School of Electronics and Computer Science
Faculty of Engineering, Sciences and Mathematics
University of Southampton

Amit Shah (ams401)
Donna Crawford (dc899)
Chris O'Neill (ckjon101)
Dave Newman (drn101)

December 16, 2004

Accounting and Billing Software for Wolfson Electronic Business Centre

Project Supervisor: Dr. Tim Chown
Second Examiner: Dr. Michael R. Poppleton

A group design project report submitted for the award
Master of Engineering in Computer Science

Abstract

The Wolfson Electronic Business Centre currently uses a complicated spreadsheet to record their client information. The aim of this project is to engineer a complete system to aid the management process. The most appropriate tools and techniques were selected from those currently used in industry.

A strong relationship was forged with the customer through regular interaction. The use of an iterative prototyping model made it possible to handle the changing requirements that stemmed from this rapport. Constant user aided testing provided feedback that allowed the development of a highly customised solution.

The project's complicated processing algorithms were presented to the customer in a simple, intuitive interface. The result was a PHP web-based system with an associated MySQL database, which is under consideration for deployment by WEB Centre.

Contents

Chapter 1

Introduction

1.1 Project Brief (Donna Crawford)

The goal of the project is to engineer back-end software for the management of Wolfson Electronic Business Centre (WEB Centre) customer information. The system will be expected to store details such as customer contact information, records of hosting services operated, domains owned for customers and the time spent implementing and maintaining the services used by each client. It is also necessary that the data can be viewed and updated by key personnel and for the software to support itemised billing for consultancy and hosting services.

1.2 Objectives (Amit Shah)

The main tasks for this project were established through discussion with the WEB Centre administrator and consisted of:

1. Management of client and staff details
2. Records of services available and which clients use those services
3. Facility to track staff and consultancy hours spent on each service
4. Automated invoice generation and covering letters
5. IP and domain name management
6. Fully functioning software for possible incorporation to the WEB Centre
7. Integration with the ECS finance department Honours system

The project ultimately aims to automate the majority of day-to-day tasks associated with the managerial side of the business and to make the information easily accessible, searchable and maintainable.

1.3 Introduction to WEB Centre (Amit Shah)

> “The Wolfson Electronic Business Centre (WEB Centre) is an industrial unit hosted on premises within the Department of Electronics and Computer Science at the University of Southampton.”

The WEB Centre specialises in serving commercial customers within the local region including Hampshire, Dorset and Sussex, but also provides a platform for University research “spin-off ventures” to have an electronic presence in the market place.

There are a variety of services available including database-driven applications, online ordering, graphics design, WAP-enabled technology and black box hosting. These can be implemented and supplied on Unix or Windows systems. The WEB Centre provides advice for both upon client request. Prices vary depending on type of service, and discounts are given for registered charity organisations.

The WEB Centre also offers specialised consultancy in fields such as IPv6, knowledge representation systems and agent-based computing solutions. Assistance is provided for the deployment of the anti-virus and spam filtering software included in the email hosting package.

Chapter 2

Analysis and Design

2.1 Current WEB Centre Model (Amit Shah)

The WEB Centre has a single administrator who manages their clientele using a basic Excel spreadsheet. The flat file contains information regarding client contact details, services purchased and domain renewal information. The data is input manually and believed to be up-to-date, although no guarantee can be made. This method of management was sufficient when WEB Centre began, but clientele growth and expansion of the services provided has led to a complicated spreadsheet susceptible to human error.

Consider domain renewal in December for example, where the administrator must manually search the spreadsheet (see Appendix A) for the addresses about to expire. Clients who have not paid for another year need to be notified; the remainder are renewed. If the scenario is extended to consider a period of three months and the small file of ten clients is increased to two hundred, the workload for the administrator will increase dramatically. In this realistic situation it is highly possible that a client could be missed or a reminder not sent. This can be complicated further by actual human business interactions where a client may wish to renew their domain and database for example, but not their extra five POP email accounts. The WEB Centre must cancel services the client is not paying for without forgetting those scheduled for renewal.

To summarise, the current system has many drawbacks:

- Information is difficult to locate: Excel has limited searching capabilities
- No record of assigned IP addresses: Impractical to generate a complete allocation listing for submission to the Internet governing bodies
- Very time-consuming to isolate an available IP address: Administrator has to ping[1] each known address in turn
- Extremely slow and complex process to find which domains require renewal in any given month
- Generating invoices to pass to the finance department is inefficient when manually browsing the Excel spreadsheet

The aforementioned reasons clearly demonstrate why the WEB Centre urgently needs an update to their business management software.

[1]PING - Packet InterNet Groper. An Internet utility used to check the connection with another site [4].

2.2 Functionality Specification (Dave Newman)

The project objectives (see Section 1.2) were abstracted to produce four use case diagrams (see Appendix B). When constructing these diagrams it became apparent that five user levels were required:

1. Super administrator
2. Administrator
3. Systems Staff
4. Finance Staff
5. Client

Access to functionality is reduced through each level of the hierarchy; these user levels were consistent throughout the project.

The use case diagrams inevitably changed over the course of the project as the regular discussions with the customer revealed modified requirements. A finalised set of use case diagrams (see Appendix C) details the revised definitions of functionality and user levels.

With the assistance of the use case diagrams, the derived functionality specification below was used as a reference throughout the system development to ensure correct implementation. The specification is appended with the user level required where appropriate:

1. General Requirements
    - (a) Accessible to authorised personnel and clients only
    - (b) Functionality must be restricted dependent on the user level
2. Client details management
    - (a) Add a new client to the system [Super administrator]
    - (b) Edit the contact details [Finance staff and above, excluding systems staff]
    - (c) Change password
        - i. Change own password (with old password confirmation) [Client Only]

        ii. Change any client password (without old password confirmation) [Administrator and above]

    (d) View own contact details [Client only]

    (e) View all client contact details [Finance staff and above]

    (f) Search for a client's contact details [Finance staff and above]

    (g) Make notes on a client [System Staff and above]

3. Staff details management

    (a) Add a new staff member to the system [Administrator and above]

    (b) Add a new administrator to the system [Super Administrator]

    (c) Edit staff member details [Administrator and above]

    (d) Edit administrator details [Super Administrator]

    (e) Edit own details [Finance staff and above]

    (f) Change password

        i. Change own password (with old password confirmation) [Finance staff and above]

        ii. Change staff member password (without old password confirmation) [Administrator and above]

        iii. Change administrator password (without old password confirmation) [Super Administrator]

    (g) View a mapping of staff to clients [Finance staff and above]

    (h) View a log of updates to the database [Administrator and above]

    (i) View all staff details [Finance staff and above]

    (j) Search for a staff member's details [Finance Staff and above]

4. Services management

    (a) Add a new service [Super Administrator]

    (b) Edit a service's details [Administrator and above]

    (c) View a list of services offered

        i. Charity and commercial rates displayed [When not logged in]

        ii. Charity, commercial or ECS rate displayed (dependent on client account type) [Client only]

        iii. Charity, commercial and ECS rate displayed [Finance staff and above]

   (d) Add a new service to a client [Administrator and above]
   (e) Modify a client's service [Administrator and above]
   (f) View own purchased services [Client only]
   (g) View all client purchased services [Finance staff and above]
   (h) Make a note on a purchased service [Finance staff and above]

5. IP/DNS features
   (a) Allocate IP address to client [Administrator and above]
   (b) Add new IP address or range of addresses that can be allocated [Super administrator]
   (c) Remove IP address or range of addresses that can be allocated [Super administrator]
   (d) View a list of available IP addresses [Administrator and above]
   (e) View all IP address records [Administrator and above]
   (f) View listing of all IP addresses with the number of domains/customers allocated to each [Administrator and above]
   (g) View DNS hostname(s) associated with each IP address [Administrator and above]

6. Staff Timesheets
   (a) Submit own timesheets
      i. As consultancy for a client [System Staff and above]
      ii. As general infrastructure for WEB Centre [Staff and above]
   (b) Edit own timesheets [System Staff and above]
   (c) Search through own timesheets [System Staff and above]
   (d) View own timesheets [System Staff and above]
   (e) View all timesheets [Administrator and above]
   (f) Edit all timesheets [Administrator and above]

7. Client Invoices
   (a) Generate invoices [Administrator and above]
   (b) Edit payment on invoices [Administrator and above]
   (c) Search for invoices [Finance staff, Administrator and above]

(d) View all invoices [Finance staff, Administrator and above]

(e) View own invoice history [Client only]

(f) Save invoices in PDF format [Client, Finance, Administrator and above]

Once a listing of functionality was completed a feasibility study was carried out on every item to determine whether it was achievable within the time constraints of the project. One such item was the generation of DNS configuration files, currently written by hand. The feasibility study revealed that every file was so unique it was near impossible to write a function that would save any significant time (see Section 3.6.2). It was therefore agreed with the customer that this area of functionality should not be implemented and time should be spent on other more practical aspects of the project instead.

This method of negotiating priorities based upon feasibility and user-importance was achieved throughout the project life cycle. An additional factor that affected decisions was the extent to which adding the feature in question would speed up the tasks in the current model. Keeping track of timesheets for example, was originally quite difficult and time-consuming for the customer as all timesheets needed to be collated and then input into a spreadsheet. Once a timesheet is submitted to the new system the money owed to staff members is calculated and the appropriate amount automatically charged to the clients where applicable.

The customer was particularly interested in the implementation of advanced features, mainly those associated with IP and DNS configurations. The listing of available IP addresses was intended to make the client-IP allocation more straightforward and to justify applications for new IP addresses from the relevant organisations such as RIPE[2]. A Domain Name Server (DNS) lookup table was another desirable feature, that would check the domain names associated with IP addresses to check consistency.

To summarise, each functionality had to be prioritised in relation to importance to the customer and feasibility so the restricted project time could be appropriately allocated.

[2]The RIPE NCC (Rseaux IP Europens Net Coordination Centre) is an independent, non-profit membership organization that supports the infrastructure of the Internet through technical co-ordination in its service region [9].

2.3 Concept Development (Donna Crawford)

The next stage was to consider the nature of the solution. It had been ascertained that WEB Centre staff and clients are likely to use Windows, Linux or Macs and therefore the finished product had to be cross-platform. The main options available under these constraints consisted of building a Java based or web based application.

One of the key requirements of the project was for data to be viewed and updated by key personnel. This ruled out the use of an isolated flat-file system since text files running on a stand-alone machine would be neither portable nor particularly flexible. To make the system accessible remotely, employing a back-end database appeared to be a suitable alternative, especially for a company that had access to these resources.

The functionality had to extend to WEB Centre's clients as well as their staff and therefore distributing a Java application to allow users to view their account details would not have been practical. Within industry it is common to offer this functionality in the form of a server-side web-based solution [2]. The advantages of this would be ease of delivery, customisation and maintainability. The system could be viewed from any location without the need for any pre-installed software, information and operations could be updated to meet changing requirements as data would not have to be static, and users would always have access to the latest product.

By modelling the project as an interactive website with an associated database the majority of WEB Centre's day-to-day managerial and financial tasks could be automated in a highly versatile and easy-to-use format.

2.4 Development Tools (Donna Crawford)

The development tools considered were Active Server Pages .net (ASP.net), Java Server Pages (JSP) and Personal Home Pages (PHP) for the main body of the project along with MySQL and Microsoft's SQL Server for the database.

ASP.net offers multi-language support for a wide range of compiled languages such as Microsoft Visual Basic and C, and even scripted languages such as VBScript and Python. The advanced debugging features, numerous "bells and whistles" and object-oriented nature each counted highly in its favour. Despite these factors ASP.net was not deemed the most suitable choice. The first concern was that it offers poor memory management where efficiency had to be a priority. Secondly, it had potential to be the least secure since it requires Internet Information Server (IIS), which has a reputation for vulnerabilities [1].

Java Server Pages on the other hand is multiplatform and supports ORACLE, MySQL and SQL Server databases. It combines Java and HTML in a manner that offers good separation of development roles, isolating the HTML code for the graphical layout of the website from the system software itself. It is scalable and thus suitable for a medium to large sized business. On the downside, it would have had the slowest response times out of the three options, which becomes an important factor when considering the number of complex database queries and calculations to be employed.

The scripting language PHP is also multiplatform and very similar in style and syntax to C, Java and Perl due to its origins. It also contains libraries to support several databases including Oracle, Sybase, MySQL and ODBC. The code is usually embedded within the HTML page but it has the advantage of being easy to work with. PHP has the added benefit of enabling rapid development of web based solutions, making it ideal for use within the ten-week period [7]. Its PDF document generation and XML parsing libraries also looked promising for a project of this nature. Unlike ASP.net, PHP has excellent memory management and is interpreted by the server much faster than JSP.

Overall PHP was the natural choice, not only because of several advantages it had over the alternatives but primarily due to the team's previous experience with that language. Each member of the group had completed the e-Business module and was familiar with the "LAMP" package i.e. working with the combination of Linux, Apache, MySQL and PHP. By selecting a known language the project start-up time was reduced from the outset: an important factor for a ten-week implementation.

2.5 Software Technique (Donna Crawford)

The project development had several layers of requirements:

- A basic layer of the necessary elements to be used regularly;
- The intermediate, non-essential tasks that would be more difficult to code but certainly useful;

- And the advanced options that although would prove challenging to implement would extend the system further still and make it highly versatile.

For this reason, iterative development appeared to be the best software engineering approach in terms of taking account of repeated periods of production, testing, regression testing and delivery.

The implied structure of regular prototyping also meant that customer feedback could be gathered at the weekly progress meetings, and therefore constant user acceptance testing could be performed. The project benefited because it was routinely checked against the specifications and problems could be located early on. However this technique often caused unexpected modifications to the user-requirements when the customer suggested extra functionality they had not considered originally. Therefore it was vital that the Gantt Chart (see Appendix E) was adhered to as closely as possible and time was built in for additional work.

2.6 Final Design (Chris O'Neill)

After completing the analysis of the system, establishing the tools to be used and the target system (see Section 2.4), it was necessary to plan the remainder of the project.

The design stated the system would be implemented using PHP to handle hypertext pre-processing and MySQL to manage the back-end database (see Section 2.3). Using the class diagram (see Appendix D) as a reference, tables were created to store client details, staff details, timesheet information, available services, purchased services and domain allocation. The DBMS supported a relational database that connected the various tables using primary keys to ensure the database schema adhered to Third Normal Form standards [11]. The following list provides a summary of each table that was created:

1. web_client — This table stores all the details related to a single client entity that may be either an individual or an organisation. There are fields containing contact information for addressing invoices and mailshots in addition to system information such as passwords and account type.
2. $web\_clientservice$ — This table provides a link between the $web\_client \longleftrightarrow web\_service$ tables. Each time a customer purchases a service (see Section 3.4) a

record will be inserted into this table detailing the costs involved and the start and end dates for this contract.

