An info system to provide details of social establishments in Leeds

Daniel Sherburn

BSc (Hons) Information Systems

2002/2003

Summary

Project Aim

The aim of this project is to provide students, particularly those without any experience of Leeds, a system that will supply useful information about student nightlife. The system will provide details on the pubs, clubs and bars in the Leeds areas of headingley, hyde park, and City centre. A purpose for the project would be for those wishing to come to Leeds University to be able to have an insight into what student nightlife in Leeds is really like. The system would also be extremely useful to the first year students who may be unfamiliar with the Leeds social scene. Currently there are systems similar to this but none that cater solely for students.

Objectives, minimum requirements and future enhancements

My Personal objectives for the project*:*

- To gain further project management skills, learning how to manage time efficiently and complete the project from start to finish.
- To enhance my web development and database skills.
- To learn new tools and techniques

The objectives for the project*:*

- To evaluate a selection of software methodologies to decide which is the most appropriate to follow.
- To evaluate possible software tools for the implementation of the project and understand why one piece of software is more appropriate than the others.
- To carry out the requirement analysis in order to design the system.
- To design, implement and test a normalised database along with a user interface.
- To design, implement and test a web interface that connects to the database.
- To evaluate the project in order to see if requirements have been met or exceeded.

Minimum requirements for the project

- To evaluate possible software tools for the implementation of the project and understand why one piece of software is more appropriate than the others.
- Produce a prototype relational database that holds information about pubs, clubs, bars and their promotions.
- Configure the database to provide a user-friendly interface that allows an administrator to add/delete/edit the information held.
- Provide a web front that allows users to access the information held in the database.

Possible Enhancements for the project

- Provide details of quizzes and food for pubs.
- Provide users with web-based search facilities so the database can be searched in correspondence with the criterion that the user selects.
- Provide a web-based message board that users can post on.
- Produce a user manual for the database
- Provide a pop-up map for each establishment giving detailed directions.

Deliverables

- A working prototype database system
- A web Interface that connects to the database
- A user manual (if future enhancement is achieved)

ACKNOWLEDGEMENTS

I would like to thank the following people who have provided me with a great level of support throughout my project.

Ann Roberts, my supervisor who has provided me with tremendous encouragement and support through all stages of the project. Thank you for all the help you have provided me with over the last year.

Sarah Fores, my assessor who provided me with useful suggestions during the progress meeting.

Stuart Roberts, who was kind enough to provide assistance with any database-related queries that I had.

Ollie Clark, who was also kind enough to provide assistance with any database-related queries that I had.

My Parents, who continue to support me in everything I do

All my friends within the school of computing, who have provided me with much needed humour and help whilst spending my time in the level 7 lab!

CONTENTS PAGE

1. INTRODUCTION

1.1 Introduction to the problem

During my time at University I have never come across a system that provides students with information about social establishments in Leeds. Currently information about promotions or social establishments is spread by word of mouth or flyers. Although this is often a useful approach of distributing information, it is not always readily available. A system that could readily provide this information would be of great benefit to students. There is often the case that students will go to a social establishment and find out that a particular promotion that they still believe existed, doesn't exist. There is a need for a system that will keep students updated with this kind of information to prevent such mistakes.

1.2 Current systems

Before I start my project, it is important to evaluate some systems that are similar to the one I wish to propose. Some systems do exist that provide some relevant information, but none that are catered for student needs. Using suitable search criteria in Google, I was able pick out a selection of systems to be evaluated to see what the benefits and drawbacks are.

1.2.1 My Leeds

The system is not specific to pubs, clubs and bars but provides information on gigs, concerts and record shops. The website is not particular user-friendly with no real menu to help the user navigate through the system. Details of bars and pubs are displayed but the information provided is very limited. The system offers no search facilities that would enable users to find details of social establishments that they were interested in. On the positive side, the system does provide a pop-up map that shows the precise location of each establishment. The clubs page provides adequate details on promotions but once again lacks any mechanism whereby users are able to refine their search. A principle of a good interface is simplicity. The interface is not simple; it is not straightforward and is clustered with unnecessary adverts and objects.

1.2.2 BBC Leeds Club Guide

This website provides a simple interface where users can retrieve information on bars and clubs. However, searching is non-existent and users are required to browse through all the clubs and bars before they identify any that will stimulate interest. The system becomes useful if the user knows which establishment they are looking for, but is not particularly useful for users who don't know what they are looking for. The system provides no information about pubs or food, which are as important to students as clubs and bars are. There is no use of images that can help the user to visualise what the establishment is like.

1.2.3 Webflyers Leeds Nightlife and Clubbing guide

This provides the most simplistic and straightforward interface out of the three systems. The user is presented with an immediate search screen that lets you search by ‘venue’, ‘night’, ‘music’, and ‘last updated’. However, the functionality of the system is poor because each search produces no results. This is likely to be due to the fact that no entries are present in the database at the current time. The fact that the system produces no search results makes it impossible to state whether the system provides details on pubs, clubs and bars.

1.2.4 Web site to provide info on Clubs, Pubs and Bars for Leeds Students Final Year Project

Fraser William Nelson Stride conducted this final year project. The reason why this has been chosen for evaluation is due to the fact that it is a project of similar nature to my own. The system was originally designed to include details of pubs, clubs and bars but due to time constraints was only able to produce information on pubs. The user interface that this system provided was one of simplicity and familiarity. The system provides a pub search page that allows a user to search on different attributes. However, this search functionality is somewhat limited due to the fact that the user can only search on individual attributes. A user may want to select multiple attributes in order to refine their search and would be unable to do this using this system.

1.2.5 Summary of current systems

From the research carried out, it seems to be the case that there is not a single system that satisfies student needs. The third system evaluated was the only system that had a simplistic, easy to use interface. The other two systems had interfaces that were designed poorly and cluttered up with too much information and too many adverts. None of the systems provided details for all pubs, clubs and bars. It was often the case that only details of clubs and bars were provided. It can be seen from the current systems available that there is a need for a system that provides students with details of pubs, clubs and bars. The interfaces from the three systems evaluated can be seen in Appendix B.

1.3 Project Management

In order to achieve the objectives set out by the project, an initial project plan had to be developed. To achieve this required following an appropriate methodology. Following the methodology would lead to key milestones that result in the completion of the project.

1.3.1 Selecting a software methodology

In order to complete this project, a structure is required that should be followed step by step. In order to do this my project must follow a software development methodology. A software development methodology can be said to be:

" A collection of philosophies, procedures, techniques, tools and documentation which aid the systems developer with the implementation of a new information system" [1].

These methodologies have the purpose of forming a requirements process so that the actual requirements of the system are not misunderstood. Many methodologies exist which I will be looking at and evaluating to decide on the best approach for my project.

1.3.2 5 Stage Software life cycle

This process involves 5 unique stages for which the software is supposed to under take during its life cycle.

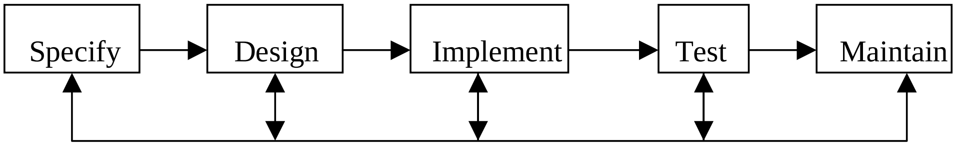


Figure 2.1: The 5 Stage Software Life Cycle [2]

"The lifecycle is an orderly set of activities conducted and managed for each development project. The processes and methods are the machinery for a lifecycle implementation" [3].

This particular approach is iterative so that previous stages in the cycle can be re-visited if they need to be. Each of the five processes specified in Figure 2.1 relate directly to the project where the problem will be specified, the system will be designed, implemented, tested and possibly have someone maintain it. This kind of methodology allows you to construct the project modularly so that various functions of the system can be produced in separate modules. Each function in a module will have been designed, implemented and tested before the next module was undertaken.

1.3.3 *SSADM*

"Structured Systems Analysis and Design Method (SSADM) is an analysis and design method that ensures an information systems specification is correctly defined and monitored." [4].

SSADM offers a hard approach to information systems development such that the stages, tasks and deliverables are specified in advance. The methodology represents a data driven method where the basic assumption is that systems have an underlying generic data structure which changes very little over time. The only thing that may change is the processing requirements. This methodology supports the following stages:

- Feasibility study
- Requirements analysis
- Requirements specification
- Logical system specification
- System design

Each of these stages is completed before the next stage is started. The problem with SSADM, is the fact that it only deals with the analysis and design stages. This is not a full scope to base a whole project on from start to finish.

1.3.4 Rapid Application Development (RAD)

"Rapid Application Development is more than a requirements elicitation method, it is an approach to software development as a whole (Hoffer et al, 1996). As the name suggests, the RAD aims at delivering solutions fast. Technical excellence is secondary to the speed of delivery" [3]. It becomes apparent that RAD would be appropriate for projects that are on a relatively small scale and bound to a tight project deadline. This project however, is stretched over a long period of time and it would be unjust to produce a solution that is on time but does not meet minimum requirements. Speed of delivery is not the most important factor for this project.

1.3.5 Evaluation of methodologies

Due to the nature of my project, it would be most appropriate to adopt the 5-stage software life cycle approach. This is due to the fact that my project involves implementing a system where all stages outlined in this cycle are going to be used. Another benefit of this approach is that the methodology allows you to re-visit earlier stages, which will be appropriate to my project as re-design and implementation maybe required. RAD prioritises speed of delivery over technical excellence so this was deemed inappropriate to this project. SSADM only deals with the analysis and design sections of a project so is deemed an inappropriate methodology considering implementation and testing is required.

1.4 Project Plan

The remainder of the report is constructed into a further 6 chapters. This section outlines what the chapters are and what they consist of. A detailed project plan with timescales can be seen in the Gantt chart in Appendix C.

- Comparison of software tools – The purpose of this chapter is to identify the possible software tools for the project and to evaluate them to see which is the most appropriate to use. (October 2002-January 2003)
- Analysis – This chapter focuses on identifying the possible users for the system and gathering together their requirements. Functional and non-functional requirements form the basis of an evaluation criterion that will be used in conjunction with the evaluation chapter. (November 2002 – January 2003)
- Design – After gathering the requirements from the analysis stage, the database can be designed using appropriate ER-modelling tools. The design of the user interface is also included in this chapter. (January 2003 – March 2003)
- Implementation – having designed the system, the time comes to implement it. The chapter shows how various components of the system were implemented and how the selected methodology was followed to construct it. (February 2003 – April 2003)
- Testing – This chapter takes into consideration two different forms of testing. Unit testing provides feedback on the functionality of the system while acceptance testing shows whether users have accepted the system. (January 2003 – March 2003)
- Evaluation – This chapter concludes the project by examining the requirements gathered in the analysis section to see whether they have been achieved. The chapter also provides information about any future improvements that could be made. (April 2003)

2. COMPARISON OF SOFTWARE TOOLS

2.1 Databases and Database Management systems

"A database is any collection of facts that are systematically organised" [5]. To be able to perform operations like insert, delete and update, the database needs to be managed by a substantial package of software. This is called a Database Management System (DBMS). "A DBMS is a generalized software package for implementing and maintaining a computerized database." [7] A DBMS is responsible for many tasks such as to provide functions that only allow users with permissions to access and modify information in the database. A database can be of two types, a flat file database or a relational database. A flat file database is the most basic type, holding all data in a single table. The simplest way to understand what this kind of database is, is to think of a filing cabinet which is filled with individual pieces of paper, where the only way to access this information is to go through them one by one. The problem with flat file databases is that they suffer from data redundancy where data is unnecessarily duplicated. A relational database counters this problem by breaking the database into different tables that reduces data redundancy. It makes use of relationships where tables can be related by a common field so the data values don't have to be duplicated in the same table. There are different DBMS that provide different languages but the one which is most common is Structured Query Language (SQL) which is de-facto standard and the first Relational database language. Almost every DBMS understands SQL so it becomes easy to import and export data from one database package to another. The language is very simple reading almost like English, which is very helpful when it comes to actually learning how to use it. Data can be manipulated easily using SQL whether it be something as simple as deleting a record from a table or something difficult like updating values when an event occurs. The possible Relational DBMS software will be looked at and compared to see which one suits the needs of my project the most.

2.1.1 Oracle Server

"Oracle is currently one of the main world producers of software, and the range of products offered has its foundation, the database management system Oracle Server, available for most types of computer." [6]. Oracle server is typically used for large organisations who require a database management system with high functionality and capacity. However, the cost of such a package makes this project infeasible as neither the university nor I hold a licence for this product. "An important strength of Oracle though, is its availability on various platforms, which facilitates the integration among databases at various levels in an organisation." [6]

2.1.2 SQL Server

SQL server uses the standardised language of SQL. SQL server is an example of a DBMS that is designed to receive many requests from various clients wishing to manipulate the database in some way or form. The package offers superior performance with users able to manipulate queries using the Query Analyser. Together with the Enterprise manager, SQL server offers a fully integrated DBMS package that can be used for all scale operations. The fact that SQL server does not provide tools to provide a user interface, limits the applications it can be used for. A minimum requirement of this project was to provide a user interface to add/delete and modify information held in the database. SQL server cannot provide this from its database management system.

2.1.3 Microsoft Access

"Access provides a database engine and a graphical user interface for data definition and manipulation, with the power of SQL" [7] Access is one of the most well known Relational DBMS on the PC. "It is considered as part of an integrated set of tools for creating and managing databases on the PC platform." [7]. Access can be used for a number of reasons whether it is for personal use or for setting up a small business. Access is designed for all types of users whether they are relatively novice or whether they are technically gifted. Access offers a wide variety of facilities such as forms, reports, queries, macros and modules that can be developed very quickly through the use of wizards which are available on this software. These are interactive programs that guide the user through different steps in order to create their object. It allows a user to create something easily that would be difficult to do if programmed. Access also conforms to the Microsoft Open Database Connectivity (ODBC) standard and this allows for other applications such as web development tools to be used as a front end to databases stored on a different server. However the downside of Microsoft Access is that it is a low capacity database, one that would not be suitable for organisations of a large nature.

2.1.4 Evaluation Of DBMS Packages

Due to the performance factor, it appears that both SQL Server and Oracle should be considered for development. However due to the fact that the university does not own a copy of Oracle makes this an unrealistic option. To purchase a copy of Oracle would require a very large sum of money that would make a project such as my own infeasible. Although most of the computers in the computing department have a licence for SQL Server, it is unlikely that many other desktop PC's will have this software. Again, a large amount would need to be spent on this software, which is simply infeasible. There is also the fact that SQL server does not provide a user interface that is required for the administrator to add/delete/edit information in the database.

This leaves one option that is feasible for this project. Microsoft Access is favourable in a number of ways. The first reason for choosing Access is due to the fact that the database I intend to produce will have a low capacity, and would not require the superior performance of SQL server or Oracle. Whilst at University, I have increased my knowledge of SQL by learning new skills in modules such as DB11, DB21, DB31 and DB32. However, these skills cannot be demonstrated in SQL Server because of the fact that it does not provide a user interface that is required for this project. Microsoft Access and SQL Server do however rely on the same language so I feel I can apply these new learnt techniques in Microsoft Access. The other major advantage of Microsoft Access is that it offers a user interface in the form of reports, forms and switchboards.

2.2 Server-side scripting tools

With a project of this nature, the system has to have the ability to respond to client responses such as entering a value into a HTML form. The data that is provided by the web client is processed by web software running on the server. This software dynamically generates a new page with regards to the users input. The server side scripting offers the advantage that the connectivity between server and client can be done much simpler than trying to do it all in the HTML. Various server-side scripting tools have been described below:

2.2.1 Common Gateway Interface (CGI)

"The CGI is the most straightforward way to process responses from the Web. In a typical set up, a HTML form is submitted to the server, and that form data is then passed through CGI to a processing program." [5]

CGI usually operates with the programming language Perl and the way it works is that it uses the embedded SQL written into it, to call CGI commands running on web servers to dynamically create new web pages. The advantage of using CGI is that it is universal, so portability is not an issue. The problem with CGI though is that it is fairly slow and requires a great deal of knowledge of the programming language. The reason that it is slow is the fact that it has to create a new instance of the processing program every time a client accesses the program through CGI.

2.2.2 Active Server Pages (ASP)

"ASP (Active Server Pages) is a technology developed by Microsoft. Pages using ASP are primarily developed in JavaScript, VBScript , or PerlScript and are integrated into the HTML of your Web pages. The ASP code is compiled on-the-fly by the server and the resulting output is standard HTML. By using ASP, "Web pages can be dynamic, full of ever-changing content, and browser independent." [8]

The way it works is that the client requests an ASP from the web server for which the ASP engine responds by executing the scripting on the page. The result is a dynamically created HTML page that is sent back to the server so the client can access it through their web browser. ASP includes the following technologies:

- ActiveX Database Objects (ADO) – are server side components that dynamically connect data in a database to a web page. They provide access to both client and server information for building dynamic web pages.
- Advanced Data Connector (ADC) – this provides client-side database access. It enables the database data to be sent to the browser so that the load on the server and the network traffic is reduced.
- VBScript – this scripting is responsible for binding together ADO and ADC into the Active Server Pages model.

2.2.3 Hypertext Pre-Processor (PHP)

PHP is similar to ASP in the way that it's a language that is embedded in the html. PHP is a simple language where "tasks accomplished with some degree of difficulty in many other languages, can be swiftly executed with but a few lines of PHP code." [9]. However PHP requires an additional layer in the ADO model to achieve data independence. This makes the process of retrieving pages much slower.

2.2.4 Evaluation of Server side Scripting tools

ASP "provides the highest level of abstraction of any of the Microsoft web database tools", [5] and has been identified as the "perfect choice for data masters without strong programming experience". The main reason for using ASP is the fact that it generates pages much faster than the other software tools. CGI scripts for example are particularly slow because they require a new instance of the processing program to be created every time a client accesses the program through CGI. The other justification for using this language is the fact that it is compatible with the other Microsoft software and applications that have been selected earlier as the most appropriate. This allows for integration between the selected tools.

2.3 Web development tools

The website produced is there to enable the user to retrieve required information from the system. The website is the interface to the people that are going to use it (i.e. students), so extra attention must be paid to Human Computer Interaction (HCI) issues and user-friendliness. There are various ways that you can create a website, some may decide to program them using specific languages or others may use an application such as Microsoft FrontPage or Macromedia Dreamweaver.

2.3.1 HTML (HyperText Markup language)

"The main function of HTML is to provide information that the browser can use to make formatting decisions for displaying the contents of a web document." [5]. Since HTML primarily a text-based language, it has the advantage that it is much easier to create and transfer onto other platforms and operating systems. It uses the notion of tags such as <html> to open the language and </html> to close it. The tags tell the browser how the page is to be formatted but the important fact to note about this is that the HTML documents are static. This means than the formatting is embedded in the web page, stored on the relative server and then accessed by the client wishing to view it. Different browsers will format the HTML in a slightly different way but will understand the HTML language. Internet Explorer and Netscape Navigator are the most common of these browsers and will be used to test whether the HTML displays correctly. A typical way of writing this HTML, if I were to construct my website in this way, would be to use notepad.

2.3.2 Microsoft FrontPage

Microsoft FrontPage is a user-friendly tool that allows the designer to select from 3 different views in order to see the progress that they have made. Normal view provides the user with an interface that allows them to create a website with various tools on offer to them via certain toolbars. The HTML is automatically created for you. FrontPage also offers you the facility to write the HTML yourself or to self modify the code that had resulted from creation in the normal view. This can be done in the HTML view. The final view, the preview view allows the user to see what they have created in FrontPage's version of a browser. "Along with its other features for creating web pages, FrontPage can be used to create web databases and pages that incorporate dynamic database content." [5] The HTML view in FrontPage can also be used to write the ASP code. FrontPage automatically recognises ASP code and formats it so the colour of the code is different. This is a useful feature that allows you to distinguish from ASP and HTML.

2.3.3 Macromedia Dreamweaver

"The world's best way to create professional websites is now the easiest way to build powerful Internet applications. For the first time, you can work in a single environment to quickly create, build, and manage websites and Internet applications." [10]. The main function of this program is its ability to create pages using Dynamic Hypertext Mark-up Language (DHTML). This gives the browser the ability to alter a web page's look and style after the document has loaded and thus making it more dynamic such that the code on a website can be changed without having to interact with the server. Dreamweaver not only offers tools for DHTML but also for creating cascading style sheets and tools for client side scripting. Dreamweaver has certain elegance about it, with the ability to create flashy graphics as well as the ability to import database definitions to a database connection. Like Microsoft FrontPage you can edit the HTML manually or through the application interface giving the user the best of both worlds in the same application.

2.3.4 Evaluation of Web development tools

The choice of software for this tool is going to be Microsoft FrontPage. The Justification for this choice is the fact that Microsoft tools, when combined, have the ability to combine as a full integrated solution. Microsoft FrontPage also offers ODBC connectivity that allows you to openly connect to a database with the help of server side scripting. The main reason however for selecting Microsoft FrontPage is the fact that it provides you with three different views outlined in section 2.3.2. The way that I plan to construct the website is to write the html in the ‘html view’ and continually flick over to the other two views to make sure the html is formatting to my satisfaction. If the html was written primarily in notepad, then time would be lost continually opening the html page in some browser to make sure formatting is correct.

2.4 Client-Side Scripting tools

Until recently the emphasis on the processing was done on the server side while the client just received it and displayed it on the screen. Now however with the introduction of the client server model, it has been identified doing some processing on the client side can reduce that network traffic. The idea is to incorporate more dynamic elements into the web page so that the browser can interpret them rather than passing it on to the server. Client-side processing becomes useful for inputting data into a web form where validation rules and constraints can be set.

2.4.1 VBScript

VBScript is based on the Visual Basic language. Although it does not offer the full functionality of the programming language, it does provide a high performance scripting language that is also easy to use. VBScript is responsible for binding together the ActiveX Database Objects (ADO) and the Advanced Database Connector (ADC) into the Active Server Pages model.

2.4.2 JavaScript

JavaScript is a “scripting language which looks like a loose amalgam of C, Java and Basic and which implements much of the functionality of applets” [11]. Developed by Netscape, JavaScript is a text-based language which is directly embedded into the web page but can also be viewed in Internet Explorer.

2.4.3 Evaluation of Client side scripting tools

The fact that I have decided upon ASP has had an impact on what scripting I use for client side processing. ASP is very compatible with VBScript in the fact that it is needed to bind together the elements of ASP, ADO and ADC. For this reason VBScript will be used as the choice of software.

3. ANALYSIS

3.1 Requirements Analysis

The purpose of this section is to identify the different users for the system and determine their requirements. There are essentially two main users for this information system, administrative users and student users that are discussed below.

3.1.1 Administrative users

The administrator(s) of the system are responsible for adding/deleting/editing information in the database and keeping the system up-to-date. There are two different ways of administrating the system. One way would be to have each owner of the social establishment update their own details themselves. Potentially this would be a very efficient system that would be kept updated by each administrator. However, a number of problems do arise from this approach:

- The most obvious problem is the fact that some of the administrators will forget to update their own establishment, resulting in the information being out of date.
- This method of keeping the system up-to-date would more than likely have the implication of training for these users. It is very unlikely that every administrator will have the computer literacy required to keep the system up-to-date.
- Due to the fact that there will be a large amount of administrative users, brings about the possibility of data inconsistency, as each user will have different ways of doing things.
- Another problem with this approach is the possibility that some of the administrative users may not accept the system. It would be extremely important that all these users accept the system otherwise it would be extremely difficult to keep it up-to-date.

The other way of administrating the system would be to have a single member who would keep the system up to date. Using this approach would require training for only one administrator and would also counter the problems that are mentioned above. The implication of this approach though is the problem of actually getting the information. The way for this to be achieved is for the administrator to set up good links with each social establishment so they would receive an email or telephone call from each owner when a modification is made. The administrator may also decide that they too can make the phone calls to find out about any new promotions that exist. The reason why this approach would be more efficient is the fact that changes to social establishments and new promotions do not happen on a regular basis so it would not take much of the administrator's time to update the database. However, finding someone to administrate the database would be a difficult task. The University Union is considering deploying the system as part of the University open day. It would be unrealistic to assume that this project will definitely go into deployment.

3.1.2 Student users

Student users are essentially those who are going to request the information entered in the database. Each user will have different requirements for what they want to get out of the system. Some students will want to use the system to find out which pubs have a pub quiz where others may want to find out which pubs have a pool table. To find out their differing requirements it is important that a way of gathering this data is constructed. Clubs and Bars will all have the same physical attributes and are only likely to differ by music, location, dress code and price. It is for this reason that no further research needs to be carried out to see what further database attributes for these entities need to be included. However, pubs will differ by physical objects such as a pool table or a jukebox. It is important to find out what pubs have different to offer so further research was carried out.

3.2 Gathering of data

One of the most important reasons for carrying out a requirements analysis is to enable you to gather data so the solution to the problem can be designed. The most obvious way of gathering such data would be to produce questionnaires that are sent out to a selected sample size and analysed upon return. However many problems arise with questionnaires, some of which are given below. [16]

- Questionnaires, like many evaluation methods occur after the event, so participants may forget important issues.
- Questionnaires are standardised so it is not possible to explain any points in the questions that participants might misinterpret. This could be partially solved by piloting the questions on a small group of students or at least friends and colleagues. It is advisable to do this anyway.
- Open-ended questions can generate large amounts of data that can take a long time to process and analyse. One way of limiting this would be to limit the space available to students so their responses are concise or to sample the students and survey only a portion of them.
- Respondents may answer superficially especially if the questionnaire takes a long time to complete. The common mistake of asking too many questions should be avoided.
- Students may not be willing to answer the questions. They might not wish to reveal the information or they might think that they will not benefit from responding perhaps even be penalised by giving their real opinion. Students should be told why the information is being collected and how the results will be beneficial. They should be asked to reply honestly and told that if their response is negative this is just as useful as a more positive opinion. If possible the questionnaire should be anonymous.

The problems with questionnaires outlined above could have a hindrance on the project in terms of inaccurate data and wasted time. The third point mentioned above identifies that some people will often just complete the questionnaire with no real effort to study the questions. The simple reason for this is the fact that they wish to complete the questionnaire in the shortest time possible. Another important point to mention is the fact that only a small amount of submitted questionnaires are returned. This has the implication of selecting a much larger sample size than is realistically needed. Even when the results are returned it is often the case that the process of sending the questionnaires out to receiving them back, is quite a lengthy period of time.

The problems pointed out above were the reasons why questionnaires were disregarded as an appropriate source of gathering information. Instead, a process of informal interviews was decided as an appropriate alternative because of the fact that you can actually interact with the potential user. Adopting this approach allowed me to clearly explain to the interviewee what the questions were intended to mean. Conducting these informal interviews in the environment of a pub helped to remind the user why they were in this particular pub and not another one. Eight different pubs were visited in the headingley and hyde park areas with 5 different people from each pub selected. Each person was asked to discuss the question:

"Why did you choose to come to this Pub tonight?"

This question helped the interviewee to have a think about the reasons why they did actually come to this Pub. This helped provide responses that I personally believe were more accurate than if they had been carried out by a questionnaire.

3.3 Analysis of results

Having completed the informal interview phase it was time to analyse the results that had been gathered. Table 3.1 shows a list of factors together with their percentage that would determine why a student would go to a particular pub.

Percentage	Factor
75%	Close to house
70%	Widecreen TV
68%	Serves a good pint
63%	Pool Table
60%	Served Food
55%	Pub quiz
48%	Jukebox
40%	Quiz machine
10%	Dartboard

Table 3.1: Factors to influence a choice of pub

The results are shown with the highest percentage factor at the top. A total of 9 factors were found, that help to determine why a person chooses to go to one pub over another. A total of 40 people were interviewed as there were 5 people from each of the 8 pubs. Each person was allowed to give as many factors as they wished. The percentage represents how many people out of the 40 interviewed considered that factor to be important.

- As the sample size was relatively small, it was decided that any factor having a percentage of 40 or above would be including in the design of the database.
- An exception to this would be for the 'dartboard' attribute to be included. This was due to the fact that there was only one pub visited that had a dartboard. Four out of five people that were interviewed in this pub went there so they could play Darts. This represents 80% of people in this pub that go there to play darts, so 'dartboard' is to be included.
- Although factors 'Close to house' and 'Serves a good pint' represent high percentages, they are not quantifiable. 'Close to house' is different for every single person so cannot be included in the database schema. 'Serves a good pint' is opinion based and not something that can be easily included in the database.

3.4 Functional Requirements

"Functional requirements capture the intended behaviour of the system. This behaviour may be expressed as services, tasks or functions the system is required to perform." [12] These point out what the system actually does in terms of its functionality. The functional requirements can be split for the two different types of users, the administrative user and the student user.

3.4.1 Administrative user

To satisfy minimum requirements:

- The system must allow for Pubs to be added, deleted and updated.
- The system must allow for Clubs to be added, deleted and updated.
- The system must allow for Bars to be added, deleted and updated.
- The system must allow for a Location to be added, deleted and updated.
- The system must allow for a Music type to be added, deleted and updated.
- The system must allow for a Club or Bar Promotion to be added, deleted and updated.

To satisfy future enhancements:

- The system could allow for details of Pub quizzes to be added, deleted and updated for any given day of the week.
- The system could allow for Food details to be added, deleted and updated for any given day of the week.

3.4.2 Student user

To satisfy minimum requirements:

- The system must allow for Pubs in the database to be attainable through a web front.
- The system must allow for Clubs in the database to be attainable through a web front.

- The system must allow for Bars in the database to be attainable through a web front.

To satisfy future enhancements:

- The system could allow a Pub to be searched on by its location.
- The system could allow a Pub to be searched on by its Food start time.
- The system could allow a Pub to be searched on by its Food finish time.
- The system could allow a Pub to be searched on by its Food day.
- The system could allow a Pub to be searched on by its Pub quiz night.
- The system could allow a Pub to be searched depending on whether it includes a pool table.
- The system could allow a Pub to be searched depending on whether it includes a quiz machine.
- The system could allow a Pub to be searched depending on whether it includes a dartboard.
- The system could allow a Pub to be searched depending on whether it includes a jukebox.
- The system could allow a Pub to be searched depending on whether it includes a wide screen TV.
- The system could allow the search to be refined by using the above criteria in one single search.
- The system could allow a club or bar to be searched on by its promotion night.
- The system could allow a club or bar to be searched on by its location.
- The system could allow a club or bar to be searched on by its Music Type.
- The system could allow a message to be posted into a forum.
- The system could allow the user to view a map of each social establishment.

3.5 Non-functional Requirements

“The term 'non-functional requirements' has traditionally been used to refer to any requirements which relate to aspects of a system other than its functionality. One quite widely accepted definition of non-functional requirements is:

'A non-functional system requirement is a restriction or constraint placed on a system service.' [14]“ [13]

- The database must provide a user-friendly interface so administrators can easily navigate to their required destination.
- The administrator must update the system every week so it is up-to-date for the commencing week.
- The administrator should supply, where possible, an image of each social establishment.

- The web front should be user-friendly and conform to specific design guidelines. IBM suggests the following design principles for which a good interface will conform to:
    - Simplicity: The interface should be kept simple and straightforward.
    - Support: Users should have control over the system so they are able to perform tasks as they would normally do on a regular basis.
    - Familiarity: All aspects of the interface should be kept the same so users are able to gain knowledge of how the system works.
    - Obviousness: The interface should use real world representations so users become familiar with icons and toolbars etc.
    - Encouragement: A user's actions should result in something that they expect and the interface should provide a way of reversing those actions.
    - Satisfaction: The interface should provide the user with a feel of accomplishment.
    - Availability: The user should be able to use any of the interface's objects at any one time in any sequence.
    - Safety: The user should be protected from making any errors. A good interface will have appropriate error handling.
    - Versatility: The interface should be flexible enough so a wide range of user skills can be accommodated.
    - Personalization: A good interface should consider that all users are different.

