are still readable and usable.*
>
> *6.3 Provide alternative content for each SCRIPT that conveys information or functionality.*
>
> *6.3 Make sure pages are still usable if programmatic objects do not function. 4*
>
> *7.1 Make sure that the page does not cause the screen to flicker rapidly.*
>
> *8.1 Provide accessible alternatives to the information in scripts, applets, or objects. 4*
>
> *14.1 Use the simplest and most straightforward language that is possible."*

The points associated with appearance and layout (1.1, 2.1, 4.1, 5.1, 5.2, 7.1 and 14.1) are irrelevant as the images and tables were merely decorative and structural respectively. Of the point associated with language the page does not have any changes to the language but because it is impossible to scan for

142. "WebXACT is a free online service that lets you test single pages of web content for quality, accessibility, and privacy issues.", it is provided by Watchfire and is located at: http://webxact.watchfire.com/

changes the warning is flagged on all submitted pages. Of the remaining warnings the application relies upon the user using at least a modern, if not the most recent, version of a browser with JavaScript enabled in order for it to work.

While the W3C recommendation[143] points out that making a site accessible benefits all users not just those with disabilities the application relies on some techniques which contraindicate screen-reader software such as JAWS[144], this will be discussed further in the conclusion.

Checking the site using Section 508[145] using WebXACT produces:

> "*This page complies with all of the automatic checkpoints of the Section 508 Accessibility Guidelines. However, it does not comply with all of the manual checkpoints, and requires manual verification.*"

Those warnings are similar in scope to the points noted above and can be addressed in the same way.

Paul Canning suggested that the application be submitted for review to the Accessify Forum Site Critique[146] section and this garnered some interesting suggestions. Of particular interest was the suggestion of redux[147] associated with replacing the drop-down mechanism with check-boxes. This would reduce

---

143. http://www.w3.org/TR/WAI-WEBCONTENT/

144. http://www.freedomscientific.com/fs_products/software_jaws.asp

145. "Section 508 (the 1998 Amendment to Section 508 of the Rehabilitation Act) requires the federal government to make all goods and services—including Web pages—fully accessible. It identifies specific standards for Internet and Web accessibility, which are often used as a basis for evaluating whether or not Web sites meet accessibility requirements." University of Maryland University College (2005) ADE Glossary [WWW] http://www.umuc.edu/ade/glossary.html

146. Accessify (2006) Web Application Accessibility [WWW] http://www.accessifyforum.com/viewtopic.php?p=35702

147. Patrick H. Lauke, webmaster at the University of Salford

markup as well as making the application less repetitive to users with a visual disability who employ screen-readers, though it would impact the present structural symmetry of the application, this might be countered with alternative styling particularly as the use of HTML tables for the structuring of page content is coming out of vogue.

Cerbera[148], also via the Accessify Forum post, pointed out technical ways of structuring a page using the check-box approach rather than drop-down lists. He noted that such changes would not only serve to aid those with disabilities but might also speed the non-disabled users interaction with the application.

148. Ben Millard, a freelance website developer

V. Summary and Conclusions

1.0 Introduction

A number of issues were raised during the development of the application; these will be covered in greater depth in the sections below.

2.0 Accessibility

Universities are obliged to follow accessibility guidelines when developing web pages and in this instance the application relies on functionality that can interfere with full compliance with these guidelines, AbilityNet recently announced the introduction of a new standard called PAS 78[149]. They say[150]:

> *"Since 1999 under the DDA all UK organisations with websites have been legally obliged to make them disabled-friendly. The new PAS 78 guidelines are a practical guide which will help site owners and developers stay on the right side of the law."*[151]

There is further debate about accessibility issues in regards to assistive technologies for disabled people on the Accessify Forum[152]. Of particular interest is this comment from Isofarro[153]:

> *"AJAX itself is not inaccessible. The problems revolve about how to dynamically update a document in a way that assistive technology can inform, or let its owner know an update has occurred or describing what has changed in a document."*

There is an excellent blog entry on accessibility issues and Ajax from Matt May[154] (Director of Technology at Blue Flavor[155], an experience design

---

149. There is no online tool presently to check conformance with PAS 78 so this was not tested for
150. AbilityNet (2006) PAS 78 Launches. [WWW] http://www.abilitynet.org.uk/content/oneoffs/pas78.htm
151. A thorough discussion of the development and ramifications of PAS 78 is written by Bruce Lawson on his blog at: http://www.brucelawson.co.uk/index.php/2006/pas-78-guide-to-good-practice-in-commissioning-accessible-websites/
152. http://www.accessifyforum.com/
153. Isofarro (2005) AJAX [WWW] http://www.accessifyforum.com/viewtopic.php?t=3516
154. http://www.bestkungfu.com/archive/date/2005/03/ajaxessibility/
155. http://www.blueflavor.com/

consultancy), which points out how screen-readers, for instance, do not take note of changes made to the page as a result of Ajax requests.

The application relies upon modern browsers and when accessed by older versions it does not presently offer any fall back methods. It is feasible to add functionality to a standard site when and where corresponding browser technology is enabled. In this way it would be possible to have an approach similar to the second iteration as standard. But where Ajax techniques are possible the script on the page would alter the DOM so that the two page model would be replaced with the final iteration. This alteration would be imperceptible to the user but would allow users of more recent browsers to have a more user-friendly application from which to choose their academic course.

Rather than following the ideal of having functionality that degrades gracefully this approach would rather provide for incremental enhancement. This would also serve to address the aspect of accessibility. The original submit button and validation page could be retained to allow disabled users as well as those with older browsers to still have the same functionality, if through a slightly different interface.

Paul Canning, the Web Development Officer for Cambridge City Council, notes[156]:

> *"The other thing I would say is don't lose touch with common sense - you may not be able to do everything technically correct but if you*

156. Canning P (2006) web application accessability [e-mail] No cache of message

explain limitations and offer an alternative route to the information then you're communicating with users and they appreciate that."

The fall back to the traditional techniques would enable this approach and satisfy statutory requirements as well. This advice is mirrored by brothercake[157], again on the Accessify Forum, when he notes[158]:

"*I appreciate there's a value for applications that would otherwise not be possible - you could quite reasonably argue that loss of accessibility is less critical if it's for something that would otherwise not be possible for anyone.*"

How proceeding from this point of view might be affected by legislation is not currently known. We might be in a position where if something isn't possible for all it will be prohibited for the majority, though it is also arguable that optimising sites for the use of people with disabilities aids all users[159].

Implementing incremental enhancements would address some of these issues and the retention of the submit button and validation page would not detract from the application as a whole but might provide for the facility to print out the selections or submit them to an academic supervisor via email, for instance. Keith[160] calls such incremental enhancements by a different term, that of progressive enhancement. He says (p.146):

"*The principle of* ***progressive enhancement*** *is based on the idea*

---

157. James Edwards, a freelance web developer

158. brothercake (2005) Accessibility of applications using XMLHttpRequest [WWW] http://www.accessifyforum.com/viewtopic.php?t=2660

159. Disability Rights Commission (2004) The Web: access and inclusion for disabled people London: DRC

160. Keith J (2005) DOM Scripting Web Design with JavaScript and the Document Object Model Berkeley: Apress

that you should begin with your core content. The content is structured using markup. The marked-up content is then enhanced. The enhancement might be stylistic, using CSS, or it might be behavioural using DOM Scripting."

3.0 Security

Something which is more and more relevant when discussing Ajax techniques is the issue of security. The Guardian recently ran a story[161] about the first security exploit to employ Ajax, the so called Samy MySpace Worm, but this was by no means the first concern raised about the security implications of Ajax. As Lorin Thwaits notes on his blog[162]:

> *"AJAX is a very straightforward way to increase the dynamic feel of any site. But we're talking about Javascript calling almost directly into server-side code, so there are some big security concerns to be worried about with this architecture. What was once one door into the server, a URL to retrieve the web page, is now accompanied by an additional door for every method provided by the server. Retrieving fairly raw data with a simple call into these methods means the code servicing those calls is likely to be closer to the database. So altogether there is more risk."*

He goes on to detail possible exploits but it is arguable that such exploits were already possible prior to the use of Ajax techniques but just not so easily accomplished using such a "simple" programming language like JavaScript. He suggests that thorough precautions be put in place to ensure that it is not possible to inject malicious data into data passed to the server using the XHR object. Further, should such precautions be circumvented measures should be taken to strengthen the server. Quinn Norton in The Guardian notes that the most safe approach would be to disable JavaScript in the browser but notes

---

161. Norton Q (2006) Ajax prepares for battle on the dark side The Guardian (Technology Guardian supplement) March 9th 2006 p.3

162. Thwaits L (2006) AJAX as an attack vector [WWW] http://geekswithblogs.net/lorint/archive/2006/02/09/68810.aspx

that doing so might impact the usability of those sites which rely upon the technique.

Keeping two copies of the data, as discussed earlier, would negate such security concerns and it might be apt to delegate this process to something like a cron[163] job (or similar) which would copy all relevant data to a fresh SQLite file at a set time every night.

At a more fundamental level JavaScript itself has issues surrounding security. Crane, Pascarello and James say[164]:

> *"We describe code as mobile if it is stored on one machine and can transmit itself across the network to execute on another. The computer receiving the mobile code needs to consider whether it should trust the sender of the code, particularly over the public Internet. To what system resources should it grant access?"*

They do go on to note that browser vendors have been conscious of this fact for a number of years and that JavaScript has little or no access to the computer's resources as it runs within a sandbox in much the same way as Java does.

---

163. "Cron is the name of program that enables unix users to execute commands or scripts (groups of commands) automatically at a specified time/date." cogNiTioN (1999) Intro to cron [WWW] http://www.unixgeeks.org/security/newbie/unix/cron-1.html

164. Crane D, Pascarello E & James D (2006) Ajax IN ACTION Greenwich: Manning

4.0 Other Issues

While the application provides the functionality for a student to correctly choose an academic program with all requisites and credit requirements fulfilled it does not take into consideration other issues which might be important to the student.

Should the student have other commitments it might be appropriate for them to be able to limit their choices in terms of time-tabling. That is to say that future applications should allow for a prospective student to choose just those modules which would allow them to drop-off and collect their children from school, for example.

5.0 Summary

The functionality of the application would benefit significantly from the provision of other criteria for module choice as outlined above. Its cultural bias might also be negated by providing the user with the option to alter the structure of the page presented to them, thus allowing them to place the years of the degree horizontally or vertically. The adoption of check-boxes rather than drop-down lists would enable users with disabilities to access the application as well as speed up the process of choosing modules and ensure that the institution would be in compliance not only with best practices but also legislation. Further analysis of the PAS 78[165] document is required in regards to this.

Despite the issues noted above the application does provide the functionality required and there is promise that such techniques could surpass requirements in future to allow for an enabling user interface.

Coding an application for the Internet has proved to be at least as problematic as for any other context. But the lack of tools has been more than made up for by the sheer quantity of help, advice and information available online.

---

165. British Standards Institution (2006) Publicly Available Specification (PAS):78 Guide to good practice in commissioning accessible websites London: BSI

VI. References

AbilityNet (2006) PAS 78 Launches [WWW] http://www.abilitynet.org.uk/content/oneoffs/pas78.htm

Accessify (2006) Web Application Accessibility [WWW] http://www.accessifyforum.com/viewtopic.php?p=35702

Agarwal-Hollands U & Andrews R (2001) From Scroll... to Codex... and Back Again Education, Communication & Information 1(1), pp.59-73

Asleson R & Schutta NT (2005) Foundations of Ajax Berkeley: Apress

Baran P (1964) Rand Memoranda on Distributed Communication [WWW] http://www.rand.org/publications/RM/RM3420/

Bennahum D (Unknown) J.C.R. Licklider (1915-1990) [WWW] http://memex.org/licklider.html

Berners-Lee T (no date) The WorldWideWeb browser [WWW] http://www.w3.org/People/Berners-Lee/WorldWideWeb.html

Berners-Lee T (1999) Weaving the Web The Past, Present and Future of the World Wide Web by its Inventor London: Orion Business Books

Bersvendson A (2005) Who created CSS? CSS Early History [WWW] http://virtuelvis.com/archives/2005/01/css-history

British Standards Institution (2006) Publicly Available Specification (PAS):78 Guide to good practice in commissioning accessible websites London: BSI

brothercake (2005) Accessibility of applications using XMLHttpRequest [WWW] http://www.accessifyforum.com/viewtopic.php?t=2660

Burrell I (2005) Where do we go from here? The Independent Media Weekly supplement March 20th 2006 pp.4-7

Bush V (1945) As We May Think Atlantic Monthly

bytown internet (no date) glossary [WWW] http://www.bytowninternet.com/glossary

c-worker (2005) Manual [WWW] http://www.c-worker.ch/txtdbapi/manual_html_eng/intro.html

Callahan E (2005) Cultural similarities and differences in the design of university websites Journal of Computer-Mediated Communication 11(1) http://jcmc.indiana.edu/vol11/issue1/callahan.html

Canning P (2006) web application accessability [e-mail] No cache of message

Champeon S (2001) JavaScript: How Did We Get Here? [WWW] http://www.oreillynet.com/pub/a/javascript/2001/04/06/js_history.html

Chapman S (no date) A Brief History of Javascript [WWW] http://javascript.about.com/od/reference/a/history.htm

Clegg C, Axtell C, Damodaran L, Farbey B, Hull R, Lloyd-Jones R, Nicholls J, Sell R & Tomlinson C (1997) Information technology: a study of performance and the role of human and organizational factors Ergonomics 40(9) pp.851-871

cogNiTioN (1999) Intro to cron [WWW] http://www.unixgeeks.org/security/newbie/unix/cron-1.html

Crane D, Pascarello E & James D (2006) Ajax IN ACTION Greenwich: Manning

Davidson J (1988) An Introduction to TCP/IP New York: Springer-Verlag

Disability Rights Commission (2004) The Web: access and inclusion for disabled people London: DRC

Dutta S (2006) Native XMLHTTPRequest object [WWW] http://blogs.msdn.com/ie/archive/2006/01/23/516393.aspx

ECMA (no date) TC39 - Programming and Scripting languages [WWW] http://www.ecma-international.org/memento/TC39.htm

Eisenhower D (1960) Farewell Radio and Television Address to the American People IN Public Papers of the Presidents Washington: GPO

Griffin S (2000) Internet Pioneers [WWW] http://www.ibiblio.org/pioneers/index.html

Garber S (2003) Sputnik and The Dawn of the Space Age [WWW] http://www.hq.nasa.gov/office/pao/History/sputnik/

Garrett JJ (2005) Ajax: A New Approach to Web Applications [WWW] http://www.adaptivepath.com/publications/essays/archives/000385.php

Goldfarb C (1990) SGML HISTORY [WWW] http://xml.coverpages.org/sgmlhist0.html

Goldfarb C & Prescod P (2001) THE XML HANDBOOK 3rd Ed. London: Prentice Hall

Gray J (2005) A Conversation with Tim Bray ACM Queue 3(1) [WWW] http://www.acmqueue.com/modules.php?name=Content&pa=showpage&pid=282&page=1

Griffin S (2000) Internet Pioneers [WWW] http://www.ibiblio.org/pioneers/index.html

Isofarro (2005) AJAX [WWW] http://www.accessifyforum.com/viewtopic.php?t=3516

Jacobs I (2005) About the World Wide Web Consortium (W3C) [WWW] http://www.w3.org/Consortium/

Keith J (2005) DOM Scripting Web Design with JavaScript and the Document Object Model Berkeley: Apress

Kleinrock L (1961) Information Flow in Large Communication Nets PhD. Thesis Proposal, Massachusetts Institute of Technology

Kleinrock L (1964) Communication Nets: Stochastic Message Flow and Design Columbus: McGraw-Hill

Kock P (2004) A history of browsers [WWW] http://www.quirksmode.org/index.html?/browsers/history.html

Krug S (2006) Don't Make Me Think! A common Sense Approach to Web Usability 2nd ed. Berkeley: New Riders

Langridge S (2005) DHTML Utopia: Modern Web Design Using Javascript & DOM Victoria: sitepoint

Leiner B & Cerf V *et al* (2003) A Brief History of the Internet Version 3.32 [WWW] http://www.isoc.org/internet/history/brief.shtml

Ley J (2006) IceBrowser and XMLHTTP Request [WWW] http://jibbering.com/blog/?p=189

Ley J (2006) Using the XML HTTP Request object [WWW]http://jibbering.com/2002/4/httprequest.html

Lipson JT & Fisher KM (1983) Technology and the classroom: Promise or threat? Theory into practice 22 pp.253-259

Mahemoff M (2005) AJAX Web Architecture podcast radio program [WWW] AJAX Web Architecture

mozilla.org (2004) About the Mozilla Foundation [WWW] http://www.mozilla.org/foundation/

MSDN (2006) JScript .NET Conditional Compilation [WWW] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/jscript7/html/jsConditionalCompilation.asp

Müller-Prove M (2005) 2 Hypertext [WWW] http://www.mprove.de/diplom/text/2_hypertext.html

Netcraft (2006) Netcraft: Web Server Survey Archives [WWW] http://news.netcraft.com/archives/web_server_survey.html

Nielson J (2000) Designing Web Usability: The Practice of Simplicity Indianapolis: New Riders

Norton Q (2006) Ajax prepares for battle on the dark side The Guardian

(Technology Guardian supplement) March 9th 2006 p.3

Olsina L, Lafuente GJ, Godoy D & Rossi G (1999) Assessing the Quality of Academic Websites: a Case Study New Review of Hypermedia and Multimedia Journal 5 pp.81-103

PBS Online (1998) Nerds 2.0.1. Timeline [WWW] http://www.pbs.org/opb/nerds2.0.1/timeline/index.html

Roberts L (1967) Multiple computer networks and intercomputer communication Proceedings of the first ACM symposium on Operating System Principles

Sayre R (2005) Ever Wonder Why It's Called "XMLHTTPRequest"? [WWW] http://www.franklinmint.fm/blog/archives/000294.html

Swann C (2004) WEB DESIGN'S EFFECT ON PERCEIVED CREDIBILITY OF ONLINE NEWS STORIES MASTER OF ARTS IN MASS COMMUNICATION UNIVERSITY OF FLORIDA

The Apache Software foundation (2005) Welcome! - The Apache HTTP Server Project [WWW] http://httpd.apache.org/

The PHP Group (2006) print_r [WWW] http://uk2.php.net/print_r

Thwaits L (2006) AJAX as an attack vector [WWW] http://geekswithblogs.net/lorint/archive/2006/02/09/68810.aspx

University of Maryland University College (2005) ADE Glossary [WWW] http://www.umuc.edu/ade/glossary.html

Välimäki M (2005) The Rise of Open Source Licensing: A Challenge to the Use of Intellectual Property in the Software Industry Helsinki: Turre Publishing

W3 (1993) World Wide Web [WWW] http://www.w3.org/History/19921103-hypertext/hypertext/WWW/TheProject.html

Waters R (2005) Byline The Financial Times, June 3rd 2005 p14

Webopedia (no date) ActiveX [WWW] http://www.webopedia.com/TERM/A/ActiveX.html

Wikipedia (2005) Bob Kahn [WWW] http://en.wikipedia.org/wiki/Bob_Kahn

Wikipedia (2005) Donald Davies [WWW] http://en.wikipedia.org/wiki/Donald_Davies

Wikipedia (2005) Hypertext [WWW] http://en.wikipedia.org/wiki/Hypertext

Wikipedia (2005) HyperTextTransfer Protocol [WWW] http://en.wikipedia.org/wiki/Http

Wikipedia (2005) Håkon Wium Lie [WWW] http://en.wikipedia.org/wiki/H%C3%A5kon_Wium_Lie

Wikipedia (2006) Internet Information Services [WWW] http://en.wikipedia.org/

wiki/Internet_Information_Services

Wikipedia (2005) Java applet [WWW] http://en.wikipedia.org/wiki/Java_applet

Wikipedia (2005) Jesse James Garrett [WWW] http://en.wikipedia.org/wiki/Jesse_James_Garrett

Wikipedia (2005) Leonard Kleinrock [WWW] http://en.wikipedia.org/wiki/Leonard_Kleinrock

