Database Driven System for Urban Lifestyle Letting

Mark Brogan
JHiS IS/MS
2004/2005)

Summary

During Easter 2004, Urban Lifestyle Letting set up a basic, static website in order to have some sort of Internet focal point. The website contained the companies contact details and had a separate webpage for every property that was available for rent. The main problem that was noticeable was that none of the staff at ULL had any knowledge of HTML and web development. A custom made solution had to be created as the user's requirements were not supported by any existing solutions.

Urban Lifestyle Letting wanted a dynamic web-based system which could be easily updated through a password protected administration area. Furthermore, Urban Lifestyle Letting also wanted a number of message boards that would allow tenants to post requests. This would allow members of staff to answer the queries in quiet moments during the day in order to reduce the number of telephone calls happening during peak periods.

In order to plan the development of the system, requirements were gathered from the end user through a number of meetings and telephone calls. A vast number of tools and methodologies were analysed in order to see which ones were best suited to the development of the system. The system was thoroughly designed, implemented and tested in order to produce a high quality solution. Finally, the system was evaluated to determine its success and how well it met the needs of the end user.

Acknowledgements

This project would not have been a success without the support from a number of individuals and for that reason I would like to show my appreciation to the following:

Brandon Bennett my Project Supervisor, who offered copious amounts of priceless suggestions, continuous positive support through weekly meetings and help in keeping me on track during difficult times.

Stuart Roberts my Project Assessor, who provided invaluable, detailed feedback following both mid-project report and progress meeting respectively. This enabled me to effectively build on my weaknesses and promote my strengths

The eight individuals who took part in my focus group, providing insightful opinions and criticisms alike, contributing to the evaluation of the solution.

All the employees at Urban Lifestyle Letting, for showing so much faith and allowing me to attain so much valuable experience from working on their system.

My three housemates, for providing methods of escapism that were contributing factors into the maintaining of my sanity.

Lucy, for her continuous support and the cooking of many meals following a hard day's work at university.

TABLE OF CONTENTS

Online Database Driven System for Urban Lifestyle Letting

CHAPTER 1 – INTRODUCTION

1.1 – Introduction to the company

Urban Lifestyle Letting (ULL) is a letting agency based in Bury, Greater Manchester. The company manages all types of property throughout Greater Manchester and prides itself on professionalism, friendliness and attention to detail. ULL is a relatively new company and was established in the summer of 2003. Currently, the company manages and rents out over 100+ properties and hope to optimize this number in order to evolve the company to the next level.

ULL is an important, intricate sector of the company founder’s property business. The company Director has another business called Northern Abodes (http://www.sell-quick.co.uk) that buys property for cash from customers that need to sell immediately for one reason or another. Furthermore, many customers who sell their home to Northern Abodes do not wish to move out into alternative accommodation. Therefore, Northern Abodes buys the property from the customer and rents it straight back to them using ULL. Additionally, many of ULL’s tenants express their wish to move out of rented accommodation and make the step to move onto the property ladder. Northern Abodes continuously buy properties for knock-down prices and can offer these properties to ULL’s tenants.

1.2 – Problem Definition

“…figures published by the UK Association of Business Recovery Professionals (R3, 2003) show that Manchester has the second greatest percentage (3.51%) of corporate insolvencies in the UK…” [1]

These figures show that ULL are participating in a competitive market and that the company needs to continue to evolve in order to find that competitive advantage. ULL have identified the Internet as the market in which they intend to target in order to progress to the next level. During Easter 2004, the company set up a basic, static website in order to have some sort of Internet focal point. The website contained the companies contact details and had a separate webpage for every property that was available for rent. The main problem that was noticeable was that none of the staff at ULL had any knowledge of HTML and web development and therefore they could not make any changes to the website.

Another problem with the existing website was that nobody really knew it existed because it was not listed in any of the major search engines. The only way people found out about the website was by word of mouth and because the URL appeared in a couple of online property directories.

Finally, the current website uses flash buttons and other flash content. On many web browsers and older computers the website is completely unusable. This can be seen in the screen capture found in Appendix B.

1.3 – Project Aim

Produce a fully working online database system for the staff at Urban Lifestyle Letting. Currently, the website that the company uses is static and requires dynamic content so that the staff can easily manage the system without having any knowledge of HTML. Also, adding dynamic content will allow potential tenants to search for wanted properties far more efficiently.

1.3.1 – Risk and Mitigation

Possible threats to the success of this project have been analysed and a number of relevant mitigations have been created. This will ensure that if and when certain problems occur, panic does not set in and the most appropriate action is taken. The Risks and Mitigations can be seen in Appendix B.

1.4 – Project Objectives

The following objectives are the anticipated outcomes and benefits following the completion of this project:

- Analyse what is required and needed by Urban Lifestyle Letting in order to produce a valuable and profitable web-based system.
- To produce the web-based application to be used by Urban Lifestyle Letting. The web-based system will allow customers:
    - To be able to find property listings that closely meets their requirements.
    - Make requests via a private message board.

  The web-based system will allow staff:
    - To be able to update the website and database contents online without any technical knowledge needed.

1.5.1 – Minimum Functional Requirements

The following minimum requirements are essential in order for the project to be a success:

1. Review and redesign the existing static HTML website.

2. Design and creation of a database containing details of all the houses that the company manages.
3. Produce an application that allows staff to update the website content online and enables customers to search for property.
4. Create user documentation instructing the staff how to use the system.
5. Evaluate the application using a number of methods.

1.5.2 – Non-functional Requirements

Cysneiros (2004) [3] believes that non-functional requirements are "…frequently neglected or forgotten in software design." It is imperative not to overlook non-functional requirements for the implementation of program functions. There is great importance in bridging the divide between the technology and the user when developing software [4]. Non-functional requirements can be successfully integrated into many UML (Unified Modelling Language) models such as Class, Sequence and Collaboration Diagrams.

By accepting and successfully implementing both functional and non-functional requirements, the software developer will have most likely produced a fully-working integrated system that the users understand and supports their day-to-day jobs.

<u>Usability of the system</u>

- The interface should be attractive, consistent and well-designed.
- Basic, easily readable fonts and text sizes should be used along with bold attractive colours.
- Must be easy to use and must only require basic I.T. skills.
- There should be logical and simple processes with reoccurring themes in order not to complicate the user.
- Data entry should be automised where appropriate.
- Fields should be hidden when they are not required.

<u>Speed and reliability of the system</u>

- The system should have a high performance and there should be little or no time delay when processing data.
- Throughout the system there must be no major functionality errors i.e. System not working correctly.
- Minor functionality errors must be kept to a minimum; otherwise the user will lose faith and become disillusioned with the system.
- The system must be tested vigorously so as to reduce the potential for a system crash to an absolute minimum. Data corruption/loss must be avoided at all costs.

Security of the system

- The system needs to be protected against unauthorised access using passwords.
- Generating data backups is essential and should be easy and quick to do. An appropriate disaster/recovery plan should be included in the user-manual.

1.6 – Project Extensions

There are many areas where a web-based system can be extended. It is important to prioritise the enhancements so that the most important ones are completed first and any luxury developments are left to the very end. Following an interview with the end-user I was able to obtain a list of features that could be added to the system and how important they were. Below are possible areas that can be investigated and pursued when the minimum requirements have been fully completed:

NB. Rank 1 = Top Priority and Rank 13 = Lowest Priority

Rank	**Description**	**Complexity**
1	Allow the administrators to alter the contents of the database online.	**Moderate/Complicated**
2	Introduce a section on the website for existing tenants, where each tenant has their own personal message board to communicate with Urban Lifestyle Letting.	**Complicated**
3	Introduce a basic area on the website for the administrators (staff at Urban Lifestyle Letting).	**Easy**
4	Design and create a prototype of an AI automated Live Help feature.	**Complicated**
5	Have the ability to store and retrieve details of current and past tenants online.	**Moderate/Complicated**
6	Investigate into search engine optimization (SEO) and optimize the website.	**Moderate**
7	Allow the administrators to backup and restore the database online.	**Complicated**
8	Allow the staff at Urban Lifestyle Letting to be able to print (or output to an Excel file) out a prioritized list of requests made by tenants.	**Complicated**
9	Implement an auto-responder when a customer sends an email.	**Moderate**
10	Add a basic FAQ section for the existing tenants.	**Easy**
11	Review the website's usability (in further detail).	**Moderate**
12	Host the website on the Internet.	**Easy**
13	Consider possible security flaws in the system (in further detail).	**Complicated**
14	Make relevant amendments to the website so that it complies as closely as possible with the World Wide Web Consortium's (W3C) [5] regulations and is compatible with the major web browsers.	**Moderate**

1.7 – Deliverables

The following are the three major deliverables that will be produced when the project has been completed:

1. The final report of the project.
2. A fully working web-based system that at the very least meets the minimum requirements previously stated.
3. A fully documented user-manual.

1.8 – Project Schedule

It is imperative to organise and plan ahead when beginning a project of this size and beyond in order to ensure that everything is completed on time. With any project, problems arise unexpectedly. One must make allowances for these by setting aside time or increasing proposed scheduled time on certain problematic areas, such as the implementation and project collation.

1.8.1 –Initial Schedule

No.	Activity	Date: from – to	Milestone/Major Deliverable
1	**Submit Project Preference Form**	**23/09/04 – 30/09/04**	✓
2	Discuss idea with supervisor	03/10/04	
3	Research	04/10/04 – 20/04/05	
4	**Submit Aims and Minimum Requirements**	**10/10/04 – 22.10.04**	✓
5	Interview with the end-user	15/10/04	
6	Compose user-requirements	16/10/04 – 27/10/04	
7	Review appropriate technologies and methodologies	10/11/04 – 05/12/04	
8	**Submit Mid-Project Report**	**01/12/04 – 10.12.04**	✓
9	End of Semester 1 - Christmas Holiday – Exam Period	10/12/04 – 21/01/05	
10	Design the system to be created using the iterative approach continuously consulting with the	11/12/04 – 28/01/05	

	end-user		
11	Prepare materials for the focus group	22/01/05 – 23/01/05	
12	Conduct focus group	23/01/05	
13	Consultation with the end-user	24/01/05	
14	Finalise design and agree on usability	29/01/05	
15	Start developing the Minimum Requirements of the system	01/02/05 – 27/02/05	
16	Start developing the Project Extensions of the system	27/02/05 – 17/03/05	
17	**Submit table of contents and draft chapter**	**03/03/05 – 11/03/05**	✓
18	**Prepare and conduct the progress meeting**	**14/03/05 – 18/03/05**	✓
19	Begin evaluating the system	19/03/05 – 26/03/05	
20	Complete all development and evaluation aspects	19/03/05 – 04/04/05	
21	OVER-FLOW	04/04/05 – 20/04/05	
22	Proof-read and collate the final report.	20/04/05 – 26/04/05	
23	**Submit final report**	**27/04/05**	✓

CHAPTER 2 – BACKGROUND RESEARCH

Throughout this chapter all major feasible tools and technologies are to be considered for the implementation of the system. It is imperative that the chosen tools are capable of producing an efficient and effective solution that meets the product requirements. Furthermore, the later stages of this chapter are dedicated to evaluating possible methodologies required for the product development.

Open-source tools are now beginning to grab the attention of many large and small businesses in the world today. It is important to seriously consider open-source tools in order to produce the final product.

"Analysts say that in the move to cut costs and drive efficiencies, enterprise users are starting to get more serious about open source products. A Forrester Research survey of 140 North American firms earlier this year found that 46% of respondents are using open source software today, and 14% have plans to bring open source into their data centers in the future.” [6]

2.1 – HTML Dreamweaver MX 2004

Dreamweaver MX 2004 is the website authoring tool that will be used to design and produce the final product following this project. Dreamweaver MX has many excellent design features supplemented by an easy to use FTP user-interface that makes it a must for this project.

“Dreamweaver has become an indispensable part of any Web-site developer's tool kit, and Dreamweaver MX 2004 is the latest incarnation of Macromedia's powerful Web-site-development tool.” [7]

2.2 – Database Management Systems (DBMS)

2.2.1 – Introduction

When storing information on a Web Server data can be easily stored in files, but this is only suitable for simple files. Furthermore, many problems occur when using simple files on the web. For example, if one user is writing data to a file and another user is downloading data from same file the data becomes in consistent (lost update problem). It is far better to use a database in order to store complex data.

Four Database Management Systems, all available on the SOC machines, were considered in this section where each was judged on its performance and characteristics: MySQL, Microsoft Access, SQL Server 2000 and PostgreSQL.

2.2.2 – MySQL

MySQL is a hugely popular Database Management System that vast amounts of people use for their web databases. It is noticeable that MySQL is often used with PHP as the development teams for the two products work closely together in order for the products to interact effectively.

“It is an industrial-strength, extremely scalable database with strong Linux and Windows support, a proven track record with large organizations, and good SQL support.”[8]

There are many features of MySQL that makes it attractive for web based systems, such as the following:

<u>It costs nothing.</u> MySQL is made available under the GNU Public License, although a commercial license can be purchased that contains support documentation.

It is platform independent. MySQL works on both Windows and Linux machines giving the user more choice.
It is commonly offered. Most web hosting companies (such as http://www.streamline.net) offer MySQL.

Although MySQL requires some technical database expertise, due to the command-line interface, there are a number of GUI administration tools [9] that allow the developer to make alterations to the database.

2.2.3 – MS Access
Microsoft Access is the best selling [10] DBMS in the world as it is included in the Microsoft Office package. Access has the ability to insert, edit, index and retrieve data through custom-made forms. Microsoft Access is a database software package. It allows the creation of interfaces and forms from which the user can manipulate data that is held in the database. An in-built wizard makes creating queries and reports easy and the validation commands, which include masking, helps make data validation a very simple task.

With the product being widely used, its place in the future is secured. With expandability in mind this is great as other people apart from those who initially designed it would be able to modify if needed. As a Microsoft product, its design is the same as all Microsoft products and therefore an immediate familiarity can be seen giving the user confidence.

2.2.4 – MS SQL Server 2000
Microsoft SQL Server is a Relational Database Management System (RDBMS) used to build business applications that require enhanced data protection. Moreover, superior performance when accessing and retrieving data is provided by the SQL server which acts as an "engine".

MS SQL Server is accompanied by excellent documentation and support combined with exceptional usability makes it a choice for many.

"According to a study conducted by Ipsos-Reid, SQL Server is installed at 53 per cent of medium and large organizations..." [11]
Subsequently, the main and palpable downside of MS SQL Server is the cost. For an SME (small to medium sized enterprise) to pay **£1,098.63** Inc VAT [12] in order to produce a web based system is a lot to ask. The quote above explains that MS SQL Server is a popular and powerful solution, but not cost-effective for a relatively small business. Although MS SQL Server offers significantly more features, it lacks the lower price that MySQL has to offer.

2.2.5 – PostgreSQL

PostgreSQL is a sophisticated, object-relational, open-source database management system that is predominantly Unix-based. PostgreSQL is supplied under the liberal BSD license meaning that it can be obtained free of charge whether it be for academic or commercial use. There are many modern features that are included in PostgreSQL such as:

- foreign keys
- complex queries
- triggers
- views
- transactional integrity
- multi-version concurrency control [13]

However, there are limitations when using open-source software such as PostgreSQL in a business environment. The lack of commercial infrastructure makes it difficult if problems arise. It would be extremely expensive and difficult to find an expert PostgreSQL programmer, as they are in limited numbers.

"If there's a problem, we say, `Give me an Oracle consultant: Then we give them $250 an hour to fix our problems. We just can't do that with a PostgreSQL person..."[14]

2.2.6 – Conclude and Evaluate

MySQL was ultimately chosen as the DBMS to be used in the production of the online database system for Urban Lifestyle Letting. The fact that Urban Lifestyle Letting is a relatively small business and that a high performance, high cost DBMS like Microsoft SQL Server would not be cost-effective. PostgreSQL depends too much on a Linux operating system and as the system is to be implemented using the Windows XP operating system it is decided that this solution may not be appropriate. It is notable that the developer of the system has no knowledge of PostgreSQL which has greatly affected the decision.

2.3 – Server-Side Scripting Languages

2.3.1 – Introduction

Many people still connect to the Internet using a dial-up modem and therefore the importance for an individual to be able load web pages as quickly as possible is a major factor. Client-side scripting is a good way to take strain off the web server. However, if many people are being turned away from a website because too many processes are being executed on their machine may result in someone not having as many customers as they could have had.

2.3.2 – PHP

PHP (Hypertext Preprocessor) is an open-source, server-side HTML embedded scripting language that is used to create dynamic web pages. The fact that PHP is open-source has resulted in the language being quickly developed by a community of followers and is widely offered on most web hosting providers.
“With more than 15 million Web sites running PHP, the language is the hottest thing to hit hosting since AOL.” [15]

PHP is very powerful as it can do anything that is possible with CGI scripts but most importantly it is compatible with many different kinds of databases and can communicate on many networks using HTTP, IMAP, SNMP, NNTP and POP. PHP code can be embedded within tags in a standard HTML file, which benefits the programmer as he/she can easily switch between HTML and PHP without having to output large amounts of HTML. Furthermore, as the PHP code is executed on the web server the visitor cannot view the code and therefore individuals who wish to copy the code are thwarted. The data to be installed in the system would not be desirable for any hacker to attempt to hack into the system. For that reason, PHP’s security problems can be overlooked at this moment in time.

2.3.3 – ASP

In 1997 Active Server Pages (ASP) was introduced by Microsoft as an alternative to CGI scripts. ASP was placed on the market to aid the creation of dynamic web content, mainly when interacting with databases. The majority of the code is written in VB Script and is executed on the web server. ASP is a very popular tool and there are many reasons for this and not just because it is supported by the biggest software developer in the world.

