**A Content Management System
to support small sports clubs**
Gary Ingham
BSc Computing (Industry)
Session 2005/2006

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of student) _____

# SUMMARY

The aim of this project was to produce an interactive web application that could be used by Leeds University Union Men's Association Football Club (LUUMAFC) to help with the management of the club and dissemination of information to its members. Usability was a central focus of the project as the application needed to be easy to set up and maintain by users with little or no technical expertise.

Background reading into the technologies available and what makes a web application easy to use early in the project identified the need to utilise an MS SQL database and online administration facility to maintain the content. Research into different development methodologies also identified the Information Systems Development Life Cycle as the most suitable development model to follow throughout the development.

User involvement throughout the development process was a priority so they were highly involved in the requirements analysis and user evaluation stages. After the implementation a detailed evaluation was conducted using user inspections and user testing techniques to evaluate the success of the system. Finally, to support the Content Management System (CMS) produced a user manual was created to assist the user.

# ACKNOWLEDGEMENTS

# CONTENTS

**CHAPTER 3 - METHODOLOGY**

**CHAPTER 4 - ANALYSIS**

**CHAPTER 5 - DESIGN**

**CHAPTER 6 - IMPLEMENTATION**

# CHAPTER 1: INTRODUCTION

## 1.1 Problem Definition

Information management and dissemination is an important issue for all businesses and organisations alike. The World Wide Web has revolutionised how information is managed and distributed by providing an online portal to share information. All large organisations are or have already taken advantage of this, but it is often the case that many small organisations lack the finances and technical expertise to explore this avenue of communication and information management.

Small sports clubs are such organisations that lack the necessary resources to establish and maintain a web presence. Leeds University Union Men's Association Football Club (LUUMAFC) is just one organisation that currently has no web presence, but would relish the opportunity to disseminate information to its 70 members and supporters on-line. Unfortunately, its members currently lack the technical skills to create their own web application.

Communication between the club and its members is currently a difficult task with each member having to be contacted individually. It is therefore believed that a web application would improve the communication process and enable the club to carry out other administrative tasks interactively online.

## 1.2 Project Aim

The aim of this project is to research the potential need for a web application amongst a wider range of small sports clubs and to produce an interactive web application for LUUMAFC to help with the day to day running of the club and dissemination of information to its members. This website must be easy to setup, customisable and easily maintained by the average computer user.

## 1.3 Project Objectives

The objectives for this project are:
1. Research the problem with the problem owner
2. Research existing internet technologies available
3. Research and choose a development methodology to follow
4. Establish a list of user requirements for LUUMAFC's
5. Research the need for a web application amongst other types of small sports clubs
6. Design a web application to incorporate the requirements of the user

7. To implement the web application design for LUUMAFC

8. Evaluate the application in terms of usability and meeting the user requirements.

## 1.4 Minimum Requirements

The following set of minimum requirements for this project are:

1. To produce a research chapter on existing solutions and technologies

2. Produce a report on the user requirements of LUUMAFC

3. Establish a generic list of requirements that satisfy all types of sports

4. Develop a web application for LUUMAFC that uses a database to manage content

5. A report of the usability testing of the application carried out by a committee member of LUUMAFC.

Further research and work to be carried out on this project, time permitting may include:

1. Incorporate the list of generic requirements into the web application produced for LUUMAFC to make it customisable and appropriate for the needs of all types of sports clubs

2. Produce a user manual to support the use of the application.

## 1.5 Deliverables

The deliverables to be produced for this project include:

1. A web application for LUUMAFC

2. Final year project report

## 1.6 Project Schedule

| Dates | Task | Milestones |
|---|---|---|
| 10-10-05  -  21-10-05 | Identify Minimum Aims and Requirements | **Submit 21-10-05** |
| 22-10-05  -  27-11-05 | Background Reading | Background Reading Summary chapter |
| 22-10-05  -  27-11-05 | Research existing solutions | Summary chapter |
| 14-11-05  -  27-11-05 | Methodologies | Methodologies chapter |
| 28-11-05  -  09-12-05 | Mid project report | **Submit 09-12-05** |
| 10-12-05  -  22-01-06 | Christmas break and exam period | |
| 23-01-06  -  05-02-06 | Requirements Analysis | Report |
| 30-01-06  -  05-03-06 | Design of solution | Design chapter |
| 20-02-06  -  09-04-06 | Implementation of solution | Implementation chapter |

| 06-03-06 - 19-03-06 | Progress meeting | **Submit 17-03-06** |
|---|---|---|
| 10-04-06 - 16-04-06 | User testing of solution | User testing chapter |
| 03-04-06 - 30-04-06 | Evaluation | Evaluation chapter |
| | Collate and write the report | **Submit 03-05-06** |

*Fig 1.0 – A table outlining the project schedule and milestones*

The above table shows the final project schedule. It splits the project up into a series of tasks and specifies time periods for each task to commence and finish. Great consideration was given to other course commitments such as coursework and examinations when deciding upon time frames for these tasks. For instance I decided that I would spend more time in semester 2 on the design and implementation of the web application due to fewer course commitments in semester 2 than semester 1. I also designed a gap in the development of this project over the Christmas period to revise for 5 examinations at the beginning of January. Work then recommenced after these examinations. A Gantt chart of this final schedule and the initial schedule are available in appendix B.

**1.7 Relevance to degree programme**

Within BSc Computing (Industry) I have studied three database modules including 'Database Principles and Practice' and most recently 'Advanced Databases' (DB11, DB21, DB31). Creating a web application with content management functionality should enable me to utilise these skills by applying a database behind my web application. Development methodologies and requirements gathering have also been studied in various Information System modules including IN11, IN21, IN23, as well as a 'Software Project Management' module (SE22). My programming skills in java and HMTL have also been aptly tested in many modules including 'Object Orientated Programming' (SE21) and 'Human Computer Interaction' (SI13), and will be further put to the test in this project. In particular my year in Industry saw me maintain and create new static web pages for a large national website. This project would not only utilise the skills learnt from this placement but expand this knowledge to include interactive features.

# CHAPTER 2: BACKGROUND READING

## 2.1 Introduction

This chapter will report on the background reading carried out to find a solution to the user's problem. It investigates the user's problem in greater depth and researches what constitutes a usable web application. It also researches the web server, database and server-side technologies that may need to be utilised to devise an appropriate solution to the problem. It discusses advantages and disadvantages of each technology to solving the problem and presents justification for the chosen technologies.

## 2.2 Research existing products

Through searching the Leeds University Union website it is apparent that there is currently no online resource available for the men's football club to utilise. Web pages do however exist for other sports clubs, but these are limited to just one introductory web page addressed solely at those students interested in joining. They do not act as a resource for the clubs to disseminate their own information or allow club members to interact with the site. This seems to highlight the possibility of a common need amongst many sports clubs in the university and perhaps even more widely spread amongst local Saturday and Sunday league teams. This is clearly an area of research that needs delving into further if I have time within this project. In the meantime more research needs to be carried out on which clubs actually have a web application and what functionality these applications have.

An interview with the tennis captain Alex Collingwood revealed because the university do not provide a website for tennis, they attempted to develop an independent website. When asked how much the tennis team used this website Alex admitted that it was never actually used, the reason for this being poor functionality. Again this highlights the possibility of a wider potential.

## 2.3 Usability

Jakob Neilsen [1] an expert in web usability, defines usability as "a quality attribute that assesses how easy user interfaces are to use. The word "usability" also refers to methods for improving ease-of-use during the design process. The importance of usability should not be underestimated in what is becoming an increasingly competitive business environment. If a user can not effectively operate your website they will simply stop using it. There are many usability frameworks and guidelines available today, but the two I will discuss are a "Web Site Usability Checklist" [2] and a general purpose checklist [3].

### 2.3.1 Framework 1

The first framework to consider is a web site usability checklist created by Larisa Thomason [2], which is based in many parts on usability guru Jakob Nielsen's work [4]. The checklist highlights 5 important components of a usable web site: design a clear and simple navigation system; keep content clear and simple; support your brand; provide for visitor feedback; and test the site on real users. This checklist is clear and simple, yet gives precise details on what a usable web site should and should not include. Overall this checklist provides a good checklist to consult at all stages in the development of my application. The full checklist can be found in appendix C.

### 2.3.2 Framework 2

Usability gurus and authors Brink, Gergle and Wood [3] highlight 10 web guidelines, which consist of general purpose principles for a quick review of a website. These guidelines can be found in appendix D. These guidelines are however, vague and in order to use them efficiently you need to be familiar with detailed guidelines first. They provide a detailed checklist of general usability issues, usually used in the evaluation of a system. This checklist checks for very specific usability aspects and requires a yes or no answer. This checklist can be found in appendix D.

### 2.3.3 Chosen usability framework

In conclusion I have decided to utilise both checklists in this project. I will use framework 1 to guide my design and implementation of the application because it is detailed and gives precise information on what a usable web site should and should not include. I will also use framework 2 as an evaluation checklist to carry out a usability inspection of my application in the testing phase.

### 2.4 User Interfaces

A user interface guides how a user interacts with a system. It should hide the complexities of the system allowing the user to operate the system easily and efficiently. An interface for a web application works in exactly the same way. Molly E Holzchlag [5] classifies 5 features of Interface Design. These are metaphor; clarity; consistency; orientation; and navigation. Metaphor refers to the symbolic representation of areas of your site e.g. using familiar images for entry points, exits and windows in an environment. Clarity demands that every asset on your website should have a reason for being there, and that this reason should be apparent to the user. E.g. images and buttons should perform their perceived purpose. The third feature to consider is consistency, which states that the use

of metaphors and navigation aids be uniformly used. Orientation means the user must know exactly where he or she is at every step of using your application. Aids to ensure this include titles, headers and footers etc. The final feature highlighted by Molly E Holzchlag [5] is the importance of good navigation, which can coincide with layout design issues.

## 2.5  Ease of Maintenance

Web sites have traditionally required the expertise of web designers for managing the content, but no more. Web content is increasingly becoming more dynamic and database driven, allowing content changes to be made by simply changing the information in text files or databases, and the changes immediately becoming evident in the web pages. A term frequently used to refer to systems that make managing content easier is a Content Management Systems.

### 2.5.1  Content Management Systems (CMS)

Russell Nakano [6] refers to a CMS as "a discipline that managers the timely, accurate, collaborative, iterative and reproducible development of a web property". CMS allow anybody within an organisation to update web pages without any prior technical expertise, which is an ideal solution to my users' problem. CMS is therefore described as a templating system, with the design layout set and the content provided by database lookups.

Updating content stored in a database therefore automatically changes the content on the web page without affecting the design layout of the web page. The simplest method to update content in the database is through the use of forms, which can be made available in the browser, thus allowing the site owner to make small or large changes quickly through an online interface. This removes the need for web page editor packages and removes the need for any hard coding by the user. The proposed web application will therefore utilise an online interface to allow the site owner to maintain the website. The term Content Management System or CMS may be used to refer to this application in the rest of the report

### 2.5.2  WYSIWYG Editors

What You See Is What You Get editors allow you to design pages without knowing html. They allow you to layout text, images and tables etc, as you would in any word processing software and the html is generated for you. The advantage of these editors is that the learning curve is low as 'what you see is what you get' making their usability consistent to all users. They can however lack control, as the software may override a user's intended design [5].

6

## 2.6 Research available technologies

## 2.6.1 Web Server technologies

There are two main types of web servers commonly used by web hosts. These are Apache web servers running on UNIX and IIS web servers running on Windows. Each web server will differ in the scripting technologies it supports so both servers must be compared to determine which platform the proposed CMS is going to run on.

The most popular web server on the internet today is Apache. It has been the most popular since April 1996 and according to a Netcraft survey [7] carried out in November 2005 "more than 70% of the websites on the internet are using Apache, making it more widely used than all other web servers combined". Apache HTTP Server is a project of Apache Software Foundation and is always being actively developed, in that those who use this software can contribute back to it in the form of enhancements and bug fixes. Furthermore, Apache is faster and more stable than many other web servers. It can also run on Windows NT/9x, Netware 5.x and above, OS/2 and most versions of UNIX.

Apache supports PHP (Pre Hypertext Processor), Perl and JavaScript languages which are examined later in the report, thus allowing access to databases such as MySQL and MS SQL Server. It does not however support ASP (Active Server pages).

Apache's popularity and widespread use alone make it an extremely attractive option, as it is important that my web application contain as fewer constraints as possible.

## 2.6.2 Client side and server side technologies

### 2.6.2.1 Pre Hypertext Processor (PHP)

PHP is a free server side scripting language that can run in both Apache and ISS web servers. The problem is not all web hosts currently provide support for it on their servers, but are doing so in increasing numbers. Free web space providers in particular are unlikely to provide this support [8].

PHP is very useful for adding interactive features to a web application and has built in facilities within the language making it extremely compatible with many types of database including MySQL, MS SQL, Dbase and Oracle. Its enormous compatibility with databases therefore makes PHP a prime candidate to be used in my web application. Although, it is not a general purpose language like Java

or Perl, PHP is said to be relatively easy to learn because it resembles the language C [9], to which I have prior knowledge which may aid development.

PHP is used within HTML and because it is processed on the server side rather than the client side, the PHP code is hidden from the client and cannot be viewed by viewing the source code of the web page.

Delving deeper into PHP we can see why some organisations prefer alternative technologies. Improperly validating variables can create security vulnerabilities which can be exploited by dishonest customers to bypass security and authorisation checks. This is often due to poor programming and would need researching further should this technology be used.

### 2.6.2.2 Active Server Pages (ASP)

ASP's are web pages that combine HTML and server side scripts. Like PHP, the server scripts are processed on the web server before being sent to the client's browser, thus only allowing the client to see the HTML code. ASP lets you use the power of the web server to process user requests and provide dynamic, individualised, content based on logic, file and database data [10]. Unlike PHP, ASP is not platform independent requiring a Microsoft Information Server or Personal Web Server to run on and therefore is not supported by Apache. It is however language independent supporting the use of any scripting language that is compatible with the Microsoft Scripting Hosts requirements, including the use of multiple languages on a single page. E.g. VB script, JavaScript and PerlScript. ASP makes creating dynamic pages easier by providing access to a set of components known as ActiveX Data Objects (ADOs), which allow easy connections to text files and databases to manipulate and display information on a web page. These components make ASP extremely compatible with any databases that run on Microsoft Operating Systems. Furthermore, ASPs are browser independent making them accessible on all browsers.

### 2.6.2.3 Perl

Perl stands for Practical Extraction and Reporting Language. It is an open source server side scripting language commonly used for creating CGI scripts to process data from HTML forms on a website. One factor that led to Perl's popularity was the availability of libraries that made it easy to write CGI programs Colburn [11]. It differs from PHP and ASP in that its code is not embedded within HTML and instead of being downloaded by the browser resides purely on the server side.

Perl was first created by Larry King for programming with the UNIX operating system but due to its power of handling large amounts of text and its use in CGI scripting it is now used for database administration. Perl, recently starting using a DBI (Database Interface) module to simplify the process of connecting and interacting with databases.

While Perl is widely used I possess no knowledge of the language, and although it resembles the C language to which I have prior knowledge, development would prove very difficult should this language be used.

### 2.6.2.4  JavaScript

JavaScript is the most popular client side scripting language and like all scripting languages can be embedded within HTML. It is an object orientated programming language that is freely available and can be used to add interactivity to web pages. It is open source and a cross platform system supported by both Windows and Linux servers, thus making it a realistic choice for my project.

Rather than the script being processed on the server, all processing is carried out in the client's browser. All major browsers like IE, Mozilla, Firefox, Netscape and Opera will sufficiently be able to process these scripts with out the need for a JavaScript application. JavaScript is also able to detect which browser the user is using and load a page designed particularly for that browser. This is beneficial as it will ensure all LUUMAFC's members and supporters view the best possible version of the website.

From a development point of view JavaScript is a language I have no experience of using. It does however encompass a similar syntax to that of Java and C, for which I do have some experience and should therefore speed up the learning process.

JavaScript's main draw back is that it requires an API (Application Programming Interface) to connect java programmes to a database, which could cause implementation problems further down the line. It is also true that JavaScript is in decline and being replaced by languages such as Coldfusion, PHP and Perl etc.

### 2.6.2.5  Evaluation of client and server side technologies

JavaScript's difficulties with database connectivity rule it out of contention leaving Perl, PHP and ASP as the remaining possibilities. While I possess no prior knowledge of either technology, I believe PHP and ASP to be used more commonly for this type of application and subsequently have chosen

not to use Perl as my scripting language. Both PHP and ASP seem indivisible in terms of database compatibility and ease of use, but providing care is taken during implementation to address PHPs security flaws, its cross platform compatibility give it a competitive advantage over ASP.

### 2.6.3  Database technologies

### 2.6.3.1  MySQL databases

MySQL is a free open source software, widely available and with good technical support. It is compatible with many Operating Systems including Windows, Linux and Mac and can be used with both Apache and Windows web servers. It can also handle large amounts of data and provides security through user authorisation and access privileges.

MySQL merges perfectly with PHP, Perl and ASP making it ideal for managing the content of a web application. These server side scripts can therefore be used to interact with a MySQL database to retrieve and add information to a database. Furthermore, MySQL is a database to which I have had some experience using in both DB21 and DB31 computing modules. This experience could prove vital during the development stage.

### 2.6.3.2  Microsoft Access databases

Microsoft Access is a very popular data management application. It is not open source but is available through the Microsoft Office software package which is now fairly standard on all PC's running a Windows Operating System. Mac and Linux operating systems will not however have access to Microsoft Access. LUUMAFC would therefore only be able to use an IIS server and windows operating system to host and edit a website. MySQL on the other hand integrates easily with Apache and other web servers using a large number of languages including PHP, ASP, Perl and Java. Access's web capabilities are also limited over the web with MySQL proving much more accessible.

Access is however advantageous to the non technical user, providing GUI's and menus so that the user can interact easily without knowing Visual Basic. It also allows the user to create a series of forms to act as an interface and employ queries to interact with the database, making it easier to use than MySQL and making it ideal for small database applications.

Access itself can also be used as an interface to access information on other databases such as MySQL. This could prove useful for a non technical user unfamiliar with SQL wanting to access information directly from a database [12].

### 2.6.3.3 PostgreSQL

PostgreSQL is an open source, platform independent database management system similar to MySQL, but less widely used. It is designed for large databases and includes more advanced features than MySQL such as the use of triggers [13]. The disadvantage of postgreSQL is that it is slower than MySQL to perform queries.

### 2.6.3.4 MS SQL Server

MS SQL server is a platform dependant database management system that runs only on ISS servers. It is designed for large databases and handles multiple users and database queries extremely well. It is also easily maintained through the Enterprise Manager GUI. The disadvantage of MS SQL is that it is not open source and expensive to purchase.

### 2.6.3.5 Evaluation of database technologies

Although Access provides useful graphical interfaces for the non-technical user, it is hoped a web application can be produced that does not require the user to modify the database directly in anyway. The design should ensure the database is modified through an online interface only.

MS SQL Sever would be a useful DBMS for this solution because it is good at handling multiple database requests by multiple uses, which could only enhance the performance of the application. The main drawback of MS SQL is its cost and because of my problem owner's small financial budget, it is an unlikely solution.

PostgreSQL and MySQL are the most viable solutions as they are both freely available and platform independent.

### 2.6.4 Leeds University IT Resources

An email enquiry to Leeds University Unions web master revealed that the central web server where LUU societies have web space supports PHP, but currently has no database connectors. It is unrealistic therefore to be able to implement this change on the union web server, resulting in the most viable option of hosting a website externally to the university. My course department however, is able to provide me with the temporary resources to implement an application on either an IIS or Apache web server, and in any language using a MySQL, MS Server or Access databases. The department are unwilling however, to continue to host my application after completion.

### 2.6.5 Chosen technologies

ASP was chosen over PHP as the scripting language for this development because it does not suffer from security weaknesses and is language independent, which enabled the use of VB script to which I am a familiar. It was hoped that this would aid the learning of ASP and assist development. The use of ASP meant an IIS sever was required for development. A MySQL database was also chosen because of its compatibility with ISS servers and because it is freely available, making it ideal for LUUMAFC who operate on a small budget. MySQL was chosen over postgreSQL because of previous experience in using it.

# CHAPTER 3: METHODOLOGY

## 3.1  Methodology comparisons

Sha Avison [14] describes a methodology as "a collection of procedures, techniques, tools and documentation aids which will help systems developers in their effort to produce a new information system".

Before the introduction of software development methodologies, systems were often poorly designed and inappropriate to the users needs as greater emphasis was always placed on programming. Without any clear structure it was also difficult to estimate project completion dates and software development projects almost always overran. Although still an issue today, it is less of a problem than before the introduction of methodologies.

Methodologies provide a more systematic approach to software development, clearly defining tasks and giving greater management control, to avoid cost and schedule overruns [14]. It is therefore essential that my project follow a methodology and this section will review a number of methodologies available.

### 3.1.1 Information System Development Life Cycle

Otherwise known as the Waterfall model this methodology divides the development into 6 discrete stages that cover every process required to develop a computer system. This enables management to take greater control over the process at each stage. It will also enable management to make more accurate predictions of project schedules, thus preventing project delays and cost overruns. The six stages of development include: feasibility study; systems analysis; system design; implementation; test; and review and maintenance. Each stage is distinctly defined with a set of sub tasks and processes ensuring that all aspects of development are covered. A stage must therefore be completed fully before moving onto the next stage. I.e. no two stages can run asynchronously.