Wikipedia (2006) Microsoft Developer Network [WWW] http://en.wikipedia.org/wiki/Msdn

Wikipedia (2006) Open Database Connectivity [WWW] http://en.wikipedia.org/wiki/Open_Database_Connectivity

Wikipedia (2005) Paul Baran [WWW] http://en.wikipedia.org/wiki/Paul_Baran

Wikipedia (2005) RAND Corporation [WWW] http://en.wikipedia.org/wiki/RAND_Corporation

Wikipedia (2005) Server-side scripting [WWW] http://en.wikipedia.org/wiki/Server-side_scripting

Williams R & Tollett J (2000) The Non-Designers Web Book 2nd ed. Berkeley: Peachpit Press

Zahas NC (2005) Professional JavaScript for Web Developers Indiana: Wrox

VII. Appendices

1.1 secondpage.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv=content-type content="text/html; charset=ISO-8859-1">
<?php
  include("txt-db-api/txt-db-api.php");
  include("resources/functions.php");
?>
    <title>
      secondpage.php
    </title>
    <link href="resources/hobbs.css" type="text/css" rel="stylesheet">
    <script type="text/javascript" src="resources/overview.js"></script>
  </head>
  <body>
    <h1>
      Module Choice Form
    </h1>
    <form name="myform" id="myform" action="thirdpagetxtdb.php" method="post">
      <table border="1" cellpadding="2" cellspacing="3">
        <tbody>
          <tr>
            <td class="title" colspan="6">
              Pathway Choice: BSc (Hons) Computer Science
            </td>
          </tr>
          <tr>
            <td class="year" colspan="2">Year 1</td>
            <td class="year" colspan="2">Year 2</td>
            <td class="year" colspan="2">Year 3</td>
          </tr>
          <tr>
            <td class="sem">Semester 1</td>
            <td class="sem">Semester 2</td>
            <td class="sem">Semester 1</td>
            <td class="sem">Semester 2</td>
            <td class="sem">Semester 1</td>
            <td class="sem">Semester 2</td>
          </tr>
<!-- 1st Row -->
          <tr>
            <td class="drop">
              <select name="module1">
                <option selected value="CSB1030C">
                  Underpinning Skills for Computer Science
                </option>
                <option value="S"> * Special * </option>
              </select>
            </td>
            <td class="drop">
              <select name="module7">
```

```
                <option selected value="CSB1002C">
                  Computer Systems Architecture
                </option>
                <option value="S"> * Special * </option>
              </select>
            </td>
            <td class="drop" rowspan="1" colspan="2">
              <select name="module12">
                <option selected value="CSD2009C">
                  Project Design and Implementation
                </option>
                <option value="S"> * Special * </option>
              </select>
            </td>
            <td class="drop" rowspan="2" colspan="1">
              <select name="module23">
                <option selected value="CSH2999C">
                  Undergraduate Project
                </option>
                <option value="S"> * Special * </option>
              </select>
            </td>
            <td class="drop" rowspan="2" colspan="1">
              <select name="module29">
                <option selected value="CSH2005C">Distributed Systems</option>
                <option value="S"> * Special * </option>
              </select>
            </td>
          </tr>
<!-- 2nd Row -->
          <tr>
            <td class="drop">
              <select name="module2">
                <option selected value="MSB1016">
                  Maths Skills for Computer Science
                </option>
                <option value="S"> * Special * </option>
              </select>
            </td>
            <td class="drop">
              <select name="module8">
                <option selected value="CSB1006C">Systems Programming</option>
                <option value="S"> * Special * </option>
              </select>
            </td>
            <td class="drop" rowspan="2" colspan="1">
              <select name="module13">
                <option selected value="CSD2007C">
                  Information Systems: Tools and Concepts
                </option>
                <option value="S"> * Special * </option>
              </select>
            </td>
            <td class="drop" rowspan="2" colspan="1">
              <select name="module18">
                <option selected value="CSD2003C">
                  Computer Networks and Communications
                </option>
                <option value="S"> * Special * </option>
              </select>
```

```
            </td>
          </tr>
<!-- 3rd Row -->
          <tr>
            <td class="drop">
              <select name="module3">
                <option selected value="CSB1039C">
                  Software Fundamentals
                </option>
                <option value="S"> * Special * </option>
              </select>
            </td>
            <td class="drop" rowspan="2" colspan="1">
              <select name="module9">
                <option selected value="CSB2002C">
                  Systems Analysis and Design
                </option>
                <option value="S"> * Special * </option>
              </select>
            </td>
            <td class="drop">
              <select name="module24">
                <option selected value="CSH1005C">
                  Computing and Society
                </option>
                <option value="S"> * Special * </option>
              </select>
            </td>
            <td class="drop">
              <select name="module30">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Designated (Level H). ";
  $option_block.="Please Choose...</option>\n";
  $option_block.=get_droptxtdb('6','CSH2005C');
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
              </select>
            </td>
          </tr>
<!-- 4th Row -->
          <tr>
            <td class="drop">
              <select name="module4">
                <option selected value="CSB1038C">
                  Object Orientated Tools and Techniques
                </option>
                <option value="S"> * Special * </option>
              </select>
            </td>
            <td class="drop">
              <select name="module14">
                <option selected value="CSD1005C">
                  Object Orientated Systems Development
                </option>
                <option value="S"> * Special * </option>
              </select>
            </td>
```

```
            <td class="drop">
              <select name="module19">
                <option selected value="CSD1013C">Project Preparation</option>
                <option value="S"> * Special * </option>
              </select>
            </td>
            <td class="drop">
              <select name="module25">
                <option selected value="CSH1012C">Project Management</option>
                <option value="S"> * Special * </option>
              </select>
            </td>
            <td class="drop">
              <select name="module31">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Designated (Level H). ";
  $option_block.="Please Choose...</option>\n";
  $option_block.=get_droptxtdb('6','CSH2005C');
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
              </select>
            </td>
          </tr>
<!-- 5th Row -->
          <tr>
            <td class="drop">
              <select name="module5">
                <option selected value="CSB1005C">Systems Modelling</option>
                <option value="S"> * Special * </option>
              </select>
            </td>
            <td class="drop">
              <select name="module10">
                <option selected value="MSB1013">Discrete Maths</option>
                <option value="S"> * Special * </option>
              </select>
            </td>
            <td class="drop">
              <select name="module15">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Designated (Level D). ";
  $option_block.="Please Choose...</option>\n";
  $option_block.= get_droptxtdb('3','CSD2009C','CSD2007C','CSD1005C');
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
              </select>
            </td>
            <td class="drop">
              <select name="module20">
                <option selected value="CSD1007C">
                  Declarative Languages
                </option>
                <option value="S"> * Special * </option>
              </select>
```

```
            </td>
            <td class="drop">
              <select name="module26">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Designated DOM (Level H). ";
  $option_block.="Please Choose...</option>\n";
  $option_block.= get_droptxtdb('5','CSH2999C','CSH1005C','CSH1012C');
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
              </select>
            </td>
            <td class="drop">
              <select name="module32">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Designated (Level H). ";
  $option_block.="Please Choose...</option>\n";
  $option_block.=get_droptxtdb('6','CSH2005C');
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
              </select>
            </td>
          </tr>
<!-- 6th Row -->
          <tr>
            <td class="drop">
              <select name="module6">
                <option selected value="CSB1003C">
                  Context of Computer Science
                </option>
                <option value="E">English Language Module</option>
                <option value="S"> * Special * </option>
              </select>
            </td>
            <td class="drop">
              <select name="module11">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Free Choice (Level B), ";
  $option_block.="Please Choose...</option>\n";
  $option_block.=get_droptxtdb('2','CSB1002C','CSB1006C','CSB2002C','MSB1013');
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  echo "$option_block";
?>
              </select>
            </td>
            <td class="drop">
              <select name="module16">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Free Choice (Level D), ";
```

```
  $option_block.="Please Choose...</option>\n";
  $option_block.= get_droptxtdb('3','CSD2009C','CSD2007C','CSD1005C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
  $option_block.=" * Nothing * </option>\n";
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
              </select>
            </td>
            <td class="drop">
              <select name="module21">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Free Choice (Level D), ";
  $option_block.="Please Choose...</option>\n";
  $option_block.= get_droptxtdb('4','CSD2003C','CSM1013C','CSD1007C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
  $option_block.=" * Nothing * </option>\n";
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
              </select>
            </td>
            <td class="drop">
              <select name="module27">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Free Choice (Level H), ";
  $option_block.="Please Choose...</option>\n";
  $option_block.= get_droptxtdb('5','CSH2999C','CSH1005C','CSH1012C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
  $option_block.=" * Nothing * </option>\n";
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
              </select>
            </td>
            <td class="drop">
              <select name="module33">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Free Choice (Level H), ";
  $option_block.="Please Choose...</option>\n";
  $option_block.=get_droptxtdb('6','CSH2005C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
```

```
  $option_block.=" * Nothing * </option>\n";
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
              </select>
            </td>
          </tr>
<!-- 7th row -->
          <tr>
            <td class="darker" colspan="2">
              <a href="http://www.c-worker.ch/txtdbapi/index_eng.php">
                <img src="./resources/txtdb-powered.png" alt="logo"
                   style="width: 80px; height: 15px;border:0;">
              </a>
            </td>
            <td class="drop">
              <select name="module17">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * This is optional...</option>\n";
  $option_block.= get_droptxtdb('3','CSD2009C','CSD2007C','CSD1005C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
  $option_block.=" * Nothing * </option>\n";
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
              </select>
            </td>
            <td class="drop">
              <select name="module22">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * This is optional...</option>\n";
  $option_block.= get_droptxtdb('4','CSD2003C','CSM1013C','CSD1007C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
  $option_block.=" * Nothing * </option>\n";
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
              </select>
            </td>
            <td class="drop">
              <select name="module28">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * This is optional...</option>\n";
  $option_block.= get_droptxtdb('5','CSH2999C','CSH1005C','CSH1012C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
```

```
  $option_block.=" * Nothing * </option>\n";
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
              </select>
            </td>
            <td class="drop">
              <select name="module34">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * This is optional...</option>\n";
  $option_block.=get_droptxtdb('6','CSH2005C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
  $option_block.=" * Nothing * </option>\n";
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
              </select>
            </td>
          </tr>
<!-
  Nice button!
  from: http://www.ozoneasylum.com/Forum2/HTML/001764.html
-->
          <tr>
            <td onclick="document.myform.submit();" onmouseout="outView(this);"
               onmouseover="overView(this);" class="cellButton" colspan="6">
              SUBMIT
            </td>
          </tr>
        </tbody>
      </table>
    </form>
    <table border="0" cellspacing="0" cellpadding="0">
      <tbody>
        <tr>
          <td class="note00">Notes:</td>
        </tr>
        <tr>
          <td class="note01">1</td>
          <td class="note02">
HND students are expected to take "Context of computer Science" as their
default choice in semester 1 (Subject to Note 3 below).
          </td>
        </tr>
        <tr>
          <td class="note01">2</td>
          <td class="note02">
Students who have A level math (or equivalent) must take the replacement module
"Analytical Techniques I".
          </td>
        </tr>
        <tr>
          <td class="note01">3</td>
          <td class="note02">
```

```
Students who, due to English not being their first language, are required to
take an English language module in semester one should take it instead of
"Context of Computer Science".
          </td>
        </tr>
        <tr>
          <td class="note01">4</td>
          <td class="note02">
The module "Project Design and Implementation" is taken over the whole of year
two. When making module choices it should be counted as 10 credits in each of
Semester one and two.
          </td>
        </tr>
        <tr>
          <td class="note01">5</td>
          <td class="note02">
"Computing and Society" is offered in both semester one and two. You may take
it either semester. If you take it in semester two you should swap it for a
semester two designated or free choice module.
          </td>
        </tr>
        <tr>
          <td class="note01">6</td>
          <td class="note02">
The project module in year three is shown as being entirely in semester one,
but in practice while most of the work should be done in semester one the
project is handed in part way through semester two. For this reason you may
wish to move ten designated or free choice credits from semester one to two
making a 50/70 credit programme.
          </td>
        </tr>
        <tr>
          <td class="note01">7</td>
          <td class="note02">
Pairs of 10 credit designated or free choice modules can be amalgamated into 20
credit modules if you wish, provided this would not take you over the
prescribed maximum number of credits in a semester. In certain circumstances
you may be permitted to do a 70/50 or 50/70 credit programme in a year. If you
wish to do this discuss it with your personal tutor. You may not do more than
70 credits of new modules in any given semester.
          </td>
        </tr>
        <tr>
          <td class="note01">8</td>
          <td class="note02">
You may also swap designated and free choice modules between semesters provided
you do the correct number of each over the year.
          </td>
        </tr>
      </tbody>
    </table>
  </body>
</html>
```

1.2 secondpagebest.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv=content-type content="text/html; charset=ISO-8859-1">
<?php
  include("txt-db-api/txt-db-api.php");
  include("resources/functions.php");
?>
    <title>
      secondpagebest.php
    </title>
    <link href="resources/hobbs.css" type="text/css" rel="stylesheet">
    <script type="text/javascript" src="resources/overview.js"></script>
  </head>
  <body>
    <h1>
      Module Choice Form
    </h1>
      <form name="myform" id="myform" action="thirdpagetxtdb.php"
         method="post">
        <table border="1" cellpadding="2" cellspacing="3">
          <tbody>
            <tr>
              <td class="darker" colspan="2">
                <p class="title">
                  Pathway Choice: BSc (Hons) Computer Science
                </p>
              </td>
            </tr>
            <tr>
              <td style="width:50%;" class="sem">Semester 1</td>
              <td style="width:50%;" class="sem">Semester 2</td>
            </tr>
            <tr>
              <td class="year" colspan="2">Year 1</td>
            </tr>
            <tr>
              <td class="drop">
                <select name="module1">
                  <option selected value="CSB1030C">
                    Underpinning Skills for Computer Science
                  </option>
                  <option value="S"> * Special * </option>
                </select>
              </td>
              <td class="drop">
                <select name="module7">
                  <option selected value="CSB1002C">
                    Computer Systems Architecture
                  </option>
                  <option value="S"> * Special * </option>
                </select>
              </td>
            </tr>
            <tr>
              <td class="drop">
```

```
    <select name="module2">
      <option selected value="MSB1016">
        Maths Skills for Computer Science
      </option>
      <option value="S"> * Special * </option>
    </select>
  </td>
  <td class="drop">
    <select name="module8">
      <option selected value="CSB1006C">
        Systems Programming
      </option>
      <option value="S"> * Special * </option>
    </select>
  </td>
</tr>
<tr>
  <td class="drop">
    <select name="module3">
      <option selected value="CSB1039C">
        Software Fundamentals
      </option>
      <option value="S"> * Special * </option>
    </select>
  </td>
  <td class="drop" rowspan="2">
    <select name="module9">
      <option selected value="CSB2002C">
        Systems Analysis and Design
      </option>
      <option value="S"> * Special * </option>
    </select>
  </td>
</tr>
<tr>
  <td class="drop">
    <select name="module4">
      <option selected value="CSB1038C">
        Object Orientated Tools and Techniques
      </option>
      <option value="S"> * Special * </option>
    </select>
  </td>
</tr>
<tr>
  <td class="drop">
    <select name="module5">
      <option selected value="CSB1005C">Systems Modelling</option>
      <option value="S"> * Special * </option>
    </select>
  </td>
  <td class="drop">
    <select name="module10">
      <option selected value="MSB1013">Discrete Maths</option>
      <option value="S"> * Special * </option>
    </select>
  </td>
</tr>
<tr>
  <td class="drop">
```

```
                <select name="module6">
                  <option selected value="CSB1003C">
                    Context of Computer Science
                  </option>
                  <option value="E">English Language Module</option>
                  <option value="S"> * Special * </option>
                </select>
              </td>
              <td class="drop">
                <select name="module11">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Free Choice (Level B), ";
  $option_block.="Please Choose...</option>\n";
  $option_block.=
get_droptxtdb('2','CSB1002C','CSB1006C','CSB2002C','MSB1013');
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  echo "$option_block";
?>
                </select>
              </td>
            </tr>
            <tr>
              <td class="year" colspan="2">Year 2</td>
            </tr>
            <tr>
              <td class="drop" colspan="2">
                <select name="module12">
                  <option selected value="CSD2009C">
                    Project Design and Implementation
                  </option>
                  <option value="S"> * Special * </option>
                </select>
              </td>
            </tr>
            <tr>
              <td class="drop">
                <select name="module13">
                  <option selected value="CSD2007C">
                    Information Systems: Tools and Concepts
                  </option>
                  <option value="S"> * Special * </option>
                </select>
              </td>
              <td class="drop">
                <select name="module18">
                  <option selected value="CSD2003C">
                    Computer Networks and Communications
                  </option>
                  <option value="S"> * Special * </option>
                </select>
              </td>
            </tr>
            <tr>
              <td class="drop">
                <select name="module14">
```

```
                    <option selected value="CSD1005C">
                      Object Orientated Systems Development
                    </option>
                    <option value="S"> * Special * </option>
                  </select>
                </td>
                <td class="drop">
                  <select name="module19">
                    <option selected value="CSD1013C">
                      Project Preparation
                    </option>
                    <option value="S"> * Special * </option>
                  </select>
                </td>
              </tr>
              <tr>
                <td class="drop">
                  <select name="module15">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Designated (Level D). ";
  $option_block.="Please Choose...</option>\n";
  $option_block.= get_droptxtdb('3','CSD2009C','CSD2007C','CSD1005C');
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
                  </select>
                </td>
                <td class="drop">
                  <select name="module20">
                    <option selected value="CSD1007C">
                      Declarative Languages
                    </option>
                    <option value="S"> * Special * </option>
                  </select>
                </td>
              </tr>
              <tr>
                <td class="drop">
                  <select name="module16">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Free Choice (Level D), ";
  $option_block.="Please Choose...</option>\n";
  $option_block.= get_droptxtdb('3','CSD2009C','CSD2007C','CSD1005C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
  $option_block.=" * Nothing * </option>\n";
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
                  </select>
                </td>
                <td class="drop">
                  <select name="module21">
<?php
```

```
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Free Choice (Level D), ";
  $option_block.="Please Choose...</option>\n";
  $option_block.= get_droptxtdb('4','CSD2003C','CSM1013C','CSD1007C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
  $option_block.=" * Nothing * </option>\n";
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
                </select>
              </td>
            </tr>
            <tr>
              <td class="drop">
                <select name="module17">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * This is optional...</option>\n";
  $option_block.= get_droptxtdb('3','CSD2009C','CSD2007C','CSD1005C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
  $option_block.=" * Nothing * </option>\n";
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
                </select>
              </td>
              <td class="drop">
                <select name="module22">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * This is optional...</option>\n";
  $option_block.= get_droptxtdb('4','CSD2003C','CSM1013C','CSD1007C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
  $option_block.=" * Nothing * </option>\n";
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
                </select>
              </td>
            </tr>
            <tr>
              <td class="year" colspan="2">Year 3</td>
            </tr>
            <tr>
              <td class="drop">
                <select name="module23">
                  <option selected value="CSH2999C">
                    Undergraduate Project
```

```
                    </option>
                    <option value="S"> * Special * </option>
                  </select>
                </td>
                <td class="drop">
                  <select name="module29">
                    <option selected value="CSH2005C">
                      Distributed Systems
                    </option>
                    <option value="S"> * Special * </option>
                  </select>
                </td>
              </tr>
              <tr>
                <td class="drop">
                  <select name="module24">
                    <option selected value="CSH1005C">
                      Computing and Society
                    </option>
                    <option value="S"> * Special * </option>
                  </select>
                </td>
                <td class="drop">
                  <select name="module30">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Designated (Level H). ";
  $option_block.="Please Choose...</option>\n";
  $option_block.=get_droptxtdb('6','CSH2005C');
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
                  </select>
                </td>
              </tr>
              <tr>
                <td class="drop">
                  <select name="module25">
                    <option selected value="CSH1012C">Project Management</option>
                    <option value="S"> * Special * </option>
                  </select>
                </td>
                <td class="drop">
                  <select name="module31">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Designated (Level H). ";
  $option_block.="Please Choose...</option>\n";
  $option_block.=get_droptxtdb('6','CSH2005C');
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
                  </select>
                </td>
              </tr>
              <tr>
                <td class="drop">
                  <select name="module26">
```

```
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Designated (Level H). ";
  $option_block.="Please Choose...</option>\n";
  $option_block.=get_droptxtdb('5','CSH2999C','CSH1005C','CSH1012C');
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
                </select>
              </td>
              <td class="drop">
                <select name="module32">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Designated (Level H). ";
  $option_block.="Please Choose...</option>\n";
  $option_block.=get_droptxtdb('6','CSH2005C');
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
                </select>
              </td>
            </tr>
            <tr>
              <td class="drop">
                <select name="module27">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Free Choice (Level H), ";
  $option_block.="Please Choose...</option>\n";
  $option_block.=get_droptxtdb('5','CSH2999C','CSH1005C','CSH1012C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
  $option_block.=" * Nothing * </option>\n";
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
                </select>
              </td>
              <td class="drop">
                <select name="module33">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Free Choice (Level H), ";
  $option_block.="Please Choose...</option>\n";
  $option_block.=get_droptxtdb('6','CSH2005C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
  $option_block.=" * Nothing * </option>\n";
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
```

```
                </select>
              </td>
            </tr>
            <tr>
              <td class="drop">
                <select name="module28">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * This is optional...</option>\n";
  $option_block.=get_droptxtdb('5','CSH2999C','CSH1005C','CSH1012C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
  $option_block.=" * Nothing * </option>\n";
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
                </select>
              </td>
              <td class="drop">
                <select name="module34">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * This is optional...</option>\n";
  $option_block.=get_droptxtdb('6','CSH2005C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
  $option_block.=" * Nothing * </option>\n";
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
                </select>
              </td>
            </tr>
            <tr>
              <td style="font-size:large;" onclick="document.myform.submit();"
                  onmouseout="outView(this);" onmouseover="overView(this);"
                   class="cellButton" colspan="2">
                  SUBMIT
              </td>
            </tr>
            <tr>
              <td class="darker" colspan="2">
                <a href="http://www.c-worker.ch/txtdbapi/index_eng.php">
                  <img src="./resources/txtdb-powered.png" alt="logo"
                     style="width:80px;height:15px;border:0;">
                </a>
              </td>
            </tr>
          </tbody>
        </table>
      </form>
    <table border="0" cellspacing="0" cellpadding="0">
      <tbody>
        <tr>
```

```
          <td class="note00">Notes:</td>
        </tr>
        <tr>
          <td class="note01">1</td>
          <td class="note02">
HND students are expected to take "Context of computer Science" as their
default choice in semester 1 (Subject to Note 3 below).
          </td>
        </tr>
        <tr>
          <td class="note01">2</td>
          <td class="note02">
Students who have A level math (or equivalent) must take the replacement module
"Analytical Techniques I".
          </td>
        </tr>
        <tr>
          <td class="note01">3</td>
          <td class="note02">
Students who, due to English not being their first language, are required to
take an English language module in semester one should take it instead of
"Context of Computer Science".
          </td>
        </tr>
        <tr>
          <td class="note01">4</td>
          <td class="note02">
The module "Project Design and Implementation" is taken over the whole of year
two. When making module choices it should be counted as 10 credits in each of
Semester one and two.
          </td>
        </tr>
        <tr>
          <td class="note01">5</td>
          <td class="note02">
"Computing and Society" is offered in both semester one and two. You may take
it either semester. If you take it in semester two you should swap it for a
semester two designated or free choice module.
          </td>
        </tr>
        <tr>
          <td class="note01">6</td>
          <td class="note02">
The project module in year three is shown as being entirely in semester one,
but in practice while most of the work should be done in semester one the
project is handed in part way through semester two. For this reason you may
wish to move ten designated or free choice credits from semester one to two
making a 50/70 credit programme.
          </td>
        </tr>
        <tr>
          <td class="note01">7</td>
          <td class="note02">
Pairs of 10 credit designated or free choice modules can be amalgamated into 20
credit modules if you wish, provided this would not take you over the
prescribed maximum number of credits in a semester. In certain circumstances
you may be permitted to do a 70/50 or 50/70 credit programme in a year. If you
wish to do this discuss it with your personal tutor. You may not do more than
70 credits of new modules in any given semester.
          </td>
```

```
        </tr>
        <tr>
          <td class="note01">8</td>
          <td class="note02">
You may also swap designated and free choice modules between semesters provided
you do the correct number of each over the year.
          </td>
        </tr>
      </tbody>
    </table>
  </body>
</html>
```