2.3.4 – CGI

Common Gateway Interface (CGI) is a set of rules for transporting data between the web server and a CGI program. The CGI program, which can be written in any major programming language such as Java, Perl or Visual Basic, is any program that is created to accept and return data that obey the rules of the Common Gateway Interface protocol.

Although, CGI has numerous advantages, such as its cross-platform environment and speed, there are numerous disadvantages that have resulted in it becoming somewhat of a used technology. Furthermore, one major problem with CGI is that every time a script is executed a new process is started, therefore slowing busy web servers down.

2.3.5 – JSP

Java Server Pages (JSP) is a scripting language, developed by Sun, that uses many features of the Java programming language and can be embedded in the HTML code of a website. It is notable that JSP is very similar to ASP but has one major advantage over ASP, which is the way it uses the Java language in the form of Java Beans. Java Beans provide JSP will many functions of Java such as its object-orientated code, class libraries and platform independence. However, JSP has disadvantages such as it requiring complex code and it is relatively slow compared to other options which is not good for interaction with databases.

2.3.6 – Conclude and Evaluate

PHP has been ultimately chosen as the server-side scripting language to be used with the production of the web-based system, mainly to communicate with the DBMS. ASP and PHP would both perform the required functions but PHP has the marginally superior cross-platform ability. Additionally, the developer has a better understanding of PHP which was a major factor.

2.4 – Client-Side Scripting Languages

2.4.1 – Introduction

Nearly all web based systems that require a user to send data contain some client-side scripting that is used for validation. Validation, before data is sent to the web server, is imperative when a user sends data as human errors are easily made and certain aspects of a web form may be misunderstood. Additionally, many web sites are automised and it is very important that the client-side scripting checks that all the required data has been submitted before, for example allowing a customer a web sites services. PHP and MySQL can be used to validate data but it is far easier to use a client side scripting language. One major advantage of client side scripting is that it takes a lot of strain off the web server as the programs are executed on the user's system.

2.4.2 – JavaScript

JavaScript is the most popular client-side scripting language currently used in order to make static HTML pages more interactive. JavaScript is an open language, developed by Netscape, which is supported by a number of established web browsers such as Internet Explorer, Netscape Navigator and emerging browsers like Mozilla and Opera.

The main advantage of JavaScript is its ability to perform validation on web forms taking the strain off the web server whilst executing the scripts on the client-side. Although JavaScript does not have the power of VB Script or Java Applets, its speed, efficiency and simplicity is usually enough for most web sites.

However, certain situations like the one below have affected customer confidence in the language and its use for validation.
“A number of popular Web sites were attacked this summer by hackers who inserted a small JavaScript file into their Web servers. The JavaScript program would run inside the Web browser of anyone who visited those sites. It would quietly download from a Russian Web site a Trojan horse program called Berbew, which included a key-stroke logging program that would allow hackers to harvest any information users typed. Thus, infected users who went to an e-commerce site and typed in their credit card numbers were giving this information to the hackers, putting the users potentially at risk of a swindle or even identity theft.” [16]

2.4.3 – VBScript

VBScript is a scripting language that has been developed by Microsoft and supported by their Internet Explorer web browser. VBScript is based on the Visual Basic programming language but is considerably simpler. In a number of ways VBScript is similar to JavaScript in the way it enables web developers to include interactive controls, such as buttons and scrollbars, and web form validation techniques.

“It's important to note that VBScript is supported only by Internet Explorer.” [17] The major downfall of VBScript is that it is only supported by Internet Explorer which is a major factor as many people and businesses are beginning to make the switch to browsers such as Mozilla Firefox, Netscape Navigator and Opera.

“Security is the reason why Jefferson County in Colorado ordered its 2,000 government workers to switch to Firefox about five months ago, said David Gallagher, the county's director of IT development. Gallagher said he came to view IE as "a VDS - a virus distribution system." [18]

2.4.4 – Conclude and Evaluate

In relation with client-side scripting, JavaScript has been chosen for the web-based system. The only real need for JavaScript in this project is for the validation of data inputted into the system. The fact that JavaScript can be used on many different browsers and that VBScript is dependant on Internet Explorer was a key factor.

2.5 – Software Development Methodologies and Approaches

2.5.1 – Introduction

It is very important to choose the best possible methodology for this specific project in order to complete the product to the best possible standard. Certain methodologies are only appropriate for certain problems and it is imperative to choose the methodology that best suits this project.

2.5.2 – Traditional Functional Decomposition

This is a method of programming decomposition in which a problem is broken up into several independent tasks, or functions, which can be run simultaneously on different processors. The tasks/functions are usually verbs that describe what needs to be done for that module. Many of these tasks can be continuously divided in order to produce smaller and smaller modules that can be programmed by individual programmers. Therefore, this methodology is very time efficient as a project manager can continuously review and monitor the progress of a project.

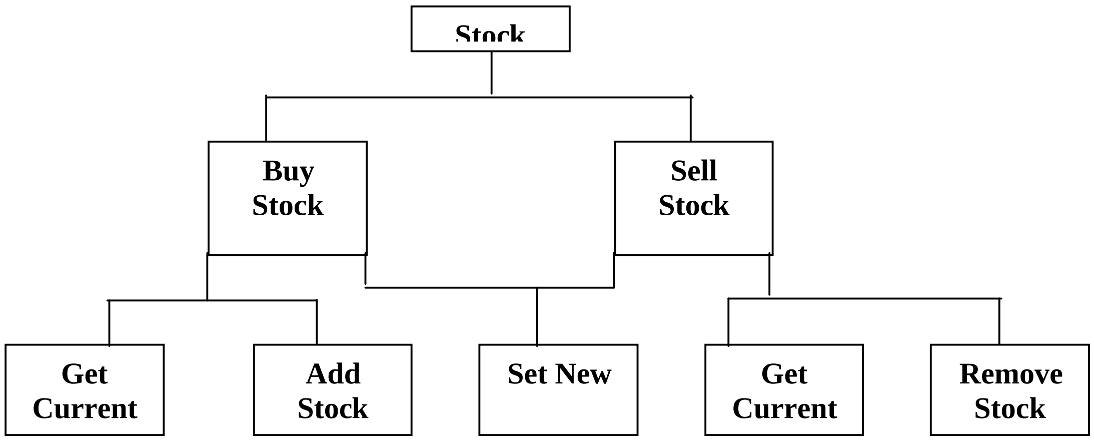


Figure 1

Although there are many positives of this approach to software development there are many faults that have been highlighted in a number of high profile failures such as the Denver International Airport Project [19]. One fault is that the functional approach is not sufficiently scalable to deal with complex, large projects. As you can see from ***Figure 2*** in Appendix B, the more complicated a project gets the more likely it is to fail. Nevertheless, this project on a large scale is considered extremely small and therefore the risk of it failing is very small.

2.5.3 – Object-Oriented Approaches

Object-Oriented Approaches differ from functional approaches as they organise the functions around the objects (nouns) that interact with each other instead of actions/commands (verbs). It is apparent that the Object-oriented approach looks at the real-world, overall problem rather than looking at each separate function as the problem which is the functional philosophy.

2.5.4 – Software Development Life Cycle (SDLC) Models

2.5.4.1 – The Waterfall Model

The Waterfall Methodology (Traditional Methodology) is a phased software development approach that requires milestones, assessments and certain deliverables at each and every phase transition. The model depicts that each phase must be followed in succession and each transitional stage is not complete until the deliverable for that specific stage has been entirely finished. In contrast, the Waterfall Model has many advantages such as it is extremely straightforward and has the shortest development time of all the other methodologies.

The main weaknesses of this method are the inflexibility for scope changes and that certain limitations/problems may not arise until further on in the development cycle, which at that stage it may be too late for any amendments. In addition, following this strict and rigid methodology it is not possible for the end-user to be able to see the final product until it has been completely finished.

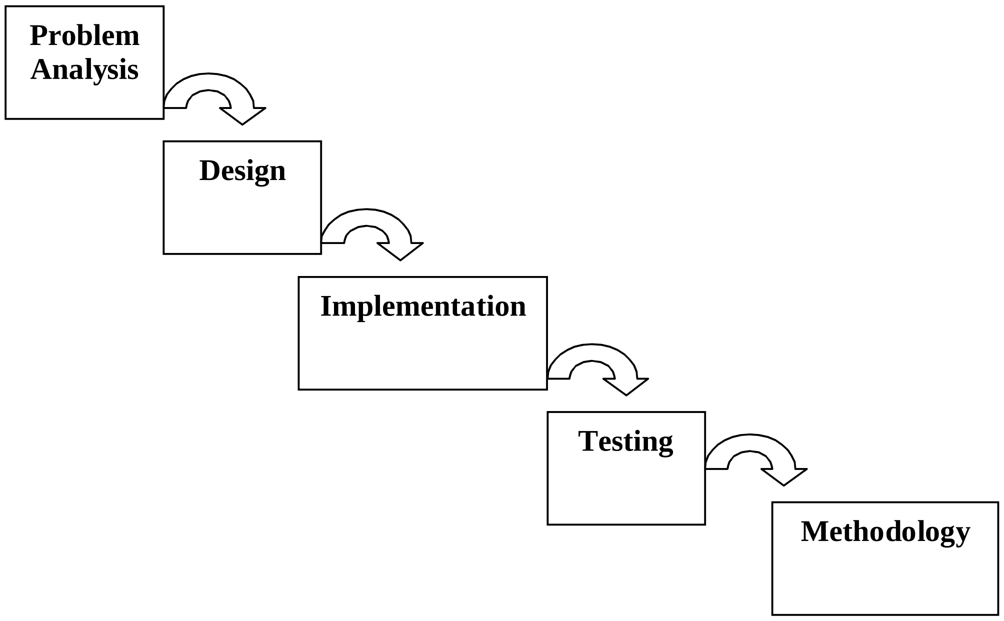


Figure 3

2.5.4.2 – Rapid Application Development (RAD)

The Rapid Application Development (RAD) model is a systems development methodology that is used to drastically decrease the time that is required to analyse, design and implement a fully working product. The RAD methodology is not suitable for all proposed projects because certain systems, such as an Air Traffic Control System, require the code to be of the highest standard and a rapidly produced system would not install any confidence into the users.

The RAD methodology uses a number of management techniques that are optimized in order to produce the product quickly:

Prototyping – This is where sample products are continuously produced until the final working model is completed to the required standard. This is an excellent method of development as the developer is able to constantly interact with the end-user and show them what the product will be like. The valuable feedback from the end-user is then used to make enhancements and changes. This approach emphasises relationship and change management.
Iteration – works closely with Prototyping as the development process cycles often many times in order to produce the required product.
Time-boxing – This is a certain management technique that concentrates on the delivery of the product above everything else. It is notable that under a time-box the scope can change but the delivery cannot.

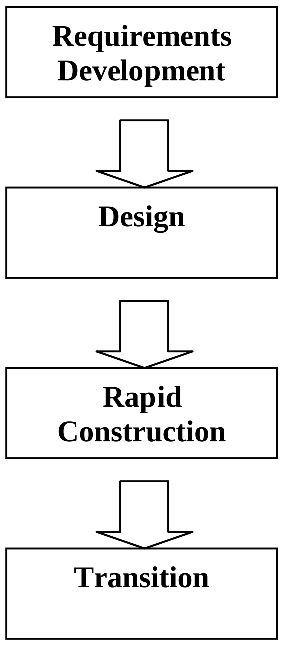


Figure 4

2.5.4.3 – Spiral Model

This software development methodology is used to combine the advantages of both top-down and bottom-up concepts by incorporating both design and prototyping in stages. Furthermore, a design goal is found at the beginning of each phase and finishes with the end-user reviewing the progress and submitting feedback. It is notable that the Spiral Model is rarely used today but has influenced the modern day approach of agile software development.

2.5.4.4 – Rational Unified Process (RUP)

This methodology is based on the object-oriented Unified Modelling Language (UML) and categorizes the development in four separate stages:

Inception – during this phase the business is reviewed and the development team decide whether or not the project is worth doing.

Elaboration – The developers identify the possible risks and mitigations that could possibly occur during the development of the project and therefore this stage is extremely important.

Construction – this stage reviews the code that has been written by testing the software that has been produced and determine whether or not it has met the requirements from the Inception phase.

Transition – Any last-minute amendments are made in this phase following feedback from the user.

The problem with the RUP is that it requires a fully evaluated prototype at the end of every phase. This will be difficult with a project that requires the developers to keep to a strict time plan.

2.5.4.5 – Conclude and Evaluate

The Rapid Application Development (RAD) methodology has been chosen as it fits in well with this project. Websites are systems that generally improve as time progresses and require a number of iterations and end-user cooperation in order to produce the final product. Furthermore, a number UML techniques will be used as the Soft Systems method of decomposing problems when they arrive. Therefore, by using the structure and flexibility of the RAD methodology (allowing for time management) and utilising a number of UML techniques (to encourage innovative thinking) will hopefully prove to be a highly effective combination.

2.6 – Information Systems Project Management

This project is relatively small compared to the developments currently happening in large organisations. Nonetheless, many lessons can be learned from past high-profile software development failures. Furthermore, irrespective of the size of project in hand, in order for a project to be a success the development team must understand competently the requirements of the customer [21]. The efforts of the development team would be wasted if the software produced is not what the customer requires. Therefore, it is imperative to be aware of the possible symptoms of software development failure, which are as follows:

- Customer frequently requesting for changes to be made, therefore confusing the development understanding of the solution.
- Customer not properly understanding their requirements.

- Tasks being overlooked by both customer and development team.
- Insufficient communication between the customer and the development team leading to misunderstandings and errors going unnoticed.
- Lack of involvement from top management. [22]
- Poor or imprecise requirement definition.

As development methods and approaches change through time, it is crucial that valuable information and insights are acquired from an effective review process in order to aid the control of future projects.

2.7 – Methods of Communication with the User

As explained previously in the last section, regular well-organised communication with the end-user is extremely important for the success of the project. Since Urban Lifestyle Letting is primarily based in Greater Manchester it is difficult to organise face-to-face meetings due to University commitments and company opening times. Therefore, it has been decided through discussion with the user that telephone meetings will occur at 12.00pm on Mondays fortnightly. In addition, all other queries and milestones can be communicated via email to damon@urbanlifestyleletting.co.uk.

Also, all completed work, software scripts and project write-up, will be uploaded to the project website at http://www.comp.leeds.ac.uk/jhs2mnb. This allows the Project Supervisor and the end-user to monitor the progress of the project. Also, this allows the developer to more effectively organise the task at hand.

2.8 – Customer Resource Management (CRM)

“Effective customer relationship management (CRM) means improved sales, support, and marketing efficiency.” [23]

In the business world today many organisations sell the same products and constantly compete with each other over the prices of these goods. It is imperative that any business in any sector of the market attempts to find their competitive advantage [24] over the other companies, in order to stabilize their immediate future. Therefore, it is noticeable that over the last few years many of these companies are attempting to improve their CRM in order to gain market share.

2.9 – Live Help

Many companies on the Internet are competing in extremely competitive markets and it is imperative to attempt to find a competitive edge [25] over competitors in order to secure market share. Live Help is a service that attempts to improve the customer-business relationship and satisfaction.

Live Help is an enquiry service that is offered to customers browsing a particular website. Live Help allows a customer to initiate a conversation with a member staff via instant messages. Furthermore, the service allows customers to obtain immediate responses to queries about stock availability, delivery times and other more ambiguous questions that would not usually be illustrated on a company website. Many people shop on the Internet because of the convenience [26] and one inconvenience of ecommerce is that emails are asynchronous and there is a waiting period for simple queries. Live Help allows simple queries to be answered instantly and maintains the customer's interest.

Gabbott et al (2003) [27] makes an observation during his research of businesses that a substantial percentage (42%) did not respond to a simple query made through their website. This demonstrates that many companies simply do not take the World Wide Web seriously as a source of communication and do not understand the impending dissatisfaction felt by customers who utilise a contact resource and are subsequently ignored.

Trust [28] has always been a major issue regarding the development and success of E-commerce. Trust is slowly gained and quickly lost [29] and for that reason companies should do everything that they possibly can in order for a customer to feel comfortable about the business's integrity. Moreover, telephone communication between a customer and a member of staff is a better way to communicate than Live Help in portraying trust. However, Live Help is a useful substitute for customers who feel uneasy using the telephone, do not have a telephone or have the inability to speak.