The advantage of using this model is that the user is clearly involved in the development. The user is involved at both the Systems Investigation and Implementation stages. At the Systems Investigation stage the user is involved to establish the user requirements and needs of the system and again at the Implementation stage for user testing and entering data. As previously highlighted user involvement is critical to the success of the system.

The drawback of this development model includes changes in user requirements. The 6 distinct stages make the model inflexible to incorporate a change in user requirements throughout the development process. This may then mean that by the end of the development the system no longer has any real use to the user.

An adaptation to this model that includes the opportunity to return to the previous stage in order to review the work carried out can overcome this problem. This is known as a Waterfall model with iterative feedback. This model allows the developer to revisit the design stage or requirements stage and incorporate any user requirement changes that may occur during development. It can however make it more difficult to predict the project schedule and there will be no clear end to each stage. Therefore management decisions will be required to decide when no more iteration's are allowed.

### 3.1.2 Rapid Application Development

Rapid Application Development or RAD proposes to do exactly as the title suggests. This methodology was designed to solve long development problems and changing user requirements. It proposes to do this through an incremental approach known as timeboxing.

The system is divided up into a number of components or timeboxes that are developed separately. Timeboxes basically specify a development start time and end time. Within each of these timeboxes functionality is added to the system very quickly until the end of the timebox is reached, which cannot be extended nor additional resources allocated. The first timebox will therefore implement all the high level user requirements first to ensure the main functionality of the system is implemented. The second and third timeboxes will then proceed to implement lower and more detailed user requirements.

To achieve this quick development commercial code is utilised where applicable, thus providing well tested solutions to the system requirements. The user also has the added advantage of working with the system at the end of each timebox allowing them to specify any changes they may want.

RAD's approach enables any change in user requirements to be included in the system because the full system is not implemented straight away. A late requirement can then be included in the next timebox. Like the Waterfall model RAD's design also prevents project overruns, through the implementation of timeboxes that can not be extended. This however results in the loss of functionality, because if there is not enough time to implement all the user requirements, the least important functionality yet to be implemented is discarded. David Avison & Guy Fitzgerald [15] refer to this as the Pareto principle or 80/20 rule.

Although delivery of the main functionality is quick and often meets the bulk of the user requirements, the system produced is often missing functionality, poorly designed and flawed.

Prototyping and JAD workshops can also be used in conjunction with RAD methodology to speed up the process of requirements gathering.

### 3.1.3 Evolutionary Development

Evolutionary development methodology is another incremental approach. The delivery of the system is split up into a number of iterations carried out one after another. Each iteration contains a requirements, analysis, and design and implementation stage. Although the first iteration will only include a small section of the requirements this approach enables lessons to be learnt from each iteration and rectified in the next iteration thus, improving the system as the development continues.

Evolutionary development is especially good for projects where user requirements are difficult to find out in advance or are likely to change throughout the development process. The requirements stage in each iteration allows these requirement changes to be addressed and implemented. In this sense it is similar to Prototyping allowing the user to monitor the development and have input throughout the development, thus ensuring they are happy with the system produced.

The disadvantages of this approach include not being able to meet the full user requirements early on and the difficulty in managing and controlling each iteration and the different stages associated with each.

This methodology can also be used in conjunction with the prototyping protocol discussed later by allowing the user to be involved throughout the development [6].

### 3.1.4 Prototyping

There are two types of prototyping methodologies. These are throw away prototyping and evolutionary prototyping. Coad and Yourdan (1991) in [15] suggest "prototyping should be used for all object-orientated projects", but prototyping is now being used in more than just object orientated projects. Its use is especially growing in web development.

Prototyping involves the user more than the Waterfall model and like Evolutionary Development and RAD allows for changes in requirements, thus reducing the projects risk of failure. With prototyping

15

there is no clear stage of establishing user requirements. Instead a prototype is produced for the user to comment on, and then either enhanced to form the final system (evolutionary prototyping) or thrown away after the user requirements have been identified. It is also important to note that it may take a number of prototypes until the user and developer agree on a set of requirements, thus running the risk that an agreement may never be reached. Development time can also be wasted by implementing things that the user later decides is not appropriate.

The advantages of this methodology however are vast. Apart from the user being involved more throughout the development than in the Waterfall model it also enables them to establish exactly what is technologically feasible by experimenting with the prototype and talking with the developer. Quite often managers can have an idea in their head of a system that is often unfeasible and as a result is unhappy with the final system. Prototyping will solve this by identifying any problems in the design early in the development process, while it can still be changed. Prototyping is also especially good for systems with a small number of users and short project duration, making it a very feasible methodology to follow in my project.

Although Prototyping has many advantages, these do not come without there drawbacks. Because there are no clearly defined stages management and control of the development process is made more difficult and as stated it could take some time for the user and developer to agree on a set of requirements. At what point would you need to say enough is enough and implement the system? Prototyping also requires a good commitment from the users of the systems in order to be successful. Without it most of the advantages of Prototyping will not be exploited. Furthermore, testing of the system developed through prototyping is made increasingly difficult by the absence of a detailed specification of requirements.

### 3.1.5 Structured Systems Analysis and Design (SSADM)

SSADM is a highly structured framework that covers all aspects of the Waterfall model with the exception of implementation and maintenance. It too is split up into series of stages each with a clearly defined set of rules and guidelines including deliverables required at each stage.

SSADM focuses greatly on the analysis and design of the system placing greater emphasis on user requirements and involvement, logical system specification and technical system options. This methodology provides a very thorough examination of the design and is best suited to large scale system development projects requiring heavy documentation, therefore is unsuitable for the purposes of this project.

## 3.2  Chosen Methodology

Through analysis of the researched methodologies it has become apparent that either Prototyping or the Waterfall model with iterative feedback is best suited for the development. While prototyping will help me to produce a solution quickly and involve the user during the physical development of the CMS, the Waterfall Model clearly sets out a stage to establish user requirements, which I believe will be critical to the success of my project.

For this reason the Waterfall model with iterative feedback will be used to structure the development of the CMS. Iterative feedback will allow for stages of the development to be returned to should a change in user requirements be needed. The development will also try to exploit the advantages of prototyping by integrating it within the implementation stage of the Waterfall model to receive feedback from the user on the design and main functionality of the system.

# CHAPTER 4: ANALYSIS

## 4.1 Introduction

Analysis forms the second stage of the Waterfall development methodology and as the model suggests [14] this section must be completed before commencing the design of the content Management System (CMS). This section will examine existing user practices to gain vital understanding of the business processes before undertaking a full requirements analysis that aims to define the systems functional architecture in terms of the operations or events that must be performed in order to achieve the project aim. Also, in this section existing alternative systems will be analysed and their limitations taken into consideration during design and implementation.

## 4.2 Why do systems fail?

System development projects can fail for numerous reasons. A common cause of failure is a lack of resources allocated to researching and understanding the problem. Understanding the problem should include sufficient liaising with users and stakeholders of the system, but often IT developers neglect this vital communication, thus fail to accurately understand the business needs of the stakeholder and inaccurately define project objectives. User involvement throughout the requirements, design and implementation stage is therefore essential.

## 4.3 Existing user practices

This section analyses the existing practices utilised to carry out the main tasks of running the football club and highlights how a CMS could significantly aid and improve on these methods. The following processes were identified from the interview conducted 27 January 2006. The minutes for this meeting can be found in appendix E.

*To make club announcements;*
Club notices are generally communicated to the club members via text message, on a notice board situated in university or at team training sessions and games by team captains. Problems encountered through this approach include; informing those absent from training and matches; text messaging is a costly process; inability to check the notice board; and difficulty in informing members of any changes. An online bulletin board facility will not only be free but is more accessible for all members with internet access.

18

*Publicise fixtures and results;*

A fixture list is attached to the notice board and a copy sent to members via email, but it is up to individual members to keep their own record of results. The problems with this approach are that members often misplace the fixture list and subsequently are unaware of upcoming fixtures and who the opposition is. This leads to problems of players being unavailable for selection due to confusion over the fixtures. Confusion is increased further by fixture changes, which are not properly documented. An online fixtures and results facility will provide one accurate copy of the fixtures that can be altered when needed as well as record the results. It will remove the need for paper copies to be supplied and emails being sent.

*Publish squad selections;*

Squad selections are currently published on the notice board in university. Team captains publish the lists as early as possible on the day before a match, but find it difficult to do this at consistent times. The accessibility of the notice board to all members on the day before a match is also a problem. An online list would allow members to view squad selections from anywhere and remove the need for text messaging between team captains and their players.

*League tables;*

The league organisers publish all results and current league tables on their own website but these are often poorly maintained and inaccurate. With no other way of finding out the results of other league matches, implementing a solution to this could prove difficult. Furthermore no records of past years are kept. In theory, providing all results are known, a simple database will allow for the implementation of such a facility.

*Publicise tour information;*

Currently information regarding a tour is disseminated via word of mouth. This information usually consists of the destination, tour dates, prices and deadline for payments. A web application would support more information such as photographs and resort information. It would also enable links to tour operator's websites.

*Order training kit;*

The current mechanism for ordering kit is a paper based system. The committee distribute forms to all members and collect in the completed forms and payments before placing a bulk order with the kit manufacturer. The problems with this mechanism are; delays in collecting completed forms; misplaced forms; and no official record of all kit orders is retained. A facility to place these orders online would speed up the process and make it easier for both members and the committee.

*Promoting sponsors;*

The club receive sponsorship every year from local organisations. To promote these organisations the sponsors logos are printed on the training kit and match kit. A web application would further increase the exposure for these organisations and perhaps encourage more sponsorship for the club.

*Publish match reports;*

Match reports are a rare commodity for the club, but occasionally reports are published in the Leeds Student newspaper. The problem however is that the newspaper is unable to publish all match reports for all teams. A web application would provide an online portal for all members and supporters to follow the team's progress.

*Publish directions;*

Directions to the universities sports facilities are published on the league organisers' website. These directions however are not very detailed. A web application would allow for the club to publish its own directions for all visiting teams to view and download.

## 4.4  User Requirements

A detailed study of the needs of the user is critical to the success of any development project. Understanding the business processes is essential to acquiring these needs. Many of the existing processes have been analysed above, but it is important to ensure a development incorporates not just the current practices but any desired or future expansion of the clubs needs. To collect these requirements meetings were held with my client LUUMAFC and other prospective clients with potentially similar problems.

It is important at this stage not only to determine and understand the functions an IT system must fulfil but to establish exactly which functions will determine whether the system is successful. These functions can be known as Key Performance Indicators (KPIs) or Critical Success Factors (CSFs).

### 4.4.1  LUUMAFC Requirements

The CSFs identified for the LUUMAFC include:

1. *The ability to inform members of important notices including upcoming events.*
2. *The ability to store and maintain current and past fixtures and results for 4 teams.*
3. *The ability to store and maintain player profiles including contact details.*
4. *The ability to publish squad selections in advance of matches.*

5. *The ability to view current and past league tables.*
6. *The ability to publicise tour information.*
7. *The ability to store and maintain photographs from social events, tour and competitive matches.*
8. *The ability for members only, to discuss topics of interest.*
9. *The ability for members only, to order training kit.*
10. *The ability to link to sponsors and league organisers.*
11. *The ability to publish directions to training ground facilities.*

Other functions identified but not critical to the success of the system include:

12. *The ability to store and maintain a list of player statistics including, goals scored, assists, bookings, and man of the match awards.*
13. *The ability to publish match reports for 4 teams.*

## 4.4.2 Men's Tennis Club Requirements

In addition to LUUMAFC requirements the tennis club view *the ability to publish match reports* as a necessity. They do not however require tour information to be published and they only have two teams compared to four football teams. Additional features the club would like to see but that are not essential are the *ability to securely pay for kit orders* and *the ability to publish video clips*. Full documentation from the meeting with the Tennis club captain can be found in appendix E.

## 4.4.3 Netball Club Requirements

The Netball Club also relate to the problems encountered by LUUMAFC, but have never considered utilising a website to help overcome these problems. The prime reason identified for this is the lack of technical skills to build a website. A full list of critical requirements can be found in appendix E. In summary, Netball has 4 teams with additional members who form a recreational netball society. There requirements differ to that of the tennis club and football club, in that they; do not essentially require: *the ability to order training kit online*, instead just require a link to the kit manufacturers website; nor *the ability to store player profiles*, as contact details are sufficient. In addition to the requirements already identified, the following functions are desirable:

1. *The ability to publish club history. E.g. On team members and successes.*
2. *The ability to publish job descriptions of committee members.*

3. *The ability to store contact lists for third parties. E.g. Printing companies, match officials, tour organisers etc.*

### 4.4.4 Generic list of user requirements

To meet the requirements of all clubs the proposed web application should incorporate the CSFs of all sports teams without supplying my client with facilities that will not be utilised and become superfluous. A generic list of 15 requirements is identified as:

1. *The ability to inform members of important notices including upcoming events.*
2. *The ability to store and maintain current and past fixtures and results for a VARIETY of teams.*
3. *The ability to store and maintain player profiles including contact details.*
4. *The ability to publish a NUMBER of squad selections in advance of matches.*
5. *The ability to publish match reports. (optional)*
6. *The ability to view current and past league tables.*
7. *The ability to publicise tour information. (optional)*
8. *The ability to store and maintain photographs from social events, tour and competitive matches.*
9. *The ability for members only, to discuss topics of interest.*
10. *The ability for members only, to order training kit. (optional)*
11. *The ability to publish directions to training ground facilities.*
12. *The ability to link to a VARIETY of third parties e.g. sponsors, league organisers, kit manufacturers.*
13. *The ability to publish club history. E.g. On team members and successes.*
14. *The ability to publish job descriptions of committee members.*
15. *The ability to store contact lists for third parties. E.g. Printing companies, match officials, tour organisers etc.*

While a set of generic requirements has been established it is abundantly clear that in order to account for the differences in number of teams and the requirements of different clubs, the web application needs to allow for certain facilities to be optional. One such facility is the ability to publish match reports. These facilities could either be specified upon the initial set up, or be maintainable at all times.

### 4.5 Alternatives and competition

A number of websites currently exist for sports teams to register and create their own site. Some charge for the privilege, but some offer free hosting. These websites allow teams to create a website within their organisations domain.

#### 4.5.1 www.esportsdesk.com

Recreation Sports Management Inc. (RSM) an American based company formed in 1997, founded esportsdesk.com to allow managers and league administrators to convey important information quickly and easily. It boasts to be "the worldwide leader in the online administration of amateur sport" [16], hosting websites for many sports teams across the world, but predominantly American sports teams. Esportsdesk.com offer one free website package and a number of packages they charge for. In summary esportsdesk.com fulfil the majority of the generic requirements identified including: *support for multiple teams; notice boards; storing current and past results; player profiles; player statistics; multiple photo albums; links; local sponsorship ad server; and a method for player discussions.* However, even though the majority of generic requirements are fulfilled there are some superseding factors that make esportsdesk.com unsuitable as a solution:

1. The user is not free to add new pages to the website, hence a number of requirements can not be fulfilled: *publish squad selections; tour information; directions; order training kit.*
2. A lot of unnecessary facilities to over complicate things.
3. No control over the presentation.
4. Cluttered by unrelated sponsorship ads. Does not conform to Thomason's [2] checklist.
5. An application owned by another organisation is not personal enough and doesn't reflect the identity of the football club.
6. As the website is a mini-site of this large site, navigating to the site is more difficult, and the URL not specific to the client.

To support my argument screen shots of the web interface for this product are contained in appendix F.

#### 4.5.2 www.web-teams.co.uk

Web-teams.co.uk is similar web based company based in the UK that offers football teams a chance to create their own free website. This website is less sophisticated than esportsdesk.com and is sport specific, but still fails to fulfil many of LUUMAFC requirements. It fails to deliver: *the ability for*

*members to discuss topics; publish squad selections; tour information; store multiple photo albums; store past results; and place kit orders.* In addition, the design of the facilities offered leaves a lot to be desired. It also suffers from the same disadvantages as esportsdesk.com:

1. The user is not free to add new pages.
2. No control over presentation.
3. It does not reflect the individual identity of the sports team. Does not conform to Thomason's [2] checklist.
4. The URL is not fully representative of the particular sports team.

To support my argument screen shots of the web interface for this product are also contained in appendix F.

### 4.5.3 Conclusion

It is apparent that solutions to help propagate information for sports clubs do exist, but also that these solutions are not appropriate to all types of sports clubs. The aim of this project to produce "A web application to support small sports teams" in particular LUUMAFC and other university sports clubs is therefore justified.

# CHAPTER 5: DESIGN

## 5.1 Introduction

Design is the next step in the Waterfall model [14] and can only take place after the requirements of the application have been gathered accurately. This section will document firstly how the system will integrate the generic Critical Success Factors (CSFs) referred to in the previous section, while abiding by Larisa Thomason's [2] 'Web site usability checklist' investigated in section 2.3.1. The design will ensure the application is easy to use and maintain by the average computer user.

## 5.2 Data

The data to be stored in this application will be stored in a MySQL database because it is freely available, compatible with ISS web servers and my chosen language, ASP. To determine the structure of the database it was essential to first define the data stores necessary to fulfil the generic requirements. This required the creation of table schemas to outline the information to be stored. An Entity Relational Diagram could then be used to represent the overall structure of the database. The database also reflects the normalisation issues referred to in more detail later.

### 5.2.1 Database Schema

The functionality and content of this application will be controlled by the database so it is advantageous to ensure the tables are created correctly. The database schema will be as follows:

- Players (*player_id* int auto, title char(4), firstname char(20), surname char(20), nickname char(20), dob datetime, email char(30), mobile number(16), position char(20), comments char(200))
- Team (*team_id* int auto, name char(20))
- News (*news_id* int auto, title char(50), news char(500), date_published datetime)
- Fixtures (*fixture_id* int auto, *team_id* int(4), date datetime, location char(1), opponent char(30), competition char(10), result char(1), score vchar(10))
- Kit_selection (*kit_id* int auto, description char(50), cost char(8))
- Kit_order (*order_id* int auto, <u>kit_id</u> int(4) , quantity int(4), size char(20), <u>player_id</u> int(4))
- External_links (*link_id* int auto, name char(20), url char(80))
- Contactlist (contact_id int auto, name char(40), organisation char(40), telephone text(16), email char(30), address1 char(30), address2 char(30), address char(30), description char(40))

- Matchreports (*report_id* int auto, title char(50), summary char(1000), author char(30), team_id int(4), date datetime, document_name char(30))
- Security (*security_id* int auto, user_id char(16), password char(10), player_id int(4))

The primary keys in the schemas above are in italics and the foreign keys underlined.

**5.2.2 Entity Relational Diagram**

An entity relationship diagram helps illustrate the conceptual design of the database showing high level abstractions, relationships and cardinalities between each relation. Each entity in the database should also have a unique identifier to identify specific instances of each entity. In the example of the Players table this could be a PlayerID property that is unique to each player. This will help when using SQL to query the database. An ER diagram also helps to model the relationship between each entity. There are three types of relationship that can be modelled: one-to-one (1:1); one-to-many (1:M); or many-to-many (N:M). A 1:1 relationship is where one instance of an entity relates to exactly one instance of another entity. A 1:M relationship means an instance of one entity can relate to multiple instances of another entity. A N:M relationship means multiple instance of one entity can relate to multiple instances of another entity. An explanation of the ER diagram below explains the relationships between each entity in more detail.
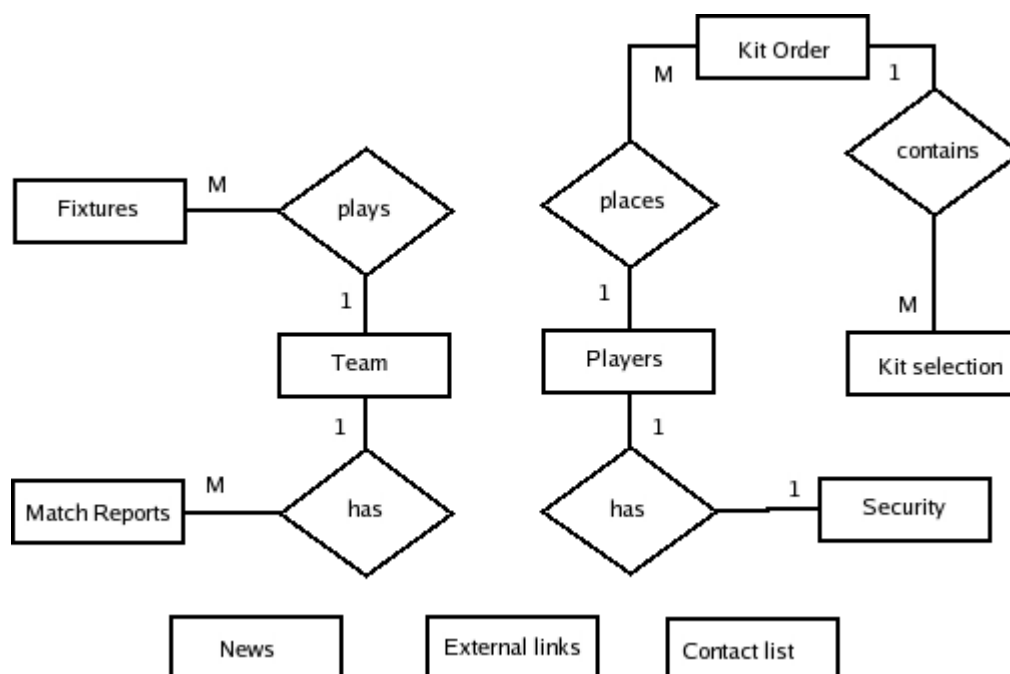


*Fig 5.1 - Entity Relationship Diagram to show proposed database structure*

The diagram in figure 5.1 shows the 10 tables represented in the database schema in section 5.2.1. Three of these tables are not related to each other and the others are related to one or more tables. The team table is related to the fixtures and match reports table by a 1:M relationship, as one team may have several fixtures and several match reports. The players tables is in a 1:M relationship with the kit order table as a player can make many orders. The players table is also in a 1:1 relationship with the security table because a player can have only one username and password, and a username and password can only belong to one player. The kit order table is in a 1:M relationship with the kit selection table because one kit order will contain many kit items.