An extended set of these design principles can be found in appendix D. The functional and non-functional requirements mentioned above provide a sound basis for an evaluation criterion. This evaluation criterion can be used in conjunction with the evaluation section to justify whether the system does what its supposed to do.

3.6 Business Rules

Business rules provide application constraints that must be conformed to in order to maintain integrity throughout the database. Listed below are selections of business rules that need to be enforced in the database together with the type of business rule that they are.

1. An administrator must enter a password to get into the system – structural
2. When a location is deleted from the database then so should the clubs, bars and pubs that reside in this location – Action Triggering
3. When a Pub is deleted from the database then so should the food details and pub quizzes that belong to it – Action Triggering
4. When a Club or Bar is deleted from the database then so should the promotion details that belong to that establishment – Action triggering

5. If an administrator adds a new promotion for a club or bar, the start date can not be earlier than the current date – Action Restricting
6. If an administrator adds a new promotion for a club or bar, the end date can not be earlier than the start date – Action Restricting
7. If food details are added for a particular pub, then the finish time cannot be earlier than the start time – Action Restricting
8. If quiz details are added for a particular pub, then the finish time cannot be earlier than the start time – Action Restricting

4. DESIGN

4.1 Database Design

The database requires a data-modelling element that consists of a logical design and a physical design. The logical design identifies the entities and their relationships that are required in order to construct an Entity Relationship Diagram. The physical design maps the logical design into physical tables.

4.1.1 Entities

"The basic object that the ER model represents is an entity which is a "thing" in the real world with an independent existence. An entity may be an object with a physical existence-a particular person, car, house, or employee-or it may be an object with a conceptual existence-a company, a job, or a university course." [7]. Entities to be included in the ER Model are Pub, Food_Day, Pub_Quiz, Night, Music, Promotion, Venue, Bar and Club.

4.1.2 Attributes

Each entity contains one or more attributes and it is these attributes that describe the entity. "A particular entity will have a value for each of its attributes. The attribute values that describe each entity become a major part of the data stored in the database." [7]. The entities included will have the following attributes:

Pub: Pub_id, Pub_name, Address, Location, Postcode, Tel_No, Description, Pool_table, Quiz_machine, Dartboard, Jukebox, Widescreen_TV, Image

Food_Day: Pub_id, Night_id, Food_details, Start_time, Finish_time

Quiz_Night: Pub_id, Night_id, Quiz_details, Start_time, Finish_time

Location: Location_id, Location

Night: Night_id

Music: Music_id, Music_type, Description

Venue: Venue_id, Venue_name, Type, Address, Location, Postcode, Tel_no, Capacity, No_of_rooms, Description, Image

Promotions: Promo_id, Promo_name, Start_date, End_date, Dresscode, Entry_Price, Drinks_offers, Venue_id, Night_id, Music_id

4.1.3 Relationships

A relationship can be classified by differing degrees. They can be one-to-one (1:1), one-to-many (1:M) or many-to-many (M:N) relationships [7]. These relationships represent the logical links between the different entities. The relationship between the entity 'Venue' and the entity 'Promotion' is a one-to-many relationship. A venue can have many promotions but a promotion can only have one venue thus classifying the relationship as one-to-many.

Entity 1	Relationship	Entity 2	Classification
Location	has	Pub	(1:M)

A particular Location such as Headingley can have many pubs but a Pub such as Skyrack can only have one Location.

Pub	has	Quiz_night	(1:M)

A pub can have many quiz nights but a particular quiz night can only be at one pub.

Pub	has	Food_Day	(1:M)

A pub can have many food days but a particular food day can only be at one pub.

Night	holds	Food_Day	(1:M)

A day of the week can have many different food days but a particular food day can only have one day of the week.

Night	holds	Quiz_night	(1:M)

A day of the week can have many different quiz nights but a particular quiz night can only have one day of the week.

Night	holds	Promotion	(1:M)

A night of the week such as Monday holds many different promotions but a particular promotion can only be on one night of the week

Venue	has	Promotion	(1:M)

A venue can have many promotions but a particular promotion can only be at one venue.

Music	has	Promotion	(1:M)

A type of music is represented in many promotions but a particular promotion only represents one type of music.

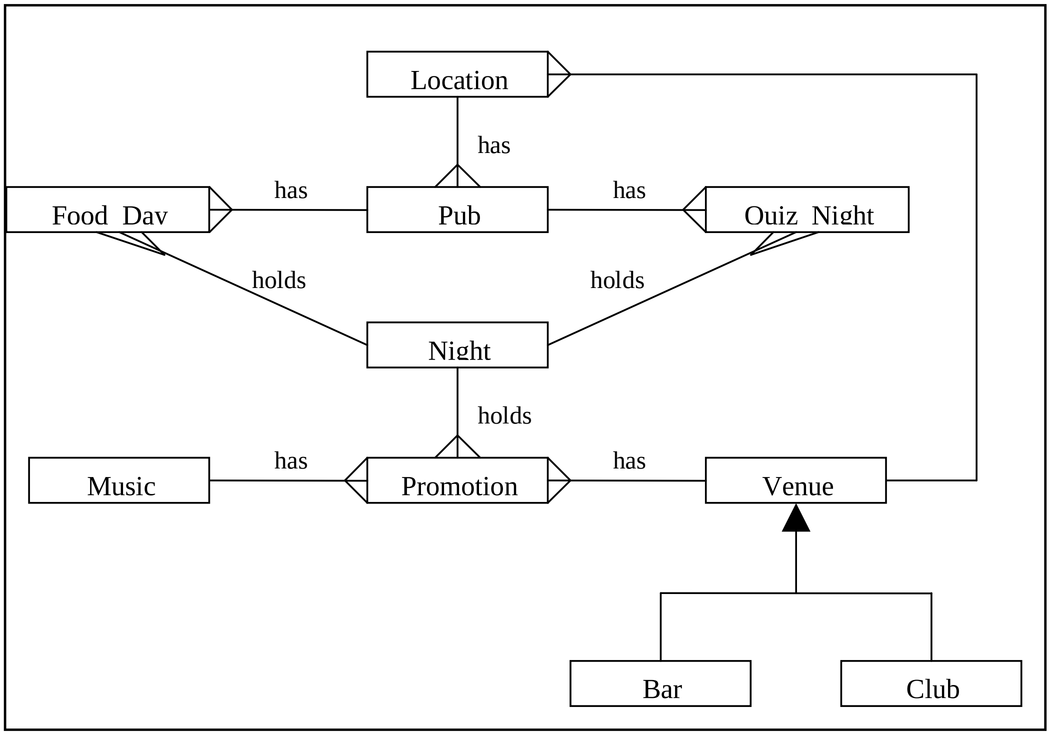


Figure 4.1: Entity Relationship Diagram

The entity relationship diagram shown in Figure 4.1 shows that entities “Bar” and “Club” inherit attributes from the entity “Venue”. The reason why the entity relationship diagram is constructed in this way is so that a Bar or a club can hold a particular promotion. This allows a promotion to change venue from a club to a bar or vice versa using this design. If separate entities “Bar” and “Club” were just used then this would not be possible. The “Bar” and “Club” entities will not be mapped to a table into the database because this would cause unnecessary duplication of data. Instead, they are to be designed as queries that inherit the data dictionary from the “Venue” table and have the “Type” attribute set as a default “Bar” or “Club” accordingly.

4.1.4 Normalisation

“The Normalisation process, as first proposed by Codd, takes a relation schema through a series of tests to “certify” whether it satisfies a certain normal form” [7]. The purpose of undergoing this normalisation process is to reduce data redundancy so that data is not stored unnecessarily. A standard technique that tests for normalisation is the use of functional dependencies that are identified as “the single most important concept in relational schema design” [7].

1st Normal Form

For a relation R to be in first normal form, all attributes must be single atomic values. “It states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute” [7]. In other words 1NF does not allow you to have a tuple of values or any set of values and must be atomic. Consider the relation ‘Pub_quiz’ with the primary key of *Pub_id:*

PUB_ID	**START_TIME**	**FINISH_TIME**	**QUIZ_DETAILS**	**NIGHT_ID**
1	20:00	23:00	Sports Quiz	{Monday, Thursday}
2	19:30	22:30	General Knowledge Quiz	{Tuesday, Wednesday, Sunday}

The relation assumes that one pub can have more than one pub quiz. However this is not in 1st normal form because the attribute ‘Night_id’ is not atomic. To achieve 1st normal form, the key must be extended so there is a separate tuple for each night of a Pub quiz.

PUB_ID	**START_TIME**	**FINISH_TIME**	**QUIZ_DETAILS**	**NIGHT_ID**
1	20:00	23:00	Sports Quiz	Monday
1	20:00	23:00	Sports Quiz	Thursday
2	19:30	22:30	General Knowledge Quiz	Tuesday
2	19:30	22:30	General Knowledge Quiz	Wednesday
2	19:30	22:30	General Knowledge Quiz	Sunday

The relation above now takes the primary key of {PUB_ID, NIGHT_ID} where a particular pub can only have one pub quiz on a particular night of the week.

The disadvantage of this method is the fact that data redundancy is introduced. However this particular method of normalisation must be used to allow the administrator to modify or enter different quiz details for each night. For example, the sports quiz at Pub ‘1’ on a Thursday might be changed so it finishes at 22:30 rather than 23:00.

2nd Normal Form

"Second normal form (2NF) is based on the concept of full functional dependency. A functional dependency X → Y is a full functional dependency if removal of any attribute A from X means that the dependency does not hold any more" [7]. The way 2nd normal form is tested is by testing for functional dependencies where the left hand side is part of the primary key. If the primary key is a single attribute than there is no need to test it. The following functional dependencies exist in the pub quiz relation:

FD1: {PUB_ID, NIGHT_ID} → START_TIME
FD2: {PUB_ID, NIGHT_ID} → FINISH_TIME
FD3: {PUB_ID, NIGHT_ID} → QUIZ_DETAILS

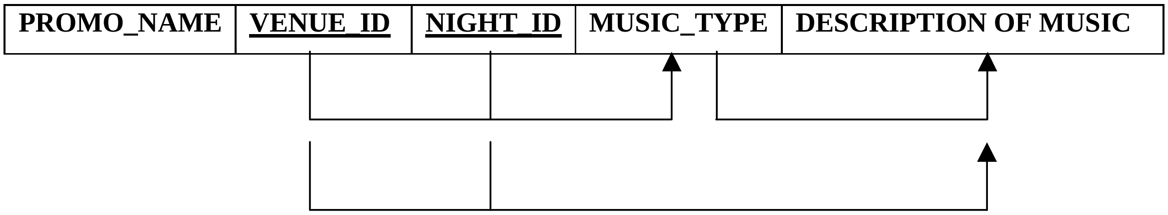
If either attribute of the primary key {PUB_ID, NIGHT_ID} were removed then the functional dependency would no longer hold. For example, if the 'NIGHT_ID' attribute was removed from the left hand side of the functional dependency, then you would be left with the following:

{PUB_ID} → START_TIME

The start time would no longer be dependent on the left hand side of the functional dependency because it can differ for every night of the week. The same can be said if the 'PUB_ID' attribute was removed because the start time would no longer be dependent on just the specific night of the week. It would vary for each different pub. The relations 'Food' and 'Promotions' were also tested for 2nd normal form as they both contained multiple attributes for the primary key.

3rd Normal Form

"Third normal form (3NF) is based on the concept of transitive dependency. A functional dependency X → Y in a relation schema R is a transitive dependency if there is a set of attributes Z that is neither a candidate key nor a subset of any key" [7]. Third normal form implies that any attribute that is a non-key cannot depend on any other non-key attribute in the same relation. Each non-key attribute must be independent of each other.



The diagram shown above represents part of the 'Promotions' relation. The functional dependency of '{VENUE_ID, NIGHT_ID} → DESCRIPTION OF MUSIC' is transitive through the attribute 'MUSIC_TYPE'. This is due to the fact that both functional dependencies '{VENUE_ID, NIGHT_ID} → MUSIC_TYPE' and 'MUSIC_TYPE → DESCRIPTION OF MUSIC' hold. Third normal form states that any non-key attribute cannot depend on any other non-key attribute in the same relation. Due to the functional dependency of ‘MUSIC_TYPE → DESCRIPTION OF MUSIC’, this relation is not in third normal form. The relation can be normalised by decomposing it into two separate relations, ‘Music’ and ‘Promotions’.

MUSIC:

MUSIC_TYPE	**DESCRIPTION OF MUSIC**

‘MUSIC_TYPE’ becomes the primary key of the relation ‘MUSIC’

PROMOTIONS:

PROMO_NAME	**VENUE_ID**	**NIGHT_ID**	**MUSIC_TYPE**

‘MUSIC_TYPE’ becomes the foreign key of the relation ‘PROMOTIONS’ with reference to the primary key ‘MUSIC_TYPE’ in the relation ‘MUSIC’.

4.1.5 Database Schema

“In a database, there is a part that is invariant in time, called the schema of the database made up of the characteristics of the data.” [6] The schema of particular relation includes the name of the relation and its corresponding attributes. The database schema constructed for the proposed database can be seen in Appendix E.

4.1.6 Data Dictionary

“Each relational Database management system manages its own data dictionary (or rather the description of the tables present in the database) using the relational schema” [6]. The data dictionary is required to identify each attributes data types and any other information that is necessary. This can be seen in Appendix F.

4.1.7 Database Population

For testing Purposes, the database is to be populated with eight pubs, eight bars and eight clubs. This will ensure that the functionality of the system is fully tested.

4.1.8 Integrity Constraints

The purpose of this section is to show how entity integrity and referential integrity will be included in the database.

Entity integrity

Entity integrity ensures that no primary key in the database schema can contain a NULL value. A primary key that is a NULL value would violate normalisation constraints.

Referential integrity

Referential integrity states that if there is a foreign key in relation R1 that references a primary key in relation R2 then the value entered in the foreign key attribute must appear in the primary key attribute. Foreign key dependencies can be used so that the first four business rules in section 3.6 are satisfied. When a tuple in the primary table is deleted, referential integrity can be modified so that all related records in the foreign key tables are also deleted. The same procedure can be done when primary tables are updated.

4.1.9 Input Validation

Validation rules apply a constraint on an attribute that restrict users input. They help to prevent unnecessary errors by setting the particular format of an attribute. For example an attribute with the validation rule of '>Date()' requires that the date entered would have to be greater than the current date in order for it to be valid. In Microsoft access, validation can be in the form of validation rules, input masks, format and default values. The following attributes are validated using one of these techniques. They are described below:

- In the tables 'Pub' and 'Venue' the 'Postcode' attribute must be in the form of 'LL09\ 0LL' where an L represents a letter and a 0 or 9 represent a number. The difference between a 0 and a 9 is the fact that the 0 must be supplied. This accommodates postcodes that are either in the format of 'LS6 1DF' or 'LS10 3GH'.
- In the tables 'Pub' and 'Venue' the 'Tel_No' attribute must be in the form of '/(00009") "0000009'. The user must enter a minimum four numbers and a maximum of five numbers for the area code. The user must also enter a minimum of six and a maximum of seven numbers to complete the remainder of the telephone number.
- n the table 'Pub' the attributes 'Pool_table', 'Jukebox', 'Dartboard', 'Quiz_machine', and 'Widescreen TV' must have a value that is either 'yes' or 'no'
- In the tables 'Food' and 'Pub_quiz' the attributes 'Finish_time' and 'End_time' must be greater than the attribute 'Start_time'. To implement this would require the use of a validation rule that would be in the form of '>Start_time'. Implementing this satisfies business rules (7) and (8).
- In the 'Promotions' table, the attribute 'Start_date' would have to be equal or greater than the current date. The attribute must have a validation rule in the form of '>=Date()'. Implementing this satisfies business rule (5).

- In the 'Promotions' table, the attribute 'End_date' would have to be equal or greater than the Start date. The attribute must have a validation rule in the form of '>=Start_date'. Implementing this satisfies business rule (6).

4.2 User interface design

Figure 4.2 shows the design of the user interface that the administrator will use to satisfy the minimum requirement of 'Configure the database to provide a user-friendly interface that allows an administrator to add/delete/edit the information held.'

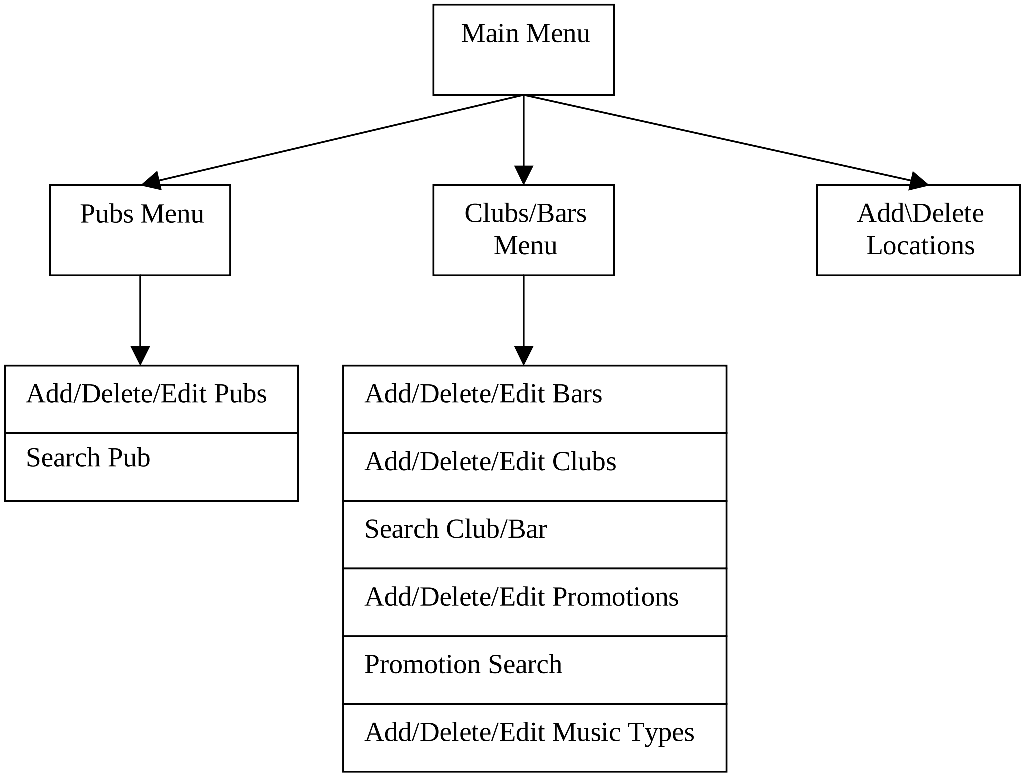


Figure 4.2: Navigation of User interface

Each individual box represents a form in the database. Each of these forms will have the ability to navigate back to their previous menu and the main menu. Figure 4.3 shows how the bottom part of the form will look.

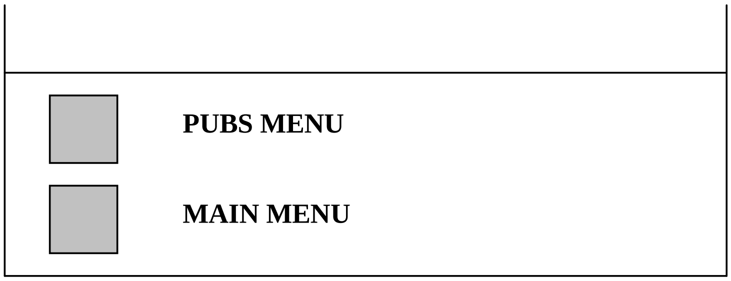


Figure 4.3: Form design

Each form will contain a combo box that is represented by a drop down menu for every foreign key dependency represented in the database. Due to the fact that referential integrity is enforced for every foreign key in the database, means that the user is limited to what they can select in this field. By providing the user with a drop down menu that looks up values from the primary table, the chance of error is greatly reduced. If the user were required to enter this value in a textbox, then they would be unsure of what values are present in the primary table. Rather than have the ID value displayed in this combo box, the user will be given a list of names (e.g. Venue names) so they can easily identify the value they wish to select.

4.3 Web interface design

The interface will consist of thirteen pages, that if implemented, will satisfy future enhancements 2,3 and 4. Only the home page, the links page and the help page will purely be in html because they are the only ones that don't need to be created dynamically. The remaining ten pages will be active server pages that are dynamically created everytime that the page is requested.

Navigation

To enable the student to navigate the web system, frames pages are to be used that will always be visible throughout the whole interface. Frames conform to the IBM guidline of familiarity where the user is presented with a familiar outlook on each interface. The interface will consist of two frames, a left sided frame and a top frame. The left-sided frame will resemble a menu that the user will use to navigate through the system. The top frame will consist of a logo and a title for the interface. The layout of the interface can be seen in figure 4.4.

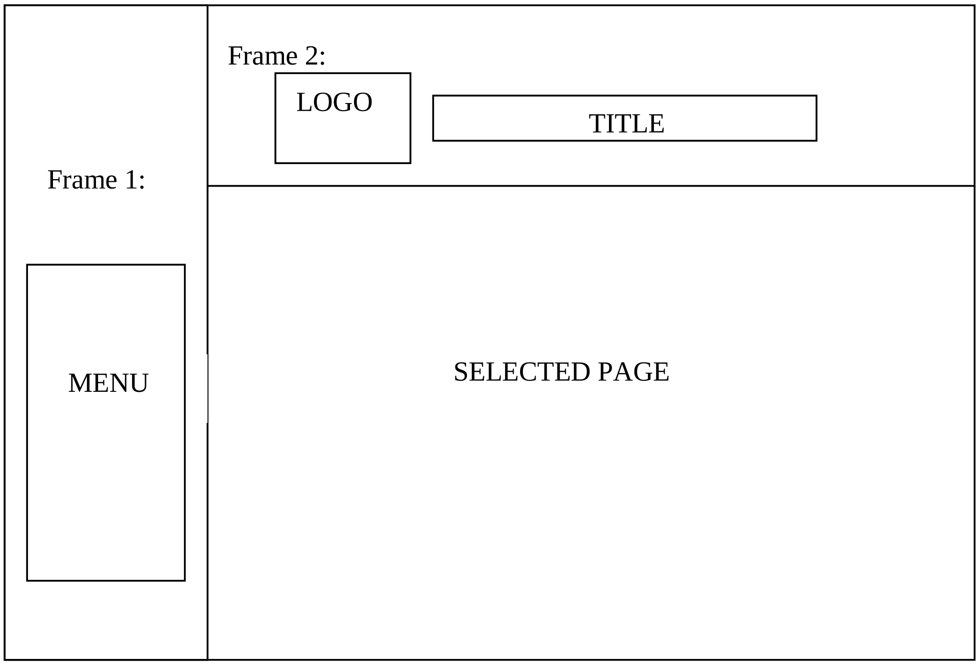


Figure 4.4: Web Interface design

When a page is requested from the menu, it is displayed as the ‘selected page’. A selection of six pages are available from the menu where further active server pages are generated as a result of a user process in some of these pages. The diagram (figure 4.5) illustrates the structure of the web interface.

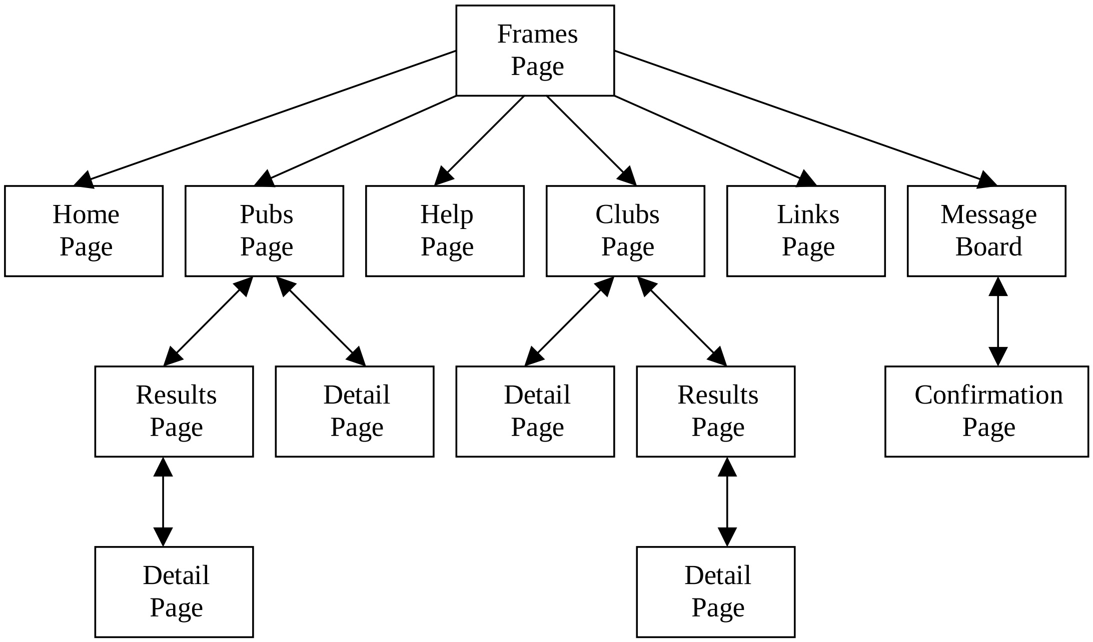


Figure 4.5: Navigation of Web interface

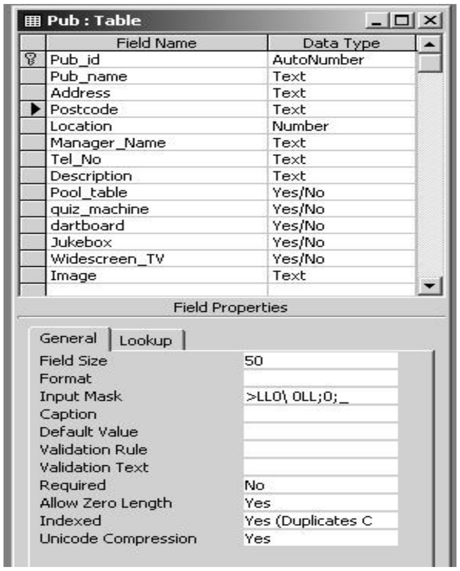
The arrows between the pages represent the hyperlinks that exist through the system. A double-ended arrow represents both pages providing hyperlinks to each other.

5. IMPLEMENTATION

5.1 Database Implementation

Implementation of the database was based on the normalised database schema constructed in the design phase. The methodology selected in section 1.3.5, allowed for the creation of modules where each module was designed, implemented and tested independently of the other modules. Such an example would be where tables were designed, implemented and tested before the design stage of the forms was started. The methodology also allowed for previous stages to be re-visited should an element of a module need re-designing.

5.1.1 Table implementation



Field Name	Data Type
Pub_id	AutoNumber
Pub_name	Text
Address	Text
Postcode	Text
Location	Number
Manager_Name	Text
Tel_No	Text
Description	Text
Pool_table	Yes/No
quiz_machine	Yes/No
dartboard	Yes/No
Jukebox	Yes/No
Widescreen_TV	Yes/No
Image	Text

Field Size	50
Format	
Input Mask	>LL0\ 0LL;0;_
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	Yes (Duplicates C
Unicode Compression	Yes

Figure 5.1: 'Pub' Table

Each entity from the ER Diagram was constructed in Microsoft Access as a 'Table'. Each of these tables were created in the design view option rather than using the table wizard because it allows the designer to have more control over each of the attribute properties. For example, design view allows you to specify the default value for each attribute. Such a property may come in use if you needed the value stored in the attribute to be the current date. Figure 5.1 shows how the Pub table was created using the design view option.

The key symbol next to the Pub_id attribute represents the primary key field. This field is unique in a way that the values cannot be duplicated. For example, only one particular pub can have a pub_id value of 3. The data type for this field is set to autonumber, which automatically increments the pub_id value by 1 for every new record entered into the table. Attributes such as "Postcode" and "Tel_No" were given an input mask to make data entry easier and to control the values users can enter in the field. Objects in a pub such as a pool table or a jukebox were given a yes/no data type where the value can be set to either yes or no depending on whether the pub has this particular object. The "Image" attribute has a simple text data type where the administrator simply enters the filename of the image should they have one (eg. Skyrack.jpg). This is so the image can be displayed by the asp page.

5.1.2 Relationship Diagram

Figure 5.2 shows the relationships between the different tables in the database. It can be seen that each table is connected through the use of primary and foreign keys.

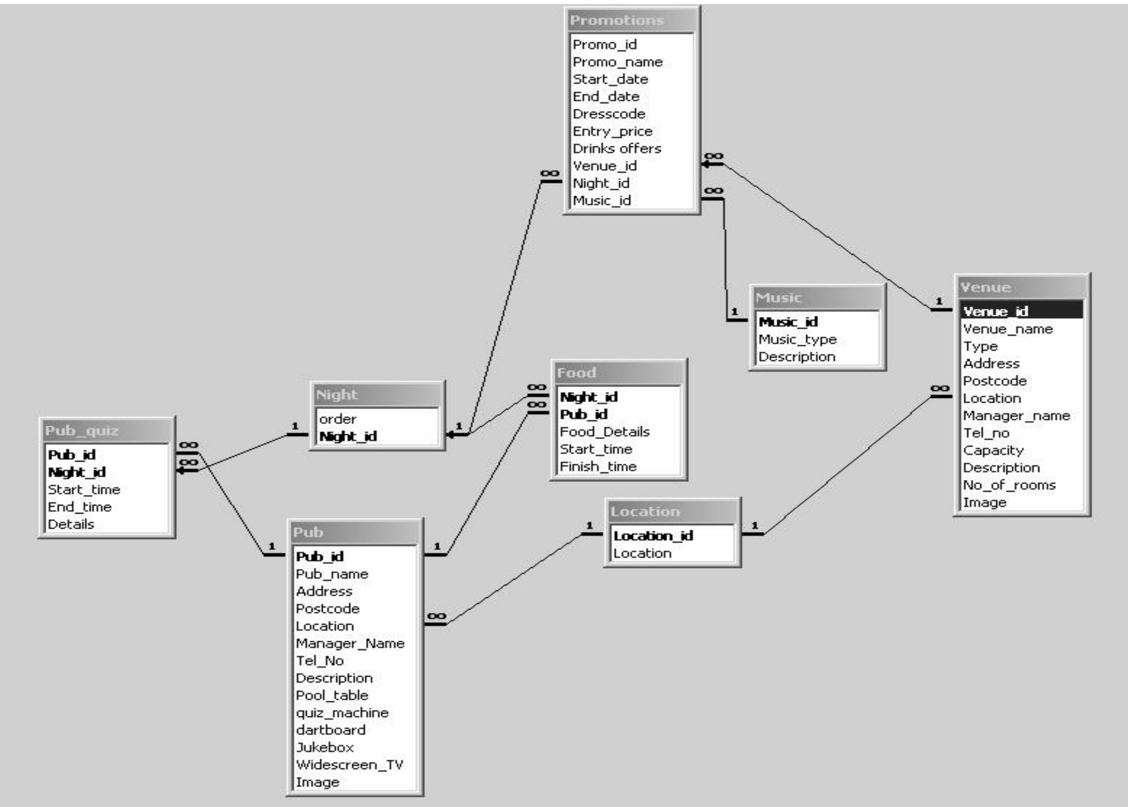
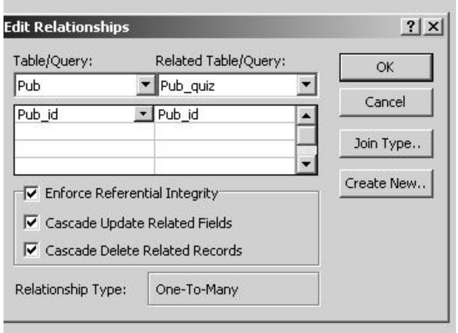


Figure 5.2: Relationship Diagram

To ensure that integrity is maintained throughout the database, Access allows you to select the “Enforce Referential Integrity” option. This ensures that users cannot add records to a relation R1 where there is no reference to a relation R2 if they are connected by some relationship.