It is noticeable that the Live Help service is modelled on behaviour that occurs in a standard shop. Customers can initiate conversation with operators as they would in a shop and make queries. Moreover, operators can observe customers viewing habits and offer them help or state special offers on goods that are relevant to what they are showing interest in. The similarities between Live Help and the behaviour in a real-life shop are depicted in the diagram below:

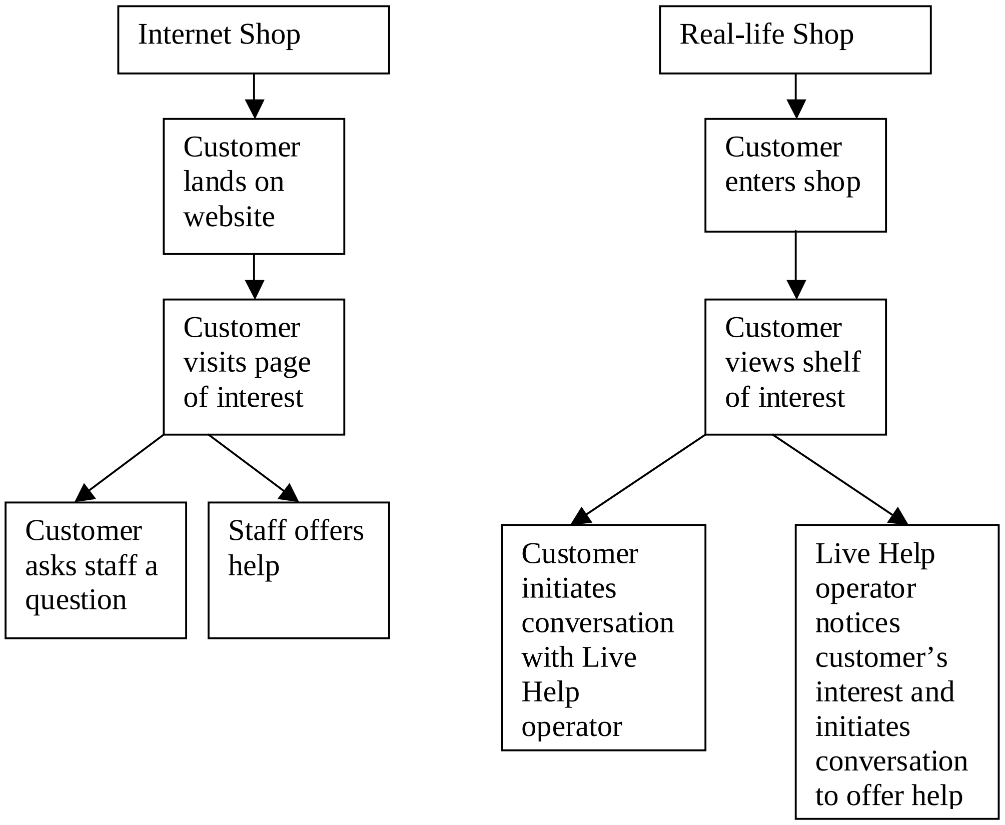

Internet Shop
Customer lands on website
Customer visits page of interest
Customer asks staff a question
Staff offers help
Real-life Shop
Customer enters shop
Customer views shelf of interest
Customer initiates conversation with Live Help operator
Live Help operator notices customer's interest and initiates conversation to offer help

Figure 5

Live Help has many advantages that are widely documented; however its limitations are ones that can severely question the service's cost-effectiveness. For example, websites that receive copious amounts of visitors will require many devoted Live Help operators which can be expensive for the company to finance. However, the main limitation of Live Help is that the service from the websites does not fit in with the behaviour of the World's Internet users. ***Figure 6,*** found in Appendix B, is taken from the 2004 TGI-EGM Colombia study where 1,620 survey respondents reported their Internet usage time by time of day.

It is noticeable from the graph that only two-thirds of the home users have accessed the Internet by six o'clock. Additionally, the average working day is between eight and six o'clock. Many websites offering Live Help do not offer the service after six o'clock. Therefore, if one assumes that the Colombia demographics are relatively similar to global web activity, the service will not be available to a third of Internet users.

2.10 – Conclusion

It is therefore apparent that an automated live help feature would be desirable for companies to enable during evenings and weekends. If the feature is fully developed it could also be incorporated full time, which would take the strain off the staff at the company and still provide a friendly interface for customers.

CHAPTER 3 – DESIGN

3.1 – Introduction to the design

Now that the system requirements have been identified and modelled, it is possible to expand the diagrams into design documentation. These can be used as a basis for constructing the actual working system.

A standard three-tiered architecture has been chosen as the overall structure of the system and the technologies used in each tier have been previously determined in Chapter 2. Stevens (1996) [31] emphasises the importance of using a three-tiered architecture opposed to a two-tiered client/server approach. Stevens highlights the need to offload mainframe functions in order to take the strain off the client, server and database and separating the application functions makes it easier to implement. In addition, Stevens states that the increased modularity avoids complications caused by modifying or replacing a tier without affecting other tiers.

The desired outcome is achieved by incorporating the middle tier to define the logic. As you can see from the diagram below, PHP is operating on the Application Tier acting as the go-between. PHP allows the HTML on the Presentation Tier [32] to communicate with the MySQL database on the Data Storage Tier.

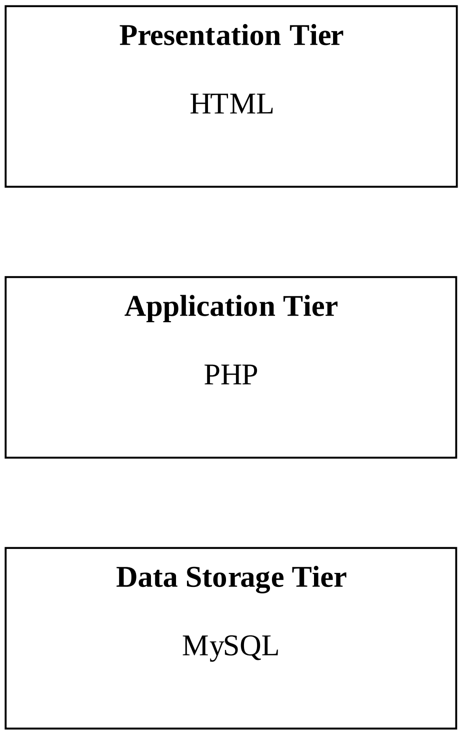


Figure 7

3.2 – Presentation Tier

3.2.1 – Human Computer Interaction (HCI)

The staff at ULL are considered to have a low to medium I.T. literacy and therefore they will be expecting an easy-to-learn and consistent layout. It is imperative that a large majority of time is spent on the usability, as surveys have previously shown that over 50% of the design and implementation effort of a project is devoted to user interface. [33]

Ensuring that a well-designed and implemented user interface is developed can have massive positive implications for the business in-hand such as the following:

- Fewer errors made the users of the system.
- Reduced time taken to perform tasks.
- Lessen the strain on the support staff.
- Eliminate the need for staff training.
- Significantly decreases user disruption.

3.2.2 – User Interface Design

Sutcliffe stated the emergence in usability and user interface design “Interface design became important because pleasant, attractive, easy-to-use software sells well.” [34] Sutcliffe was indicating that the underlying purpose of a computer application was to aid the user with their day-to-day work and not hinder any progress with any complications.

When creating a sophisticated computer application for users with limited IT skills, it is imperative to have a competent Graphical User Interface (GUI) [35] that simplifies complicated operations. Many software developers overlook creating a good user interface, which often determines the acceptance of the application in the marketplace [36]. It is possible that the end-user will find the application too complicated and difficult to use and could result in a perfectly working system not being accepted. It is very important that the developer should attempt to design a GUI that looks professional and easy for the user to use. The diagram below shows the intended user interface for the public website. It is intended to have the quick search feature and email form on each page in order to encourage the user. Furthermore, the company logo is going to be placed at the top of every page to remind the customer of the company name and the navigation toolbar is to be placed at the top of every page for familiarity.

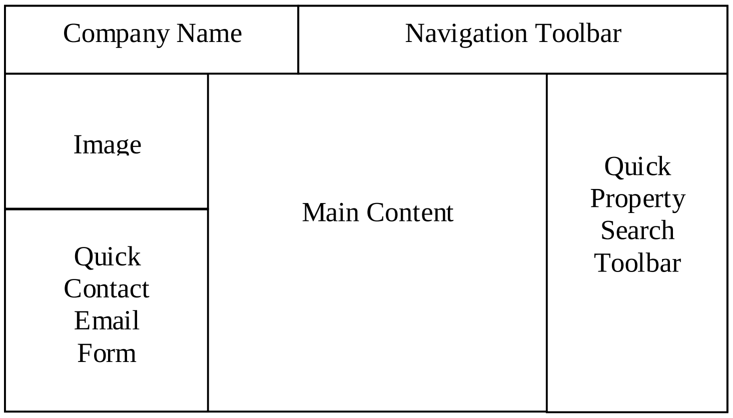


Figure 8

3.2.3 – Check Against Acceptance Tests

It was decided to check the design against the acceptance tests. The purpose of doing this was to see if the design was capable of producing the intended solution in order to be accepted by the end user.

Test 1

Load web page – **This will be possible when typing http://www.urbanlifestyleletting.org.uk in a web browser as the web scripts will be uploaded, using FTP, to the web host.**

Login to the private area – **As you can see from System Navigation Layout, there is a login script and in the database tables there are “password” and “username" attributes.**

Pass Criteria

Web page should successfully load

After entering a valid username and password the user should be taken to the main menu

Test 2

Add a new property to the system – **As you can see from the Design Level Class diagram, there is a method called "add_property()" that allows a property to be added to the system.**

Amend the details of that property – **Within the Design diagram there is a method called "edit_property()" that allows a property to be amended.**

Search and list the information on that property – **This can be achieved using the "find_property()" method found in the Design diagram.**

Delete that property – **The method "delete_property()" will allow a property to be deleted.**

Print list of properties – **A list of properties will be able to be printed using the method "print_property_list()".**

Pass Criteria

Successful addition of a new property

The details of that particular property should be successfully changed

The relevant property should be found and its details listed

The property should be successfully removed from the system

Test 3

Add a new tenant to the system – **Within the Design diagram there is a method called "add_tenant()" that allows a tenant to be added.**

Amend the details of that tenant – **This can be achieved using the "edit_tenant()" method found in the Design diagram.**

Search and list the information on that tenant – **This can be achieved using the "find_tenant()" method found in the Design diagram.**

Delete that tenant – **The method "delete_tenant()" will allow a tenant to be deleted.**

Print list of tenants – **A list of tenants will be able to be printed using the method "print_tenant_list()".**

Pass Criteria

Successful addition of a new tenant

The details of that particular tenant should be successfully changed

The relevant tenant should be found and its details listed

The tenant should be successfully removed from the system

Test 4

Backup Database – **As you can see from the System Navigation Layout, a script exists which will backup the database.**

Restore Database – **Once again looking at the System Navigation Layout, a script exists which will restore the database.**

Pass Criteria

A physical backup of the database and data stored in the tables must be produced

The required backup must be able to be selected and restored to the web host database

Test 5

View a message forum – **This will be achieved by accessing the relevant script that can be seen on the System Navigation Layout.**

Make a post on that message forum – **The method "add_post()" will allow this to be achieved.**

Pass Criteria

The required message forum must be able to be selected and shown

Successful post made on the message forum

Test 6

Add a user to the system – **Within the Design diagram there is a method called "add_user()" that allows a user to be added.**

Amend the details of that user – **This can be achieved using the "edit_user()" method found in the Design diagram.**

Find and show a user's username and password – **This can be achieved using the "find_user()" method found in the Design diagram.**

Remove a user – **The method "delete_user()" will allow a user to be deleted.**

Pass Criteria

Successful addition of a user to the system

The details of that particular user should be successfully changed

The relevant user should be found and its details listed

The user should be successfully removed from the system

3.3 – Application Tier

3.3.1 – Application Design

The Design Level Class diagram [37] is an extremely important aspect of the design phase in this project. This is because it provides the basis for the entire system as it details the modules, attributes, methods and relationships.

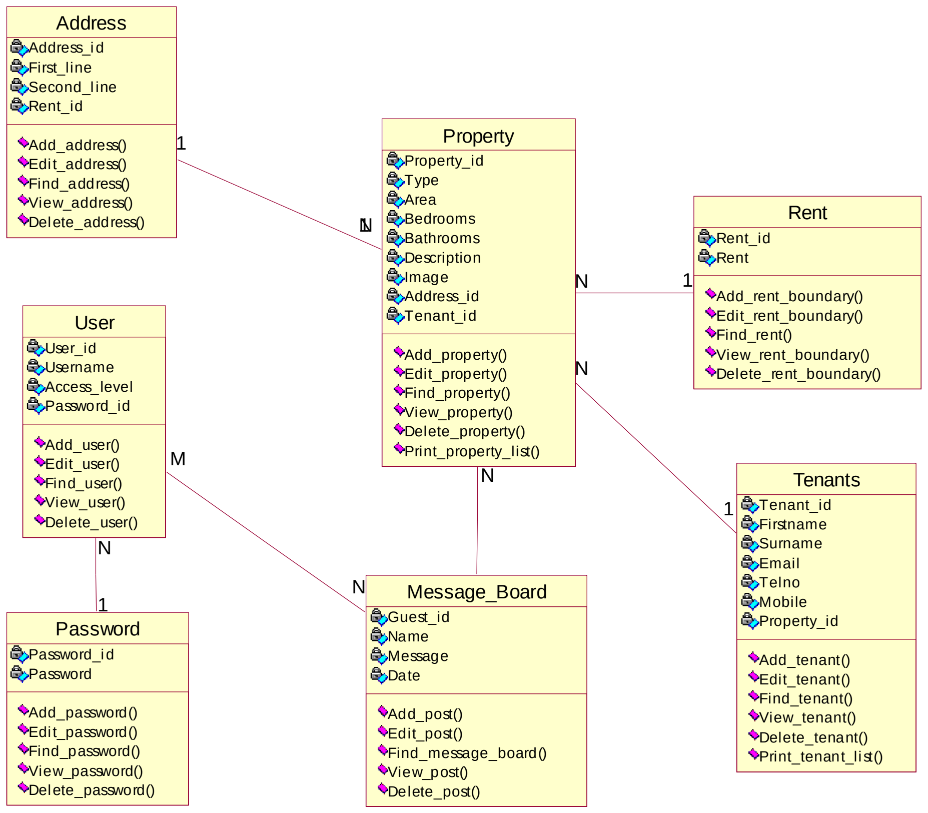
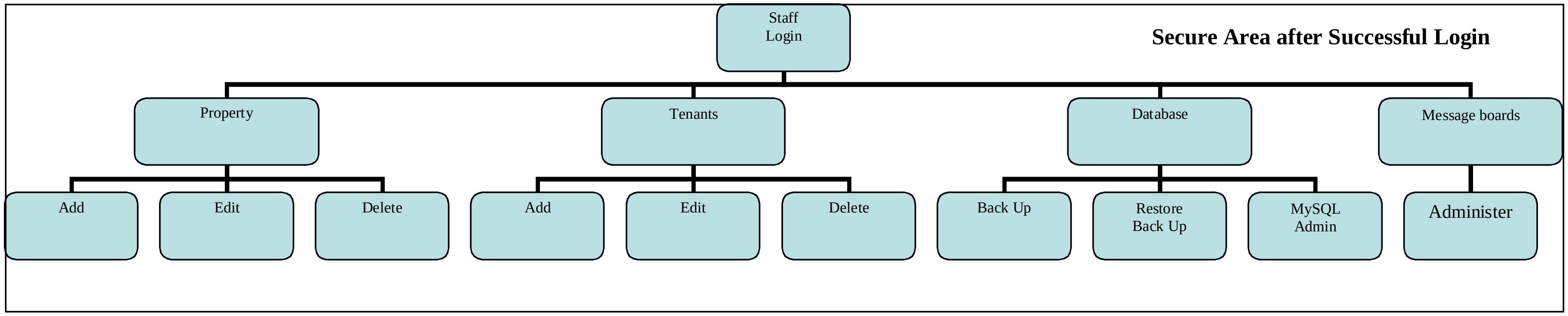


Figure 9

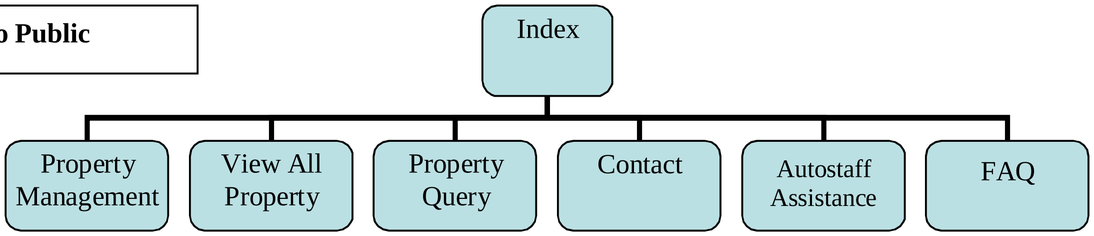
Figure 10 below shows the System Navigation Layout of all the forms to be included.

Online Database Driven System for Urban Lifestyle Letting

Staff Web-based System



Website Available to Public



Message Boards Available to Current Tenants

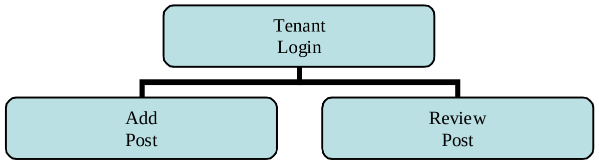


Figure 10 – **System Navigation Design**

3.4 – MySQL database design (data modelling) – Database server

Ullman (2003) [38] states that the primary step in designing a database is establishing its structure. A well organised structure is essential for the long-term success of the data management. This is achieved using normalisation where the data is organised into its most natural, stable, subject-orientated, shareable and non-redundant form.

3.4.1 – Normalisation

The concept of normalisation was initially developed by E.F. Codd of the IBM Research Laboratory, California. Codd (1970) [39] proposed a collection of rules defining the creation of well-structured relations for a particular relational database implementation.

It is imperative that the developer understands clearly what the user requires for the intended application. This requires extensive communication with the user to understand how the information will be accessed, which will therefore determine the modelling. Ullman (2003) highlights that one of the best methods to establish what data should be entered into a database is to elucidate what questions will be asked of it and what data will be included in the answers.

3.4.2 – Keys

When creating a Relational Database, the designer must take into consideration the keys existing in tables. The Primary Key is a unique identifier that:

- Must always contain a value and cannot be NULL.
- Must have a value that is different to every other record in the table.
- Must have a value that is constant and never changes.