### 5.2.3 Integrity Constraints

The tables in the proposed database structure are related together through the use of primary keys and foreign keys, emphasised in the database schema. Each table in that database includes a primary key to uniquely identify each record in the database. This is necessary as records can have the same value for fields such as 'Name'. The primary key in each table in this database design is a numeric ID field of type autonumber to ensure no record has the same identifier. In order to form a relationship between two tables the joining tables must also contain the primary key of the table it is forming a relationship with. This key is usually referred to as a foreign key. Using primary and foreign keys in this way will enforce referential integrity by ensuring no record can be added to a table unless the foreign key in that relation points to a valid record in the other relation [17].

### 5.2.4 Normalisation

Normalisation is used to define the structural properties of relations. In 3NF a new relation should be created for each functional dependency in the database to avoid duplication and redundancy of data [18]. A functional dependency "describes the relationship between attributes" [17]. That is to say that if the same field was stored in two tables, we might forget to update the data in both tables. Esen Ozkaran [18] states that normalisation should be a reversible process to maintain consistency, so that if you return to the original form of the database you can do so without losing or altering any data.

### 5.3 Maintenance

### 5.3.1 Form Design

Online forms will act as the data entry interface that will allow web site users and site administrators to add and amend data stored in the database. As discussed in depth in chapter two, forms provide an easy way for site administrators to update the database by hiding the complicated coding and

removing the need for any prior or special knowledge from the user. By providing the user with an easy mechanism to manage the database the user is able to effortlessly control the content of the web application. To avoid any unnecessary complication of these forms, they will be grey clearly structured and contain no graphics. This conforms to the usability principles highlighted in [2]. Appendix G contains a complete design of each form, stating which generic requirement it fulfils, which type of user will operate the form and which tables in the database will be affected.

The requirements not achieved through the use of tables and forms will utilise a WYSIWYG editor discussed in detail in chapter 2. This editor will be gained from an online source and will be fully referenced in the implementation chapter.

## 5.4 Presentation

### 5.4.1 Web Interface

The five features of interface design identified by Molly E Holzchlag [5] discussed in the research section have all been considered in the design of a web interface for the proposed application. The design will also conform to Thomason's usability checklist [2]. Molly E Holzchlag [20] describes 4 ways to layout a web page: using standard HTML; table-based layout; frame-based layout; and Cascading Stylesheets (CSS). HTML is most commonly used to structure text and language, while table-based layouts provide a more accessible and flexible formatting option than frame-based layouts. CSS are commonly used for design elements but can incur difficulties with browser compatibilities, therefore usually good for positioning and font styles. Another design factor requiring consideration is page dimensions. Computer screens are typically smaller than most opened books or magazines. A common mistake in web design is spreading the width of page graphics beyond the area most viewers can see on their screens. According to Web Style Guide: 2nd edition [19] "the safe area for web page graphics is determined by two factors: the minimum screen size in common use and the width of paper used to print web pages". The most common display today is an 800 x 600-pixel screen. The web interface chosen below will be built to these dimensions using tables for layout, html for structuring language and CSS to control the positioning and text style of assets on each page. A full collection of the designs considered are contained in appendix G.
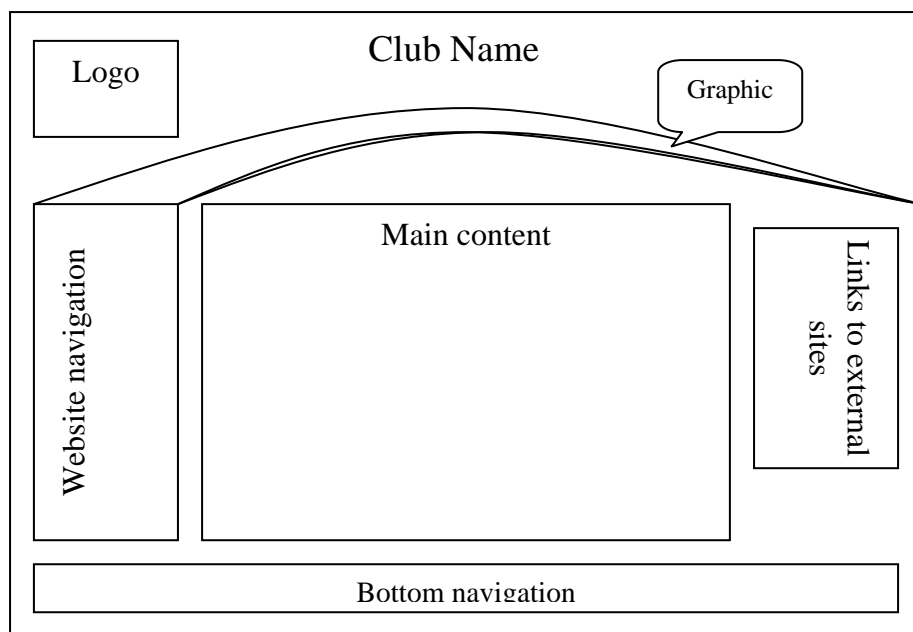
*Fig 5.15 - Web interface design*

This design in figure 5.15 is relatively simple with clear navigation on the left, right and bottom that should enhance usability and consistency and in doing so conform to Thomason's web site usability checklist [2]. The main navigation on the left is separated from external links and given more emphasis by blending into the graphic. A bottom navigation is common on many web sites and will hold links such as 'About Us' and 'Contact Us'. The limitation of this design however, is that the presence of the graphic stretching across the entire screen consequently means the dimensions of the web site will be fixed and so will appear small on larger screen displays.

The main content section of each web page will be determined by either database lookups or by the use of WYSIWYG text editors operated by the user. A detailed design of those pages required to meet the generic set of user requirements are contained in appendix G.

**5.4.2  Administration Interface**

The administration interface is only intended for a limited number of users in control of the content on the web site. The design therefore needs more emphasis on practicality than appearance. The proposed interface is a simple column structure that clearly defines the different tasks available to the user. See figure 5.16.
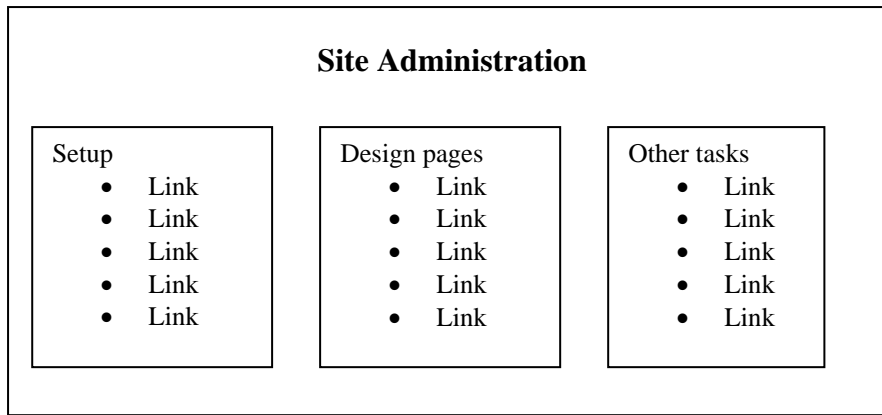
```
┌─────────────────────────────────────────────────────────────┐
│                   Site Administration                       │
│                                                             │
│  ┌──────────────┐   ┌──────────────┐   ┌──────────────┐     │
│  │ Setup        │   │ Design pages │   │ Other tasks  │     │
│  │    • Link    │   │    • Link    │   │    • Link    │     │
│  │    • Link    │   │    • Link    │   │    • Link    │     │
│  │    • Link    │   │    • Link    │   │    • Link    │     │
│  │    • Link    │   │    • Link    │   │    • Link    │     │
│  │    • Link    │   │    • Link    │   │    • Link    │     │
│  └──────────────┘   └──────────────┘   └──────────────┘     │
│                                                             │
└─────────────────────────────────────────────────────────────┘
```

*Fig 5.16 - Site administration interface design*

### 5.4.3 Navigation

Thomason's usability checklist [2] in accordance with Nielsen [1] research suggest that a user must know where they are at all times in the application and be able to answer the questions *"Where am I? Where have I been? Where can I go?* ". To achieve this Thomason [2] states that navigation must be consistent and that text links should always be appropriate and emphasised using CSS. The main navigation should also avoid dynamic menu systems created using flash and javascript etc. Nielsen [4] also states that the "content should account for at least 80 percent…navigation should be kept below 20%".

The main navigation on the left and detailed in appendix G will contain links to the pages that accomplish all of the primary generic requirements (1-10 in the list) allowing users to access all main areas of the site with no more than 2 mouse clicks. The user will have the option to include some of these links or discard them depending on their requirements. The left navigation will consist of rollover buttons marked by plain text to emphasise when a link is selected.

The bottom navigation bar below will be organised horizontally and appear underlined when the mouse hovers over the link.

The right hand side navigation will contain links to external websites. The user will be given control over these links and a database lookup will provide the display in the right navigation section. These will be text links that underline when a mouse hovers over them. To be consistent all links on the site will follow this style.

### 5.4.4 Colours

The use of colours on a website is very important. Holschlag [5] states that colours have a potent psychological impact e.g. red for love, passion, heat and flame; green for fertility, nature and earth; and blue for truth, nature and peace. It is also well documented that high contrast is very important for web usability, such that dark colours should be used against light colours and visa versa. In accordance with my chosen usability framework the colours must primarily reflect the club colours of my end user Leeds University Men's Association Football Club. These colours must also be from the web safe 216 RGB colour palette. The colours I have therefore chosen to use consistently throughout the application are dark green (hexadecimal #018424, RGB 1, 132, 36), light green (#3FBE46, RGB 1, 132, 36), white (RGB 255, 255, 255) and grey (EAEAEA, RGB 234, 234, 234).

### 5.4.5 Graphics

Graphics can help the user to understand what a link or button does, but it is important that the use or function of the graphic is abundantly obvious to the user and consistently used throughout the application. If a user needs to hover over the graphic to see what it does then it is not worth including. Author, web designer and usability analyst Vincent Flanders [21] describes navigation systems where it is inordinately difficult for users to discern the destinations of navigational hyperlinks as Mystery Meat Navigation (MMN). Any graphics I will therefore use will be chosen carefully and supported by alt text.

# CHAPTER 6:  IMPLEMENTATION

## 6.1  Introduction

The implementation phase of the project is the development of the designs produced during the design phase, documented in section 5 above. Through a series of screen shots, code snippets and descriptions this section will show how the generic requirements of the application have been met, and where necessary the differences between the proposed designs and actual implementation.

## 6.2  Data stores

### 6.2.1  Tables

Perhaps the biggest and most important change is the database employed. During the implementation phase difficulties in connecting to a MySQL database were encountered. The main cause of this problem was a missing ASP component installed on the universities IIS server. The solution to this problem was to use a MS SQL Server. There were 14 tables implemented in the final application, 4 more than anticipated in the design section. These tables include; admin_security; temp_security; temp_players; and club_details. The main rationale for implementing the temp_security and temp_players tables is to enable the user to manage who can be a member of the web site. The details sent via the registration form are therefore stored in these tables until the person is approved or disapproved. The club_details table was constructed to help make the application generic to all sports by allowing the name of the club and web site to be changed. More details on these tables are available in appendix H. Each table in the database contains a primary key field which uniquely identifies each record in the table. This field is of type 'autonumber' and increments by 1 each time a new record is added to the database. Most tables also include a foreign key allowing a relationship between two tables to be formed.

### 6.2.2  Text files

A WYSIWYG text editor is used to control the content of many of the web pages including: the homepage; squad selections; player statistics; tour; discussion forum homepage; about us; and the how to find us page. The text editor was freely obtained online from Web Wiz Guide [19]. The text editor converts the contents of the editor to html and writes this to a text file. The web page subsequently reads the contents of this text file to provide the display on a page. A detailed explanation including code snippets of this implementation is contained in appendix H.

### 6.3 Web Interface

As stated in chapters 2 and 5, good usability is critical to a successful solution to my end users problem. The design chosen in section 5.4.1 was implemented because of its clear layout and navigation structure. The final interface can be viewed in figure 6.1.



*Fig 6.2 – A screen shot of the final web interface*

### 6.3.1 Layout

The layout of the interface was constructed using a series of tables in html. A graphical representation of how these tables are organised can be seen in figure 6.3 in appendix H. Code snippets to show how these tables were implemented are also detailed here.

### 6.3.2 Navigation

The main navigation on the left is constructed using a table. Each section of the site is represented by a green rollover button that changes to dark green when the mouse hovers over it. Details on the full implementation of the left navigation, including code snippets can be viewed in appendix H. The bottom navigation bar contains text links to the pages 'About us', 'How to find us' and 'Contact us'. To emphasise these text links in accordance with Thomason's usability checklist [2] I use CSS to

underline the links when a user hovers the mouse over them. Further details on the use of CSS are reported in section 6.3.5. The right hand-side navigation contains links to external web sites and is controlled by the web site administrator from the administration interface. To ensure these links are consistent, CSS is again used. The links are dark green on a light grey background and are emphasised in red when the mouse hovers over the link. All hyperlinks that appear in the main content follow this style. As discussed in section 5.4.3 a user must be able to answer the questions: *Where am I? Where have I been? Where can I go?* To ensure this is possible, a trace of the users movements to a given page are displayed in the top left of the main content section.

### 6.3.3 Colours and graphics

The intended four colours determined in the design phase, section 5.4.4, were used in the implementation of the web interface. These colours have been carefully used to ensure good contrast throughout the web site. This is in accordance with Thomason's usability checklist [2]. The club name is written in white text on a dark green background and the main navigation is a lighter shade of green. Light grey is used as the background colour for the right hand-side navigation with dark green links. Black text is used on a white background to provide the main content while page headers can be in either shade of green. A visual representation of the use of these colours can be seen in figure 6.2 above.

The only graphics used on the web interface include the rollover buttons for the main navigation, the light green arc to separate the title from the content and the clubs logo.

### 6.3.4 CSS

A Cascading Style Sheet is used to control the font style, colour, size and positioning of text on a page. This file is called global.css and enables the appearance and layout of all pages on the web to be changed by changing just this file, thus ensuring a consistent layout throughout. The CSS syntax is made up of three parts: a selector, a property and a value. E.g. selector { property: value }. There are two different types of selector you can use to select the text you want to style, the class selector or id selector. I use the class selector. The example code snippet below shows how the CSS has been implemented to control the layout and style of the bottom navigation bar.

```
.baseline {
        font-family: Verdana, Arial, Helvetica, sans-serif;
        color: #FFFFFF;
        font-size: 12px;
        text-decoration: none;
```

```
                }

        a.baseline:hover {
                color: #FFFFFF;
                text-decoration: underline;
        }
```

The html code of a web page must also reference this class using the following code:

```
<p align="center" ><a href="aboutus.asp" class="baseline">About Us</a>
```

## 6.4  Administration Interface

### 6.4.1  Layout

The final layout differs to the proposed design. Instead of a 3 column structure the implemented interface is only a 2 column structure. The original design proposed a column each for set up tasks, page editing and administrative tasks. In reality there are few set up tasks to perform so it seemed a waste of page space to allocate an entire column to this section. The setup tasks are therefore displayed at the top of the screen and the page editing and administration tasks split into 2 columns beneath. Figure 6.6 in appendix H shows a screen shot of the final administration interface.

### 6.4.2  Navigation and graphics

Links to each facility on the administration interface consist of textual links supported by graphics with alt text. The graphics used are clear and the function they perform obvious to the administrator.

### 6.4.3  Functionality

The functions available to the user are those that help meet the generic requirements identified in section 4.5.5 and help solve the problem described in section 1.1. A detailed explanation of each of these features and how they work is contained in appendix H.

## 6.5  Security

Security is always an issue where sensitive information is stored. In this application three aspects of security were considered: security of the data stored in the database; access to the online administration facility; and access to restricted areas of the web site intended for members only. To ensure security of data in the database the database is protected by a username and password to

prevent unauthorised users from accessing the information held in the tables. To prevent unauthorised users from accessing restricted pages the user will be required to enter log in details before the page can be displayed. The same method is applied to the site administration facility. Appendix H discusses in more detail how security was implemented.

**6.6  User manual**

A user manual was written after completion of the testing phase to help support the use of the administration facility. This manual is made available online at www.leedsunisports.co.uk/usermanual.htm.

**6.7  Role out onto the WWW**

Rolling out the application from its development environment to the World Wide Web proved to be a tricky process. Searching for a cheap and affordable hosting company that met the hosting needs of the application was more difficult than anticipated. While cheap hosting packages for a MySQL database were plentiful, cheap support for MS SQL databases were few and far between. Nevertheless cheap hosting was found for £30 with Fast-trak.net. Setup with Fast-trak however, failed as support requests were ignored. I was subsequently forced to go elsewhere and received a package with unlimited database support for £66 per annum. After buying the domain name www.leedsunisports.co.uk the website was launched on the 1st May 2006.

# CHAPTER 7: TESTING

## 7.1  Introduction

The Waterfall methodology [14] discussed in section 3.1.1 states that the testing phase must commence only when the implementation phase is complete. However, the iterative approach adopted in this project allows for stages to be returned to. Due to technical difficulties in implementing two requirements at this stage: discussion forum and photo gallery, testing commenced in the absence of these two facilities. As highlighted by the methodology in chapter 3 user involvement is critical to any successful project development and this includes the testing phase. Testing is ultimately an investigation into the correctness of a system in order to evaluate it. It aims to find all the errors and examine if the project aim and system specifications have been successfully delivered. Authors and usability experts Tom Brink, Darren Gergle and Scott Wood [3] state that there are three types of usability testing: usability inspections; group walkthroughs and user testing. I will converge the usability inspection and user testing techniques to reinforce the results returned.

## 7.2  Usability inspection

A usability inspection is an evaluation based on general design principles or a specific list of guidelines and is usually carried out by the interface designer or developer. The guidelines used for the inspection are the general purpose usability checklist [3] in appendix D, discussed in more detail in (section 2.3.2). The disadvantage of this approach is that it does not involve the user so it is unclear if a problem identified by the developer constitutes a problem for the user. For this reason user testing is still critical to a successful testing conclusion.

The usability inspection returned mostly positive results. It showed that the layout and design meet all of the usability guidelines in the checklist and that the navigation in the application is good. The application also successfully met all of the HCI (Human Computer Interaction) guidelines in the checklist. The use of colour and content were also determined to be good. The inspection did, however, highlight a weakness in error tolerance. Users in the site administration need to be warned of risky or costly actions before they proceed and need a source of help, in the event of errors occurring.

The full results of the usability inspection are contained in appendix I.

## 7.3  User testing

User testing aims to identify extremely specific problems and involves observing users performing specific activities in order to identify what problems they have as they use the application. The user testing in this project will focus on three main areas: testing of the presentation interface by a potential user of the web site; testing of the administration interface by the potential site administrator of LUUMAFC; and a full test of setting up and maintaining the application carried out by a member of a separate sports club.

### 7.3.1 Presentation interface

The user was supplied with specific activities to perform and observations made on the users reactions and movement through the application. The user was also encouraged to comment on any design or functionality issues that arose.

User testing of the presentation interface showed that the usability of the web site is good with the user showing good memory and recall of the site structure by correctly navigating to each page via the shortest route possible. Testing also highlighted errors in the registration process, which are easily rectified, and expressed the need to give the squad selections page its own section and link in the main navigation.

A detailed test plan with results and comments made by the user are contained in appendix I.  Here I also propose corrective action to be taken for each problem that occurred.

### 7.3.2 Administration interface

The potential administrator for LUUMAFC, Adam McKenzie, was supplied with a similar but more extensive and lengthy set of activities to perform. These activities covered the entire range of administration functions available to the application user.

User testing of the administration interface showed similar results to that of the presentation interface. It highlighted the same functionality problems with the registration process and was also easy to use. The user found that the layout and use of graphics on the administration interface was an aid to understanding administration tasks. Overall the design of the interface was concluded as 'ok', with plenty of room for improvement, but also that the design of this interface was not a key determinant in the success of the application. The user believed the application was on track to fulfil all the agreed needs of LUUMAFC.

The full results of these tests are contained in appendix I. Again, corrective action is given to solve these problems.

### 7.3.3 Set up and maintenance

The Leeds University Tennis Club captain Alex Collingwood, was more than willing to take part in the testing process, in the pursuit of an application that would fulfil the needs of the tennis club. Alex was supplied the application template with no content in place, and asked to create a website for the tennis club unaided. This testing process aimed to find out if the application successfully delivers on the generic user requirements and on how easy the application is to set up and maintain. I observed Alex's actions and asked him to express his opinions and thoughts on the application. The main findings are summarised below and the full test results are available in the appendix I.