Table/Query: Pub
Related Table/Query: Pub_quiz

Pub_id	Pub_id

☑ Enforce Referential Integrity
☑ Cascade Update Related Fields
☑ Cascade Delete Related Records

Relationship Type: One-To-Many

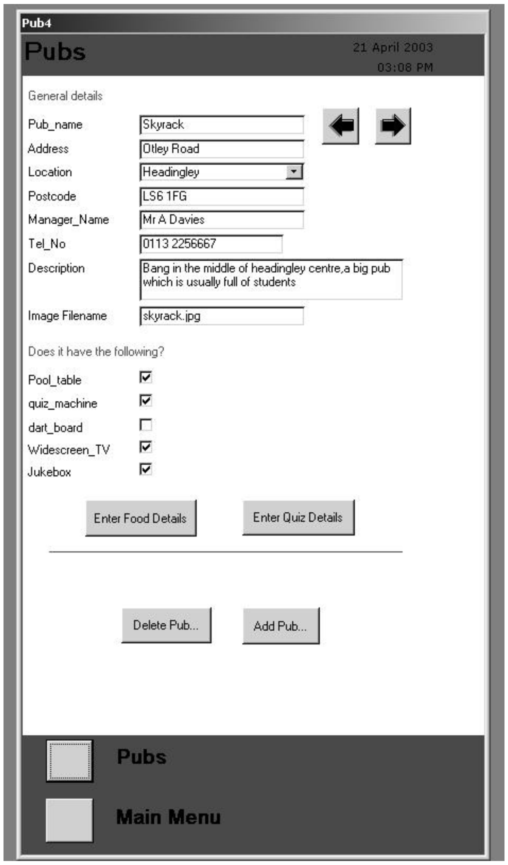
Figure 5.3: Defining relationships

Figure 5.3 shows how these relationships are defined. Another useful feature offered by Access is the “Cascade Delete Related Records” option. This will delete any records in the foreign key table that reference the record in the primary table that is being deleted.

The example shown in Figure 5.3 shows the relationship between the "Pub" table and the "Pub_quiz" table. If a Pub P is deleted from the relation "Pub" then the effect of referential integrity will be to delete all pub quizzes from the relation "Pub_quiz" that reference Pub P.

5.1.3 Form Implementation

The user interface for the administrator of the database was created through the use of forms. Forms provide an abstract method of data entry, data deletion and editing of data in a database. A total of 26 forms were created, three of which were menu forms and eight of which were subforms.



Pub4

Pubs

21 April 2003
03:08 PM

General details

Pub_name: Skyrack
Address: Otley Road
Location: Headingley
Postcode: LS6 1FG
Manager_Name: Mr A Davies
Tel_No: 0113 2256667
Description: Bang in the middle of headingley centre,a big pub which is usually full of students
Image Filename: skyrack.jpg

Does it have the following?

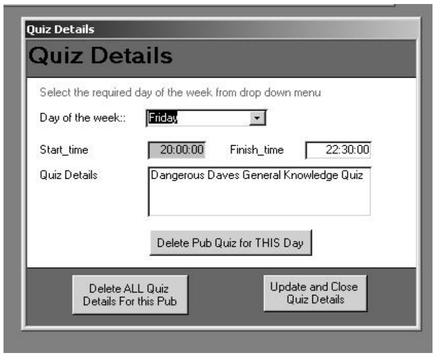
Pool_table ☑
quiz_machine ☑
dart_board ☐
Widescreen_TV ☑
Jukebox ☑

Figure 5.4: 'Pubs' Form

Figure 5.4 shows the layout of the pubs form.

Microsoft Access has the facility of a form wizard where you are prompted for the required table(s) and field(s) that are to appear in the form. After creating the form, it can be edited in the design view to get an appropriate format. In each case I decided against displaying the "ID" field in each form due to the fact that it would be unnecessary because this field generates its own value with an autonumber data type. Each form has the ability to return to previous menus by selecting the required navigational command button at the bottom of the form. The form also allows you to enter/view food and quiz details for a particular pub. The way this works is by running an event procedure in Visual Basic when the button is clicked

The visual Basic code filters the records in the Pub_quiz table to just those with the Pub_id that matches the current record on the form. After the command button is clicked the Pub_quiz form is displayed as shown in figure 5.5.


Quiz Details

Quiz Details

Select the required day of the week from drop down menu

Day of the week:: Friday

Start_time 20:00:00 Finish_time 22:30:00

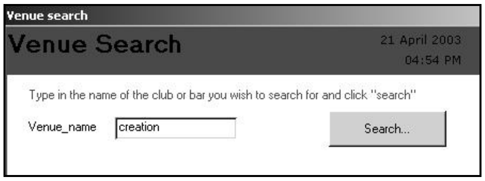
Quiz Details Dangerous Daves General Knowledge Quiz

Delete Pub Quiz for THIS Day
Delete ALL Quiz Details For this Pub
Update and Close Quiz Details

Figure 5.5: 'Quiz Details' Form

The 'Day of the week' is an unbound combo box that retrieves its option values from the "Night" table. The administrator can simply select from the drop down menu the required day of the week to enter or view the quiz details. The code that filters the quiz details for each different pub can be shown in Appendix G. The code to link the Food Details is constructed the same.

After conducting several informal interviews with students in the school of computing, it was decided that there was a need for search forms so that pubs, clubs, bars and promotions could easily be searched on. The administrator simply enters into the text box the name they wish to search for.


Venue search

Venue Search

21 April 2003
04:54 PM

Type in the name of the club or bar you wish to search for and click "search"

Venue_name creation

Figure 5.7: 'Venue_search' Form

The result this has, is to produce a subform in the main form itself of the details that relate directly to the search criteria. This subform is based on the following query:

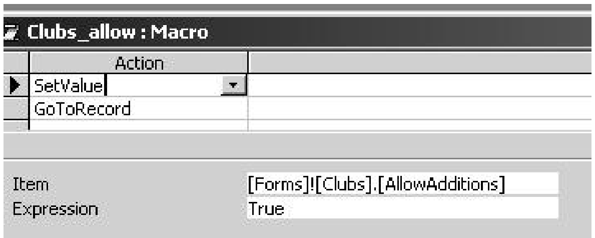
```
SELECT * FROM Venue WHERE Venue_name LIKE [forms].[venuesearch].[venue_name]
```

Each club and bar form allows you to view or enter promotions for any particular night of the week. Using visual basic code that is similar to that shown in Appendix G, the promotions are filtered from the "promotions" table by the venue_id. The venue_id filtered is equal to that in the current club or bar form. The administrator is able to select from a drop down menu a night of the week that they wish to view or enter details on. If the night of the week selected already has a promotion for that particular venue then the promotion details will be displayed in the subform below the 'Night of the week' drop down menu. If there is no promotion on that particular night then the promotion subform will have blank fields that the administrator may wish to fill in if they want to enter a promotion for that night of the week. A particular promotion may change from one venue to another so a form based on the promotions table was created where the administrator can select a new venue for the promotion. The venue is selected from a combo box. This combo box is dynamic in the way that only the venues entered in the "venue" table will be available for selection. Other forms in the database allow for locations and types of music to be added, deleted and modified. All the forms implemented in the database can be seen in Appendix H.

5.1.4 Macro Implementation

"A macro is a set of one or more actions that each perform a particular operation, such as opening a form or printing a report. Macros can help you to automate common tasks. For example, you can run a macro that prints a report when a user clicks a command button" [15].

Macros have had a significant use in this project, performing a numerous amount of tasks from opening a form to running a piece of SQL. There are 47 different macros that are functional in the database, although a lot of them perform a similar task. A common macro that was present in the database was to allow a user to add a record in the database. By default, the "allow additions" function in each form property is set to "no" to prevent users from accidentally adding records when they skip through each record in the form. A macro was created that changed this "allow additions" property to "yes" in the current form. This macro was applied to a command button that it would be run when the button is clicked. The macro design can be shown in figure 5.8.



Action
SetValue
GoToRecord

Item [Forms]![Clubs].[AllowAdditions]
Expression True

Figure 5.8: 'Clubs_allow' Macro design

The other macro that was commonly present in the database had the functionality of closing the current form and opening another form. This gave a user-friendly approach where only one form was displayed at any one time. For example, if a user was currently on the 'Main menu' form but wished to navigate to the 'Pubs' form then the macro would perform the action of closing the 'Main menu' form and opening the 'Pubs' form.

5.2 Implementation of server side scripting

Having identified that ASP was the most appropriate tool to write the scripting code in, the next step required learning the technology, understanding it and finally writing it.

A connection between the web application and the database was defined by creating a connection string shown in figure 5.9.

```
Holds the Database Connection Object
Dim adoConn
Creates an ADO connection object
Set adoConn = Server.CreateObject("ADODB.Connection")
Sets an active connection to the Connection object using a DSN-less connection
adoConn.Open
"DRIVER={Microsoft Access Driver (*.mdb)}; DBQ=" & Server.MapPath("proj.mdb")
```

Figure 5.9: Database connection string

The website consisted of six pages which could be accessed from the navigational menu in the left sided frame. These interfaces were the Home page, Pubs page, Clubs and Bars page, Message Board forum, Help page and the Links page. The screenshots of these pages are all show in Appendix I.

5.2.1 Home Page

The home page gives a brief description of what the site is all about and what it has to offer to the user. The home page also provides navigational links to the other five interfaces mentioned above.

5.2.2 Pubs Page

This page allowed the user to select a pub from a drop down menu or to enter some search criteria in order to retrieve a set of results that matched their criteria.

Selecting a Pub

The user is presented with a drop down menu that is dynamically created so that each addition/deletion or edit to a pub results in the list being automatically changed. This becomes particular useful if a student has a particular pub in mind which they want to look at without having to go through the process of specifying search criteria.

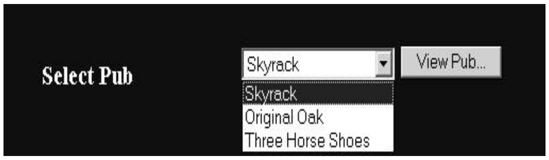


Figure 5.10: Selecting a Pub

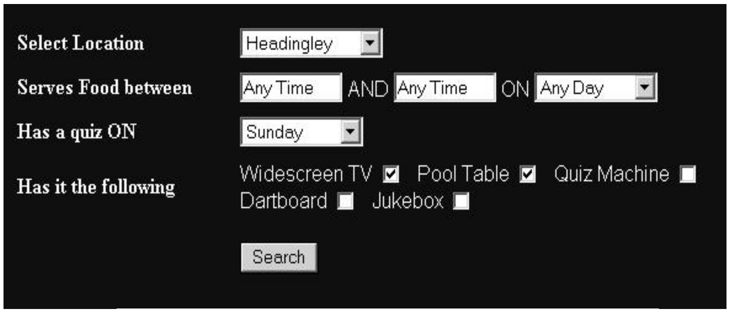
By clicking the "view pub" submit button, the pub detail page opens up which gives more information on that particular pub. This is shown in figure 5.16

The way this works is for the pub detail page to create a recordset that selects everything from the table "Pub" where the 'Pub_id' is the same as the 'Pub_id' selected in the drop down menu. Running the following query within it creates the recordset:

```
SELECT * FROM Pub WHERE Pub_id " & _
" LIKE '%" & request.form("selPub") & _
"%'
```

Defining search criteria

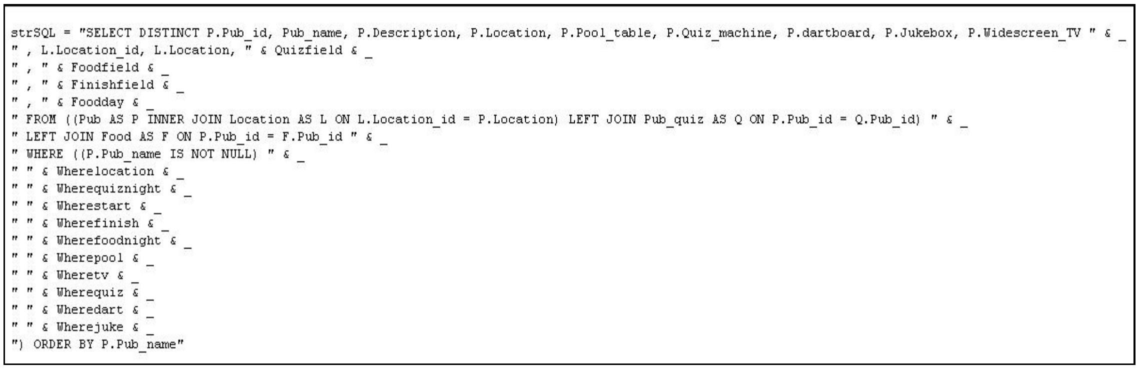
If a user is looking for a pub with particular attributes then they can enter their relevant criteria from the search options. The search criteria shown in figure 5.11 would select all pubs in headingley that have a pub quiz on a Sunday that also have a widescreen TV and a pool table. The search page has been constructed so that the user is able to select as many or as little search criteria as they wish to select.



Select Location: Headingley
Serves Food between: Any Time AND Any Time ON Any Day
Has a quiz ON: Sunday
Has it the following: Widescreen TV ☑ Pool Table ☑ Quiz Machine □ Dartboard □ Jukebox □

Figure 5.11: 'Pubs' Search page

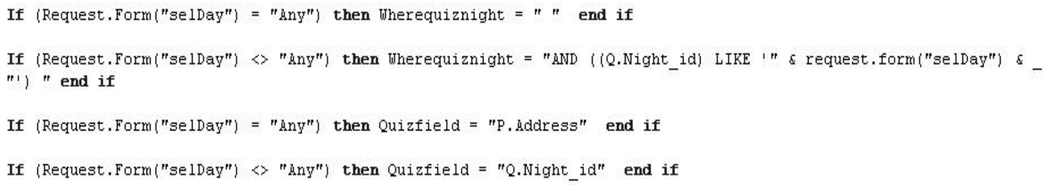
If a particular criteria is not selected (ie left as "Any…") then they are simply left out of the WHERE clause. The following query in figure 5.12 is executed once the search button is clicked:



```
strSQL = "SELECT DISTINCT P.Pub_id, Pub_name, P.Description, P.Location, P.Pool_table, P.Quiz_machine, P.dartboard, P.Jukebox, P.Widescreen_TV " & _
" , L.Location_id, L.Location, " & Quizfield & _
" , " & Foodfield & _
" , " & Finishfield & _
" , " & Foodday & _
" FROM ((Pub AS P INNER JOIN Location AS L ON L.Location_id = P.Location) LEFT JOIN Pub_quiz AS Q ON P.Pub_id = Q.Pub_id) " & _
" LEFT JOIN Food AS F ON P.Pub_id = F.Pub_id " & _
" WHERE ((P.Pub_name IS NOT NULL) " & _
" " & Wherelocation & _
" " & Wherequiznight & _
" " & Wherestart & _
" " & Wherefinish & _
" " & Wherefoodnight & _
" " & Wherepool & _
" " & Wheretv & _
" " & Wherequiz & _
" " & Wheredart & _
" " & Wherejuke & _
") ORDER BY P.Pub_name"
```

Figure 5.12: SQL query string

Figure 5.13 shows an example of an IF statement that is used to construct the query string in figure 5.12. This basically says that if the quiz night is left as "Any" then the 'where clause' for the quiz night is set to empty. However if any other value than "Any" is selected (i.e. Sunday), then the 'where clause' will be set to the value that you select in the drop down menu. The same kind of IF statement in figure 5.13 is used for the entire search criteria, which in turn constructs the query string.

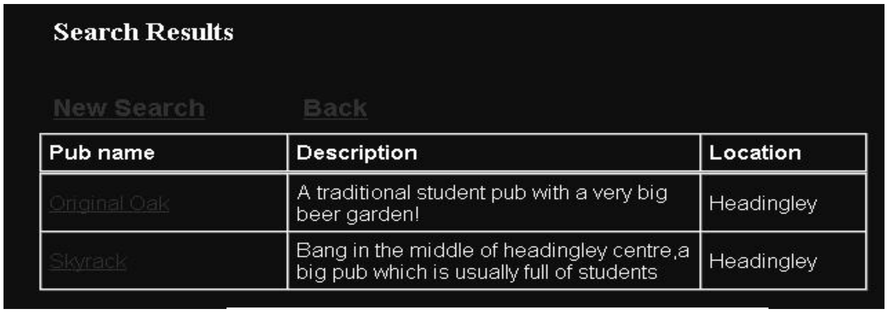


```
If (Request.Form("selDay") = "Any") then Wherequiznight = " "  end if

If (Request.Form("selDay") <> "Any") then Wherequiznight = "AND ((Q.Night_id) LIKE '" & request.form("selDay") & _
"') " end if

If (Request.Form("selDay") = "Any") then Quizfield = "P.Address"  end if

If (Request.Form("selDay") <> "Any") then Quizfield = "Q.Night_id"  end if
```

Figure 5.13: An IF statement that selects the Pub quiz night

5.2.3 Pubs results page

Once the search criteria have been selected, the user will click the search button that will post the criteria to an SQL query shown in figure 5.12. The user is presented with a table of results that match their search criteria, shown in Figure 5.14.



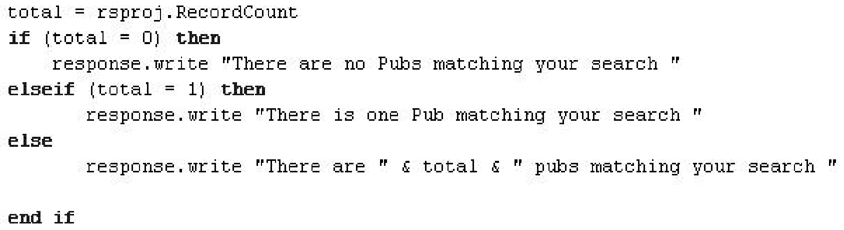
Search Results

New Search   Back

Pub name	Description	Location
Original Oak	A traditional student pub with a very big beer garden!	Headingley
Skyrack	Bang in the middle of headingley centre,a big pub which is usually full of students	Headingley

Figure 5.14: Pub Search results

The results page also includes a piece of code that counts the number of search results found. This code is shown in figure 5.15.



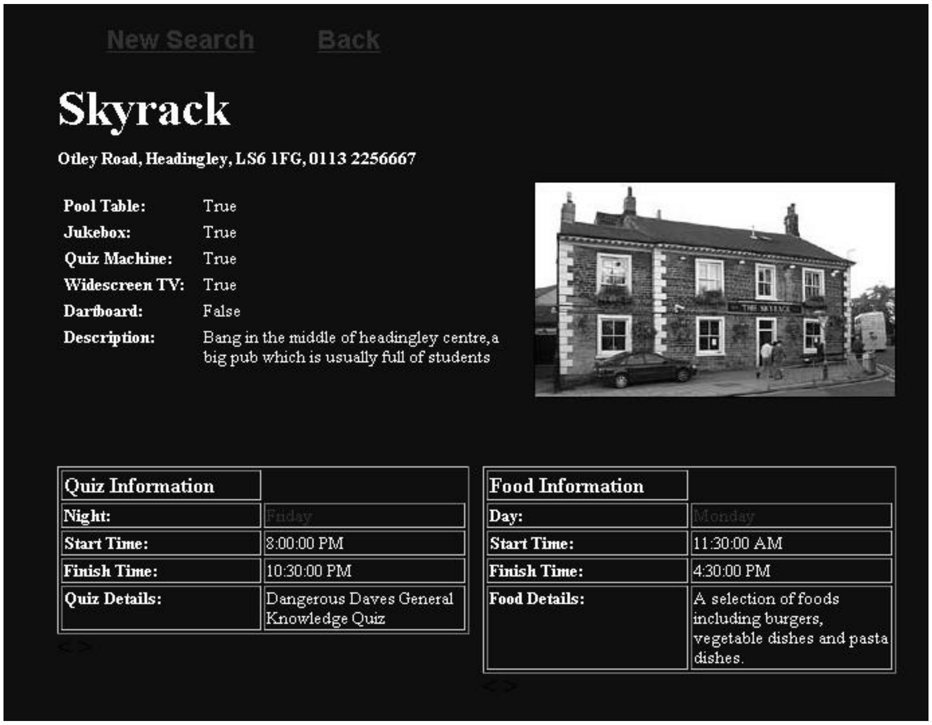
```
total = rsproj.RecordCount
if (total = 0) then
    response.write "There are no Pubs matching your search "
elseif (total = 1) then
       response.write "There is one Pub matching your search "
else
       response.write "There are " & total & " pubs matching your search "

end if
```

Figure 5.15: ASP code to count the number of records in the recordset.

The variable 'total' is used to count the number of pubs that are in the recordset. For grammatical purposes a different line is output if there are no search results, one search result or more than one search result.

When a recordset of results is displayed on the screen, the pub name attribute in the results table becomes a hyperlink to the pub detail page. To view the pub in more detail, the user can simply click on the name of the pub that will take them to the pub detail page. The way this works is the browser forwards the 'Pub_id' value as a URL parameter to the Pub detail page. The pub detail page then selects all the details of the pub where the 'Pub_id' value is equal to that of what has been sent by the Pub results page.

5.2.4 The Pub Detail Page



Skyrack

Otley Road, Headingley, LS6 1FG, 0113 2256667

Pool Table:	True
Jukebox:	True
Quiz Machine:	True
Widescreen TV:	True
Dartboard:	False
Description:	Bang in the middle of headingley centre, a big pub which is usually full of students

Quiz Information	
Night:	Friday
Start Time:	8:00:00 PM
Finish Time:	10:30:00 PM
Quiz Details:	Dangerous Daves General Knowledge Quiz

Food Information	
Day:	Monday
Start Time:	11:30:00 AM
Finish Time:	4:30:00 PM
Food Details:	A selection of foods including burgers, vegetable dishes and pasta dishes.

Figure 5.16: The Pub detail page

The detail page provides the user with more information on their requested pub. Referencing the "image" attribute in the "Pub" table provides the image. It can be presented using the following code:

```
<img src=<%=(rsPub.Fields.Item("image").Value)%>>
```

Each record from the "Pub_quiz" table and "Food" table is displayed where the 'Pub_id' value in these tables' match the 'Pub_id' sent as the URL parameter from the pub results page. These records are looped till the end of the recordset and displayed under each other for the different days of the week. When the case arrives that a pub has no quiz nights for example, then the table will not be displayed at all. Should the user wish to look at another one of their search results they can simply click "Back" which will navigate them to the results table. This becomes quite useful because it eliminates the need to repeat the search over and over.

5.2.5 Clubs and Bars Page

This page was constructed in a very similar format to the Pub page with a little variation on the search criteria. The user is required to select a night of the week from the drop down menu as they are searching for clubs or bars with a promotion on that particular day. A user can narrow down their search results by selecting other criteria from the drop down menus. They have the choice of selecting whether they only want to select only clubs or bars in their search, a specific type of music and a specific location. If any of these are left as "Any.." then the attribute is not considered in the where clause. Like the Pubs page, a query string is constructed depending on what values are selected in the search page and consequentially stored in a recordset. The results from this recordset are output in the table format similar to the results shown in figure 5.14. The screenshots from the clubs and bar pages can be seen in Appendix I.

5.2.6 Message Board Forum

To create this message board required inputting data into a table. The implications of this meant a slight alteration to my database design. A new database table was added with the following schema:

Message_Board (<u>Message_id</u>, Name, Message, Date)

The date attribute required the current date so rather than have to enter it in manually, a function can be used so the date is entered in automatically. By setting the default value in this attribute to '=Date()' has the effect of the current date always being displayed.

A simple interface was then created so that the user can enter their name and the message that they wish to post (figure 5.17). This was included on the same page as the message board itself.

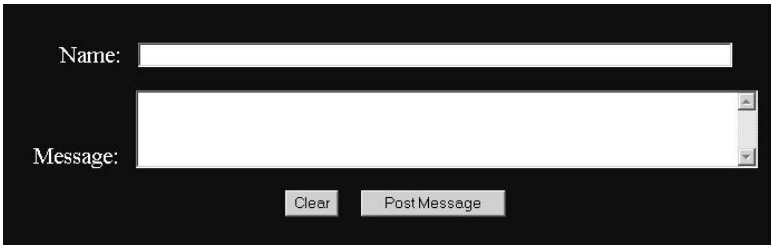
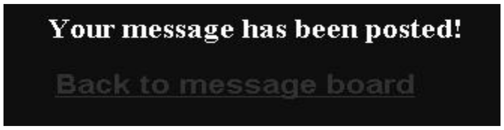


Figure 5.17: Message Board input

The following SQL was then embedded within the message board asp page so that the data posted was input into the "message_board" table in the database.

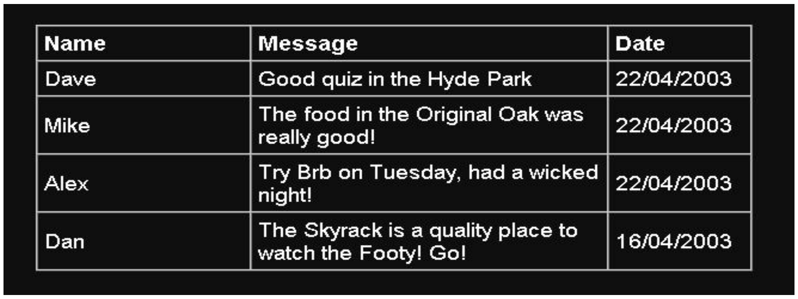
```
 strSQL = " INSERT INTO message_board (Name, Message)" &_
" Values ( '"& Request.Form("T1") &"', '"& Request.Form("T2")&"') "
```

Once the user had clicked ‘Post message’ they received acknowledgement that their message had been posted. The asp page generated also provides a link back to the message board so that the user can see their posted message displayed on the screen. The results from the “message_board” database table are displayed ordered by the attribute ‘Date’. The reason for this was the fact that the most recent message post would be displayed at the top (figure 5.18).



Your message has been posted!

Back to message board



Name	Message	Date
Dave	Good quiz in the Hyde Park	22/04/2003
Mike	The food in the Original Oak was really good!	22/04/2003
Alex	Try Brb on Tuesday, had a wicked night!	22/04/2003
Dan	The Skyrack is a quality place to watch the Footy! Go!	16/04/2003

Figure 5.18: The message board

5.2.7 Help Page

The help page provides users with information on how to use the information system effectively. The important thing to consider is the fact that all people have different levels of computer literacy so you cannot assume that they will just be able to use the system instantaneously. The help page has been provided generally for those who are not familiar with these types of systems. The page has been constructed purely in html and provides examples of how to use the system as well as useful information. This can be seen in appendix I.

5.2.8 Links Page

This page was constructed due to the possible usage of the system for the university open day. The page provides useful links to other university websites that may be appropriate to the student user.

6. TESTING

There are two main reasons for testing. The first is to ensure that each system function works without errors and the second is to test whether the users accept the new system as an alternative to a current system. These two reasons can be categorised into two different types of testing, unit testing and acceptance testing.

6.1 Unit Testing

Unit testing identifies any errors that may be present in the functionality of the system. This became very important because any errors that were found needed to be identified quickly, in order to be modified. The methodology mentioned in section 1.3.5 allowed the system to be split into modules that could each be designed, implemented and tested individually. Unit testing fitted in well with this methodology because errors could be quickly spotted and changed accordingly. It would be foolish to implement the whole of the system without any testing in between because you may find that the errors reside at the very base of the system. An implication of a bad database design could lead to further complications for the implementation of the project. This is why it is so important to make sure each module is thoroughly tested before the next one is undertaken.

Unit testing identifies how robust the system is and how well it handles the errors that may be present in the system. Errors in systems do and will always occur so it is important that you know how to deal with them once they arise. One way to identify any errors that may exist is to draw up a detailed test plan that tests each functionality of the system.

6.1.1 Testing the database functions

The Database has been tested using the integrity constraints and validation rules in sections 4.1.8 and 4.1.9. The first thing to be tested is the login to the system. Any user that does not have a password should not be able to use the system. The function was tested to find, as expected, that users could only enter the system if they had a password.

Integrity can be tested to ensure that both referential integrity and entity integrity are satisfied. Table 6.1 outlines some of the referential integrity tests that were carried out.

ACTION	EXPECTED RESULT	ACTUAL RESULT
Delete Location “Headingley”	“Headingley” deleted from “Location” table and any pubs, clubs and bars that are in this location consequentially deleted.	Same as expected
Delete Venue “Creation”	“Creation” deleted from “Venue” table and any Promotions that are run for this	Same as expected

	venue consequentially deleted from the "Promotions" table.	
Delete Pub "Skyrack"	"Skyrack" deleted from "Pub" table and any Pub quiz or food details that are run for this pub are consequentially deleted from their respected tables.	Same as expected

Table 6.1: Referential Integrity Testing

Each database table was tested for entity integrity by leaving the primary key attribute(s) NULL. The system provides a relevant error message explaining to the user that this attribute cannot contain a NULL value. This need not be tested for any attribute that has the data type of 'autonumber' due to the fact that on record entry this will automatically be generated.

6.1.2 Validation testing

The other main area for testing the database is the validation rules. These are tested to make sure that user inputs the right data into the field. The idea of validation testing is not to ensure that every field entry is accurate, but to reduce the possibility of errors by limiting what the users can input. Table 6.2 shows the tests that were run for the postcode field. Each other validation rule was also tested in the same way. These can be seen in Appendix J.

ACTION	EXPECTED RESULT	ACTUAL RESULT
Enter Postcode LS6 1DF	Action ok	Same as expected
Enter Postcode LS16 1DF	Action ok	Same as expected
Enter Postcode LSJ 1DF	User prevented from entering any postcode not in the 'LL09 0LL' format.	Same as expected
Enter Postcode LS166 1DF	User prevented from entering any postcode not in the 'LL09 0LL' format.	Same as expected
Enter Postcode LS6	Error message displayed	Same as expected

Table 6.2: Validation Testing

6.1.3 Testing SQL queries in ASP code

To consider how robust the system is, each Active Server Page (ASP) must be tested. The pub search asp page includes two drop down menus that are dynamically created from the database. To test these effectively would require an addition/deletion and an edit to the database to see whether the asp page can dynamically output these changes. Table 6.3 shows the test results if a new pub is added/deleted or edited in the database. Table 6.4 shows the test results if a new location is added/deleted or edited in the database.