3. $web\_domains$ — Any domains that are hosted by the WEB Centre of behalf of their clients will be listed in this table. Each entry in this table contains a reference to a $web\_ipaddresses$ entry which then gives more information on which of the WEB Centre's server the domain is located.

4. $web\_ipaddresses$ — Every IP address that has been allocated to the WEB Centre by their ISP is in this table. Each record contains information regarding the type of hosting present at this IP address and whether not it is available to accept further domains.

5. $web\_permissions$ — Each of the permission levels in the WEB Centre system is entered in this table. This allows for expansion to include new levels of security at a future date. The permissions listed here are linked to the web_staff table to control each staff member's access (see Section 3.2)

6. $web\_popaccounts$ — Clients may purchase differing numbers of $POPaccounts$ when they order the email account service; this service a special case and has an additional table in the database to handle the extra information.

7. $web\_service$ — All services offered by the WEB Centre are listed in this table. Each service has 3 rates associated with it that will be shown to clients depending on their type. Staff members can modify the data in this table to introduce global pricing changes if necessary. A flag in this table allows services to be disabled without deleting them for ease of system management and to ensure the integrity of the database.

8. web_staff — Staff members are listed in this table, giving them a means of authentication to the system, contact details and a permission level (see Section 3.2). The staff listed here are linked to timesheets (see Section 3.5.1), purchased services and staff notes.

9. $web\_staffnotes$ — Staff members have the facility to make comments regarding clients and services that have been purchased. Records in this database consist of a textual comment field and an id field linking them to either a specific client or a specific purchased service.

10. $web\_timesheet$ — All work performed by a member of the WEB Centre staff will be listed in this table (see Section 3.5.1); it acts as a hub between the staff table web_staff and the purchased services table $web\_clientservice$ to show hours

worked for billing purposes (see Section 3.5.2). Each time work is carried out by a staff member they are given the option to write a brief note on the work performed.

In the same way that the class diagram provided a reference for the creation of the database, the use case diagrams facilitated the construction a series of skeleton web pages outlining the functionality of the final system. This allowed both the designers and the customer to discuss the functions available in each web page the system at a very early stage of the project. An advantage of this method of development is that the purpose of each page would be well defined and easily translatable to a section of the functional specification. Those pages that performed similar functions could also be identified early and a library of common functions was developed.

The pages were divided into core functionality and additional features (based on the functional specification in Section 2.2) as follows:

1. Core functionality
   (a) $Login/Logout$ — This basic functionality is required to authenticate both staff members and clients to the system. Authentication is in the form of a user password combination that will compared to the data stored in the database (see Section 3.2).
   (b) $ServicesOffered$ — Displays a list of the services that are currently available for purchase by clients. Services that have been disabled will only be shown on this page if the user is logged in as with administrative access. The information shown on this page will vary depending on access level (see Section 3.4.3).
   (c) $ClientAccounts$ — Provides an administrator with the option to create a new client account or search the database for existing clients. Once located, the administrator will be able to modify the client's record if desired (see Section 3.4.1).
   (d) $WebsiteOptions$ — Miscellaneous features are provided here, giving users the option to change the password they use to authenticate themselves. Administrators have the ability to view a transaction log of all operations performed by staff on the database (see Section 3.4.2).

   (e) $StaffMembers$ — Similar in design to the client accounts section, these pages give administrators the ability to add and manage staff members. The ability to view which staff members have had dealings with which clients in the past is also available here.

2. Intermediate features

   (a) $StaffTimesheets$ — A required feature of the website that ties staff members, hours worked and purchased services, this section will be the most heavily used area of the site (see Section 3.5.1). Staff members can submit timesheet information and administrators can view the activity of the staff. The information stored in this section forms part of that used when invoices are generated to bill clients (see Section 3.5.2).

   (b) $ClientInvoices$ — An administrative feature used to keep tracking of the WEB Centre's finances. The pages in this section handle information exchanged in both directions: invoices to be sent to clients can be generated and payments received from clients can be entered into the database (see Section 3.5.2).

3. Advanced features

   (a) $IPAllocation$ — This feature allows an administrator to track all IP addresses allocated to the WEB Centre. Administrators can use the functions available here to determine which addresses are available for use by new clients or clients who have purchased new hosting services. The facility exists to generate a report detailing the full IP allocation listing for all addresses leased by the WEB Centre.

   (b) $DNSRecords$ — These pages provide the functionality to generate DNS server configuration information for use with the BIND DNS server. This feature enables administrators to maintain integrity between the database of hosted domains and the DNS configuration files.

   (c) $HONOURsChecking$ — Client contact details change from time to time and this feature provides the functionality to perform a complete comparison between the data stored in the WEB Centre database and the Finance Department's contact database. Any inconsistencies will be reported to an administrator for further attention.

A Gantt chart was created to handle time management across the ten weeks of the project (see Appendix E). Core functionality and intermediate features were given the maximum priority with additional aspects being implemented in later weeks. It was known at this point that it would most likely be used in a real business environment and therefore work could continue beyond the requirements of the Group Design Project.

Chapter 3

Implementation of Design

3.1 System Setup (Donna Crawford)

It was vital to set up the system in a manner that would provide solid foundations for the remainder of the project. This initially involved ensuring the file space was publicly viewable and then creating page holders containing purely a title and short description for each web page (see Section 2.6).

The next step was to write two plain-text files comprising of the header and footer HTML information common to all pages. The technique avoided repetition and allowed easy editing of the core functionality. The page holders were then updated with a pair of PHP *include* commands at the top and bottom of each file in order to import the code from the header and footer files respectively.

The skeleton of the system also required the addition of a simple navigation menu to provide access to all pages and demonstrate the clear structure of the product. The preliminary layout was intentionally simple (see Section 2.6) and consisted of a centered table with WEB Centre's company banner at the top and two columns on the next row for the menu and the page content.

Finally, with the file system in place the database could be initialised with the information collected in the class diagram (see Appendix D). At this point the freeware tool phpMyAdmin[1] was installed to provide a visual interface for constructing the database tables and displaying the data graphically. The alternative was to enter this information via typing command line[2] instructions into a text prompt, which was not considered as effective. With this solid foundation in place, work could begin on the next phase of development.

3.2 User Levels and Security (Donna Crawford)

The Use Case diagrams (see Appendix C) were used to illustrate the relationships between the four staff levels and the system functionality. This information subsequently had to be applied within the software itself to restrict user access where necessary. In order to implement these constraints member verification, validation and login sessions

[1]PHPmyAdmin is a software tool providing rapid development of MySQL databases: http://www.phpmyadmin.net/

[2]MySQL provides a basic command line interface where SQL queries and other commands can be submitted.

were essential.

The two methodologies considered were *cookies* and PHP *sessions* whereby each valid user is assigned a unique id that, depending on the approach used, could be stored either in a cookie or propagated in the URL. Sessions ensure the data is available even when cookies are disabled but they entail using long URLs that cannot be bookmarked, have to accommodate users who experiment with editing the session id and they tend to utilise more of the server's resources. Cookies on the other hand are very easy to use and store session-persistent data on the visitor's computer to keep the server load to a minimum. Users who block cookies will have problems logging in; this issue is resolved by providing support documentation.

The login system first prompts the user for the client or staff username and password via an HTML form. It is then checked against the client and staff databases for matching details. If any are found a cookie is saved on the user's computer containing an MD5[3] encrypted combination of their private $user\_id$, $account\_type$ and $username$. Even if the cookie content is decrypted the resulting data would not contain sufficient information to allow someone to log in; this is because the username is not stored in a recognisable format and the password is omitted. Invalid or partial cookies are immediately nulled by the system.

Once a user has successfully logged in, code placed in the header file checks their permission level and sets a series of boolean values accordingly:

- $loggedin
- $client
- $financial staff
- $staffplus (i.e. systems staff and above)
- $adminplus
- $superadmin

These booleans are convenient for avoiding code repetition in the body of the program, and as quick abbreviations they also reduce the code-length whilst increasing clarity.

[3]Message Digest 5 is a one-way encryption algorithm developed by Prof. Ronald L. Rivest: see RFC1321 for further details.

To avoid any malicious interference, attempts to pass values to the pages directly for example, every variable used is initialised as null.

Another method of bypassing the system might be to attempt to load any $include$ pages of imported code manually. To ensure that is never an option, $htaccess$ file permissions were configured to block the loading of certain web pages and folders server side. The $.htaccess$ file is a web server configuration file that can be set to restrict access or redirect requests for certain files to a $403ForbiddenAccess$ or a $404PageNotFound$ error page respectively.

The system had to connect to the database with suitable security. To reduce exposure, all references to the MySQL database's single access name and password had to be stored in one isolated place. PHP variables do not exist outside of functions they are embedded within. Therefore it made sense to store the connection-details inside a function located in a randomly named file. The function is designed to take a query, submit it to the database and return the result. Visitors to the site would not be able to discover the folder or file name since PHP commands are run at the server and thus the code and connection-details are always kept private. To maintain complete security on this sensitive data the file and folder permissions were set to be hidden to the public and only accessible from the server.

The user levels and software security allowed the development of a cleaner navigation menu by hiding links for pages the user had been blocked from viewing.

3.3 User Interface (Donna Crawford)

Designing the user interface comprised of a series of decisions concerning navigation, layout, scale and colour scheme. Whilst planning these sections one of the most important factors to focus on was the necessity for the project to support multi-platform cross-browser usage on Windows, Linux and Mac computers. The range of popular browsers selected for testing purposes included Internet Explorer, Netscape, Opera, Mozilla, Konqueror and Safari.

When initially considering menu navigation, research on accepted formats was performed. It was found that menus were placed either on the top or left hand side of most websites; Sub-menus were found in drop down bars under their relevant headings in the left hand menu or in a separate right hand section [5][8]. The top menu approach was dismissed as it was more suitable in systems with fewer pages. Drop down menus would have been too time-consuming due to the difficulties with implementing reliable code to run on each of the platforms mentioned previously. It had thus been determined the main part of the menu had to appear in the left hand pane, and that the sub menu would be on the right hand side (see Figure 3.1).

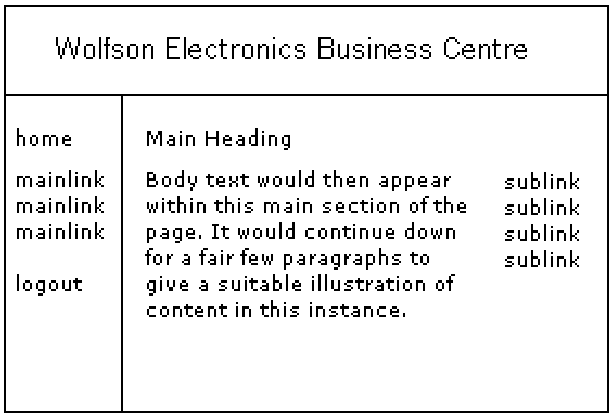

Wolfson Electronics Business Centre
home
mainlink
mainlink
mainlink
logout
Main Heading
Body text would then appear within this main section of the page. It would continue down for a fair few paragraphs to give a suitable illustration of content in this instance.
sublink
sublink
sublink
sublink

Figure 3.1: Right hand sub-menu system

This style had shown promise but was declined because the width of the main body could not support the inevitably large tables of data associated with database-driven applications. The solution was to place all the menu links in the left hand column. Since the list would consist of around thirty page entries it made sense to only show the relevant sub-links (see Figure 3.2). After minor modifications the result was a menu system that was intuitive and easy to use.

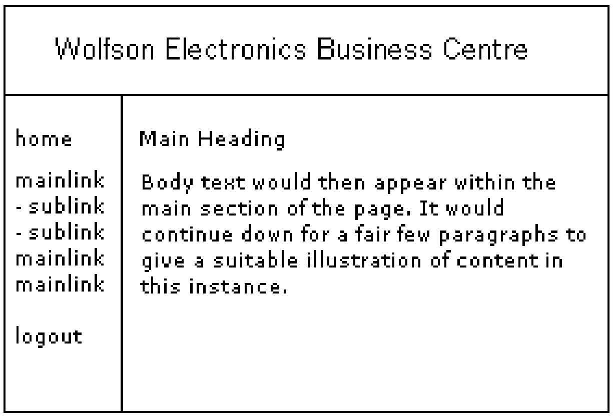

Wolfson Electronics Business Centre
home
mainlink
- sublink
- sublink
mainlink
mainlink
logout
Main Heading
Body text would then appear within the main section of the page. It would continue down for a fair few paragraphs to give a suitable illustration of content in this instance.

Figure 3.2: Left hand sub-menu system

Regarding the size and scale of the solution, most monitors can support screen resolutions of 1280×1024, 1280×960 or 1024×768 pixels on the high end of the market or either 800×600 or 640×480 pixels on the low end of the spectrum [12]. To avoid restricting any users, the system was intended to support resolutions as low as 800×600.

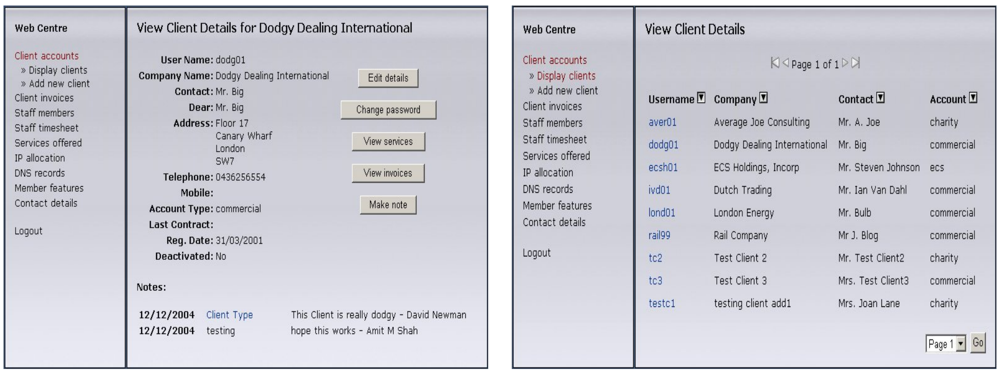
It would not have been worth planning for anything smaller though, because 640×480 was the standard for Microsoft Windows '95 released nine years ago. The width of the site was designed to fit with these figures, leaving enough room to display the vertical scrollbar so the pages could be easily browsed from top to bottom.