1.3 secondpagebestfast.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv=content-type content="text/html; charset=ISO-8859-1">
<?php
  include("resources/functions.php");
?>
    <title>secondpagebestfast.php</title>
    <link href="resources/hobbs.css" type="text/css" rel="stylesheet">
    <script type="text/javascript" src="resources/overview.js"></script>
  </head>
  <body>
    <h1>Module Choice Form</h1>
    <form name="myform" id="myform" action="thirdpagetxtdb.php" method="post">
      <table border="1" cellpadding="2" cellspacing="3">
        <tbody>
          <tr>
            <td class="darker" colspan="2">
              <p class="title">Pathway Choice: BSc (Hons) Computer Science</p>
            </td>
          </tr>
          <tr>
            <td style="width:50%;" class="sem">Semester 1</td>
            <td style="width:50%;" class="sem">Semester 2</td>
          </tr>
          <tr>
            <td class="year" colspan="2">Year 1</td>
          </tr>
          <tr>
            <td class="drop">
              <select name="module1">
                <option selected value="CSB1030C">
                  Underpinning Skills for Computer Science
                </option>
                <option value="S"> * Special * </option>
              </select>
            </td>
            <td class="drop">
              <select name="module7">
                <option selected value="CSB1002C">
                  Computer Systems Architecture
                </option>
                <option value="S"> * Special * </option>
              </select>
            </td>
          </tr>
          <tr>
            <td class="drop">
              <select name="module2">
                <option selected value="MSB1016">
                  Maths Skills for Computer Science
                </option>
                <option value="S"> * Special * </option>
              </select>
            </td>
            <td class="drop">
```

```
    <select name="module8">
      <option selected value="CSB1006C">Systems Programming</option>
      <option value="S"> * Special * </option>
    </select>
  </td>
</tr>
<tr>
  <td class="drop">
    <select name="module3">
      <option selected value="CSB1039C">
        Software Fundamentals
      </option>
      <option value="S"> * Special * </option>
    </select>
  </td>
  <td class="drop" rowspan="2">
    <select name="module9">
      <option selected value="CSB2002C">
        Systems Analysis and Design
      </option>
      <option value="S"> * Special * </option>
    </select>
  </td>
</tr>
<tr>
  <td class="drop">
    <select name="module4">
      <option selected value="CSB1038C">
        Object Orientated Tools and Techniques
      </option>
      <option value="S"> * Special * </option>
    </select>
  </td>
</tr>
<tr>
  <td class="drop">
    <select name="module5">
      <option selected value="CSB1005C">Systems Modelling</option>
      <option value="S"> * Special * </option>
    </select>
  </td>
  <td class="drop">
    <select name="module10">
    <option selected value="MSB1013">Discrete Maths</option>
      <option value="S"> * Special * </option>
    </select>
  </td>
</tr>
<tr>
  <td class="drop">
    <select name="module6">
      <option selected value="CSB1003C">
        Context of Computer Science
      </option>
      <option value="E">English Language Module</option>
      <option value="S"> * Special * </option>
    </select>
  </td>
  <td class="drop">
    <select name="module11">
```

```
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Free Choice (Level B), ";
  $option_block.="Please Choose...</option>\n";
  $option_block.=get_dropmysql('2','CSB1002C','CSB1006C','CSB2002C','MSB1013');
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  echo "$option_block";
?>
              </select>
            </td>
          </tr>
          <tr>
            <td class="year" colspan="2">Year 2</td>
          </tr>
          <tr>
            <td class="drop" colspan="2">
              <select name="module12">
                <option selected value="CSD2009C">
                  Project Design and Implementation
                </option>
                <option value="S"> * Special * </option>
              </select>
            </td>
          </tr>
          <tr>
            <td class="drop">
              <select name="module13">
                <option selected value="CSD2007C">
                  Information Systems: Tools and Concepts
                </option>
                <option value="S"> * Special * </option>
              </select>
            </td>
            <td class="drop">
              <select name="module18">
                <option selected value="CSD2003C">
                  Computer Networks and Communications
                </option>
                <option value="S"> * Special * </option>
              </select>
            </td>
          </tr>
          <tr>
            <td class="drop">
              <select name="module14">
                <option selected value="CSD1005C">
                  Object Orientated Systems Development
                </option>
                <option value="S"> * Special * </option>
              </select>
            </td>
            <td class="drop">
              <select name="module19">
                <option selected value="CSD1013C">Project Preparation</option>
                <option value="S"> * Special * </option>
              </select>
```

```
            </td>
          </tr>
          <tr>
            <td class="drop">
              <select name="module15">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Designated (Level D). ";
  $option_block.="Please Choose...</option>\n";
  $option_block.=get_dropmysql('3','CSD2009C','CSD2007C','CSD1005C');
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
              </select>
            </td>
            <td class="drop">
              <select name="module20">
                <option selected value="CSD1007C">
                  Declarative Languages
                </option>
                <option value="S"> * Special * </option>
              </select>
            </td>
          </tr>
          <tr>
            <td class="drop">
              <select name="module16">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Free Choice (Level D), ";
  $option_block.="Please Choose...</option>\n";
  $option_block.=get_dropmysql('3','CSD2009C','CSD2007C','CSD1005C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
  $option_block.=" * Nothing * </option>\n";
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
              </select>
            </td>
            <td class="drop">
              <select name="module21">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Free Choice (Level D), ";
  $option_block.="Please Choose...</option>\n";
  $option_block.=get_dropmysql('4','CSD2003C','CSM1013C','CSD1007C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
  $option_block.=" * Nothing * </option>\n";
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
```

```
            </select>
          </td>
        </tr>
        <tr>
          <td class="drop">
            <select name="module17">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * This is optional...</option>\n";
  $option_block.=get_dropmysql('3','CSD2009C','CSD2007C','CSD1005C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
  $option_block.=" * Nothing * </option>\n";
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
            </select>
          </td>
          <td class="drop">
            <select name="module22">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * This is optional...</option>\n";
  $option_block.=get_dropmysql('4','CSD2003C','CSM1013C','CSD1007C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
  $option_block.=" * Nothing * </option>\n";
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
            </select>
          </td>
        </tr>
        <tr>
          <td class="year" colspan="2">Year 3</td>
        </tr>
        <tr>
          <td class="drop">
            <select name="module23">
              <option selected value="CSH2999C">
                Undergraduate Project
              </option>
              <option value="S"> * Special * </option>
            </select>
          </td>
          <td class="drop">
            <select name="module29">
              <option selected value="CSH2005C">Distributed Systems</option>
              <option value="S"> * Special * </option>
            </select>
          </td>
        </tr>
        <tr>
          <td class="drop">
```

```
              <select name="module24">
                <option selected value="CSH1005C">
                  Computing and Society
                </option>
                <option value="S"> * Special * </option>
              </select>
            </td>
            <td class="drop">
              <select name="module30">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Designated (Level H). ";
  $option_block.="Please Choose...</option>\n";
  $option_block.=get_dropmysql('6','CSH2005C');
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
              </select>
            </td>
          </tr>
          <tr>
            <td class="drop">
              <select name="module25">
                <option selected value="CSH1012C">Project Management</option>
                <option value="S"> * Special * </option>
              </select>
            </td>
            <td class="drop">
              <select name="module31">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Designated (Level H). ";
  $option_block.="Please Choose...</option>\n";
  $option_block.=get_dropmysql('6','CSH2005C');
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
              </select>
            </td>
          </tr>
          <tr>
            <td class="drop">
              <select name="module26">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Designated (Level H). ";
  $option_block.="Please Choose...</option>\n";
  $option_block.=get_dropmysql('5','CSH2999C','CSH1005C','CSH1012C');
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
              </select>
            </td>
            <td class="drop">
              <select name="module32">
<?php
  $option_block="<option selected value=\"0\">";
```

```
  $option_block.=" * Designated (Level H). ";
  $option_block.="Please Choose...</option>\n";
  $option_block.=get_dropmysql('6','CSH2005C');
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
              </select>
            </td>
          </tr>
          <tr>
            <td class="drop">
              <select name="module27">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Free Choice (Level H), ";
  $option_block.="Please Choose...</option>\n";
  $option_block.=get_dropmysql('5','CSH2999C','CSH1005C','CSH1012C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
  $option_block.=" * Nothing * </option>\n";
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
              </select>
            </td>
            <td class="drop">
              <select name="module33">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * Free Choice (Level H), ";
  $option_block.="Please Choose...</option>\n";
  $option_block.=get_dropmysql('6','CSH2005C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
  $option_block.=" * Nothing * </option>\n";
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
              </select>
            </td>
          </tr>
          <tr>
            <td class="drop">
              <select name="module28">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * This is optional...</option>\n";
  $option_block.=get_dropmysql('5','CSH2999C','CSH1005C','CSH1012C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
  $option_block.=" * Nothing * </option>\n";
```

```
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
              </select>
            </td>
            <td class="drop">
              <select name="module34">
<?php
  $option_block="<option selected value=\"0\">";
  $option_block.=" * This is optional...</option>\n";
  $option_block.=get_dropmysql('6','CSH2005C');
  $option_block.="<option value=\"E\">";
  $option_block.=" * External Module * ";
  $option_block.="</option>\n";
  $option_block.="<option value=\"0\">";
  $option_block.=" * Nothing * </option>\n";
  $option_block.="<option value=\"S\">";
  $option_block.=" * Special * </option>\n";
  echo "$option_block";
?>
              </select>
            </td>
          </tr>
          <tr>
            <td style="font-size:large;" onclick="document.myform.submit();"
               onmouseout="outView(this);" onmouseover="overView(this);"
               class="cellButton" colspan="2">
              SUBMIT
            </td>
          </tr>
        </tbody>
      </table>
    </form>
    <table border="0" cellspacing="0" cellpadding="0">
      <tbody>
        <tr>
          <td class="note00">Notes:</td>
        </tr>
        <tr>
          <td class="note01">1</td>
          <td class="note02">
HND students are expected to take "Context of computer Science" as their
default choice in semester 1 (Subject to Note 3 below).
          </td>
        </tr>
        <tr>
          <td class="note01">2</td>
          <td class="note02">
Students who have A level math (or equivalent) must take the replacement module
"Analytical Techniques I".
          </td>
        </tr>
        <tr>
          <td class="note01">3</td>
          <td class="note02">
Students who, due to English not being their first language, are required to
take an English language module in semester one should take it instead of
"Context of Computer Science".
          </td>
```

```
        </tr>
        <tr>
          <td class="note01">4</td>
          <td class="note02">
The module "Project Design and Implementation" is taken over the whole of year
two. When making module choices it should be counted as 10 credits in each of
Semester one and two.
          </td>
        </tr>
        <tr>
          <td class="note01">5</td>
          <td class="note02">
"Computing and Society" is offered in both semester one and two. You may take
it either semester. If you take it in semester two you should swap it for a
semester two designated or free choice module.
          </td>
        </tr>
        <tr>
          <td class="note01">6</td>
          <td class="note02">
The project module in year three is shown as being entirely in semester one,
but in practice while most of the work should be done in semester one the
project is handed in part way through semester two. For this reason you may
wish to move ten designated or free choice credits from semester one to two
making a 50/70 credit programme.
          </td>
        </tr>
        <tr>
          <td class="note01">7</td>
          <td class="note02">
Pairs of 10 credit designated or free choice modules can be amalgamated into 20
credit modules if you wish, provided this would not take you over the
prescribed maximum number of credits in a semester. In certain circumstances
you may be permitted to do a 70/50 or 50/70 credit programme in a year. If you
wish to do this discuss it with your personal tutor. You may not do more than
70 credits of new modules in any given semester.
          </td>
        </tr>
        <tr>
          <td class="note01">8</td>
          <td class="note02">
You may also swap designated and free choice modules between semesters provided
you do the correct number of each over the year.
          </td>
        </tr>
      </tbody>
    </table>
  </body>
</html>
```

1.4 functions.php

```
<?php

  // These are for the second pages. The 1st is to get stuff from TXTDB,
  // the 2nd gets stuff from MySQL.

  function get_droptxtdb(){
    // Connect to database
    $db=new Database("hobbs");
    // Sort out query
    $query="SELECT Code, Title FROM module WHERE Semester = '".
    func_get_arg(0)."'";
    $num_of_args=func_num_args();
    for($loop=1;$loop<$num_of_args;$loop++){
      $query.=" AND Code !='".func_get_arg($loop)."'";
    }
    $query.=";";
    $rs=$db->executeQuery($query);
    // Sort out results
    $stuff="";
    while($rs->next()){
      $vals=$rs->getCurrentValues();
      $Code=$vals[0];
      $Title=$vals[1];
      $stuff.="<option value=\"$Code\">$Title</option>\n";
    }
    return $stuff;
  }

  function get_dropmysql(){
    // Connect to database and select proper database
    $link=mysql_connect('localhost', 'root', 'batman')
      or die('Could not connect: ' . mysql_error());
    mysql_select_db('hobbs') or die('Could not select database');
    // This sorts out our query
    $num_of_args=func_num_args();
    $query="SELECT Code, Title FROM modules WHERE Semester = '".
    func_get_arg(0)."'";
    for($loop=1;$loop<$num_of_args;$loop++){
      $query.=" AND Code !='".func_get_arg($loop)."'";
    }
    $query.=";";
    // This sorts out our results
    $stuff="";
    $result = mysql_query($query) or die('Query failed: ' . mysql_error());
    while($row = mysql_fetch_array($result, MYSQL_NUM)){
      $stuff.="<option value=\"$row[0]\">$row[1]</option>\n";
    }
    mysql_close($link);
    return $stuff;
  }

// General worker functions here:

  function check_not_exists_test($work,$reference,$a){
    for ($i=0;$i<count($reference[$a]);$i++){
      if ($work == $reference[$a][$i]){
        echo "<font color=\"red\"><b>Repitition of:</b></font><br>";
```

```
        return 1;
        break;
      }
    }
    return 0;
  }

  function what_else($work){
    if($work == "S")
      echo "<font color=red><b>Please check</b></font><br>Special";
    else{
      if($work == "E") echo "External";
      else echo "No Entry";
    }
  }

  function check_co_req($reference,$a){
    $something=$reference[$a];
    for($i=0;$i<count($something);$i++){
      $db=new Database("hobbs");
      $rs=$db->executeQuery("SELECT requisites.module FROM requisites, module
WHERE module.Code = '$something[$i]' AND module.Req_Code = requisites.req_num
AND requisites.type = 'co';");
      while($rs->next()){
        $vals = $rs->getCurrentValues();
        if(!(in_array($vals[0],$something))){
          printf("<font color=\"red\"><b>ERROR:</b></font><br><font size=\"-
2\">%s has Co-Requisite %s</font><br>",$something[$i],$vals[0]);
        }
      }
    }
  }

  function check_co_req_test($reference,$a){
    $something=$reference[$a];
    for($i=0;$i<count($something);$i++){
      $db=new Database("hobbs");
      $rs=$db->executeQuery("SELECT requisites.module FROM requisites, module
WHERE module.Code = '$something[$i]' AND module.Req_Code = requisites.req_num
AND requisites.type = 'co';");
      while($rs->next()){
        $vals=$rs->getCurrentValues();
        if(!(in_array($vals[0],$something))){
          return 1;
        }
      }
    }
  }

  function check_pre_req($reference,$a){
    $something = array();
    if($a == "2") $something = $reference[1];
    else{
      for($i = 1; $i < $a; $i++){
        $something = array_merge($something, $reference[$i]);
      }
    }
    $somethingelse = $reference[$a];
    for($i = 0; $i < count($somethingelse); $i++){
      $query = "SELECT requisites.module FROM requisites, module ";
```