3.4.3 – ER Diagram

Figure 12 shows the Entity Relationship Diagram created for this project. The ER model shows the type of information that will be stored in the database

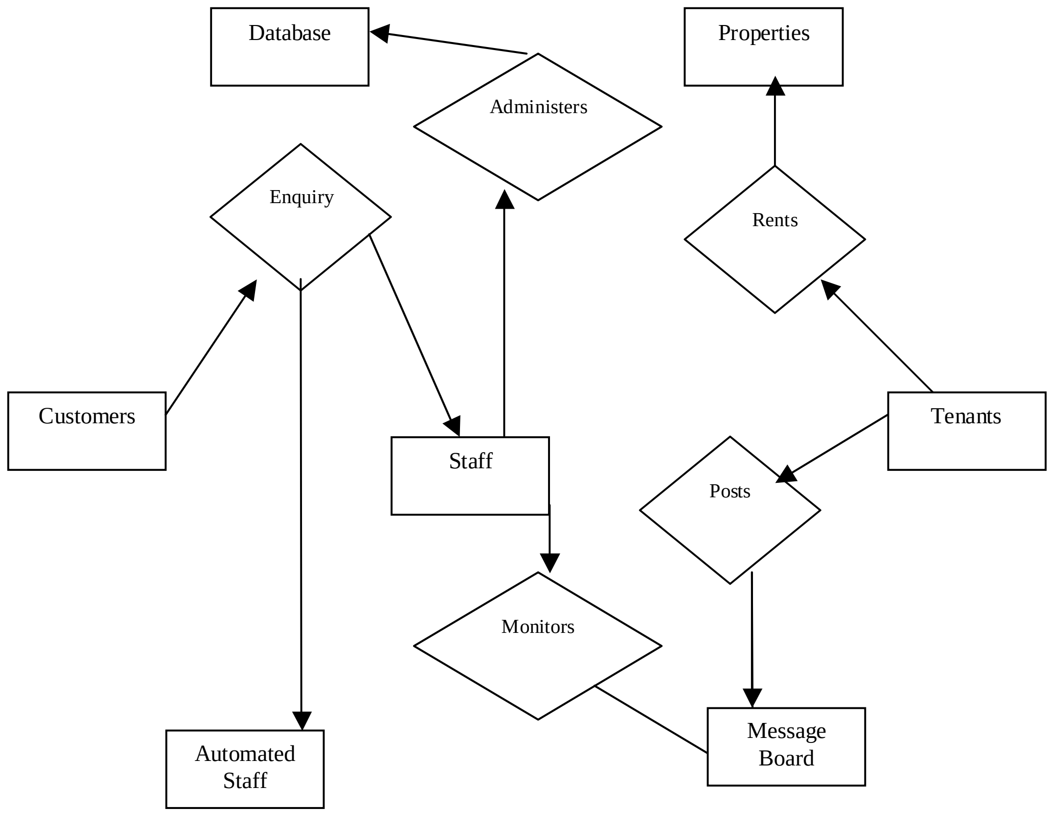


Figure 11

3.4.4 – Relationships

When the keys and fields have been defined it is important to establish how the data in the different tables relate to each other via one-to-one, one-to-many and many-to-many relationships.

3.4.5 – First Normal Form (1NF) [40]

In order for a database to qualify being in First Normal Form (1NF) each of the columns must only contain atomic values. With regards to this project, the database would not be in 1NF if a field for an address existed. This is because the address has many values such as the “street address”, “town”, “city” and “postcode”. Therefore, a separate field would be created in the database to contain the previous data.

3.4.6 – Second Normal Form (2NF) [41]

In order for a database to qualify being in Second Normal Form (2NF) it should already be in 1NF and that all columns existing in a table that are not a key must directly relate only to the primary key. With regards to the database in this project, it would not be in 2NF if the field “area” existed in the same table as the field “street_name”. This is because it would be very likely that there will be multiple instances of the same value stored in the “area” column. The way to overcome this problem will be to separate these columns into their own tables resulting in each value stored only once.

It is therefore fair to say that the process of Normalisation is implemented by making more and more tables until the redundancies have been eradicated.

3.4.7 – Third Normal Form (3NF) [42]

A database is considered to be in Third Normal Form (3NF) when it is in 2NF and the non-key fields are mutually independent. By making sure that the database used in this project is in 3NF will help to maintain system stability and endurance [43]. It is desirable to create one-to-many relationships between the tables in order to progress to 3NF. If many-to-many relationships appear between tables, it is likely that they may have to be re-structured. As a result the following schemas were produced:

PROPERTY

Property_id	Type	Area	Bedrooms	Bathrooms	Description	Image	Map	Address_id	Tenant_id

ADDRESS

Address_id	First_line	Second_line	Rent_id

RENT

Rent_id	Rent

TENANTS

Tenant_id	Firstname	Surname	Email	Tel_no	Mobile	Property_id

USER

User_id	Username	Access_level	Password_id

PASSWORD

Password_id	Password

MESSAGE_BOARD

Guest_id	Name	Message	Date

3.4.8 – Data Types

Once all the tables and columns of the database had been identified, each field's data type was determined to identify what type of information was to be stored in each field.

3.4.9 – Design Summary

The system has been thoroughly designed at all three tiers and is ready to be implemented. The deployment and component diagram that can be seen below illustrates how the physical components of the system, the relevant hardware and software, are connected and maps out where they will operate.

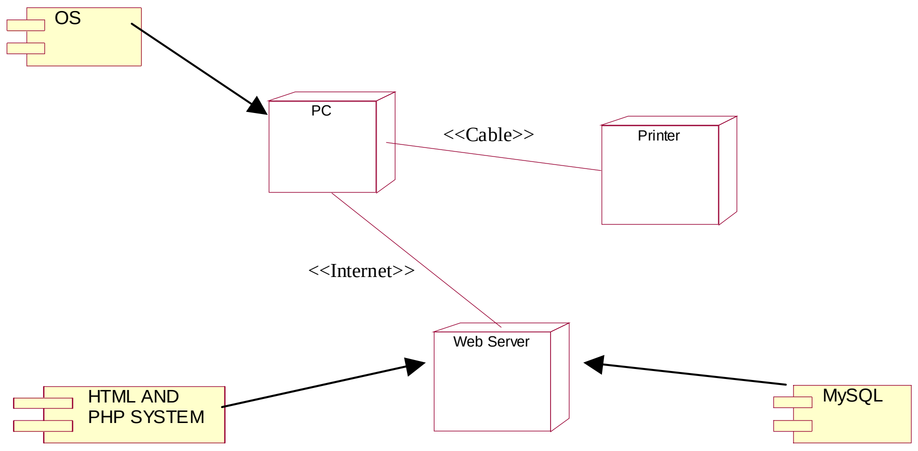


Figure12

CHAPTER 4 – IMPLEMENTATION

4.1 – Introduction to the Implementation

The implementation section of this project illustrates how the developer went about creating the system in relation to the requirement's of the user and the system design.

4.2 – Installation

The system was firstly created on the developer's local Apache [44] Web Server with PHP[1][45] installed allowing the PHP scripts to be executed. Additionally, MySQL [46] was installed locally in order to interact with the PHP scripts. By default Apache is not programmed to automatically execute PHP scripts and therefore a new file type called "application/x-httpd-php" was created in the "http.ini" initialisation file in order to overcome this issue.

[1] http://www.php.net

4.3 – Uploading the Live System onto the World Wide Web

When the initial system had been created on the developer’s local machine, it was imperative to upload the scripts using FTP (File Transfer Protocol) onto the company’s web server, provided by “Streamline” [47], in order to see how the system appears in different browsers, screen resolutions and Operating Systems. Streamline’s web servers are hosted using Apache and PHP is installed on all hosting packages.

4.4 – Presentation Tier Implementation

The process of implementing the Presentation Tier began with creating the interface for the public website. The use of Frames is a HTML technique that creates a fixed, static feature to a website usually in the form of the navigation buttons/menu and company logo. Frames would drastically cut down on the amount of HTML code that would have to be written. However, frames are not used as much today as they were several years ago [48]. This is because they present problems [49] to a number of search engines when they come to index the webpage and browser compatibility issues [50]. Therefore, as Urban Lifestyle Letting has stated that they wish to be recognized on the major search engines, frames had to be immediately dismissed. As a replacement for Frames, HTML Tables were used to create the general layout and structure of the web pages. They make it easy for search engines to crawl, accomplishes the same look/interaction as Frames and only requires one HTML file for the entire web page [51].

4.4.1 – Presentation Tier Challenging Aspects

4.4.1.1 – Search Engine Optimisation (SEO)

Keyword rich title tags, keyword rich body text and links from important websites are commonly known as the most effective [52] methods of Search Engine Optimization. Urban Lifestyle Letting wanted their site to be found on the major search engines when the company name was entered. Therefore, by having the company name as a title tag on each page meant that this could be achieved. The website was designed to include a number of references to important key phrases in the HTML such as “Letting Agent”, “Property to Rent” and “Rent Property Lancashire”.

Although Meta data does not have the effect on search engine rankings as it use to, there still is enough evidence [53] to suggest that it still has a large influence on the position of a web site on the major search engines.

```
<META content="index, follow" name=robots>
<META content="text/html; charset=windows-1252" http-equiv=Content-
Type>
<META name="author" content="Urban Lifestyle Letting, Bury, Bolton
and Blackburn Lancashire">
<META NAME="DESCRIPTION" CONTENT=" Urban Lifestyle Letting. Property
to let in Manchester, Bury, Lancashire, Bolton, Blackburn, Preston,
Stockport, Blackley, Middleton, Rochdale">
<META NAME="KEYWORDS" CONTENT=" Urban Lifestyle Letting, rent, let,
Manchester, Bury, Lancashire, Bolton, Blackburn, Preston, Stockport,
Blackley, Middleton, Rochdale">
<META NAME="RATING" CONTENT="General">
<META NAME="REVISIT-AFTER" CONTENT="1 Week">
```

4.4.1.2 – JavaScript Validation

JavaScript Validation was used to test the correctness and integrity of the data entered into the system by the user through a web form. Client-side validation has a number of benefits as the data is checked on the server-side before it is uploaded, therefore taking strain off the web server. In addition, JavaScript validation enables the user to rectify any human errors that they may have made which avoids them from sending the form again with the correct details. When writing the JavaScript code for validation, it was imperative to check for required fields left empty, text fields being entered into numerical fields, fields being entered that were too long/short and others that would question the integrity of the database.

4.4.1.3 – Robots.txt – Security Issue

When search engine spiders crawl through web pages it will search the root domain for a special file called “robots.txt” (http://www.urbanlifestyleletting.org.uk/robots.txt). This file is extremely important as it instructs the web crawler which pages it may index in its search engine (The Robots Exclusion Standard [54]). Regarding the system created during this project, it is important to instruct the web crawler, in the “Disallow” section seen below, not to index the administration sections that are accessible to staff only at Urban Lifestyle Letting.

```
User-agent: *
Disallow: admin.htm
Disallow: tenants.php
Disallow: backup.php
Disallow: restore.php
```

4.5 – Data Storage Tier Implementation

When implementing the MySQL database it was imperative to closely reference the Design Level Class diagram, ER diagram and the database tables found in Chapter 3. The tool MySQL Control Center [55] was used as the graphical interface in order to simplify the process of adding, amending and deleting aspects of the database. Otherwise, one would have to connect to the database through a command line interface with limited graphical, automated features.

4.5.1 – Data Storage Tier Challenging Aspects

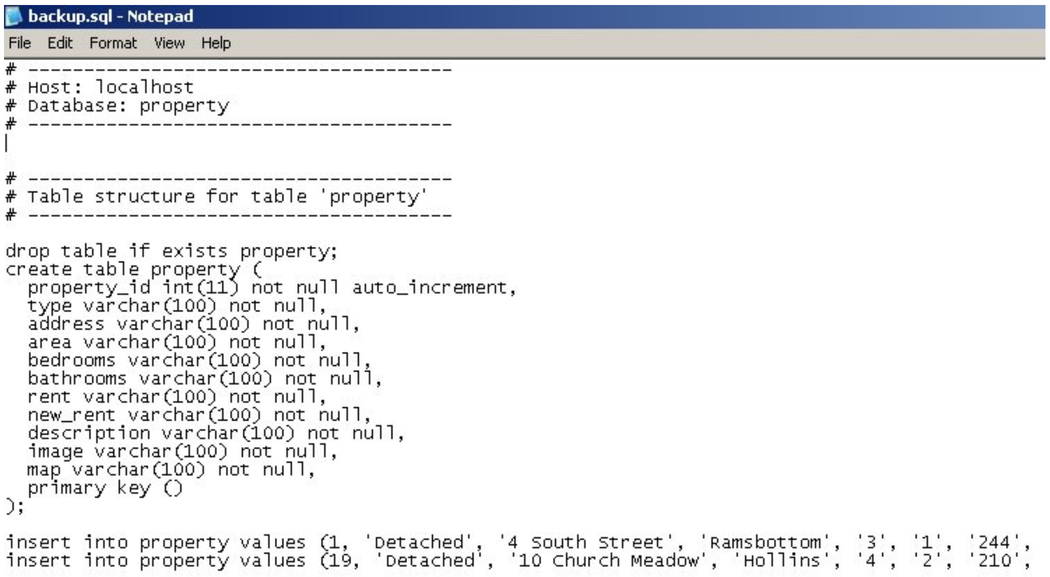
4.5.1.1 – Quick Search

The Quick Search feature that appears on each of the company's public web pages proved to be a demanding aspect. A number of contrasted inputs had to be taken from the HTML drop-down menus and placed into relevant PHP variables. The PHP variables then had to be entered into SQL queries that were used to query the database. The results from the query are then illustrated on the web page to the user.

4.5.1.2 – Backup Database Code

As the staff at Urban Lifestyle Letting have limited IT skills it was very important to have a simple method of backing up the business database. The web hosting company Streamline obvious back up their servers at regular checkpoints using RAID (Redundant Array of Inexpensive Disks). Therefore, if the server was to crash the majority of the data would be recoverable. However, if the system was to be penetrated by a computer hacker or accidentally corrupted by a member of staff, the company would have to rely on the web host to provide a backup. The PHP script implemented to backup the database was very complicated and entailed a number of challenging aspects. Of these aspects, all of the contents of the database had to be selected using a query and produced into SQL executable form.

It was also imperative that the backup of the database tables was outputted into an SQL file so that "backup.sql" could easily be reinstalled by the user. The screen capture below shows the SQL created by using the backup script. The SQL in "backup.sql" can either be entered manually into the database or be inserted by automatically by another script such as "restore.php" implemented in this project.



backup.sql - Notepad

File Edit Format View Help

```
# --------------------------------------
# Host: localhost
# Database: property
# --------------------------------------
|

# --------------------------------------
# Table structure for table 'property'
# --------------------------------------

drop table if exists property;
create table property (
  property_id int(11) not null auto_increment,
  type varchar(100) not null,
  address varchar(100) not null,
  area varchar(100) not null,
  bedrooms varchar(100) not null,
  bathrooms varchar(100) not null,
  rent varchar(100) not null,
  new_rent varchar(100) not null,
  description varchar(100) not null,
  image varchar(100) not null,
  map varchar(100) not null,
  primary key ()
);

insert into property values (1, 'Detached', '4 South Street', 'Ramsbottom', '3', '1', '244',
insert into property values (19, 'Detached', '10 Church Meadow', 'Hollins', '4', '2', '210',
```

Figure 13

4.5.1.3 – Restore Database Code

When implementing the “restore.php” script, that takes the backup and restores into the database, it was very important to take a number of precautions in order not to cause unnecessary strain on the web server and basically inform the user of any errors that may have occurred. A switch statement was used to inform the user of the specific error that may have happened, which can be seen in Appendix D.

4.5.1.4 – Storing Images Using MySQL

This resulted in being a complicated issue when it came to outputting the relevant image to accompany the relevant property information. It is advisable not to store the image in the database [56] as a BLOB field because it is slower due to additional processing/data transfer. This problem was solved by entering the full image name into the database, i.e. “house.jpg”, and using the following code that selects the relevant image name from the database and places it into the “$img” variable and subsequently print the image out onto the screen.

```
$img = $row['image'];
print "<img src=\"images/$img\">";
```

4.6 – Application Tier Implementation

The final stage in the implementation of the system was the PHP coding on the Application Tier allowing the user’s selections made from the Presentation Tier to interact with the Database on the Data Storage Tier. In order to connect to the MySQL database using PHP the following code was used with an example query:

```
mysql_connect("hostname", "username", "password");
```

```
        mysql_select_db("database");
        $result = mysql_query(
                "select * from property");
```

4.6.1 – Reusing of code

Many PHP scripts were implemented and therefore it was crucial to re-use code where possible in order to make it easier to track down errors and save time.

4.6.2 – Application Tier Challenging Aspects

4.6.2.1 – Password Protection

When implementing a web based system it is extremely important that only the authorised users are able to view sensitive web pages. It is possible to use the extremely simple code below to protect certain pages on the system.

```
<?php
$username = "username";
$password = "password";
        if ($_POST['txtUsername'] != $username or $_POST['txtPassword'] != $password) {
?>
```

The code in question would be acceptable if only one page needed protecting, however within this project many pages need protecting and therefore it would be unsuitable for a user to have to enter their password and username every time they access password protected pages. The solution that was eventually implemented was to initialise a session based variable by creating a cookie, as seen in Appendix D, which states whether or not the user is allowed access on a particular page and is carried forward until the browser is closed.

Each private page has the following code to check if the user has a valid cookie and if they do not they are automatically directed to the password screen:

```
if($_COOKIE["auth"]<"1")   {
                        header("Location: index.php");
        } else {
                        include "header.php";
```

4.6.2.2 – Contact Details

When using the standard “mailto” method of emailing (seen below) it can often cause an inconvenience to the user as their local email server is initiated.

```
<form action="mailto:enquiries@northernabodes.co.uk"
method=POST  enctype="text/plain">
```

The user cannot choose which email address they wish to submit and is dependant on the email server running local on that specific machine. Furthermore, the “mailto” function makes the company exposed to producers of spam emails. This is because there are many web crawlers [57] existing on the Internet that search the HTML code for the “mailto” function and subsequently add the following email address to their database of contacts. One way to overcome this problem is to write a “sendmail” function in PHP in order to send the email. The PHP code sits on the Web Server and therefore makes the companies email address inaccessible to the general public (unlike using the “mailto” function in HTML that can easily be accessed by “viewing the page source”). “Sendmail” operates by taking the details from a contact details form then constructing and sending an email as a result, as seen in Appendix D.