Finally, the colour scheme had to be fairly neutral for a business whilst still providing enough contrast to be clearly visible on the client's laptop computer. Laptop screens are known for their issues where colours can appear dull and sometimes even indistinguishable from other tones if the screen is tilted back to a slightly incorrect angle or if the monitor is viewed at a non-perpendicular angle [6]. After considerable experimentation black, metallic silver, electric blue and a contrasting burgundy brown were chosen to offer a sharp palette.

3.4 Basic Functionality

3.4.1 Profile Management (Amit Shah and Dave Newman)

The client and staff profile management were implemented first in accordance with the agreed customer specification (see Section 2.2) and to allow the creation of records to be used by the more advanced functionality. The left-hand side of Figure 3.3 shows the details of a client that is displayed when they view their profile; a staff member can view this information by selecting a client from the listings shown on the right-hand side.



View Client Details for Dodgy Dealing International

User Name:	dodg01
Company Name:	Dodgy Dealing International
Contact:	Mr. Big
Dear:	Mr. Big
Address:	Floor 17 Canary Wharf London SW7
Telephone:	0436256554
Mobile:	
Account Type:	commercial
Last Contract:	
Reg. Date:	31/03/2001
Deactivated:	No

Notes:

12/12/2004	Client Type	This Client is really dodgy - David Newman
12/12/2004	testing	hope this works - Amit M Shah

View Client Details

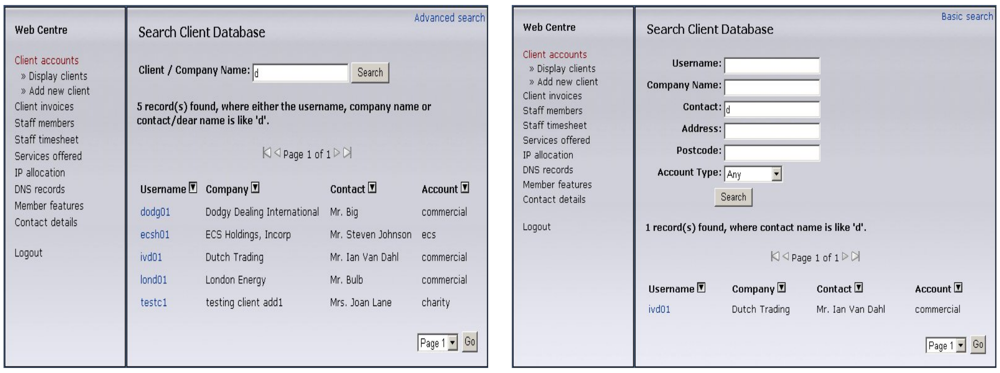
Page 1 of 1

Username	Company	Contact	Account
aver01	Average Joe Consulting	Mr. A. Joe	charity
dodg01	Dodgy Dealing International	Mr. Big	commercial
ecsh01	ECS Holdings, Incorp	Mr. Steven Johnson	ecs
ivd01	Dutch Trading	Mr. Ian Van Dahl	commercial
lond01	London Energy	Mr. Bulb	commercial
rail99	Rail Company	Mr J. Blog	commercial
tc2	Test Client 2	Mr. Test Client2	charity
tc3	Test Client 3	Mrs. Test Client3	commercial
testc1	testing client add1	Mrs. Joan Lane	charity

Figure 3.3: Client Profile and Client Listing

The expansion of WEB Centre leads to an obvious need for client and staff search facilities and both basic and advanced functions were implemented. The basic search uses a single entry field to match either the client/company name or the staff username.

The advanced search enables a more specified search to be carried out based upon client account-types and address details, or email information and access privileges for staff (see Figure 3.4). The query results are displayed in a similar manner to the full listings and can thus also link to individual profile details.



Search Client Database

Client / Company Name: d

5 record(s) found, where either the username, company name or contact/dear name is like 'd'.

Page 1 of 1

Username	Company	Contact	Account
dodg01	Dodgy Dealing International	Mr. Big	commercial
ecsh01	ECS Holdings, Incorp	Mr. Steven Johnson	ecs
ivd01	Dutch Trading	Mr. Ian Van Dahl	commercial
lond01	London Energy	Mr. Bulb	commercial
testc1	testing client add1	Mrs. Joan Lane	charity

Search Client Database

Username:	
Company Name:	
Contact:	d
Address:	
Postcode:	
Account Type:	Any

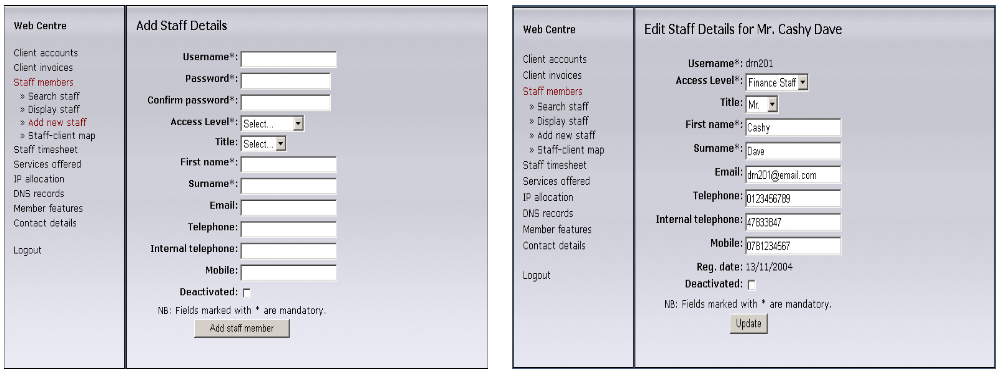
1 record(s) found, where contact name is like 'd'.

Page 1 of 1

Username	Company	Contact	Account
ivd01	Dutch Trading	Mr. Ian Van Dahl	commercial

Figure 3.4: Basic and Advanced Searching Facilities

Once the ability to search and view accounts was complete, the mechanism to add and edit records was implemented. The order of tasks meant that direct results could be obtained during development. The appearance of the add and edit forms are very similar (see Figure 3.5) and use the $POST$ method within HTML forms to submit data. The input is error-checked to ensure all mandatory fields are entered and that each field is in the appropriate format. Correct data is submitted to the database and the user is informed of the success. Incorrect data however, redisplays the form with an appropriate error message. For data consistency reasons usernames are not editable after the user account has been created.



Add Staff Details

Username*:	
Password*:	
Confirm password*:	
Access Level*:	Select...
Title:	Select...
First name*:	
Surname*:	
Email:	
Telephone:	
Internal telephone:	
Mobile:	
Deactivated:	☐

NB: Fields marked with * are mandatory.

Edit Staff Details for Mr. Cashy Dave

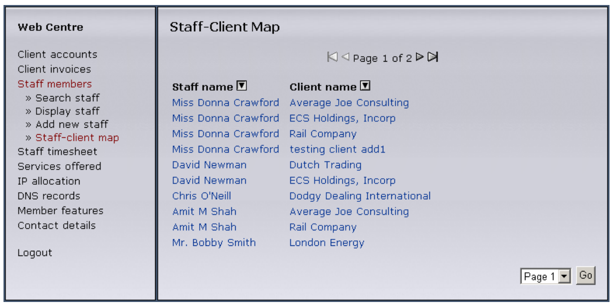
Username*:	drn201
Access Level*:	Finance Staff
Title:	Mr.
First name*:	Cashy
Surname*:	Dave
Email:	drn201@email.com
Telephone:	0123456789
Internal telephone:	47833847
Mobile:	0781234567
Reg. date:	13/11/2004
Deactivated:	☐

NB: Fields marked with * are mandatory.

Figure 3.5: Add and Edit Pages

The client profile page links to functionality associated with the particular client. Staff members with appropriate privileges can edit client details as well as view the client's purchased services and invoices. In addition the staff can make and edit notes on a client, but can only view those submitted by lower level staff. Each note has three sections: the date it was submitted or last edited, the title of the note and the note body, appended with the author.

To help the WEB Centre keep track of the staff working on each client account, a unique staff-client mapping was made. This is possible because every staff timesheet is referenced to a particular client. In order to keep the mapping current, only timesheets submitted within the previous twelve months are used (see Figure 3.6).



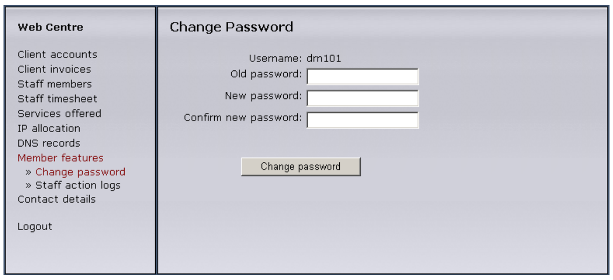
Staff-Client Map

Staff name	Client name
Miss Donna Crawford	Average Joe Consulting
Miss Donna Crawford	ECS Holdings, Incorp
Miss Donna Crawford	Rail Company
Miss Donna Crawford	testing client add1
David Newman	Dutch Trading
David Newman	ECS Holdings, Incorp
Chris O'Neill	Dodgy Dealing International
Amit M Shah	Average Joe Consulting
Amit M Shah	Rail Company
Mr. Bobby Smith	London Energy

Figure 3.6: Staff-Client Mapping

A requirement of both the staff and client account management is the ability to change passwords. The security for this functionality is very important as the entire system is dependent on the confidentiality of a user's password, thus a successful attack could result in an unauthorised user being considered as a Super Administrator.

The functionality specification (see Section 2.2) stated that all users should be able to change their passwords but only certain users should be able to change the password's of others. Confirmation of the old password is required to protect against an unauthorised user changing the logged-in users password (see Figure 3.7).



Change Password

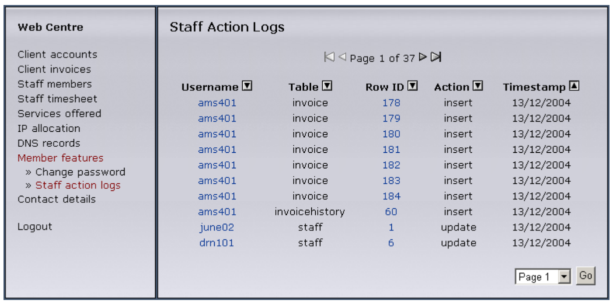
Username: drn101
Old password:
New password:
Confirm new password:

Figure 3.7: Change Password

The implementation of functionality to change another users password was more difficult. A potential security hole was from URL[4] injection because the username is passed in the URL header when legitimately changing the password. This means that an unauthorised user could pass a modified URL when attempting to gain access to another's change password page. To protect against this, the change password page checks the user has sufficient access permissions to change the password declared in the header. This is particularly important as changing passwords for users other than yourself does not require old password confirmation.

3.4.2 User Editing History (Dave Newman)

For database maintenance purposes, a log is made in the $web\_editinglog$ table of any changes or updates carried out by the users of the software: this information is viewable to those at the adminstrator level only. Each record in the log table contains a link to the profile page of the user who made the ammendment so that the adminstrator can quickly identify or contact that person (see Figure 3.8).



Staff Action Logs

Page 1 of 37

Username	Table	Row ID	Action	Timestamp
ams401	invoice	178	insert	13/12/2004
ams401	invoice	179	insert	13/12/2004
ams401	invoice	180	insert	13/12/2004
ams401	invoice	181	insert	13/12/2004
ams401	invoice	182	insert	13/12/2004
ams401	invoice	183	insert	13/12/2004
ams401	invoice	184	insert	13/12/2004
ams401	invoicehistory	60	insert	13/12/2004
june02	staff	1	update	13/12/2004
drn101	staff	6	update	13/12/2004

Figure 3.8: Log listing

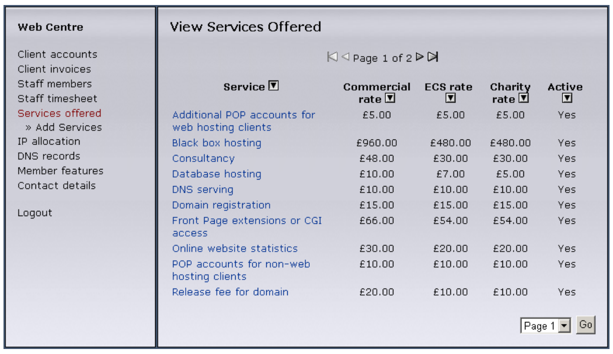
3.4.3 Services (Dave Newman)

Services was broken down into two sections: a section for the services offered by WEB Centre and another for those purchased by a client. Both of these sections needed the facility to view, add and edit records.

It was more logical for the system to display services offered before enabling a client to purchase one and was thus implemented in this order, despite the $web\_services$ table already being populated by phpMyAdmin. Figure 3.9 shows that the layout remains

[4]Universal Resource Locator

consistent with the staff and client listings and also has the ability to link to a more detailed view of the service offered.



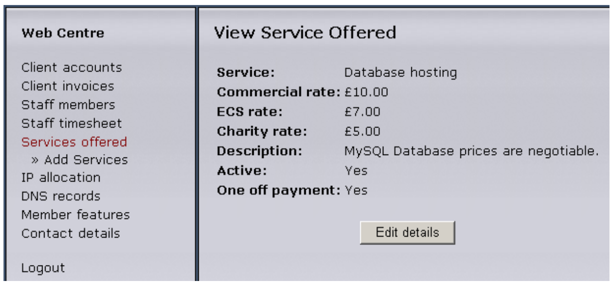
View Services Offered

Page 1 of 2

Service	Commercial rate	ECS rate	Charity rate	Active
Additional POP accounts for web hosting clients	£5.00	£5.00	£5.00	Yes
Black box hosting	£960.00	£480.00	£480.00	Yes
Consultancy	£48.00	£30.00	£30.00	Yes
Database hosting	£10.00	£7.00	£5.00	Yes
DNS serving	£10.00	£10.00	£10.00	Yes
Domain registration	£15.00	£15.00	£15.00	Yes
Front Page extensions or CGI access	£66.00	£54.00	£54.00	Yes
Online website statistics	£30.00	£20.00	£20.00	Yes
POP accounts for non-web hosting clients	£10.00	£10.00	£10.00	Yes
Release fee for domain	£20.00	£10.00	£10.00	Yes

Figure 3.9: Service Listing

The full view of offered service details is again similar to the client and staff profile views (see Figure 3.10) with an option to edit details for users with appropriate access privileges. Both the offered services listing and its profile vary depending on who is viewing the page. When a user is not logged in only the active services with commercial and charity rates are displayed; otherwise the software recognises the client account type so only the designated rates are shown. Staff members, on the other hand, can view all three rates and whether the service is currently available. The price for a purchased service is set depending on the account type or using the flexible custom price field.