```
      $query.= "WHERE module.Code = '$somethingelse[$i]' ";
      $query.= "AND module.Req_Code = requisites.req_num ";
      $query.= "AND requisites.type = 'pre';";
      $db = new Database("hobbs");
      $rs = $db->executeQuery($query);
      while($rs->next()){
        $vals = $rs->getCurrentValues();
        if(!(in_array($vals[0], $something))){
          printf("<font color=\"red\"><b>ERROR:</b></font><br><font size=\"-
2\">%s has Pre-Requisite %s</font><br>",$somethingelse[$i],$vals[0]);
        }
      }
    }
  }

  function check_pre_req_test($reference,$a){
    $something=array();
    if ($a == "2") $something = $reference[1];
    else{
      for($i=1;$i<$a;$i++){
        $something = array_merge($something, $reference[$i]);
      }
    }
    $somethingelse = $reference[$a];
    for($i=0;$i<count($somethingelse);$i++){
      $db=new Database("hobbs");
      $rs=$db->executeQuery("SELECT requisites.module FROM requisites, module
WHERE module.Code = '$somethingelse[$i]' AND module.Req_Code =
requisites.req_num AND requisites.type = 'pre';");
      while($rs->next()){
        $vals = $rs->getCurrentValues();
        if(!(in_array($vals[0],$something))){
          return 1;
        }
      }
    }
  }

/* Used to get the credit value from txtdb for the specified module code,
these are then added to the totals for the year and semester. */

  function get_credits($work){
    $db=new Database("hobbs");
    $rs=$db->executeQuery("SELECT Credits FROM module WHERE Code = '$work';");
    $rs->next();
    $vals=$rs->getCurrentValues();
    return $vals[0];
  }

/* Used to get the credit value from MySQL for the specified module code,
these are then added to the totals for the year and semester. */

  function get_creditsmysql($work){
    // Connect to database and select proper database
    $link=mysql_connect('localhost','root','batman')
    or die('Could not connect: ' . mysql_error());
    mysql_select_db('hobbs')or die('Could not select database');
    $query="SELECT Credits FROM modules WHERE Code = '$work';";
    $result=mysql_query($query)or die('Query failed: ' . mysql_error());
    $vals=mysql_fetch_array($result);
```

```
    mysql_close($link);
    return $vals[0];
  }

/*
<!-- This is what the table used to look like:
+-----------------+
| TITLE           |
+-----+-----+-----+
|YEAR1|YEAR2|YEAR3|
+--+--+--+--+--+--|
| I|II| I|II| I|II|
+--+--+--+--+--+--|
|01|07| 12  |  |  |
+--+--+--+--+23|29|
|02|08|  |  |  |  |
+--+--+13|18+--+--|
|03|  |  |  |24|30|
+--+09+--+--+--+--+
|04|  |14|19|25|31|
+--+--+--+--+--+--+
|05|10|15|20|26|32|
|--+--+--+--+--+--+
|06|11|16|21|27|33|
+--+--+--+--+--+--+
|     |17|22|28|34|
+-----+--+--+--+--+
|XXXX SUBMIT XXXXX|
+-----------------+

Where as this is what it looks like now, much nicer I think:

+-----------------+
|      TITLE      |
+--------+--------+
|   I    |   II   |
+--------+--------+
|      YEAR 1     |
+--------+--------+
|   01   |   07   |
+--------+--------+
|   02   |   08   |
+--------+--------+
|   03   |        |
+--------+   09   |
|   04   |        |
+--------+--------+
|   05   |   10   |
+--------+--------+
|   06   |   11   |
+--------+--------+
|      YEAR 2     |
+--------+--------+
|       12        |
+--------+--------+
|   13   |   18   |
+--------+--------+
|   14   |   19   |
+--------+--------+
|   15   |   20   |
```

```
+--------+--------+
|   16   |   21   |
+--------+--------+
|   17   |   22   |
+--------+--------+
|       YEAR 3     |
+--------+--------+
|   23   |   28   |
+--------+--------+
|   24   |   29   |
+--------+--------+
|   25   |   30   |
+--------+--------+
|   26   |   31   |
+--------+--------+
|   27   |   32   |
+--------+--------+
|XXXX SUBMIT XXXXX|
+-----------------+ --> */
?>
```

1.5 hobbs.css

```
body{
  font-weight:bold;
  font-family:Ariel,Helvetica,sans-serif;
  font-size:small;
  color:#000000;
  background-color:#CCCCCC;}
table{
  text-align:left;
  background-color:#CCCCCC;
  width:96%;}
.darker{
  background-color:#999999;}
.lighter{
  background-color:#000000;
  color:#FFFFFF;}
h1{
  text-align:center;
  text-decoration:underline;}
.subheader{
  text-align:center;
  font-size:smaller;}
#image002{
  position:absolute;
  left:50%;
  top:50%;
  margin-top:-250px;
  margin-left:-350px;
  width:700px;
  height:500px;}
.cellButton{
  font-size:larger;
  font-weight:bolder;
  text-align:center;
  background-color:#999999;
  color:#000000;
  border:#ffffff 1px solid;
  cursor:pointer;}
.cellButton2{
  font-size:larger;
  font-weight:bolder;
  text-align:center;
  background-color:#000000;
  color:#ffffff;
  border:#ffffff 1px solid;
  cursor:pointer;}
.title{
  font-weight:bolder;
  background-color:#999999;
  font-size:large;}
.year{
  font-weight:bolder;
  background-color:#999999;
  font-size:large;
  text-align:center;
  width:32%;}
.sem{
  text-align:center;
```

```
  background-color:#999999;
  font-size:large;
  font-weight:bolder;
  width:16%;}
.drop{
  text-align:center;
  vertical-align:middle;
  background-color:#CCCCCC;}
.note00{
  vertical-align:top;
  width:5%;
  font-size:smaller;
  text-align:center;
  text-decoration:underline;}
.note01{
  vertical-align:top;
  text-align:center;
  font-size:smaller;}
.note02{
  vertical-align:top;
  font-size:smaller;
  font-weight:normal;}
img{
  display:block;
  margin: auto;}
```

1.6 overview.js

```
function overView(that){
  that.className = 'cellButton2';
}
function outView(that){
  that.className = 'cellButton';
}
```

2.1 choice.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv=content-type content="text/html; charset=ISO-8859-1">
<?php
include("./resources/functions.php");
include("./resources/prefs.php");
/*********************************************************************/
/* This array holds all of the drop down values for the degree. With  */
/* it we call the MySQL database just the number of times required to */
/* fill it for each semester instead of for each module, (6 not 42).   */
/*********************************************************************/
$modules=array();
for($i=0;$i<($periodsOfStudyPerYear*$years);$i++){
  $modules[$i]=getDropText($i+1);
}
?>
    <title>Module Choice Form</title>
    <link href="./resources/degree.css" type="text/css" rel="stylesheet">
    <script type="text/javascript" src="./resources/overview.js"></script>
  </head>
  <body>
    <h1>Module Choice Form</h1>
    <form name="myform" id="myform" action="validation.php" method="post">
      <table border="1" cellpadding="2" cellspacing="3">
        <tbody>
          <tr>
            <td class="title" colspan="<?php echo $periodsOfStudyPerYear; ?>">
<?php print $degreeTitle; ?>
            </td>
          </tr>
<?php
for($a=0;$a<$years;$a++){
  echo '<tr>';
  echo '<td class="year" colspan="'.$periodsOfStudyPerYear.'">';
  echo 'Year '.($a+1);
  echo '</td>';
  echo '</tr>';
  echo '<tr>';
  for($b=0;$b<$periodsOfStudyPerYear;$b++){
    echo '<td style="width:';
    echo (100/$periodsOfStudyPerYear).'%;" class="sem">';
    echo 'Semester '.($b+1);
    echo '</td>';
  }
  echo '</tr>';
  for($c=(($a*$periodsOfStudyPerYear)*$maxModulesPerSemester);
     $c<((($a*$periodsOfStudyPerYear)*$maxModulesPerSemester)+
     $maxModulesPerSemester);$c++){
    echo '<tr>';
    for($d=0;$d<$periodsOfStudyPerYear;$d++){
      echo '<td class="drop">';
      echo '<select class="droplist" name="';
      echo (($maxModulesPerSemester*$d)+$c).'">';
      echo $modules[(($a*$periodsOfStudyPerYear)+$d)];
      echo '</select>';
```

```
      echo '</td>';
    }
    echo '</tr>';
  }
}
?>
          <tr>
            <td onclick="document.myform.submit();" onmouseout="outView(this);"
               onmouseover="overView(this);" class="cellButton"
               colspan="<?php echo $periodsOfStudyPerYear; ?>">
              SUBMIT
            </td>
          </tr>
        </tbody>
      </table>
    </form>
  </body>
</html>
```

2.2 validation.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv=content-type content="text/html; charset=ISO-8859-1">
<?php
include("./resources/functions.php");
include("./resources/prefs.php");
/*********************************************************************/
/* This array will hold all of the values posted from the Module     */
/* choice form. It will serve as our main testing bed and will be    */
/* accessed a lot... I think ;-)                                     */
/*********************************************************************/
$choices=array();
for($a=0;$a<($years*$periodsOfStudyPerYear);$a++){
  for($b=0;$b<$maxModulesPerSemester;$b++){
    $choices[$a][$b]=getNameCredits($choices[$a],
    $_POST[(($a*$maxModulesPerSemester)+$b)],$choices);
  }
  $choices[$a][credits]=countCredits($choices[$a]);
}
$errors=0;
?>
    <title>Module Validation Form</title>
    <link href="./resources/degree.css" type="text/css" rel="stylesheet">
    <script type="text/javascript" src="./resources/overview.js"></script>
  </head>
  <body>
    <h1>Module Validation Form</h1>
    <table border="1" cellpadding="2" cellspacing="3">
      <tbody>
        <tr>
          <td class="title" colspan="<?php echo $periodsOfStudyPerYear; ?>">
<?php print $degreeTitle; ?>
          </td>
        </tr>
<?php
for($a=0;$a<$years;$a++){
  echo '<tr>';
  echo '<td class="year" colspan="'.$periodsOfStudyPerYear.'">';
  echo 'Year '.($a+1);
  echo '</td>';
  echo '</tr>';
  echo '<tr>';
  for($b=0;$b<$periodsOfStudyPerYear;$b++){
    echo '<td style="width:'.(100/$periodsOfStudyPerYear).'%;" class="sem">';
    echo 'Semester '.($b+1);
    echo '</td>';
  }
  echo '</tr>';
  for($c=0;$c<$maxModulesPerSemester;$c++){
    echo '<tr>';
    for($d=0;$d<$periodsOfStudyPerYear;$d++){
      $currentChoice=$choices[(($a*$periodsOfStudyPerYear)+$d)][$c];
      echo '<td class="drop">';
      if($currentChoice[code]=="0") echo "Nothing";
      elseif($currentChoice[code]=="S") echo "Special";
```

```
      elseif($currentChoice[code]=="R") echo "Repitition";
      else{
        echo $currentChoice[title];
        echo ' ('.$currentChoice[code].') ';
        echo $currentChoice[credits].' credits';
        $errors+=checkRequisites($currentChoice,$choices);
      }
      echo '</td>';
    }
    echo '</tr>';
  }
  $creditsThisYear=0;
  for($b=0;$b<$periodsOfStudyPerYear;$b++){
    $creditsThisYear+=$choices[(($a*$periodsOfStudyPerYear)+$b)][credits];
  }
  for($b=0;$b<$periodsOfStudyPerYear;$b++){
    echo '<td class="note">';
    $credits=$choices[(($a*$periodsOfStudyPerYear)+$b)][credits];
    if($credits<(($creditsAYear/2)-$leeway)){
      echo 'Credits: '.$credits.' <img src="./resources/error.gif" alt=""
onmouseover="return escape(\'Too Few Credits. You must do '.($creditsAYear/2).'
credits, plus or minus '.$leeway.' credits, as long as you do not do more or
less than '.$creditsAYear.' credits in a year.\')">';
    }
    elseif($credits>(($creditsAYear/2)+$leeway)){
      echo 'Credits: '.$credits.' <img src="./resources/error.gif" alt=""
onmouseover="return escape(\'Too Many Credits. You must do '.($creditsAYear/
2).' credits, plus or minus '.$leeway.' credits, as long as you do not do more
or less than '.$creditsAYear.' credits in a year.\')">';
    }
    else{
      if($creditsThisYear<$creditsAYear){
        echo 'Credits: '.$credits.' <img src="./resources/error.gif" alt=""
onmouseover="return escape(\'While you have done enough credits for this
semester you have not done enough for the year. You must do '.($creditsAYear/
2).' credits, plus or minus '.$leeway.' credits, as long as you do not do more
or less than '.$creditsAYear.' credits in a year.\')">';
      }
      elseif($creditsThisYear>$creditsAYear){
        echo 'Credits: '.$credits.' <img src="./resources/error.gif" alt=""
onmouseover="return escape(\'While you have done enough credits for this
semester you have done too many for the year. You must do '.($creditsAYear/2).'
credits, plus or minus '.$leeway.' credits, as long as you do not do more or
less than '.$creditsAYear.' credits in a year.\')">';
      }
      else{
        echo 'Credits: '.$credits;
      }
    }
    echo '</td>';
  }
}
if($errors>0){
  echo '<tr>';
  echo '<td colspan="'.$periodsOfStudyPerYear.'">';
  echo 'There are errors, run your mouse over the <img src="./resources/
error.gif" alt="" onmouseover="return escape(\'Indicates 1 or more errors.\')">
tags above for details.';
  echo '</td>';
  echo '</tr>';
```

```
}
?>
      </tbody>
    </table>
    <pre>
<?php
print_r($choices);
?>
    </pre>
    <script language="JavaScript" type="text/javascript"
       src="./resources/wz_tooltip.js"></script>
  </body>
</html>
```

2.3 sqlite_choice_xhtml.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
  <head>
<?php
include("./resources/sqlite_functions.php");
include("./resources/prefs.php");
/*********************************************************************/
/* This array holds all of the drop down values for the degree. With  */
/* it we call the SQLite database just the number of times required to*/
/* fill it for each semester instead of for each module, (6 not 42).   */
/*********************************************************************/
$modules=array();
for($i=0;$i<($periodsOfStudyPerYear*$years);$i++){
  $modules[$i]=getDropText($i+1);
}
?>
    <title>Module Choice Form</title>
    <link href="./resources/degree.css" type="text/css" rel="stylesheet"/>
    <script type="text/javascript" src="./resources/overview.js"></script>
  </head>
  <body>
    <h1>Module Choice Form</h1>
    <form id="myform" action="sqlite_validation_xhtml.php" method="post">
      <table border="1" cellpadding="2" cellspacing="3">
        <tbody>
          <tr>
            <td class="title" colspan="<?php echo
$periodsOfStudyPerYear; ?>"><?php print $degreeTitle; ?></td>
          </tr>
<?php
for($a=0;$a<$years;$a++){
  echo '<tr>';
  echo '<td class="year" colspan="'.$periodsOfStudyPerYear.'">';
  echo 'Year '.($a+1);
  echo '</td>';
  echo '</tr>';
  echo '<tr>';
  for($b=0;$b<$periodsOfStudyPerYear;$b++){
    echo '<td style="width:'.(100/$periodsOfStudyPerYear).'%;" class="sem">';
    echo 'Semester '.($b+1);
    echo '</td>';
  }
  echo '</tr>';
  for($c=(($a*$periodsOfStudyPerYear)*$maxModulesPerSemester);
   $c<((($a*$periodsOfStudyPerYear)*$maxModulesPerSemester)+
   $maxModulesPerSemester);$c++){
    echo '<tr>';
    for($d=0;$d<$periodsOfStudyPerYear;$d++){
      echo '<td class="drop">';
      echo '<select class="droplist"
         name="'.(($maxModulesPerSemester*$d)+$c).'">';
      echo $modules[(($a*$periodsOfStudyPerYear)+$d)];
      echo '</select>';
      echo '</td>';
  }
    echo '</tr>';
```

```
  }
}
?>
          <tr>
            <td onclick="document.getElementById('myform').submit();"
               onmouseout="outView(this);" onmouseover="overView(this);"
               class="cellButton"
               colspan="<?php echo $periodsOfStudyPerYear; ?>">
              SUBMIT
            </td>
          </tr>
        </tbody>
      </table>
    </form>
    <p>
      <a href="http://validator.w3.org/check?uri=referer"><img
         src="./resources/valid-xhtml10.png"
         alt="Valid XHTML 1.0 Strict"
         style="height:31px;width:88;border:0px;"/></a>
      <a class="right" href="http://jigsaw.w3.org/css-validator/"><img
         src="./resources/vcss.gif"
         alt="Valid CSS!" style="height:31px;width:88;border:0px;"/></a>
    </p>
  </body>
</html>
```

2.4 sqlite_validation_xhtml.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
  <head>
<?php
include("./resources/sqlite_functions.php");
include("./resources/prefs.php");
/*********************************************************************/
/* This array will hold all of the values posted from the Module     */
/* choice form. It will serve as our main testing bed and will be    */
/* accessed a lot... I think ;-)                                     */
/*********************************************************************/
$choices=array();
for($a=0;$a<($years*$periodsOfStudyPerYear);$a++){
  for($b=0;$b<$maxModulesPerSemester;$b++){
    $choices[$a][$b]=getNameCredits($choices[$a],
    $_POST[(($a*$maxModulesPerSemester)+$b)],$choices);
  }
  $choices[$a][credits]=countCredits($choices[$a]);
}
$errors=0;
?>
    <title>Module Validation Form</title>
    <link href="./resources/degree.css" type="text/css" rel="stylesheet"/>
    <script type="text/javascript" src="./resources/overview.js"></script>
  </head>
  <body>
    <h1>Module Validation Form</h1>
    <table border="1" cellpadding="2" cellspacing="3">
      <tbody>
        <tr>
          <td class="titleNew" colspan="<?php echo $periodsOfStudyPerYear; ?>">
<?php echo $degreeTitle."\n"; ?>
          </td>
        </tr>
<?php
for($a=0;$a<$years;$a++){
  echo ' <tr>'."\n";
  echo ' <td class="yearNew" colspan="'.$periodsOfStudyPerYear.'">';
  echo 'Year '.($a+1);
  echo '</td>'."\n";
  echo ' </tr>'."\n";
  echo ' <tr>'."\n";
  for($b=0;$b<$periodsOfStudyPerYear;$b++){
    echo ' <td style="width:'.(100/$periodsOfStudyPerYear).'%;" ';
    echo 'class="semNew">';
    echo 'Semester '.($b+1);
    echo '</td>'."\n";
  }
  echo ' </tr>';
  for($c=0;$c<$maxModulesPerSemester;$c++){
    echo "\n".' <tr>';
      for($d=0;$d<$periodsOfStudyPerYear;$d++){
        $currentChoice=$choices[(($a*$periodsOfStudyPerYear)+$d)][$c];
        echo "\n".' <td class="drop">'."\n";
        if($currentChoice[code]=="0"){
          echo <<<END
```

```
<table border="0" cellpadding="1" cellspacing="0">
  <tr>
    <td style="width:32px;" class="dropTable"></td>
    <td class="dropTable">Nothing</td>
  </tr>
</table>
END;
        }
        elseif($currentChoice[code]=="S"){
          echo <<<END
<table border="0" cellpadding="1" cellspacing="0">
  <tr>
    <td style="width:32px;" class="dropTable"></td>
    <td class="dropTable">Special</td>
  </tr>
</table>
END;
        }
        elseif($currentChoice[code]=="R"){
          echo <<<END
<table border="0" cellpadding="1" cellspacing="0">
  <tr>
    <td style="width:32px;" class="dropTable">
      <img src="./resources/error.gif" alt="" onmouseover="return escape('You
chose the same module twice.');"/>
    </td>
    <td class="dropTable">Repitition</td>
  </tr>
</table>
END;
        }
        else{
          echo <<<END
<table border="0" cellpadding="1" cellspacing="0">
  <tr>
    <td style="width:32px;" class="dropTable">
END;
          $errors+=checkRequisites($currentChoice,$choices,
             $periodsOfStudyPerYear);
          echo <<<END
      </td>
    <td class="dropTable">$currentChoice[title] ($currentChoice[credits]
credits)
    </td>
  </tr>
</table>
END;
        }
        echo "\n".' </td>';}
        echo "\n".' </tr>';
      }
      $creditsThisYear=0;
      $halfCredits=($creditsAYear/2);
      for($b=0;$b<$periodsOfStudyPerYear;$b++){
        $creditsThisYear+=$choices[(($a*$periodsOfStudyPerYear)+$b)][credits];
      }
      echo "\n".' <tr>';
      for($b=0;$b<$periodsOfStudyPerYear;$b++){
        echo "\n".' <td class="note">'."\n";
        $credits=$choices[(($a*$periodsOfStudyPerYear)+$b)][credits];
```