The testing found that the application is fairly easy to use and that it would only get easier when the user becomes more familiar with the site. The problems identified in the other testing processes also reoccurred. These include: problems with registering member security details; navigating back to the administration page from some forms did not always work; and pages did not automatically refresh after records were updated. Testing also found that certain field properties of the security table and fixtures table needed altering to allow more characters to be entered.

A major suggestion for improving the application was that the user should have the ability to change the colour scheme of the web site. It is imperative that the user is able to do this if the application is to be widely adopted by different sports clubs.

### 7.4 Conclusion

The usability inspection and user testing performed all returned similar results, suggesting the results obtained can be trusted and corrective action should be taken to remedy the problems identified. It can also be concluded from the user testing that most of the generic requirements have been met. The one requirement not fulfilled is the ability for facilities to be optional, which would give the user control over which pages are online and offline. Also, although not part of the generic requirements, the user needs to be able to change the colour scheme of the application. The fact that each user would choose to use this application suggests that the application is an adequate solution for different types of sports clubs and has gone some way to being generic. It is evident however that improvement is necessary to make it completely generic.

# CHAPTER 8: EVALUATION

## 8.1 Introduction

This chapter is an evaluation of the solution to determine if it is successful. It discusses the success of the solution in meeting the user requirements and usability targets. It also evaluates the methodology and testing methods used in the development process. Finally the chapter goes onto evaluate the solution against alternative solutions.

## 8.2 Evaluation criteria

The second usability framework [3] discussed in section 2.3.2, provides a detailed checklist to evaluate the usability of the application. This checklist was used in the testing phase to carry out a usability inspection, the results of which will be used to evaluate the usability of the application. As well as evaluating the usability of the application it is necessary to evaluate the applications success in terms of meeting the user requirements. This will be done using the test results and by a comparison of the requirements identified in section 4.4.1. A comparison of the application against the alternative solutions analysed in section 4.5 will also be made to help determine if the application is successful.

## 8.3 Meeting user requirements

LUUMAFCs requirements of the web application were agreed early in the information system development life cycle. The requirements gathering phase identified 11 functional requirements or CSFs (Critical Success Factors) that needed to be incorporated in a web application for it to be successful in solving their problem. There were also two non-essential requirements desired.

*The ability to inform members of important notices including upcoming events:*
This was achieved through the online bulletin board. The user is able to add, edit and delete articles published on this page through the online site administration interface. For details on how this was implemented refer to appendix H. User testing also showed that the users involved were very impressed with the bulletin board and believed it would be extremely useful.

*The ability to store and maintain current and past fixtures and results for 4 teams:*
This was achieved as the user is able to add, edit and delete fixtures for each team in the club. At the end if the season the user is able to store the seasons results by selecting the 'archive results' option in the online administration.

*The ability to store and maintain player profiles including contact details:*

This was achieved by requiring club members to register with the web site. Their details are stored and their profiles, minus contact details, are displayed on the player profiles page.

*The ability to publish squad selections in advance of matches:*

This was achieved using a WYSIWYG text editor to enable the user to change the layout and content of the page easily. A separate page is designed for each team in the club. Appendix H discusses in more detail how the text editor works. The results of the user testing in appendix I showed that these pages successfully met the requirement, but emphasised that the squad selection pages needed to be given more emphasis in the navigation.

*The ability to view current and past league tables:*

This requirement has only partially been achieved. It was determined too difficult to acquire the results of all fixtures in the league, so it was decided to link to the league organisers web site to view these tables.

*The ability to publicise tour information:*

This was achieved through the implementation of a tour page. The user is able to control the content on this page using the WYSIWYG text editor.

*The ability to store and maintain photographs from social events, tour and competitive matches.*

This was achieved by implementing a photo gallery that allows the administrator to create categories and display photographs in each category. The user is also allowed to interact with the gallery by porting comments to the photographs.

*The ability for members only, to discuss topics of interest:*

This has been achieved through the implementation of a discussion forum. Members of the club can register with this forum and post their thoughts and opinions. The site administrator is also given complete control over the forum. Unfortunately this facility was not implemented at the time of testing so no user evaluation of this function was carried out.

*The ability for members only, to order training kit:*

This facility has been achieved through the inclusion of an online order form. To ensure this form is available to members only, they are required to login in order to access the form. The orders made through this form are saved in the database and a display of these orders can be viewed from the site administration interface. The user is also able to control the kit selection available to the user through

this facility. While the functionality of this process is effective the design of the kit order form and the order report produced have ample room for improvement.

*The ability to link to sponsors and league organisers:*
This requirement has been achieved by allowing the user to add and delete links to other web sites in the right hand-side navigation of the web interface. Appendix H discusses how this is implemented. User testing of this function showed it to work perfectly every time.

*The ability to publish directions to training ground facilities:*
This has been achieved through the inclusion of a 'How to find us' page. This page can be accessed from the bottom navigation bar. The user is again able to control the content of this page using the text editor.

The two non-essential requirements identified include:

*The ability to store and maintain a list of player statistics including, goals scored, assists, bookings, and man of the match awards:*
This requirement was achieved by providing the user with a player statistics page that could be edited using the text editor. This implementation is not ideal as a good statistics page would utilise a database to store the data. It was however, impossible to design a statistics table in the database that could be used by all types of sports so it was decided that a table created in the text editor would be the best solution.

*The ability to publish match reports for 4 teams:*
This was achieved by allowing the user to add, edit and delete match reports. The user can upload a full report and write a summary of this report. User comments during the testing process showed that they were very happy with how this facility works.

## 8.4 Meeting minimum requirements

All five minimum requirements specified in section 1.4 of this report have been successfully met.

*To produce a research chapter on existing solutions and technologies:*
Chapter 2 of this report analyses two existing solutions available to solve the user's problem. It also researches web server, database and sever-side technologies that could have been used to create a solution to the problem.

*A report on the user requirements:*

The Waterfall methodology followed for the development of this application required the user requirements to be gathered in the analysis phase. These requirements are detailed in chapter 4 of this report.

*Establish a generic list of user requirements:*

Requirements of other sports teams that shared LUUMAFC's problem were also gathered during the analysis phase. A list of 15 requirements were generated from the requirements of each club that were deemed crucial to the success of an application that could be used by all sports clubs. These requirements are detailed in chapter 4.

*Develop a web application for LUUMAFC that uses a database to manage content:*

As covered in this chapter, the application successfully meets the user requirements of LUUMAFC, thus can be considered successful.

*A report of the usability testing of the application by a committee member of LUUMAFC:*

Thomason's [2] usability checklist was used to guide the design and implementation of the application. Testing of how successful the usability was, was conducting using 2 methods: a usability inspection; and user testing. Brink, Gergle and Woods' [3] usability evaluation checklist was used for the usability inspection and the user testing was carried out by two committee members of LUUMAFC and one from Leeds University Men's Tennis Club. A report on this process is in chapter 7.

## 8.5  Usability

The usability evaluation checklist [3] used in the usability inspection looked at nine areas of usability: architecture and navigation; layout and design; content; forms and interaction; graphics; colour; typography; error tolerance; and platform and implementation. The results of this checklist highlighted both strengths and weaknesses in usability. The application scored perfect results in the layout and design and forms and interaction sections and almost full results in the architecture and navigation section. The use of colour and content were also concluded to be good. The results of these sections alone suggest that the application is easy to use, but there were weaknesses that questioned this. The main weakness was error tolerance. It emerged from the evaluation checklist [3] that this was extremely poor. Users were not warned of risky or costly actions before they proceeded and could not reverse these actions if a mistake was made. This is a major concern and a very damaging finding to this evaluation.

User testing supported the findings of the layout, design and navigation architecture of the application, with all three users involved in the testing finding it easy to use. In light of all these results the usability of the application can be concluded as good, but with room for improvement on error handling.

## 8.6  Effectiveness of methodology

The methodology followed to develop a solution to the problem identified was the Waterfall model with iterative feedback. This was also to be combined with prototyping to gain the users thoughts on the progress of the development early in the production process. The Waterfall model with iterative feedback was chosen for its advantages in providing greater management control of the development process and making it easier to predict and manage project schedules. It also encouraged greater user involvement than other methodologies and allowed stages of the development to be returned to should there be any change in user requirements.

The Waterfall model proved to be the correct development approach. Time management was made easy by the model as development stages were broken up into clearly defined tasks making them easy to manage and predict. This is evident from the few schedule changes that were necessary. The decision to include iterative feedback proved vital in the development process. Although there was no user requirement changes during the development process iterative feedback enabled stages to be revisited should there be the need to. This was the case between the implementation and testing stages. Due to difficulties in implementing the discussion forum and photo gallery with the development resources available in the School of Computing, a management decision was necessary to undertake the testing stage without these two facilities in place to prevent the project over running. The Waterfall model states that should iterative feedback be included, management decisions would be necessary to decide when a stage is complete [14]. This is also evident here. Finally, the user's acceptance of the final application in the testing phase can suggest that the Waterfall model included an acceptable amount of user interaction during the development cycle.

## 8.7  Effectiveness of testing

The testing process involved both a user inspection and user testing of the application to evaluate if it meets the system requirements and solves the problem. It was also designed to identify any errors along the way so that corrective action could be taken to solve them.

The usability inspection utilised the usability evaluation checklist [8] to determine how usable the system was. The checklist proved a very useful tool to assess the usability of the application as it was easy to use, adequately detailed and tested all aspects of usability. The drawback of this tool however, was that it did not provide a quantitative evaluation of the application. That is to say there was no way of knowing if the results constituted a pass or a fail. This testing approach would have been more effective if the checklist was accompanied by a rating system.

The user testing was separated into 3 distinct parts: usability of the web interface; the usability and functionality of the administration interface; and how easy the application is to set up and maintain. In the first two parts the users were provided with specific tasks to perform, whereas the user involved the third test was told only to try and set up the website. The tasks outlined in appendix I for each user proved to be a valuable tool for testing as they ensured all aspects of the application were sufficiently tested. Furthermore, the tasks aided the users understanding of how the application worked and helped them to learn how to use each function. The third user test required the user to setup and add content to the application unaided and without any guidelines to follow. This test was designed to see if it could be widely adopted and setup by any sports team. The user was able to successfully setup up the application so the test can be deemed successful.

In conclusion converging both testing techniques worked extremely well as both gained useful results that were similar thus making them more reliable and trustworthy.

## 8.8  User evaluation

The user testing carried out in the testing stage of the development life cycle aimed to evaluate the user's acceptance of the application as well as its success in meeting the user requirements. Three different types of user were involved in this process to gain a full representation of how the application is perceived. The users included a member of LUUMAFC who will use the web site to gain information; a committee member of LUUMAFC who will be the future administrator of the application and in control of the content and maintenance of the application; and a committee member from Leeds University Men's Tennis Club. All users were encouraged to express their opinions on all aspects of the application throughout the testing and asked to comment on the functionality and usability as a whole at the end. Involving 3 different users during the testing phase was beneficial as it provided a more conclusive set of results.

The first user was in positive acceptance of the application. He expressed that the overall design was good and it was very easy to use. The only improvement suggested was to give the squad selections page more emphasis in the main navigation because it was a main feature.

The user who tested the functionality of the administration facility was highly positive about the functions available and how easy they were to use. On the design of the administration interface, he believed it left a lot to be desired but expressed that this didn't have a damaging influence over how he perceived the application as a whole. Overall, the user was happy to acknowledge that the application was usable.

The third user supported the comments of both users with similar comments, thus the application can be deemed to have successfully gained the acceptance of all the users.

**8.9  Comparison of alternative solutions**

Two online solutions were discussed and analysed in section 4.5 of this report. These solutions include esportsdesk.com and web-teams.co.uk. Esportsdesk.com is an American based website that caters for all types of sports, whereas web-teams.co.uk provides for football teams only. To evaluate the success of this application we can compare the functional and usability differences of these alternative solutions with the application, and against the user requirements in section 4.4.1.

Functionally the application is superior to both alternatives, meeting all the requirements of the user. Those requirements not fulfilled by both alternatives include: *publishing squad selections; publishing tour information; ordering training kit; publishing directions to training ground facilities; and the ability to link to third party websites*. Esportsdesk also fails to offer members the opportunity to discuss topics of interest.

The usability of esportsdesk.com was found to be poor. The user was unable to control the colour or layout of pages, pages were cluttered with unrelated sponsorship ads and navigation was extremely poor. In comparison web-teams.co.uk was very easy to use, but its design was very bland. In their favour though it does allow the user to easily personalise the colour of their mini-site to reflect its brand, something not achieved in this web application.

Overall the functional advantages and proven usability of this CMS in the testing stage suggest it is a much more attractive and capable solution to the users problem than these alternatives.

**8.10  Time management and schedule**

The Waterfall methodology adopted in this project helped to manage the schedule of the project by splitting the development process up into 6 discrete stages. A schedule was then devised to manage

each stage of the development. This initial schedule proved to be too ambitious in the first 10 weeks of the project as at the mid project report stage requirements gathering had not begun as planned due to an over run in the background reading. It was therefore necessary to make changes to the initial schedule for the mid project report. The initial schedule can be seen in appendix B.

The final schedule allocates an extension of 2 weeks for the background reading and delays the requirements gathering stage by 4 weeks. To allow for these changes to take effect without causing the development to over run, requirements gathering was allocated only 2 weeks instead of the initial 3, and the design stage was put back one week. The final change was to extend the design and implementation stages by 1 week.

Upon reflection the schedule was not strictly followed as little reference was made to the schedule throughout the development process and tasks did not always begin and end when the schedule suggested. The Waterfall methodology however, did ensure that the tasks were carried out in the order planned and that each task was completed before the next task begun, with exception to the instances where tasks were aloud to overlap.

## 8.11  Possible improvements

The evaluation of the application against the user requirements showed all of them were met, but it doesn't mean to say that improvements are not possible. The requirements analysis also identified 15 requirements common to different sports teams suffering from the same problem as LUUMAFC. It was beyond the scope of this project to produce an application for all types of sports, but nevertheless the final application achieved all of the requirements to some degree and was accepted by Leeds University Men's Tennis Club. User testing identified a number of new features that could be implemented to improve the application and make it more appealing to all sports teams. These include: enabling the user to control the colour of the web interface in order to reflect their identity; allowing them to choose if a page is online or offline; and allowing them to create new pages through the administration facility.

It is not just new features that can make improvements but also changes to existing features and functions implemented. The player statistics page could be improved by utilising the database to store the data, rather than a text file. This feature was implemented in this way due to time constraints and difficulties in designing a data store that could be used by all types of sports. The implementations of the discussion forum and photo gallery were also quick solutions due to a lack of time and do not reflect the design of the rest of the site. It would be an improvement if theses features were developed as part of the web application rather than external features linked through the web interface.

An improvement to the functionality would include allowing the user to delete multiple records at once as currently the user must select individual records and delete them separately. The authorised user could also be allowed to upload their photos directly to the photo gallery, or have them approved by the site administrator.

The design could be improved by including small amounts of flash graphics, such as the latest headline scrolling across the top of the page rather than displaying it static in the latest headline box underneath the main navigation.

The user testing also highlighted small changes such as giving the squad selections page its own section in the navigation which was carried out after the testing.

# CHAPTER 9: CONCLUSION

## 9.1  Evaluation of entire project

The aim of this project was to research the potential needs of a web application amongst small sports teams and to develop a Content Management System (CMS) for LUUMAFC to help disseminate information to its members and improve other business processes. The development focused primarily on the usability of such an application, under strict usability guidelines [9], and the functionality needs of the user. It was recognised early in the project that a usable solution must utilise a database to drive the content of the application and that this must be updated through an online administration facility.

The CMS produced successfully met the usability and functional requirements of the user and gained their acceptance. The roll out of this application onto the World Wide Web shows the project was successful. The user testing and evaluation of the application did however highlight room for improvements and expansion. Although all the user requirements were successfully met the application is not yet completely generic to all sports as the user is still restricted to what functions they can perform. The application could therefore be developed further to include additional functionality and allow the user greater control.

# References

1. Nielsen, Jakob, (2003), *Usability 101: An Introduction to Usability,* http://www.useit.com/alertbox/20030825.html [3rd February 2006]

2. Thomason, Larisa, (2004), *Web Site Usability Checklist,* http://www.netmechanic.com/news/vol7/design_no4.htm [18th April 2006]

3. Brink, Tom, Gergle, Darren, Wood, Scott, (2002), *Usability Evaluation,* in: Usability for the web: designing websites that work**,** pp. 405-441

4. Nielsen J, (1999), *Site Design, in:* Designing Web Usability, 1st ed, pp. 188-193

5. Holzschlag, Molly E, (1998), *The Art if Interface, in:* Web by Design: the complete guide!, pp. 77-83, Sybex

6. Nakano, Russell, (2002), Web Content Management: a collaborative approach,  Addison-Wesley

7. Netcraft (2005), November Web Server Survey 2005- http://news.netcraft.com/archives/2005/11/07/november_2005_web_server_survey.html [5th December 2005]

8. Maxfield, Wade, (2001), MySQL and PHP from Scratch

9. Heng, SL, Christopher (2000) – www.javascriptkit.com/howto/phpcgi.shtml [2nd December 2005]

10. Jones, Russel, A, (2000), *Behind the Scenes – How Active Server Pages Work,* in: Mastering Active Server Pages 3, pp. 4-32, Sybex

11. Colburn, Rafe, (2003), Sams Teach Yourself CGI in 24 Hours

12. Dubois, Paul, (2003), *Migrating from Microsoft Access to MySQL, version 1.01,* http://www.kitebird.com/articles/access-migrate.html [6th December 2005]

13. Jepson, B, PostgreSQL vs. MySQL: building better databases, www.webtechniques.com/archives/2001/09/jepson/ [6th Decemeber 2005]

14. Avison, David, Shah, Hanifa, (1997), *Information System Development life cycle, in:* The Information Systems Development Life Cycle: A first course in Information Systems, pp. 67-87, McGraw-Hill

15. Avison, David, Fitzgerald, Guy, (2002), *Methodologies, in:* Information Systems Development, Methodology, Techniques and Tools, 3rd ed, pp. 347-433, McGraw-Hill

16. www.esportsdesk.com [08 March 2006]

17. Connolly, T, Begg, C. 2005, Database Systems. A Practical Approach to Design, Implementation, and Management**,** 4td ed. Addison-Wesley

18. Ozkarahan, Esen, Database Management: Concepts, Designs and Practice, pp 114-116

19. Lynch, Horton, (2004), *Web Style Guide,* 2nd ed,
    http://www.webstyleguide.com/page/dimensions.html [14th April 2006]

20. Holzschlag, Molly E, (1998), *Layout Technology*, *in:* Web by Design: the complete
    guide!, pp. 375-395, Sybex

21. Flanders, Vincent, *Websites that suck,*
    www.webpagesthatsuck.com/mysterymeatnavigation.html [14th April 2006]

22. Web Wiz Guide, *Web Wiz Rich text Editor*, http://www.webwizguide.info/asp/default.asp
    [12th March 2006]

23. The ASP Resource Index, www.aspin.com [30th April 2006]

24. Web Wiz Guide, *Forums,* www.webwizguide.info/web_wiz_forums/default.asp [29th
    April 2006]

# Appendix A – User Reflections

The problem to be solved in this project was ideal for me. I am a keen sportsman and enjoy web development so from the very start of the project I was excited to begin the development of a solution. Looking back on the entire project I am reasonably happy with the quality of this report, but feel the Content Management System (CMS) produced had not fulfilled my own expectations. I believe if more time was available I could significantly improve both the functionality and design of the application, and plan do so upon completion of this project.

I feel my personal time management throughout this project was good as I put in more than the recommended 400 hours work and allocated sufficient time to each section of this report. Having said this I was poor at following a schedule and did not always perform sections in a logical order. This was especially evident between the design and implementation sections as some aspects of the functionality had not been fully designed before an attempt was made to implement them. Because of this, some aspects of the functionality took longer to implement than probably would have, had the design been in place. This was a waste of valuable time.

I have taken a tremendous personal gain from this project by learning new skills that will help me in industry. I have learnt an entire new language, ASP, from scratch and expanded my web development knowledge by learning how to add interactive features to a website. I also gained valuable knowledge of using a MS SQL server to drive the content of a web application and how to roll out a web application onto the World Wide Web.

If I was to attempt this project again I would try to follow a schedule more closely and complete all the background research before the half way mark to allow more time for development and the report write up. I would also ensure a detailed design of the proposed application was complete before commencing development. My research showed that the main reason projects failed was because programmers were too quick to commence programming yet I did the same thing in this project. I was just lucky it didn't have a negative impact.