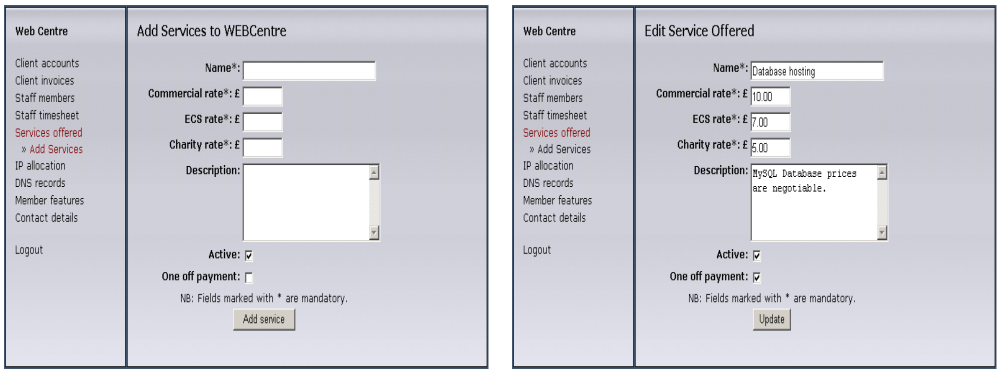
View Service Offered

Service:	Database hosting
Commercial rate:	£10.00
ECS rate:	£7.00
Charity rate:	£5.00
Description:	MySQL Database prices are negotiable.
Active:	Yes
One off payment:	Yes

Figure 3.10: Service Profile

A consistent style has been applied to the forms used when adding and editing services (see Figure 3.11). Unlike the staff and client profiles, editing a service offered allows the user to change the name of the service. The problem with data consistency does not apply here because the auto-incrementing *serviceid* is used as the primary key in database queries.

The implementation of the services purchased by a client followed a similar development schedule. A listing was generated that could be accessed from the client profile



Add Services to WEBCentre

Name*:	
Commercial rate*: £	
ECS rate*: £	
Charity rate*: £	
Description:	
Active:	☑
One off payment:	☐

NB: Fields marked with * are mandatory.

Edit Service Offered

Name*:	Database hosting
Commercial rate*: £	10.00
ECS rate*: £	7.00
Charity rate*: £	5.00
Description:	MySQL Database prices are negotiable.
Active:	☑
One off payment:	☑

NB: Fields marked with * are mandatory.

Figure 3.11: Service Adding and Editing Forms

pages (see Section 3.4.1), after which an option to purchase another service for that client was added. Each service also contains a link to its full details if the viewer has appropriate access privileges (see Figure 3.9).

Adding and editing of services purchased is more complex than services offered due to several special cases. When adding a new client-service for example, the number of $POPaccounts$ required is only necessary if the service opted for is either "Additional POP accounts for web hosting clients" or "POP accounts for non-web hosting clients" and should thus only be visible after the service selection has been made to prevent user confusion.

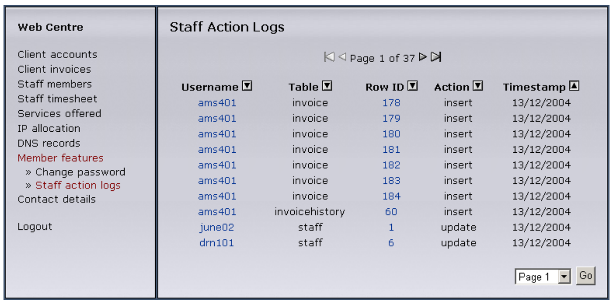
The services pages operate in a very similar way to staff and client pages with exception to the special cases in adding and editing. This has two important benefits: consistency throughout the sections makes the user interface more intuitive, and it also allows extensive code reuse that speeds up production.

Also in "view purchased service" there is a listing of notes made by staff on the particular purchased service. These notes offer the same functionality as that of client staff notes (see Section 3.4.1), therefore notes can be added, edited and deleted in the same way. The only difference is that when a staff-note record is stored in the database, the $clientserviceid$ is left blank and only the $clientid$ is completed.

3.4.4 Usability Features (Dave Newman)

Usability was a major issue of concern as the display content grew, and it became clear that intuitive methods were required to enable the user to find particular records. Paging of query results was one such method: the user defines the number of records per page in their customisable options.

Navigation through these pages was obviously required, in the form of buttons to step forward, backward, skip to first and skip to last. As there can be a large number of pages in a listing a "go to page" option was implemented to enable direct access to a known page (see Figure 3.12).



Staff Action Logs

Page 1 of 37

Username	Table	Row ID	Action	Timestamp
ams401	invoice	178	insert	13/12/2004
ams401	invoice	179	insert	13/12/2004
ams401	invoice	180	insert	13/12/2004
ams401	invoice	181	insert	13/12/2004
ams401	invoice	182	insert	13/12/2004
ams401	invoice	183	insert	13/12/2004
ams401	invoice	184	insert	13/12/2004
ams401	invoicehistory	60	insert	13/12/2004
june02	staff	1	update	13/12/2004
drn101	staff	6	update	13/12/2004

Figure 3.12: Paging

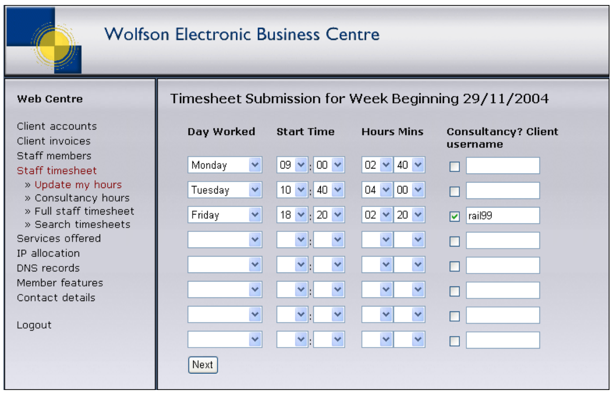
To find a particular record, the user can also sort the fields in the record listings both in ascending and descending order. This was implemented by passing two variables in the page header: the first specifies the sorting priority of the fields, the second is a bitmask defining how to order each field. A binary numeric system was used for the bitmasks so that every conceivable combination of individual field ordering could be achieved. The sorting is presented to the user in form of arrow buttons that depress upon selection.

3.5 Intermediate Functionality

3.5.1 Staff Timesheets (Donna Crawford)

The staff timesheets feature has several purposes. Primarily, it would be used by staff members to record the number hours they had worked either for specific clients in a $consultancy$ capacity or on the WEB Centre $infrastructure$. In order to submit a new timesheet the user is first prompted as to which week the entry will apply to. This information is presented in a drop down list containing the starting date for the current week, for example $Mon13/12/2004$, along with the dates for the four Mondays prior to that. The restrictions ensure that timesheets cannot be submitted for future times or sufficiently far in the past to cause interference with the payment of wages. The super administrator has the additional control of being able to alter this period from four weeks as he sees fit.

After the staff member has selected the correct week, a list of drop down boxes are displayed (see Figure 3.13) from which details of the hours worked can be input. If any times have already been submitted for that period these figures will appear above the list. The values required for the process include: the weekday name, start time, number of hours and minutes worked (where the smallest unit of time is twenty minutes), the clients username and specific *serviceid* if the work was consultancy related, and any suitable comments relevant to the task. The content of this form is then shown in a simple table format so it might be checked before being saved to the database. If any errors are present the user can easily return to the previous webpage and make any changes necessary.



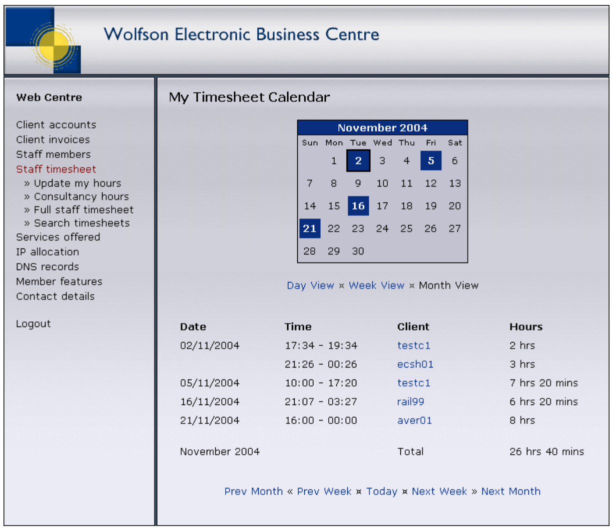
Timesheet Submission for Week Beginning 29/11/2004

Day Worked	Start Time	Hours Mins	Consultancy? Client username
Monday	09 : 00	02 40	☐
Tuesday	10 : 40	04 00	☐
Friday	18 : 20	02 20	✓ rail99
	:		☐
	:		☐
	:		☐
	:		☐
	:		☐

Figure 3.13: Timesheet Entry

Once a timesheet had been submitted, the result would be visible from the main Timesheet Calendar page. The layout was intended to depict the chosen month in a calendar style view (see Figure 3.14) since this appeared to be the most efficient method of providing an intuitive overview of the data. The currently selected date is outlined with a black border and any days worked for that particular month are highlighted with blue squares[5]. Situated below the calendar is a list of the hours worked on that day and the details for each entry; by default the page initially loads with the current date selected. Regarding the navigation of the system, clicking on any given day causes the timesheet for that period to be displayed and in conjunction with this, a set of links are used to change the selection period from a single day, to a week or even month view. The latter is particularly useful if the user needs to print a copy of their timesheet for any given month. Shortcut *previous* and *next* links have also been added to allow the user to

[5]A simple colour change might have proved difficult to detect on a laptop so highlighting was implemented instead (see Section 3.3).


Wolfson Electronic Business Centre

Web Centre

Client accounts
Client invoices
Staff members
Staff timesheet
» Update my hours
» Consultancy hours
» Full staff timesheet
» Search timesheets
Services offered
IP allocation
DNS records
Member features
Contact details

Logout

My Timesheet Calendar

November 2004

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

Day View ¤ Week View ¤ Month View

Date	Time	Client	Hours
02/11/2004	17:34 - 19:34	testc1	2 hrs
	21:26 - 00:26	ecsh01	3 hrs
05/11/2004	10:00 - 17:20	testc1	7 hrs 20 mins
16/11/2004	21:07 - 03:27	rail99	6 hrs 20 mins
21/11/2004	16:00 - 00:00	aver01	8 hrs
November 2004		Total	26 hrs 40 mins

Prev Month « Prev Week ¤ Today ¤ Next Week » Next Month

Figure 3.14: Timesheet Calendar

move through the system in units of weeks or months. Timesheet entries for individual days can also be edited by clicking on a small icon displaced when viewing that date.

The customer had naturally asked that consultancy hours should have to be approved by an administrator before being billed to clients. The Consultancy page thus lists all entries made by system staff categorised as unpaid and consultancy. The administrator can then view the details along with any comments made and choose to update the status to $approved$ or $rejected$, at which point it will be removed from this list.

The Full Staff Timesheet page requires a day, week or month to be entered and then presents a fully detailed listing of all staff members who worked during that period. This administration feature could be used to evaluate the most active staff members, assess which clients require the more support than others or even to check the system's calculations relating to staff pay.

The Search Timesheets function is another administrator level feature. It allows a variety of input to be specified in the search fields, including staff username, stall level, client username, a $searchfrom$ date, a $searchto$ date and comment text. Submitting the report produces a simple table of timesheet entries that match the criteria, using previous and next page links if appropriate.

The staff timesheet functionality contains a considerable amount of complicated programming, which ties in to several other areas of the project but it is vital in order to reducing the customer's current workload.

3.5.2 Invoice Generation (Amit Shah)

WEB Centre's main request was to have the new software to facilitate the generation of invoices for clients who owed money for services used. Upon customer discussion it became clear that the interface presented must be very simple with a single button to generate all, which presented a challenging task.

Generically speaking the solution was to calculate the total costs that every client had individually accumulated and to deduct any amounts previously paid. This meant tracking the period of time any service was used by a client via timestamps for the start and finish[6]. Computing the number of months a client had used a service was therefore simple, and the costs incurred were obtained by multiplying this by the monthly rate. There was a potential of redundancy however, because even in five years time the calculations would start from the $start\_date$ and re-calculate amounts that one already knew had been paid. The remedy to this was to store more recent timestamps when the client made a payment to make the due balance $£0$ (Equation 3.1).

$$\begin{aligned} recent\_start\_date &= recent\_start\_date + months\_paid\_for \\ months\_paid\_for &= \frac{amount\_paid\_for\_service}{service\_monthly\_rate} \end{aligned} \tag{3.1}$$

Services are not just monthly however, and adjustments must be made for one-off charges and payments. This was managed using an active status, whereby once the amount due was $£0$ the service was marked as inactive to the invoice generation system saving redundant processing. The other non-monthly service is consultancy hours, which was assigned a similar system of recording the last known amount of hours already paid for.

The manner in which these fields are used is key to the success of the invoice generation. To counter the numerous varieties of special cases that may occur in this area a rather elegant algorithm was formed:

The combined flexibility of the database and the algorithm meant that the user simply had to enter the invoice date and press the generate-all button. The software then calculates the balance due for all services used by each client regardless of expiry time or whether there were gaps in use, or even if clients used different dates for their operating periods.

[6]The current date was used if no finish date had been entered under the assumption that the service was still in use.

Algorithm 1 Algorithm used to generate invoices

```
1:  for each client do
2:      for each service that has not been paid for (regardless of whether they are in use
        at this time) do
3:          if one-off service then
4:              record the calculated amount due and the service name
5:          else if consultancy hours then
6:              use the most recent hours field and record the amount due, the hours being
                billed and service name
7:          else
8:              It is a monthly service so calculate the amount due from the most recent
                start date to either the finish date if it exists or the billing date if not. Record
                the amount due and the service name
9:          end if
10:         Create an invoice record with an itemised bill in CSV format
11:     end for
12: end for
```

The WEB Centre also introduced new special circumstances during the half-way point in the project whereby not only individual clients would be invoiced, but the ability to select a subset of clients would be appreciated. Fortunately the algorithm to generate invoices for all clients was based upon a loop that considered each client in turn and thus easily extensible to incorporate the new functionality.