```
        if($credits<(($creditsAYear/2)-$leeway)){
          echo <<<END
<table border="0" cellpadding="1" cellspacing="0">
  <tr>
    <td style="width:32px;" class="noteTable">
      <img src="./resources/error.gif" alt="" onmouseover="return escape('Too
Few Credits. You must do $halfCredits credits, plus or minus $leeway credits,
as long as you do not do more or less than $creditsAYear credits in a
year.');"/>
    </td>
    <td class="noteTable">Credits: $credits</td>
  </tr>
</table>
END;
          $errors++;
        }
        elseif($credits>(($creditsAYear/2)+$leeway)){
          echo <<<END
<table border="0" cellpadding="1" cellspacing="0">
  <tr>
    <td style="width:32px;" class="noteTable">
      <img src="./resources/error.gif" alt="" onmouseover="return escape('Too
Many Credits. You must do $halfCredits credits, plus or minus $leeway credits,
as long as you do not do more or less than $creditsAYear credits in a
year.');"/>
    </td>
    <td class="noteTable">Credits: $credits</td>
  </tr>
</table>';
END;
          $errors++;
        }
        else{
          if($creditsThisYear<$creditsAYear){
            echo <<<END
<table border="0" cellpadding="1" cellspacing="0">
  <tr>
    <td style="width:32px;" class="noteTable">
      <img src="./resources/error.gif" alt="" onmouseover="return escape('While
you have done enough credits for this semester you have not done enough for the
year. You must do $halfCredits credits, plus or minus $leeway credits, as long
as you do not do more or less than $creditsAYear credits in a year.');"/>
    </td>
    <td class="noteTable">Credits: $credits</td>
  </tr>
</table>
END;
            $errors++;
          }
          elseif($creditsThisYear>$creditsAYear){
            echo <<<END
<table border="0" cellpadding="1" cellspacing="0">
  <tr>
    <td style="width:32px;" class="noteTable">
      <img src="./resources/error.gif" alt="" onmouseover="return escape('While
you have done enough credits for this semester you have done too many for the
year. You must do $halfCredits credits, plus or minus $leeway credits, as long
as you do not do more or less than $creditsAYear credits in a year.');"/>
    </td>
    <td class="noteTable">Credits: $credits</td>
```

```
  </tr>
</table>';
END;
            $errors++;
          }
          else{
            echo <<<END
<table border="0" cellpadding="1" cellspacing="0">
  <tr>
    <td style="width:32px;" class="noteTable"></td>
    <td class="noteTable">Credits: $credits</td>
  </tr>
</table>
END;
          }
        }
        echo "\n".' </td>';
      }
      echo "\n".' </tr>'."\n";
   }
   if($errors>0){
     echo <<<END
<tr>
  <td colspan="$periodsOfStudyPerYear">
    <table border="0" cellpadding="1" cellspacing="0">
      <tr>
        <td style="width:32px;">
          <img src="./resources/error.gif" alt="" onmouseover="return
escape('Indicates 1 or more errors.');"/>
        </td>
        <td>There are errors, run your mouse over the <img src="./resources/
error.gif" alt=""/> tags above for details.
        </td>
      </tr>
    </table>
  </td>
</tr>
END;
}
?>
      </tbody>
    </table>
    <p>
      <a href="http://validator.w3.org/check?uri=referer">
        <img src="./resources/valid-xhtml10.png"
           alt="Valid XHTML 1.0 Strict"
           style="height:31px;width:88;border:0px;"/>
      </a>
      <a class="right" href="http://jigsaw.w3.org/css-validator/">
        <img src="./resources/vcss.gif" alt="Valid CSS!"
           style="height:31px;width:88;border:0px;"/>
      </a>
    </p>
    <script type="text/javascript" src="./resources/wz_tooltip.js"></script>
  </body>
</html>
```

2.5 functions.php

```
<?php
error_reporting(0);
/*************************************************************************/
/* This function is called by the global arrays from choice.php and      */
/* returns a string with the results of a database access already        */
/* formatted as a html drop-down list sort of way                        */
/*************************************************************************/
function getDropText($semester){
  $link=mysql_connect("localhost","root","batman")
     or die('Could not connect: '.mysql_error());
  mysql_select_db('degree')
     or die('Could not select database');
  $query="SELECT code, title, credits FROM modules WHERE semester = '";
  $query.=$semester."';";
  $result=mysql_query($query)
     or die('Query failed: '.mysql_error());
  mysql_close($link);
  $list="<option value=\"0\">Please Choose</option>";
  while($row=mysql_fetch_array($result,MYSQL_NUM)){
    $list.='<option value="'.$row[0].'">'.$row[1].' (';
    $list.=$row[0].') '.$row[2].' credits</option>';
  }
  $list.="<option value=\"S\">Special</option>";
  return $list;
}
/*************************************************************************/
/* This function takes the modules array and a module code and returns*/
/* the modules array pupulated with the code initially and, after        */
/* accessing MySQL and if the code is a valid code, the title and        */
/* credits appropriate to that code. It also check for repitition.       */
/*************************************************************************/
function getNameCredits($choice,$code,$main){
  switch($code):
    case "0":
      $choice=array(code=>"0");
      return $choice;
      break;
    case "S":
      $choice=array(code=>"S");
      return $choice;
      break;
    default:
      if(checkArray($code,$main)){
        $choice=array(code=>"R");
        return $choice;
        break;
      }
      else{
        $choice=array(code=>$code);
        $link=mysql_connect("localhost","root","batman")
           or die('Could not connect: '.mysql_error());
        mysql_select_db('degree')
           or die('Could not select database');
        $query="SELECT title, credits FROM modules WHERE code = '";
        $query.=$code."';";
        $result=mysql_query($query)
           or die('Query failed: '.mysql_error());
```

```
        mysql_close($link);
        $row=mysql_fetch_assoc($result);
        $choice[title]=$row[title];
        $choice[credits]=$row[credits];
        $choice=getRequisites($choice,$code,"pre");
        $choice=getRequisites($choice,$code,"co");
        return $choice;
      }
  endswitch;
}
/*********************************************************************/
/* This moves through the main $choices array and checks for         */
/* repitition of [code] values, if it finds it it returns true. Not  */
/* an overly complex result but it might be useful for it to return  */
/* the position of the [code] when it's found.                       */
/*********************************************************************/
function checkArray($needle,$haystack){
  $found = false;
  for($a=0;$a<count($haystack);$a++){
    for($b=0;$b<count($haystack[$a]);$b++){
      if(in_array($needle,$haystack[$a][$b])){
        $found = true;
      }
    }
  }
  return $found;
}
/*********************************************************************/
/* This function recieves a semester array and returns the credits due*/
/* on that array. A for loop evaluates the 1st condition only once so */
/* the array is only counted the once.                               */
/*                                                                   */
/*********************************************************************/
function countCredits($semester){
  $credits = 0;
  for($i=count($semester)-1; $i>=0; $i--) {
    $credits += $semester[$i][credits];
  }
  return $credits;
}
/*********************************************************************/
/* This function grabs the requisites (both co- and pre-) and puts   */
/* them in as array's, both co- and pre-. This is used by the        */
/* checkRequisites function to flag up any errors in the main page,  */
/* it access the database a lot though.                              */
/*********************************************************************/
function getRequisites($choice,$code,$type){
  $link=mysql_connect("localhost","root","batman")
     or die('Could not connect: '.mysql_error());
  mysql_select_db('degree')
     or die('Could not select database');
  $query="SELECT req_code FROM requisites WHERE type = '";
  $query.=$type."' AND code = '".$code."';";
  $result=mysql_query($query)
     or die('Query failed: '.mysql_error());
  mysql_close($link);
  while($row=mysql_fetch_array($result,MYSQL_NUM)){
    $choice[$type][]=$row[0];
  }
  return $choice;
```

```
}
/*************************************************************************/
/* This function writes to the browser if there are errors in pre- or */
/* co- requisites for a given $choice. It also returns the number of  */
/* errors which is used to calculate whether or not a message appears */
/* at the bottom of the table.                                        */
/*************************************************************************/
function checkRequisites($subject,$main){
  $errorLevel=0;
  $errorMessage="<img src=\"./resources/error.gif\" alt=\"\"
onmouseover=\"return escape('";
  foreach($subject as $value){
    if(is_array($value)){
      for($a=0;$a<count($value);$a++){
        if(!checkArray($value[$a],$main)){
          $errorMessage.=" ".$value[$a]." required!";
          $errorLevel++;
        }
      }
    }
  }
  if($errorLevel>0) echo $errorMessage."')\">";
  return $errorLevel;
}
/*************************************************************************/
/* This rather tasty function is by Marco Stumper and is located here:*/
/* <http://uk2.php.net/manual/en/function.in-array.php#45516>         */
/* It is a good example of Function Recursion in that the function    */
/* calls itself in order to check through a multi-dimensional array.  */
/*************************************************************************/
function in_array_multi($needle, $haystack){
  $found = false;
  foreach($haystack as $value){
    if((is_array($value) && in_array_multi($needle, $value)) || $value ==
$needle){
      $found = true;
    }
  }
  return $found;
}
?>
```

2.6 degree.css

```
body{
  font-family:Ariel,Helvetica,sans-serif;
  color:#000000;
  background-color:#CCCCCC;}
table{
  background-color:#CCCCCC;
  color:#000000;
  width:100%;}
.darker{
  background-color:#999999;
  color:#000000;}
h1{
  text-align:center;}
.cellButton{
  font-size:2em;
  font-weight:bolder;
  text-align:center;
  background-color:#999999;
  color:#000000;
  border:#ffffff 1px solid;
  padding: 0.2em 0em 0em 0em;
  cursor:pointer;}
.cellButton1{
  font-size:2em;
  font-weight:bolder;
  text-align:center;
  background-color:#000000;
  color:#ffffff;
  border:#ffffff 1px solid;
  padding: 0.2em 0em 0em 0em;
  cursor:pointer;}
.title{
  font-weight:bolder;
  background-color:#999999;
  color:#000000;
  font-size:1.2em;}
.titleNew{
  font-weight:bolder;
  background-color:#999999;
  color:#000000;
  font-size:1.2em;
  padding-left:36px;}
.year{
  font-weight:bolder;
  background-color:#999999;
  font-size:2em;
  text-align:center;
  border:#000000 1px solid;
  padding: 0.2em 0em 0em 0em;
  color:#FFFFFF;}
.yearNew{
  font-weight:bolder;
  background-color:#999999;
  font-size:2em;
  border:#000000 1px solid;
  padding: 10px 0px 2px 36px;
  color:#FFFFFF;}
```

```
.sem{
  text-align:center;
  background-color:#999999;
  font-size:1.2em;
  font-weight:bolder;
  color:#000000;
  width:50%;}
.semNew{
  text-align:left;
  background-color:#999999;
  color:#000000;
  font-size:1.2em;
  font-weight:bolder;
  padding-left:36px;}
.note{
  background-color:#999999;
  color:#000000;
  font-size:0.8em;}
.noteTable{
  background-color:#999999;
  color:#000000;}
.drop{
  font-weight:normal;
  color:#000000;
  font-size:0.7em;
  background-color:#FFFFFF;}
.dropTable{
  font-weight:normal;
  color:#000000;
  background-color:#FFFFFF;}
.droplist{
  width:100%;
  font-size:smaller;}
```

2.7 sqlite_functions.php

```php
<?php
/*************************************************************************/
/* This function is called by the global arrays from choice.php and      */
/* returns a string with the results of a database access already        */
/* formatted as a html drop-down list sort of way                        */
/*************************************************************************/
function getDropText($sem){
  $dbhandle = sqlite_open('./sqlitedb/degree.sqlite');
  $query="SELECT code, title, credits FROM modules WHERE semester = ";
  $query.=$sem;
  $result = sqlite_array_query($dbhandle, $query);
  $list="<option value=\"0\">Please Choose</option>";
  foreach($result as $row){
    $list.='<option value="'.$row[0].'">'.$row[1].' (';
    $list.=$row[0].') '.$row[2].' credits</option>';
  }
  $list.="<option value=\"S\">Special</option>";
  sqlite_close($dbhandle);
  return $list;
}
/*************************************************************************/
/* This function takes the modules array and a module code and returns*/
/* the modules array pupulated with the code initially and, after        */
/* accessing MySQL and if the code is a valid code, the title and        */
/* credits appropriate to that code. It also check for repitition.       */
/*************************************************************************/
function getNameCredits($choice,$code,$main){
  switch($code):
    case "0":
      $choice=array(code=>"0");
      return $choice;
      break;
    case "S":
      $choice=array(code=>"S");
      return $choice;
      break;
    default:
      if(checkArray($code,$main)){
        $choice=array(code=>"R");
        return $choice;
        break;
      }
      else{
        $choice=array(code=>$code);
        $dbhandle = sqlite_open('./sqlitedb/degree.sqlite');
        $query='SELECT title, credits FROM modules WHERE code = "'.$code.'";';
        $result=sqlite_query($dbhandle,$query);
        $row = sqlite_fetch_array($result);
        $choice[title]=$row[title];
        $choice[credits]=$row[credits];
        $choice=getRequisites($choice,$code,"pre");
        $choice=getRequisites($choice,$code,"co");
        sqlite_close($dbhandle);
        return $choice;
      }
  endswitch;
}
```

```
/*************************************************************************/
/* This moves through the main $choices array and checks for             */
/* repitition of [code] values, if it finds it it returns true. Not      */
/* an overly complex result but it might be useful for it to return      */
/* the position of the [code] when it's found.                           */
/*************************************************************************/
function checkArray($needle,$haystack){
  $found = false;
  for($a=0;$a<count($haystack);$a++){
    for($b=0;$b<count($haystack[$a]);$b++){
      if(in_array($needle,$haystack[$a][$b])){
        $found = true;
      }
    }
  }
  return $found;
}
/*************************************************************************/
/* This function recieves a semester array and returns the credits due*/
/* on that array. A for loop evaluates the 1st condition only once so */
/* the array is only counted the once.                                   */
/*                                                                       */
/*************************************************************************/
function countCredits($semester){
  $credits = 0;
  for($i=count($semester)-1; $i>=0; $i--) {
    $credits += $semester[$i][credits];
  }
  return $credits;
}
/*************************************************************************/
/* This function grabs the requisites (both co- and pre-) and puts       */
/* them in as array's, both co- and pre-. This is used by the            */
/* checkRequisites function to flag up any errors in the main page,      */
/* it access the database a lot though.                                  */
/*************************************************************************/
function getRequisites($choice,$code,$type){
  $dbhandle = sqlite_open('./sqlitedb/degree.sqlite');
  $query="SELECT req_code FROM requisites WHERE type = '";
  $query.=$type."' AND code = '".$code."';";
  $result = sqlite_array_query($dbhandle, $query);
  foreach($result as $row){
    $choice[$type][]=$row[0];
  }
  sqlite_close($handle);
  return $choice;
}
/*************************************************************************/
/* This function writes to the browser if there are errors in pre- or */
/* co- requisites for a given $choice. It also returns the number of  */
/* errors which is used to calculate whether or not a message appears */
/* at the bottom of the table.                                           */
/*************************************************************************/
function checkRequisites($subject,$main,$sems){
  $errorLevel=0;
  $errorMessage="\n"." ";
  $errorMessage.="<img src=\"./resources/error.gif\" alt=\"\" ";
  $errorMessage.="\n"." ";
  $errorMessage.="onmouseover=\"return escape('";
  foreach($subject as $value){
```

```
    if(is_array($value)){
      for($a=0;$a<count($value);$a++){
        if(!checkArray($value[$a],$main)){
          $errorMessage.=getWhatWhen($value[$a],$sems)." required<br/>";
          $errorLevel++;
        }
      }
    }
  }
  if($errorLevel>0) echo $errorMessage."');\"/>";
  return $errorLevel;
}
/*************************************************************************/
/*                                                                       */
/*                                                                       */
/*                                                                       */
/*                                                                       */
/*************************************************************************/
function getWhatWhen($code,$sems){
  $dbhandle=sqlite_open('./sqlitedb/degree.sqlite');
  $query='SELECT title, semester FROM modules WHERE code = "'.$code.'";';
  $result=sqlite_query($dbhandle,$query);
  $row=sqlite_fetch_array($result);
  $title=$row[title];
  $year=floor($row[semester] / $sems) + 1;
  $sem=($semester % $sems) + 1;
  sqlite_close($dbhandle);
  return $title." (Year ".$year." Semester ".$sem.")";
}
/*************************************************************************/
/* This rather tasty function is by Marco Stumper and is located here:*/
/* <http://uk2.php.net/manual/en/function.in-array.php#45516>          */
/* It is a good example of Function Recursion in that the function     */
/* calls itself in order to check through a multi-dimensional array.   */
/*************************************************************************/
function in_array_multi($needle, $haystack){
  $found = false;
  foreach($haystack as $value){
    if((is_array($value)&&in_array_multi($needle,$value))||$value==$needle){
      $found = true;
    }
  }
  return $found;
}
?>
```

2.8 prefs.php