# Appendix B – Final Project Schedule

| Task | Month | | | | | | | | | | | | | | | | | | | | | | | | |
|------|-------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| | Oct-05 | | | | Nov-05 | | | | Dec-05 | | | Jan-06 | | Feb-06 | | | | Mar-05 | | | | Apr-05 | | | | May-05 |
| **Week ending** | 09-Oct | 16-Oct | 23-Oct | 30-Oct | 06-Nov | 13-Nov | 20-Nov | 27-Nov | 04-Dec | 11-Dec | | 29-Jan | 05-Feb | 12-Feb | 19-Feb | 26-Feb | 05-Mar | 12-Mar | 19-Mar | 26-Mar | 02-Apr | 09-Apr | 16-Apr | 23-Apr | 30-Apr | 07-May |

Gantt chart tasks (CHRISTMAS EXAM PERIOD BREAK spanning Dec-05 / Jan-06):

- Identify Minimum Aim and Requirements — 09-Oct to 16-Oct
- Background Reading — 23-Oct to 27-Nov
- Research existing solutions — 23-Oct to 27-Nov
- Requirements Analysis — 29-Jan to 05-Feb
- Methodologies — 20-Nov
- Mid project report — 04-Dec to 11-Dec
- Design of solution — 05-Feb to 12-Mar
- Implementation of solution — 26-Feb to 09-Apr
- Progress Meeting — 12-Mar to 19-Mar
- User testing of solution — 16-Apr
- Evaluation — 02-Apr to 30-Apr
- Collate and write the report — 23-Oct to 11-Dec and 29-Jan to 07-May

# Initial Project Schedule

| Task | Month | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|-------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Week ending** | | **Oct-05** | | | | **Nov-05** | | | | **Dec-05** | | **Jan-06** | | **Feb-06** | | | | **Mar-05** | | | | **Apr-05** | | | | | **May-05** |
| | | 09-Oct | 16-Oct | 23-Oct | 30-Oct | 06-Nov | 13-Nov | 20-Nov | 27-Nov | 04-Dec | 11-Dec | 29-Jan | 05-Feb | 12-Feb | 19-Feb | 26-Feb | 05-Mar | 12-Mar | 19-Mar | 26-Mar | 02-Apr | 09-Apr | 16-Apr | 23-Apr | 30-Apr | 07-May |
| Identify Minimum Aim and Requirements | | ▬ | | | | | | | | | | | | | | | | | | | | | | | | |
| Background Reading | | | | ▬ | ▬ | | | | | | | | | | | | | | | | | | | | | |
| Research existing solutions | | | | ▬ | ▬ | ▬ | ▬ | | | | | | | | | | | | | | | | | | | |
| Requirements Analysis | | | | | | | | ▬ | ▬ | | | | | | | | | | | | | | | | | |
| Methodologies | | | | | | | | ▬ | | | | | | | | | | | | | | | | | | |
| Mid project report | | | | | | | | | | ▬ | | | | | | | | | | | | | | | | |
| Design of solution | | | | | | | | | | | | ▬ | ▬ | | | | | | | | | | | | | |
| Implementation of solution | | | | | | | | | | | | | | | | ▬ | ▬ | ▬ | ▬ | ▬ | | | | | | |
| Progress Meeting | | | | | | | | | | | | | | | | | | | ▬ | | | | | | | |
| User testing of solution | | | | | | | | | | | | | | | | | | | | | | | ▬ | | | |
| Evaluation | | | | | | | | | | | | | | | | | | | | | | ▬ | ▬ | ▬ | | |
| Collate and write the report | | | ▬ | ▬ | ▬ | ▬ | ▬ | ▬ | ▬ | ▬ | ▬ | ▬ | ▬ | ▬ | ▬ | ▬ | ▬ | ▬ | ▬ | ▬ | ▬ | ▬ | ▬ | ▬ | ▬ | ▬ |

CHRISTMAS EXAM PERIOD BREAK

54

# Appendix C – Larisa Thomason's Website Usability Checklist

1. **Design a clear and simple navigation system.** According to Web usability expert, Jakob Nielsen, a good navigation system should answer three questions:

   1. Where am I?
   2. Where have I been?
   3. Where can I go?

   Your site's navigation system will answer all three questions if you're careful to include these basic elements:

   - **Keep it consistent.** The navigation system should be in the same place on every page and have the same format. Visitors will get confused and frustrated if links appear and disappear unpredictably. Consider using Server Side Includes for your main navigation system to make certain the navigation system stays consistent.

   - **Use appropriate text inside links.** Don't make your visitors guess where a link is going to take them. Visitors should be able to anticipate a link's destination by reading the text in the link or on the navigation button. This isn't the time to be cute or obscure - visitors don't have the time or patience for it.

     If there's any question about a link's destination, clarify the issue with a TITLE attribute that explains exactly where the link goes.

   - **Use CSS to emphasize text links.** Some designers dislike underlined text links inside page content - although visitors expect to be able to click on underlined text. If you decide to remove this important visual navigation clue, style your links with CSS to replace underlining with another, consistent visual technique like a background color, different font, or text color that indicates a hyperlink.

   - **Always include text links.** You can create some great looking menus using JavaScript or other scripting language, but never rely completely on a dynamic menu system. Some users may have problems using a mouse to navigate through the menu and others may be listening to the page using a screen reader. Every page should have basic text links that link to all major sections of the site.

   - **Add a text-based site map.** Large or complex sites should always have a text-based site map in addition to text links. Every page should contain a text link to the site

map. Lost visitors will use it to find their way, while search engines spiders will have reliable access to all your pages.

- **Include a home page link inside your main navigation system.** Visitors may enter your site via an internal page, but hopefully they'll want to head for the home page next.

- **Site logo links to home page.** Most sites include their logo somewhere at the top of every page - generally in the top, left-hand corner. Visitors expect this logo to be a link to your site's home page. They'll often go there before looking for the home link in the navigation system.

- **Include a site search box.** A robust site search feature helps visitors quickly locate the information they want. Make the search box prominent and be sure that it searches all of your site - and only your site. We've run across far too many Web sites that include a "Search the Web" search box on their home page. The result? Visitors hardly get to the site before the search function sends them to another site!

2. **Keep the content clear and simple.** You may attract visitors with an eye-catching design, but content is what keeps them at the site and encourages them to return. Content is also the best way to boost your site in search engine rankings.

Always keep search engines in mind when you write content, but remember that your ultimate audience is human visitors. Present your content with humans in mind.

- **Don't save the best for last.** Place your most important content high on the page. Think of a newspaper: the top story is always prominently displayed above the fold. Check your page display at in a number of different screen resolutions to make sure that your most important content is visible when the page loads.

- **Make page content easy to scan.** You'll spend hours - maybe days - writing your page content and it's really annoying to think that visitors may read less than half of it. Format your content so that it's easy to scan. Emphasize important points (or product characteristics) with a combination of header tags, bold type, color, or lists.

- **Avoid using text inside images whenever possible.** Text in images is invisible to search engine spiders and to visitors who may have images turned off in their browsers or who use assistive technologies like screen readers.

- **Add ALT and TITLE attributes to all images.** Each image should have a descriptive ALT attribute and TITLE attribute associated with it - particularly images that are also links to other pages. That way, they can quickly jump to the page they're interested in without having to wait for the entire page to load.

- **Contrast, contrast, contrast!** Be careful with background images and colors because they can obscure the text content on the page. Make sure you have a good reason to deviate from the successful dark text on a light background model. Visitors can't buy your products if they can't read the content.

3. **Support your brand.** A good brand creates or reinforces a user's impression of the site. When your site is strongly branded, that means that visitors will think of you first when they go shopping for your product or service.

   Branding on a Web site takes time, effort, and close attention to page design and layout.

   - **Keep colors and typefaces consistent.** Visitors should never click on an internal link in your site and wonder if they've left your Web site. Choose your colors and fonts carefully and use them consistently throughout the site.

   - **Keep page layout consistent.** Use a Web site template to enforce a uniform page structure. Visitors should be able to predict the location of important page elements after visiting just one page in your site.

   - **Custom error page.** Create a useful custom error page that helps visitors if they should click on a broken internal link or type a URL incorrectly. The custom error page should reflect the site's overall color, type, and layout structure as much as possible and provide useful links to help visitors find what they're looking for.

   - **Create a good tagline and use it on every page.** A good tagline clearly and concisely explains your "value proposition" or what makes your site stand out from competing sites. It should be memorable and reinforce your brand in one quick phrase.

4. **Provide for visitor feedback.** Forms are critical to the success of ecommerce sites. Without forms, you can't have a shopping cart. But any site usually needs at least one form to allow for user feedback. A form helps you hide from email spiders and also helps you control how user feedback is formatted and sent.

57

- **Keep feedback forms short and clearly note which information is required to successfully submit the form.** Take care to design accessible forms that all visitors can use.

- **Remember your international users and don't require information they may not have - like area codes or ZIP codes.**

- **Present complete contact information including your business phone number and postal address.** A street address is preferred, but you may want to use a PO box if yours is a home-based business. Visitors will probably prefer to contact you using email or a form, but they feel more comfortable with a site that allows other contact methods.

5. **Test the site on real users.** Remember that you're the designer so of course you effortlessly use the navigation system, love the content, and understand the value proposition. But now it's time to get user feedback - before your online users start sending it in.

Usability testing **helps you replicate the experience of the average Web site user and correct problems before online visitors find them.** It also gives you valuable answers to other questions:

- **Do visitors enjoy using the site?** If so, they'll stay longer and read more content.

- **Do they understand the purpose of the site?** If not, there's no compelling reason to return.

- **Is there any incentive to return after the first visit?** Your site should try to be the ultimate authority on the Web for your topic. A site with depth and breadth encourages visitors to bookmark it and refer friends interested in the same topic.

- **Can they recover from errors?** Usability testing is the best way to test how well your site search, site map, forms, and custom error pages function. They should all work together to guide a visitor through the site and help him get where he's going. Frustrated visitors aren't likely to return - ever.

# Appendix D

10 Web Guidelines from "Usability for the web: designing websites that work"

1. Content and scope
2. Speed
3. Navigation
4. Appropriateness of task
5. Visual design
6. Compatibility
7. Simplicity
8. Consistency and contrast
9. Error handling
10. Respect for the user

A detailed general-purpose checklist can be seen in appendix I.

# Appendix E – Requirements Gathering

## MEETING WITH LEEDS UNIVERSITY MENS ASSOCIATION FOOTBALL CLUB

| | | | |
|---|---|---|---|
| Date | 27 January 2006 | Start time | 3pm |
| Location | University Union (Arc conference room) | End time | 3:45pm |
| Chair | Gary Ingham | Secretary | Gary Ingham |
| Present | Shaun Cooke<br>Chris Nunn<br>Nick O'Donoghue | Copies to | |
| Apologies | Thomas Gregson<br>Martin Cocoron | | |

1. Welcome and apologies

   1.1 I welcomed all members to the meeting and thanked them for their support of this project.

   1.2 LUUMAFC committee members Thomas Gregson and Martin Corcoron sent their apologies.

2. General

   2.1 LUUMAFC are an independent club affiliated with the University and consists of 4 teams. It has its own brand logo and its club colours are green and white.

   2.2 The club do not currently possess a web site, but it has been on the agenda for 2 years. The main reason the web site has so far not been implemented is due to a lack of skills.

   2.3 The main reason for a website is to communicate with the clubs 70 members.

3. Web site requirements

   3.1 The committee need to notify members of important announcements i.e. social events, training schedules, general notices. These notices need to be viewable by all members and supporters. **ACTION: Main requirement**

   3.2 Currently, squad selections are published on notice boards in university prior to matches. These must be available online. **ACTION: Main requirement**

   3.3 Fixtures and results for all teams must be available. It would be good to store past results as well. **ACTION: Present fixtures compulsory. Past results strongly advised.**

   3.4 League tables currently stored on league organiser's site. They should be available on this application. I explained that it would be difficult to maintain bearing in mind the difficulty to find out all the match results from other universities. The best option would be to link to the organisers site. **ACTION: Separate page but link to external web site**

3.5 The committee want to publish photographs of matches, social events and tour. **ACTION: Main requirement**

3.6 It would be useful if club members place a training kit order with the kit secretary online. An email order to the secretary would be sufficient. This is to be available to club members only. This suggests web site registration necessary. **ACTION: Main requirement**

3.7 Possibly the ability to publish match reports. Committee members disagreed on the importance of this facility.

3.8 Provide a discussion forum. This is to be available to members only. **ACTION: Main requirement**

3.9 It would be good to have a profiles page, of all the club members. The details to be shown include: Name; Nickname, Dob, Position, Additional comments. These would be added once at the beginning of the season. **ACTION: Main requirement**

3.10 The option of including a player statistics section. Providing it is not to difficult and time consuming to maintain. **ACTION: Not compulsory**

3.11 It would be compulsory to provide links to BUSA and NUL (league organisers). **ACTION: Main requirement**

3.12 The web site should publish directions to the clubs facilities, to be viewed by other universities. **ACTION: Main requirement**

3.13 Providing a page to supply information about tour is needed. **ACTION: Main requirement**

4. Any other business

   4.1. The committee members were happy to be involved in the testing stage of the application and expressed their willingness to help at any stage.

   4.2. I promised to show the committee the application regularly throughout the development and inform them of any problems I may have in meeting their requirements.

5. Date and chair of next meeting

   5.1. No official meeting is planned at this time.

**MEETING WITH LEEDS UNIVERSITY MENS TENNIS CLUB**

| | | | |
|---|---|---|---|
| Date | 24 February 2006 | Start time | 4pm |
| Location | University Union (Arc conference room) | End time | 4:45pm |
| Chair | Gary Ingham | Secretary | Gary Ingham |
| Present | Clare Cochrane<br>Linzi McMaster | Copies to | |
| Apologies | | | |

3. Welcome and apologies

    1.3  I welcomed all members to the meeting and thanked them for their time and support of this project.

4. General

    3.14 The club have never considered a web site. The main reason being that they didn't know how to build one.

    3.15 The main reason to use a website would be to communicate with club members.

    3.16 The club has 4 teams.

4    Web site requirements

    4.1  A website would need to publish latest notices, such as social events and squad lists. I suggested a separate section for squad lists, which they agreed would be ideal.

    4.2  They would like the application to allow for past members to stay in touch. Possibly a discussion forum open to present and previous members.

    4.3  An application would need to store fixtures for 4 teams in 2 different leagues: Busa and NUL.

    4.4  League tables must also be stored. Clare pointed out that the tables they refer to are on the BUSA website and are kept up to date.

    4.5  When prompted, Clare expressed that the club have never written match reports but that if it is easy to use, they may start to do so.

    4.6  Club members order there training kit individually so the application would require only a link to the kit manufacturers' web site.

    4.7  It would be extremely useful to use the web site to generate a contact list i.e. mobile number, email and student number. This would allow the committee to keep in contact with all members. These details must not be available for all to see.

4.8 As well as a contact list for members it would be good to store contact lists for third parties such as suppliers and t-shirt printers. This would help pass on information to new committee members at the end of each season.

4.9 Publishing job descriptions of each committee member would also be beneficial.

4.10 It would be good to have a photo gallery with different categories. All members should be able to upload their photos. Under advice, Clare and Linzi both agreed it would be a good idea to ensure that a committee member approved these photos before they are published on the live site.

4.11 It would be good to have a page about existing members. This would include a paragraph about them.

4.12 When asked about storing player statistics, the Netball club said they wouldn't use such a facility.

4. Any other business

4.1. The netball team would be interested in considering the possibility of using my application

5. Date and chair of next meeting

5.1. No official meeting is planned at this time.

| Date | 22 February 2006 | Start time | 4pm |
|---|---|---|---|
| Location | University Union (Arc conference room) | End time | 4:30pm |
| Chair | Gary Ingham | Secretary | Gary Ingham |
| Present | Alex Collingwood | Copies to | |
| Apologies | | | |

5. Welcome and apologies

1.4 I welcomed Alex to the meeting and thanked him for his time to support this project.

6. General

4.13 The Tennis club would be interested in a web site. A member has previously tried to implement a website but it never took off because due to poor design.

4.14 They have a large member's base of 50 -70 people, but only 2 teams consisting of 15 -20 members.

4.15 The main reason for a web site would be to improve communications with its members. They are constantly gaining new members throughout the season making it difficult to keep in contact with them all. It would be good to notify them of training times.

4.16 The web site must be simple to use and update.

4.17 Alex said it didn't matter if the web site is part of the union or separate, but expressed that it would be good if it was recognised as part of the union.

5 Web site requirements

5.1 The club want to be able to notify members of training times and cancellations.

5.2 The application needs to store fixtures and results, including previous years.

5.3 It needs to store profiles of the committee members and 1st and 2nd team members (15-20).

5.4 It would be good to be able to publish match reports, viewable by all web site visitors.

5.5 A photo gallery is compulsory, to publish photographs of matches and social events. A member should be able to upload there own photographs. I suggested members being able to upload photographs, but that the administrator must approve which photos go on the live site. Alex agreed this would be the best option.

5.6 The application should store the league tables as well as tournament tables (held 3 or 4 times a year).

5.7 A discussion forum would be useful and should be available to all club members.

5.8 It would be advantageous for members to order training kit online. Being able to pay online would also be good. I told Alex this may be out of the project scope.

5.9 When prompted, Alex expressed the need for links to league and tournament organisers. He also stated that these may change throughout the year so it must be possible to change these.

5.10 When asked for any extra extensions, Alex suggested a video gallery to play clips of matches.


4. Any other business

4.1. Alex expressed his desire to use my application at the end of development should it meet the requirements.

4.2. Alex was also happy to assist in the testing phase.

5. Date and chair of next meeting

5.1. No official meeting is planned at this time.

**Netball Requirements**

Those functions or operations required by the Netball Club include:

1. *The ability to inform members of important notices including upcoming events.*
2. *The ability to store and maintain current and past fixtures and results for 4 teams.*
3. *The ability to store current and passed league tables.*
4. *The ability to publish squad selections in advance of matches.*
5. *The ability to link to the kit supplier.*
6. *The ability to store contact lists for all members.*
7. *The ability for all members to upload and store photographs.*
8. *The ability to store club history. E.g. On team members and successes.*
9. *The ability to publish tour information.*
10. *The ability to publish job descriptions of committee members.*
11. *A discussion forum.*
12. *The ability to store contact lists for third parties. E.g. Printing companies, match officials, tour organisers etc.*

The functions desired by the Netball team but not essential include:

13. *The ability to publish match reports.*
14. *A subsection for recreational netball to publish general notices.*
15. *To store the contact details of recreational members.*

# Appendix F – Alternatives and competition

**Esportsdesk**
www.esportsdesk.com

This is the interface for esportsdesk. It shows the difficulty in accessing individual club web sites.

Difficult to navigate to a sports clubs mini-site

Promotion links for the company
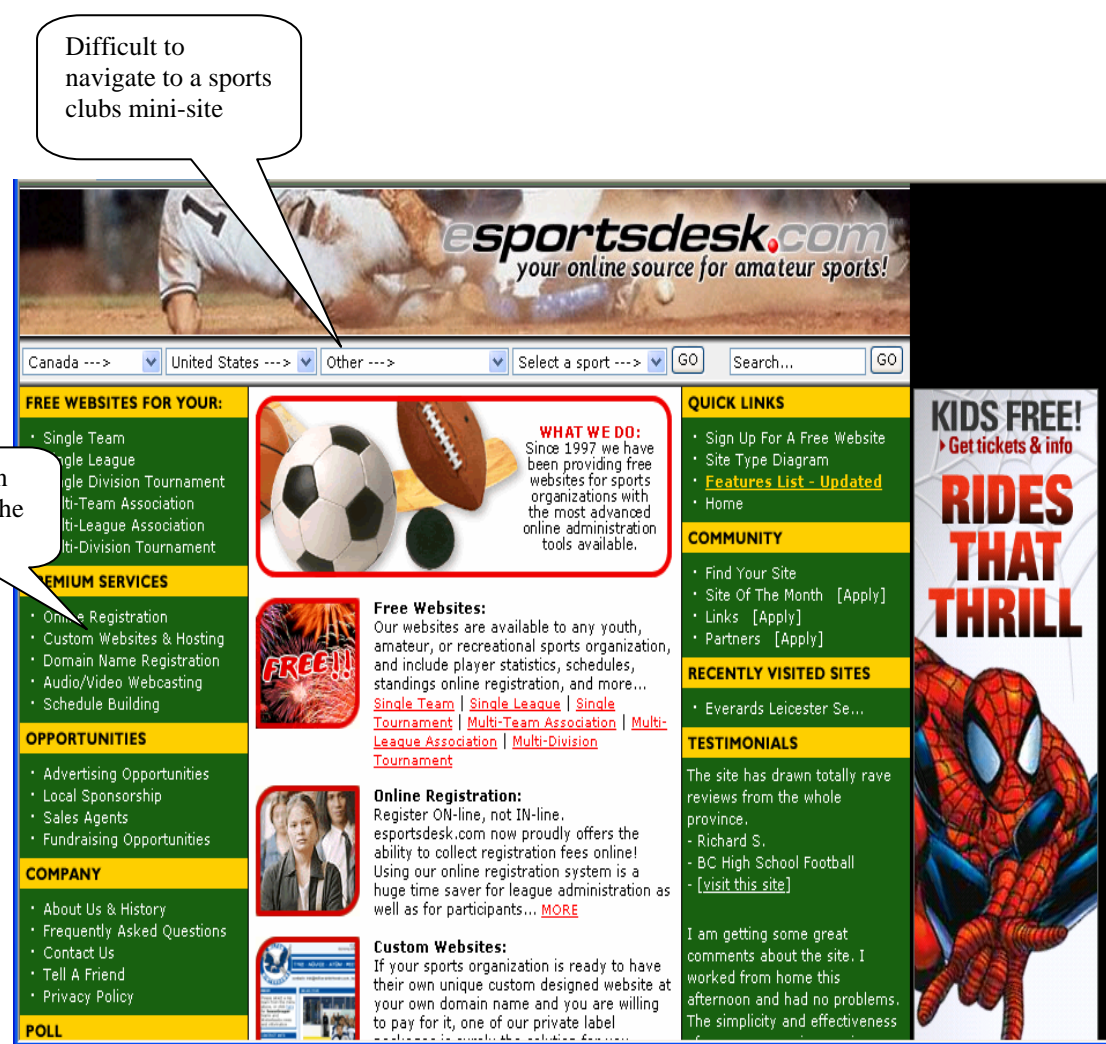


Below is a screen shot of the homepage for a sports club registered with esportsdesk. The page is bland and cluttered with sponsorship unrelated to the club. The page lacks identity with the club name the only asset on the page to show that this is the homepage of the 'Greater Manchester In-line Hockey league'.