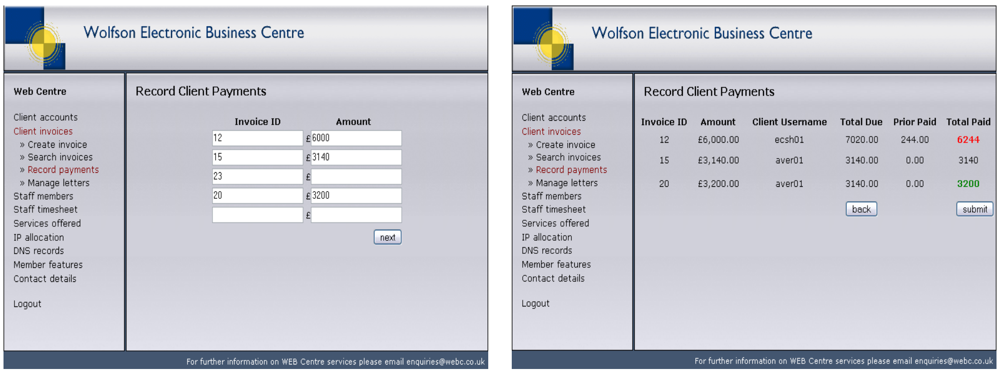
Every time a batch of invoices is generated, common attributes to each invoice are stored in an invoice history log. This log also contains the number of invoices generated and records the batch type[7]. Administrators can browse the invoice history to locate particular batches, or search for a particular invoice as appropriate.

When the invoices have been generated the user is presented with a summary that shows the number of invoices created and provides options to view, save or print all within the batch. Both the save and print facilities use a PDF format of the invoices to either store to the user's desktop or transmit to the user's printer (see Section 3.5.3). Upon viewing a batch each invoice is displayed one-at-a-time in an HTML format with browsing features (see Section 3.4.4) and the option to save or print just the single invoice. In addition, an itemised breakdown is included to reveal exactly where the costs were incurred.

Once the clients have received the generated invoices and corresponding covering letters the need to record payments becomes evident. The process is straightforward, and

[7]Batch type is single, subset or all depending on the option selected for the invoice generation.

has been designed to reduce human error that is induced during long periods of data entry. The user is presented with two fields: one for the invoice reference and one for the amount being paid. To avoid repetition and reduce the workload placed on the user an option is provided whereby they can select how many payments they wish to enter in one batch. It is not necessary to use all of the requested entry fields because the software only considers pairs of fields with data in both. Instead of allowing the user to submit the payments straightaway a next button displays a brief summary of client details, the amount due for the invoice and whether any prior payments have been made (see Figure 3.15).



Record Client Payments

Invoice ID	Amount
12	£ 6000
15	£ 3140
23	£
20	£ 3200
	£

Record Client Payments

Invoice ID	Amount	Client Username	Total Due	Prior Paid	Total Paid
12	£6,000.00	ecsh01	7020.00	244.00	6244
15	£3,140.00	aver01	3140.00	0.00	3140
20	£3,200.00	aver01	3140.00	0.00	3200

For further information on WEB Centre services please email enquiries@webc.co.uk

Figure 3.15: Making Invoice Payments

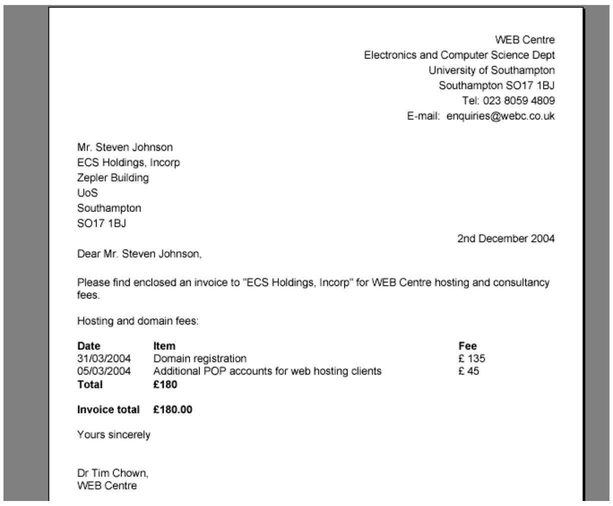
The absence of a known invoice payment in the confirmation screen indicates to the user that an entry field has been skipped and should thus go back and complete the details. Crucially the end column displays the amount now due for each invoice with positive discrepancies highlighted in green indicating possible overpay and negative discrepancies in red where money is still owed. This provides instant feedback for the user and the opportunity to rectify any input errors by selecting the back button. The web page remembers every data entry in the current batch including those not actually used in the case of an invoice reference with no payment amount or vice-versa to reduce user workload.

Upon submission of the confirmed payment batch the user is informed of the number of successful payment updates.

3.5.3 PDF Generation (Dave Newman)

A free PHP library for generating PDF (Postscript Data File) documents was found called FPDF[8]. Although this library is not as powerful or as quick at generation as commercial libraries, it was sufficient for the PDF generation tasks of the system.

The PDF invoice layout conforms to the existing WEB centre standard (see Figure 3.16). PDF documents can be generated for individual or batches of invoices. If a value for $invoicehist$ is passed in the URL header, (e.g. $pdfgen.php?invoicehist = 59$), then a PDF with a page for each invoice in that batch is generated. Alternatively, if a value for $invoice$ is passed (e.g. $pdfgen.php?invoice = 166$) then that specific invoice is generated. PDFs are generated by extracting the data from the invoice records and the associated $invoicehistory$ and client records; they can then be either saved to disk or printed out.



WEB Centre
Electronics and Computer Science Dept
University of Southampton
Southampton SO17 1BJ
Tel: 023 8059 4809
E-mail: enquiries@webc.co.uk

Mr. Steven Johnson
ECS Holdings, Incorp
Zepler Building
UoS
Southampton
SO17 1BJ

2nd December 2004

Dear Mr. Steven Johnson,

Please find enclosed an invoice to "ECS Holdings, Incorp" for WEB Centre hosting and consultancy fees.

Hosting and domain fees:

Date	**Item**	**Fee**
31/03/2004	Domain registration	£ 135
05/03/2004	Additional POP accounts for web hosting clients	£ 45
Total	**£180**	

Invoice total **£180.00**

Yours sincerely

Dr Tim Chown,
WEB Centre

Figure 3.16: Generated PDF

As all invoices are permanently stored in the database a PDF invoice can be generated whenever a user wishes. This safeguards against the loss of saved PDF files or printed hard copies.

[8]Free PDF can be found at http://www.fpdf.org/

3.6 Advanced Functionality

3.6.1 IP Allocation (Chris O'Neill)

Every computer connected to the Internet is allocated a unique IP address that can be used for identification. It is possible for a single host computer to be referenced by many IP addresses but each IP address must be allocated to one and only one host on the given network.

On a local network consisting of five hosts it is easy to ensure there are no IP address conflicts on the network. When dealing with a network with over one hundred connected hosts it becomes much more complicated to ensure that no clashes between IP addresses exist without keeping accurate assignment records. In the case of the Internet, there are entire organisations such as RIPE, whose sole task is to manage the assignment of IP addresses. ISPs must do the same for their customers, which includes the WEB Centre.

In the same way that a single host may be allocated multiple IP addresses, a single IP address can be host to many websites: this is called 'virtual hosting'. Maintaining accurate records of the customers' websites being hosted on each IP address is of the utmost importance to any hosting company and the WEB Centre is no exception.

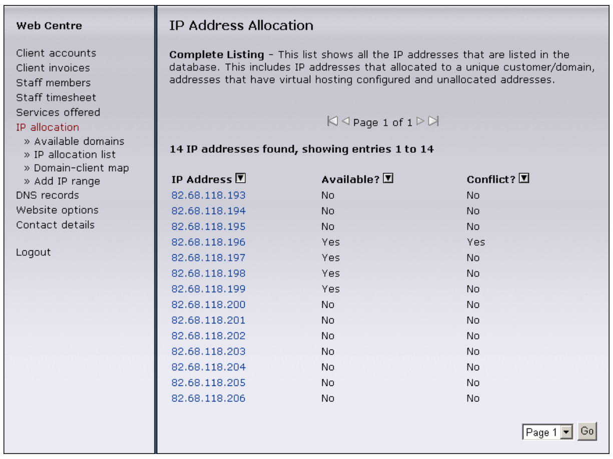
The system has been designed to be capable of storing a list of the IP addresses that have been assigned to the WEB Centre by their ISP. IP addresses can be selected and marked as configured for virtual hosting, in which case many domains can be hosted from that single IP address. This type of hosting is preferable for small, low traffic sites as it minimises IP address usage.

IP address allocation is handled using a single table in the database called $web\_ipaddresses$. This table contains a list of each IP address that has been allocated to the WEB Centre along with two flags. The field $conflict$ indicates whether or not this IP address is configured for virtual hosting and that conflicts are allowed because the host is capable of dealing with many domains at once. The field $available$ simply indicates whether or not this IP address is available to accept new domains. An address used for virtual hosting will continue to be available even after many domains have been hosted on it already; other addresses will be flagged as unavailable as soon as a single domain is hosted on them.

It is possible to mark an IP address as being reserved, in which case the domain that is hosted on the IP address will be marked as $internal$. Devices such as routers and WEB Centre servers should all be marked on this list as using IP addresses to allow for accurate record-keeping. This accurate IP address tracking will be important when the WEB Centre needs to request more IP addresses as the allocation organisations requires proof that ISPs have efficiently used a sufficient proportion of their current allotment.

The system includes web pages to facilitate the management of IP addresses for each of the purposes described above. IP addresses can be added to and removed from the database by specifying the network address (a base IP address) and the subnet mask[9]. This method of dealing with IP address ranges has been chosen because it is the standard technique for dealing with address ranges on all platforms.

A list of all IP addresses that the system is tracking can be generated in various formats, each tailored to a specific purpose. The standard view is used by administrators to display a list of IP addresses that have been allocated and which client(s) they have been allocated to (see Figure 3.17). The format allows the administrator to select a specific



IP Address Allocation

Complete Listing - This list shows all the IP addresses that are listed in the database. This includes IP addresses that allocated to a unique customer/domain, addresses that have virtual hosting configured and unallocated addresses.

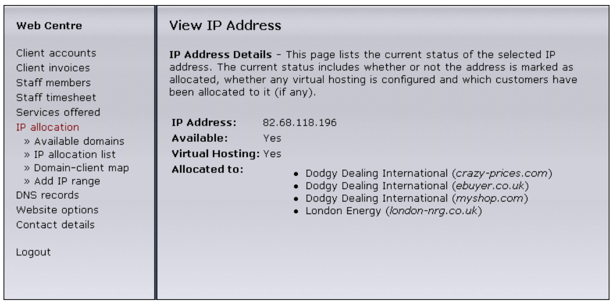
14 IP addresses found, showing entries 1 to 14

IP Address	Available?	Conflict?
82.68.118.193	No	No
82.68.118.194	No	No
82.68.118.195	No	No
82.68.118.196	Yes	Yes
82.68.118.197	Yes	No
82.68.118.198	Yes	No
82.68.118.199	Yes	No
82.68.118.200	No	No
82.68.118.201	No	No
82.68.118.202	No	No
82.68.118.203	No	No
82.68.118.204	No	No
82.68.118.205	No	No
82.68.118.206	No	No

Figure 3.17: IP Address Allocation Listing

IP address from the list and view extended information regarding it (see Figure 3.18). The information includes a list of the customers that are using this address and the domains that are being hosted on it. If the address is not marked for virtual hosting, there will only be a single entry in this list. An address marked for virtual hosting could potentially have any number of entries, but in reality the processing power of the server hosting the sites usually places an upper limit. Tracking the processing power of each server and recommending limits is currently beyond the scope of this project.

[9]The subnet mask specifies the allowed range of addresses starting at the base IP address



View IP Address

IP Address Details - This page lists the current status of the selected IP address. The current status includes whether or not the address is marked as allocated, whether any virtual hosting is configured and which customers have been allocated to it (if any).

IP Address: 82.68.118.196
Available: Yes
Virtual Hosting: Yes
Allocated to:

- Dodgy Dealing International (*crazy-prices.com*)
- Dodgy Dealing International (*ebuyer.co.uk*)
- Dodgy Dealing International (*myshop.com*)
- London Energy (*london-nrg.co.uk*)

Figure 3.18: Detailed View of IP Address Information

As mentioned earlier, should new IP addresses be necessary, the WEB Centre would be expected to demonstrate efficient usage of existing addresses. A list can be generated that details each IP address currently allocated to the WEB Centre with a short description of what it is used for i.e. internal usage or customer hosting. In the case of the latter, the number of domains hosted in conjunction with the hosted domain names will also be displayed.

IP addresses can be assigned on a per customer basis using the 'IP Address Assignment' page. This page allows an administrator to select a customer web service, stored in the table $web\_service$, and link it to an IP address by creating an entry in the $web\_domains$ table. In the case of an address marked as $internal$ a 255 character comment can be made to describe what the address is reserved for.

3.6.2 BIND/DNS Configuration (Chris O'Neill)

The Domain Name System (DNS) is used to convert between the domain names that typical users see when accessing a web site, www.webc.co.uk for example, and the IP address of the server hosting that service, 62.189.30.180 for example.

When a user attempts to access a given host, a DNS lookup request is sent to the user's ISP's DNS server(s). If this server does not know the answer to the request it will perform a lookup of its own on the root DNS servers for the $.uk$, $.co.uk$ and $.webc.co.uk$ domain. The result will be a nameserver that knows the IP address associated with every $.webc.co.uk$ host, including our initially desired www host. This server will then be contacted and queried for the answer to the initial request and the answer ($www.webc.co.uk \longleftrightarrow$ 62.189.30.180) will be returned to the user.

The most common DNS server software used in the world today is BIND (Berkeley Internet Name Domain), which is understandably the software used by the WEB Centre. The BIND configuration files must store a list of all the hostnames handled by the ISP running it along with the each associated IP address. At the time of writing, the WEB Centre has been allocated a range consisting of 256 IP addresses. Assuming every IP address had been assigned to a single domain and there was no virtual hosting in operation, the BIND configuration files would be storing 256 entries, each of which must be kept up to date at all times; the customer's site would become inaccessible otherwise.

At the request of the customer, the IP allocation data in the database could be used to generate the BIND DNS configuration files. This would ensure that the data was kept up to date with only a small amount of administrator intervention and would virtually guarantee data integrity between the database and the DNS server.