```
<?php
$periodsOfStudyPerYear=2;
$years=3;
$maxModulesPerSemester=7;
$degreeTitle="Pathway Choice: BSc (Hons) Computer Science";
$creditsAYear=120;
$leeway=10;
?>
```

3.1 index.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
  <head>
    <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type"/>
    <title>corner test</title>
    <link href="./new.css" type="text/css" rel="stylesheet"/>
    <script type="text/javascript" src="./js/xmlhttp.js"></script>
    <script type="text/javascript" src="./js/whitespace.js"></script>
    <script type="text/javascript" src="./js/functions.js"></script>
  </head>
  <body>
    <table class="main" border="0" cellpadding="0" cellspacing="0">
      <tbody>
        <tr><td colspan="2"><img alt="" src="./images/middle.gif"/></td></tr>
        <tr><td colspan="2"><img alt="" src="./images/top.gif"/></td></tr>
        <tr><td colspan="2" class="title">Module Choice Form</td></tr>
        <tr><td colspan="2"><img alt="" src="./images/bottom.gif"/></td></tr>
        <tr><td colspan="2">
          <img alt="" src="./images/middlelittle.gif"/>
        </td></tr>
        <tr><td colspan="2"><img alt="" src="./images/top.gif"/></td></tr>
        <tr><td colspan="2" class="subtitle">
          Pathway Choice: BSc (Hons) Computer Science
        </td></tr>
        <tr><td colspan="2"><img alt="" src="./images/bottom.gif"/></td></tr>
        <tr><td colspan="2"><img alt="" src="./images/middle.gif"/></td></tr>
        <tr><td colspan="2"><img alt="" src="./images/top.gif"/></td></tr>
        <tr><td class="year" colspan="2">Year 1</td></tr>
        <tr><td colspan="2"><img alt="" src="./images/bottom.gif"/></td></tr>
        <tr><td colspan="2">
          <img alt="" src="./images/middlelittle.gif"/>
        </td></tr>
        <tr><td colspan="2"><img alt="" src="./images/top.gif"/></td></tr>
        <tr>
          <td class="sem">Semester 1</td>
          <td class="sem">Semester 2</td>
        </tr>
        <tr>
          <td id="sem1">
            <table border="0" cellpadding="0" cellspacing="0">
              <tbody>
                <tr>
                  <td id="err0" class="error"></td>
                  <td>
                    <select class="droplist" id="select1"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSB1030C">Underpinning Skills for Computer Science (10)</option>
<option value="CSB1005C">System Modelling (10)</option>
<option value="CSB1039C">Software Fundamentals (10)</option>
<option value="CSB1038C">Object Orientated Tools and Techniques (10)</option>
<option value="MSB1016">Maths Skills for Computer Science (10)</option>
<option value="CSB1003C">Context of Computer Science (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
```

```
                </tr>
                <tr>
                  <td class="error" id="err1"></td>
                  <td>
                    <select class="droplist" id="select2"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSB1030C">Underpinning Skills for Computer Science (10)</option>
<option value="CSB1005C">System Modelling (10)</option>
<option value="CSB1039C">Software Fundamentals (10)</option>
<option value="CSB1038C">Object Orientated Tools and Techniques (10)</option>
<option value="MSB1016">Maths Skills for Computer Science (10)</option>
<option value="CSB1003C">Context of Computer Science (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err2"></td>
                  <td>
                    <select class="droplist" id="select3"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSB1030C">Underpinning Skills for Computer Science (10)</option>
<option value="CSB1005C">System Modelling (10)</option>
<option value="CSB1039C">Software Fundamentals (10)</option>
<option value="CSB1038C">Object Orientated Tools and Techniques (10)</option>
<option value="MSB1016">Maths Skills for Computer Science (10)</option>
<option value="CSB1003C">Context of Computer Science (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err3"></td>
                  <td>
                    <select class="droplist" id="select4"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSB1030C">Underpinning Skills for Computer Science (10)</option>
<option value="CSB1005C">System Modelling (10)</option>
<option value="CSB1039C">Software Fundamentals (10)</option>
<option value="CSB1038C">Object Orientated Tools and Techniques (10)</option>
<option value="MSB1016">Maths Skills for Computer Science (10)</option>
<option value="CSB1003C">Context of Computer Science (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err4"></td>
                  <td>
                    <select class="droplist" id="select5"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSB1030C">Underpinning Skills for Computer Science (10)</option>
<option value="CSB1005C">System Modelling (10)</option>
<option value="CSB1039C">Software Fundamentals (10)</option>
<option value="CSB1038C">Object Orientated Tools and Techniques (10)</option>
<option value="MSB1016">Maths Skills for Computer Science (10)</option>
```

```
<option value="CSB1003C">Context of Computer Science (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err5"></td>
                  <td>
                    <select class="droplist" id="select6"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSB1030C">Underpinning Skills for Computer Science (10)</option>
<option value="CSB1005C">System Modelling (10)</option>
<option value="CSB1039C">Software Fundamentals (10)</option>
<option value="CSB1038C">Object Orientated Tools and Techniques (10)</option>
<option value="MSB1016">Maths Skills for Computer Science (10)</option>
<option value="CSB1003C">Context of Computer Science (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err6"></td>
                  <td>
                    <select class="droplist" id="select7"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSB1030C">Underpinning Skills for Computer Science (10)</option>
<option value="CSB1005C">System Modelling (10)</option>
<option value="CSB1039C">Software Fundamentals (10)</option>
<option value="CSB1038C">Object Orientated Tools and Techniques (10)</option>
<option value="MSB1016">Maths Skills for Computer Science (10)</option>
<option value="CSB1003C">Context of Computer Science (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error"><img src="images/error.gif" alt="error"
                     onmouseover="ddrivetip('Too few credits.')"
                     onmouseout="hideddrivetip()" />
                  </td>
                  <td class="result">Credits:  
                    <span id="sem1result" class="red">00</span>
                  </td>
                </tr>
              </tbody>
            </table>
          </td>
          <td id="sem2">
            <table border="0" cellpadding="0" cellspacing="0">
              <tbody>
                <tr>
                  <td class="error" id="err7"></td>
                  <td>
                    <select class="droplist" id="select8"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSB1006C">Systems Programming (10)</option>
<option value="CSB2002C">Systems Analysis and Design (20)</option>
```

```
<option value="CSB1027C">Experimentation in Computer Science (10)</option>
<option value="MSB1013">Discrete Maths (10)</option>
<option value="CSB1002C">Computer System Architecture (10)</option>
<option value="CSB1035C">Basic Internet Configuration (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err8"></td>
                  <td>
                    <select class="droplist" id="select9"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSB1006C">Systems Programming (10)</option>
<option value="CSB2002C">Systems Analysis and Design (20)</option>
<option value="CSB1027C">Experimentation in Computer Science (10)</option>
<option value="MSB1013">Discrete Maths (10)</option>
<option value="CSB1002C">Computer System Architecture (10)</option>
<option value="CSB1035C">
  Basic Internet Configuration (CSB1035C) 10 credits
</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err9"></td>
                  <td>
                    <select class="droplist" id="select10"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSB1006C">Systems Programming (10)</option>
<option value="CSB2002C">Systems Analysis and Design (20)</option>
<option value="CSB1027C">Experimentation in Computer Science (10)</option>
<option value="MSB1013">Discrete Maths (10)</option>
<option value="CSB1002C">Computer System Architecture (10)</option>
<option value="CSB1035C">
  Basic Internet Configuration (CSB1035C) 10 credits
</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err10"></td>
                  <td>
                    <select class="droplist" id="select11"
                      onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSB1006C">Systems Programming (10)</option>
<option value="CSB2002C">Systems Analysis and Design (20)</option>
<option value="CSB1027C">Experimentation in Computer Science (10)</option>
<option value="MSB1013">Discrete Maths (10)</option>
<option value="CSB1002C">Computer System Architecture (10)</option>
<option value="CSB1035C">
  Basic Internet Configuration (CSB1035C) 10 credits
</option>
<option value="S">Special</option>
                    </select>
```

```
                </td>
              </tr>
              <tr>
                <td class="error" id="err11"></td>
                <td>
                  <select class="droplist" id="select12"
                     onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSB1006C">Systems Programming (10)</option>
<option value="CSB2002C">Systems Analysis and Design (20)</option>
<option value="CSB1027C">Experimentation in Computer Science (10)</option>
<option value="MSB1013">Discrete Maths (10)</option>
<option value="CSB1002C">Computer System Architecture (10)</option>
<option value="CSB1035C">
  Basic Internet Configuration (CSB1035C) 10 credits
</option>
<option value="S">Special</option>
                  </select>
                </td>
              </tr>
              <tr>
                <td class="error" id="err12"></td>
                <td>
                  <select class="droplist" id="select13"
                     onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSB1006C">Systems Programming (10)</option>
<option value="CSB2002C">Systems Analysis and Design (20)</option>
<option value="CSB1027C">Experimentation in Computer Science (10)</option>
<option value="MSB1013">Discrete Maths (10)</option>
<option value="CSB1002C">Computer System Architecture (10)</option>
<option value="CSB1035C">
  Basic Internet Configuration (CSB1035C) 10 credits
</option>
<option value="S">Special</option>
                  </select>
                </td>
              </tr>
              <tr>
                <td class="error" id="err13"></td>
                <td>
                  <select class="droplist" id="select14"
                     onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSB1006C">Systems Programming (10)</option>
<option value="CSB2002C">Systems Analysis and Design (20)</option>
<option value="CSB1027C">Experimentation in Computer Science (10)</option>
<option value="MSB1013">Discrete Maths (10)</option>
<option value="CSB1002C">Computer System Architecture (10)</option>
<option value="CSB1035C">
  Basic Internet Configuration (CSB1035C) 10 credits
</option>
<option value="S">Special</option>
                  </select>
                </td>
              </tr>
              <tr>
                <td class="error">
                  <img src="images/error.gif" alt="error"
                     onmouseover="ddrivetip('Too few credits.')"
```

```
                    onmouseout="hideddrivetip()" />
                </td>
                <td class="result">Credits:  
                  <span id="sem2result" class="red">00</span>
                </td>
              </tr>
            </tbody>
          </table>
        </td>
      </tr>
      <tr><td colspan="2"><img alt="" src="./images/bottom.gif"/></td></tr>
      <tr><td colspan="2"><img alt="" src="./images/middle.gif"/></td></tr>
      <tr><td colspan="2"><img alt="" src="./images/top.gif"/></td></tr>
      <tr><td class="year" colspan="2">Year 2</td></tr>
      <tr><td colspan="2"><img alt="" src="./images/bottom.gif"/></td></tr>
      <tr><td colspan="2">
        <img alt="" src="./images/middlelittle.gif"/>
      </td></tr>
      <tr><td colspan="2"><img alt="" src="./images/top.gif"/></td></tr>
      <tr>
        <td class="sem">Semester 1</td>
        <td class="sem">Semester 2</td>
      </tr>
      <tr>
        <td id="sem3">
          <table border="0" cellpadding="0" cellspacing="0">
            <tbody>
              <tr>
                <td id="err14" class="error"></td>
                <td>
                  <select class="droplist" id="select15"
                    onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSD2007C">Information Systems: Tools and Concepts (20)</option>
<option value="CSD1002C">Computer Graphics Programming (10)</option>
<option value="CSD2002C">Computer Intergrated Manufacture (20)</option>
<option value="CSD1003C">Artificial Intelligence (10)</option>
<option value="CSD1005C">Object Orientated Systems Development (10)</option>
<option value="CSD1011C">Personal Computer Architecture (10)</option>
<option value="CSD2009C">Project Design and Implementation (20)</option>
<option value="CSD1063C">Rapid Application Development (10)</option>
<option value="CSD1008C">Real Time Programming (10)</option>
<option value="CSD1046C">Representations of Computer Images (10)</option>
<option value="S">Special</option>
                  </select>
                </td>
              </tr>
              <tr>
                <td class="error" id="err15"></td>
                <td>
                  <select class="droplist" id="select16"
                    onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSD2007C">Information Systems: Tools and Concepts (20)</option>
<option value="CSD1002C">Computer Graphics Programming (10)</option>
<option value="CSD2002C">Computer Intergrated Manufacture (20)</option>
<option value="CSD1003C">Artificial Intelligence (10)</option>
<option value="CSD1005C">Object Orientated Systems Development (10)</option>
<option value="CSD1011C">Personal Computer Architecture (10)</option>
<option value="CSD2009C">Project Design and Implementation (20)</option>
```

```
<option value="CSD1063C">Rapid Application Development (10)</option>
<option value="CSD1008C">Real Time Programming (10)</option>
<option value="CSD1046C">Representations of Computer Images (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err16"></td>
                  <td>
                    <select class="droplist" id="select17"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSD2007C">Information Systems: Tools and Concepts (20)</option>
<option value="CSD1002C">Computer Graphics Programming (10)</option>
<option value="CSD2002C">Computer Intergrated Manufacture (20)</option>
<option value="CSD1003C">Artificial Intelligence (10)</option>
<option value="CSD1005C">Object Orientated Systems Development (10)</option>
<option value="CSD1011C">Personal Computer Architecture (10)</option>
<option value="CSD2009C">Project Design and Implementation (20)</option>
<option value="CSD1063C">Rapid Application Development (10)</option>
<option value="CSD1008C">Real Time Programming (10)</option>
<option value="CSD1046C">Representations of Computer Images (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err17"></td>
                  <td>
                    <select class="droplist" id="select18"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSD2007C">Information Systems: Tools and Concepts (20)</option>
<option value="CSD1002C">Computer Graphics Programming (10)</option>
<option value="CSD2002C">Computer Intergrated Manufacture (20)</option>
<option value="CSD1003C">Artificial Intelligence (10)</option>
<option value="CSD1005C">Object Orientated Systems Development (10)</option>
<option value="CSD1011C">Personal Computer Architecture (10)</option>
<option value="CSD2009C">Project Design and Implementation (20)</option>
<option value="CSD1063C">Rapid Application Development (10)</option>
<option value="CSD1008C">Real Time Programming (10)</option>
<option value="CSD1046C">Representations of Computer Images (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err18"></td>
                  <td>
                    <select class="droplist" id="select19"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSD2007C">Information Systems: Tools and Concepts (20)</option>
<option value="CSD1002C">Computer Graphics Programming (10)</option>
<option value="CSD2002C">Computer Intergrated Manufacture (20)</option>
<option value="CSD1003C">Artificial Intelligence (10)</option>
<option value="CSD1005C">Object Orientated Systems Development (10)</option>
<option value="CSD1011C">Personal Computer Architecture (10)</option>
<option value="CSD2009C">Project Design and Implementation (20)</option>
```

```
<option value="CSD1063C">Rapid Application Development (10)</option>
<option value="CSD1008C">Real Time Programming (10)</option>
<option value="CSD1046C">Representations of Computer Images (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err19"></td>
                  <td>
                    <select class="droplist" id="select20"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSD2007C">Information Systems: Tools and Concepts (20)</option>
<option value="CSD1002C">Computer Graphics Programming (10)</option>
<option value="CSD2002C">Computer Intergrated Manufacture (20)</option>
<option value="CSD1003C">Artificial Intelligence (10)</option>
<option value="CSD1005C">Object Orientated Systems Development (10)</option>
<option value="CSD1011C">Personal Computer Architecture (10)</option>
<option value="CSD2009C">Project Design and Implementation (20)</option>
<option value="CSD1063C">Rapid Application Development (10)</option>
<option value="CSD1008C">Real Time Programming (10)</option>
<option value="CSD1046C">Representations of Computer Images (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err20"></td>
                  <td>
                    <select class="droplist" id="select21"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSD2007C">Information Systems: Tools and Concepts (20)</option>
<option value="CSD1002C">Computer Graphics Programming (10)</option>
<option value="CSD2002C">Computer Intergrated Manufacture (20)</option>
<option value="CSD1003C">Artificial Intelligence (10)</option>
<option value="CSD1005C">Object Orientated Systems Development (10)</option>
<option value="CSD1011C">Personal Computer Architecture (10)</option>
<option value="CSD2009C">Project Design and Implementation (20)</option>
<option value="CSD1063C">Rapid Application Development (10)</option>
<option value="CSD1008C">Real Time Programming (10)</option>
<option value="CSD1046C">Representations of Computer Images (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error">
                    <img src="images/error.gif" alt="error"
                       onmouseover="ddrivetip('Too few credits.')"
                       onmouseout="hideddrivetip()" />
                  </td>
                  <td class="result">Credits:  
                    <span id="sem3result" class="red">00</span>
                  </td>
                </tr>
              </tbody>
            </table>
          </td>
```

```
          <td id="sem4">
            <table border="0" cellpadding="0" cellspacing="0">
              <tbody>
                <tr>
                  <td class="error" id="err21"></td>
                  <td>
                    <select class="droplist" id="select22"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSD1009C">Industrial Applications of Computers (10)</option>
<option value="CSD1007C">Declarative Languages (10)</option>
<option value="CSD1006C">Database Programming (10)</option>
<option value="CSD2003C">Computer Networks and Communications (20)</option>
<option value="CSD1001C">Computer Graphics Applications (10)</option>
<option value="CSD1047C">Computer Processing of Electronic Images (20)</option>
<option value="CSD1004C">Object Orientated Data Structures (10)</option>
<option value="CSD1013C">Project Preparation (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err22"></td>
                  <td>
                    <select class="droplist" id="select23"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSD1009C">Industrial Applications of Computers (10)</option>
<option value="CSD1007C">Declarative Languages (10)</option>
<option value="CSD1006C">Database Programming (10)</option>
<option value="CSD2003C">Computer Networks and Communications (20)</option>
<option value="CSD1001C">Computer Graphics Applications (10)</option>
<option value="CSD1047C">Computer Processing of Electronic Images (20)</option>
<option value="CSD1004C">Object Orientated Data Structures (10)</option>
<option value="CSD1013C">Project Preparation (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err23"></td>
                  <td>
                    <select class="droplist" id="select24"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSD1009C">Industrial Applications of Computers (10)</option>
<option value="CSD1007C">Declarative Languages (10)</option>
<option value="CSD1006C">Database Programming (10)</option>
<option value="CSD2003C">Computer Networks and Communications (20)</option>
<option value="CSD1001C">Computer Graphics Applications (10)</option>
<option value="CSD1047C">Computer Processing of Electronic Images (20)</option>
<option value="CSD1004C">Object Orientated Data Structures (10)</option>
<option value="CSD1013C">Project Preparation (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err24"></td>
                  <td>
```

```
                    <select class="droplist" id="select25"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSD1009C">Industrial Applications of Computers (10)</option>
<option value="CSD1007C">Declarative Languages (10)</option>
<option value="CSD1006C">Database Programming (10)</option>
<option value="CSD2003C">Computer Networks and Communications (20)</option>
<option value="CSD1001C">Computer Graphics Applications (10)</option>
<option value="CSD1047C">Computer Processing of Electronic Images (20)</option>
<option value="CSD1004C">Object Orientated Data Structures (10)</option>
<option value="CSD1013C">Project Preparation (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err25"></td>
                  <td>
                    <select class="droplist" id="select26"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSD1009C">Industrial Applications of Computers (10)</option>
<option value="CSD1007C">Declarative Languages (10)</option>
<option value="CSD1006C">Database Programming (10)</option>
<option value="CSD2003C">Computer Networks and Communications (20)</option>
<option value="CSD1001C">Computer Graphics Applications (10)</option>
<option value="CSD1047C">Computer Processing of Electronic Images (20)</option>
<option value="CSD1004C">Object Orientated Data Structures (10)</option>
<option value="CSD1013C">Project Preparation (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err26"></td>
                  <td>
                    <select class="droplist" id="select27"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSD1009C">Industrial Applications of Computers (10)</option>
<option value="CSD1007C">Declarative Languages (10)</option>
<option value="CSD1006C">Database Programming (10)</option>
<option value="CSD2003C">Computer Networks and Communications (20)</option>
<option value="CSD1001C">Computer Graphics Applications (10)</option>
<option value="CSD1047C">Computer Processing of Electronic Images (20)</option>
<option value="CSD1004C">Object Orientated Data Structures (10)</option>
<option value="CSD1013C">Project Preparation (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err27"></td>
                  <td>
                    <select class="droplist" id="select28"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSD1009C">Industrial Applications of Computers (10)</option>
<option value="CSD1007C">Declarative Languages (10)</option>
<option value="CSD1006C">Database Programming (10)</option>
```

```
<option value="CSD2003C">Computer Networks and Communications (20)</option>
<option value="CSD1001C">Computer Graphics Applications (10)</option>
<option value="CSD1047C">Computer Processing of Electronic Images (20)</option>
<option value="CSD1004C">Object Orientated Data Structures (10)</option>
<option value="CSD1013C">Project Preparation (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error">
                    <img src="images/error.gif" alt="error"
                       onmouseover="ddrivetip('Too few credits.')"
                       onmouseout="hideddrivetip()" />
                  </td>
                  <td class="result">Credits:  
                    <span id="sem4result" class="red">00</span>
                  </td>
                </tr>
              </tbody>
            </table>
          </td>
        </tr>
        <tr><td colspan="2"><img alt="" src="./images/bottom.gif"/></td></tr>
        <tr><td colspan="2"><img alt="" src="./images/middle.gif"/></td></tr>
        <tr><td colspan="2"><img alt="" src="./images/top.gif"/></td></tr>
        <tr><td class="year" colspan="2">Year 3</td></tr>
        <tr><td colspan="2"><img alt="" src="./images/bottom.gif"/></td></tr>
        <tr><td colspan="2">
          <img alt="" src="./images/middlelittle.gif"/>
        </td></tr>
        <tr><td colspan="2"><img alt="" src="./images/top.gif"/></td></tr>
        <tr>
          <td class="sem">Semester 1</td>
          <td class="sem">Semester 2</td>
        </tr>
        <tr>
          <td id="sem5">
            <table border="0" cellpadding="0" cellspacing="0">
              <tbody>
                <tr>
                  <td id="err28" class="error"></td>
                  <td>
                    <select class="droplist" id="select29"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSH1002C">Advanced Systems Programming (10)</option>
<option value="CSH1068C">Dynamic Object Systems (10)</option>
<option value="CSH1069C">
  Emerging Paradigms in Artificial Intelligence (10)
</option>
<option value="CSH1070C">Enterprise Programming (10)</option>
<option value="CSH1073C">Knowledge Management (10)</option>
<option value="CSH1074C">Neural Networks (10)</option>
<option value="CSH1012C">Project Management (10)</option>
<option value="CSH2999C">Undergraduate Project (20)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
```

```
                <tr>
                  <td class="error" id="err29"></td>
                  <td>
                    <select class="droplist" id="select30"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSH1002C">Advanced Systems Programming (10)</option>
<option value="CSH1068C">Dynamic Object Systems (10)</option>
<option value="CSH1069C">
  Emerging Paradigms in Artificial Intelligence (10)
</option>
<option value="CSH1070C">Enterprise Programming (10)</option>
<option value="CSH1073C">Knowledge Management (10)</option>
<option value="CSH1074C">Neural Networks (10)</option>
<option value="CSH1012C">Project Management (10)</option>
<option value="CSH2999C">Undergraduate Project (20)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err30"></td>
                  <td>
                    <select class="droplist" id="select31"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSH1002C">Advanced Systems Programming (10)</option>
<option value="CSH1068C">Dynamic Object Systems (10)</option>
<option value="CSH1069C">
  Emerging Paradigms in Artificial Intelligence (10)
</option>
<option value="CSH1070C">Enterprise Programming (10)</option>
<option value="CSH1073C">Knowledge Management (10)</option>
<option value="CSH1074C">Neural Networks (10)</option>
<option value="CSH1012C">Project Management (10)</option>
<option value="CSH2999C">Undergraduate Project (20)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err31"></td>
                  <td>
                    <select class="droplist" id="select32"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSH1002C">Advanced Systems Programming (10)</option>
<option value="CSH1068C">Dynamic Object Systems (10)</option>
<option value="CSH1069C">
  Emerging Paradigms in Artificial Intelligence (10)
</option>
<option value="CSH1070C">Enterprise Programming (10)</option>
<option value="CSH1073C">Knowledge Management (10)</option>
<option value="CSH1074C">Neural Networks (10)</option>
<option value="CSH1012C">Project Management (10)</option>
<option value="CSH2999C">Undergraduate Project (20)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
```

```
                <tr>
                  <td class="error" id="err32"></td>
                  <td>
                    <select class="droplist" id="select33"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSH1002C">Advanced Systems Programming (10)</option>
<option value="CSH1068C">Dynamic Object Systems (10)</option>
<option value="CSH1069C">
  Emerging Paradigms in Artificial Intelligence (10)
</option>
<option value="CSH1070C">Enterprise Programming (10)</option>
<option value="CSH1073C">Knowledge Management (10)</option>
<option value="CSH1074C">Neural Networks (10)</option>
<option value="CSH1012C">Project Management (10)</option>
<option value="CSH2999C">Undergraduate Project (20)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err33"></td>
                  <td>
                    <select class="droplist" id="select34"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSH1002C">Advanced Systems Programming (10)</option>
<option value="CSH1068C">Dynamic Object Systems (10)</option>
<option value="CSH1069C">
  Emerging Paradigms in Artificial Intelligence (10)
</option>
<option value="CSH1070C">Enterprise Programming (10)</option>
<option value="CSH1073C">Knowledge Management (10)</option>
<option value="CSH1074C">Neural Networks (10)</option>
<option value="CSH1012C">Project Management (10)</option>
<option value="CSH2999C">Undergraduate Project (20)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err34"></td>
                  <td>
                    <select class="droplist" id="select35"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSH1002C">Advanced Systems Programming (10)</option>
<option value="CSH1068C">Dynamic Object Systems (10)</option>
<option value="CSH1069C">
  Emerging Paradigms in Artificial Intelligence (10)
</option>
<option value="CSH1070C">Enterprise Programming (10)</option>
<option value="CSH1073C">Knowledge Management (10)</option>
<option value="CSH1074C">Neural Networks (10)</option>
<option value="CSH1012C">Project Management (10)</option>
<option value="CSH2999C">Undergraduate Project (20)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
```

```
                <tr>
                  <td class="error">
                    <img src="images/error.gif" alt="error"
                       onmouseover="ddrivetip('Too few credits.')"
                       onmouseout="hideddrivetip()" />
                  </td>
                  <td class="result">Credits:  
                    <span id="sem5result" class="red">00</span>
                  </td>
                </tr>
              </tbody>
            </table>
          </td>
          <td id="sem6">
            <table border="0" cellpadding="0" cellspacing="0">
              <tbody>
                <tr>
                  <td class="error" id="err35"></td>
                  <td>
                    <select class="droplist" id="select36"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSH1010C">Investigating Computer Use (10)</option>
<option value="CSH1005C">Computing and Society (10)</option>
<option value="CSH2005C">Distributed Systems (20)</option>
<option value="CSH1071C">Geographical Information Systems (10)</option>
<option value="CSH1072C">Innovations from ICT (10)</option>
<option value="CSH2033C">Internet Databases (20)</option>
<option value="CSH1014C">Quality Management (10)</option>
<option value="CSH1075C">Software Interface Design (10)</option>
<option value="CSH1055C">Software Maintenance and Testing (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err36"></td>
                  <td>
                    <select class="droplist" id="select37"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSH1010C">Investigating Computer Use (10)</option>
<option value="CSH1005C">Computing and Society (10)</option>
<option value="CSH2005C">Distributed Systems (20)</option>
<option value="CSH1071C">Geographical Information Systems (10)</option>
<option value="CSH1072C">Innovations from ICT (10)</option>
<option value="CSH2033C">Internet Databases (20)</option>
<option value="CSH1014C">Quality Management (10)</option>
<option value="CSH1075C">Software Interface Design (10)</option>
<option value="CSH1055C">Software Maintenance and Testing (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err37"></td>
                  <td>
                    <select class="droplist" id="select38"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
```

```
<option value="CSH1010C">Investigating Computer Use (10)</option>
<option value="CSH1005C">Computing and Society (10)</option>
<option value="CSH2005C">Distributed Systems (20)</option>
<option value="CSH1071C">Geographical Information Systems (10)</option>
<option value="CSH1072C">Innovations from ICT (10)</option>
<option value="CSH2033C">Internet Databases (20)</option>
<option value="CSH1014C">Quality Management (10)</option>
<option value="CSH1075C">Software Interface Design (10)</option>
<option value="CSH1055C">Software Maintenance and Testing (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err38"></td>
                  <td>
                    <select class="droplist" id="select39"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSH1010C">Investigating Computer Use (10)</option>
<option value="CSH1005C">Computing and Society (10)</option>
<option value="CSH2005C">Distributed Systems (20)</option>
<option value="CSH1071C">Geographical Information Systems (10)</option>
<option value="CSH1072C">Innovations from ICT (10)</option>
<option value="CSH2033C">Internet Databases (20)</option>
<option value="CSH1014C">Quality Management (10)</option>
<option value="CSH1075C">Software Interface Design (10)</option>
<option value="CSH1055C">Software Maintenance and Testing (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err39"></td>
                  <td>
                    <select class="droplist" id="select40"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSH1010C">Investigating Computer Use (10)</option>
<option value="CSH1005C">Computing and Society (10)</option>
<option value="CSH2005C">Distributed Systems (20)</option>
<option value="CSH1071C">Geographical Information Systems (10)</option>
<option value="CSH1072C">Innovations from ICT (10)</option>
<option value="CSH2033C">Internet Databases (20)</option>
<option value="CSH1014C">Quality Management (10)</option>
<option value="CSH1075C">Software Interface Design (10)</option>
<option value="CSH1055C">Software Maintenance and Testing (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err40"></td>
                  <td>
                    <select class="droplist" id="select41"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSH1010C">Investigating Computer Use (10)</option>
<option value="CSH1005C">Computing and Society (10)</option>
<option value="CSH2005C">Distributed Systems (20)</option>
```

```
<option value="CSH1071C">Geographical Information Systems (10)</option>
<option value="CSH1072C">Innovations from ICT (10)</option>
<option value="CSH2033C">Internet Databases (20)</option>
<option value="CSH1014C">Quality Management (10)</option>
<option value="CSH1075C">Software Interface Design (10)</option>
<option value="CSH1055C">Software Maintenance and Testing (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error" id="err41"></td>
                  <td>
                    <select class="droplist" id="select42"
                       onchange="checkValue(this.value,this.id);">
<option value="0">Please Choose</option>
<option value="CSH1010C">Investigating Computer Use (10)</option>
<option value="CSH1005C">Computing and Society (10)</option>
<option value="CSH2005C">Distributed Systems (20)</option>
<option value="CSH1071C">Geographical Information Systems (10)</option>
<option value="CSH1072C">Innovations from ICT (10)</option>
<option value="CSH2033C">Internet Databases (20)</option>
<option value="CSH1014C">Quality Management (10)</option>
<option value="CSH1075C">Software Interface Design (10)</option>
<option value="CSH1055C">Software Maintenance and Testing (10)</option>
<option value="S">Special</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td class="error">
                    <img src="images/error.gif" alt="error"
                       onmouseover="ddrivetip('Too few credits.')"
                       onmouseout="hideddrivetip()" />
                  </td>
                  <td class="result">Credits:  
                    <span id="sem6result" class="red">00</span>
                  </td>
                </tr>
              </tbody>
            </table>
          </td>
        </tr>
        <tr><td colspan="2"><img alt="" src="./images/bottom.gif"/></td></tr>
        <tr><td colspan="2"><img alt="" src="./images/middle.gif"/></td></tr>
      </tbody>
    </table>
    <p>
      <a href="http://validator.w3.org/check?uri=referer"><img
           style="border:0;width:88px;height:31px;"
           src="http://www.w3.org/Icons/valid-xhtml10"
           alt="Valid XHTML 1.0 Strict" /></a>
    </p>
    <p>
      <a href="http://jigsaw.w3.org/css-validator/check/referer"><img
           style="border:0;width:88px;height:31px;"
           src="http://jigsaw.w3.org/css-validator/images/vcss"
           alt="Valid CSS!" /></a>
    </p>
    <div id="dhtmltooltip"></div>
```

```
    <script type="text/javascript" src="./js/dhtmltooltip.js"></script>
  </body>
</html>
```