4.6.2.3 – Automated Live Help Feature

The prototype automated live help feature was an extremely difficult task to implement. It was imperative for the program to respond with meaningful answers to key phrases typed in by the user. Also, it was also important for the program to store certain relevant responses, generally about the property they may be interested in, that would be used to connect to the database at the end of the conversation and illustrate the results on the web page. The latter caused problems as arrays could not be used because PHP Scripts are executed on the web server and therefore could not be used to store the user’s responses as the page is reloaded after each question. This issue was overcome using MySQL tables that stored the user’s responses. However, another problem arose when it came to selecting the user’s responses from the databases in order to query them with the list of properties. The following query was used to select the last record entered into the relevant table which was subsequently placed into a variable.

```
mysql_query("select eliza_bedrooms from eliza_bedrooms order by eliza_id desc limit 1");
```

The user’s responses which were placed into the “$new” variables were then inserted into the following query in order to select any relevant properties from the database.

```
$result4 = mysql_query(
                    "select * from property where area = '$new3' and bedrooms ='$new1' and
type ='$new2'");
```

As this feature of the project is only an extension it would be extremely difficult to implement, as it is a project on its own, the part of the program that uses pattern matching to find key phrases entered by the user to provide meaningful responses. Therefore, the PHPEliza code found at [58] was adapted to respond to any key phrases recognised. The PHPEliza code references a file called “knowledge.txt” that stores all the key phrases and their responses. The Key phrases and responses were altered in order to make it relevant to Urban Lifestyle Letting. For example, key phrases such as “Deposit Required” and “Property for Sale” were implemented into “knowledge.txt” along with the relevant responses.

4.6.2.4 – Image sizes

The physical size of the images used in the system proved to be a problem. All of the images that were provided by Urban Lifestyle Letting were of a high quality and extremely large, with some of the images being up to a Megabyte in size. Firstly, this created the problem of space on the web server as Urban Lifestyle Letting has only 50 Megabytes of space. Furthermore, the more notable issue was the loading of the web pages with images so large in size. For instance, consider a web page of 10K in size and contains 5 images that are 500K each, the total size is 2510K. Therefore, using a 56K dial-up connection the page will take at least 359 seconds (almost six minutes) to load. It is widely thought in the USA [59] that it will not be until 2007 where half of the households there will be equipped with a broadband connection. Therefore, indicating that there are still a large number of households using dial-up connections who need to be catered for.

This problem was overcome, using Macromedia Fireworks 2004, by firstly reducing the dimensions (keeping the same height and width ratios) and “exporting” the images to an approximate size of 8K, changing the images from “JPEG” format to web friendly “Gif”. Obviously, the quality of the images was drastically reduced but still appearing to a reasonable quality when published on the Internet. Therefore, the same 10K web page with 5 8K images is 50K in total and taking 7 seconds to load in comparison to almost 6 minutes with uncompressed images.

4.7 – User Documentation

A fully documented User Manual has been created for the end user to instruct them how to use the system and can be found in Appendix D.

4.8 – Implementation Summary

The system was developed using the RAD methodology outlined in section **2.3.4.2**. The reiterations and constant reviews took place in order to produce a professional solution to the customer's requirements. MySQL Administrator (found at www.mysql.com/products/administrator) was added to the user administration area towards the end of the implementation. This tool will allow future system developers to make amendments to the existing database tables quickly and easily.

CHAPTER 5 – TESTING

The process of system testing, which typically takes 50% or more of the resources for software development projects [60], ensures that the design mirrors the "…performance, safety, durability, and reliability requirements" [61]. A computer system is usually tested by the two main processes of "validation", which the integrity and correctness of the data are established, and "verification", confirming that the predefined requirements and specifications are met.

Black Box Testing

This method of testing, also known as "functional testing", is used to check the outputted data produced by a software program where certain relevant inputs relate to "…the functional expectations of the system" [62].

White Box Testing

In comparison to "Black Box testing" this method, often know as "Glass Box", "Structural" and "Open Box", of software testing requires the relevant knowledge of the programming code in order to examine the outputs from the application [63]. The programmer, who knows exactly what the program should do, will be able to see if the system deviates from the proposed objective. One limitation of this method of testing is that it does not highlight any faults caused by exclusion in the code.

It is notable that both white and black box processes of testing are essential in order to scrutinize the program for undesired features [64].

5.1 – Test Plan

Test Plans are an essential [65] aspect of software development, as they ensure that the system produced meets the system requirements for which they have been developed. The overall rationale of a testing strategy produced by the test plan is to manage the testing procedure so that it is efficient, effective and economic.

There are always going to be human errors in software development and for this reason boundary test data [66] must be applied for thorough testing. Boundary testing refers to the testing of data inputted into the system and begins with “good values”, moving onto “reasonable but invalid”, to “extreme” and “invalid”. The reason for entering boundary test data is to begin with “good values” and “expected bad values”, if the system fails the software developer is made aware that the system is not ready for testing. If this system suitably accepts these values, it is a good idea to enter “reasonable/predictable” mistakes as the users are likely to make such mistakes. Finally, in order to find faults with the system’s functionality, “extreme errors” should be entered.

Good Values

This is where “valid and current” test data is entered into the system and it is imperative at this point not to try and cause errors.

Expected Bad Values

This checks certain problems caused by fields being left blank, dates being valid etc.

Reasonable and Predictable Mistakes

This checks for errors caused by often entering valid data but not accepted by the system. For example, an individual may enter a first and middle name in a “first name” on a web form or entering a business name into an “address form” which may not be handled by the system. This sort of error is generally caused by the design and implementation of the web form.

Extreme Errors

This is achieved by the individual attempting to basically crash the system by entering absurd values into the web forms. This is usually achieved by entering stupidly long values, enter numbers into text fields and text into numerical fields.

5.2 – Unit Testing

Unit testing is an essential [67] method where each individual module is evaluated in order to discover any errors that may exist within the program. This is generally achieved where the modules are being developed and before they are merged into the main nucleus of the system. It is imperative to test the system early in its development and consistently in order limit the chance of errors residing in the base of the system.

This method of testing proved to be very effective in discovering minor trivial faults as many errors residing in the code were dealt with when it came to the attention of the developer. MySQL and PHP proved to be two excellent choices as both produced useful error messages that allowed the programmer to easily track down and rectify the faults.

5.3 – Security checking

As the system created has a private area containing sensitive data that is only available to authorised individuals, it was crucial to test the security [68]. The security testing checked for the following security faults:

- Allowing a user to perform an unauthorised action
- Failing to ensure privacy of data by using inadequate encryption
- Failing to identify and authenticate a user
- Failing to protect the content against authorised usage
- Allowing the data integrity to be violated
- Allowing undetected intrusions

Furthermore, Urban Lifestyle Letting has been advised to check their website log which is stored on their web server at least twice a week. The website log shows if any unauthorised individuals have attempted to hack into their web server which could be potentially very harmful.

5.4 – Website Checklist Testing

When the website has been fully developed with all the content, vast range of images and appropriate colour scheme it is necessary [69] for it to be compared to a website checklist as many issues are overlooked by the web developer. The web site check list, which can be seen in Appendix D, has been created by the system developer and has been used to ensure that the website created for this project has many of the desired features found in professional websites on the Internet.

5.5 – System Performance Testing

One further method of system testing is the examination of the system performance, which is thought as being a necessity by a number of academics [70]. There are a number of tools [71] on the Internet that allow the software developer to enter the URL, http://www.urbanlifestyleletting.org.uk, and the tool subsequently performs a number of tests on the web page. Many of these tools check a website for the download speed of the web page, the size of the HTML, the size of the images, the total number of objects and many more features.

5.6 – Acceptance Testing

Acceptance testing [72], often known as "Beta testing", "QA testing", "Application testing" and "End-user testing", is the process of the end-user testing the system, where he/she either accepts or repudiates the new system depending on the results from the tests. This concluding method of testing is performed when the system has been completed and tested and relates to whether or not the system as a whole achieves what was previously stated in the system requirements. The Acceptance tests can be seen in Appendix D.

5.7 – System Changeover

When a company changes from using their legacy system to the newly developed system it can have undesired, unexpected effects [73], from the new system completely crashing to users not accepting it. There are four main ways of switching from legacy systems to newly designed applications, they are as follows:

Direct – this is the method of completely shutting down the old, legacy system and replacing it with the new, fully-tested system. However, the most apparent limitation of this method is that it is potentially disastrous if the new system crashes. Furthermore, this method has the problem of being unable to compare results from the legacy system to the new system once the changeover has take place. It would be very difficult to make alterations to the new system without affecting the overall operability. This sort of changeover methodology is best done at the weekend or preferably over a holiday period as to limit the effects it would have on the running of the business. Direct Changeover can be a very efficient method because it instantly becomes operational and avoids duplication of data. However, it requires extreme and thorough test planning in order to be a successful procedure.

Parallel – the purpose of this approach is to keep the old system running along side the new system until the new application is running as intended. Therefore, it is possible to compare two sets of results from the respective systems for likeness. In addition, if inconsistencies appear, the new system can be taken offline and repaired with the older system being in operation. The main disadvantage of using this method is that it requires a lot of labour time in order to run both systems.

Phased – is where certain areas of the system are introduced in stages, i.e. the property database first, then tenant details etc. A new area of the proposed system is introduced stage by stage until the final phase has been run in parallel and the legacy system is discarded. The main benefit from this is that it lessens the disruption to the system in operation. Furthermore, it is possible to learn from mistakes made during previous stages of changeover. The main problem of this method is that it takes a long period of time to completely switch systems when different stages are taken.

Pilot – is a conversion method where a system is usually tested in one specific location in order to get the new system stable before converting to all other locations. Urban Lifestyle Letting currently only has one office, but this method of changeover would become a viable option in the future as the company expects to expand.

Conclusion

As the system proposed in this project is online, it is only really viable to remove the old system and directly introduce the new system on the web host. Although it would be possible to purchase a new web host and have both web systems running in parallel, this would create extra costs as the system will have already been tested on a local Apache [74] web server. Therefore, "Direct Changeover" is the method chosen due to its proficient and cost-efficient attributes.

5.8 – Testing Summary

A number of different and contrasting methods of testing, from Unit Testing to a Website Checklist, have been adopted to make sure that the system has been comprehensively tested before going live.

CHAPTER 6 – EVALUATION

6.1 – Introduction to the Evaluation

The process of system evaluation is the gathering and analyzing of information in order to determine if a system is carrying out the specific objectives that have previously been stated [75]. The evaluation process is primarily used to learn where a system has been a success, what areas need improving and any other possible enhancements that could be required.

6.2 – Minimum Functional Requirements Reviewed

In order for the project to be any sort of success it was incredibly important for the minimal requirements to be achieved. The minimum requirements were the top priority when it came to the system implementation, all of which were fully developed and extensively enhanced.

No.	Minimum Requirement	Requirement Solution	Achieved?
1	Review and redesign the existing static HTML website.	A new, fully developed website was created.	✓

2	Design and creation of a database containing details of all the houses that the company manages.	The database was developed and extended beyond the minimum requirement.	✓
3	Produce an application that allows staff to update the website content online and enables customers to search for property.	This feature was implemented and extended beyond the minimum requirement.	✓
4	Create user documentation instructing the staff how to use the system.	This document has been created and is found in APPENDIX ?	✓
5	Evaluate the application using a number of methods.	The application has been evaluated in section 6.2.	✓

6.3 – Non-Functional Requirements Reviewed

Usability of the system

- The interface should be attractive, consistent and well-designed.

Page links positioned at the top of the page, relevant images and a quick search feature on each page has provided a professional, comprehendible website.

- Basic easily readable fonts and text sizes should be used along with bold attractive colours.

Black text on a white background and a pale blue and white colour scheme has resulted in an eye catching website.

- Must be easy to use and must only require basic I.T. skills.

The content management area of the system allows details to be easily changed and with the user documentation resolving any other issues.

- There should be logical and simple processes with reoccurring themes in order not to complicate the user.

On the public website and on the private system any page can be reached by a single click of the button.

- Data entry should be automised where appropriate.

A number of drop down menus have been used.

Speed and reliability of the system

- The system should have a high performance and there should be little or no time delay when processing data.

The MySQL database has proved to be very efficient when dealing with queries.

- Throughout the system there must be no major functionality errors i.e. System not working correctly.

The system has been thoroughly checked for these problems.

- Minor functionality errors must be kept to a minimum; otherwise the user will lose faith and become disillusioned with the system.

The system has been thoroughly checked for these problems.

- The system must be tested vigorously so as to reduce the potential for a system crash to an absolute minimum. Data corruption/loss must be avoided at all costs.

The system has been thoroughly checked for these problems.

<u>Security of the system</u>

- The system needs to be protected against unauthorised access using passwords.

This has been achieved by using a session variable password system protecting sensitive pages.

- Generating data backups is essential and should be easy and quick to do. An appropriate disaster/recovery plan should be included in the user-manual.

The process of creating backups and restoring data is supported in the administration section and documented in the user manual.

6.4 – Project Extensions Reviewed

The table below reviews the Project Extensions outlined in section **1.6.** The table illustrates if the relevant extensions have been achieved and if so, how they were achieved.

Rank	Description	Complexity	Extension Solution	Achieved?
1	Allow the administrators to alter the contents of the database online.	Moderate/ Complicated	The staff are able to enter a password protected area and alter the database contents.	✓
2	Introduce a section on the website for existing tenants, where each tenant has their own personal message board to communicate with Urban Lifestyle Letting.	Complicated	Each tenant has their own password and they are able to post messages on their personal message board that is reviewed by a member of staff.	✓
3	Introduce a basic area on the website for the administrators (staff at Urban Lifestyle Letting).	Easy	Staff have access to the password protected administration area.	✓

4	Design and create a prototype of an AI automated Live Help feature.	Complicated	This feature was developed and would be considered a partial success. One section of the program looked at key words entered by the customer and provided intelligent responses. However, it is possible to outsmart the program to gain irrelevant responses. The other part of the program successfully stores the customer's response and queries the database to find properties to meet the needs.	✓
5	Have the ability to store and retrieve details of current and past tenants online.	Moderate/ Complicated	This feature has been fully implemented and can be accessed from the administration area.	✓
6	Investigate into search engine optimization (SEO) and optimize the website.	Moderate	Meta Tags, Title Tags and robots.txt were used and has resulted in the website being indexed by the major search engines.	✓
7	Allow the administrators to backup and restore the database online.	Complicated	This feature has been fully implemented and can be accessed from the administration area.	✓
8	Allow the staff at Urban Lifestyle Letting to be able to print (or output to an Excel file) out a prioritized list of requests made by tenants.	Complicated	It was discovered that this would not be required during implementation. However, the company have instructed that if they require this feature in the future then it will be	

			implemented.	
9	Implement an auto-responder when a customer sends an email.	Moderate	This feature was implemented on the web host control panel.	✓
10	Add a basic FAQ section for the existing tenants.	Easy	It was discovered that this would not be required during implementation.	
11	Review the website's usability (in further detail).	Moderate	This has been done using the website checklist.	✓
12	Host the website on the Internet.	Easy	Found at http://www.urbanliftyleletting.org.uk	✓
13	Consider possible security flaws in the system (in further detail).	Complicated	Partially completed by looking into security flaws of PHP and SQL injection.	✓
14	Make relevant amendments to the website so that it complies as closely as possible with the World Wide Web Consortium's (W3C) [76] regulations and is compatible with the major web browsers.	Moderate	Completed using the tool found at http://validator.w3.org/	✓

6.5 – Deliverables

The following three deliverables were required to be produced and submitted to Urban Lifestyle Letting.

1. The final report of the project – **Completed**

2. A fully working web-based system that at the very least meets the minimum requirements previously stated – **Completed – Found at http://www.urbanlifestyleletting.org.uk**

3. A fully documented user-manual – **Completed – Found in APPENDIX ?**

6.6 – Technology/Tools Selection

PHP and MySQL have proved to excellent software development tools due to the high quality of the solution produced. The error messages and online documentation have aided the implementation process and solved many difficult problems. The versatility and usability of

JavaScript, used mostly in validating input forms, proved to be a good choice as the universal use of the language meant that any difficulties could be easily rectified.

6.7 – Time Management

Comparing the initial project schedule found in section 1.8 and the actual schedule found in Appendix B, it is clearly noticeable that the project went relatively smoothly. The system developer was away from university for a month, due to surgery, beginning the 2nd November and therefore the three weeks that were designated for "Overflow" proved to be a major underlining factor in the overall success of the project. For that reason, the project schedule has illustrated why it is an irreplaceable feature in any software development project.

6.8 – User Evaluation

It was very important to conduct a user evaluation [77] in order to discover how effective and useful the implemented system is deemed to be. Moreover, it was decided that the highly recommended [78] focus group tool would be used to conduct the qualitative research into extracting valuable opinions of the system.

Focus groups also appeared to be the pre-eminent method of evaluating the automated live help feature. This is because the automated live help feature is a prototype and the discussion would bring new ideas which could improve on the program in the future, possible leading to it becoming a commercial commodity.

The system that has been created has three generally different aspects which all needed to be evaluated. The public website, the private password protected area and the automated live help feature will be used by users of varying levels of IT skills and therefore it was extremely important to invite specific individuals [79] to the focus group in order to reflect this occurrence. As a heterogeneous group of participants has been decided, it is thought by a number of researchers [80] that the group must consist of at least 6 people in order to have a lively discussion and less than 12 in order to involve each person.