After some research had been conducted regarding the format of the BIND configuration files, it was determined that such a system would be possible provided more information was stored in the database. The additional information would need to include data regarding the specific hostnames assigned to each IP address rather than simply the domain hosted. The system would need to be flexible as the customer had warned of special cases, but the only way to add the level of flexibility desired would be to generate template configuration files and have the administrator edit them each time. Further discussions with the customer revealed that these special cases were so abundant that virtually every customer would require a change to their entries in some way. These modifications would have to be made to each customer every time the database was updated so the decision was made to abandon this feature as the primary goal of maintaining database $\longleftrightarrow$ configuration file integrity could not be achieved whilst providing the required level of flexibility. Appendix F shows the functionality that was achieved before work on this feature was discontinued.

3.6.3 Registrar Nameserver Configuration (Chris O'Neill)

In order to perform a DNS lookup correctly (see Section 3.6.2), the information at each stage must be accurate and up to date. There are two areas to check to ensure the information is current: one is the local data stored in the BIND configuration files and the other is the data held by the domain registrar. The data stored at the domain registrar, Nominet[10] for example, includes the nameservers that are listed as authoritative for

[10]Nominet UK registers all .uk domain names: see http://www.nominet.org.uk/

that domain. Every domain hosted by the WEB Centre should have the WEB Centre's BIND server IP address(es) listed as the authoritative servers for the domain.

Although the information stored in the local BIND configuration files is too highly customised to be generated automatically, it would be possible for the system to verify the data stored by the registrar via 'screen-scraping' the results of a registry $WhoIs$ query. Unfortunately, this idea was put forward by the customer in the later stages of the project so no development has taken place yet.

A page will be created that will allow the administrator to enter the current IP addresses of the WEB Centre nameservers. PHP code behind the page will then perform a $WhoIs$ query on each of the domains listed in the domains table $web\_domains$ and parse the resulting web page to determine the listed name servers.

The nameservers given by the $WhoIs$ query will be compared to the values entered by the administrator and the domain will be flagged as ok if the two match. If the two do not match, the administrator will be alerted that there is a problem and given more information about the error: the full $WhoIs$ query result.

A disadvantage of the 'screen-scraping' method is that even very subtle changes to the parsed web page can result in a failed 'read'. The goal is to develop a system that can intelligently read the information presented on the results page to find the nameserver IP addresses.

If this system does not prove successful, it may be possible to find a web service that will allow the queries to be performed faster and in a consistent format.

3.6.4 HONOURs System Integration (Chris O'Neill)

Discussions with the customer regarding invoicing (see Section 3.5.2) led to another potential project expansion regarding the HONOURs finance system. The contact information maintained by the finance department is stored separately to the information stored by the WEB Centre itself, which introduces the potential for data integrity and consistency problems.

It is not possible to directly access the financial database as this is stored in a proprietary format and may contain confidential information. Integration with this system requires the use of a ‘screen-scraping’ technique performed on a user-generated list of all WEB Centre customers.

A unique ID field related to each client in the HONOURs system could be stored in the WEB Centre’s main client database. Each client’s details could then be read from the screen-scraped $contact\_details$ report and compared to the details stored in the main database. Any conflicting data would be brought to the attention of an administrator or a member of the finance staff.

Chapter 4

Systems Testing Plan

4.1 Maintaining Standards (Chris O'Neill)

When dealing with a project of this scale it is important to maintain quality throughout. A high standard of code is important and rigorous testing must be carried out to ensure that the code is robust, especially given that the high profile of the website will reflect upon the WEB Centre.

Frequent code review sessions ensured the level of code produced was of the standard that would be required by the customer. The Fagan Inspection [3] process was used during the code review meetings in addition to the constant regression testing employed by individual members during development. Weekly prototype builds were demonstrated to the customer, providing regular user acceptance testing feedback; this allowed for a natural iterative design process where prototypes were produced and refined throughout the development cycle.

The customer's summary of their experience with the most recent prototype is extremely encouraging and is as follows:

> "The product appears to meet most of the discussed requirements, and our initial impression is quite positive. With the vendor having shown such enthusiasm in determining our needs to date, we are confident that the final product will be most satisfactory."

A full report of the customer's evaluation based on the most recent prototype can be found in Appendix H. The results of the in-house testing for this prototype include a series of figures to show the approximate page loading times (see Appendix G). This timing data was recorded using the development server and will vary from the performance experienced on the customer's production system; they will however, serve as a reference point for future releases and the final production system.

4.2 Final In-House Alpha Testing (Chris O'Neill)

The alpha testing phase will be carried out after all features have been implemented. Core and advanced functionality will be programmed by this time and this a comprehensive testing plan needs to be developed.

A check on quality will need to be performed on all code written by this point to ensure it is of the high standard expected by the customer (see Section 4.1). Comments should be placed at appropriate points in all code to allow for easy modification at a future date, possibly by a new developer. Each function that has been coded will be tested to ensure that it gives an appropriate response for the full range of possible inputs. The comments listed with each function will be compared to the output given to make certain the comments are accurate and up to date.

Once all functions have been individually tested, entire features will undergo the same examination. Each page contains a description of its purpose and the facilities it provides; this information will be used to perform a complete test of each page. It is imperative that all pages are fully tested using an extensive range of inputs as many pages will be accessible by clients of the WEB Centre. The development team must consider at all times during the alpha testing phase that this product will be used in a business environment and will be representative of the WEB Centre's quality of service.

4.3 System Installation and Beta Testing (Chris O'Neill)

The beta testing phase will take place after the system has successfully passed the alpha test and will take the form of a User Acceptance Test. The system has been constantly reviewed by the customer to ensure that the desired functionality has been implemented and this testing phase will be the conclusion to this testing. The customer will be comparing the finished system to the functional specification and testing accordingly.

There will be a period of parallel running where the new system will run alongside the existing flat-file system. The system will be set up on the customer's system using their hardware configuration to provide a test-bed as close to the final installed system as possible. If the results generated by the old and news systems differ then an investigation can be performed to determine which system generated the erroneous data and the reason why.

Once the system has passed the beta testing phase it will be made available to client's of the WEB Centre; for this reason it must be of the highest quality possible and all bugs must be removed. The system will be used for the invoicing of the WEB Centre customers and mistakes cannot be made when dealing with financial information. Reliability will also be tested to ensure the system is stable and resilient against hacking attempts.

Chapter 5

Team Management

5.1 Team Dynamics (Amit Shah and Dave Newman)

Unlike other teams, made up of friends through the use of the preferences form, our group consisted of people who had never previously worked together. The result was a lack of collective team focus that hindered the initial progress, which was not ideal with such a short timescale. Fortunately this soon changed as the group learnt of the common interests that had brought us together: web-based design.

The team decided to tackle the design of a database layout first by providing an initial foundation for the allocation of discrete tasks to be completed individually. After reviewing each other's strengths, individual roles became more apparent and thus the allocation of the aforementioned tasks became easier. Team members who were more experienced in web page generation created templates for others to use, learn and build upon. This method allowed an otherwise steep learning curve to become more manageable and enabled the team to progress collectively forwards and thus catch up on 'lost time'. The design of the site layout was again allocated to the team member who had the most experience in that field, so time was efficiently spent in terms of quality and productivity. Comments and feedback from the rest of the team were always welcome however obscure or insignificant they may have been.

Through drawing upon the strengths of each team member and because the project had discrete areas of implementation, the division and allocation of tasks was straightforward for an iterative prototype approach. The early part of the project life cycle was led by expertise and current skills so a high quality of foundation was set. The middle phase incorporated work completed in pairs and involved more collaboration, allowing each member to learn from an "expert" and ideas to be exchanged for larger segments of the project. The final phase was heavily dependent on individual team members being allocated sections to complete so that integration was consistent in that area (see Section 5.2 for more detail).

Roles were not constrained to implementation and code however, there were members within the team whose main strengths for this project were their literacy and concerns for customer satisfaction and usability. The final site for example, will adhere to rules dictated by the results of customer negotiation on the user interface.

5.2 Integration (Chris O'Neill)

Due to the nature of a website, a collection of pages divided into discrete sections, the workload was easily distributed across the members of the project according to abilities and specialties. The decision was made to avoid the use of a concurrent versioning system such as CVS for two primary reasons. Past experiences using similar systems had dissuaded some group members from using such systems and the extra time required to set up such a system would have reduced time available for coding. Each member of the team focused on their assigned pages and maintain their code in the best way they saw fit.

A global 'changelog' was kept on the website to keep other group members apprised of changes to any pages. In the event that changes to a global library, all group members would be notified in advanced to prevent any concurrent modifications that could result in a loss of code.

Weekly progress meetings ensured that code could be examined and quality-checked (see Section 4.1) but also allowed the team to provide progress updates. New tasks were assigned as those completed were marked off on the project Gantt chart.

5.3 Project Evolution (Chris O'Neill)

In addition to working on this project, each member of the group was also involved in various other projects. Not all members of the group selected the same module options and this meant that some members had to deal with deadlines from other projects during this WEB Centre project's life cycle. This eventuality was planned for from the onset of the project by creating a list of what other commitments each member of the project had and what other deadlines they would be required to meet. These deadlines could then be taken into account when designing the time management plan that is summarised in the initial Gantt chart.

Where possible, priority was given to this project, as it was by far the largest of those being worked on by any members of the group. Whilst ensuring that this project was given high-priority, care was taken by all members of the group to maintain a high standard across all the projects they were undertaking.

The initial project's time management plan was adhered to as strictly as possible because, as mentioned above, it was based on when group members were available and any deviation could potentially introduce large delays in the project. By the fourth week of the project, this strategy had been shown to work well as the project was very much on-track with each planned intermediate deadline being achieved.

A notable omission from the Gantt chart is the time required to plan and execute the various progress seminars during the course of the project. Work on these presentations was divided equally between group members, causing slightly less work than originally anticipated to be completed in affected weeks. This did not influence deadlines overall however as team members put in extra time to ensure weekly tasks were still completed on schedule.

At the time of the final progress seminar, the core project features were nearing completion. Based on customer feedback, addition core functionality that was not initially listed in the Gantt chart had also been implemented. The advanced features that were planned could not be implemented in time for this progress seminar and it was known that many could not be implemented in time for the final deadline. Each had been investigated by the group and discussed in detail with the customer at the weekly progress meetings and relative priorities were assigned to the advanced features and the core functionality. The decision was made to ensure the core functionality was completed by the time this report was compiled and to continue work on the advanced functionality afterwards. The system this project describes is required to be a complete, business-ready solution and quality management was placed above all other considerations.

The customer has been aware of (and advocating) the focus on quality management during the project and is aware that though this report will be compiled and submitted after only ten weeks of work, the project will be continued above and beyond the requirements for the Group Design Project.

Chapter 6

Conclusion and Future Work (Amit Shah)

A healthy relationship was forged with the customer, WEB Centre, through the use of weekly status meetings. This system of continuous project prototyping and user interaction provided a constructive opportunity for discussion throughout the development life cycle, ensuring the team maintained a strong understanding of the requirements. In addition, regular team meetings were scheduled to review the progress made and consider any queries or ideas that had arisen during the week.

The structure of the meetings was ideal for an iterative development model [A3] and meant that regular user-aided testing (UAT) was possible (see Section 4.1). By demonstrating the latest iteration of the product and recording the customer's feedback it was possible to appropriately modify the specification in preparation for the next prototype.

The team was required to produce a fully functioning system within a very short period; therefore the allocation of tasks was crucial to the success of the project. It was necessary for each team member to work on a section that would move the project forwards without impeding the progress of others. The team was able to manage the division of labor without any setbacks (see Section 4.1) and in cases where components were heavily dependent on each other it was ensured that the appropriate members collaborated (see Section 5.1).

In producing a complete system, the management of component integration is of vital importance to guarantee updates are not lost in the process. The team controlled versioning through a common system of record keeping and the assignment of independent tasks. This technique was implemented instead of using commercial software such as CVS, which was only possible due to the divisible nature of a web-based solution (see Section 5.2).

The combination of constant UAT, healthy customer liaison and strong team understanding resulted in a project that not only met, but exceeded the original specification. The management of system staff, services and invoices was achieved in addition to the initially requested accounting and billing capabilities (see Section 3.4.1). Intricate mechanisms were used to incorporate automated core features such as invoice generation and reverse DNS configuration (see Section 3.5.2 and Section 3.6.2); although complex in operation, they are presented to the user in a simple layout that is intuitive and easy to use (see Section 3.3). Another bonus to the web-based solution is that the system is not constrained to staff usage; clients can access the site with an appropriate user level (see Section 3.2) to view their own details, services and invoices, which utilises the data already stored. This strategy provides a platform for the clients to review their account details and inform the WEB Centre of any changes they would like made to their current hosting package or contact details.

The success of this group project is indicated by the WEB Centre's consideration to not only incorporate the web-based solution into their business but to also hire the team as software consultants to maintain the fully functioning system.

During development of the software both the team and the customer considered numerous methods of expanding the project. Functionality that passed the feasibility test was incorporated into a revised specification, but several promising ideas had to be categorised as impractical to implement within the ten-week period. This information was not discarded however, as certain elements could be added at a later stage and were considered as future work:

- Implementation of "add notes" or "add events" to any date within the timesheet calendar to record extra information such as meeting times or holidays
- Email notifications: new work available to staff, or service newsletters and price changes for clients
- Text reminders to staff about timesheet information

- RSS[1] news feeds concerning current offers or server status
- WAP support for the web-based software solution

[1]"Really Simple Syndication (RSS) is a lightweight XML format designed for sharing headlines and other Web content."[10]

Bibliography

[1] CERT/CC. Cert advisory ca-2002-09 multiple vulnerabilities in microsoft iis. URL http://www.cert.org/advisories/CA-2002-09.html, 2002.

[2] eBay Inc. Ebay uk — the uk's online marketplace. URL www.ebay.co.uk, Copyright ©1995-2004, 2004.

[3] Michael Fagan. Design and code inspections to reduce errors in program development. *IBM Systems Journal*, 15(3):182–211, 1976.

[4] C Hunt. *TCP/IP Network Administration*. O'Reilly, 1997.

[5] IBM. Ibm employment — uk. URL http://www-05.ibm.com/employment/uk/index.html, 2004.

[6] Portrait Displays Inc. The viewing angle problem. URL http://personalcomputing.portrait.com/us/products/lc_lcd_whitepaper.html ©1996-2004 Portrait Displays, Inc., 2004.

[7] GBdirect Ltd. Perl asp vbscript php jsp comparison: Comparing web scripting and website programming languages. URL http://training.gbdirect.co.uk/courses/php/comparison_php _versus_perl_vs_asp_jsp_vs_vbscript_web_scripting.html, 2004.