URL does not represent clubs identity

Main navigation small

Cluttered with sponsorship

**Web-teams.co.uk**
www.web-teams.co.uk

Below is a screen shot of one football clubs homepage. The usability of this page is poor with the main navigation too small and the page content too long. This is common on all pages.

Main navigation small

Page is too long

# Appendix G – Design

## Form design

The following forms are necessary to achieve the generic requirements listed in chapter 4. A description of the purpose of each form including the tables that will be affected by its use and the requirement each helps realize, accompanies each diagram.

**Registration form**
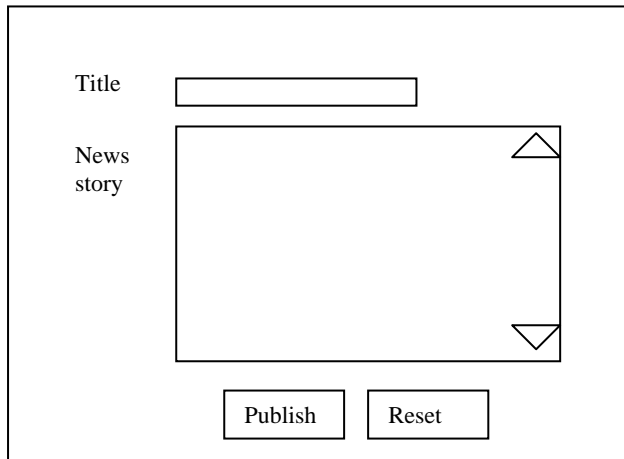


*Fig5.2 – Registration form*

This form will be used to allow players to register as members of the website. The form will populate the Players table with the members' personal details and the Security table with the members' username and password. This form will be used by all users. Together this form and the Players table will aid the successful delivery of generic requirement number 3: *"The ability to store and maintain player profiles including contact details".*

**News form**
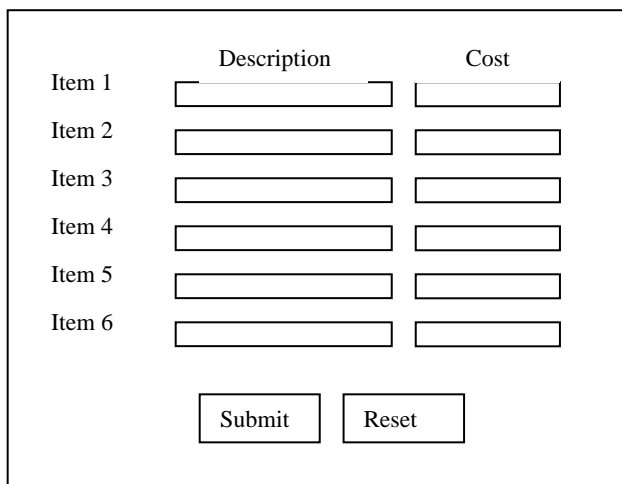


*Fig 5.3 – Add new form*

This form populates the News table and is for use by site administrators only. It will assist in achieving generic requirement number 1, *"The ability to inform members of important notices including upcoming events".*

**Kit selection**



*Fig 5.4 – Kit selection form*

This form will populate the Kit_selection table and will be used by site administrators to determine the choice of kit available to all members.

**Kit ordering**



*Fig 5.5 – Kit order form*

This facility is for members only so will require members to login in order to access this form. The form will take the player_id of the member and store the form details in the Kit_order table. The member must first confirm these order details. This form will assist in achieving requirement number 10, *"The ability for members only, to order training kit. (optional)"*

**Match reports**



*Fig 5.6 – Add match reports form*

This form is to be used by the site administrator. Clicking the 'Add' command button will add the form data including the name of the uploaded document to the Matchreports table. This facility will support generic requirement number 5: *"The ability to publish match reports. (optional)"*

**Fixtures**



*Fig 5.7 – Add fixtures form*

The site administrator will be asked for the number of fixtures he/she wishes to enter in the database, and that number of rows will be displayed on the next form. The data entered in this form will be stored in the Fixtures table. A similar form will also be required to amend the details of any fixture. For this form the user will be asked to select the fixture they wish to change. This supports the delivery of generic requirement number 2, *"The ability to store and maintain current and past fixtures and results for a VARIETY of teams."*

**External Links**
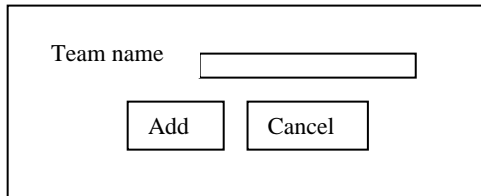


*Fig 5.8 – Add external links form*

The site administrator will use this form to add links to external web sites. The data will be stored in the external_links table. This will achieve requirement 12, *"The ability to link to a VARIETY of third parties e.g. sponsors, league organisers, kit manufacturers."*

**Teams**

Requirements analysis showed that sports clubs have varying numbers of teams so it is important to store each team in the database.
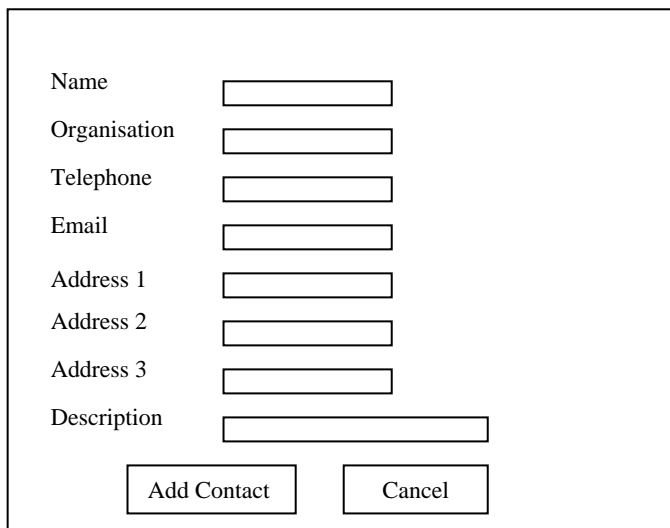


*Fig 5.9 – Forms to add team names*

The only detail required in the Teams table is the team name, as the team ID is automatically generated. It may also be necessary to provide the user with a mechanism for editing this name or deleting teams from the Team table.


**Contact details**



*Fig 5.10 – Contacts form*

This form will be used by the site administrator to store contact details of third parties and organisations e.g. sponsors, suppliers and match officials etc. The details stored in the contact_list table are likely to be viewable only by the site administrator. A similarly structured form is all so necessary to allow the site administrator to amend these details. This form helps achieve requirement number 15, *"The ability to store contact lists for third parties. E.g. Printing companies, match officials, tour organisers etc."*

# Interface design

**Design One**
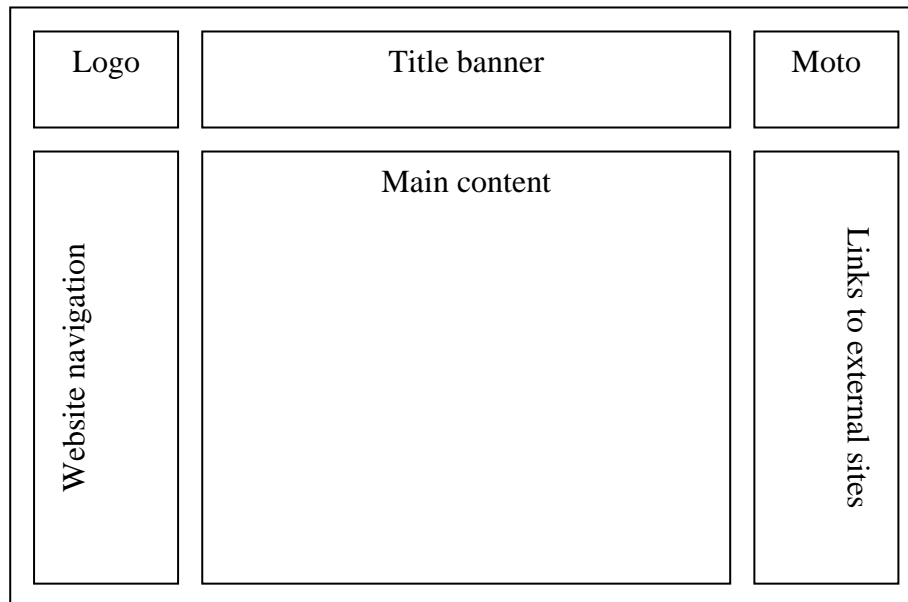


*Fig 5.11 – Interface design one*

The logo is located in the top left hand corner adjacent to the title, and the main navigation of the web site located on the left of the screen, as advised by my chosen usability framework.

Advantages of this interface include:

- Quick and simple to implement using a table-based layout
- The layout will allow for the dimensions of the interface to adapt to the screen size of the user thus the web site will appear consistent to all users
- Clear navigation.

Disadvantages:

- Plain and boring
- Unlikely to impress the user.

**Design Two**
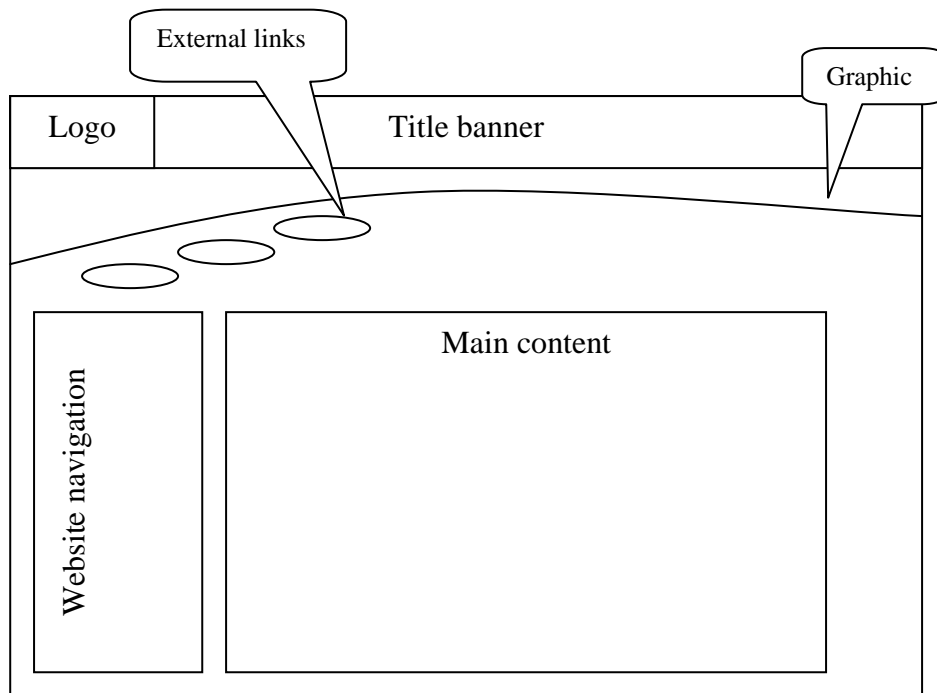
*Fig 5.12 – Interface design two*

Again, the logo is situated in the top left hand corner adjacent to the title located centre. The title banner is then separated by an arced graphic to add more shape to the design. Immediately beneath this graphic or buttons that link to external web sites e.g. sponsors.

Advantages of this interface include:

- Clear navigation
- Aesthetically pleasing.

Disadvantages:

- External links given more emphasis than main navigation
- Graphic means the layout dimensions must be a fixed size.

**Design Three**
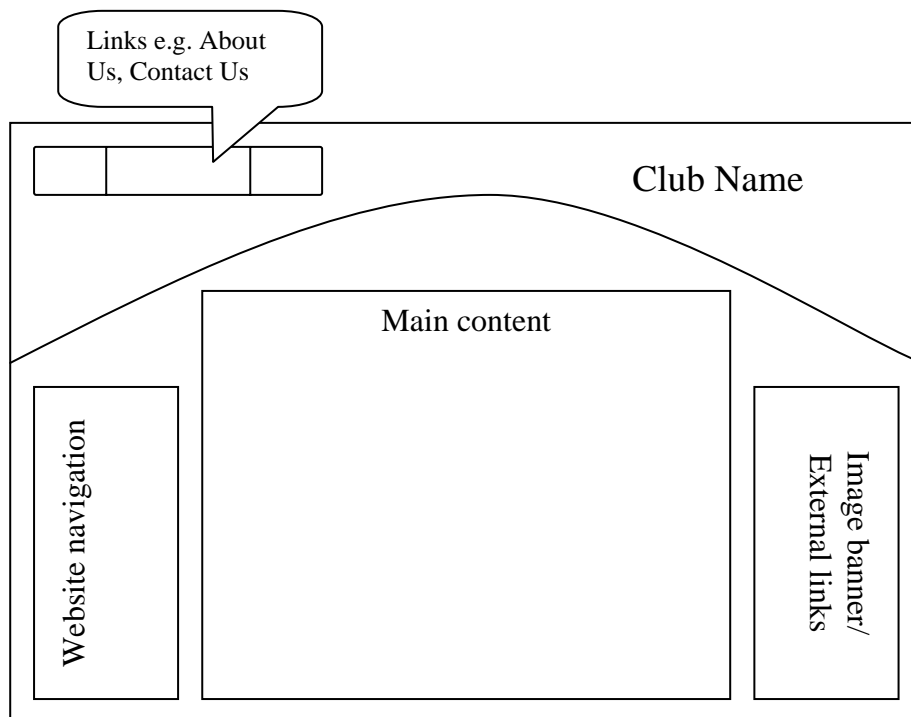


*Fig 5.13 – Design three*

Similar to design two, but with the external links located to the right hand side of the main content. Again the main navigation is located to the left.

**Design Four**
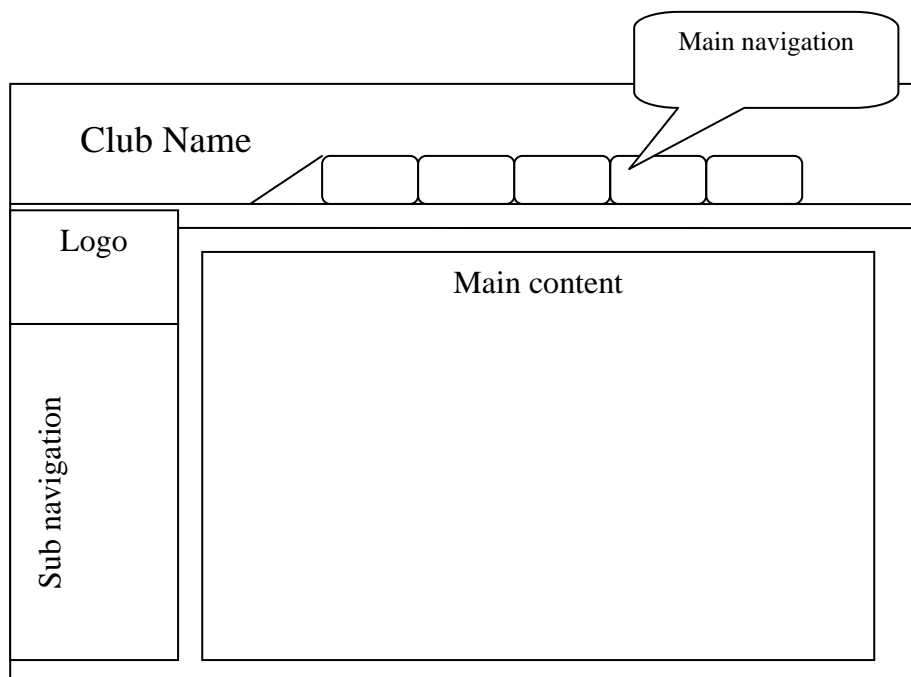


*Fig 5.14 – Design four*

This design is different to all the designs considered so far with the main navigation along the top of the screen and additional links on the left. The logo is located underneath the club name. This design would also adapt to the screen size of the user.

**Design Five**
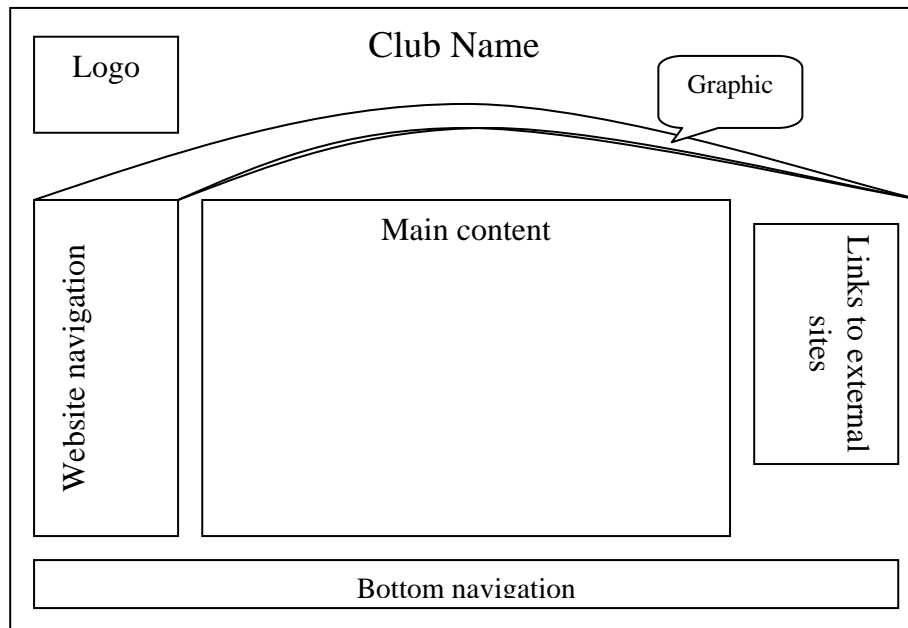


*Fig 5.15 – Design five*

The graphic on this design means that the dimensions are fixed and will appear the same size regardless of screen size. This design boasts a bottom navigation common on many sites that will hold links to pages such as 'About Us' and 'Contact Us'. The main navigation is separated from the external links and given more emphasis by blending into the graphic.

# Page Designs

The following page designs are for those pages that utilise the database to store the content of a page. These pages fulfil the following generic requirements listed in chapter 4:

*1. The ability to inform members of important notices including upcoming events.*

*2. The ability to store and maintain current and past fixtures and results for a VARIETY of teams.*

*3. The ability to store and maintain player profiles including contact details.*

*5. The ability to publish match reports. (optional)*

*8. The ability to store and maintain photographs from social events, tour and competitive matches.*

**News**

| **Latest News** |
| --- |
| Title |
| Content |
| *Date pub* |
| Title |
| Content |
| *Date pub* |

The content of the latest news page will be provided by a database lookup from the News table. The news articles will be displayed with the most recently published at the top of the page. For usability purposes the page will display no more than 10 articles on a single page.

*Fig 5.16 – Design of the bulletin board*

**Fixtures**

| **Fixtures** | | | | | |
| --- | --- | --- | --- | --- | --- |
| Date | Location | Vs | Competition | Result | Score |
| | | | | | |

The fixtures table will be laid out in a table, with each row of the table provided by a database lookup. The rows read from the database will be ordered by date in ascending order.

*Fig 5.17 – Design of fixtures page*

**Match reports**

| | | |
|---|---|---|
| **Match Reports** | | |
| Date | Vs Opponent | Score |
| **Title** **Author** **Summary** | | |
| | | Link full document |
| Date | Vs Opponent | Score |
| **Title** **Author** **Summary** | | |
| | | Link full document |

The content on this page will be provided by a database lookup from the Matchreports table. Those records entered into the table using the Match reports form will be displayed in this format.

*Fig 5.18 – Design of match reports page*

**Player profiles**

| | |
|---|---|
| **Player Profiles** | |
| A B C D E F …. X Y Z | |
| Pic | Title Name Nickname Position Comments |
| Pic | Title Name Nickname Position Comments |
| *Page [1] [2] [3]* | |

The content on this page will be provided by a database lookup to the Players table. The records will be ordered alphabetically by name and store no more than 20 profiles per page. The web user will be provided with the facility to search through the profiles alphabetically or by visiting each page in turn.

*Fig 5.19 – Design of profiles page*

**Gallery**

| | | | | |
|---|---|---|---|---|
| **Gallery** | | | | |
| Category | | | | |
| Recently added… | | | | |
| Pic | Pic | Pic | Pic | Pic |
| | | | | *View full* |
| Category | | | | |
| Recently added… | | | | |
| Pic | Pic | Pic | Pic | Pic |
| | | | | *View full* |

The photo gallery page will consist of those photos most recently added in each category with a link to a full archive of picture thumbnails.

*Fig 5.20 – Design of photo gallery*

**Log in**

| Members Log In |
| --- |
| Please enter your username and password. |
| Username     ▭ |
| Password     ▭ |
| Log In |

A member must enter is or her details in order to access the kit ordering form and discussion forum. A database look up will verify the details entered by the member.