Eight participants were invited to the focus group with four being students from Leeds University School of Computing, in order to discuss the more technical aspects, and four members of the general public who could possible highlight a number of issues that possibly may be overlooked.

Each of the eight participants were asked to perform the following tasks before attending the focus group in order to become familiar with the system and have an opinion that could be voiced. The tasks were as follows:

- Navigate through the public website
- Search for a property using the quick search feature
- Send an email to Urban Lifestyle Letting using the quick contact form found on each page
- Have a number of conversations with the automated live help feature
- Access the private pages using the username "pentagon52" and password "135redplanet13"
- Add, edit and delete properties
- Search and view a property
- Add, edit and delete tenant
- Search and view a tenant

The responses gained from the focus group on the whole tended to be positive. Everybody involved in the focus group agreed that the layout of the website was very professional and allowed easy navigation. A number of individuals commented on the contact details of the company being visible on each page being a desirable feature as it could possible encourage somebody to contact Urban Lifestyle Letting there and then. Furthermore, the quick search feature proved to be very popular with it "allowing me to find the property that I was looking for". However, one of the participants expressed that it could be enhanced by incorporating more variables into the search criteria therefore narrowing down the number of properties found.

The three participants of the focus group who were considered to be less experienced with information systems proved to be very important when evaluating the user interface and navigability of the private area. All three agreed that the private areas were easy to navigate due to the menu buttons and that the input forms were self explanatory. The participants from the SOC provided valuable input regarding the more technical aspects. One participant mentioned that the "JavaScript validation worked well…but maybe more drop down menus, radio buttons etc. could be used to limit any input errors."

Many interesting responses were taken from the discussion on the prototype automated live help feature. Everybody agreed that the feature was an excellent idea and one individual from the SOC mentioned "…that if it was professionally developed it could provide a solution to many online websites". Furthermore, the majority of the participants were impressed with the way the automated member of staff asked a series of questions and produced a number of properties that would possibly suit them. One limitation of the automated live help feature was regarding the matching of key phrases to key responses, as he found that by entering a certain phrase produced an irrelevant response.

In conclusion, all of the participants present agreed that the system was good enough to "go live" and be a major success with Urban Lifestyle Letting's day-to-day operations. Regarding the automated live help feature, all agreed that it had a number of excellent attributes and if fine tuned in a number of areas it could be used commercially.

6.9 – Methodology Chosen

The RAD methodology worked extremely well as it allowed the developer to produce a number of iterations of the system to show the end user. The end user attempted to quickly respond with feedback and the product continued to evolve in this manner. The UML models that were produced during the analysis and design stages, aided in the dissemination of problems so that they could be intuitively resolved.

6.10– User Satisfaction

The Director of Urban Lifestyle Letting Mr Damon Brown, expressed his delight with the system created and quoted "…the new web system will improve the efficiency of the company and allow us to spend more time on taking the company forward. Thanks for all your hard work!"

6.11 – Summary of the Evaluation

This chapter was incorporated to fully and comprehensively discover how successful the system and the overall process has been. The Evaluation has successfully identified a number of areas that could possible be expanded upon or done in a completely different manner. Therefore, the Evaluation could be considered as being a vital in order to learn a number of lessons to be applied in the future.

CHAPTER 7 – FURTHER DEVELOPMENTS AND CONCLUSIONS

7.1 – Further Developments

The web system itself was designed and implemented in order to be used by the staff at Urban Lifestyle Letting. However, the AutoStaff "Live Help" feature is an initial prototype that can be fully implemented in the future. As with Joseph Weizenbaum's Eliza [81] program, it is relatively simple to tell that one is not speaking to a real person as the program only uses basic tricks such as string substitution and canned responses based on keywords. As explained in the design, PHP was chosen as the platform to produce the application. The major factor that made it appear that the customer was not speaking to a real person was the instant reply. It would have been possible to set a time delay, using the code below, using a client-side JavaScript in order to make it appear that the program was thinking.

```
function think(){
                document.Eliza.input.value = "";
                if( elizaresponse != "" ){ respond(); }
                else { setTimeout("think()", 250); }
        }
```

However, as explained previously in the implementation, this was not possible due to the need to use a Server-side scripting language in order connect to the MySQL database.

The automated live help feature could also be developed further by being incorporated with existing Live Help programs, allowing the company to switch on the automated version during evening and weekends.

7.2 – Conclusions

Following the results from the Evaluation in Chapter 6, it is fair to say that the project on Urban Lifestyle Letting has been very successful. Furthermore, all the Aims (**1.3.1 – Project Aim),** Objectives (**1.4 – Project Objectives**) and Requirements (**1.5.1 – Minimum Functional Requirements**) have been accomplished to a very high standard.

BIBIOGRAPHY

[1] Journal of Electronic Commerce in Organizations. Hershey: Oct-Dec 2004. Vol. 2, Iss. 4; pg. 20

[2] http://www.urban-lifestyle.co.uk

[3] Luiz Marcio Cysneiros, Julio Cesar Sampaio do Prado Leite. IEEE Transactions on Software Engineering. New York: May 2004. Vol.30, Iss. 5; pg. 328

[4] Ward and Pepper: Strategic Planning for Information Systems 2003

[5] http:// www.w3c.org

[6] Network World. Framingham: Oct 18, 2004. Vol. 21, Iss. 42; pg. 26

[7] Macworld. San Francisco: Jan 2004. Vol. 21, Iss. 1; pg. 26

[8] Richard V. Dragan. [0][0]PC Magazine[0]. [0]New York:[0][0] Nov 16, 2004[0].[0] Vol.23[0][0], Iss. 20; [0][0] pg. 101

[9] http://dev.mysql.com/downloads/

[10] http://www.bellsupportweb.ca/glossary.htm

[11] Computing Canada. Willowdale: Jul 27, 2001. Vol. 27, Iss. 16; pg. 20

[12] http://www.spellsoftware.com/ss/shop/pc/viewPrd.asp?pid=2021

[13] http://www.postgresql.org/docs/7.4/static/preface.html#INTRO-WHATIS

[14] Computerworld. Framingham: Sep 10, 2001. Vol. 35, Iss. 37; pg. 48

[15] CRN. Jericho: Oct 18, 2004. , Iss. 1117; pg. 59

[16] Security Management. Arlington: Sep 2004. Vol. 48, Iss. 9; pg. 38

[17] http://www.javascriptkit.com/javatutors/vbalert3.shtml

[18] Computerworld. Framingham: Nov 29, 2004. Vol. 38, Iss. 48; pg. 1

[19] Montealegre R & Keil M, De-escalating Information Technology Projects: Lessons from the Denver International Airport, MIS Quarterly, Vol 24, No. 3, Sept 2000, pp.417-447

[20] Applied Software Measurement: Assuring Productivity and Quality, Capers Jones, McGraw Hill 1997

[21] Information systems project management: The price of failure: John McManus, Trevor Wood-Harper. Management Services. Enfield: May 2003.Vol.47, Iss. 5; pg. 16

[22] Donaldson A.J.M., Narrative Case Study of the Denver Airport Baggage Handling System, SFC TR 2002-01, May 2002.

[23] CRM Meets Business Intelligence: James R Borck. InfoWorld. San Mateo: Jan 10, 2005.Vol.27, Iss. 2; pg. 39, 1 pgs

[24] Strategic Planning for Information Systems; Ward 2002

[25] Johnson & Scholes: Investigating Corporate Strategy

[26] Paul N Romani. The American Salesman. Burlington: Mar 2005.Vol.50, Iss. 3; pg. 3, 8 pgs

[27] Mark Gabbott, Chris Dubelaar, Yelena Tsarenko. Journal of Asia Pacific Marketing. Patrington: 2003.Vol.2, Iss. 2; pg. 42

[28] S Srinivasan. Information Management & Computer Security. Bradford: 2004.Vol.12, Iss. 1; pg. 66

[29] W.S. Whyte: Enabling E-Business

[30] http://www.zonalatina.com/Zldata386.htm

[31] Consider three-tier client/server: Stevens, Larry: Datamation; Feb 15, 1996; 42, 4

[32] DiscMath II intelligent tutoring system: Middle tier design and implementation by Jiang, Yajie, M.C.S., The University of New Brunswick (Canada), 2000, 109 pages; AAT MQ62130

[33] Strategic directions in human-computer interaction Brad Myers, Jim Hollan, Isabel Cruz. ACM Computing Surveys. Baltimore: Dec 1996.Vol.28, Iss. 4; pg. 794, 16 pgs

[34] Sutcliffe, Alistair (1988): Some Experiences in Integrating Specification of Human Computer Interaction within a Structured System Development Method

[35] Palmer, Terry. Systems International. London: March 1987. Vol 15, iss 3; pg.91, 2pgs

[36] http://ezinarticles.com/?On-the-Importance-Good-User-Interface-Design

[37] http://www.agilemodeling.com/style/classDiagram.htm

[38] Larry Ullman: MySQL: Visual QuickStart Guide; 2003

[39] E.F. Codd: Relational Model of Data for Large Shared Data Banks

[40] Fundamentals of Database Systems – 2003 – Elmasri and Navathe

[41] Fundamentals of Database Systems – 2003 – Elmasri and Navathe

[42] Fundamentals of Database Systems – 2003 – Elmasri and Navathe

[43] MySQL: Visual Quick Start Guide: Larry Ullman: 2003

[44] http://www.apache.org

[45] http://www.php.net

[46] http://www.mysql.com

[47] http://www.streamline.net

[48] www.websearchworkshop.co.uk/glossary.htm

[49] staff.washington.edu/rells/topics/frames.html

[50] www.azatiko.com/glossary/f.php

[51] http://www.macromedia.com/support/dreamweaver/layout/frames_or_tables/fr

ames_or_tables04.html

[52] Gunning for Google: . Catalog Age. New Canaan: Feb 2005. Vol.22, Iss. 2; pg. 0_1, 2 pgs

[53] The impact of metadata implementation on webpage visibility in search engine results (Part II) Jin Zhang Alexandra Dimitroff. Information Processing & Management. Oxford: May 2005.Vol.41, Iss. 3; pg. 691

[54] http://www.robotstxt.org/wc/norobots.html

[55] http://www.mysql.com/products/mysqlcc/

[56] http://www.aota.net/forums/printthread.php?t=18784

[57] http://www.kuro5hin.org/story/2003/12/16/114327/74

[58] http://www.techdose.com/tutorials/php/PHPLiza/

[59] Got Broadband? Matthew Haeberle. Chain Store Age. New York: Jan 2005.Vol.81, Iss. 1; pg. 64, 1 pgs

[60] More testing should be taught Terry Shepard, Margaret Lamb, Diane Kelly. Association for Computing Machinery. Communications of the ACM. New York: Jun 2001.Vol.44, Iss. 6; pg. 103, 6 pgs

[61] Neglect testing at your own peril Patrick D T O'Connor. IEEE Spectrum. New York: Jul 2001.Vol.38, Iss. 7; pg. 18

[62] The three faces of testing Linda Hayes. Datamation. Barrington: Sep 1997.Vol.43, Iss. 9; pg. 32, 1 pgs

[63] Development and Application of a White Box Approach to Integration Testing Haley, Allen, Zweben, Stuart. The Journal of Systems and Software. New York: Nov 1984.Vol.4, Iss. 4; pg. 309, 7 pgs

[64] www.scism.sbu.ac.uk/law/Section5/chap3/s5c3p23.html

[65] Hidden rewards of your Y2K efforts: IT objectives are more easily attained Steve Jeffries, Steve Robertson. Apparel Industry Magazine. Atlanta: Jun 1999.Vol.60, Iss. 6; pg. 70, 2 pgs

[66] RUN AND GROWTH THROUGH BOUNDARY TESTING MCCONNELL, CHARLES R.. Manage. Dayton: MAR./APR. 1977.Vol.29, Iss. 2; pg. 14

[67] Unit Testing Emerges As Software Necessity Scott Thomas. Wireless Systems Design. Cleveland: Jun 2004.Vol.9, Iss. 5; pg. 13, 1 pgs

[68] Testing, testing... a rigorous process Roger Fournier. InformationWeek. Manhasset: Apr 27, 1998., Iss. 679; pg. 8A, 8 pgs

[69] Building a better Website; Here are some tips from legal technology consultants about how to get the most from your law practice's Website. YOUR WEBSITE CHECKLIST 1. Be careful with graphics. 2. Provide the basics. 3. Strut your stuff. 4. Structure for clients. 5. Design for clients, too. 6. Avoid legalese. 7. Be a

content provider. 8. It's a marketing tool. 9. Grow your Website Jean Cumming. National. Ottawa: Nov 2002.Vol.11, Iss. 7; pg. 43, 3 pgs

[70] Year 2000 capacity and performance issues Ellen M Friedman, Jerry L Rosenberg. Capacity Management Review. Naples: Sep 1997.Vol.25, Iss. 9; pg. 1, 11 pgs

[71] http://www.websiteoptimization.com/services/analyze/wso.php

[72] Testing, testing... a rigorous process Roger Fournier. InformationWeek. Manhasset: Apr 27, 1998., Iss. 679; pg. 8A, 8 pgs

[73] Juggling act: Bill Roberts. Electronic Business. Highlands Ranch: May 2001.Vol.27, Iss. 5; pg. 54, 1 pgs

[74] http://www.apache.org

[75] System evaluation and quality improvement Boloix, Germinal. The Journal of Systems and Software. New York: Mar 1997.Vol.36, Iss. 3; pg. 297, 15 pgs

[76] www.w3c.org

[77] Design and evaulation of a task-based digital library for the academic community N Meyyappan, Schubert Foo, G G Chowdhury. Journal of Documentation. Bradford: 2004.Vol.60, Iss. 4; pg. 449

[78] Best practices: Choosing the right methods for evaluation Kelly Hannum. Leadership in Action. San Francisco: Jan/Feb 2004. Vol. 23, Iss. 6; p. 15 (6 pages)

[79] Miles, M. B., & Huberman, A. M. (1984). Qualitative data analysis: A sourcebook of new methods. Beverly Hills, CA: Sage.

[80] Brown, J. B. (1999). The use of focus groups in clinical research. In (Eds.) Crabtree, B. F., & Miller, William L. Doing qualitative research (2nd ed.) (pp. 109-124). Thousand Oaks: Sage.

[81] www-ai.ijs.si/eliza/eliza.html

APPENDIX A – PERSONAL REFLECTION

In reflection, I believe that the production of this project has on the whole been a major success. Looking at the final system and the documentation I believe that I have followed the initial information, given by Urban Lifestyle Letting, extremely well and successfully produced a fully working system that meets the user's requirements and passed all the acceptance tests.

I feel the key to the success of the project was that I started immediately back in September and continued to work throughout the whole period at a steady pace, which resulted in nothing being left to the last minute. Furthermore, it was crucial that I began the project immediately as unfortunately I had an operation in November which resulted in not being able to work on the project for a month. Therefore, the three weeks that I designated for "Overflow" in the initial project schedule proved to be priceless due to the unexpected period away from university. Admittedly, the final week was a little hectic as new tasks were discovered either unfinished or overlooked, added with the collation of the whole project made it all quite strenuous. The standard of work I feel is high and represents well the amount of time and effort that has been invested in this project.

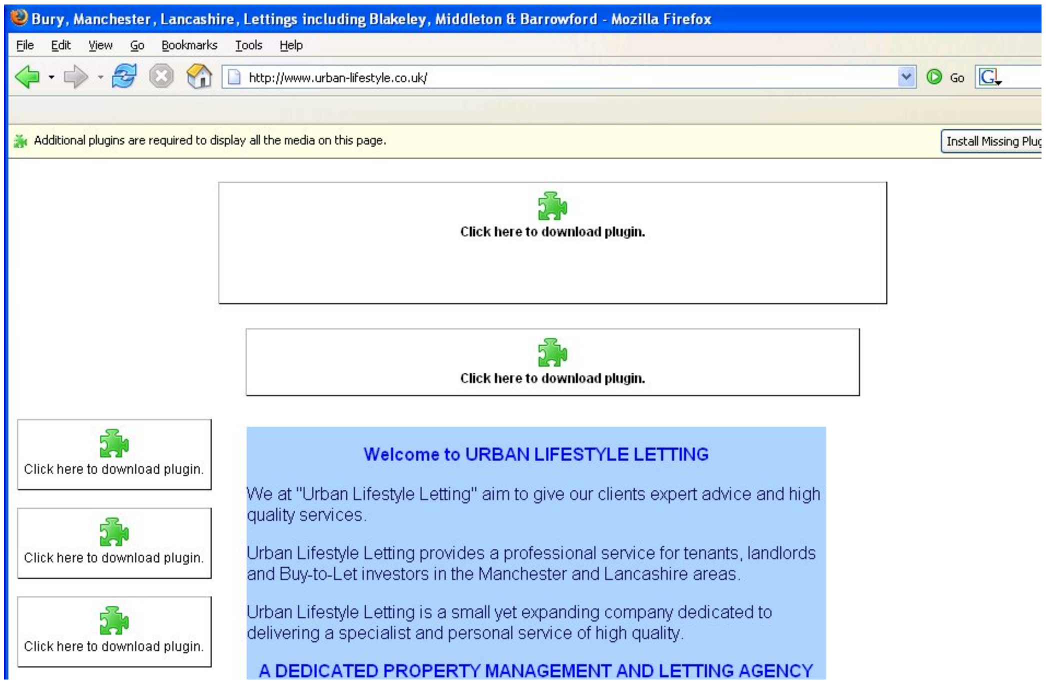
Time management was another key area that was crucial to the success of the project. If I started to get behind at the beginning it would have been possible for the work to mount up and cause serious problems towards the final weeks. In my opinion, all the work that has been produced in this project was required in order to produce the system to meet Urban Lifestyle Letting's needs.