ACTION	EXPECTED RESULT	ACTUAL RESULT
Add Pub “Test”	Pub “Test” available in the drop down menu	Same as expected
Edit Pub “Test” to “Test1”	Pub “Test1” available in the drop down menu Pub “Test” removed from drop down menu	Same as expected
Delete Pub “Test1”	Pub “Test1” removed from drop down menu	Same as expected

Table 6.3: Testing Asp for add/deletion/edit of a Pub

ACTION	EXPECTED RESULT	ACTUAL RESULT
Add Location “Test”	Location “Test” available in the drop down menu	Same as expected
Edit Location “Test” to “Test1”	Location “Test1” available in the drop down menu Location “Test” removed from drop down menu	Same as expected
Delete Location “Test1”	Location “Test1” removed from drop down menu	Same as expected

Table 6.4: Testing Asp for add/deletion/edit of a Location

To test all of the pub search functions, each possible selection from the user interface (shown in figure 5.11) was tested to make sure that the correct results were displayed to the user. These selections were not only tested on their own, but also with multiple selections. The results should be displayed in correspondence with the criterion that the user selects. Table 6.5 shows the structure of the test plan that is to be used for this particular test. A full testing plan for testing the ASP query code can be found in Appendix K. This appendix also covers test plans for the Club/Bar Search page.

ATTRIBUTE:	SELECTION
Location:	Headingley
Food Start time:	Any Time
Food Finish time:	Any Time

Food Day:	Any Day
Quiz Night:	Any Day
Widescreen TV:	(not selected)
Jukebox:	(not selected)
Pool table	(not selected)
Dartboard	(not selected)
Quiz machine	(not selected)
SQL Query	SELECT DISTINCT P.Pub_id, Pub_name, P.Description, P.Location, P.Pool_table, P.Quiz_machine, P.dartboard, P.Jukebox, P.Widescreen_TV, L.Location_id, L.Location , P.Address , P.Tel_No , P.Postcode , P.Image FROM ((Pub AS P INNER JOIN Location AS L ON L.Location_id = P.Location) LEFT JOIN Pub_quiz AS Q ON P.Pub_id = Q.Pub_id) LEFT JOIN Food AS F ON P.Pub_id = F.Pub_id WHERE ((P.Pub_name IS NOT NULL) AND ((L.Location_id) LIKE '1')) ORDER BY P.Pub_name
Expected result	There are 8 Pubs matching your search
Actual result	Same as expected

Table 6.5: Example Test Plan for search pages

The final phase of unit testing involves testing the use of the message board. The message board interface includes two text boxes for which the user enters both their name and their message. Initial faults were found with this message board in the situation where either or both of the text boxes were left empty. The methodology I had selected in section 1.3.5 allowed me to go to the previous stage of implementation and address this error. To fix it required client side scripting that was run on the clients machine if such an error was to occur. This was implemented using Visual Basic script that had the functionality of displaying a dialogue message if the user left any of the text fields blank.

6.1.4 Results of Unit testing

Each database test that was carried out resulted in what had been expected. This included any integrity tests and validation tests carried out on the database. Once this particular unit was tested and working the next unit could then be implemented and tested. The testing of the ASP code encountered a few minor problems. The first problem was the data provided in the food start and finish time text boxes. Extensive testing of this found that errors do occur when any values entered into the text boxes are not a time datatype or represent the string "Any Time". To fix this required a different approach to how a user would input the data. Instead of manually inputting the time into the text box, the user would be able to select from a drop down menu a particular time from a list. The lists of times are based on the hour. This method prevents the user from receiving an asp scripting error but does have the disadvantage that the times a user can input or select are less flexible.

6.2 Acceptance Testing

The main reason for this type of testing is to determine whether the user accepts the system. It can be used to determine whether user requirements have been understood. The fact that users were involved in the requirements stage of the project increased the likelihood of the system being accepted. If the analysis, design and implementation stages were carried out carefully then there should be a higher chance that the user will accept the system. To test this involved ten of my fellow piers on a range of different courses who would use the system and mark it out of a criterion shown in table 6.6.

Criteria	Range
How simple was the system to navigate?	1-10 (1 – V.Difficult, 10 – V.Simple)
Were all aspects of the system kept familiar?	1-10 (1 – Not at all, 10 – Yes, completely)
Did your actions result in something you expected?	1-10 (1 – Not at all, 10 – Yes, completely)
Does the interface provide a way of reversing those actions?	1-10 (1 – Not at all, 10 – Yes, completely)
Did the system provide the results you were looking for?	1-10 (1 – Not at all, 10 – Yes, completely)
Do you feel that the system facilitates for differing user abilities?	1-10 (1 – Not at all, 10 – Yes, completely)
To what extent does the interface stimulate interest?	1-10 (1 – Not at all, 10 – Yes, completely)
If the system was to be deployed then how likely are you to use it?	1-10 (1 – Not at all, 10 – All the time)

Table 6.6: An evaluation criterion to test users acceptance of the system

Results of acceptance testing

Having demonstrated the system to each of the ten users, the general consensus was that the system more than satisfied user requirements. The results in table 6.7 summarise the average results for the criterion shown in table 6.6.

Criteria	Av. Result
How simple was the system to navigate?	8.5
Were all aspects of the system kept familiar?	8.4
Did your actions result in something you expected?	8

Does the interface provide a way of reversing those actions?	7.6
Did the system provide the results you were looking for?	8.2
Do you feel that a high level of computer literacy is required to use the system?	7.7
To what extent does the interface stimulate interest?	7.5
If the system was to be deployed then how likely are you to use it?	6

Table 6.7: Testing results for evaluation criterion in table 6.6

The results show that the student user is satisfied with all aspects of the system, with each criterion scoring high average marks. A separate comment made by a few of my piers was that the interface would look more attractive using more images and colour schemes. The implication of this however is that the html or asp page requested by the user would take longer to download. This is not particularly desirable for the user. The download rate of a particular page could be the factor for a user to abandon the system altogether. This is my justification for not using as many graphics or images on the system interface. The users also suggested than an online map that specifically located the social establishment would be desirable. This was included as part of the future enhancements and will be discussed in the next chapter.

Involving the user in this testing phase provided useful feedback on possible enhancements to the system. It is important to make sure user requirements are always considered in order for them to accept the system.

7. EVALUATION

7.1 Why we need an Evaluation?

Evaluation is required to identify whether the system does what its supposed to do. Sections 3.4 and 3.5 identify user needs in the form of functional and non-functional requirements. These are brought together to form an evaluation criterion for which each requirement can be evaluated against the system implemented. The way the system is going to be evaluated is by identifying whether the minimum requirements have been met and exceeded.

7.2 Exceeding Minimum requirements

This section identifies the minimum requirements that are evaluated to see whether they have been met, exceeded or not met at all.

To evaluate possible software tools for the implementation of the project and understand why one piece of software is more appropriate than the others.

Before starting any design or implementation stage of the project, it was important to understand the various tools that could be used and more importantly, to understand why one particular tool is better than the other. I had no previous knowledge on tools that are responsible for server side or client side scripting so it was important to research these to find out their respected advantages and disadvantages. After conducting my research, I selected the following tools and languages: Microsoft Access, Microsoft FrontPage, ASP scripting language and VBScript. To evaluate whether these were the right choice of tools, I have used the following justifications:

- Microsoft Access – The only DBMS available that offers a user interface
- Microsoft FrontPage – The only web editor available that allows you to select from 3 different views.
- ASP scripting language – Produces dynamic web pages faster than PHP or CGI.
- VBScript – Combines best with the ASP scripting language.

Having understood all the different tools available, I was able to decide which tools were more suitable to this project based on their relative advantages and disadvantages. Having gained the knowledge that I have, it is fair to say that I have satisfied this requirement.

Produce a prototype relational database that holds information about pubs, clubs, bars and their promotions.

To identify whether this minimum requirement has been met, the functional requirements in section 3.4 can be used that directly relate to this minimum requirement.

The system must allow for Pubs to be added, deleted and updated

This requirement has been successfully implemented. The system provides an interface for the administrative user whereby they can add a pub, delete a pub, and update a pub

The system must allow for Clubs to be added, deleted and updated.

This requirement has been successfully implemented. The system provides an interface for the administrator user whereby they can add a club, delete a club, and update a club.

The system must allow for Bars to be added, deleted and updated.

This requirement has been successfully implemented. The system provides an interface for the administrator user whereby they can add a bar, delete a bar, and update a bar.

The system must allow for a Location to be added, deleted and updated.

This requirement has been successfully implemented. The system provides an interface for the administrator user whereby they can add a location, delete a location, and update a location

The system must allow for a Music type to be added, deleted and updated.

This requirement has been successfully implemented. The system provides an interface for the administrator user whereby they can add a music type, delete a music type, and update a music type

The system must allow for a Club or Bar Promotion to be added, deleted and updated.

This requirement has been successfully implemented. The system provides an interface for the administrator user whereby they can add a promotion, delete a promotion, and update a promotion for a particular club or bar.

A future enhancement relating to this minimum requirement was that the system could provide details on pub quizzes and Pub Food. This future enhancement has been successfully implemented so the user is able to add/delete and update pub quiz details and/or food details for any particular pub on any given day of the week.

Configure the database to provide a user-friendly interface that allows an administrator to add/delete/edit the information held.

This minimum requirement has not only been met, but also exceeded. To enable the user to carry out the first minimum requirement in a user-friendly way required the use of forms in Microsoft Access. Forms for each of the functional requirements outlined in section 3.4 were produced. Producing forms for menu screens that helped navigate the user through the system exceeded the minimum requirement. The system was also configured to produce error dialogues should the user make a data entry mistake. This helps the user to understand why an error was produced and how they go about correcting it.

Provide a web front that allows users to access the information held in the database.

This minimum requirement was met by providing a dynamically created drop down menu where the student was able to select a pub, club or bar that would result in the details of this establishment being displayed on the screen. A possible enhancement to this was to provide users with web-based search interface where they could search for social establishments depending on the criterion they select. This was fully implemented whereby users could search for pubs, clubs and bars depending on the criteria they are searching on. The implementation went further to provide the users with a set of results that matched their specified criteria. The user could then view further details by selecting an establishment from the set of results. Section 3.4 includes functional requirements that would satisfy future enhancements for the student user. These are shown in table 7.1 and also outlines whether they have been implemented.

FUNCTIONAL REQUIREMENT	**IMPLEMENTED**
The system could allow a Pub to be searched on by its location.	Yes
The system could allow a Pub to be searched on by its Food start time.	Yes
The system could allow a Pub to be searched on by its Food finish time.	Yes
The system could allow a Pub to be searched on by its Food day.	Yes
The system could allow a Pub to be searched on by its Pub quiz night.	Yes
The system could allow a Pub to be searched depending on whether it includes a pool table.	Yes
The system could allow a Pub to be searched depending on whether it includes a quiz machine.	Yes
The system could allow a Pub to be searched depending on whether it includes a jukebox.	Yes
The system could allow a Pub to be searched depending on whether it includes a wide screen TV.	Yes
The system could allow the search to be refined by using the above criteria in one single search	Yes
The system could allow a club or bar to be searched on by its promotion night.	Yes
The system could allow a club or bar to be searched on by its location.	Yes
The system could allow a club or bar to be searched on by its Music Type.	Yes
The system could allow a message to be posted into a forum.	Yes
The system could allow the user to view a map of each social establishment.	No

Table 7.1: Functional Requirements that satisfy Future Enhancements

The Web Interface was evaluated against the IBM guidelines for good interface design specified in Appendix D**.**

- ***Simplicity:*** The interface provided is simple with no adverts or unnecessary objects.
- ***Support:*** The system allows the user to have control over it by including familiar functions such as drop down menus.
- ***Familiarity:*** All aspects of the interface have the same layout, format and functionality so users are able to gain knowledge of how the system works.
- ***Obviousness:*** The interface makes use of real life representations such as buttons for navigational links. A button represents something that is pressed.
- ***Encouragement:*** The system responds to a request by doing something that a user expects. For example a user performing a search would expect results at the end of it. The system also provides the user with a way of re-visiting pages so they can go back to the search results without having to perform the search again.
- ***Satisfaction***: The system provides this by letting the user know they have performed an action. For example, a user is notified that their message has been posted.
- ***Availability:*** The frames page provides the user with an ever-present menu that they can select their required page from.
- ***Safety:*** The user is protected from making errors using VBScript that notifies them if they make an error. For example, an error message would be displayed if they had left a message text box empty.
- ***Versatility:*** Acceptance testing provided me with the information that users do not have to have a high level of computer literacy to use this system. The system also provides a help page.

7.3 Summary of further enhancements

Providing further enhancements to the system has clearly exceeded the minimum requirements that I set out to achieve. Table 7.2 summarises which future enhancements were achieved.

FUTURE ENHANCEMENT	**ACHIEVED?**
Provide details of quizzes and food for pubs.	ü
Provide users with web-based search facilities so the database can be searched in correspondence with the criterion that the user selects.	ü
Provide a web-based message board that users can post on.	ü
Produce a user manual for the database	ü
Provide a pop-up map for each establishment giving detailed directions.	X

Table 7.2: Future Enhancements Achieved

The reason why the last enhancement was not achieved was due to time constraints. The project required at least three weeks to complete the write-up. The user manual for the database had only just been completed with three weeks until the project deadline so it was an unrealistic aim to implement the final enhancement.

7.4 Advantages over other systems

The major advantage that this system brings is its ability to combine different searches in order to produce a set of results that are more acquainted to the user requirements. The interface provided is much simpler and is constructed in a way so users can be familiar with it. The system also provides a message board that encourages users to communicate with each other on favoured places or promotion nights.

7.5 Future Improvements

Described below are a set of future improvements that if implemented, would extend the systems functionality and usage.

Providing a dictionary for the message board

Currently, the message board is not administrated by anybody. An implication of this is that users could post offensive messages onto the message board. An improvement would be to contain a dictionary of words that are considered offensive, and provide a mechanism to restrict users from posting anything that is offensive.

Provide a pop-up map for each establishment giving detailed directions.

Providing a map of where a social establishment is located would be very beneficial to the user as it would remove the need to search for maps once an appropriate establishment has been found from the search. The map could be displayed dynamically in the same way that the image is displayed for each of the social establishments. As well as the map, detailed instructions of how to get to the establishment could also be included.

Provide a separate service where users can register and receive automatic emails with regards to the kind of venues/music they are into.

The service would allow users to register and provide information about their favourite establishments and music. This information would be used so that automatic emails could be sent to each different user dependent on their interests and tastes. The system would provide a more personalised service that would cater for user needs.

7.6 Conclusion

It seemed unrealistic to assume that the system produced will be put into deployment. The reality is that the system was carried out as a final year project and thus may come to nothing but that. However, there is a possibility that the system could be used in conjunction with the student unions university open day. This would provide a useful guide to those considering a university life in the city of Leeds.

Although the system is unlikely to go into deployment it has been a success in a number of ways. Firstly the project has met all the minimum requirements. Secondly, the project has exceeded minimum requirements by implementing five out of six possible enhancements. The system was also a success in the fact that it compared more favourably to the systems currently available.

The objectives pointed out in the summary section of the project were clearly satisfied by making good use of project management. The methodology selected was clearly the right choice allowing me to design, implement and test sections of the project in modules. Although it may be argued that the fifth stage in the software life cycle (Maintain) was not carried out, the methodology was still required to carry out the remaining stages in the project. Even though there is not a specific administrator for the system, the system still provides the functionality for maintaining it via a user interface. The project was set a strict deadline that was met, minimum requirements and future enhancements were carried out. In this respect, the project can be considered a success.

REFERENCES

[1] Avison D. E., and Fitzgerald G., (1995), *Information Systems Development: Methodologies, Techniques and Tools,* McGraw-Hill International

[2] Halpin, Terry, (1995), *Conceptual schema & relational database design***,** Prentice Hall

[3] Maciaszek leszek A., (2001), *Requirements analysis and system design: developing systems with uml*, Pearson education limited

[4] Lonsdale systems, (2002), Methodologies, http://members.iinet.net.au/~lonsdale/seng/se03.htm [12/01/2003]

[5] Ashenfelter John Paul, (1998), *Choosing a database for your website*, Wiley Publishers

[6] Paolo Atzeni/Stefano Ceri, Stefano Paraboschi and Riccardo Torlone, (1999), *Database systems: concepts, languages and architectures*, McGraw-hill publishing company

[7] Elmasri and Navathe, (2000), *Fundamentals of Database systems Third Edition*, Addison-Wesley

[8] Power ASP, (May 2003), Featured content, http://www.powerasp.com [12/04/2003]

[9] Gilmore W.J., (1999), DevShed . *PHP. What is it?*, http://www.devshed.com/Server_Side/PHP/Introduction/page1.html [16/01/2003]

[10] Macromedia Dreamweaver MX, (2003), http://www.macromedia.com/software/dreamweaver/ [14/01/2003]

[11] Ince, Darrel, (2002), *Developing Distributed and E-commerce Applications*, Addison-Wesley

[12] The Architecture Discipline, (2000), http://www.bredemeyer.com/use_cases.htm [04/04/2003]

[13] Requirements Engineering: Non-functional requirements, http://homepages.feis.herts.ac.uk/~3com0027/NONF(31).HTML [04/04/2003]

[14] Sommerville I., (1992), *Software Engineering (4th ed)*, Addison-Wesley

[15] Macros: what are they and how they work?, *Microsoft Access 200 Help* [20/04/2003]

[16] Learning technology dissemination initiative, Questionnaires: Advantages and Disadvantages http://www.icbl.hw.ac.uk/ltdi/cookbook/info_questionnaires/ [17/03/2003]

Appendix A – Personal Reflection

The purpose of this appendix was to reflect on my personal experience of the project rather than on the solution to the problem. The project has been both a challenging experience as well as a satisfying experience. The project has been the most difficult task while I have been at University but one that has provided me with an insight into how projects are managed in the real world. The project also gave me a chance to learn new skills and develop existing skills particularly in databases and web development. The project was extremely challenging and often quite stressful. The biggest challenge I faced was sticking to a tight project schedule that I had drawn up before the project was started. I had been naïve to think that I would complete my implementation by the end of March, leaving all April to complete the report. I had quickly realised that my current schedule was unrealistic, and one that had to be revised in order to follow it. I had not anticipated the amount of work that was required for such a project. Being a 40 credits project, the time that should be spent on this is 350 hours. To produce the solution and report that I have, required a far greater number of hours, which I found particular satisfaction with upon completion.

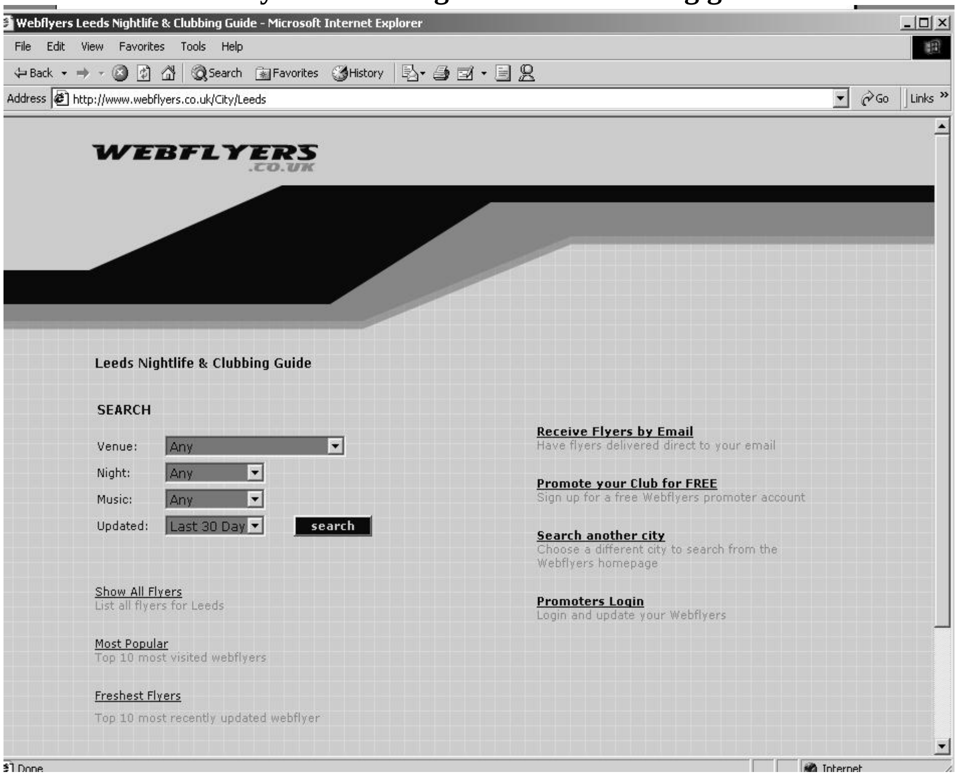
I believe the project aims I set out to achieve have been achieved. I now believe I have some vital experience in project management that I will be able to provide to my future employer. Managing the project from start to finish had its up's and its down's but was extremely satisfying to complete it before the deadline. There were factors that hindered the project such as the closure of the lab, which put the implementation stage back. Also, the implementation completion date was unrealistic resulting in a revised project schedule. This provided me with less time to complete the report for the project, meaning after-hours time was required in level 7 laboratory.

Having reflected on my project, I would offer the following advice to a student who wishes to take a project of a similar nature.

- Don't be too unrealistic with your project schedule and prioritise getting the implementation stage complete with at least a month to go before the deadline. This will provide you with sufficient time to write your report.
- Follow your methodology as it is actually useful!
- Having selected the appropriate software tools to use, make sure you understand them fully before using them otherwise you will waste time trying to work out any errors that occur
- Finally, I would recommend that any work that requires the use of the computing laboratories be completed a week before the deadline. The lab gets extremely busy and you will probably have to queue a long time for a computer.

Appendix B – Interfaces of Current Systems

Webflyers Leeds Nightlife and Clubbing guide


Webflyers Leeds Nightlife & Clubbing Guide - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://www.webflyers.co.uk/City/Leeds
Go Links
WEBFLYERS .CO.UK
search
Done
Internet

Leeds Nightlife & Clubbing Guide

SEARCH

Venue: Any

Night: Any

Music: Any

Updated: Last 30 Day

Show All Flyers
List all flyers for Leeds

Most Popular
Top 10 most visited webflyers

Freshest Flyers
Top 10 most recently updated webflyer

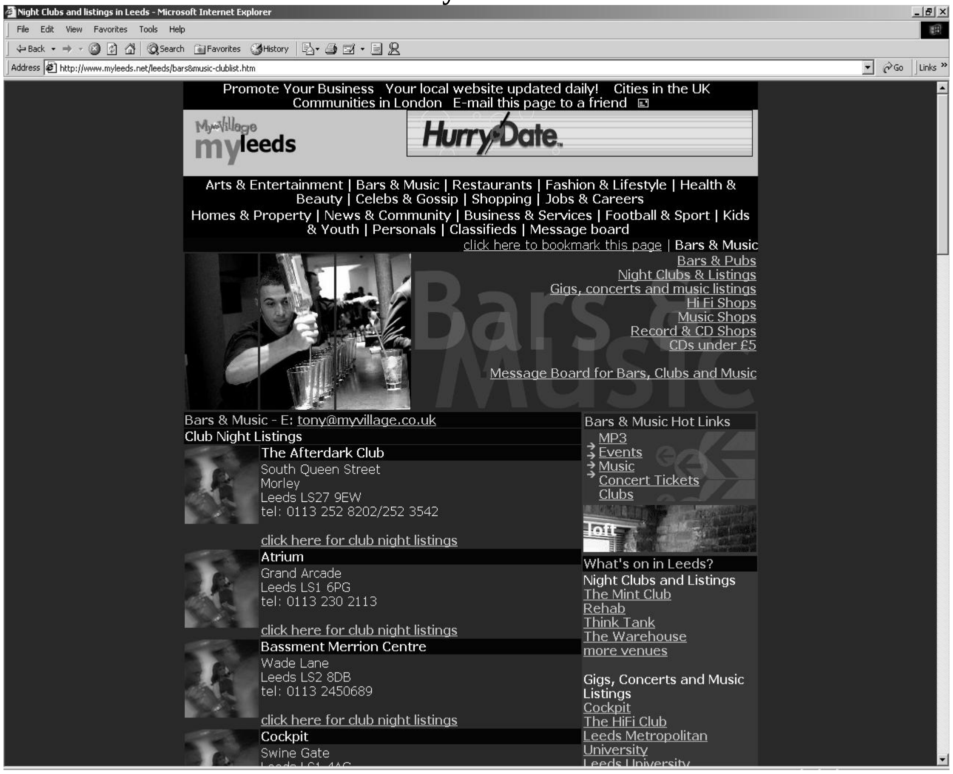
Receive Flyers by Email
Have flyers delivered direct to your email

Promote your Club for FREE
Sign up for a free Webflyers promoter account

Search another city
Choose a different city to search from the Webflyers homepage

Promoters Login
Login and update your Webflyers

My Leeds


Night Clubs and listings in Leeds - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://www.myleeds.net/leeds/bars&music-clublist.htm
Go Links
My Village myleeds
HurryDate
Bars & Music
loft

Promote Your Business   Your local website updated daily!   Cities in the UK
Communities in London   E-mail this page to a friend

Arts & Entertainment | Bars & Music | Restaurants | Fashion & Lifestyle | Health & Beauty | Celebs & Gossip | Shopping | Jobs & Careers
Homes & Property | News & Community | Business & Services | Football & Sport | Kids & Youth | Personals | Classifieds | Message board

click here to bookmark this page | Bars & Music

Bars & Pubs
Night Clubs & Listings
Gigs, concerts and music listings
Hi Fi Shops
Music Shops
Record & CD Shops
CDs under £5

Message Board for Bars, Clubs and Music

Bars & Music - E: tony@myvillage.co.uk

Club Night Listings

The Afterdark Club
South Queen Street
Morley
Leeds LS27 9EW
tel: 0113 252 8202/252 3542

click here for club night listings

Atrium
Grand Arcade
Leeds LS1 6PG
tel: 0113 230 2113

click here for club night listings

Bassment Merrion Centre
Wade Lane
Leeds LS2 8DB
tel: 0113 2450689

click here for club night listings

Cockpit
Swine Gate

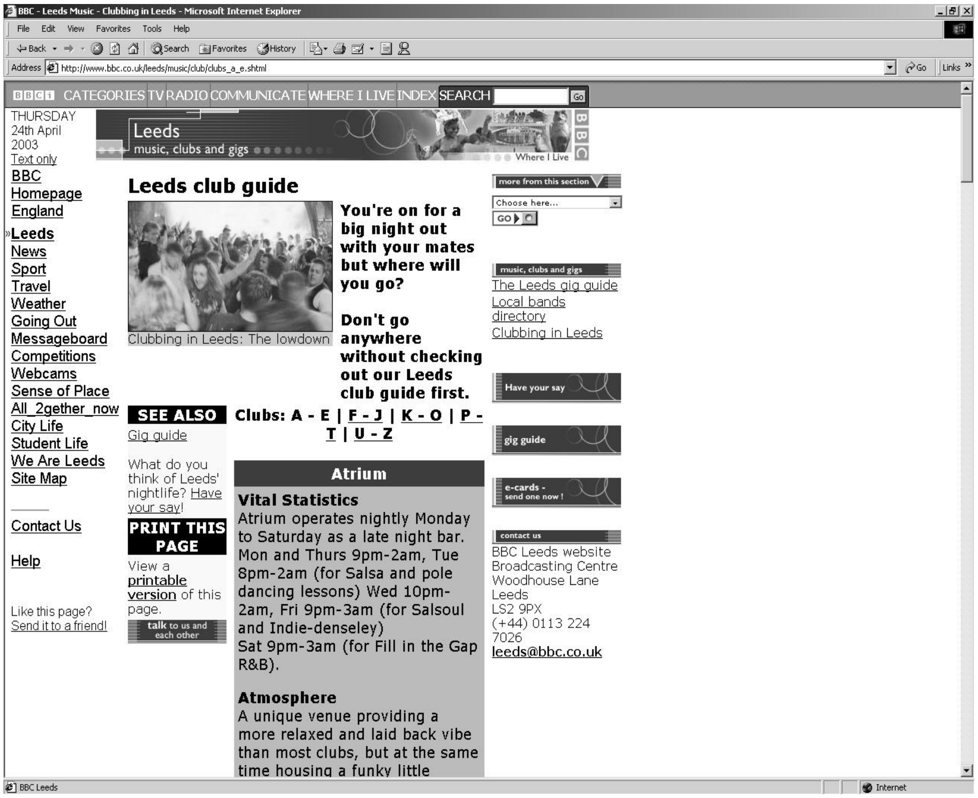
Bars & Music Hot Links
MP3
Events
Music
Concert Tickets
Clubs

What's on in Leeds?

Night Clubs and Listings
The Mint Club
Rehab
Think Tank
The Warehouse
more venues

Gigs, Concerts and Music Listings
Cockpit
The HiFi Club
Leeds Metropolitan University
Leeds University

BBC Leeds Club Guide


BBC - Leeds Music - Clubbing in Leeds - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://www.bbc.co.uk/leeds/music/club/clubs_a_e.shtml
Go Links
BBCi CATEGORIES TV RADIO COMMUNICATE WHERE I LIVE INDEX SEARCH Go
Leeds
music, clubs and gigs
Where I Live
BBC
more from this section
Choose here...
GO
music, clubs and gigs
Have your say
gig guide
e-cards - send one now !
contact us
talk to us and each other
BBC Leeds
Internet

THURSDAY
24th April
2003
Text only

BBC Homepage
England

»Leeds
News
Sport
Travel
Weather
Going Out
Messageboard
Competitions
Webcams
Sense of Place
All_2gether_now
City Life
Student Life
We Are Leeds
Site Map

Contact Us

Help

Like this page?
Send it to a friend!

Leeds club guide

Clubbing in Leeds: The lowdown

You're on for a big night out with your mates but where will you go?

Don't go anywhere without checking out our Leeds club guide first.

SEE ALSO
Gig guide

What do you think of Leeds' nightlife? Have your say!

PRINT THIS PAGE
View a printable version of this page.

Clubs: A - E | F - J | K - O | P - T | U - Z

Atrium

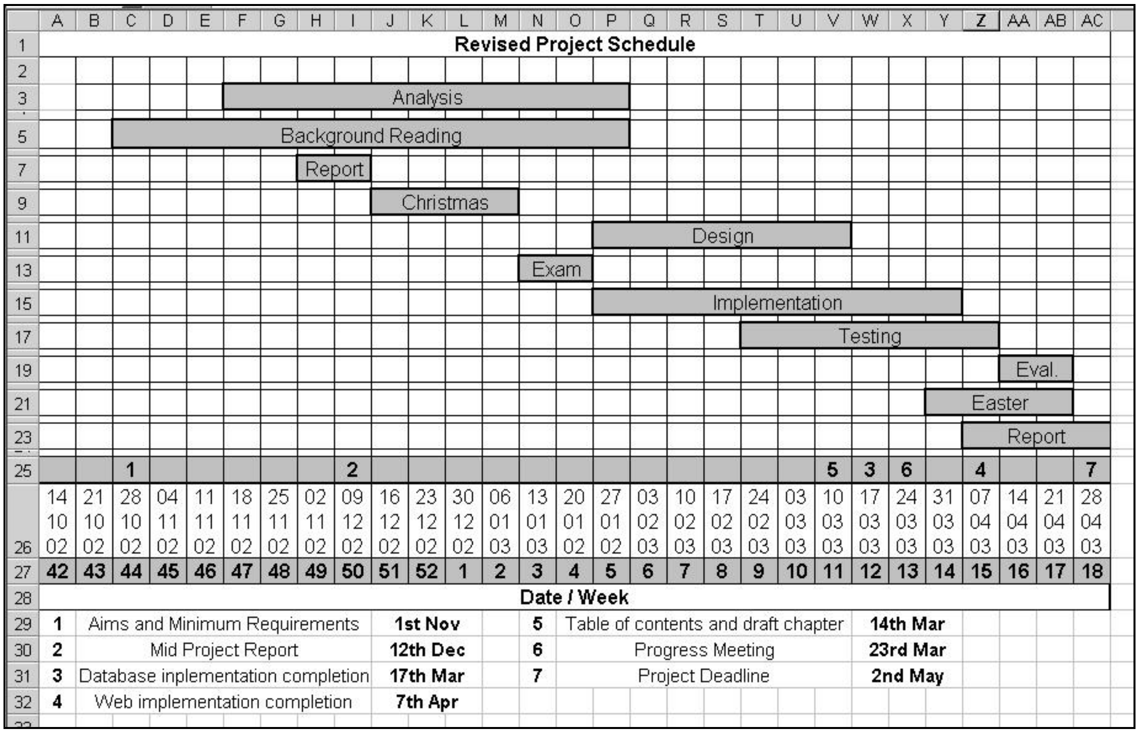
Vital Statistics
Atrium operates nightly Monday to Saturday as a late night bar. Mon and Thurs 9pm-2am, Tue 8pm-2am (for Salsa and pole dancing lessons) Wed 10pm-2am, Fri 9pm-3am (for Salsoul and Indie-denseley)
Sat 9pm-3am (for Fill in the Gap R&B).