[8] Microsoft. Microsoft download center. URL http://www.microsoft.com/downloads/search.aspx?displaylang=en, 2004.

[9] RIPE NCC. About the ripe ncc. URL http://www.ripe.net/info/ncc/index.html, 2004.

[10] Web Reference. Introduction to rss. URL http://www.webreference.com/authoring/languages/xml/rss/intro/, 2004.

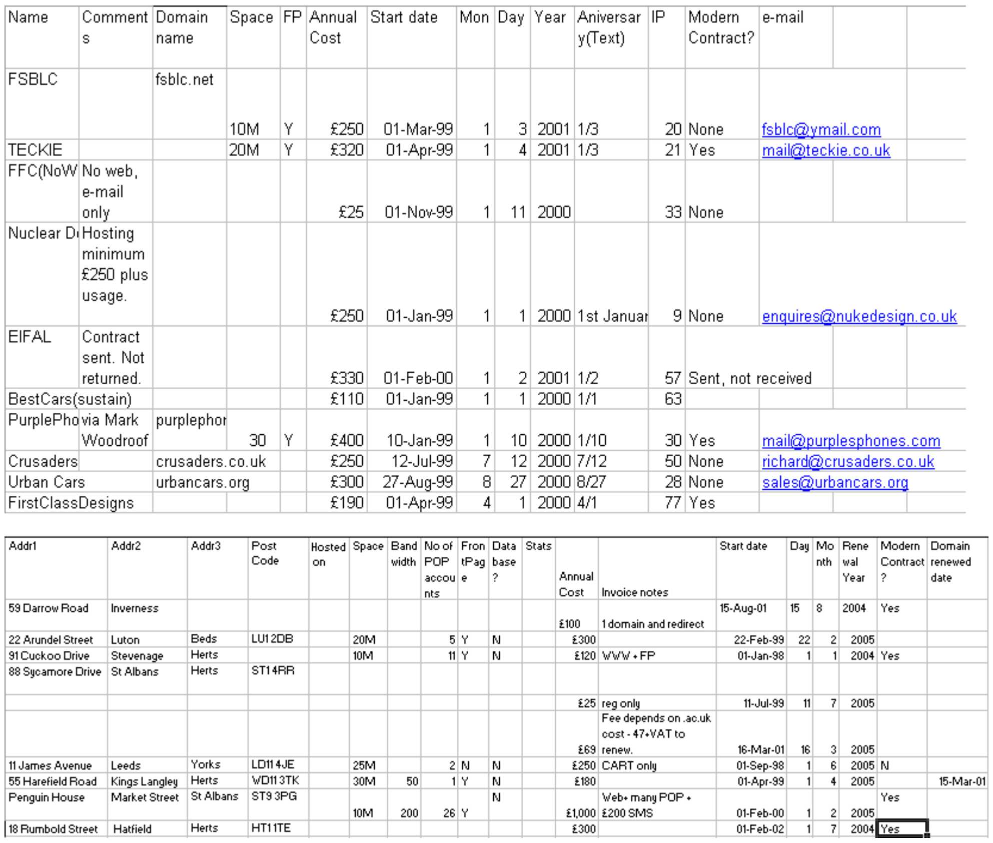
[11] L Ullman. *MySQL: Visual QuickStart Guide*. Peachpit Press, 2002.

[12] Chad Upsdell. Browser news: Statistics — trends — learn about trends in browsers, colour-depths, and resolutions. URL http://www.upsdell.com/BrowserNews/stat_trends.htm#res, 2004.

Appendices

Appendix A

Sample of Current WEB Centre Spreadsheet



Name	Comments	Domain name	Space	FP	Annual Cost	Start date	Mon	Day	Year	Aniversary(Text)	IP	Modern Contract?	e-mail
FSBLC		fsblc.net	10M	Y	£250	01-Mar-99	1	3	2001	1/3	20	None	fsblc@ymail.com
TECKIE			20M	Y	£320	01-Apr-99	1	4	2001	1/3	21	Yes	mail@teckie.co.uk
FFC(NoW	No web, e-mail only				£25	01-Nov-99	1	11	2000		33	None	
Nuclear D	Hosting minimum £250 plus usage.				£250	01-Jan-99	1	1	2000	1st Januar	9	None	enquires@nukedesign.co.uk
EIFAL	Contract sent. Not returned.				£330	01-Feb-00	1	2	2001	1/2	57	Sent, not received	
BestCars(sustain)					£110	01-Jan-99	1	1	2000	1/1	63		
PurplePho	via Mark Woodroof	purplephor	30	Y	£400	10-Jan-99	1	10	2000	1/10	30	Yes	mail@purplesphones.com
Crusaders		crusaders.co.uk			£250	12-Jul-99	7	12	2000	7/12	50	None	richard@crusaders.co.uk
Urban Cars		urbancars.org			£300	27-Aug-99	8	27	2000	8/27	28	None	sales@urbancars.org
FirstClassDesigns					£190	01-Apr-99	4	1	2000	4/1	77	Yes	

Addr1	Addr2	Addr3	Post Code	Hosted on	Space	Band width	No of POP accounts	FrontPage	Data base ?	Stats	Annual Cost	Invoice notes	Start date	Day	Month	Renewal Year	Modern Contract ?	Domain renewed date
59 Darrow Road	Inverness										£100	1 domain and redirect	15-Aug-01	15	8	2004	Yes	
22 Arundel Street	Luton	Beds	LU1 2DB		20M		5	Y	N		£300		22-Feb-99	22	2	2005		
91 Cuckoo Drive	Stevenage	Herts			10M		11	Y	N		£120	WWW + FP	01-Jan-98	1	1	2004	Yes	
88 Sycamore Drive	St Albans	Herts	ST1 4RR															
											£25	reg only	11-Jul-99	11	7	2005		
											£69	Fee depends on .ac.uk cost - 47+VAT to renew.	16-Mar-01	16	3	2005		
11 James Avenue	Leeds	Yorks	LD11 4JE		25M		2	N	N		£250	CART only	01-Sep-98	1	6	2005	N	
55 Harefield Road	Kings Langley	Herts	WD11 3TK		30M	50	1	Y	N		£180		01-Apr-99	1	4	2005		15-Mar-01
Penguin House	Market Street	St Albans	ST9 3PG		10M	200	26	Y	N		£1,000	Web+ many POP + £200 SMS	01-Feb-00	1	2	2005	Yes	
18 Rumbold Street	Hatfield	Herts	HT1 1TE								£300		01-Feb-02	1	7	2004	Yes	