Overall, the experience that I have gained from doing this project has been immense. I have learnt the value of good communication with the end-user, focus group participants, project supervisor and assessor. I also learnt that, unlike other university coursework's, you cannot get every little detail 100% right due to the time restriction. There comes a moment where you must move on even if an activity is not completed to total satisfaction. It has been a difficult seven months of work, but looking at what I have achieved and the skills that I have developed I feel that it has been a success and I am proud of what I have done.

I would advise all future students to start early, work consistently, communicate effectively with their supervisor and concentrate initially on passing the minimum requirements. It is imperative for an individual not to over face themselves.

APPENDIX B

Screen Capture of Urban Lifestyle Letting's Previous Website.


Bury, Manchester, Lancashire, Lettings including Blakeley, Middleton & Barrowford - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.urban-lifestyle.co.uk/
Go
Additional plugins are required to display all the media on this page.
Install Missing Plug
Click here to download plugin.

Welcome to URBAN LIFESTYLE LETTING

We at "Urban Lifestyle Letting" aim to give our clients expert advice and high quality services.

Urban Lifestyle Letting provides a professional service for tenants, landlords and Buy-to-Let investors in the Manchester and Lancashire areas.

Urban Lifestyle Letting is a small yet expanding company dedicated to delivering a specialist and personal service of high quality.

A DEDICATED PROPERTY MANAGEMENT AND LETTING AGENCY

[2]

Increasing Size of a Project Equals an Increasing Chance of Failure

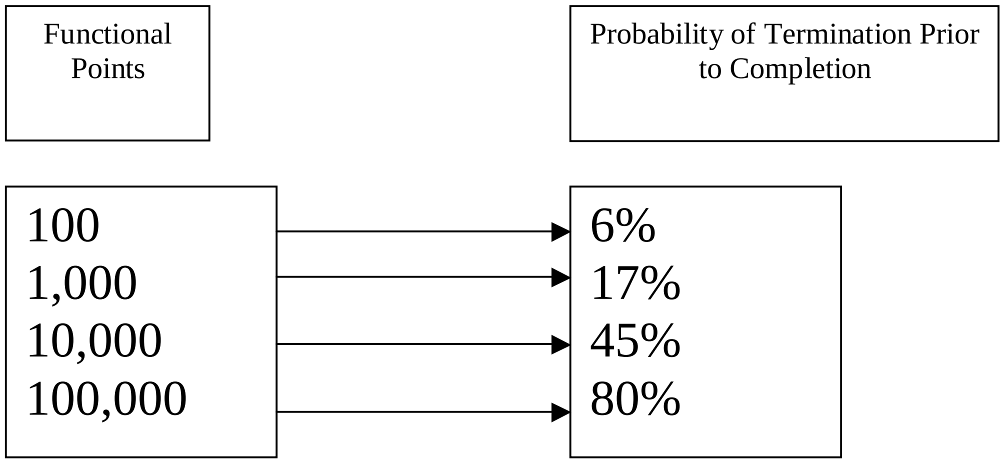


Figure 2 [20]

Risk and mitigation

Reference No.	Risk	Mitigation
1.1	Either not understanding the problem or	**Regular communication with**

	misinterpreting it.	**the end-user of the system**.
1.2	Missing out vital activities.	**Constant revision of the activities proposed.**
1.3	Setting unrealistic targets and/or resource limits.	**As above.**
1.4	Unrealistic allocation of time to an activity.	**Do preliminary time charts and then constantly revise and monitor them.**
1.5	Being over-cautious leading to wasting time on the project.	**Be reasonable and focus at the task at hand.**

System Requirements Specification

Reference No.	Risk	Mitigation
2.1	Generally misinterpreting or forgetting any user requirements.	**A brainstorming session with the end-user would cover most if not all aspects of the requirements.**
2.2	Incorrect UML models/diagrams. This includes incorrect syntax or simply incorrect or missed out sections.	**Once the diagrams have been drawn they need to be printed out in Rational Rose. This will highlight any syntax errors and will also clear up any faults.**

Design of the System

Reference No.	Risk	Mitigation
3.1	Incorrect attributes or missing out vital attributes.	**Thoroughly following on from initial requirements specification. I.e. keeping the design level class diagrams as similar as possible to the concept level class diagram.**
3.2	Incorrect methods or missing out vital methods.	**As above.**
3.3	Identifying the flow of events incorrectly.	**Re-assess with the end-user.**
3.4	Incorrect actors identified or missed out.	**As above.**
3.5	Ambiguous identification of software and hardware to be used.	**Don't be too precise but ensure that a certain degree of detail is attained.**
3.6	Interface that does not include all the required functions.	**This will be highlighted in testing but can be avoided by cross-referencing with the system requirements.**
3.7	Assigning incorrect data structure. I.e. too much validation, allowing NULL, etc.	**Simply be reasonable and sensible with validation.**
3.8	Selecting inappropriate software tools.	**Ensure adequate research is carried out and that all possible options are considered.**
3.9	Misunderstanding of the system design.	**Reassess the situation at hand.**
3.10	Too much alteration of the design can waste resources such as time.	**Only make necessary changes.**

Implementation/Testing

Reference No.	Risk	Mitigation
4.1	Unable to implement the functions that	**This will NOT happen as the**

	have been designed.	**design will be not over-complicated.**
4.2	Creating a test plan that does not cover all aspects of the system. However, to a certain extent it will be impossible to cover ALL aspects.	**Ensure the major aspects of the system are fully tested.**
4.3	Incorrect recording of results leading to faults in the system.	**Full attention to the task at hand should combat this issue.**
4.4	Ensure that any modifications will not affect other modules in the system.	**Ensure that if/when any modifications are made that they are saved under a new version name. This will then allow the referring back to the old system.**

Integration

Reference No.	Risk	Mitigation
5.1	Over modification leading to waste of resources and also slippage away from the project time plan.	**This will NOT happen as the design will be not over-complicated.**
5.2	Creating a test plan that does not cover all aspects of the system. However, to a certain extent it will be impossible to cover ALL aspects.	**Discuss thoroughly with the end-user and decide if the modifications are essential or not.**
5.3	Putting too much time into the User Manual thus wasting resources.	**Interaction with the end-user to discover their level of literacy.**
5.4	Missing out any resources that have been used throughout the duration of the project.	**Keeping details on all forms of resources as the project progresses.**
5.5	Future changes need to be made to the system by another programmer.	**Comment all coding fully and use meaningful variable names.**

Diagram Showing the Business Use Case for Urban Lifestyle Letting

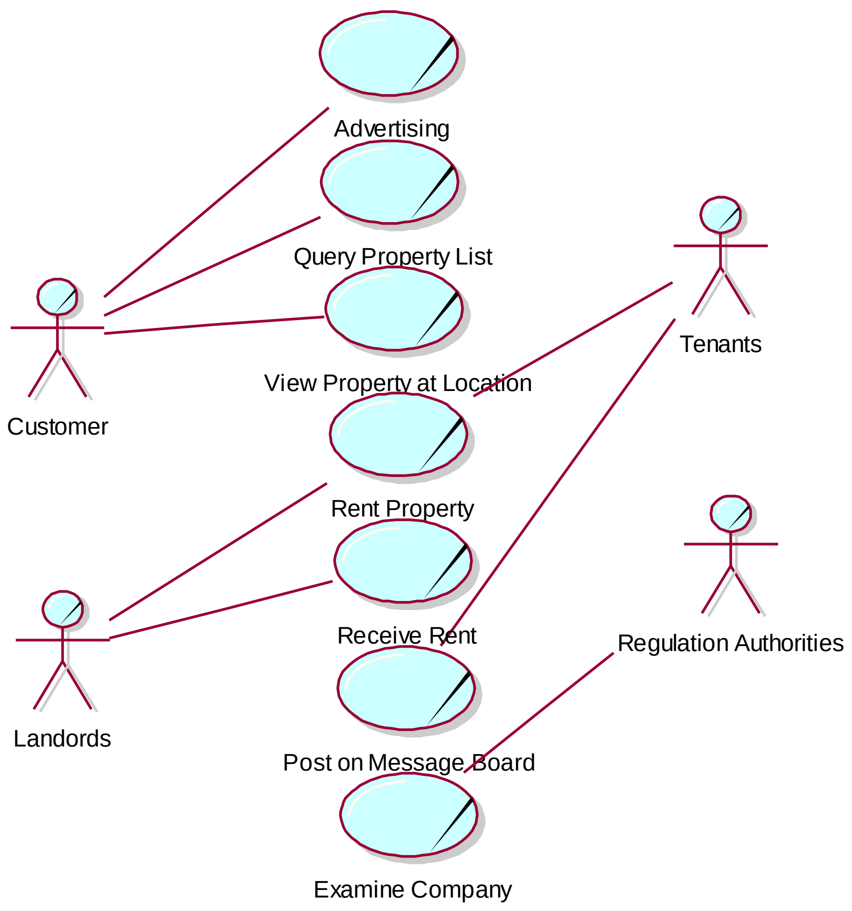
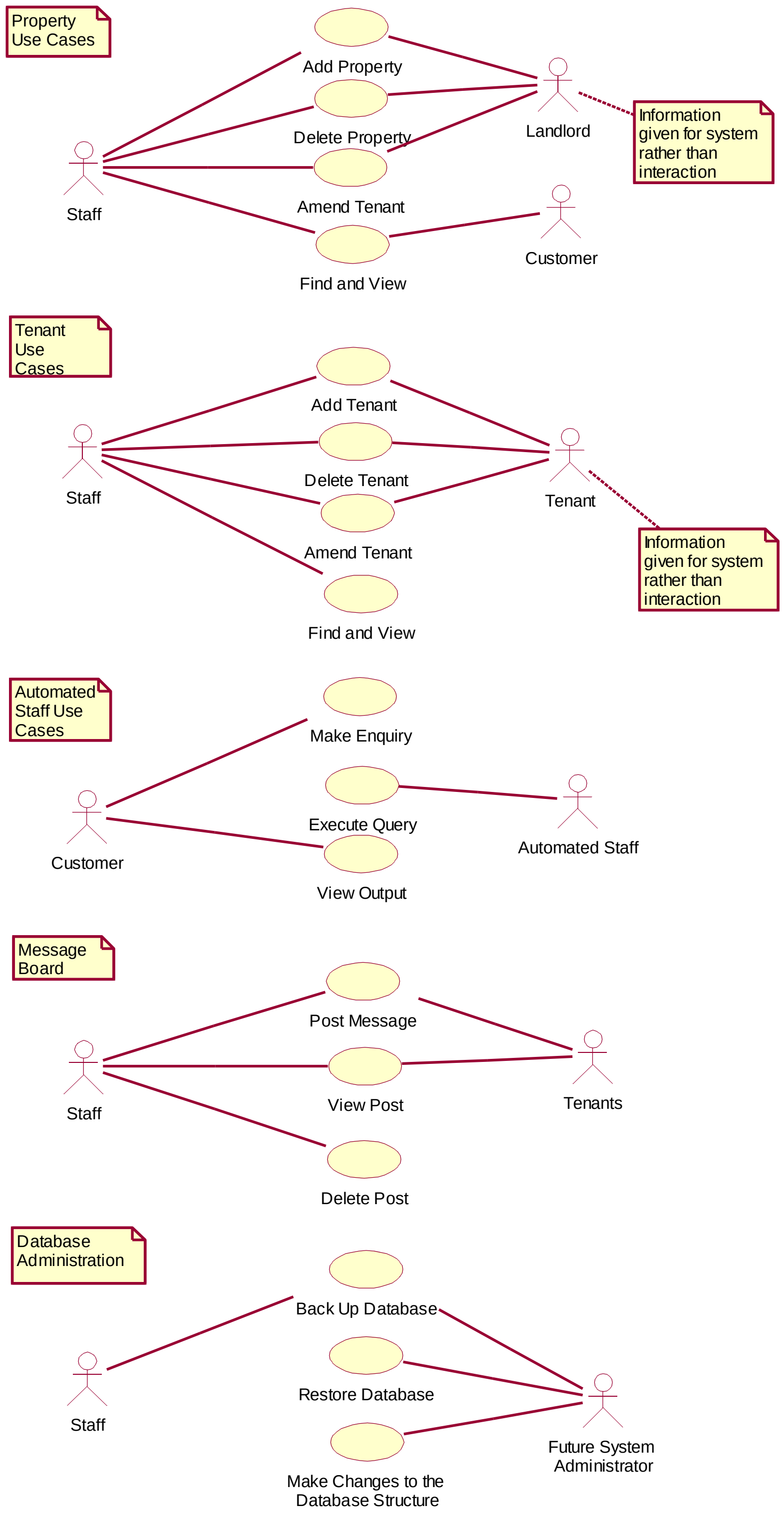


Diagram Showing the Use Case Diagrams for the System Development



Graph Showing the Cumulative Internet Usage by Time of Day and Place of Usage.

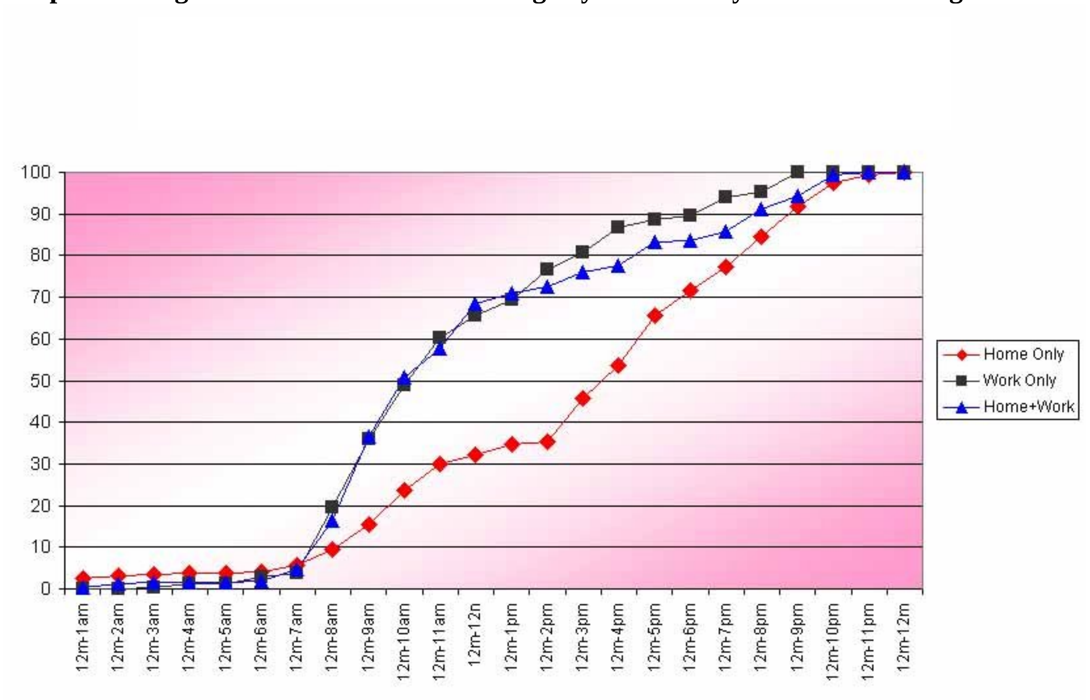


Figure 6 [30]

1.8.1 – RevisedSchedule

No.	Activity	Date: from – to	Milestone/Major Deliverable
1	**Submit Project Preference Form**	**23/09/04 – 30/09/04**	✓
2	Discuss idea with supervisor	035/10/04	
3	Research	04/10/04 – 10/04/05	
4	**Submit Aims and Minimum Requirements**	**10/10/04 – 22.10.04**	✓
5	Interview with the end-user	25/10/04	
6	Compose user-requirements	16/10/04 – 23/10/04	
7	Review appropriate technologies and methodologies	10/11/04 – 10/12/04	
8	**Submit Mid-Project Report**	**01/12/04 – 10.12.04**	✓
9	End of Semester 1 - Christmas Holiday – Exam Period	10/12/04 – 21/01/05	
10	Design the system to be created	11/12/04 – 20/01/05	

	using the iterative approach continuously consulting with the end-user		
11	Prepare materials for the focus group	22/01/05 – 23/01/05	
12	Conduct focus group	23/04/05	
13 14	Consultation with the end-user Finalise design and agree on usability	24/01/05 25/01/05	
15	Start developing the Minimum Requirements of the system	01/02/05 – 22/02/05	
16	Start developing the Project Extensions of the system	22/02/05 – 17/03/05	
17	**Submit table of contents and draft chapter**	**03/03/05 – 11/03/05**	✓
18	**Prepare and conduct the progress meeting**	**14/03/05 – 18/03/05**	✓
19	Begin evaluating the system	22/03/05 – 26/03/05	
20	Complete all development and evaluation aspects	19/03/05 – 07/04/05	
21	OVER-FLOW	04/04/05 – 20/04/05	
22	Proof-read and collate the final report.	25/04/05 – 26/04/05	
23	**Submit final report**	**27/04/05**	✓

APPENDIX D

Acceptance Testing

<u>Acceptance Test Plan</u>

Test 1

Load web page

Login to the private area

Pass Criteria

Web page should successfully load

After entering a valid username and password the user should be taken to the main menu

Test 2

Add a new property to the system

Amend the details of that property

Search and list the information on that property

Delete that property

Print list of properties

Pass Criteria

Successful addition of a new property

The details of that particular property should be successfully changed

The relevant property should be found and its details listed

The property should be successfully removed from the system

Test 3

Add a new tenant to the system

Amend the details of that tenant

Search and list the information on that tenant

Delete that tenant

Print list of tenants

Pass Criteria

Successful addition of a new tenant

The details of that particular tenant should be successfully changed

The relevant tenant should be found and its details listed

The tenant should be successfully removed from the system

Test 4

Backup Database

Restore Database

Pass Criteria

A physical backup of the database and data stored in the tables must be produced

The required backup must be able to be selected and restored to the web host database

Test 5

View a message forum

Make a post on that message forum

Pass Criteria

The required message forum must be able to be selected and shown

Successful post made on the message forum

Test 6

Add a user to the system

Amend the details of that user

Find and show a user's username and password

Remove a user

Pass Criteria

Successful addition of a user to the system

The details of that particular user should be successfully changed

The relevant user should be found and its details listed

The user should be successfully removed from the system

User Documentation

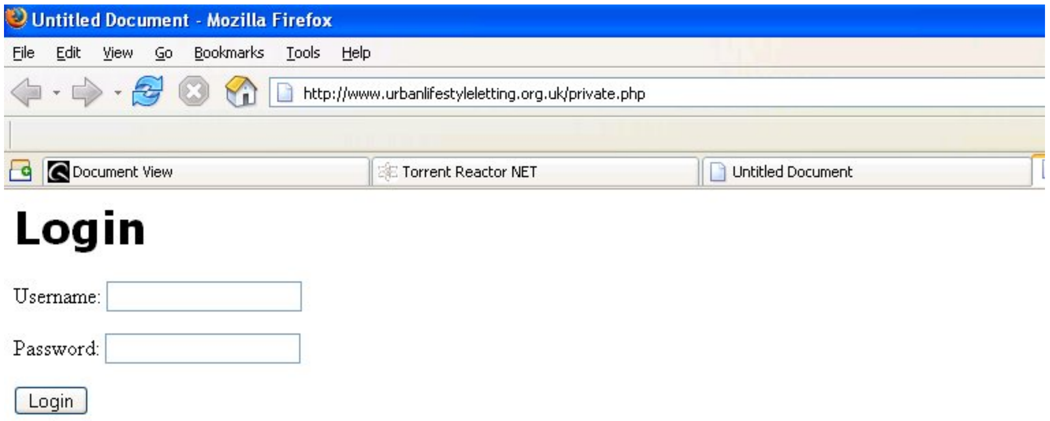
How to Use the System

The screen-shot below is the opening page that the user is greeted with when http://www.urbanlifestyleletting.org.uk/admin.php is entered into a web browser, such as Internet Explorer, and a valid username and password has been accepted. Each of the buttons (e.g. "Add Property") represents other individual pages which provide different operations. For example, if you wanted to add a property to the system you would simply click on the "Add Property" button and it will take you to the page.