Atmosphere
A unique venue providing a more relaxed and laid back vibe than most clubs, but at the same time housing a funky little

The Leeds gig guide
Local bands directory
Clubbing in Leeds

BBC Leeds website
Broadcasting Centre
Woodhouse Lane
Leeds
LS2 9PX
(+44) 0113 224 7026
leeds@bbc.co.uk

Appendix C – Gantt chart



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
25			1						2													5	3	6		4			7
26	14 10 02	21 10 02	28 10 02	04 11 02	11 11 02	18 11 02	25 11 02	02 11 02	09 12 02	16 12 02	23 12 02	30 12 02	06 01 03	13 01 03	20 01 02	27 01 02	03 02 03	10 02 03	17 02 03	24 02 03	03 03 03	10 03 03	17 03 03	24 03 03	31 03 03	07 04 03	14 04 03	21 04 03	28 04 03
27	42	43	44	45	46	47	48	49	50	51	52	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
28	Date / Week																												

1	Aims and Minimum Requirements	1st Nov	5	Table of contents and draft chapter	14th Mar
2	Mid Project Report	12th Dec	6	Progress Meeting	23rd Mar
3	Database inplementation completion	17th Mar	7	Project Deadline	2nd May
4	Web implementation completion	7th Apr			

The Gantt chart above has been revised from the one that was submitted with the mid-term project report. This was due to the fact that the earlier Gantt chat was too unrealistic.

Appendix D – IBM Principles For Good Interface Design

Design basics

The design principles presented here combine traditional wisdom with extensions to address the evolution of future interfaces. Existing design principles are based on our own experiences in user interface design, on the design experiences of others, and on insights from linguistics and psychology. We have extended these design principles to address evolving interfaces that will provide a more friendly appearance and behaviour in the future. The increasing use of 3-D and real-world representations as well as the blossoming popularity of the Internet and the World Wide Web have strongly influenced these progressions.

The most recent influence on these principles has come from our design experience in creating an object-oriented user interface (OOUI). IBM pioneered OOUI architecture and design. Popular operating systems such as Windows 95, IBM OS/2 Warp, and CDE for Unix provide varying degrees of object-orientation for users.

In order to effectively apply these design principles, you need to understand users' tasks and requirements. Understanding and applying principles will be meaningless if users are unhappy with the final product.

Our goal for user interface design is to have the interface positively support users' endeavours and never intrude adversely. The interface should be transparent to the task the user is trying to accomplish and be efficient, satisfying, and fun to use.

Design Principles

Simplicity: Don't compromise usability for function

Keep the interface simple and straightforward. Users benefit from function that is easily accessible and usable. A poorly organized interface cluttered with many advanced functions distracts users from accomplishing their everyday tasks. A well-organized interface that supports the user's tasks fades into the background and allows the user to work efficiently.

Basic functions should be immediately apparent, while advanced functions may be less obvious to new users. Function should be included only if a task analysis shows it is needed. Therefore, keep the number of objects and actions to a minimum while still allowing users to accomplish their tasks.

Support: Place the user in control and provide proactive assistance

To give users control over the system, enable them to accomplish tasks using any sequence of steps that they would naturally use. Don't limit them by artificially restricting their choices to your notion of the "correct" sequence.

The system should also allow users to establish and maintain a working context, or frame of reference. The current state of the system and the actions that users can perform should be obvious. Users should be able to leave their systems for a moment or a day and find the systems in the same familiar state when they return. This contextual framework contributes to their feeling of stability. Most users perform a variety of tasks, being expert at some and novice at others. In addition to providing assistance when requested, the system should recognize and anticipate the user's goals, and offer assistance to make the task easier. Ideally, assistance should provide users with knowledge that will allow them to accomplish their tasks quickly. Intelligent assistance is like the training wheels on a bicycle - at some point, most users will want to take them off and go forward on their own. The assistance should allow them to become independent at some point when they choose to be so.

Familiarity: Build on users' prior knowledge

Allow users to build on prior knowledge, especially knowledge they have gained from experience in the real world. A small amount of knowledge, used consistently throughout an interface, can empower the user to accomplish a large number of tasks. Concepts and techniques can be learned once and then applied in a variety of situations. Users should not have to learn new things to perform familiar tasks. The use of concepts and techniques that users already understand from their real world experiences allows them to get started quickly and make progress immediately.

The metaphors used in today's user interfaces tend to be inadequate when compared to the real world. Through the use of visuals and interaction techniques that more closely resemble users' real world experiences, there should be little need to continue reliance on such metaphors.

In the past, designers tended to invoke a principle of consistency when no single design alternative appeared to be the best answer. By choosing to be consistent with something the user already understands, an interface can be made easier to learn, more productive, and even fun to use.

Avoid the tendency to employ consistency without understanding your users, their tasks, and their shared experiences. When choosing a dimension within which to be consistent, seek to understand what the user expects and be consistent with those expectations. Providing a familiar experience is the ultimate use of consistency in which a truly intuitive interface will result.

Obviousness: Make objects and their controls visible and intuitive

Where you can, use real-world representations in the interface. Real-world representations and natural interactions (direct action) give the interface a familiar look and feel and can make it more intuitive to learn and use. Icons and windows were early attempts to draw on user experiences outside the computing domain. As we move toward real-world representations, reliance on such computer artefacts should decline. In an object-oriented interface the objects and concepts presented to users parallel familiar things from the real world; for example:

Trash can - when we throw things away we usually use some type of trash receptacle or "trash can". An object on the desktop displayed as a trash can communicates to users that it is a place for discarding things. It should look like the real object rather than like an abstract container, and the user should be able to show its contents in a meaningful way.

Telephone - the actions we take with telephones are so familiar to most of us that they require little thought. A telephone object on the desktop indicates to users that it will allow them to perform phone-related tasks, and users will expect it to behave like the real thing.

The controls of the system should be clearly visible and their functions identifiable. Visual representations provide cues and reminders that help users understand roles, remember relationships, and recognize what the computer is doing. For example, the numbered buttons on the telephone object indicate that they can be used to key in a telephone number.

Allow users to interact directly with objects and minimize the use of indirect techniques. Identifying an object and doing something with it (like picking up the handset of a phone to answer it) usually are not separate actions in the real world. Likewise, with direct action techniques, explicit selection is not necessary because selection is implicit in the actions users take with objects. Real-world 3D interfaces are especially conducive to direct interaction.

Encouragement: Make actions predictable and reversible

A user's actions should cause the results the user expects. In order to meet those expectations, the designer must understand the user's tasks, goals, and mental model. Use terms and images that match users' task experience, and that help users understand the objects and their roles and relationships in accomplishing tasks.

Users should feel confident in exploring, knowing they can try an action, view the result, and undo the action if the result is unacceptable. Users feel more comfortable with interfaces in which their actions do not cause irreversible consequences.

Even seemingly trivial user actions, such as deselection or moving objects, should be reversible. For example, a user who spends several minutes deliberating and selecting individual files to be archived from a group will be very upset if all the files are accidentally deselected and the deselection cannot be undone.

Avoid bundling actions together, because the user may not anticipate the side effect. For example, if a user chooses to cancel a request to send a note, only the send request should be cancelled. Do not bundle another action, such as deletion of the note, with the cancel request. Rather than implementing composite actions, make actions independent and provide ways to allow users to combine them when they wish.

Satisfaction: Create a feeling of progress and achievement

Allow the user to make uninterrupted progress and enjoy a sense of accomplishment. Reflect the results of actions immediately; any delay intrudes on users' tasks and erodes confidence in the system. Immediate feedback allows users to assess whether the results were what they expected and to take alternative action immediately. For example, when a user chooses a new font, the font of all applicable text, or of sample text, should change immediately. The user can then decide if the effect is what was desired and, if not, can change it before switching attention to something else.

Offer a preview of the results of an action when it would be inconvenient for a user to apply the action and then reverse it. For example, if a user wants to bold, underscore, and use Helvetica font in certain places throughout a document, provide a sample part of that document with those changes applied, allowing the user to decide if that is the right action to take. This saves the user a lot of time by not having to reverse the action that's been applied to an entire document and enhances the user's confidence in the system.

Avoid situations where users may be working with information that is not up-to-date. Information should be updated immediately or refreshed as soon as possible so that users are not making incorrect decisions or assumptions. If, for some reason, the results of a refresh cannot be displayed immediately, the situation should be communicated to users. This becomes especially important in networked environments where it is more difficult to maintain state between networked systems dynamically. For example, most Web browsers display a completion percentage in the information area so that users know the progress of the graphics loading process.

Safety: Keep the user out of trouble

Users should be protected from making errors. The burden of keeping the user out of trouble rests on the designer. The interface should provide visual cues, reminders, lists of choices, and other aids, either automatically or on request. Humans are much better at recognition than recall. Contextual and hover help, as well as agents, can provide supplemental assistance. Simply stated, eliminate the opportunity for user error and confusion.

Users should never have to rely on their own memory for something the system already knows, such as previous settings, file names, and other interface details. If the information is in the system in any form, the system should provide it.

Two-way communication may be necessary at times to allow users to clarify or confirm requests, or to remedy a problem. In the past, many interfaces have treated communication with users as primarily one-way, computer-to-user. The communication should be interactive - as rich in presentation and interaction capabilities as the rest of the interface. It should present relevant information, provide access to related information and help, and allow users to make task-specific decisions to continue. For instance, spell check, as designed in some systems, highlights potentially misspelled words as users work, allowing them to either select a new word or continue to work until they reach a point where they can go back and validate the potentially misspelled words.

Adopt the following design perspective: users know what they want to accomplish, but sometimes they find it difficult to express their desires using the objects and actions provided, and the system is unable to recognize their request. Two-way communication may be used to help users reach their goals.

Versatility: Support alternate interaction techniques

Allow users to choose the method of interaction that is most appropriate to their situation. Interfaces that are flexible in this way are able to accommodate a wide range of user skills, physical abilities, interactions, and usage environments.

Each interaction device is optimised for certain uses or users and may be more convenient in one situation than another. For example, a microphone used with voice-recognition software can be helpful for fast entry of text or in a hands-free environment. Pen input is helpful for people who sketch, and mouse input works well for precisely indicating a selection. Alternative output formats, such as computer-generated voice output for foreign language instruction, are useful for some purposes. No single method is best for every situation.

Users should be allowed to switch between methods to accomplish a single interaction. For example, allow the user to swipe-select using the mouse, then to adjust the selection using the keyboard. At the same time, users should not be *required* to alternate between input devices to accomplish what they perceive as a single step or a series of related steps in a task. For example, it would be tedious to require the use of a mouse for scrolling while editing text from the keyboard. Users should be able to complete an entire useful sequence through the same input device.

Providing a range of interaction techniques recognizes that users are individuals with different abilities and situations. The differences include disabilities, preferences, and work environments.

Personalization: Allow users to customize

The interface should be tailorable to individual users' needs and desires. No two users are exactly alike. Users have varying backgrounds, interests, motivations, levels of experience, and physical abilities. Customization can help make an interface feel comfortable and familiar.

Personalizing a computer interface can also lead to higher productivity and user satisfaction. For example, allowing users to change default values can save them time and hassle when accessing frequently used functions.

In an environment where multiple users are using a shared machine, allow the users to create their own system personality and make it easy to reset the system. In an environment where one user may be using many computers, make personalization information portable so the user can carry that "personality" from one system to another.

Affinity: Bring objects to life through good visual design

The goal of visual design in the user interface is to surface to the user in a cohesive manner all aspects of the design principles. Visual design should support the user model and communicate the function of that model without ambiguities. Visual design should not be the "icing on the cake" but an integral part of the design process. The final result should be an intuitive and familiar representation that is second nature to users.

The following are visual design principles that promote clarity and visual simplicity in the interface:

Subtractive design - reduce clutter by eliminating any visual element that doesn't contribute directly to visual communication.

Visual hierarchy - by understanding the importance of users' tasks, establish a hierarchy of these tasks visually. An important object can be given extra visual prominence. Relative position and contrast in color and size can be used.

Affordance - when users can easily determine the action that should be taken with an object, that object displays good affordance. Objects with good affordance usually mimic real world objects.

Visual scheme - design a visual scheme that maps to the user model and lets the user customize the interface. Do not eliminate extra space in your image just to save space. Use white space to provide visual "breathing room."

Appendix E – Database Schema

Pub

Pub_id	Pub_name	Address	Location	Postcode	Tel No	Description
Pool_table	Quiz_machine	Dartboard	Jukebox	Widescreen_TV	Image	

Food

Pub_id	**Night_id**	Food_details	Start_time	Finish_time

Pub_quiz

Pub_id	**Night_id**	Quiz_details	Start_time	Finish_time

Location

Location_id	Location

Night

Night_id

Venue

Venue_id	Venue_name	Type	Address	Location	Postcode	Tel No	Description

Capacity	No_of_rooms	Image

Promotions

Promo_id	Promo_name	Start_date	End_date	Dresscode	Entry Price

Drinks_offers	**Venue_id**	**Night_id**	Music_id

Music

Music_id	Music_type	Description

Message_Board

Message_id	Name	Message	Date

Appendix F – Data Definition

Pub

Attribute	Data Type	Length	Keys	Allow NULLS	Input mask
Pub_id	Autonumber	Long Int	Primary	No	
Pub_name	Text	50		No	
Address	Text	50			
Location	Number	Long Int	Foreign	No	
Postcode	Text				>LL09\ 0LL
Tel_No	Text				\(00009") "0000009
Description	Text	200			
Pool_table	Yes/No				
Quiz_machine	Yes/No				
Dartboard	Yes/No				
Jukebox	Yes/No				
Widescreen_TV	Yes/No				
Image	Text	50			

Pub_quiz

Attribute	Data Type	Length	Keys	Allow NULLS	Validation Rule	Input mask
Pub_id	Number	Long Int	Primary/Foreign	No		
Night_id	Number	Long Int	Primary/Foreign	No		
Quiz_Details	Text	200				
Start_time	Date/time					00:00
Finish_time	Date/time				>Start_time	00:00

Food

Attribute	Data Type	Length	Keys	Allow NULLS	Validation Rule	Input mask
Pub_id	Number	Long Int	Primary/Foreign	No		
Night_id	Number	Long Int	Primary/Foreign	No		
Food_Details	Text	200				
Start_time	Date/time					00:00
Finish_time	Date/time				>Start_time	00:00

Location

Attribute	Data Type	Length	Keys	Allow NULLS
Location_id	Autonumber	Long Int	Primary	No
Location	Text	50		No

Night

Attribute	Data Type	Length	Keys	Allow NULLS
Night_id	Text	50	Primary	No

Music

Attribute	Data Type	Length	Keys	Allow NULLS
Music_id	Autonumber	Long Int	Primary	No
Music_Type	Text	50		No
Description	Text	200		

Venue

Attribute	Data Type	Length	Keys	Allow NULLS	Input mask
Venue_id	Autonumber	Long Int	Primary	No	
Venue_name	Text	50		No	
Type	Text	50			
Address	Text	50			
Location	Number	Long Int	Foreign	No	
Postcode	Text				>LL09\ 0LL
Tel_No	Text				\(00009") "0000009
Capacity	Number	10			
No_of_rooms	Number	5			
Description	Text	200			
Image	Text	50			

Promotions

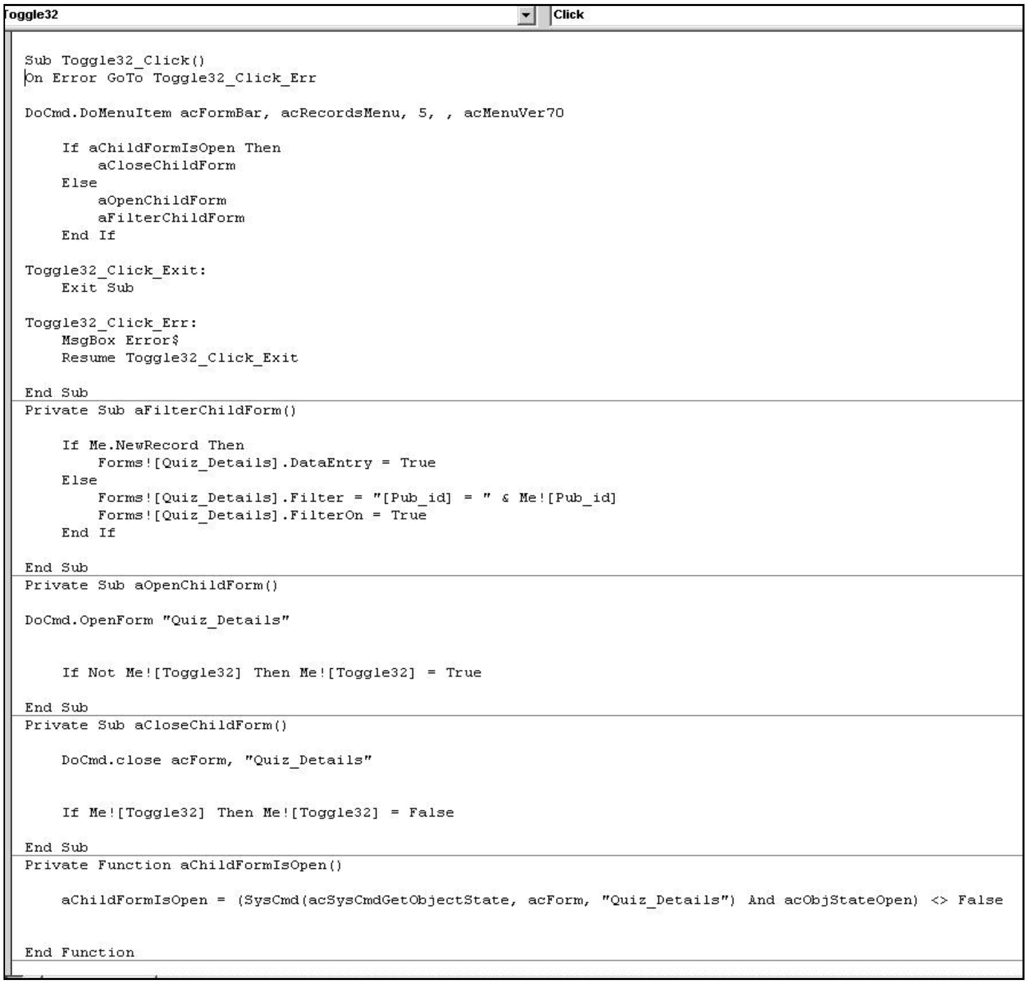
Attribute	Data Type	Length	Keys	Allow NULLS	Validation Rule	Input mask
Promo_id	Autonumber	Long Int		No		
Promo_name	Text	50				
Start_date	Text				>Date()	0/00/0000
End_date	Date/time				>Start_date	0/00/0000
Dresscode	Text	50				
Entry_Price	Currency	2 d.p.				
Drinks_offers	Text	100				
Venue_id	Number	Long Int	Primary/ Foreign	No		
Night_id	Number	Long Int	Primary/ Foreign	No		
Music_id	Text	200	Foreign	No		

Message_Board

Attribute	Data Type	Length	Keys	Allow NULLS	Input mask
Message_id	Autonumber	Long Int	Primary	No	
Name	Text	50		No	
Message	Text	400		No	
Date	Date/time			No	=Date()

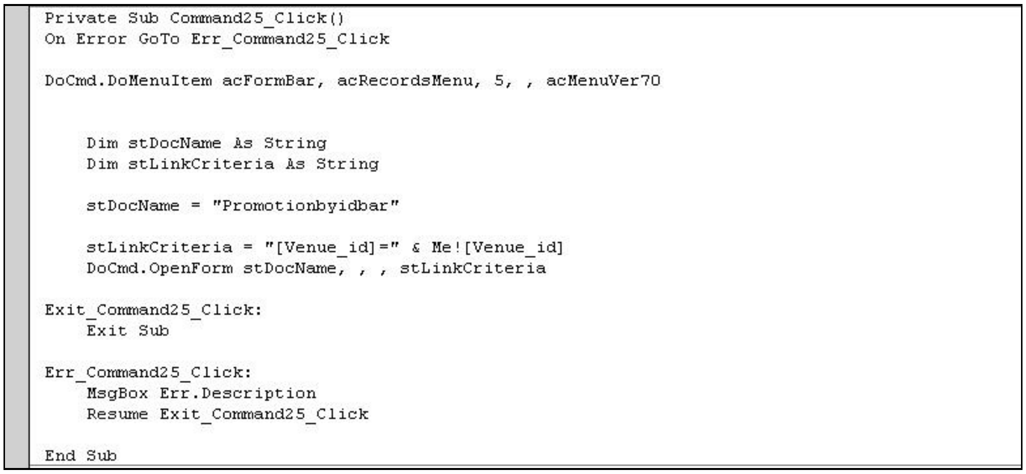
Appendix G – Visual Basic Code

Code to link the “Pub_quiz” Form to the “Pub” Form:



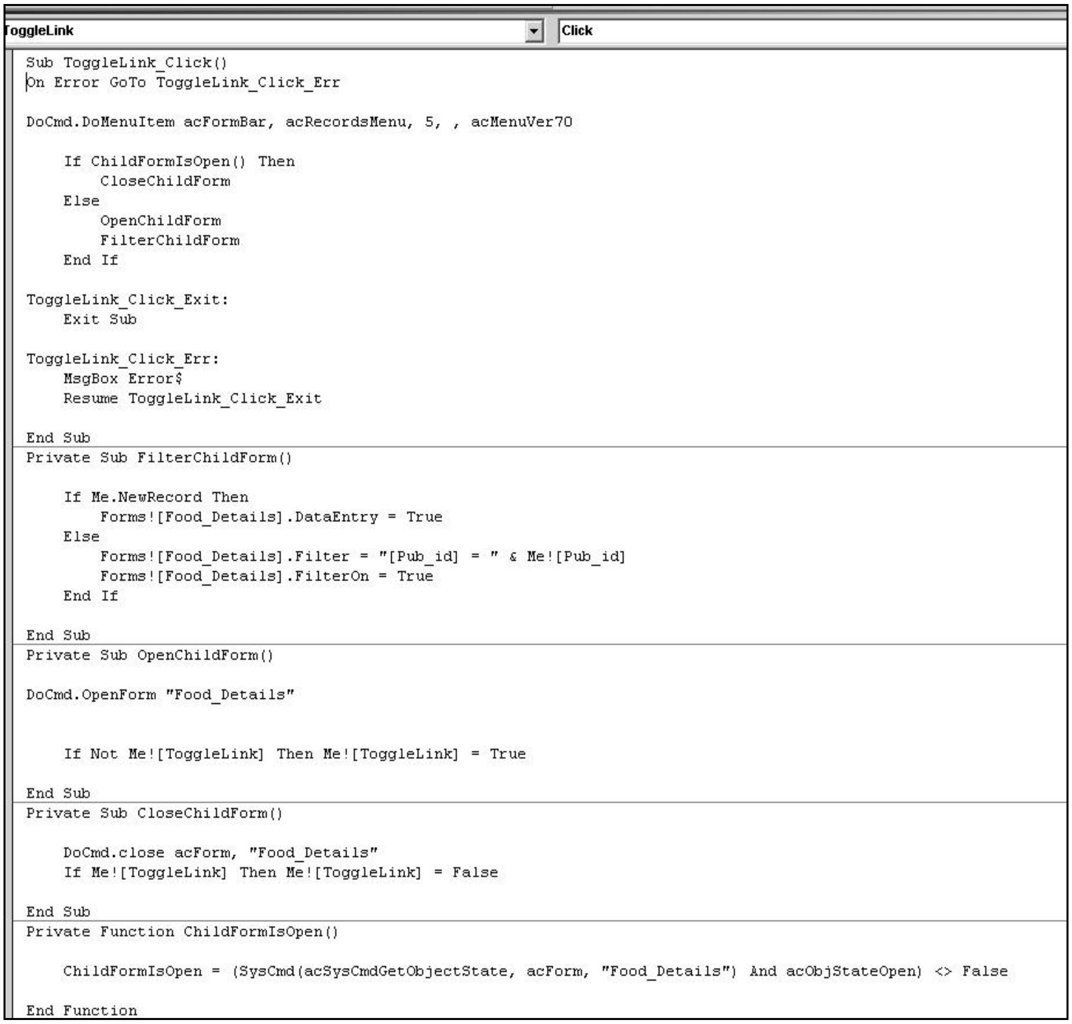
```
Sub Toggle32_Click()
On Error GoTo Toggle32_Click_Err

DoCmd.DoMenuItem acFormBar, acRecordsMenu, 5, , acMenuVer70

    If aChildFormIsOpen Then
        aCloseChildForm
    Else
        aOpenChildForm
        aFilterChildForm
    End If

Toggle32_Click_Exit:
    Exit Sub

Toggle32_Click_Err:
    MsgBox Error$
    Resume Toggle32_Click_Exit

End Sub
Private Sub aFilterChildForm()

    If Me.NewRecord Then
        Forms![Quiz_Details].DataEntry = True
    Else
        Forms![Quiz_Details].Filter = "[Pub_id] = " & Me![Pub_id]
        Forms![Quiz_Details].FilterOn = True
    End If

End Sub
Private Sub aOpenChildForm()

DoCmd.OpenForm "Quiz_Details"


    If Not Me![Toggle32] Then Me![Toggle32] = True

End Sub
Private Sub aCloseChildForm()

    DoCmd.close acForm, "Quiz_Details"


    If Me![Toggle32] Then Me![Toggle32] = False

End Sub
Private Function aChildFormIsOpen()

    aChildFormIsOpen = (SysCmd(acSysCmdGetObjectState, acForm, "Quiz_Details") And acObjStateOpen) <> False


End Function
```

Code to link the “Promotions” Form to the “Club” and “Bar” Forms:



```
Private Sub Command25_Click()
On Error GoTo Err_Command25_Click

DoCmd.DoMenuItem acFormBar, acRecordsMenu, 5, , acMenuVer70


    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "Promotionbyidbar"

    stLinkCriteria = "[Venue_id]=" & Me![Venue_id]
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Command25_Click:
    Exit Sub

Err_Command25_Click:
    MsgBox Err.Description
    Resume Exit_Command25_Click

End Sub
```

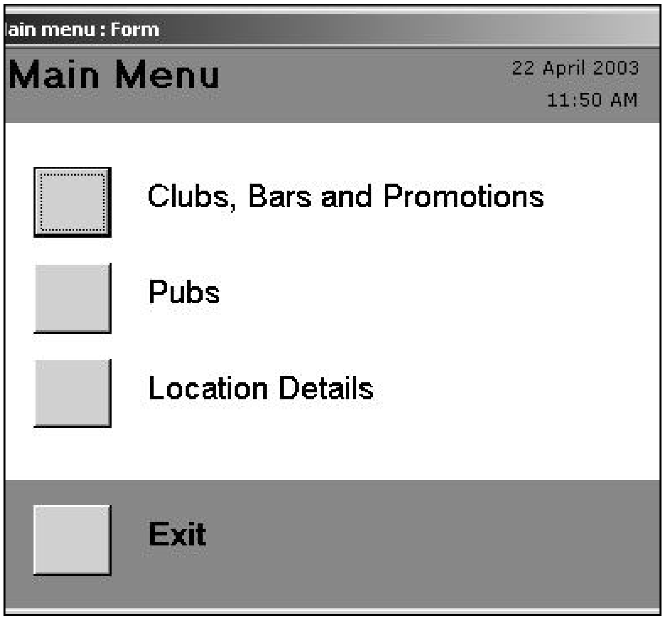
Code to link the “Food” Form to the “Pub” Form:



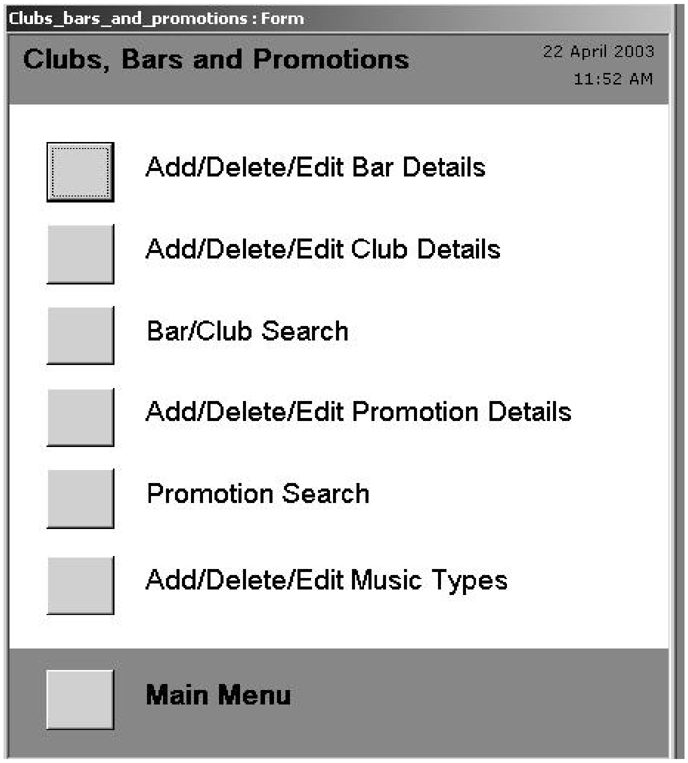
```
Sub ToggleLink_Click()
On Error GoTo ToggleLink_Click_Err

DoCmd.DoMenuItem acFormBar, acRecordsMenu, 5, , acMenuVer70

    If ChildFormIsOpen() Then
        CloseChildForm
    Else
        OpenChildForm
        FilterChildForm
    End If

ToggleLink_Click_Exit:
    Exit Sub

ToggleLink_Click_Err:
    MsgBox Error$
    Resume ToggleLink_Click_Exit

End Sub
Private Sub FilterChildForm()

    If Me.NewRecord Then
        Forms![Food_Details].DataEntry = True
    Else
        Forms![Food_Details].Filter = "[Pub_id] = " & Me![Pub_id]
        Forms![Food_Details].FilterOn = True
    End If

End Sub
Private Sub OpenChildForm()

DoCmd.OpenForm "Food_Details"


    If Not Me![ToggleLink] Then Me![ToggleLink] = True

End Sub
Private Sub CloseChildForm()

    DoCmd.close acForm, "Food_Details"
    If Me![ToggleLink] Then Me![ToggleLink] = False

End Sub
Private Function ChildFormIsOpen()

    ChildFormIsOpen = (SysCmd(acSysCmdGetObjectState, acForm, "Food_Details") And acObjStateOpen) <> False

End Function
```