*Fig 5.21 – Login form*

Additional pages are required to meet the remaining generic list of requirements, but I will give the site administrator complete control over the layout of the content on these pages. To achieve this I will provide the user with a WYSIWYG editor. Below I have listed the page needed and the requirement it fulfils.

| Page | Requirement met |
| --- | --- |
| Squad selections | 4. *The ability to publish a NUMBER of squad selections in advance of matches.* |
| Tables | 6. *The ability to view current and past league tables.* |
| Tour | 7. *The ability to publicise tour information. (optional)* |
| How to find us | 11. *The ability to publish directions to training ground facilities.* |
| About us | 13. *The ability to publish club history. E.g. On team members and successes.* |
| | 14. *The ability to publish job descriptions of committee members.* |

*Fig 5.22 – Table to show additional pages that will be implemented*

# Navigation

**Left Navigation**

> Home

> News

> Fixtures & Results

    > 1$^{st}$ team

    > 2$^{nd}$ team

    > + additional teams (optional)

>Match

    > Squad selections

    > Match reports (optional)

    > Player statistics (optional)

> League Tables

> Player Profiles

> Gallery

    > Matches

    > Social

    > Tour (optional)

> Discussion Forum

> Tour (optional)

> Kit Order (optional)

**Bottom Navigation**

About Us | How to find us | Leeds University | Contact Us | Site Admin

*Fig 5.23 – Design of bottom navigation bar*

# Appendix H – Implementation

**Database Implementation**

Tables not included in the design phase:

| Page name | Purpose |
|---|---|
| Admin_security | Holds the username and password to access the site administration facility. |
| Temp_players | Holds the details submitted via the registration form, until the user is approved or disapproved. |
| Temp_security | Holds the security details of any user requesting membership of the web site. These details are held until registration is approved or disapproved. |
| Club_details | Holds the name of the club |

*Fig. 6.1 Tables not included in design*

**Implementation of the WYSIWYG text editor**

The primary reason for the inclusion of a What You See Is What You Get text editor to control the content of this web application is to create an application that is easy to setup and maintain by the average computer user. A WYSIWYG text editor fulfils this objective by automatically generating the html for the text typed into the editor by the user. Code can then be written to save the contents of the editor to a text file. This is done by:

*In aboutustexteditor.asp*

In the text editor file we include the following code to post the contents of the text editor to the asp page we want to display the content.

```
<!-- Posts the contents of the text editor to the asp page as a string -->
Dim strFormName = "myForm"
<form name="myForm" method="post" action="../../aboutus.asp">
…
<input type="submit" name="Submit" value="Confirm changes" onclick="window.close();"
>
```

*In aboutus.asp*

In this page we read in the data posted from the text editor and write it to a text file. E.g. aboutus.txt.

```
<!-- Reads in data posted from the editor  -->
Dim strEditorContent    'Holds the content from the editors text area

strEditorContent = Request.Form("textarea")

'If there is nothing to preview then say so
If strEditorContent = "" OR IsNull(strEditorContent) OR (InStr(1,  strEditorContent,     "<br
/>", 1) = 1 AND Len(strEditorContent) = 8) Then
        strEditorContent = "<br /><br /><div align=""center"">" &
strTxtThereIsNothingToShow & "</div><br /><br />"

Else

<!-- Write the contents of the text editor to aboutus.txt -->

MyFile = Server.MapPath("textfiles/aboutus.txt")
Set FSO = Server.CreateObject("Scripting.FileSystemObject")
Set TSO = FSO.OpenTextFile(MyFile, 2, Create)
TSO.write strEditorContent & vbcrlf
```

The next thing we do is read in the contents of the text file, to provide the content for the page. We do this using:

```
<!-- Read the contents of a text file -->
strFileName = Server.MapPath("textfiles/aboutus.txt")

' Instantiate the FileSystemObject
Set objFSO = Server.CreateObject("Scripting.FileSystemObject")

' Use Opentextfile Method to Open the text File
Set TS = objFSO.OpenTextFile(strFileName, ForReading, Create)

If Not TS.AtEndOfStream Then
   Do While Not TS.AtendOfStream
      strLine = TS.ReadLine ' Read one line at a time
      Response.Write strLine ' Display that line
   Loop
End If
```

**Implementation of the web interface**

Below is a screen shot of the final interface. It is consistent across all pages of the site.

*Fig 6.2 Screen shot of the final interface*

**Table based layout**

The layout of the interface was constructed using table tags in html. Figure 6.3 shows a graphical representation of how these tables were implemented.
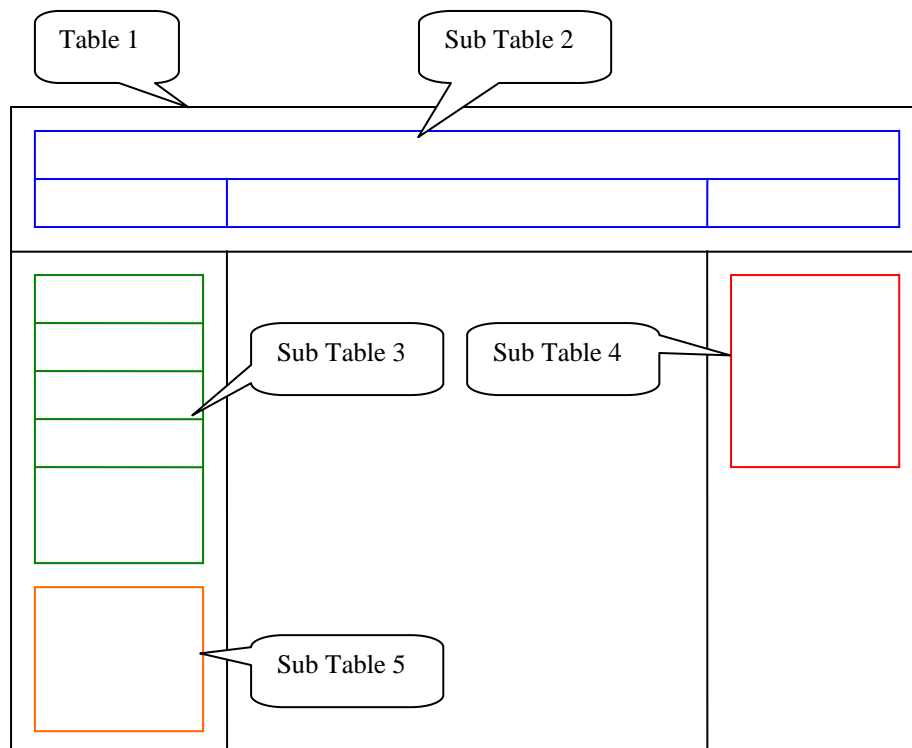


*Fig 6.3 – Shows the table based layout of the interface*

85

**Implementation of the navigation**

The code snippet below shows the code used to organise the left navigation. This code is contained in the file navigation.asp.

```
<TABLE WIDTH=125 BORDER=0 CELLPADDING=0 CELLSPACING=0 HEIGHT=
"367">
  <TR>
    <TD height="2" width="149"><IMG SRC="images/navspacer2.gif" WIDTH=147
HEIGHT=2 ALT=""></TD>
    <TD ROWSPAN=10 width="8" height="54"><IMG SRC="images/spacer.gif"
WIDTH=8 HEIGHT=8 ALT=""></TD>
  </TR>
  <TR>
    <TD height="22" width="149"><a onmouseover = "image1.src =
'images/home_button2sel.gif'; " onmouseout = "image1.src = 'images/home_button2.gif';"
href="index.asp"><img name = "image1" src= "images/home_button2.gif" border=0
width="147" height="22" align="top"></a>
    </TD>
  </TR>
…
…
</TABLE>
```

A server-side includes is then used to include this file in each web page.

```
<!--#include file="navigation.asp" -->
```
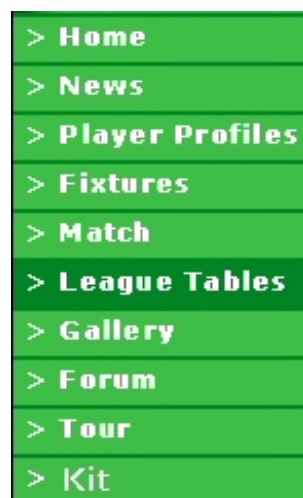


*Fig 6.4 – Screen shot of the main navigation*

A path of the pages visited is displayed in the top left of the main content section on all pages to show the user where they are in the application. This conforms to Thomason's [2] usability checklist.

*Fig 6.5 – Screen shot of the squad selections page*

**Implementation of the Site Administration interface**

Figure 6.6 shows a screen shot of the administration interface. As intended icons are used to support links and make it clear to the user what function will be performed by clicking on the link. In case the icons are unclear to some users, the icons are supported by alt text, which appears when the mouse is positioned above the icon.

# Site Administration

## Setting up the website

▷ **Set up club details**

**Editing page content...**

▷ Homepage

▷ News

▷ Player profiles

▷ Fixtures

| 1st team | 2nd team | 3rd team | 4th team |
|----------|----------|----------|----------|
|          |          |          |          |

Archive fixtures

▷ Match

▷ Squad Selections

| 1st team | 2nd team | 3rd team | 4th team |
|----------|----------|----------|----------|
|          |          |          |          |

▷ Match Reports

▷ Player statisics

▷ League tables

▷ Gallery

> Upload a picture

▷ Forum

▷ Tour

▷ Latest Headline

## Right Hand Side Navigation

▷ External Links

## Bottom Navigation

▷ About us

▷ How to find us

▷ Contact us

Help

**Administration tasks**

▷ Control Registration Requests

▷ Cancel Membership

▷ Manage members contact list

▷ Manage third party contact list

   ▷ Add contact

   ▷ Edit contact

   ▷ Delete contact

▷ View email list

▷ Manage kit selection

   ▷ View orders

**Administration facilities**

This section of the report gives a detailed explanation of those facilities made available to the user that help meet the generic requirements identified in section 4.5.5. Supported by screen shots and code snippets it details how these facilities work and how they are implemented.

A major part of all the facilities is adding data to a table and reading records from a table, so we will first cover examples of the code used to do this.

**Adding form data to a table**

Below in an example of the code required to add form data to a table.

```
<%
'Add a new record
objRS.AddNew
objRS.Fields("title") = Request.Form("title")
objRS.Fields("news") = Request.Form("content")
objRS.Fields("date_published").Value = Now()

'Write the updated fields
objRS.Update
%>
```

**Displaying records from a table**

An example of the code used to display the contents of a table is given below.

```
<%
objRS.Open "SELECT * FROM news ORDER BY news_id DESC", adoConn

do until objRS.eof

        'Create a table and loop through the records in the table'
        Response.Write "<table align='center' width='100%' border='0'
        cellpadding='2'><tr><td bgcolor='#3FBE46'><font color='#ffffff' face='Verdana'
        size='2'><b>"
        Response.Write objRS("title")

        Response.Write "</b></td></tr><tr><td background='images/pixi_grey.gif'><font
        face='Verdana' size='2'>"
        Response.Write objRS("news")

        Response.Write "</td></tr><tr><td align='right'><i>Posted  "
        Response.Write objRS("date_published")
        Response.Write "</td></tr></table><br>"

    objRS.movenext
    loop
```

```
        %>
```

**Setting up the club name and team names**

To enable the application to be used by different sports clubs the user is able to customize the club name and team names. The user does this through the administration interface using the form in figure 6.7. The user selects the name they want to change and re-enters a new value. These changes are immediately evident on the web interface.



*Fig 6.7 - Club details form*

**Implementing the generic requirements**

*Requirement 1: The ability to inform members of important notices including upcoming events.*

This requirement is fulfilled through a bulletin board facility solely designed and developed by myself. Important notices are added to the news table through the use of the form in fig and this data displayed on the news.asp page.

*Fig 6.8 – The add news article form*

The code to perform this operation is detailed above. The news.asp page then displays each record in the database using the code also detailed above. Figure 6.9 shows a screen shot of this display.
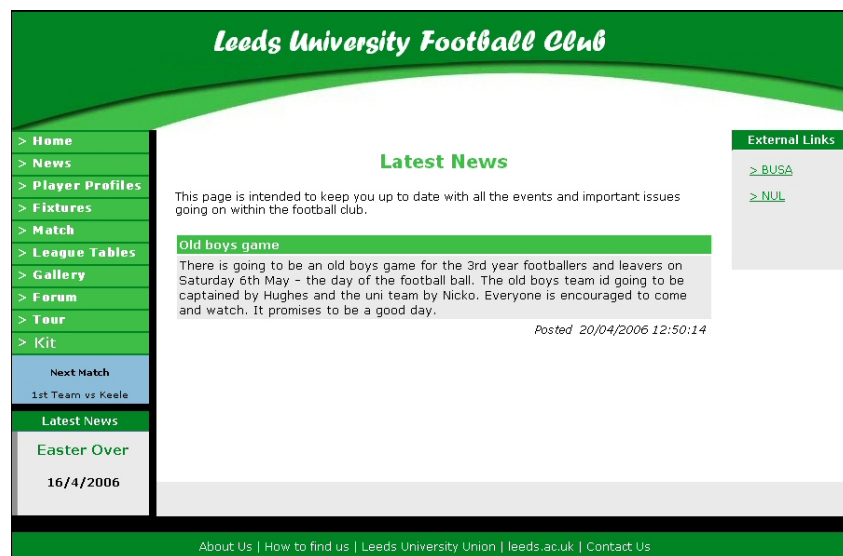


*Fig 6.9 – A screen shot of the bulletin board page*

*Requirement 2: The ability to store and maintain current and past fixtures and results for a VARIETY of teams.*

The fixtures for each team are entered separately in the following form:

## Add 1st team fixture

| Date | Location | Opponent | Competition | Result | Score |
|------|----------|----------|-------------|--------|-------|
|      | H ▾      |          |             |        |       |

Submit  Reset

BACK

*Fig 6.10 – Add fixtures form*

Data from this form is added to the fixtures table and displayed on the fixtures.asp page.



*Fig 6.11 – Screen shot of the fixtures page*

*Requirement 3: The ability to store and maintain player profiles including contact details.*

The profile of a player is added to the temp_players table when they register with the web site. This is done using the registration form below:

*Fig 6.12 – A screen shot of the registration page*

The web site administrator must then approve or disapprove the registration request. If the registration is approved the data in the temp_players table is moved to the players table using the following code:

```
'Takes a hidden id to identify the record to be approved
temp_id=Request.Form("hidden_temp_id")

'Open query string to move date to players table
objRS.Open "INSERT INTO players (title, firstname, surname, nickname, dob, email,
mobile, position_played, comments) SELECT title, firstname, surname, nickname, dob,
email, mobile, position_played, comments FROM temp_players WHERE temp_id = '" &
temp_id & "'", adoConn

'Query string to move security details from temp_security
objRS1.Open "INSERT INTO security (user_id, password) SELECT user_id, password
FROM temp_security WHERE temp_id ='" & temp_id & "'", adoConn
```

If the member is disapproved the details are deleted from the temp_players and temp_security table.

*Requirement 4: The ability to publish a NUMBER of squad selections in advance of matches.*

The layout and content of the squad selections pages are determined solely by the administrator, using the WYSIWYG editor. Details on how the text editor is implemented are discussed above. The

content of this text editor is displayed on one of the four squad selections pages created. A squad page was created for each team in the club. Below is an example of how a squad selection page may look.



*Fig 6.13 – Screen shot of a squad selections page*

*Requirement 5: The ability to publish match reports. (optional)*

The user is able to add, edit and delete match reports through the online administration. To add a match report the user must fill in a summary of the report and upload the full word document using the form in figure 6.14. The contents of this form are then stored in the match reports table, and the matchreports.asp page displays the contents of this table on the interface in figure 6.15.

*Fig 6.14 - Add match reports form*



*Fig 6.15 - Match reports page*

*Requirement 6: The ability to view current and past league tables.*

To achieve this requirement the page gives a description of the leagues participated in and links to the tables on the league organiser's web site.

*Requirement 7: The ability to publicise tour information. (optional)*

The administrator is able to control the content of the tour page using the text editor. Again the layout is entirely the administrator's choice.

*Requirement 8: The ability to store and maintain photographs from social events, tour and competitive matches.*

The design and layout of the photo gallery is different to the layout and design of other pages. This is because the script to implement the photo gallery was freely obtained from [23]. Figure 6.16 shows how the photo gallery looks.
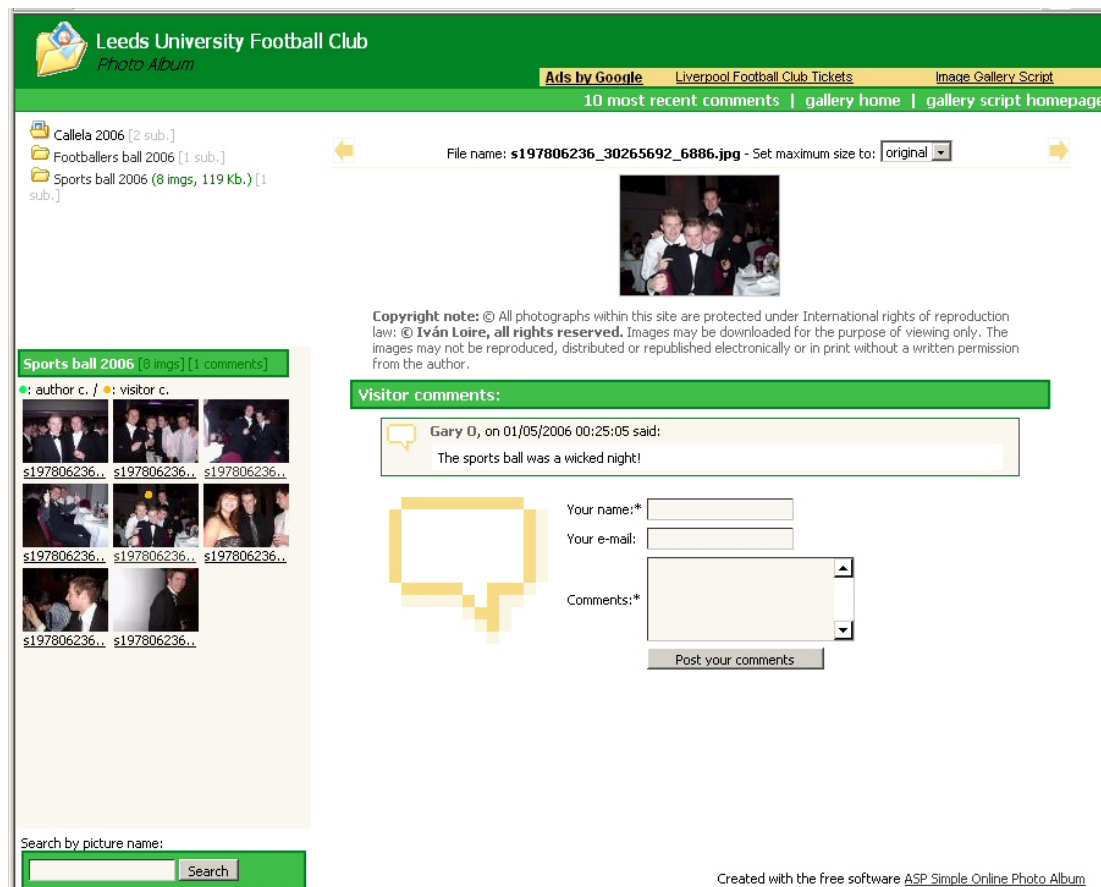


*Fig 6.16 - Photo Gallery*

The administrator uploads the photos to the server and the gallery displays them as thumb nails. The user is able to click on these thumb nails to view a larger picture of the photo and to attach comments.

*Requirement 9: The ability for members only, to discuss topics of interest.*

The design and layout of the discussion forum is also different to that of the web interface. Again the reason for this change is that the script to implement this feature was freely obtained from [24]. To access this forum the user must navigate through the discussion forum homepage. The implemented forum appears as in figure 6.17.



*Fig 6.17 - Discussion forum*

*Requirement 10: The ability for members only, to order training kit. (optional)*

Information on kit items available is stored in the kit_selection table. The administrator is able to add and edit kit information in this table through the kit selection form below.

## Update kit selection

To edit an item of kit, click on the kit ID in the left hand column.