3.2 new.css

```
body{
 font-family:       Ariel,Helvetica,sans-serif;
 color:             #000000;
 background-color: #CCCCCC;
}
table.main{
 background-color: #999999;
 color:             #000000;
 width:             770px;
 position:          absolute;
 left:              50%;
 margin-left:       -385px;
}
table img {
 display:           block;
}
td.title{
 padding-left:      15px;
 font-weight:       900;
 font-size:         2em;
}
td.subtitle{
 padding-left:      15px;
 font-weight:       bolder;
 font-size:         1.2em;
}
td.year{
 text-align:        center;
 font-weight:       bolder;
 font-size:         2em;
 padding:           5px 0px 0px 0px;
 background-color: #999999;
 color:             #FFFFFF;
}
td.sem{
 text-align:        center;
 font-size:         1.2em;
 font-weight:       bolder;
 padding:           0px 0px 5px 0px;
 background-color: #999999;
 color:             #000000;
 width:             50%;
}
td.error{
 width:             39px;
 height:            23px;
 padding:           0px 5px 0px 5px;
}
select.droplist{
 width:             331px;
 font-size:         0.7em;
}
td.result{
 text-align:        right;
 font-size:         1.2em;
 font-weight:       bolder;
 width:             331px;
```

```
 padding:           5px 0px 0px 0px;
}
span.red{
 color:             red;
 background-color: #999999;
}
span.yellow{
 color:             yellow;
 background-color: #999999;
}
span.green{
 color:             green;
 background-color: #999999;
}
#dhtmltooltip{
 position:          absolute;
 color:             #000000;
 width:             331px;
 border:            1px solid black;
 padding:           5px 2px 0px 2px;
 background-color: #CCCCCC;
 visibility:        hidden;
 z-index:           1000;
 font-size:         0.8em;
}
```

3.3 checkValue.php

```
<?php
/*****************************************************************************/
/* We need this for access to some variables                                 */
/*****************************************************************************/
include("./prefs.php");
/*****************************************************************************/
/* This is the xml header.                                                   */
/*****************************************************************************/
header('Content-Type: text/xml');
echo '<?xml version="1.0" standalone="yes"?>';
/*****************************************************************************/
/* These are global variables. We need these to initailise the return values */
/* or to access the SQLite database.                                         */
/*****************************************************************************/
$credits = 0;
$db = "./degree.sqlite";
$handle = sqlite_open($db) or die ("Could not open database");
/*****************************************************************************/
/* We need to have access to the string that is passed to this script, this  */
/* string is comprised of:                                                   */
/*  1, the id of the select                                                  */
/*  2, the selected value (Either a valid code, "0" or "S"))                 */
/*  3, the other selected values for that semester                           */
/*  4, the other values it'd be possible to select                           */
/*  All of these are seperated with a ":".                                   */
/*****************************************************************************/
$requestString = $_GET['requestString'];
/*****************************************************************************/
/* In order to access the different elements in the string we explode it,    */
/* giving us access to the stuff we need. $pieces is an array of these       */
/* elements.                                                                 */
/*****************************************************************************/
$pieces = explode(":", $requestString);
/*****************************************************************************/
/* 1, the id of the select                                                   */
/*****************************************************************************/
$id = $pieces[0];
/*****************************************************************************/
/* from which we can calculate the semester which it belongs to              */
/*****************************************************************************/
$semester = ceil((trim($id, "\x3a..\x7f")/$maxModulesPerSemester));
/*****************************************************************************/
/* 2, the selected value (Either a valid code, "0" or "S"))                  */
/*****************************************************************************/
$selected = $pieces[1];
/*****************************************************************************/
/* We'll echo this as the header of the xml                                  */
/*****************************************************************************/
echo "<select id=\"".$id."\" value=\"".$selected."\">";
echo "<semester id=\"".$semester."\">";
/*****************************************************************************/
/* We know that there will be a certain number of modules which are of       */
/* certain interest so we strip them out into another array to work on 'em   */
/*****************************************************************************/
$modulesThisSemester = array_slice($pieces, 2, $maxModulesPerSemester);
/*****************************************************************************/
/* We also know that the last lot of module codes will be interest so we'll  */
```

```
/* strip them out as well... though elements in the arrays might be repeated,*/
/* so we'll get rid of them as well to create a better array                  */
/****************************************************************************/
$otherPossibleModules = array_slice($pieces, 2+$maxModulesPerSemester);
$possibleModules = array_diff($otherPossibleModules, $modulesThisSemester);
/****************************************************************************/
/* For each module we'll need to find out how many credits they are worth   */
/****************************************************************************/
foreach($modulesThisSemester as $totalModule) {
  $query = "SELECT credits FROM modules WHERE code = '$totalModule'";
  $result = sqlite_query($handle, $query)
     or die("Error in query: "
     .sqlite_error_string(sqlite_last_error($handle)));
/****************************************************************************/
/* If there is a result                                                     */
/****************************************************************************/
  if (sqlite_num_rows($result) > 0) {
/****************************************************************************/
/* Add the credits to the variable initalised at the top of the page        */
/****************************************************************************/
    $credits += sqlite_fetch_single($result);
  }
}
sqlite_close($handle);
/****************************************************************************/
/* And echo the total credits in the final xml file                         */
/****************************************************************************/
printf("%02d", $credits);
echo "</semester>";
/****************************************************************************/
/* We need to know what modules are required by the selected module if it   */
/* isn't a special module (i.e. "0" or "S")                                 */
/****************************************************************************/
if ($selected !== "0" || $selected !== "S") {
  getRequisites($selected, "requires");
  getRequiredBy($selected, "requiredBy", "module");
}
/****************************************************************************/
/* NEEDS WORK!                                                              */
/* For each possible module (x) we need to find which modules (y) would be  */
/* required by (x). We also need to be able to see which other modules (z)  */
/* are required by (for want of a better term) y. In terms of pseudo xml:   */
/* <mightHaveBeen id="code">                                                */
/*   <whichIsRequiredBy id="code">                                          */
/*     <module>code</module>                                                */
/*     <module>code</module>                                                */
/*   </whichIsRequiredBy>                                                   */
/*   <whichIsRequiredBy id="code">                                          */
/*     <module>code</module>                                                */
/*     <module>code</module>                                                */
/*     <module>code</module>                                                */
/*   </whichIsRequiredBy>                                                   */
/* </mightHaveBeen>                                                         */
/* <mightHaveBeen id="code">                                                */
/*   <whichIsRequiredBy id="code">                                          */
/*     <module>code</module>                                                */
/*   </whichIsRequiredBy>                                                   */
/*   <whichIsRequiredBy id="code">                                          */
/*     <module>code</module>                                                */
/*   </whichIsRequiredBy>                                                   */
```

```
/* </mightHaveBeen>                                                          */
/*****************************************************************************/
foreach ($possibleModules as $possible) {
  echo "<mightHaveBeen id=\"".$possible."\">";
  getRequiredBy($possible, "whichIsRequiredBy", "module");
  echo "</mightHaveBeen>";
}
/*****************************************************************************/
/* Simply put this function gets the requisites of a given module code and   */
/* echos the result as xml                                                   */
/*****************************************************************************/
function getRequisites($passedCode, $passedString) {
  global $db, $handle;
  $query = "SELECT req_code FROM requisites WHERE code = '$passedCode'";
  $result = sqlite_query($handle, $query)
     or die("Error in query: "
     .sqlite_error_string(sqlite_last_error($handle)));
  if (sqlite_num_rows($result) > 0){
    $resultset = sqlite_fetch_all($result);
    foreach ($resultset as $entry) {
      echo "<".$passedString.">".$entry['req_code']."</".$passedString.">";
    }
  }
}
/*****************************************************************************/
/* This function returns all the modules which a given module is required by,*/
/* but it goes further and finds out what other modules are required by it,  */
/* the results are echoed as nicely formated xml again.                      */
/*****************************************************************************/
function getRequiredBy($passedCode, $passedString, $otherPassedString) {
  global $db, $handle;
  $query = "SELECT code FROM requisites WHERE req_code = '$passedCode'";
  $result = sqlite_query($handle, $query)
     or die("Error in query: "
     .sqlite_error_string(sqlite_last_error($handle)));
  if (sqlite_num_rows($result) > 0){
    $resultset = sqlite_fetch_all($result);
    foreach ($resultset as $entry) {
      echo "<".$passedString." id=\"".$entry['code']."\">";
      getRequisites($entry['code'], $otherPassedString);
      echo "</".$passedString.">";
    }
  }
}
sqlite_close($handle);
echo "</select>";
?>
```

3.4 xmlhttp.js

```
/*****************************************************************************/
/* Not from here: <http://kewlio.com/serendipity/index.php?/archives/        */
/* 180-The-getHTTPObject-function.html> but it does give a nice explanation  */
/* for the code!                                                             */
/* Actually from: <http://jibbering.com/2002/4/httprequest.html>             */
/*****************************************************************************/
function getHTTPObject() {
  var xmlhttp=false;
  /*@cc_on @*/
  /*@if (@_jscript_version >= 5)
  // JScript gives us Conditional compilation, we can cope with old IE
versions.
  // and security blocked creation of the objects.
  try {
    xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
  } catch (e) {
    try {
      xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    } catch (E) {
      xmlhttp = false;
    }
  }
  @end @*/
  if (!xmlhttp && typeof XMLHttpRequest!='undefined') {
    try {
      xmlhttp = new XMLHttpRequest();
    } catch (e) {
      xmlhttp=false;
    }
  }
  if (!xmlhttp && window.createRequest) {
    try {
      xmlhttp = window.createRequest();
    } catch (e) {
      xmlhttp=false;
    }
  }
  return xmlhttp;
}
```