Appendix H – Database Interface screenshots

Main Menu Form

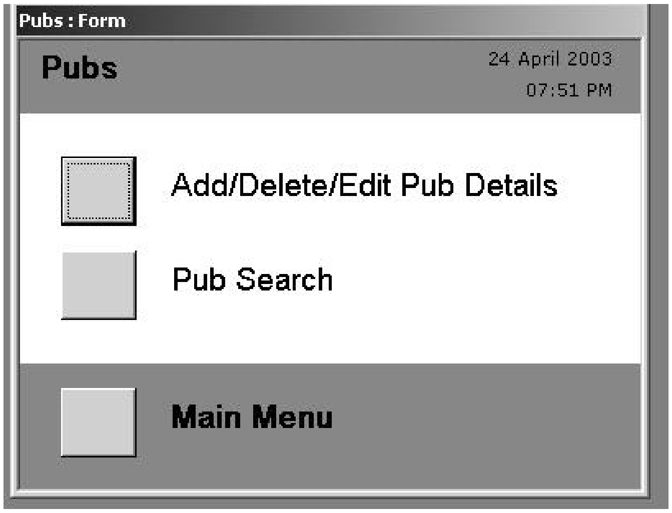


Clubs, Bars and Promotions Menu Form

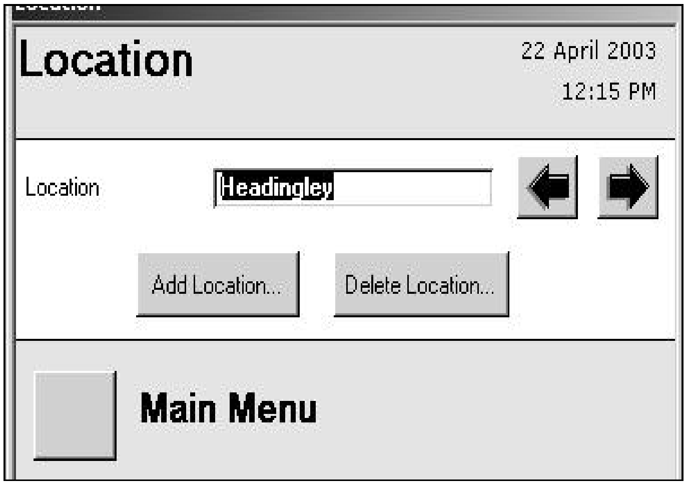


Clubs_bars_and_promotions : Form
Clubs, Bars and Promotions
22 April 2003
11:52 AM
Add/Delete/Edit Bar Details
Add/Delete/Edit Club Details
Bar/Club Search
Add/Delete/Edit Promotion Details
Promotion Search
Add/Delete/Edit Music Types
Main Menu

Pubs Menu Form

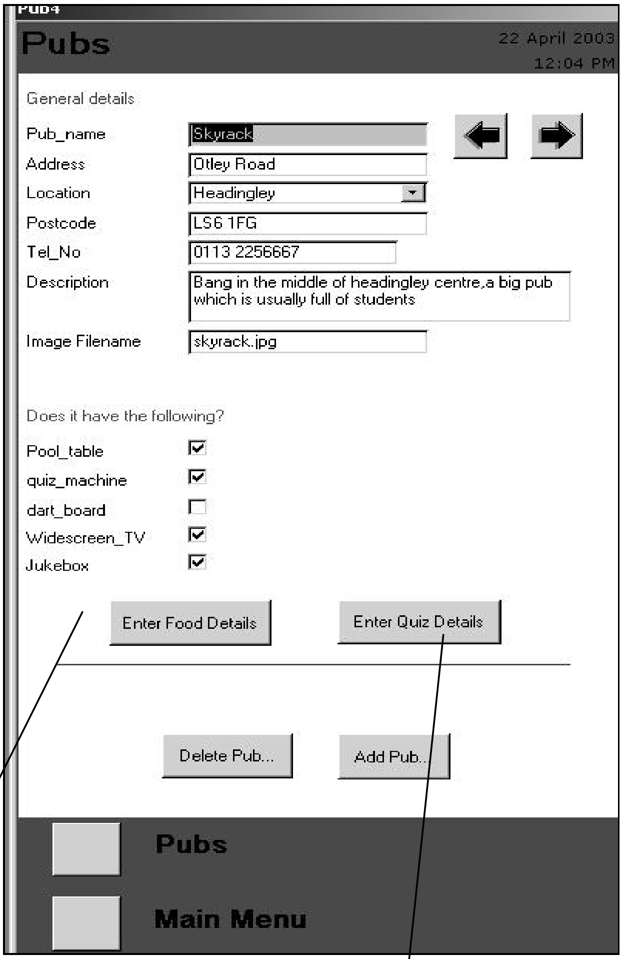


Location Form



Location Headingley

Pubs Form



Pub4

Pubs

22 April 2003
12:04 PM

General details

Pub_name: Skyrack

Address: Otley Road

Location: Headingley

Postcode: LS6 1FG

Tel_No: 0113 2256667

Description: Bang in the middle of headingley centre,a big pub which is usually full of students

Image Filename: skyrack.jpg

Does it have the following?

Pool_table ☑

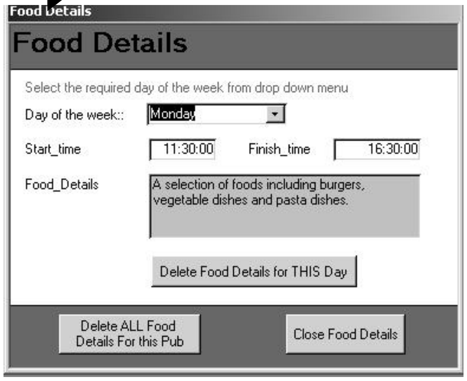
quiz_machine ☑

dart_board □

Widescreen_TV ☑

Jukebox ☑

Food Details Form



Food Details

Food Details

Select the required day of the week from drop down menu

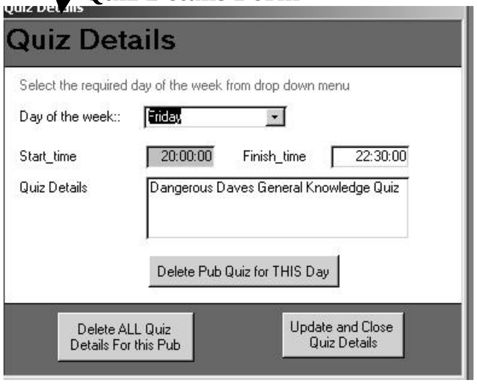
Day of the week:: Monday

Start_time 11:30:00 Finish_time 16:30:00

Food_Details: A selection of foods including burgers, vegetable dishes and pasta dishes.

Delete Food Details for THIS Day
Delete ALL Food Details For this Pub
Close Food Details

Quiz Details Form



Quiz Details

Quiz Details

Select the required day of the week from drop down menu

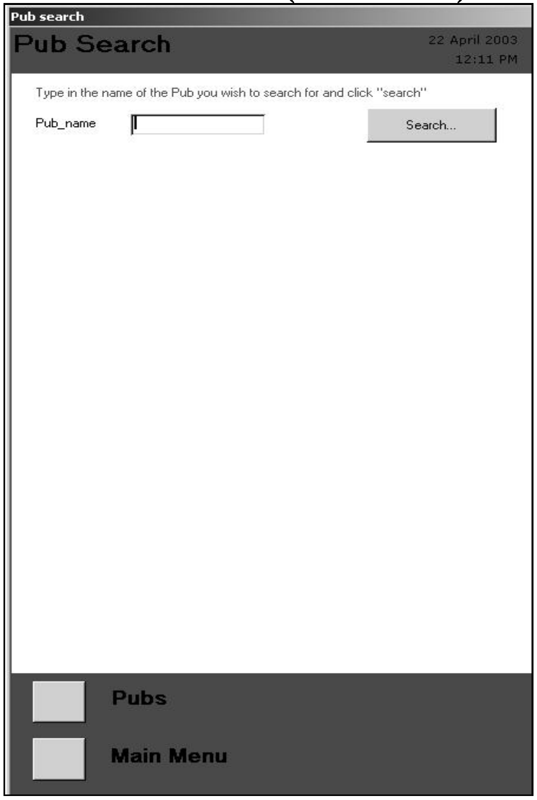
Day of the week:: Friday

Start_time 20:00:00 Finish_time 22:30:00

Quiz Details: Dangerous Daves General Knowledge Quiz

Delete Pub Quiz for THIS Day
Delete ALL Quiz Details For this Pub
Update and Close Quiz Details

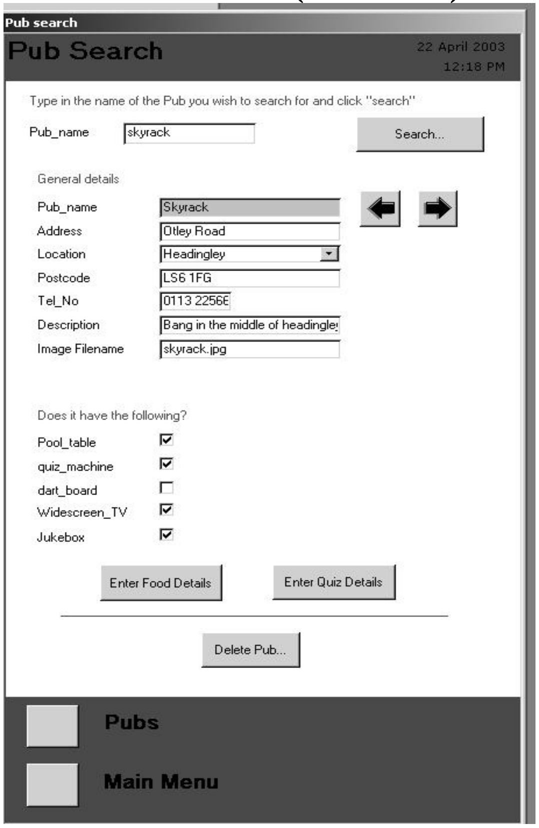
Pub Search Form (Before search)



Type in the name of the Pub you wish to search for and click "search"

Pub_name

Pub Search Form (After search)



Type in the name of the Pub you wish to search for and click "search"

Pub_name: skyrack

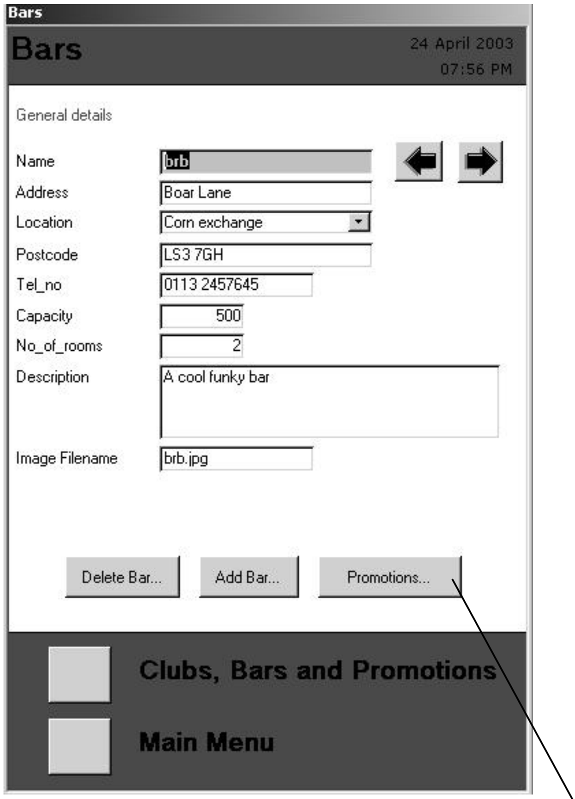
General details

Pub_name	Skyrack
Address	Otley Road
Location	Headingley
Postcode	LS6 1FG
Tel_No	0113 22568
Description	Bang in the middle of headingley
Image Filename	skyrack.jpg

Does it have the following?

Pool_table	☑
quiz_machine	☑
dart_board	☐
Widescreen_TV	☑
Jukebox	☑

Bars Form



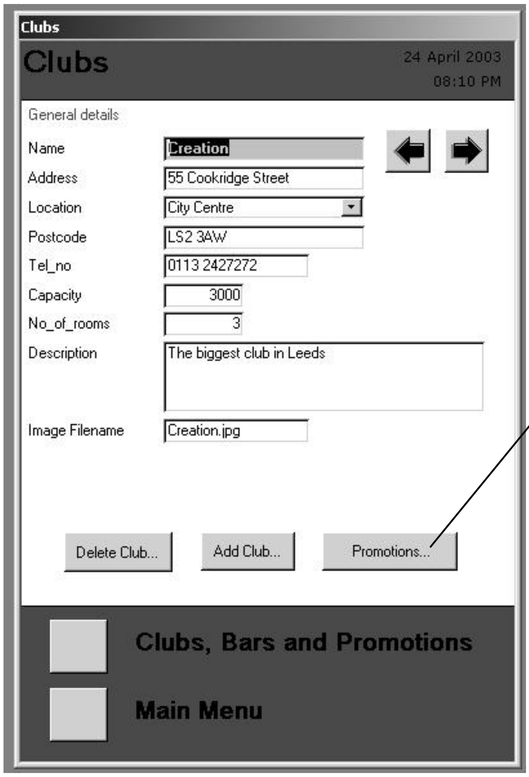
Bars

24 April 2003
07:56 PM

General details

Name: brb
Address: Boar Lane
Location: Corn exchange
Postcode: LS3 7GH
Tel_no: 0113 2457645
Capacity: 500
No_of_rooms: 2
Description: A cool funky bar
Image Filename: brb.jpg

Clubs Form



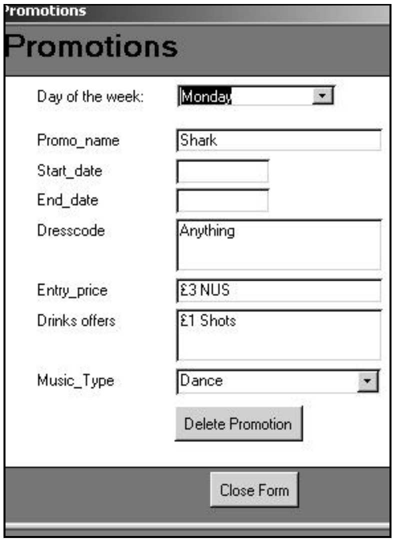
Clubs

24 April 2003
08:10 PM

General details

Name: Creation
Address: 55 Cookridge Street
Location: City Centre
Postcode: LS2 3AW
Tel_no: 0113 2427272
Capacity: 3000
No_of_rooms: 3
Description: The biggest club in Leeds
Image Filename: Creation.jpg

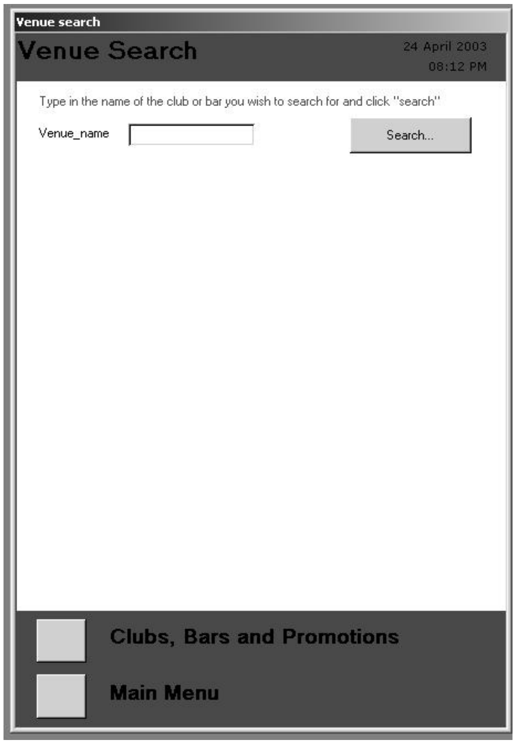
Promotions (by venue id) Form



Promotions

Day of the week: Monday
Promo_name: Shark
Start_date:
End_date:
Dresscode: Anything
Entry_price: £3 NUS
Drinks offers: £1 Shots
Music_Type: Dance

Club/Bar Search Form (Before Search)

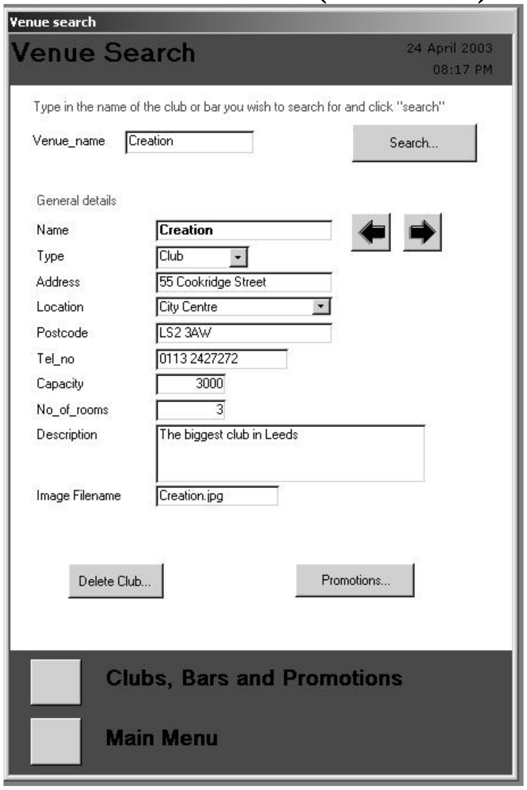


24 April 2003
08:12 PM

Type in the name of the club or bar you wish to search for and click "search"

Venue_name

Club/Bar Search Form (After Search)



24 April 2003
08:17 PM

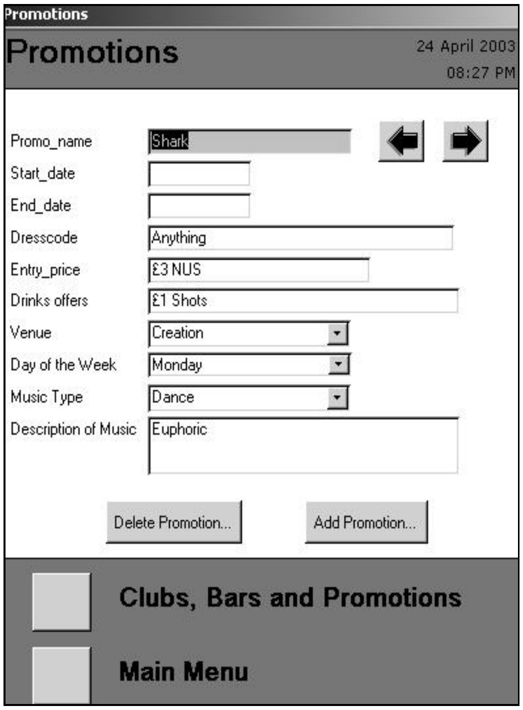
Type in the name of the club or bar you wish to search for and click "search"

Venue_name Creation

General details

Name	Creation
Type	Club
Address	55 Cookridge Street
Location	City Centre
Postcode	LS2 3AW
Tel_no	0113 2427272
Capacity	3000
No_of_rooms	3
Description	The biggest club in Leeds
Image Filename	Creation.jpg

Promotions Form



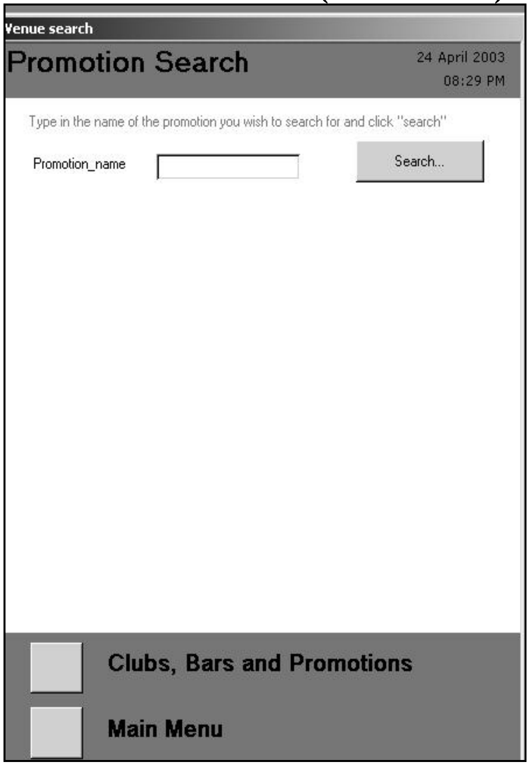
Promotions

Promotions

24 April 2003
08:27 PM

Promo_name	Shark
Start_date	
End_date	
Dresscode	Anything
Entry_price	£3 NUS
Drinks offers	£1 Shots
Venue	Creation
Day of the Week	Monday
Music Type	Dance
Description of Music	Euphoric

Promotion Search Form (Before Search)



Venue search

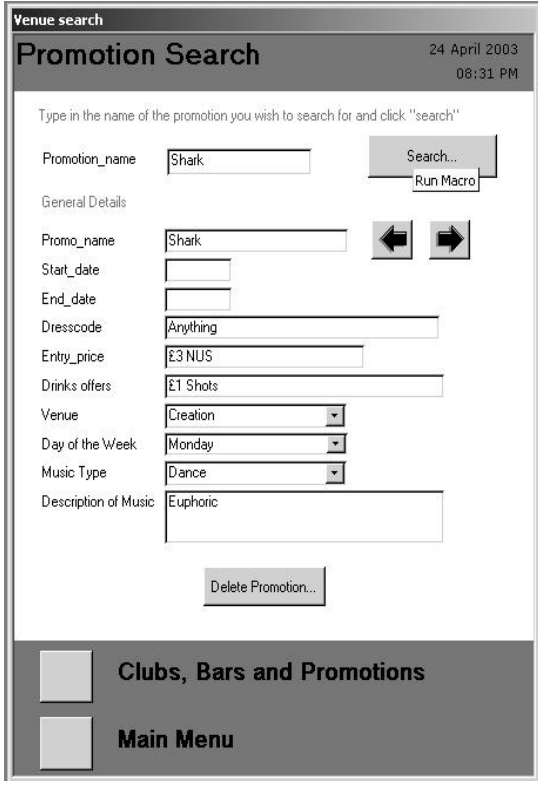
Promotion Search

24 April 2003
08:29 PM

Type in the name of the promotion you wish to search for and click "search"

Promotion_name

Promotion Search Form (After Search)



Venue search

Promotion Search

24 April 2003
08:31 PM

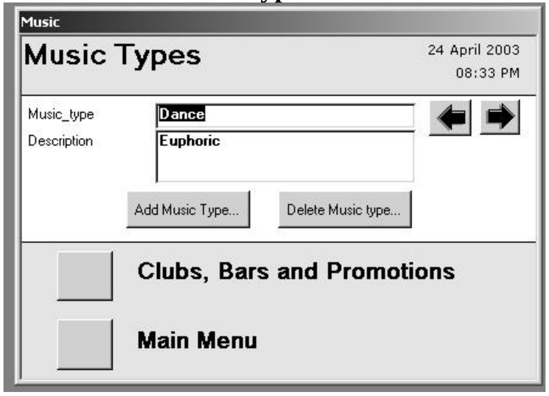
Type in the name of the promotion you wish to search for and click "search"

Promotion_name: Shark

General Details

Promo_name: Shark
Start_date:
End_date:
Dresscode: Anything
Entry_price: £3 NUS
Drinks offers: £1 Shots
Venue: Creation
Day of the Week: Monday
Music Type: Dance
Description of Music: Euphoric

Music Types Form



Music

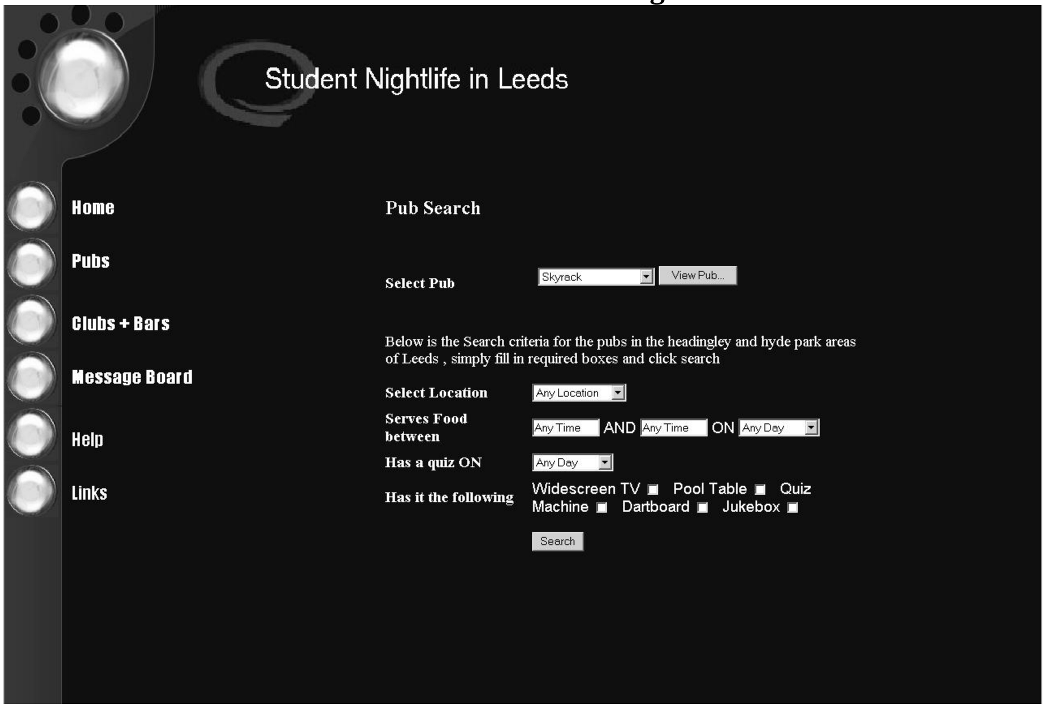
Music Types

24 April 2003
08:33 PM

Music_type: Dance
Description: Euphoric

Appendix I – Web Interface screenshots

Pub Search Page



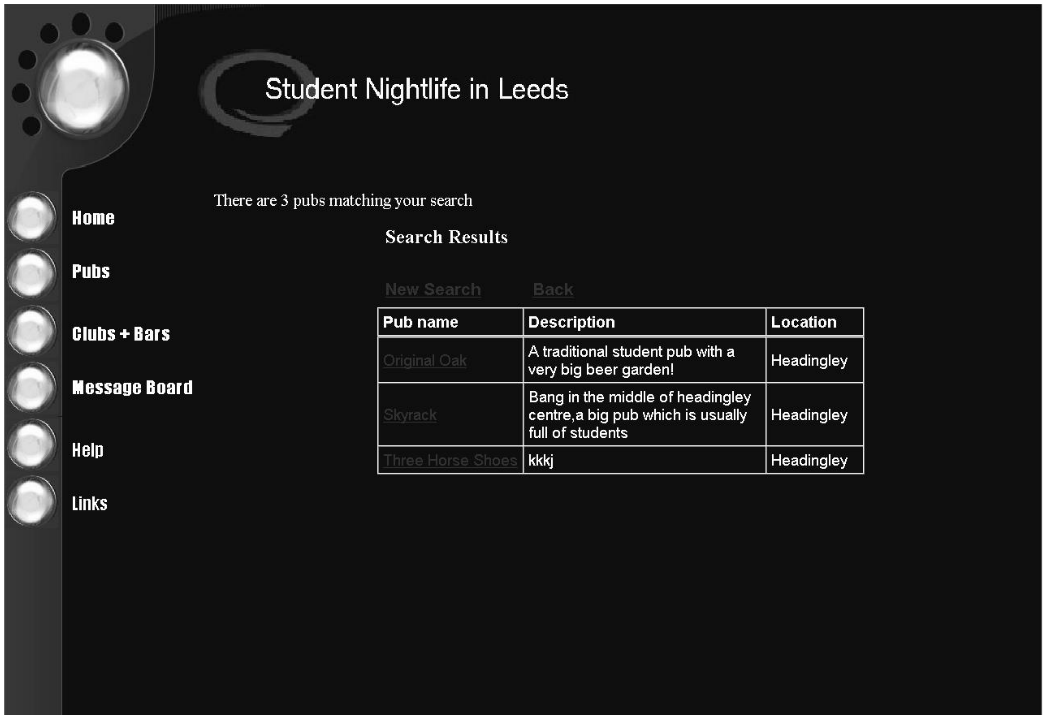
Pub Search

Select Pub: Skyrack

Below is the Search criteria for the pubs in the headingley and hyde park areas of Leeds , simply fill in required boxes and click search

Select Location	Any Location
Serves Food between	Any Time AND Any Time ON Any Day
Has a quiz ON	Any Day
Has it the following	Widescreen TV □ Pool Table □ Quiz Machine □ Dartboard □ Jukebox □

Pub Results Page

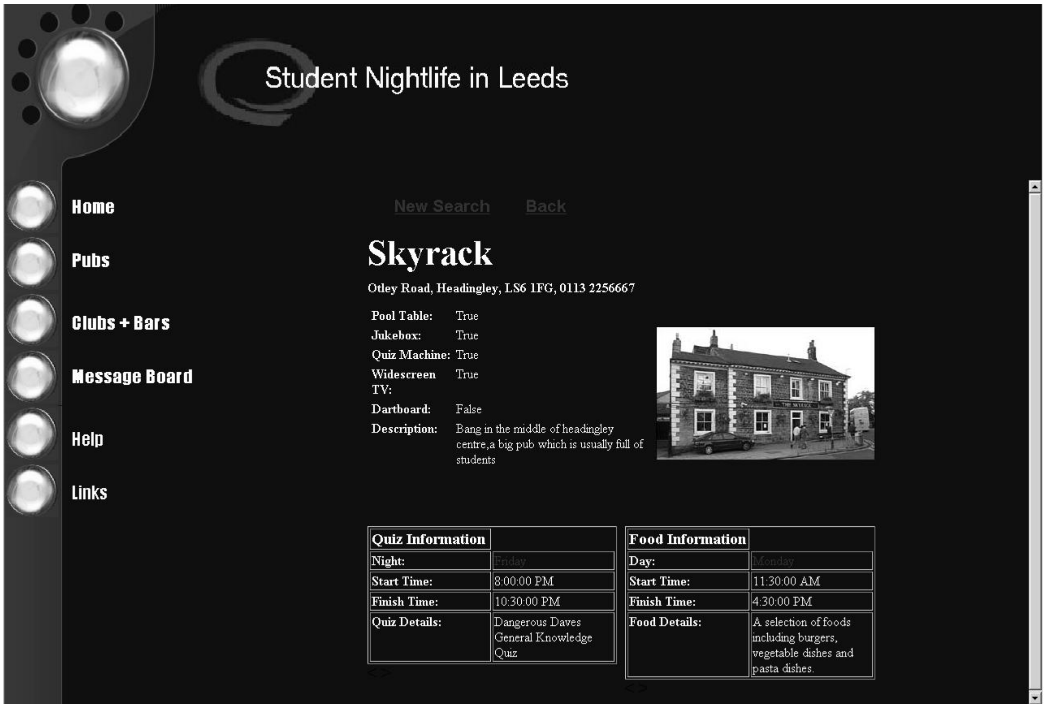


There are 3 pubs matching your search

Search Results

Pub name	Description	Location
Original Oak	A traditional student pub with a very big beer garden!	Headingley
Skyrack	Bang in the middle of headingley centre,a big pub which is usually full of students	Headingley
Three Horse Shoes	kkkj	Headingley

Pub Detail Page



Skyrack

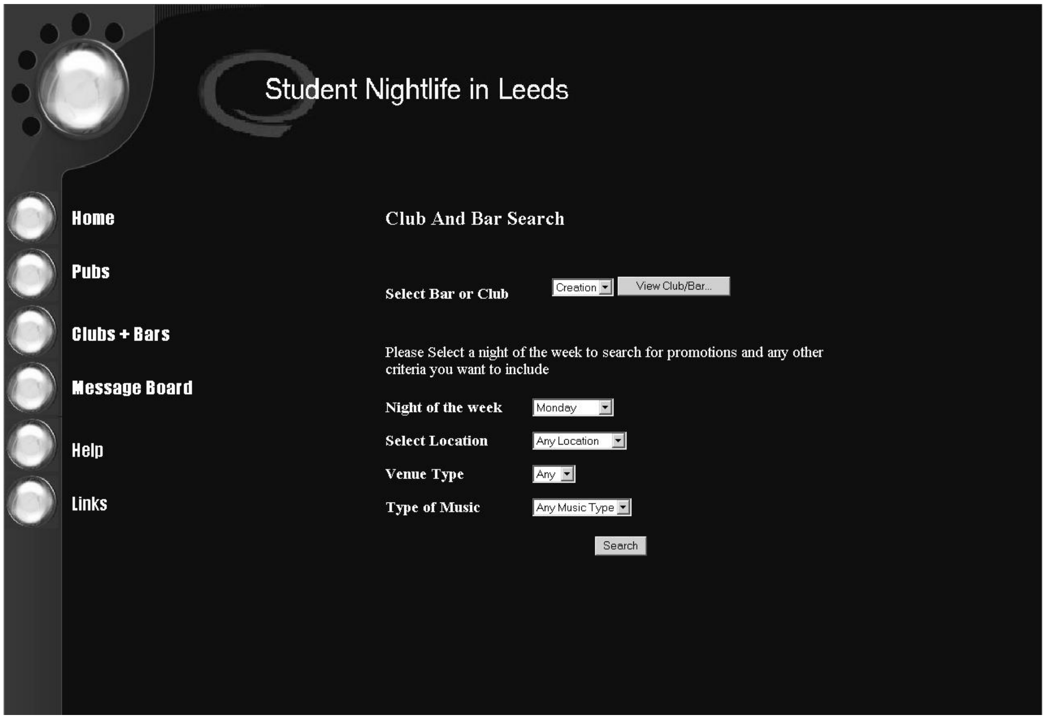
Otley Road, Headingley, LS6 1FG, 0113 2256667

Pool Table:	True
Jukebox:	True
Quiz Machine:	True
Widescreen TV:	True
Dartboard:	False
Description:	Bang in the middle of headingley centre,a big pub which is usually full of students

Quiz Information	
Night:	Friday
Start Time:	8:00:00 PM
Finish Time:	10:30:00 PM
Quiz Details:	Dangerous Daves General Knowledge Quiz

Food Information	
Day:	Monday
Start Time:	11:30:00 AM
Finish Time:	4:30:00 PM
Food Details:	A selection of foods including burgers, vegetable dishes and pasta dishes.