Figure A.1: WEB Centre spreadsheet with dummy data

Appendix B

Initial Use Case Diagrams

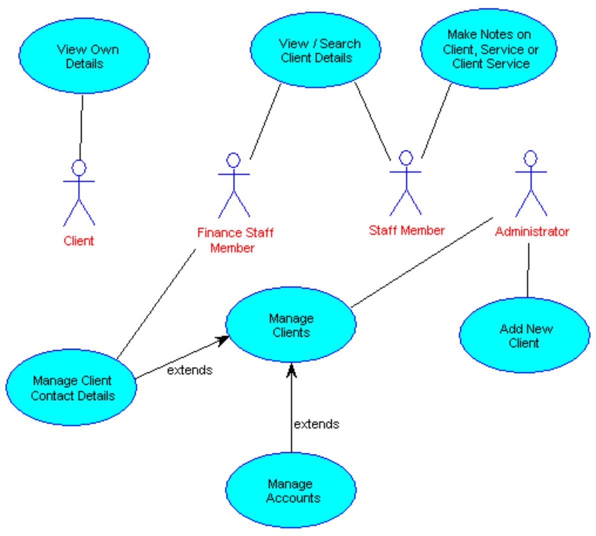


Figure B.1: Initial Client Use Case

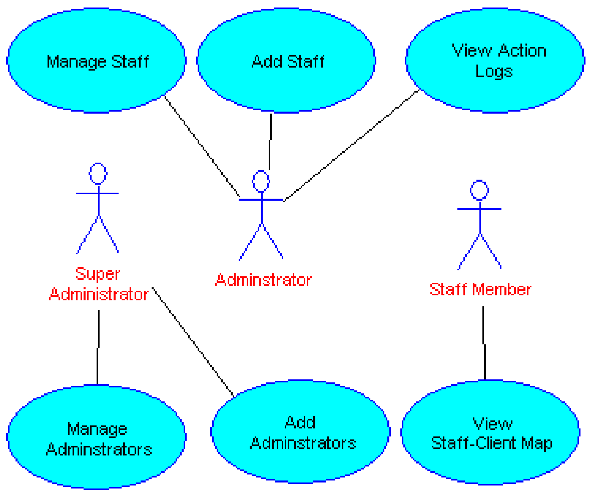


Figure B.2: Initial System Staff Use Case

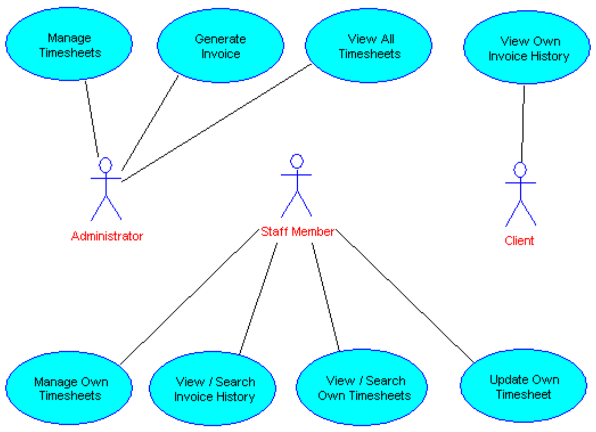


Figure B.3: Initial Finance Staff Use Case

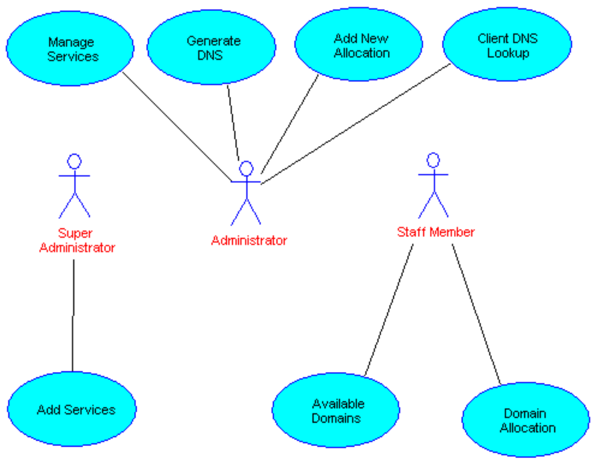


Figure B.4: Initial Services Use Case

Appendix C

Final Use Case Diagrams

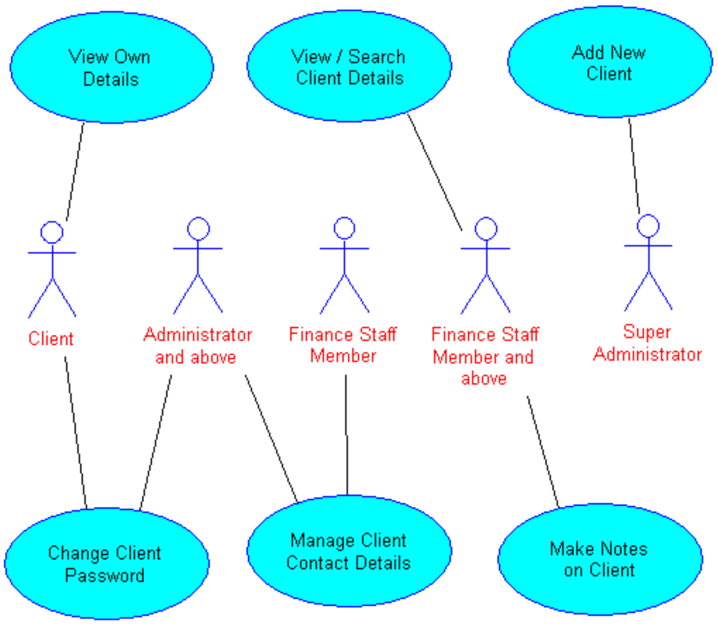


Figure C.1: Final Client Use Case

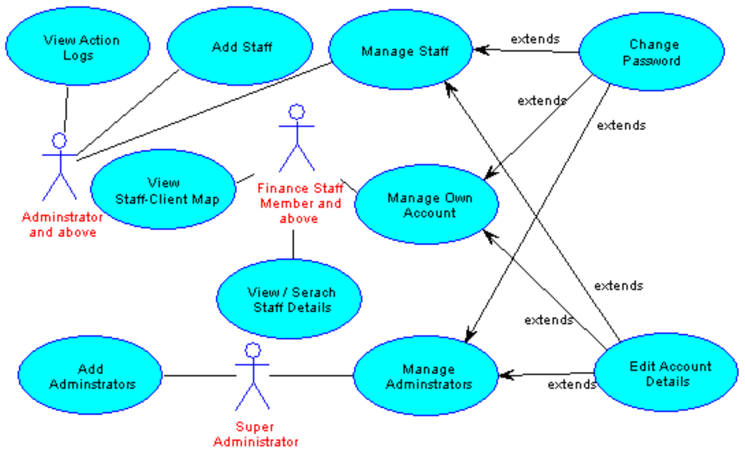


Figure C.2: Final System Staff Use Case

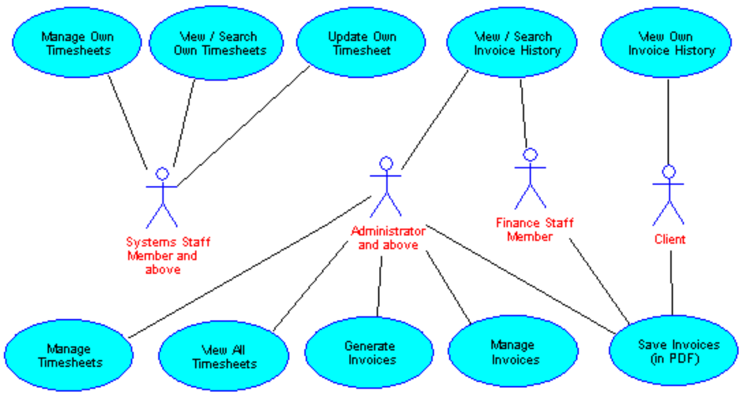


Figure C.3: Final Finance Staff Use Case

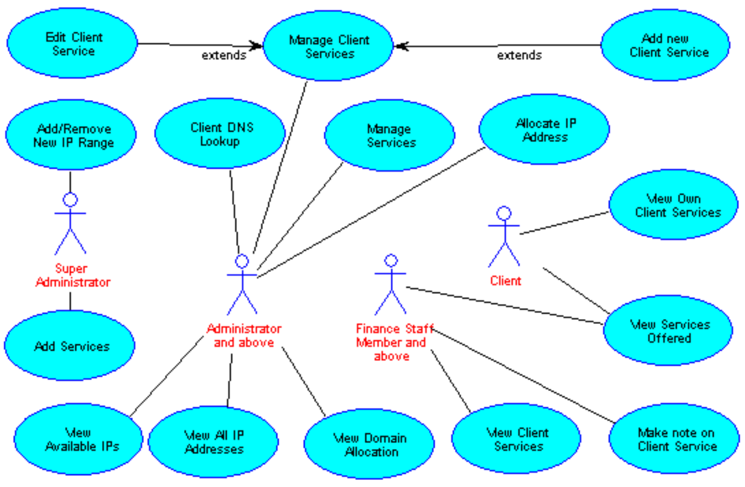


Figure C.4: Final Services Use Case

Appendix D

Class Diagram

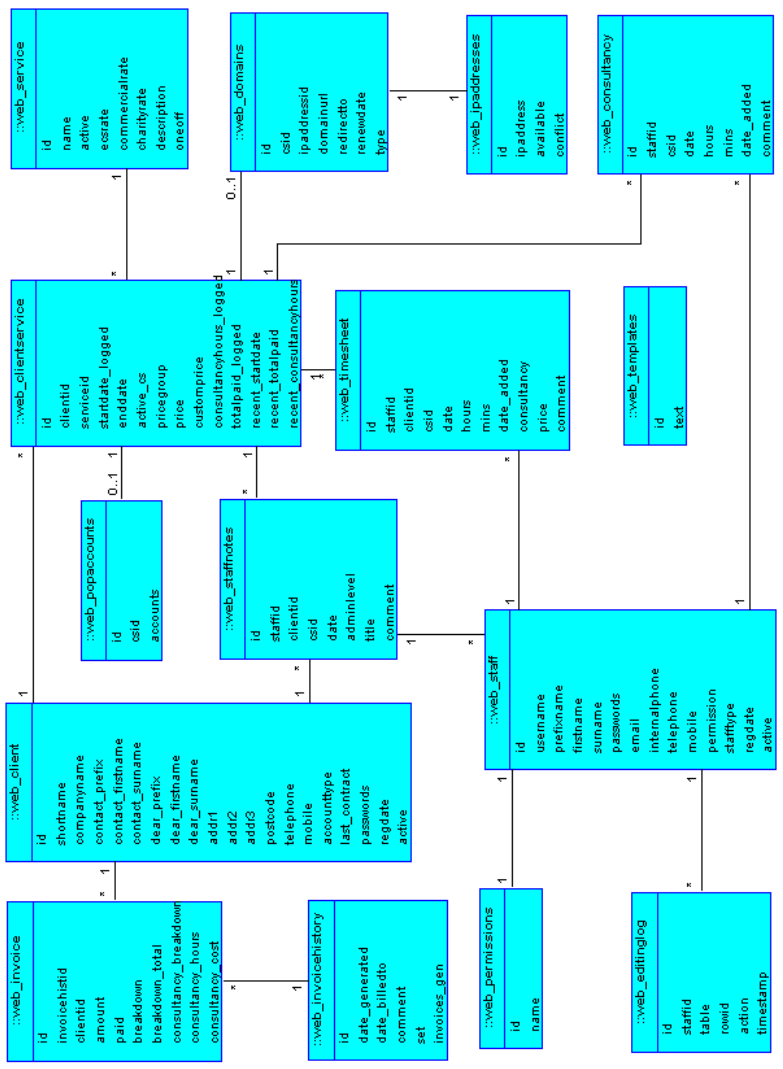


Figure D.1: Final Class Diagram

Appendix E

Gantt Chart Evolution

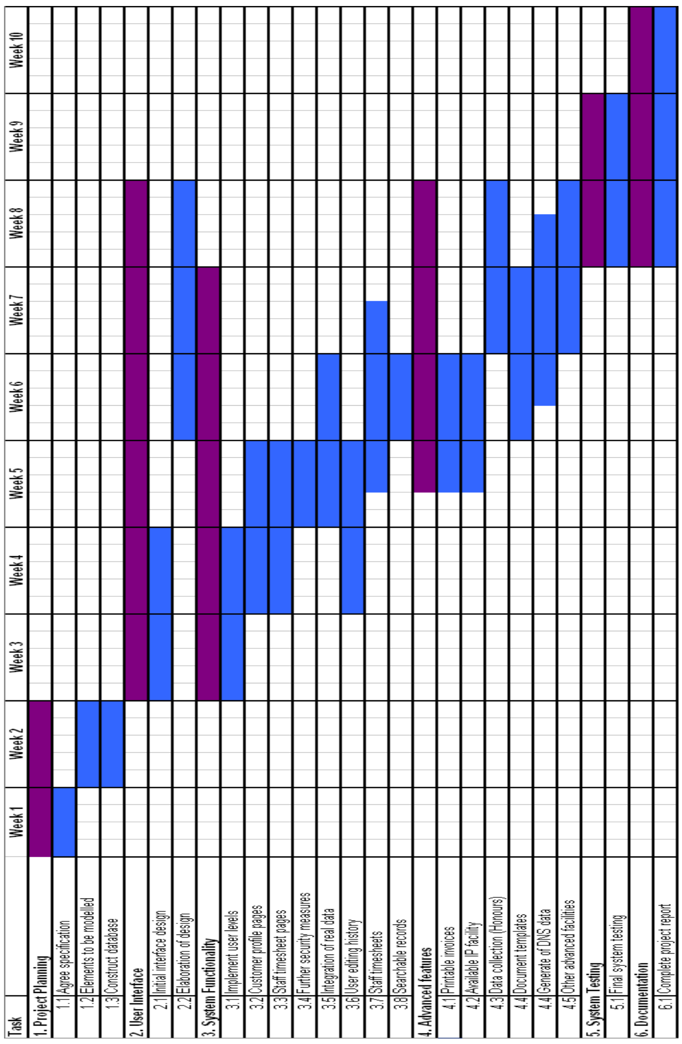
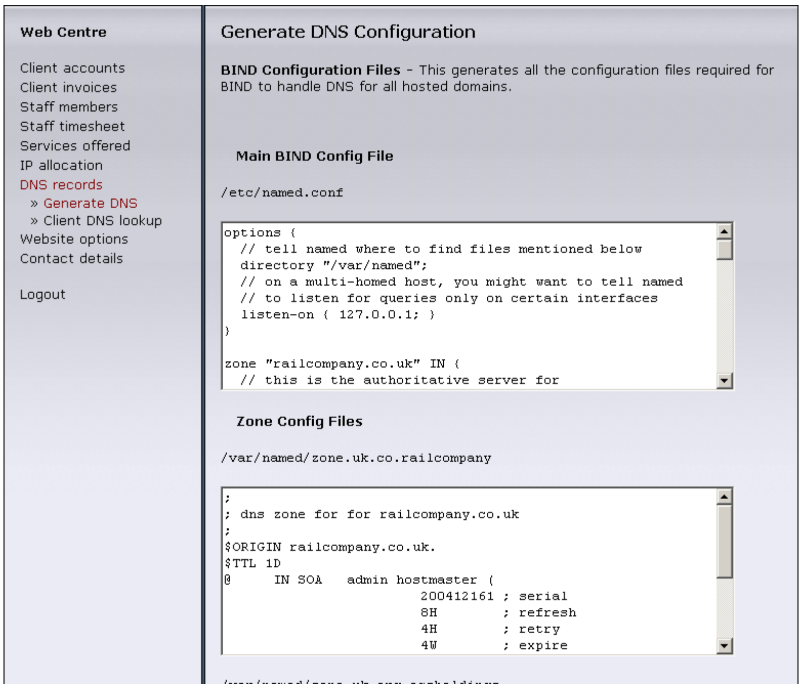


Figure E.1: Initial Gantt Chart

Appendix F

BIND DNS Progress



Generate DNS Configuration

BIND Configuration Files - This generates all the configuration files required for BIND to handle DNS for all hosted domains.

Main BIND Config File

/etc/named.conf

```
options {
  // tell named where to find files mentioned below
  directory "/var/named";
  // on a multi-homed host, you might want to tell named
  // to listen for queries only on certain interfaces
  listen-on { 127.0.0.1; }
}

zone "railcompany.co.uk" IN {
  // this is the authoritative server for
```

Zone Config Files

/var/named/zone.uk.co.railcompany

```
;
; dns zone for for railcompany.co.uk
;
$ORIGIN railcompany.co.uk.
$TTL 1D
@      IN SOA   admin hostmaster (
                        200412161 ; serial
                        8H        ; refresh
                        4H        ; retry
                        4W        ; expire
```

Figure F.1: BIND Configuration File Output

Appendix G

Page Load Times

Function	Time (seconds)
Index page	0.011121
Edit client details	0.016361
Search client database	0.021330
Display clients list	0.084690
Add new client	0.023171
Change client password	0.265766
View client's services	0.048813
Edit a client's service	0.052336
Add note to a client's service	0.015268
Add services to client	0.074340
Make client note	0.051202
View client note	0.017503
Invoice history	0.043588
View invoice history batch	0.014132
Load invoice from batch	0.022306
Generate all invoices	0.028181

Continued on next page

Function	Time (seconds)
Generate invoice for client	0.035410
Generate subset of 7 invoices	0.013764
Search invoices	0.021274
Record invoice payments	0.018750
Edit staff details	0.022282
Search staff database	0.024823
Display stall list	0.302810
Add staff member	0.014385
View staff to client map	0.043329
View staff timesheet for day	0.016158
View staff timesheet for week	0.014149
View staff timesheet for month	0.017022
Add staff timesheet for week	0.016153
Edit staff timesheet	0.014161
Full staff timesheet for month	0.303110
Search staff timesheet	0.310732
View list of services	0.167499
View service details	0.027905
Edit service details	0.014125
Add service	0.031878
View IP allocations	0.024095
View IP address details	0.018736
Complete IP allocations table	0.034163
Domain-client map	0.017728
Add IP range	0.126345
DNS records	0.109377
Generate DNS	0.275854
Client DNS lookup	0.107422

Appendix H

Customer Feedback Report (15/12/2004)

Customer Feedback	Development Team Comment
UI design simple, accessible (with exceptions), and ‘cool’ colour scheme, however the HTML ‘wobbles’ a bit with the table shifting around depending on its contents	Table widths and HTML bugs will be corrected after the content is finalised. The pages will then be tested across as many different browsers as possible.
Instigated an HTML validity check (it failed). Suggest vendor performs standards compliance checks so as to be confident of portability across different browsers.	This has now been completed.
Login isn’t secured(!). SSL transport is a MUST for the protection of login credentials	SSL will be implemented on the beta test system and the final system, but not before.
Recovery of passwords not implemented	This feature is not yet implemented.
“Page 1 of 0” when no record data available	This message will be removed.
Annotated pop-ups or ‘?’ links might aid less proficient users/administrators understand purpose of fields in data entry screens	This will be implemented in the final system.

Continued on next page

Customer Feedback	Development Team Comment
Add client: “last contract” not marked as mandatory, although entering account details omitting last contract fails (without meaningful error message)	This field is not mandatory. A bug currently prevents the user proceeding without it filled in.
Not escaping meta-chars in SQL submissions (when “add client”, or“add note” in client view) tested with company name that has an apostrophe mark in it, things go ‘bang’	Standard PHP functions will be used to escape meta-characters and sanitise user input.
No notion of session time-out (closed browser, restarted a couple of hours later, was still authenticated!)	Session expiry times were intentionally long to aid with system testing/development. They will be reduced for the final system.
No obvious route back to “view client details” when, e.g., viewing services	A link to return to view client details from view services and similar links will be implemented where it will improve the user interface.
Forcing two-digit DD/MM is tiresome	The system will be modified to accept D/M, DD/MM or DDMM depending on customer preference.
Has invoice printing/saving been implemented?	These features have been partially implemented but under development and were hidden in the customer prototype.
Dialogues such as that on ‘Generate new invoices’ would benefit from a ‘search clients’ button, or the ability to generate for a subset of clients rather than all or one	This has been implemented but obviously it is not sufficiently clear, so the user interface will be re-designed to make it clear this feature is available.
“60” minutes in ‘Hours worked’ counter-intuitive	An error check will be implemented to ensure the number of minutes specified is between 0 and 59.

Continued on next page

Customer Feedback	Development Team Comment
Full timesheet/search timesheet features not implemented	These features have been implemented but were hidden for minor re-development in the customer prototype.
How to map staff to clients? (and what does that mean?)	A message on the staff-client map will be included to explain what it is. The clarity of the documentation for this feature will be improved.
Timesheet data input fails. Data claims to go in, but nothing appears in calendar view, or in full staff timesheet view	These features have been implemented but were hidden for minor re-development in the customer prototype.
Timesheet calendar: day/week/month view all appear to be month view when there's no data there	A message will be displayed to indicate that no work has been logged in this situation.
Product prices: what does "active" mean in this context?	'Active' services are those currently offered by the WEB Centre. Services can be disabled to remove them from the price listings without deleting them.
How to find "overall" summary of client outstanding amount of invoices un-paid?	This feature has not yet been implemented.
Can we take payment for $>$ 1 invoice at once and have that reconciled in the view of the client's account status?	This functionality has been implemented. Obviously this is not sufficiently intuitive so either the user interface will be changed to improve this.
IP ranges: Precisely what is wrong with trying to instantiate a /8? but seriously - a /23 is not unreasonable for an ISP to have to manage.if the only interface is to manually manipulate each individual IP address, then that's not particularly useful	The ability to manipulate more than 256 addresses at once was disabled for this prototype. This restriction will be removed for the final product and confirmation screens will be introduced.

Continued on next page

Customer Feedback	Development Team Comment
CIDR notation (i.e. $192.168.2.1/24$ rather than $192.168.2.1/255.255.255.0$) is becoming predominant these days	The final system will allow either notation to be used.
Available domains - Free IP Addresses ? What's the purpose of the first sub-item under the IP allocation menu?	This feature lists the IP addresses that are available for hosting new websites.
"Member features" is a poor choice of section name	This will be corrected in the next revision.