3.5 whitespace.js

```
/** This library is from:
 * http://developer.mozilla.org/en/docs/Whitespace_in_the_DOM
 */

/**
 * Throughout, whitespace is defined as one of the characters
 * "\t"  TAB   \u0009
 * "\n"  LF    \u000A
 * "\r"  CR    \u000D
 * " "   SPC   \u0020
 *
 * This does not use Javascript's "\s" because that includes non-breaking
 * spaces (and also some other characters).
 */

/**
 * Determine whether a node's text content is entirely whitespace.
 *
 * @param nod  A node implementing the |CharacterData| interface (i.e.,
 *             a |Text|, |Comment|, or |CDATASection| node
 * @return     True if all of the text content of |nod| is whitespace,
 *             otherwise false.
 */

function is_all_ws( nod )
{
  // Use ECMA-262 Edition 3 String and RegExp features
  return !(/[^\t\n\r ]/.test(nod.data));
}

/**
 * Determine if a node should be ignored by the iterator functions.
 *
 * @param nod  An object implementing the DOM1 |Node| interface.
 * @return     true if the node is:
 *                1) A |Text| node that is all whitespace
 *                2) A |Comment| node
 *             and otherwise false.
 */

function is_ignorable( nod )
{
  return ( nod.nodeType == 8) || // A comment node
         ( ( nod.nodeType == 3) && is_all_ws(nod) ); // a text node, all ws
}

/**
 * Version of |previousSibling| that skips nodes that are entirely
 * whitespace or comments. (Normally |previousSibling| is a property
 * of all DOM nodes that gives the sibling node, the node that is
 * a child of the same parent, that occurs immediately before the
 * reference node.)
 *
 * @param sib  The reference node.
 * @return     Either:
 *               1) The closest previous sibling to |sib| that is not
 *                  ignorable according to |is_ignorable|, or
```

```
 *               2) null if no such node exists.
 */

function node_before( sib )
{
  while ((sib = sib.previousSibling)) {
    if (!is_ignorable(sib)) return sib;
  }
 return null;
}

/**
 * Version of |nextSibling| that skips nodes that are entirely
 * whitespace or comments.
 *
 * @param sib  The reference node.
 * @return     Either:
 *               1) The closest next sibling to |sib| that is not
 *                  ignorable according to |is_ignorable|, or
 *               2) null if no such node exists.
 */

function node_after( sib )
{
  while ((sib = sib.nextSibling)) {
    if (!is_ignorable(sib)) return sib;
  }
  return null;
}

/**
 * Version of |lastChild| that skips nodes that are entirely
 * whitespace or comments. (Normally |lastChild| is a property
 * of all DOM nodes that gives the last of the nodes contained
 * directly in the reference node.)
 *
 * @param sib  The reference node.
 * @return     Either:
 *               1) The last child of |sib| that is not
 *                  ignorable according to |is_ignorable|, or
 *               2) null if no such node exists.
 */

function last_child( par )
{
  var res=par.lastChild;
  while (res) {
    if (!is_ignorable(res)) return res;
    res = res.previousSibling;
  }
  return null;
}

/**
 * Version of |firstChild| that skips nodes that are entirely
 * whitespace and comments.
 *
 * @param sib  The reference node.
 * @return     Either:
 *               1) The first child of |sib| that is not
```

```
 *                  ignorable according to |is_ignorable|, or
 *               2) null if no such node exists.
 */

function first_child( par )
{
  var res=par.firstChild;
  while (res) {
    if (!is_ignorable(res)) return res;
    res = res.nextSibling;
  }
  return null;
}

/**
 * Version of |data| that doesn't include whitespace at the beginning
 * and end and normalizes all whitespace to a single space. (Normally
 * |data| is a property of text nodes that gives the text of the node.)
 *
 * @param txt  The text node whose data should be returned
 * @return     A string giving the contents of the text node with
 *             whitespace collapsed.
 */

function data_of( txt )
{
  var data = txt.data;
  // Use ECMA-262 Edition 3 String and RegExp features
  data = data.replace(/[\t\n\r ]+/g, " ");
  if (data.charAt(0) == " ")
    data = data.substring(1, data.length);
  if (data.charAt(data.length - 1) == " ")
    data = data.substring(0, data.length - 1);
  return data;
}
```

3.6 functions.js

```
// Global variables:
var http = getHTTPObject();
var isWorking = false;
var URL = "./php/checkValue.php?requestString=";
/*****************************************************************************/
/* Basic one! Everything comes from this one function. To start we need to   */
/* know whether the value is repeated in the document. If it is we need to   */
/* blank it, even if it isn't we still need to prepare the string for        */
/* submission to the php script. We then need to handle the response with the*/
/* handler.                                                                  */
/*****************************************************************************/
function checkValue(value,id) {
  unFlagError(id);
  if (checkExists(value,id)) {
    alert("Repitition");
    document.getElementById(id).value = "0";
  }
  getData(value,id);
}
/*****************************************************************************/
/* We need to pass a number of variables to the php script:                  */
/*  1, the id of the select                                                  */
/*  2, the selected value                                                    */
/*  3, the other selected values for that semester                           */
/*  4, the other possible options for that select                            */
/* All of these need to be seperated with a ":".                             */
/*****************************************************************************/
function getData(value,id) {
/*****************************************************************************/
/* 1, the id of the select                                                   */
/* 2, the selected value                                                     */
/*****************************************************************************/
  var requestString = id+":"+value;
/*****************************************************************************/
/* 3, the other selected values for that semester                            */
/* Note: The 7 on the next line will be replaced with a php variable         */
/*****************************************************************************/
  var semester = document.getElementById("sem"+(Math.ceil(id.slice(6)/7)));
  var selectObjects = semester.getElementsByTagName("select");
  for (var x = selectObjects.length-1; x >= 0; x--) {
    requestString+=":"+selectObjects[x].value;
  }
/*****************************************************************************/
/* 4, the other possible options for that select                             */
/*****************************************************************************/
  var otherOptions =
     document.getElementById(id).getElementsByTagName("option");
  for (var x = otherOptions.length-2; x >= 1; x--){
    requestString+=":"+otherOptions[x].getAttribute('value');
  }
  if (!isWorking && http) {
    http.open("GET", URL + requestString, true);
    http.onreadystatechange = handleHttpResponse;
    isWorking = true;
    http.send(null);
  }
}
```

```
/*****************************************************************************/
/* This is the powerhouse of the client-side script and handles the XML      */
/* returned from the server-side script.                                     */
/*****************************************************************************/
function handleHttpResponse() {
  if (http.readyState == 4) {
    if (http.responseText.indexOf('invalid') == -1) {
      var xmlDocument = http.responseXML;
      var semester = xmlDocument.getElementsByTagName('semester').item(0);
      var originalID =
         xmlDocument.getElementsByTagName('select').item(0).getAttribute('id');
      var semesterID = semester.getAttribute('id');
      var totalCredits = first_child(semester).data;
      var semesterResult = document.getElementById("sem"+semesterID+"result");
      if (totalCredits == 60) {
        semesterResult.className = "green";
        first_child(semesterResult).data = totalCredits;
        unFlagError("sem"+semesterID+"result");
      }
      else if (totalCredits == 00) {
        semesterResult.className = "red";
        first_child(semesterResult).data = totalCredits;
        flagError("sem"+semesterID+"result", "Too few credits.");
      }
      else if (totalCredits >= 50 && totalCredits < 60) {
        semesterResult.className = "yellow";
        first_child(semesterResult).data = totalCredits;
        unFlagError("sem"+semesterID+"result");
      }
      else if (totalCredits <= 70 && totalCredits > 60) {
        semesterResult.className = "yellow";
        first_child(semesterResult).data = totalCredits;
        unFlagError("sem"+semesterID+"result");
      }
      else if (totalCredits > 70) {
        semesterResult.className = "red";
        first_child(semesterResult).data = totalCredits;
        flagError("sem"+semesterID+"result", "Too many credits.");
      }
      else if (totalCredits < 50) {
        semesterResult.className = "red";
        first_child(semesterResult).data = totalCredits;
        flagError("sem"+semesterID+"result", "Too few credits.");
      }
/*****************************************************************************/
/* This bit gets all requirements of the module and checks to see if they are*/
/* filled, if not it flags an error.                                         */
/*****************************************************************************/
      var requiredModules = xmlDocument.getElementsByTagName('requires');
      if (requiredModules.length > 0){
        var alertMessage = "";
        for (var x = requiredModules.length-1; x >= 0; x--) {
          var returnedRequirement = first_child(requiredModules[x]).data;
          if (!checkExists(returnedRequirement,originalID)) {
            alertMessage += returnedRequirement+":";
          }
        }
        if (alertMessage != "") {
          flagError(originalID,formatMessage(alertMessage))
        }
```

```
      }
/*****************************************************************************/
/* This bit deals with the chunk of xml which looks like this:               */
/* <requiredBy id="CSB1005C">                                                */
/*   <module>CSB1038C</module>                                               */
/* </requiredBy>                                                             */
/* <requiredBy id="CSB1039C">                                                */
/*   <module>CSB1038C</module>                                               */
/* </requiredBy>                                                             */
/* <requiredBy id="CSD1005C">                                                */
/*   <module>CSB1006C</module>                                               */
/*   <module>CSB1038C</module>                                               */
/* </requiredBy> this is for module with code CSB1038C.                      */
/* What we need to do is grab all the "requiredBy" elements:                 */
/*****************************************************************************/
      var requiredBys = xmlDocument.getElementsByTagName('requiredBy');
/*****************************************************************************/
/* Check to see if there are requiredBy's                                    */
/*****************************************************************************/
      if (requiredBys.length > 0) {
/*****************************************************************************/
/* If there are we need to loop though them                                  */
/*****************************************************************************/
        for (var x = requiredBys.length-1; x >= 0; x--) {
/*****************************************************************************/
/* We need to get the id of the module which requires the id of out module   */
/*****************************************************************************/
          var requiredBy = requiredBys[x].getAttribute('id');
/*****************************************************************************/
/* If it exists on the page                                                  */
/*****************************************************************************/
          if (checkExists(requiredBy)) {
/*****************************************************************************/
/* Get the id of the select which has it                                     */
/*****************************************************************************/
            var requiredByID = checkExists(requiredBy);
/*****************************************************************************/
/* Then we need to hold all of the modules from the xml which the module is  */
/* said to require and initialise a variable which will tell us if the       */
/* requirements are filled                                                   */
/*****************************************************************************/
            var requiredByModules =
               requiredBys[x].getElementsByTagName("module");
            var requirementsFilled = true;
/*****************************************************************************/
/* Then we need to loop though the page checking to see if they are there    */
/*****************************************************************************/
            for (var y = requiredByModules.length-1; y >= 0; y--) {
/*****************************************************************************/
/* Get the module that is required                                           */
/*****************************************************************************/
              var requiredByModulesCode =
                 first_child(requiredByModules[y]).data;
/*****************************************************************************/
/* If it doesn't exist                                                       */
/*****************************************************************************/
              if (!checkExists(requiredByModulesCode)) {
/*****************************************************************************/
/* add one to out testing variable                                           */
/*****************************************************************************/
```

```
                requirementsFilled = false;
              }
            }
            if (requirementsFilled == true) {
              unFlagError(requiredByID);
            }
          }
        }
      }
/*************************************************************************/
/* This bit deals with the chunk of xml which looks like this:           */
/* <mightHaveBeen id="CSB1003C">                                         */
/* </mightHaveBeen>                                                      */
/* <mightHaveBeen id="MSB1016">                                          */
/*   <whichIsRequiredBy id="CSD1002C">                                   */
/*     <module>MSB1016</module>                                          */
/*     <module>CSB1006C</module>                                         */
/*   </whichIsRequiredBy>                                                */
/*   <whichIsRequiredBy id="CSB1027C">                                   */
/*     <module>MSB1016</module>                                          */
/*     <module>CSB1039C</module>                                         */
/*   </whichIsRequiredBy>                                                */
/*   <whichIsRequiredBy id="MSB1013">                                    */
/*     <module>MSB1016</module>                                          */
/*   </whichIsRequiredBy>                                                */
/*   <whichIsRequiredBy id="CSB1002C">                                   */
/*     <module>MSB1016</module>                                          */
/*   </whichIsRequiredBy>                                                */
/*   <whichIsRequiredBy id="CSD1046C">                                   */
/*     <module>MSB1016</module>                                          */
/*     <module>CSB1006C</module>                                         */
/*   </whichIsRequiredBy>                                                */
/* </mightHaveBeen>                                                      */
/* Which is generated by the following call:                             */
/* GET http://camshag.34sp.com/7/php/checkValue.php?requestString=select4:-> */
/* CSB1038C:0:0:0:CSB1038C:CSB1039C:CSB1005C:CSB1030C:CSB1003C:MSB1016:->    */
/* CSB1038C:CSB1039C:CSB1005C:CSB1030C                                   */
/*************************************************************************/
      var mightHaveBeens = xmlDocument.getElementsByTagName('mightHaveBeen');
      if (mightHaveBeens.length > 0) {
        for (var x = mightHaveBeens.length-1; x >= 0; x--) {
          var whichIsRequiredBys =
             mightHaveBeens[x].getElementsByTagName('whichIsRequiredBy');
          if (whichIsRequiredBys.length > 0) {
            for (var y = whichIsRequiredBys.length-1; y >= 0; y--) {
              var whichIsRequiredBy =
                 whichIsRequiredBys[y].getAttribute('id');
              if (checkExists(whichIsRequiredBy)) {
                var whichIsRequiredByID = checkExists(whichIsRequiredBy);
                var whichIsRequiredByModules =
                   whichIsRequiredBys[y].getElementsByTagName("module");
                var alertMessage = "";
                var requirementsFilled = true;
                for (var z = whichIsRequiredByModules.length-1; z >= 0; z--) {
                  var whichIsRequiredByModulesCode =
                     first_child(whichIsRequiredByModules[z]).data;
                  alertMessage += whichIsRequiredByModulesCode+":";
                  if (!checkExists(whichIsRequiredByModulesCode)) {
                    requirementsFilled = false;
                  }
```

```
                }
                if (requirementsFilled == false) {
                  flagError(whichIsRequiredByID,formatMessage(alertMessage));
                }
              }
            }
          }
        }
      }
/*************************************************************************/
/* The end!                                                              */
/*************************************************************************/
      isWorking = false;
    }
  }
}
/*************************************************************************/
/* This function takes one parameter (The id of the select) and returns the  */
/* "Uncle" DOM node to that select. Because we're using a table for layout we*/
/* can be assured that there is only 2 nodes (TDs) for the parent (a TR) thus*/
/* the reference returned is cool!                                       */
/*************************************************************************/
function getUncle(id) {
  var pos = document.getElementById(id);
  var par = pos.parentNode;
  return node_before(par);
}
/*************************************************************************/
/* This function takes the id of the select, creates a DOM node with the     */
/* error image, finds the selects uncle and if there isn't already an error  */
/* message there puts one there. This will be updated in order to place code */
/* for the error pupup later on.                                         */
/*************************************************************************/
function flagError(id, message) {
  var uncle = getUncle(id);
  if (!first_child(uncle)) {
    var errorImg = document.createElement('img');
    errorImg.setAttribute('alt','error');
    errorImg.setAttribute('src','./images/error.gif');
    if (message != null) {
      errorImg.onmouseover = function(){ ddrivetip(message); };
      errorImg.onmouseout = function(){ hideddrivetip(); };
    }
    uncle.appendChild(errorImg);
  }
}
/*************************************************************************/
/* We know that there can only be one image in the uncle's collection of     */
/* child nodes so this just deletes the first child of the uncle. Dead simple*/
/* really but it doesn't need to be overly complex does it?              */
/*                                                                       */
/*************************************************************************/
function unFlagError(id) {
  var uncle = getUncle(id);
  if (first_child(uncle)) {
    uncle.removeChild(first_child(uncle));
  }
}
/*************************************************************************/
/* This function takes both the value and id and checks to see if the value  */
```

```
/* repeated in any other select other than the one that is given to it. If it*/
/* is repeated then true is returned. Otherwise false is returned. The loop  */
/* reduces the processing by getting the length of the array just once.       */
/*****************************************************************************/
function checkExists(value,id) {
  var selectObjects = document.getElementsByTagName("select");
  for (var x = selectObjects.length-1; x >= 0; x--) {
    if (selectObjects[x].value != "S" && selectObjects[x].value != "0") {
      if (selectObjects[x].value == value) {
        if (id == undefined) {
          return selectObjects[x].id;
        }
        if (selectObjects[x].id != id) {
          return true;
        }
      }
    }
  }
  return false;
}
/*****************************************************************************/
/* This function formats the alert text so that the string looks correct in  */
/* terms of placement of commas and ampersands.                              */
/*****************************************************************************/
function formatMessage(alertMessage) {
  alertMessage = alertMessage.slice(0,alertMessage.length-1);
  messages = alertMessage.split(":");
  alertMessage = "";
  for (var x = messages.length-1; x >= 0; x--) {
    if (x == 0) { alertMessage += getTAP(messages[x])+" required."; }
    else if (x == 1) { alertMessage += getTAP(messages[x])+" &amp; "; }
    else { alertMessage += getTAP(messages[x])+", "; }
  }
  return alertMessage;
}
function getTAP(module) {
  var optionObjects = document.getElementsByTagName("option");
  for (var x = optionObjects.length-1; x >= 0; x--) {
    if (optionObjects[x].getAttribute('value') == module) {
      var holdingText = first_child(optionObjects[x]).data;
      var holdingPlace = optionObjects[x].parentNode.getAttribute('id');
      break;
    }
  }
  var fragment = holdingText.split("(");
  var title = fragment[0];
  var semester = Math.ceil(holdingPlace.slice(6)/7);
  var year = Math.ceil(semester/2);
  var semester = semester%2;
  if (semester == 0) { semester = 2; }
  returnedMessage = title+" &#40;Year "+year+", Semester "+semester+"&#41;";
  return returnedMessage;
}
```

3.7 dhtmltooltips.js

```
/***********************************************
 * Cool DHTML tooltip script- © Dynamic Drive DHTML code library
 * (www.dynamicdrive.com)
 * This notice MUST stay intact for legal use
 * Visit Dynamic Drive at http://www.dynamicdrive.com/ for full source code
 ***********************************************/
var offsetxpoint=20   //Customize x offset of tooltip
var offsetypoint=-20  //Customize y offset of tooltip
var ie=document.all
var ns6=document.getElementById && !document.all
var enabletip=false
if (ie||ns6)
  var tipobj=document.all? document.all["dhtmltooltip"] :
     document.getElementById? document.getElementById("dhtmltooltip") : ""
function ietruebody(){
  return (document.compatMode && document.compatMode!="BackCompat")?
     document.documentElement : document.body
}
function ddrivetip(thetext, thecolor, thewidth){
  if (ns6||ie){
    if (typeof thewidth!="undefined") tipobj.style.width=thewidth+"px"
    if (typeof thecolor!="undefined" && thecolor!="")
      tipobj.style.backgroundColor=thecolor
    tipobj.innerHTML=thetext
    enabletip=true
    return false
  }
}
function positiontip(e){
  if (enabletip){
    var curX=(ns6)?e.pageX : event.clientX+ietruebody().scrollLeft;
    var curY=(ns6)?e.pageY : event.clientY+ietruebody().scrollTop;
    //Find out how close the mouse is to the corner of the window
    var rightedge=ie&&!window.opera?
       ietruebody().clientWidth-event.clientX-offsetxpoint :
       window.innerWidth-e.clientX-offsetxpoint-20
    var bottomedge=ie&&!window.opera?
       ietruebody().clientHeight-event.clientY-offsetypoint :
       window.innerHeight-e.clientY-offsetypoint-20
    var leftedge=(offsetxpoint<0)? offsetxpoint*(-1) : -1000
    //if the horizontal distance isn't enough to accomodate the width
    //of the context menu
    if (rightedge<tipobj.offsetWidth)
      //move the horizontal position of the menu to the left by it's width
      tipobj.style.left=ie?
         ietruebody().scrollLeft+event.clientX-tipobj.offsetWidth+"px" :
         window.pageXOffset+e.clientX-tipobj.offsetWidth+"px"
    else if (curX<leftedge)
      tipobj.style.left="5px"
    else
      //position the horizontal position of the menu where the mouse
      //is positioned
      tipobj.style.left=curX+offsetxpoint+"px"
    //same concept with the vertical position
    if (bottomedge<tipobj.offsetHeight)
      tipobj.style.top=ie?
        ietruebody().scrollTop+event.clientY-
```

```
        tipobj.offsetHeightoffsetypoint+"px" : window.pageYOffset+e.clientY-
        tipobj.offsetHeight-offsetypoint+"px"
    else
      tipobj.style.top=curY+offsetypoint+"px"
    tipobj.style.visibility="visible"
  }
}
function hideddrivetip(){
  if (ns6||ie){
    enabletip=false
    tipobj.style.visibility="hidden"
    tipobj.style.left="-1000px"
    tipobj.style.backgroundColor=''
    tipobj.style.width=''
  }
}
document.onmousemove=positiontip
```

Little Bits of Changes:

Comparing traditional web applications with applications built using the AJAX paradigm in the context of a module choice milieu within an academic community.

Dominic Myers