Club/Bar Search Page



Club And Bar Search

Select Bar or Club: Creation

Please Select a night of the week to search for promotions and any other criteria you want to include

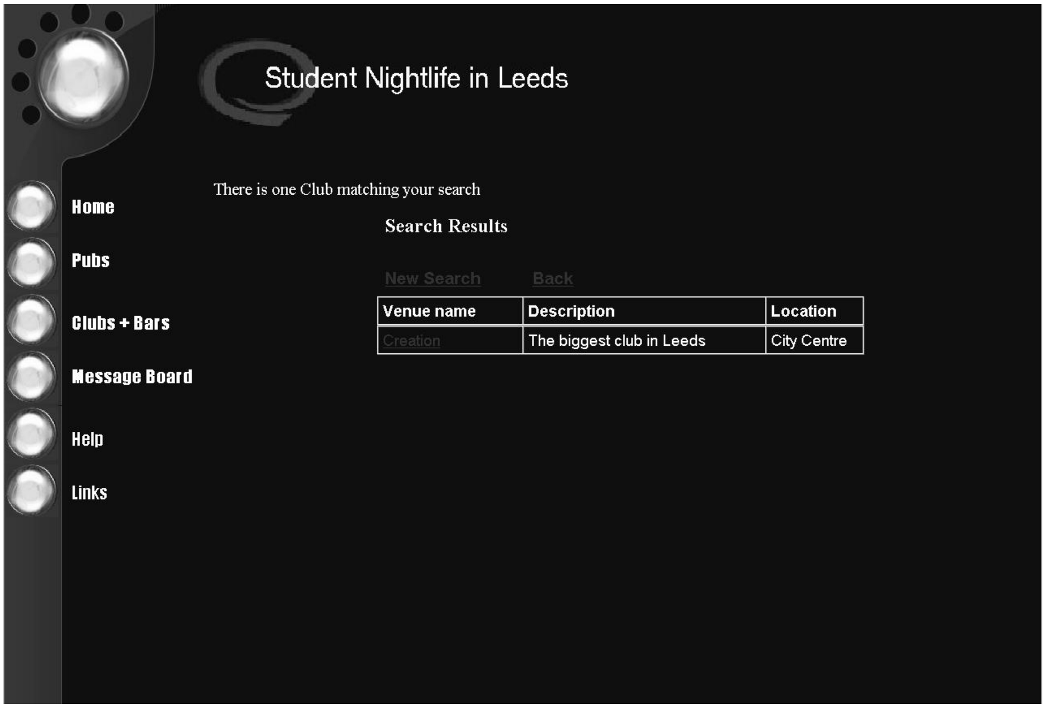
Night of the week: Monday

Select Location: Any Location

Venue Type: Any

Type of Music: Any Music Type

Club/Bar Results Page

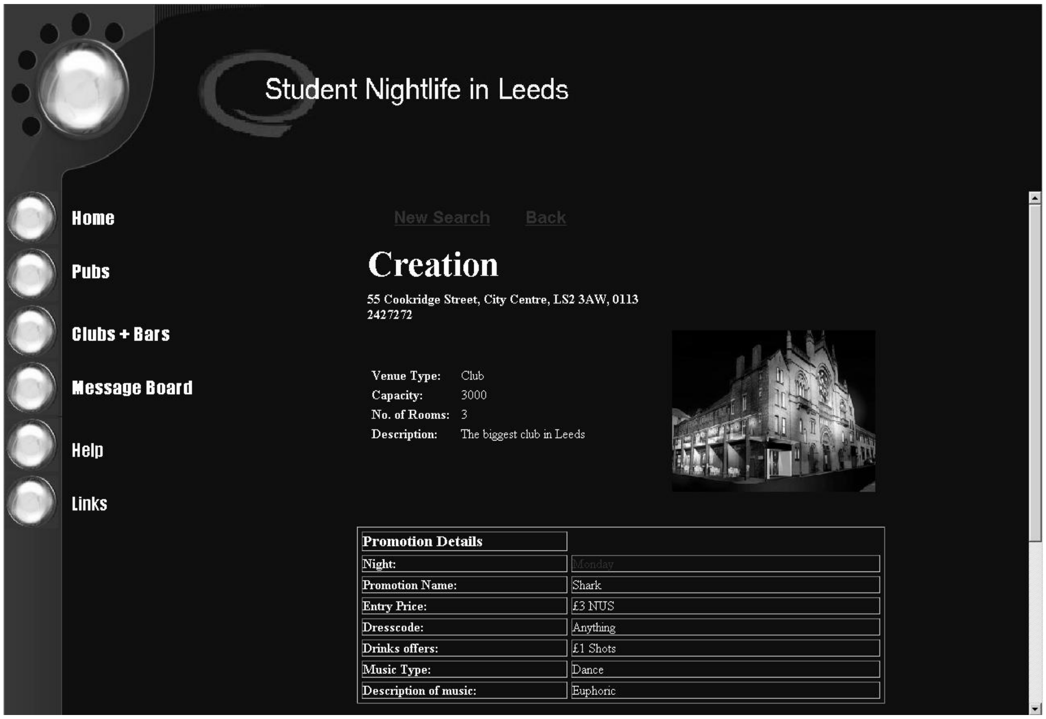


There is one Club matching your search

Search Results

Venue name	Description	Location
Creation	The biggest club in Leeds	City Centre

Club/Bar Detail Page



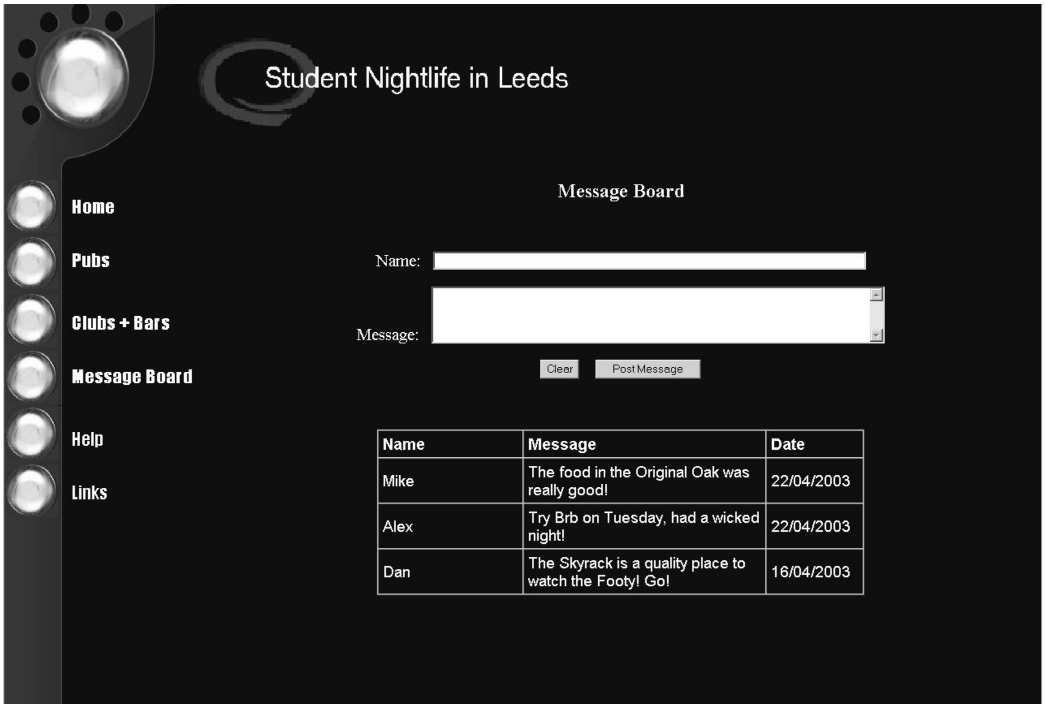
Creation

55 Cookridge Street, City Centre, LS2 3AW, 0113 2427272

Venue Type:	Club
Capacity:	3000
No. of Rooms:	3
Description:	The biggest club in Leeds

Promotion Details	
Night:	Monday
Promotion Name:	Shark
Entry Price:	£3 NUS
Dresscode:	Anything
Drinks offers:	£1 Shots
Music Type:	Dance
Description of music:	Euphoric

Message Board Page



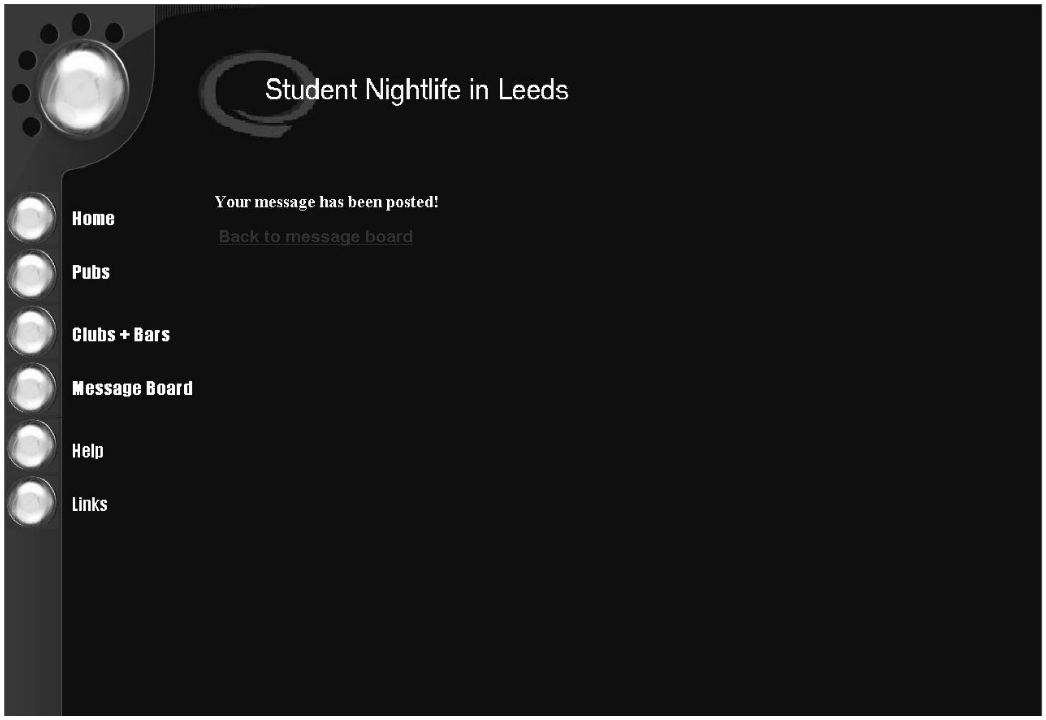
Message Board

Name:

Message:

Name	Message	Date
Mike	The food in the Original Oak was really good!	22/04/2003
Alex	Try Brb on Tuesday, had a wicked night!	22/04/2003
Dan	The Skyrack is a quality place to watch the Footy! Go!	16/04/2003

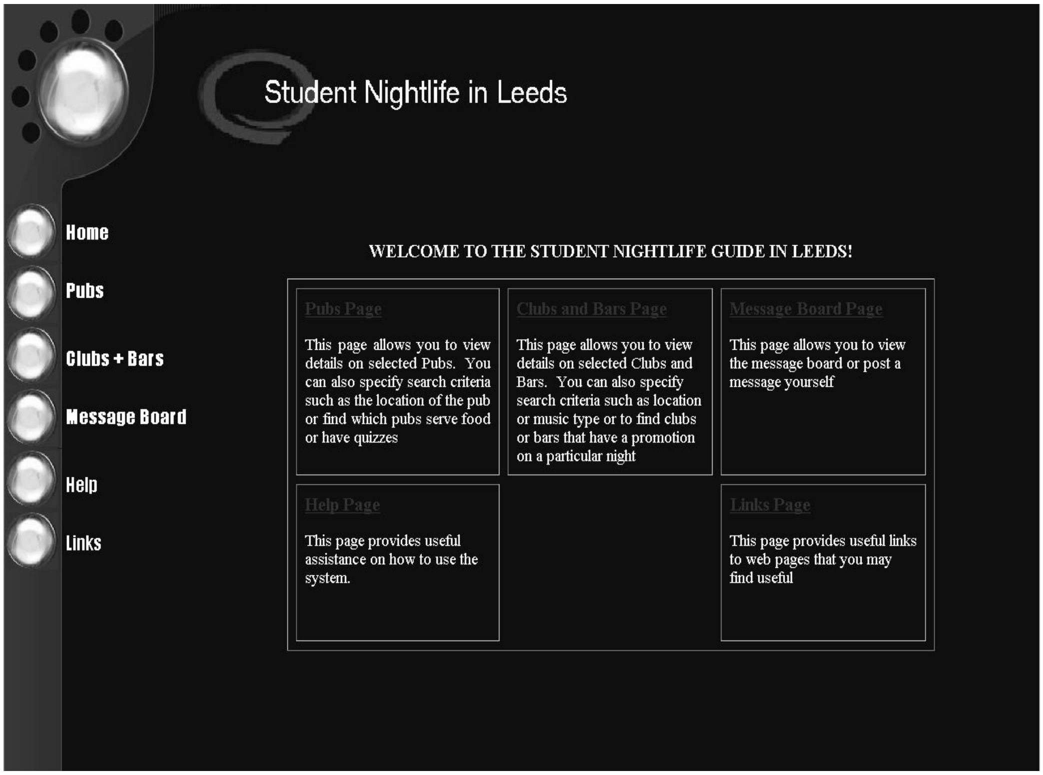
Message Post acknowledgement Page



Your message has been posted!

Back to message board

Home Page



WELCOME TO THE STUDENT NIGHTLIFE GUIDE IN LEEDS!

Pubs Page

This page allows you to view details on selected Pubs. You can also specify search criteria such as the location of the pub or find which pubs serve food or have quizzes

Clubs and Bars Page

This page allows you to view details on selected Clubs and Bars. You can also specify search criteria such as location or music type or to find clubs or bars that have a promotion on a particular night

Message Board Page

This page allows you to view the message board or post a message yourself

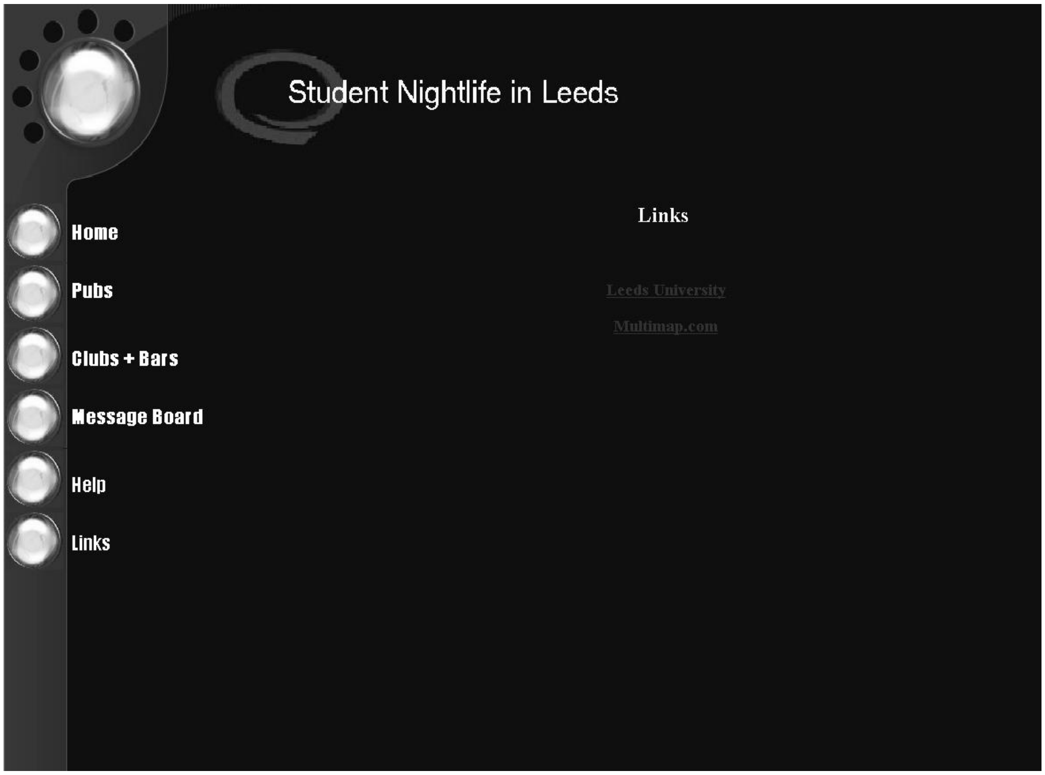
Help Page

This page provides useful assistance on how to use the system.

Links Page

This page provides useful links to web pages that you may find useful

Links Page

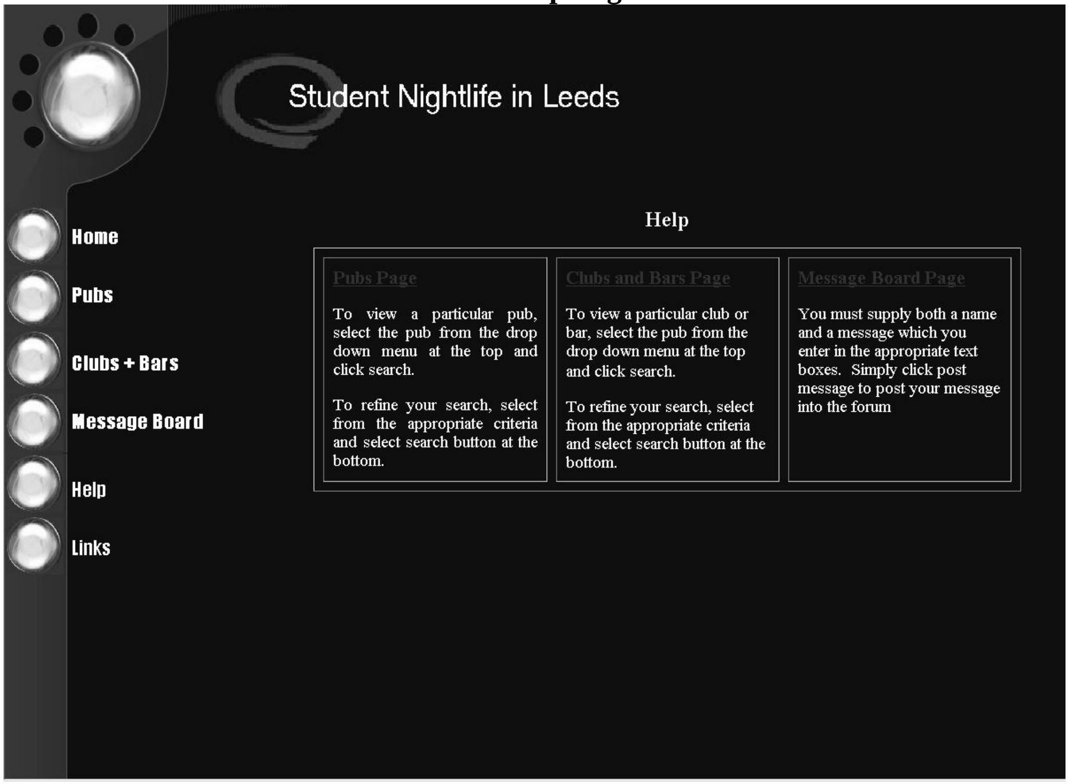


Links

Leeds University

Multimap.com

Help Page



Help

Pubs Page	Clubs and Bars Page	Message Board Page
To view a particular pub, select the pub from the drop down menu at the top and click search. To refine your search, select from the appropriate criteria and select search button at the bottom.	To view a particular club or bar, select the pub from the drop down menu at the top and click search. To refine your search, select from the appropriate criteria and select search button at the bottom.	You must supply both a name and a message which you enter in the appropriate text boxes. Simply click post message to post your message into the forum

Appendix J – Validation Testing

Testing Attribute "Tel_No" in "Pub" and "Venue" tables

ACTION	EXPECTED RESULT	ACTUAL RESULT
Enter Tel No 0113 2454545	Action ok	Same as expected
Enter Tel No 01132 245454	Action ok	Same as expected
Enter Tel No 011 22222222	Error message displayed	Same as expected
Enter Tel No 0113 24545454	User prevented from entering any Tel No not in the '(00009) 0000009 format.	Same as expected
Enter Tel No 011A 2454545	Error message displayed	Same as expected

Testing Attribute "Start_date"

ACTION	EXPECTED RESULT	ACTUAL RESULT
Enter Start_date "16/04/2003" in "Promotions" table when the current date is "16/04/2003"	Action ok	Same as expected
Enter Start_date "17/04/2003" in "Promotions" table when the current date is "16/04/2003"	Action ok	Same as expected
Enter Start_date "15/04/2003" in "Promotions" table when the current date is "16/04/2003"	Error message displayed on validation rule.	Same as expected
Enter Start_date "16/03/2003" in "Promotions" table when the current date is "16/04/2003"	Error message displayed on validation rule.	Same as expected

Testing Attribute "End_date"

ACTION	EXPECTED RESULT	ACTUAL RESULT
Enter End_date "16/04/2003" in "Promotions" table when the Start_date is "16/04/2003"	Action ok	Same as expected
Enter End_date "17/04/2003" in "Promotions" table when the Start_date is "16/04/2003"	Action ok	Same as expected
Enter End_date "15/04/2003" in "Promotions" table when the Start_date is "16/04/2003"	Error message displayed on validation rule.	Same as expected
Enter End_date "16/03/2003" in "Promotions" table when the Start_date is "16/04/2003"	Error message displayed on validation rule.	Same as expected

Testing Attribute "Finish_time" in "Pub_quiz" table

ACTION	EXPECTED RESULT	ACTUAL RESULT
Enter Finish_time "16:30" in "Pub_quiz" table when the Start_time is "11:30"	Action ok	Same as expected
Enter Finish_time "11:31" in "Pub_quiz" table when the Start_time is "11:30"	Action ok	Same as expected
Enter Finish_time "10:30" in "Pub_quiz" table when the Start_time is "11:30"	Error message displayed on validation rule.	Same as expected
Enter Finish_time"11:30" in "Pub_quiz" table when the Start_time is "11:30"	Error message displayed on validation rule.	Same as expected

Testing Attribute “Finish_time” in “Food” table

ACTION	**EXPECTED RESULT**	**ACTUAL RESULT**
Enter Finish_time “16:30” in “Food” table when the Start_time is “11:30”	Action ok	Same as expected
Enter Finish_time “11:31” in “Food” table when the Start_time is “11:30”	Action ok	Same as expected
Enter Finish_time “10:30” in “Food” table when the Start_time is “11:30”	Error message displayed on validation rule.	Same as expected
Enter Finish_time“11:30” in “Food” table when the Start_time is “11:30”	Error message displayed on validation rule.	Same as expected

Appendix K – Query testing Results in ASP

Pub search test results

ATTRIBUTE:	**SELECTION**
Location:	Any Location
Food Start time:	Any Time
Food Finish time:	Any Time
Food Day:	Any Day
Quiz Night:	Any Day
Widescreen TV:	(not selected)
Jukebox:	(not selected)
Pool table	(not selected)
Dartboard	(not selected)
Quiz machine	(not selected)
SQL Query	SELECT DISTINCT P.Pub_id, Pub_name, P.Description, P.Location, P.Pool_table, P.Quiz_machine, P.dartboard, P.Jukebox, P.Widescreen_TV, L.Location_id, L.Location , P.Address , P.Tel_No , P.Postcode , P.Image FROM ((Pub AS P INNER JOIN Location AS L ON L.Location_id = P.Location) LEFT JOIN Pub_quiz AS Q ON P.Pub_id = Q.Pub_id) LEFT JOIN Food AS F ON P.Pub_id = F.Pub_id WHERE ((P.Pub_name IS NOT NULL)) ORDER BY P.Pub_name
Expected result	There are 9 Pubs matching your search (selects all the pubs from the pub table)
Actual result	Same as expected

ATTRIBUTE:	**SELECTION**
Location:	Headingley
Food Start time:	Any Time
Food Finish time:	Any Time
Food Day:	Any Day
Quiz Night:	Any Day
Widescreen TV:	(not selected)
Jukebox:	(not selected)
Pool table	(not selected)
Dartboard	(not selected)

Quiz machine	(not selected)
SQL Query	SELECT DISTINCT P.Pub_id, Pub_name, P.Description, P.Location, P.Pool_table, P.Quiz_machine, P.dartboard, P.Jukebox, P.Widescreen_TV, L.Location_id, L.Location , P.Address , P.Tel_No , P.Postcode , P.Image FROM ((Pub AS P INNER JOIN Location AS L ON L.Location_id = P.Location) LEFT JOIN Pub_quiz AS Q ON P.Pub_id = Q.Pub_id) LEFT JOIN Food AS F ON P.Pub_id = F.Pub_id WHERE ((P.Pub_name IS NOT NULL) AND ((L.Location_id) LIKE '1')) ORDER BY P.Pub_name
Expected result	There are 8 Pubs matching your search
Actual result	Same as expected

ATTRIBUTE:	**SELECTION**
Location:	Headingley
Food Start time:	11:30
Food Finish time:	16:00
Food Day:	Any Day
Quiz Night:	Any Day
Widescreen TV:	(not selected)
Jukebox:	(not selected)
Pool table	(not selected)
Dartboard	(not selected)
Quiz machine	(not selected)
SQL Query	SELECT DISTINCT P.Pub_id, Pub_name, P.Description, P.Location, P.Pool_table, P.Quiz_machine, P.dartboard, P.Jukebox, P.Widescreen_TV, L.Location_id, L.Location , P.Address , F.Start_time , F.Finish_time , P.Image FROM ((Pub AS P INNER JOIN Location AS L ON L.Location_id = P.Location) LEFT JOIN Pub_quiz AS Q ON P.Pub_id = Q.Pub_id) LEFT JOIN Food AS F ON P.Pub_id = F.Pub_id WHERE ((P.Pub_name IS NOT NULL) AND ((L.Location_id) LIKE '1') AND ((F.Start_time) >= #11:30#) AND ((F.Finish_time) >= #16:00#)) ORDER BY P.Pub_name
Expected result	There are 4 Pubs matching your search
Actual result	Same as expected

ATTRIBUTE:	**SELECTION**
Location:	Corn Exchange
Food Start time:	11:30
Food Finish time:	16:00

Food Day:	Any Day
Quiz Night:	Any Day
Widescreen TV:	(not selected)
Jukebox:	(not selected)
Pool table	(not selected)
Dartboard	(not selected)
Quiz machine	(not selected)
SQL Query	SELECT DISTINCT P.Pub_id, Pub_name, P.Description, P.Location, P.Pool_table, P.Quiz_machine, P.dartboard, P.Jukebox, P.Widescreen_TV, L.Location_id, L.Location , P.Address , F.Start_time , F.Finish_time , P.Image FROM ((Pub AS P INNER JOIN Location AS L ON L.Location_id = P.Location) LEFT JOIN Pub_quiz AS Q ON P.Pub_id = Q.Pub_id) LEFT JOIN Food AS F ON P.Pub_id = F.Pub_id WHERE ((P.Pub_name IS NOT NULL) AND ((L.Location_id) LIKE '2') AND ((F.Start_time) >= #11:30#) AND ((F.Finish_time) >= #16:00#)) ORDER BY P.Pub_name
Expected result	There are no Pubs matching your search
Actual result	Same as expected

ATTRIBUTE:	**SELECTION**
Location:	Headingley
Food Start time:	11:30
Food Finish time:	16:00
Food Day:	Monday
Quiz Night:	Any Day
Widescreen TV:	(not selected)
Jukebox:	(not selected)
Pool table	(not selected)
Dartboard	(not selected)
Quiz machine	(not selected)
SQL Query	SELECT DISTINCT P.Pub_id, Pub_name, P.Description, P.Location, P.Pool_table, P.Quiz_machine, P.dartboard, P.Jukebox, P.Widescreen_TV, L.Location_id, L.Location , P.Address , F.Start_time , F.Finish_time , F.Night_id FROM ((Pub AS P INNER JOIN Location AS L ON L.Location_id = P.Location) LEFT JOIN Pub_quiz AS Q ON P.Pub_id = Q.Pub_id) LEFT JOIN Food AS F ON P.Pub_id = F.Pub_id WHERE ((P.Pub_name IS NOT NULL) AND ((L.Location_id) LIKE '1') AND ((F.Start_time) >= #11:30#) AND ((F.Finish_time) >= #16:00#) AND ((F.Night_id) LIKE 'Monday')) ORDER BY P.Pub_name

Expected result	There are 2 Pubs matching your search
Actual result	Same as expected