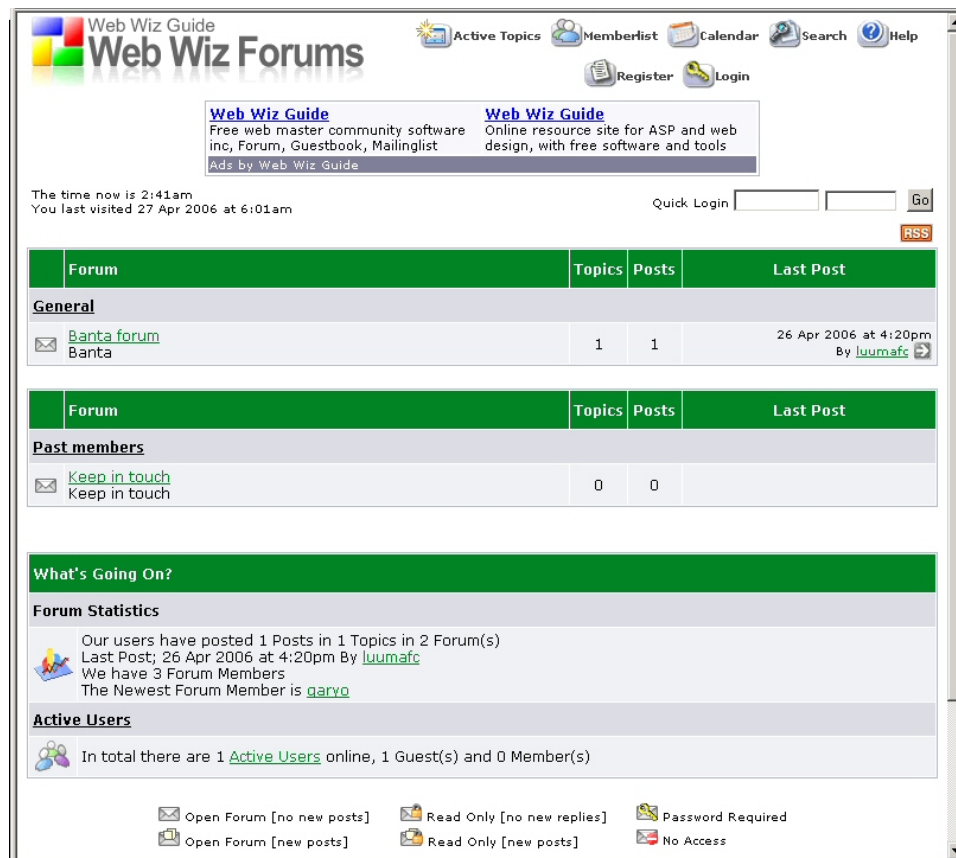| KIT_ID | DESCRIPTION | COST |
|---|---|---|
| 1 | White match shorts | £5.00 |
| 2 | White match socks | £5.00 |
| 3 | Blue Nike training t-shirt | £10.00 |
| 4 | Blue Nike training jumper | £15.00 |
| 5 | Blue Nike tracksuit bottoms | £10.00 |
| 6 | Blue Nike Rain Jacket | £25 |
| 7 | | |
| 8 | | |
| 9 | | |
| 10 | | |

BACK

*Fig 6.18 Kit selection form*

The following code is used to produce this selection form:

```
<table border="1" width="60%">
<tr>
<%
for each x in objRS.Fields
  response.write("<th><font face='Verdana,Arial' size='2'>" & ucase(x.name) & "</th>")
next
%>
</tr>
<% do until objRS.EOF %>
<tr>
<form method="post" action="updatekitselection.asp">

<%
for each x in objRS.Fields
  if lcase(x.name)="kit_id" then%>
    <td width='10%'>
    <input type="submit" name="kit_id" value="<%=x.value%>">
    </td>
  <%else%>
    <td><font face="Verdana,Arial" size="2"><%Response.Write(x.value)%></td>
  <%end if
next
%>
</form>
<%objRS.MoveNext%>
</tr>

<%
loop
adoconn.close
```

98

```
%>
</table>
```

To order kit from this selection a registered member of the web site must first login in order to use the kit ordering form in figure 6.19. This ensures kit orders can only be placed by registered members of this site. Also to ensure this form cannot be reached by typing in the exact URL, the page checks for a session id using the following code:

```
<%
If Session("Valid") <> "True" Then
Response.Redirect "kitorder.asp"
End if
%>
```



*Fig 6.19 - Kit ordering form*

The information entered in this form is added to the kit_order table, and can be viewed in the site administration section.

**Kit Orders**

| Name | Item | Size | Quantity |
|------|------|------|----------|
| Gary Ingham | White match shorts | medium | 1 |
| Gary Ingham | White match socks | medium | 1 |
| Gary Ingham | Blue Nike training t-shirt | large | 2 |
| Gary Ingham | Blue Nike training jumper | medium | 1 |

BACK

*Fig 6.20 - Kit orders*

*Requirement 11: The ability to publish directions to training ground facilities.*

This requirement is fulfilled through the 'How to find us' page. Again the layout and content of this page is controlled using the text editor.

*Requirement 12: The ability to link to a VARIETY of third parties e.g. sponsors, league organisers, kit manufacturers.*

The site administrator is able to add links to external web sites in the right hand-side navigation of each web page. This is achieved by entering the link details in the following form:

**Add External Link**

To add a link choose the name you wish the link to appear as and enter the URL of the page you want it to link to.

Name: [                    ] e.g. Sports

URL: [                    ] e.g. http://www.sports.com

Add Link    Reset

BACK

*Fig 6.21 - Add external link form*

The details entered in this form are added to the external links table and a database lookup provides the content for the right hand-side navigation (fig 6.22).

*Fig 6.22 - Right hand-side navigation*

It is also possible to delete these links from the administration interface.

*Requirement 13: The ability to publish club history. E.g. On team members and successes.*

This requirement is met through the 'About us' page. The administrator controls the content of this page through the text editor.

*Requirement 14: The ability to publish job descriptions of committee members.*

I judged it unnecessary to provide a separate page to fulfil this requirement, as I this could also be fulfilled through the 'About us' page.

*Requirement 15: The ability to store contact lists for third parties. E.g. Printing companies, match officials, tour organisers etc.*

Contact details for organisations and other third parties are stored in the contact list table. These details can be added through the use of the add contact form in figure 6.23.

Once successfully added they can be viewed on the administration interface.

## External Contact List

| Name | Organisation | Tel | Email | Address | Description |
|------|-------------|-----|-------|---------|-------------|
| Gary Ingham | | 07793005812 | gary_ingham@hotmail.com | | Web owner |

BACK

*Fig 6.24 - Table to show contact details for third parties*

**Implementing security**

To prevent unauthorised access to the discussion forum, kit ordering facility and site administration the user must supply a username and password. The login page for each of these facilities is as in figure 6.25.



*Fig 6.25 - Login form*

The username and password entered into the form are then queried against the security table to check that they are authorised. This is done using the following query:

"Select * FROM players p, security s WHERE s.user_id = '"& username & "' AND s.password = '"& password & "' AND s.security_id = p.player_id",

If the query does not return any records the user can not proceed to the next level and is presented with a failed login message.

# Appendix I - Testing

**User testing of the presentation interface**

This was carried out by LUUMAFC committee member Shaun Cooke. Shaun's comments on the interface include:

*On the design:*
- "The overall design is good".
- The Kit button in the main navigation appears to have a different font.

*On usability:*
- Squad selection pages are too hidden. He believes the Match section is probably the correct section for it, but still feels it should be emphasised more.
- Impressed by the league tables page. Links to the tables were extremely quick.

*On the functionality:*
- Apart from the Kit ordering section everything seems to work as it should be expected.
- Believes the News page will be extremely useful.

**User testing of the administration interface**

This was carried out by LUUMAFC potential web site administrator, Adam McKenzie. Adam's comments on the system include:

*On the design:*
- "The main thing is it works, so the design doesn't matter too much, but generally it is ok"

*On usability:*
- The list structure made it easy to determine each administrative task.
- Adam thought the graphics were clear and that if you didn't know what each graphic was for, the alt text helped you determine this.

*On the functionality:*
- He thought the functionality was great and believes the application will be a great success after the registration is fixed, and the discussion forum and gallery have been implemented.
- He believed it would be good to have a separate page to write training schedules down.

**Set up and maintenance testing**

This was carried out by Alex Collingwood the Leeds University Men's Tennis club captain.

The testing process found the following problems:

| Problems | Corrective Action |
|---|---|
| Clicking 'Back' on some administration functions did not re-open the administration page. | Check all administration scripts correctly set the session id. |
| Records should be refreshed after being updated by the forms. | Add code to each update facility to automatically refresh page. |
| When you click in a text field on a form the cursor doesn't start at the left. | Revise code. |
| A username should be allowed to be longer than 16 to allow for email addresses to be used. | Edit properties of temp_security and security table. |
| On the fixtures page the date and results format need specifying somewhere on the form. Also the results should be a drop down box. The score field also needs to allow for data larger than 10 characters long. E.g. 6-0 6-0 6-0. | Include an example next to the text field on the form. Extent the Score field to 20 characters. |
| From kit page, when you click forgotten password the link does not lead to a page. | Correct hyperlink. |
| A member's username and password did not work to enter the kit form. | Revise code. |

Alex commented that:
- It would be good to be able to freely change the colour of the site. Tennis colours are red, blue and white.
- Changing page content is extremely easy.
- The news facility is very straight forward and useful.
- The fixtures facility will be very useful, as the additional team spaces can be used for tournaments.
- The match report and player statistics facilities are good.
- It will be easier to use once you get the hang of it and know your way around better.

- Being able to approve and disapprove members will be a useful facility if the errors are fixed.

- The title font is not very good. It would be good if this could be changed.

- The tennis team are sure to use the application if these problems are solved.

| Site Administration Test Plan | | | | | |
|---|---|---|---|---|---|
| Test no | Task | Expected result | Result | User comments | Corrective action |
| 1 | From the home page, log in using 'luumafc' and 'footy' as the password | The user encounters an error message. | As expected. | | None |
| 2 | Log in using 'luumafc' and 'football' | The user is presented with the administration interface | As expected. | | None |
| 3 | Change the name of the club | Click in 'setup club details' and change club name successfully. | Attempted to change the title on the homepage. After instruction, successfully changed, but when click 'back' it appeared as if no change was made. | Confused if it had been updated or not. | Make setup link more obvious. Include setup instructions. The page needs to automatically refresh after update. |
| 4 | Rename one of the teams | Team name successfully changed. | Failed to enter 'Best team in the club'. | | Extend number of characters for the team name. |
| 5 | Reduce the number of teams in the club to 3 | In 'setup club details' the user selects ID 4 and deletes the name. | Tried delete the team ID. It said record changed but nothing was done. | | None |
| 6 | Add another team to the web site | The user adds a new team to ID 4. | As expected. | Again unclear if it's updated. It says updated but when go back to club setup page, old value still there. | Make the page refresh after update. |
| 7 | Add a sentence to the homepage saying "This website is in its testing phase" | User clicks on correct graphic link and successfully adds the text. | | As expected. | None |
| 8 | Re-edit the homepage and format the text however you like | User encounters no errors. The changes are made successfully. | As expected. | | None |
| 9 | Add a news article to say that the website is currently in its testing | User clicks correct link. Article is added with no | As expected. | | None |

| | | | | | |
|---|---|---|---|---|---|
| | phase | errors. | | | |
| 10 | Edit the text of this news article in some way | User clicks on edit graphic. Article changed. | As expected. | | Impressed by the News facility. Both design and for value it adds to system. |
| 11 | Delete the news article | Article is permanently deleted. | As expected. | Needs to refresh | Make the page refresh after update. |
| 12 | Edit a registered members profile | User clicks on correct link. Profile altered successfully. | As expected. Changed the text to caps in all fields. | | None |
| 13 | Add a 3rd team fixture for Saturday 14th October 2006 against Central Lancashire in the NUL league. Leave the result and score blank. | Fixture added successfully. | Hovered mouse over edit, but didn't click. Read alt text and then clicked on correct link. Fixture added successfully. | | None |
| 14 | Add the result as a 3.0 win | Result and score added. | Tried adding 3-0 but sidn't work. Fault was actually that there was white space after 'W'. | | Ensure mouse cursor starts at 1st character in text box. Or for a quick fix - allow for 2 characters |
| | Edit the date of this fixture. | Date successfully changed and fixture appear in appropriate place of fixture table. | The date changed to 01/01/1990 | | Look at code. |
| 15 | Add the same fixture to the 1st team, 2nd team and 4th team | Fixture added. | As expected. | | None |
| 16 | Edit the 2nd team squad selection page by adding your name as a player. | Correct link clicked. Users name added to 2nd team squad selection page. | As expected. | | None |
| 17 | Edit the 4th team squad selection list by adding a link to the league tables pages | User clicks on hyperlink item and enters a link that works. | As expected. | | None |
| 18 | Add a new match report to the web site. The match can be against | NB - Document uploading absent at time of testing. | As expected, but appeared at the bottom of the match | | Check code for ordering by date. |

| | | | | | |
|---|---|---|---|---|---|
| | Central Lancashire on the 14<sup>th</sup> October 2006. The content is up to you | Match report added successfully. Link to document does not work. | reports page. The date should place it at the top. | | |
| 19 | Edit the text of this match report | Details changed. | As expected. | Unclear if updated. | Make the page refresh after update. |
| 20 | Delete this match report | Report is permanently deleted. | As expected. | Needs to go back to selection of match reports to delete and refresh. | Revise code to redirect to this page after deletion. |
| 21 | Design the player statistics page from scratch, to hold the players name, number of assists and number of goals scored | User will hopefully create a table. | As expected. | | None |
| 22 | Edit the 'tour' page to include a sub title 'Highlights from 2006'. The title must be the same font style as other sub titles on the page | Correct heading applied. | Searched through colours and font size. | | User needs to be informed of each heading style. Perhaps user manual, or note above text editor. |
| 23 | Edit the latest new headline to say "This web site is currently being testing" | User selects correct link and adds the text. | As expected. | | None |
| 24 | Add a link to the BBC web site that will appear in the right hand navigation on the web site (view the web interface if necessary) | The user selects the correct menu option, and adds a hyperlink that works. | As expected. | | None |
| 25 | Delete this external link | The link is permanently deleted. | As expected. | Refresh needed. | Make the page refresh after update. |
| 26 | Edit the 'About Us' page | Page successfully edited. | As expected. | | None |
| 27 | Try editing the 'How to find us' page | Page successfully edited | As expected. | | None |
| 28 | Two members are waiting to be approved for the web site. View the details of these potential members | User selects correct menu option. | As expected. | | None |
| 29 | Approve John Thompson | John Thompson is registered | Appears as if approved. A | | Revise code for |

| | | and his details disappear off screen. His profile is added to profiles page. | check of database shows, security details not successfully approved. | | approving registration requests. |
|---|---|---|---|---|---|
| 30 | Disapprove Joe Bloggs | Joe Bloggs details disappear. His profile is not added to the profiles page. | As expected. | | None |
| 31 | View John Thompson's email address. | User selects correct link. John Thompson email address is viewable on this page. | As expected. | | None |
| 32 | Cancel John Thompson's membership | John's details are deleted permanently. His profile no longer appears on the profiles page. | As expected. | Needs to refresh. | Make the page refresh after update. |
| 33 | Add a referees contact details. The referees name is Joe Bloggs, his telephone number is 07795671234. Assume any other details you wish | The details are stored in the database. | As expected. | | None |
| 34 | Edit the referees details by adding his email address: "joebloggs@hotmail.com" | The user can successfully add the email address. | As expected. | Needs to refresh. | Make the page refresh after update. |
| 35 | Delete the referee | The referees details are permanently deleted and his details are no longer available. | As expected. | Needs to refresh. | Make the page refresh after update. |
| 36 | View club members email addresses | A list of email addresses is available. | As expected. | | None |
| 37 | Add a new item of training kit to the kit selection. Add a "Blue Nike rain jacket" | The user selects the correct menu option, and successfully adds the kit to item 6. | As expected. | Needs to refresh. | Make the page refresh after update. |

| Web Interface Test Plan | | | | | |
|---|---|---|---|---|---|
| Test no | Task | Expected result | Result | User comments | Corrective action |
| 1 | Spend a couple of minutes browsing the website to familiarise yourself | User builds a mental structure of the web site. Allows me to check for recognition and recall | | On the fixtures page a loss should be represented as the score should be separate by '-' instead of a full stop. Squad selections page clear. | None needed. The fixtures form accepts '1-2', so it is easily changed. |
| 2 | From homepage, locate the 3rd team fixtures | 2 mouse clicks, maximum 3 | As expected. Used links in main navigation. | | None |
| 3 | Locate the 1st team fixtures | User clicks on link in left navigation (1 click), or follows link at bottom of 3rd team fixtures page (2 clicks) | As expected. User uses link in left navigation. | | None |
| 4 | Locate 3rd team league tables page. Click on BUSA and NUL links. | 2 mouse clicks to external site | As expected. | | None |
| 5 | Locate 2nd team squad selection page | The user should get there in 3 mouse clicks | As expected. | Squad selection pages are too hidden. | Add rollover button in main navigation bar. |
| 6 | Locate 3rd team squad selection page | User should use link at bottom of 2nd team squad selection page | As expected. | | None |
| 7 | Locate 4th team squad selection | User should click on the link at bottom of 3rd team squad selection page | As expected. | | None |
| 8 | Find information about LUUMAFC | User should go to 'About us' page | Slight hesitation but used correct link. | | None |
| 9 | Find out what the latest club announcement is | User should move to News page | User interpreted this to be the Latest news box under main navigation. | | None |
| 10 | Locate match report for 2nd team Vs Leeds Metropolitan | User should locate the match reports page and select the | As expected. | | None |

| | | | | | |
|---|---|---|---|---|---|
| | University on 19/03/2006 | correct match report | | | |
| 11 | Find the directions to the club | Go to 'How to find us' page | As expected. | | None |
| 12 | Contact the club with an enquiry | User moves to 'Contact us' page, fills in the form and sends successfully. Admin receives an email enquiry. | Used correct link, but form did not work. Page layout of error message was distorted | | Code alterations to the contact form. Correct error message page. |
| 13 | Register as a member | User move straight to registration area via homepage. Successful registration request made. [approved by administrator] | As expected. | | None |
| 14 | Make a training kit order | User move directly to kit area. Successful login. User fills in form and submits. | 2 failed login attempts. The security details are not being approved successfully. | | Revise code to approve membership. |
| 15 | Move to change password area. Enter ANY incorrect user details. | User presented with error message and allowed to try again. | As expected. | | None |
| 16 | Now type in correct details and change password. | Password change is successful | As expected. | | None |
| 17 | Try logging in to kit area with old log in details | Login failed, and error message presented | Login failed | | Revise code to approve membership. |
| 18 | Try logging in with new details | Login successful | Login failed | | Revise code to approve membership. |
| 19 | View your player profile | User moves to profile page | As expected. | | None |
| 20 | Change your profile details in anyway and return to profiles page to view the changes | User clicks on home > personal details. Successful login. Successful change. | Failed login | | Revise code to approve membership. |

# Usability Inspection Results

## A Detailed, General-Purpose Checklist

### Architecture and Navigation

- ☑ Does the structure fit the purpose?
- ☑ Is the navigation scheme clear?
- ☑ Where are you?
- ☐ How do you find what you want?
- ☑ Is there a reasonable number of navbar choices?
- ☑ Are navbar choices logically ordered?
- ☑ Do link names match page names?
- ☑ Are links clearly marked?
- ☑ Is there a clearly marked link back to the home page?
- ☑ Is there an option to search for information?
- ☑ Is there a site map?
- ☑ Does every page make it clear which web site you're in?
- ☑ Does the user have control over navigation?

### Layout and Design

- ☒ Does page size exceed window size?
- ☑ Is layout consistent between pages?
- ☑ Is there a clear focal point on each page?
- ☑ Does the layout work visually?
- ☑ Is alignment used effectively?
- ☑ Is grouping used effectively?
- ☑ Is there good contrast?
- ☒ Is the layout cluttered?
- ☑ Is it aesthetically pleasing?

### Content

- ☑ Is the text clear and concise?
- ☑ Is text organized in small chunks?
- ☒ Are there spelling or grammar errors?
- ☑ Is it clear where to go next?
- ☑ Do dialogues follow a logical progression?
- ☑ Do pages include introductory text?
- ☑ Do multimedia components support the task?
- ☑ Are units of measure clear and unambiguous for international use?
  ☑ Date and time? ☑ Phone numbers? ☑ Address and postal codes?

### Forms and Interaction

- ☑ Do forms support the task?
- ☑ Is it clear where to go next?
- ☑ Are dialogue methods concise and consistent?
- ☑ Are form elements used properly?
- ☑ Are elements grouped properly?
- ☑ Are there clear Submit buttons?

### Graphics

- ☑ Is image quality adequate?
- ☑ Do the images include alternate text?
- ☒ Do the images include size information?
- ☐ Do the images use a consistent light source?
- ☐ Are images stored for maximum compression?
- ☐ Is mouse-over feedback provided? Is it useful?
- ☐ Are animations useful? Are there too many? Are they properly compressed?

### Color

- ☑ Is the choice of colors appropriate for site?
- ☒ Are too many colors used?
- ☑ Are colors used consistently?
- ☒ Are graphics colors dithered?
- ☐ Do color choices work in grayscale?

### Typography

- ☑ Is the text legible?
- ☑ Is the font size large enough?
- ☑ Is the font color appropriate and is there sufficient contrast?
- ☑ Is the text formatted for 10 to 12 words per line?
- ☑ Are there sufficient margins?
- ☑ Are typefaces used properly and consistently?

### Error Tolerance

- ☑ Do users need to remember items across pages or sessions?
- ☑ Are confirmations provided before risky or costly actions?
- ☒ Do error pages provide useful information?
- ☒ Do search-error pages provide search broadening tips?
- ☒ Are risky or costly actions reversible?
- ☑ Are entry errors caught locally?
- ☒ Is help available?
- ☒ Is help task-oriented?
- ☐ Is help contextual?

### Platform and Implementation

- ☑ Is load-time fast enough? Does it load in 3 to 15 seconds?
- ☑ Do all the links work?
- ☒ Are there broken images?
- ☒ Are pages written to be found by search engines?
- ☑ Does the site work with user's browser?
- ☑ Does the site work with user's hardware platform?
- ☐ Does the site work on high- and low-resolution monitors?
- ☒ Are nonstandard plug-ins required? Are they necessary or useful?