ATTRIBUTE:	**SELECTION**
Location:	Headingley
Food Start time:	11:30
Food Finish time:	16:00
Food Day:	Monday
Quiz Night:	Sunday
Widescreen TV:	(not selected)
Jukebox:	(not selected)
Pool table	(not selected)
Dartboard	(not selected)
Quiz machine	(not selected)
SQL Query	SELECT DISTINCT P.Pub_id, Pub_name, P.Description, P.Location, P.Pool_table, P.Quiz_machine, P.dartboard, P.Jukebox, P.Widescreen_TV, L.Location_id, L.Location , Q.Night_id , F.Start_time , F.Finish_time , F.Night_id FROM ((Pub AS P INNER JOIN Location AS L ON L.Location_id = P.Location) LEFT JOIN Pub_quiz AS Q ON P.Pub_id = Q.Pub_id) LEFT JOIN Food AS F ON P.Pub_id = F.Pub_id WHERE ((P.Pub_name IS NOT NULL) AND ((L.Location_id) LIKE '1') AND ((Q.Night_id) LIKE 'Sunday') AND ((F.Start_time) >= #11:30#) AND ((F.Finish_time) >= #16:00#) AND ((F.Night_id) LIKE 'Monday')) ORDER BY P.Pub_name
Expected result	There is one Pub matching your search
Actual result	Same as expected

ATTRIBUTE:	**SELECTION**
Location:	Headingley
Food Start time:	Any Time
Food Finish time:	Any Time
Food Day:	Any Day
Quiz Night:	Any Day
Widescreen TV:	(not selected)
Jukebox:	(not selected)
Pool table	(not selected)
Dartboard	Yes

Quiz machine	(not selected)
SQL Query	SELECT DISTINCT P.Pub_id, Pub_name, P.Description, P.Location, P.Pool_table, P.Quiz_machine, P.dartboard, P.Jukebox, P.Widescreen_TV, L.Location_id, L.Location , P.Address , P.Tel_No , P.Postcode , P.Image FROM ((Pub AS P INNER JOIN Location AS L ON L.Location_id = P.Location) LEFT JOIN Pub_quiz AS Q ON P.Pub_id = Q.Pub_id) LEFT JOIN Food AS F ON P.Pub_id = F.Pub_id WHERE ((P.Pub_name IS NOT NULL) AND ((L.Location_id) LIKE '1') AND ((P.Dartboard) LIKE ON)) ORDER BY P.Pub_name
Expected result	There is one Pub matching your search
Actual result	Same as expected

ATTRIBUTE:	**SELECTION**
Location:	Headingley
Food Start time:	Any Time
Food Finish time:	Any Time
Food Day:	Any Day
Quiz Night:	Any Day
Widescreen TV:	Yes
Jukebox:	Yes
Pool table	Yes
Dartboard	(not selected)
Quiz machine	(not selected)
SQL Query	SELECT DISTINCT P.Pub_id, Pub_name, P.Description, P.Location, P.Pool_table, P.Quiz_machine, P.dartboard, P.Jukebox, P.Widescreen_TV, L.Location_id, L.Location , P.Address , P.Tel_No , P.Postcode , P.Image FROM ((Pub AS P INNER JOIN Location AS L ON L.Location_id = P.Location) LEFT JOIN Pub_quiz AS Q ON P.Pub_id = Q.Pub_id) LEFT JOIN Food AS F ON P.Pub_id = F.Pub_id WHERE ((P.Pub_name IS NOT NULL) AND ((L.Location_id) LIKE '1') AND ((P.Pool_table) LIKE ON) AND ((P.Widescreen_TV) LIKE ON) AND ((P.Jukebox) LIKE ON)) ORDER BY P.Pub_name
Expected result	There are 5 Pubs matching your search
Actual result	Same as expected

ATTRIBUTE:	**SELECTION**
Location:	Headingley
Food Start time:	Any Time

Food Finish time:	Any Time
Food Day:	Any Day
Quiz Night:	Any Day
Widescreen TV:	Yes
Jukebox:	(not selected)
Pool table	Yes
Dartboard	(not selected)
Quiz machine	Yes
SQL Query	SELECT DISTINCT P.Pub_id, Pub_name, P.Description, P.Location, P.Pool_table, P.Quiz_machine, P.dartboard, P.Jukebox, P.Widescreen_TV, L.Location_id, L.Location , P.Address , P.Tel_No , P.Postcode , P.Image FROM ((Pub AS P INNER JOIN Location AS L ON L.Location_id = P.Location) LEFT JOIN Pub_quiz AS Q ON P.Pub_id = Q.Pub_id) LEFT JOIN Food AS F ON P.Pub_id = F.Pub_id WHERE ((P.Pub_name IS NOT NULL) AND ((L.Location_id) LIKE '1') AND ((P.Pool_table) LIKE ON) AND ((P.Widescreen_TV) LIKE ON) AND ((P.Quiz_machine) LIKE ON)) ORDER BY P.Pub_name
Expected result	There are 4 Pubs matching your search
Actual result	Same as expected

ATTRIBUTE:	**SELECTION**
Location:	Headingley
Food Start time:	Any
Food Finish time:	fnnngringior
Food Day:	Any Day
Quiz Night:	Any Day
Widescreen TV:	(not selected)
Jukebox:	(not selected)
Pool table	(not selected)
Dartboard	(not selected)
Quiz machine	(not selected)
SQL Query	SELECT DISTINCT P.Pub_id, Pub_name, P.Description, P.Location, P.Pool_table, P.Quiz_machine, P.dartboard, P.Jukebox, P.Widescreen_TV, L.Location_id, L.Location , P.Address , P.Tel_No , P.Postcode , P.Image FROM ((Pub AS P INNER JOIN Location AS L ON L.Location_id = P.Location) LEFT JOIN Pub_quiz AS Q ON P.Pub_id = Q.Pub_id) LEFT JOIN Food AS F ON P.Pub_id = F.Pub_id WHERE ((P.Pub_name IS NOT NULL) AND ((L.Location_id) LIKE '1') AND ((P.Pool_table) LIKE ON) AND ((P.Widescreen_TV) LIKE ON) AND ((P.Quiz_machine) LIKE ON))

	ORDER BY P.Pub_name
Expected result	There are no pubs matching your search
Actual result	Syntax error

Club/Bar Search test results

ATTRIBUTE:	**SELECTION**
Night:	Monday
Location:	Any Location
Venue Type:	Any
Music Type:	Any Music Type
SQL Query	SELECT *, P.Night_id, P.Music_id FROM (Venue V INNER JOIN Location L ON L.Location_id = V.Location) LEFT JOIN Promotions P ON V.Venue_id = P.Venue_id WHERE (((P.Night_id) LIKE 'Monday')) ORDER BY V.Venue_name
Expected result	There are 5 clubs/bars matching your search
Actual result	Same as expected

ATTRIBUTE:	**SELECTION**
Night:	Tuesday
Location:	Corn exchange
Venue Type:	Any
Music Type:	Any Music Type
SQL Query	SELECT *, P.Night_id, P.Music_id FROM (Venue V INNER JOIN Location L ON L.Location_id = V.Location) LEFT JOIN Promotions P ON V.Venue_id = P.Venue_id WHERE (((P.Night_id) LIKE 'Tuesday') AND ((L.Location_id) LIKE '2')) ORDER BY V.Venue_name
Expected result	There are 2 clubs/bars matching your search
Actual result	Same as expected

ATTRIBUTE:	**SELECTION**
Night:	Wednesday
Location:	Any Location
Venue Type:	Club
Music Type:	Any Music Type
SQL Query	SELECT *, P.Night_id, P.Music_id FROM (Venue V INNER JOIN Location L ON L.Location_id = V.Location) LEFT JOIN Promotions P ON V.Venue_id = P.Venue_id WHERE (((P.Night_id) LIKE 'Wednesday') AND ((V.Type) LIKE 'Club')) ORDER BY V.Venue_name
Expected result	There are 3 clubs matching your search
Actual result	Same as expected

ATTRIBUTE:	**SELECTION**
Night:	Thursday
Location:	Corn Exchange
Venue Type:	Bar
Music Type:	Funky House
SQL Query	SELECT *, P.Night_id, P.Music_id FROM (Venue V INNER JOIN Location L ON L.Location_id = V.Location) LEFT JOIN Promotions P ON V.Venue_id = P.Venue_id WHERE (((P.Night_id) LIKE 'Thursday') AND ((L.Location_id) LIKE '2') AND ((V.Type) LIKE 'Bar') AND ((P.Music_id) LIKE '2')) ORDER BY V.Venue_name
Expected result	There are 2 bars matching your search
Actual result	Same as expected

Appendix L – User Manual

Administrative Guide to the Leeds Student Nightlife System

USER GUIDE 2003

CONTENTS PAGE

1. Using the Main Menu

After logging onto the system, you are presented with the main menu shown in Figure 1.

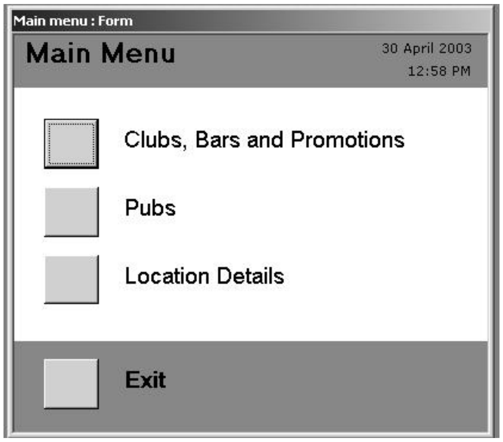


Figure 1: The Main menu

Clubs, Bars and Promotions – accesses a separate menu that allows you to add/delete/edit/search details of bars, clubs and their promotions.

Pubs – accesses a separate menu that allows you to add/delete/edit search details of Pubs.

Location Details – accesses a form that allows you to add/delete/edit Location areas.

Exit – Exits the system.

2. Using the Clubs, Bars and Promotions Menu

After selecting "Clubs, Bars and Promotions" from the main menu, you are presented with the following menu in figure 2.

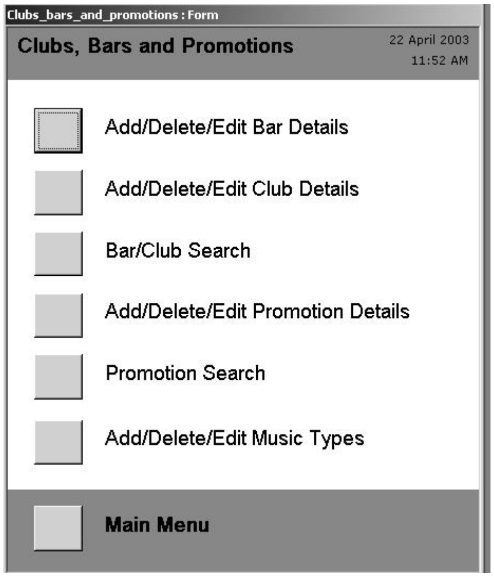
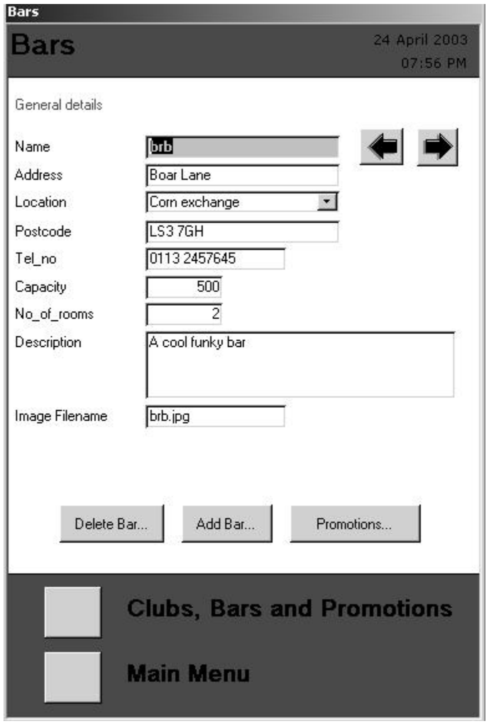

Clubs_bars_and_promotions : Form
Clubs, Bars and Promotions
22 April 2003
11:52 AM
Add/Delete/Edit Bar Details
Add/Delete/Edit Club Details
Bar/Club Search
Add/Delete/Edit Promotion Details
Promotion Search
Add/Delete/Edit Music Types
Main Menu

Figure 2: Clubs/Bars/Promotions

2.1 Using the Add/Delete/Edit Bar Details Form



Bars

Bars

24 April 2003
07:56 PM

General details

Name	brb
Address	Boar Lane
Location	Corn exchange
Postcode	LS3 7GH
Tel_no	0113 2457645
Capacity	500
No_of_rooms	2
Description	A cool funky bar
Image Filename	brb.jpg

Figure 3: Bars Form

2.1.1 Adding a Bar

a) To add a Bar into the database, simply click the 'Add Bar…' Button

b) Enter the details in the blank fields shown in figure 3. (In the case of an ▾ on a field, you are required to select the field entry from the drop down menu.

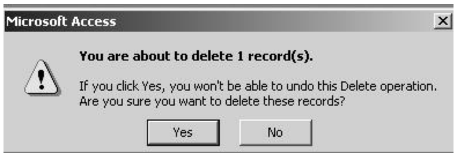
c) If you have an image of the Bar, Place the file in the same directory as this Database System. Then, in the 'Image' field, enter the filename of the Bar image.

d) Click "Update Bar details" button to enter your bar.

2.1.2 Deleting a Bar

a) To Delete a Bar from the database, simply click the 'Delete Bar…' Button

b) Click 'Yes' to delete the record or 'No' to cancel the action. Shown in figure 4.



Microsoft Access

You are about to delete 1 record(s).

If you click Yes, you won't be able to undo this Delete operation.
Are you sure you want to delete these records?

Figure 4: Deleting a Record

2.1.3 Adding a Promotion for a Bar

a) To add a Promotion for a particular Bar, simply click the 'Promotions…" Button.

b) When presented with the Promotions form, select the required night of the week from the drop down menu. Shown in figure 5.

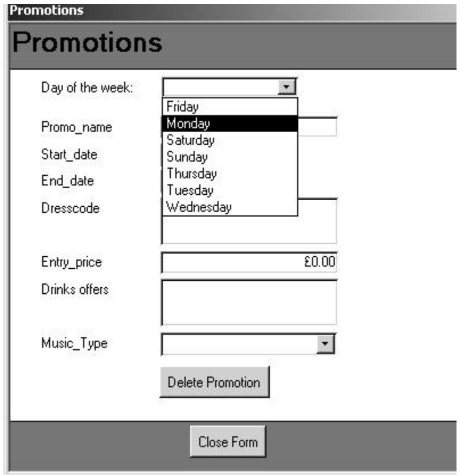


Figure 5: Promotions Form

c) Once the desired night of the week is selected, the remainder of the empty fields can be filled in to complete the promotion form.
d) Click 'Close Form' to return to the 'Bars' Form.

2.1.4 Deleting a Promotion from a Bar

a) Click the 'Promotions…' Button on the form.
b) Select the required night of the week from the drop down menu.
c) Click the 'Delete Promotion' Button
d) Confirm the deletion by selecting 'Yes' in a message box like the one shown in figure 4.

2.1.5 Editing a Promotion for a Bar

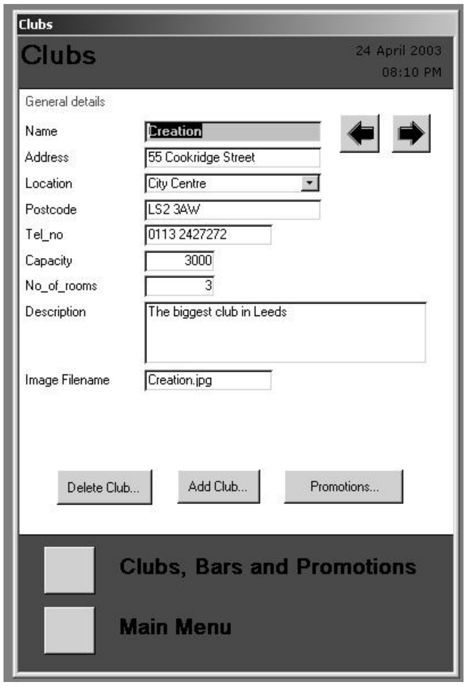
a) Click the 'Promotions…' Button on the form.
b) Select the required night of the week from the drop down menu.
c) Edit the appropriate fields in the form.
d) Click the 'Close Form' Button to return to the Bar form.

2.1.6 Navigating to Other Menus

a) Click 'Clubs, Bars and Promotions' at the bottom of the Form to return to the previous Menu.
b) Click 'Main Menu' at the bottom of the Form to return to the Main menu.

2.2 Using the Add/Delete/Edit Club Details Form

Please refer to section 2.1 on 'Using the Add/Delete/Edit Bar Details Form'. The operations carried out are executed in the same way as Bars.



Clubs

Clubs

24 April 2003
08:10 PM

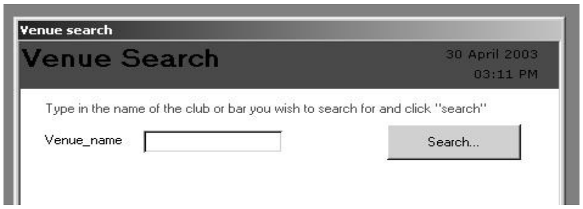
General details

Name	Creation
Address	55 Cookridge Street
Location	City Centre
Postcode	LS2 3AW
Tel_no	0113 2427272
Capacity	3000
No_of_rooms	3
Description	The biggest club in Leeds
Image Filename	Creation.jpg

Figure 6: Clubs Form

2.3 Using the Bar/Club Search Form

a) Enter the name of the Club or Bar u wish to search for in the text box shown in figure 7.



Venue search

Venue Search

30 April 2003
03:11 PM

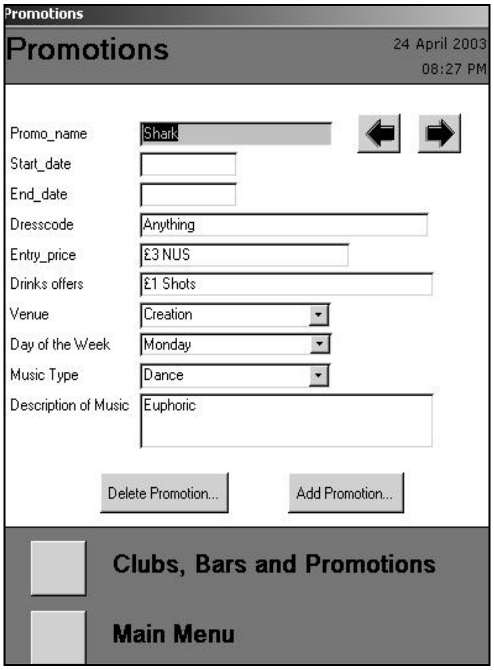
Type in the name of the club or bar you wish to search for and click "search"

Venue_name

Figure 7: Club/Bar Search Form

b) Click the “Search” button to display the details of your requested venue.
c) To edit any details in the search result, simply click on the required fields and change the information in them accordingly.

2.4 Using the Add/Delete/Edit Promotion Details Form


Promotions

Promotions

24 April 2003
08:27 PM

Promo_name	Shark
Start_date	
End_date	
Dresscode	Anything
Entry_price	£3 NUS
Drinks offers	£1 Shots
Venue	Creation
Day of the Week	Monday
Music Type	Dance
Description of Music	Euphoric

Figure 8: Promotions Form

2.4.1 Adding a Promotion

a) To add a promotion, simply click the ‘Add Promotion…’ Button
b) Enter the details in the empty fields provided by the form. (Please note a Venue and a night of the week must be selected from the drop down menus or you will receive an error message.
c) Click “Update Promotion” to enter the promotion.

2.4.2 Deleting a Promotion

a) To delete a Promotion, simply click the ‘Delete Promotion..’ Button.

b) To confirm the deletion select ‘Yes’ from the message box shown in figure 4.

2.4.3 Changing Venue for a promotion

a) Whilst viewing the required promotion, select a new Venue from the drop down menu. This can be seen in figure 9.

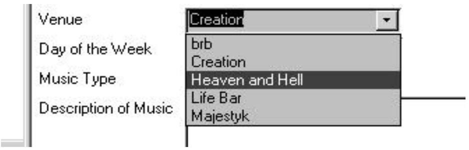

Venue: Creation

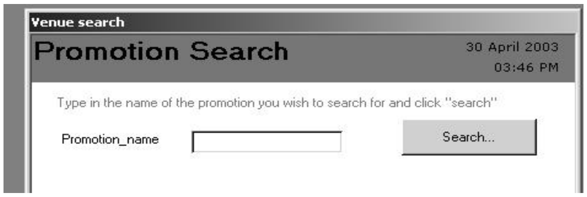
Figure 9: Venue Select

b) Selecting this sets the displayed promotion to the new venue. If a Club or Bar that you select already has a promotion on the selected night, then you will be provided with an appropriate error message.

c) Any details of the particular promotion can also be edited by modifying the required fields.

2.5 Using the Promotion Search Form

1. Enter the name of the Promotion you wish to search for in the text box shown in figure 10.



Type in the name of the promotion you wish to search for and click "search"

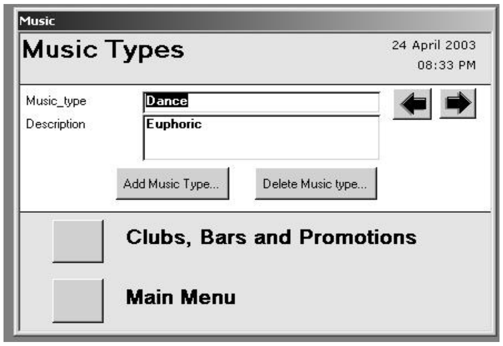
Promotion_name

Figure 10: Promotion Search Form

2. Click the "Search…" Button to find details on your requested Promotion.
3. To edit any details in the search result, simply click on the required fields and change the information in them accordingly.

2.6 Using the Add/Delete/Edit Music Types Form

The arrows provided allow you to scroll through the current music types that exist in the database. Should there be a music type that you feel should be available, then simply click "Add Music Type…" and enter the name and the description.
If you feel that a music type is no longer required then you can remove it by clicking the button "Delete music Type…"



Music_type: Dance
Description: Euphoric

Figure 11: Music Types Form

3.Using the Pubs Menu

After selecting "Pubs" from the main menu, you are presented with the following menu in figure 12.

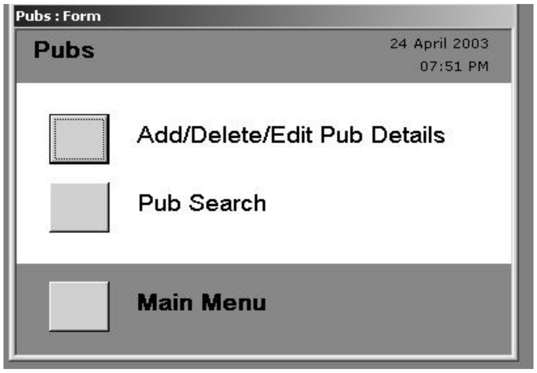
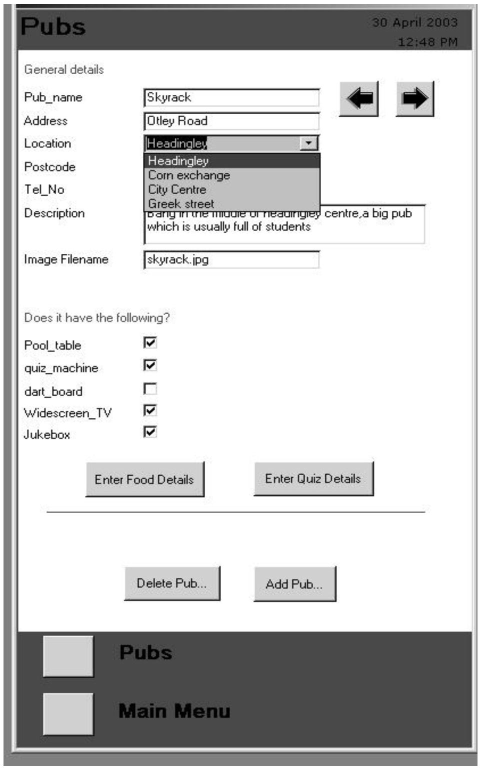


Figure 12: The Pubs Menu

3.1 Using the Add/Delete/Edit Pub Details Form



General details

Pub_name	Skyrack
Address	Otley Road
Location	Headingley ▾
Postcode	
Tel_No	
Description	Bang in the middle of headingley centre,a big pub which is usually full of students
Image Filename	skyrack.jpg

Headingley
Corn exchange
City Centre
Greek street

Does it have the following?

Pool_table	☑
quiz_machine	☑
dart_board	□
Widescreen_TV	☑
Jukebox	☑

Figure 13: Pubs Form

3.1.1 Adding a Pub

a) To add a Pub into the database, simply click the 'Add Pub…' Button

b) Enter the details in the blank fields shown in figure 13. (In the case of an ▼ on a field, you are required to select the field entry from the drop down menu.

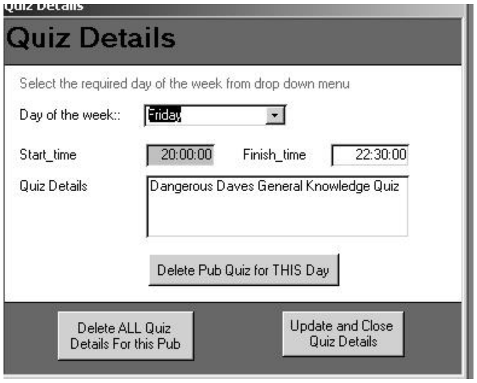
c) If you have an image of the Pub, Place the file in the same directory as this Database System. Then, in the 'Image' field, enter the filename of the Pub image.

d) Click “Update Pub details” button to enter your Pub.

3.1.2 Adding Quiz Details for a Pub

a) To add a Pub quiz for a pub, simply click the ‘Enter Quiz Details…” button on the required pub form. This provides you with a separate interface for entering the details that is shown in figure 14.

b) Select the desired day of the week from the drop down menu.


Quiz Details

Select the required day of the week from drop down menu

Day of the week:: Friday

Start_time 20:00:00 Finish_time 22:30:00

Quiz Details Dangerous Daves General Knowledge Quiz

Quiz Details
Delete Pub Quiz for THIS Day
Delete ALL Quiz Details For this Pub
Update and Close Quiz Details

Figure 14: Quiz Details Form

c) Enter the details along with the start time and finish time of the quiz.

d) Click the “Update and Close Quiz Details” button to register your pub quiz

3.1.3 Deleting Quiz Details for a Pub

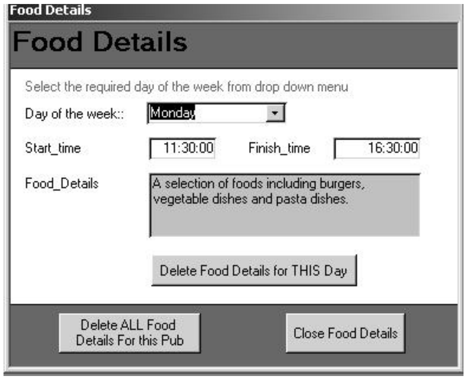
a) To delete a Pub quiz for a certain day, select the day of the week from the drop down menu and click the “Delete Pub for THIS Day” button.

b) To delete all Pub quizzes for the particular Pub, click the “Delete ALL Quiz Details For this Pub” button.

3.1.4 Adding Food Details for a Pub

a) To add a Pub quiz for a pub, simply click the ‘Enter Food Details…” button on the required pub form. This provides you with a separate interface for entering the details that is shown in figure 15.

b) Select the desired day of the week from the drop down menu.



Food Details

Select the required day of the week from drop down menu

Day of the week:: Monday

Start_time 11:30:00 Finish_time 16:30:00

Food_Details A selection of foods including burgers, vegetable dishes and pasta dishes.

Food Details
Delete Food Details for THIS Day
Delete ALL Food Details For this Pub
Close Food Details

Figure 15: Food Details Form

c) Enter the details along with the start time and finish time of the Food day.

d) Click the "Update and Close Food Details" button to register your Food day.

3.1.5 Deleting Food Details for a Pub

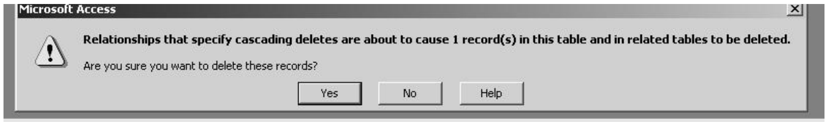
a) To delete a certain Food day, select the day of the week from the drop down menu and click the "Delete Pub for THIS Day" button.

b) To delete all Food details for a particular pub, click the "Delete ALL Food Details For this Pub" button.

3.1.6 Deleting a Pub

a) To delete a Pub from the database, simply click the "Delete Pub…" button.

b) If a Pub has related pub quizzes or food details then you will be prompted with a message box asking you if you are sure you want to delete. This is shown in figure 16.



Microsoft Access

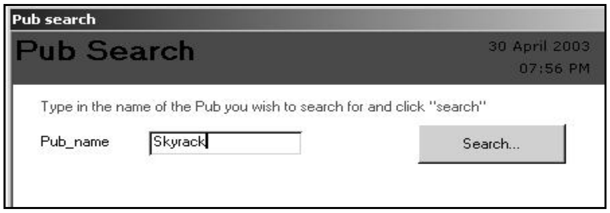
Relationships that specify cascading deletes are about to cause 1 record(s) in this table and in related tables to be deleted.

Are you sure you want to delete these records?

Figure 16: Deleting a record

3.2 Using the Pub Search Form

a) Enter the name of the Pub you wish to search for in the text box shown in figure 18.



Pub search

Pub Search

30 April 2003
07:56 PM

Type in the name of the Pub you wish to search for and click "search"

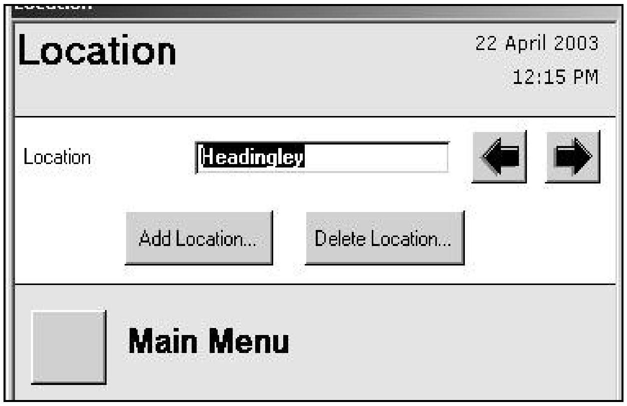
Pub_name Skyrack

Figure 16: Searching for a Pub

b) Click the "Search" button to display details of your requested Pub.
c) To edit any details in the search result, simply click on the required fields and edit them accordingly.

4. <u>Using the Location Details Form</u>

This form allows you to scroll through the different locations using the arrows on the form. Should there be a time when a new are of Leeds is introduced to the city, then you can compensate for that by clicking the "Add new Location..." button. Should a location cease to exist, you can remove it by clicking the "Delete Location…". This has the effect of deleting any pubs, clubs and bars that have this location.



Location

22 April 2003
12:15 PM

Location: Headingley

Figure 17: Location Form

Appendix M – Sample ASP code

The following few pages provide the asp code for Clubs/bar page. This includes the search page, the result page, and the detail page.