It is assumed that the users of this system have basic I.T. skills and are able to use drop down menus, print out pages and other simple operations.

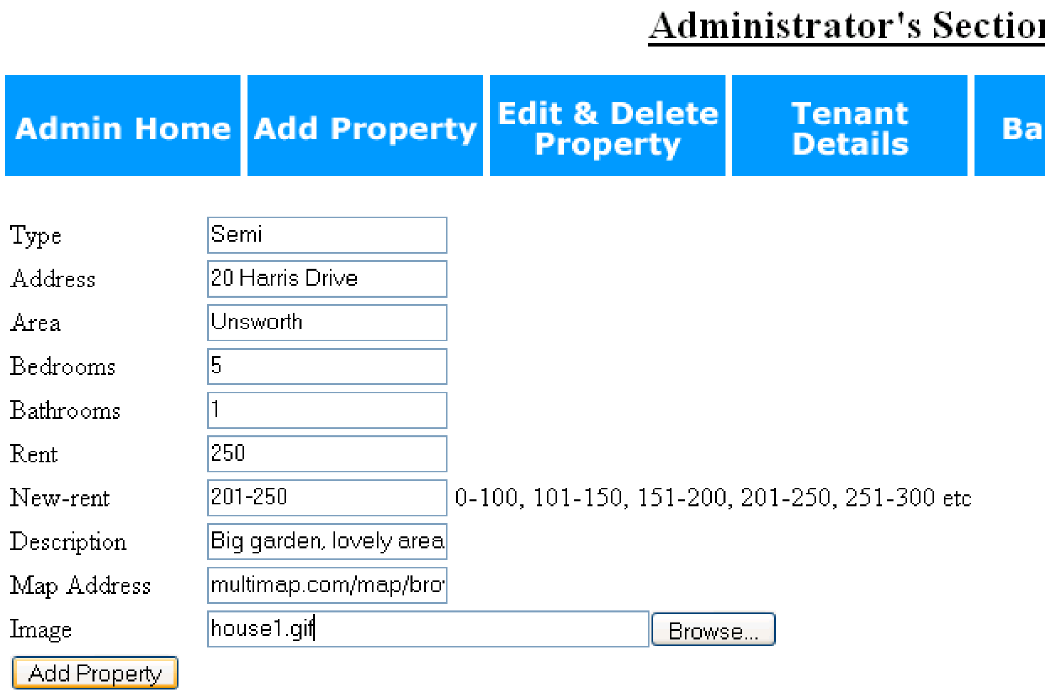
Access the System

To access the private system simply enter the URL http://www.urbanlifestyleletting.org.uk/private.php which will bring you to the password screen seen below. Enter your password and user and click on the "Login" button that will take you to the main menu.


Untitled Document - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.urbanlifestyleletting.org.uk/private.php
Document View
Torrent Reactor NET
Untitled Document
Login
Username:
Password:
Login

Add Property

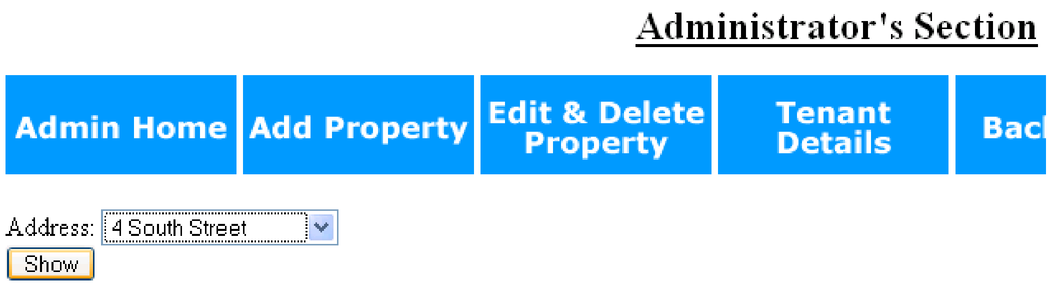
To add a property to the database, firstly click on the "Add Property" button that will take you to the screen below. Add the relevant details and simply click on the "Add Property" button at the bottom of the page.



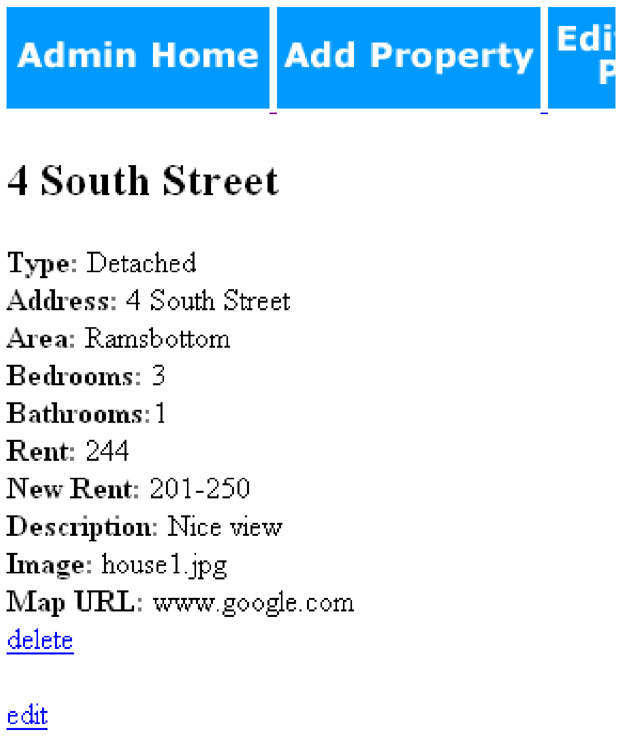
Type	Semi
Address	20 Harris Drive
Area	Unsworth
Bedrooms	5
Bathrooms	1
Rent	250
New-rent	201-250 0-100, 101-150, 151-200, 201-250, 251-300 etc
Description	Big garden, lovely area
Map Address	multimap.com/map/bro
Image	house1.gif

Search and View Details of Property

To find a specific property simply select the wanted property from the drop-down menu, seen below, and click on the “Show” button.



When the “Show” button has been clicked the relevant property will be shown in the following format seen below.



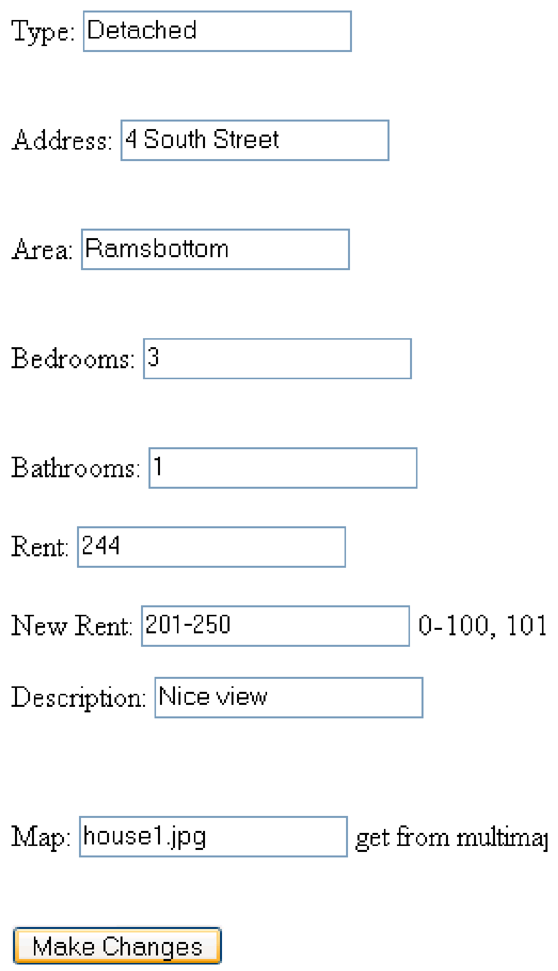
4 South Street

Type: Detached
Address: 4 South Street
Area: Ramsbottom
Bedrooms: 3
Bathrooms: 1
Rent: 244
New Rent: 201-250
Description: Nice view
Image: house1.jpg
Map URL: www.google.com
delete

edit

Edit & Delete Property

When the property has been selected and viewed, as seen in the previous screen capture, it can be deleted from the database, edited or nothing at all. In order to “Edit” the property simply click on the “Edit” button which will take you to the screen below. When you are happy with the changes that you have made click on the “Make Changes” button.



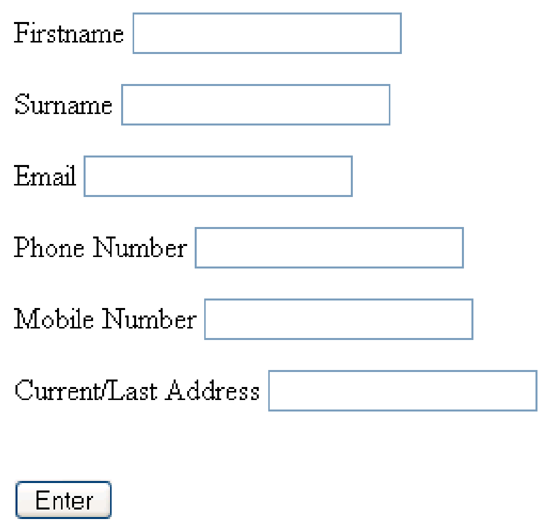
Type:	Detached
Address:	4 South Street
Area:	Ramsbottom
Bedrooms:	3
Bathrooms:	1
Rent:	244
New Rent:	201-250 0-100, 101
Description:	Nice view
Map:	house1.jpg get from multimaj

In order to "Delete" a property from the database simply click on the "Delete" button and then on the "Click here to permanently delete the property from the database" link.

<u>Tenant Details</u>

<u>Add Tenant</u>

To add a tenant to the database, firstly click on the "Tenants Details" and "Enter a New Contact Here!" buttons that will take you to the screen below. Add the relevant details and simply click on the "Enter" button at the bottom of the page.

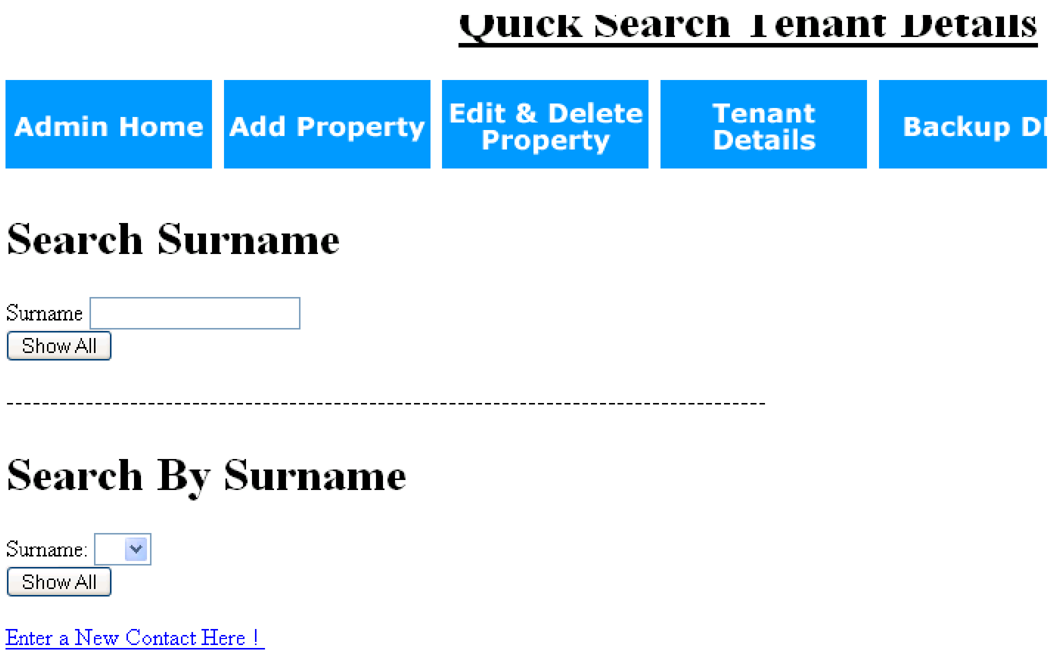


Firstname	
Surname	
Email	
Phone Number	
Mobile Number	
Current/Last Address	

Enter

Search and View Details of a Tenant

To find a specific tenant you can select the wanted tenant from the drop-down menu and click "Show All" or enter the name into the search box and click "Show All".



Quick Search Tenant Details

Search Surname

Surname

Show All

Search By Surname

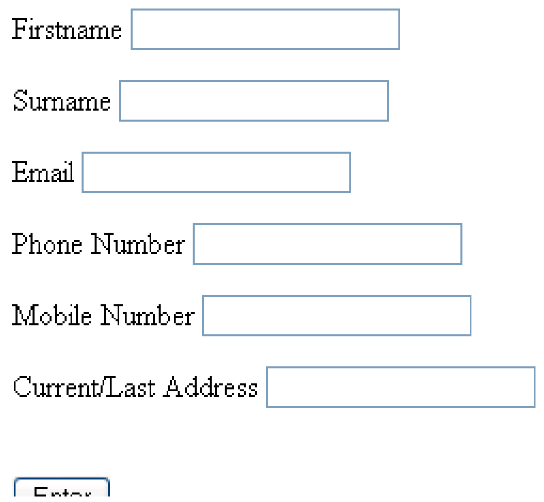
Surname:

Show All

Enter a New Contact Here !

Edit & Tenant

When the tenant has been selected and viewed it can be deleted from the database, edited or nothing at all. In order to "Edit" the tenant simply click on the "Edit" button which will take you to the screen below. When you are happy with the changes that you have made click on the "Enter" button.



Firstname

Surname

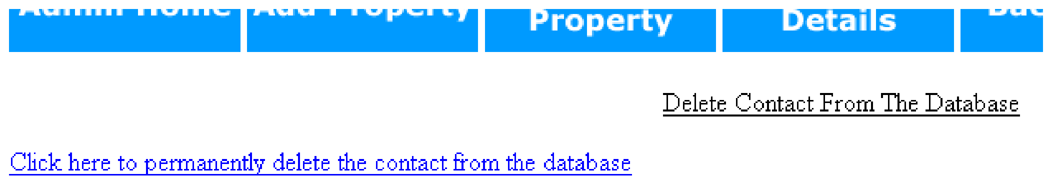
Email

Phone Number

Mobile Number

Current/Last Address

In order to “Delete” a tenant from the database simply click on the “Delete” button and then the “Click here to permanently delete the contact from the database” link.



Delete Contact From The Database

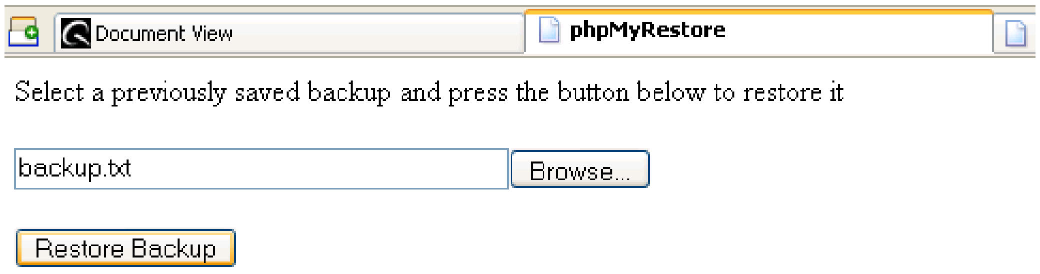
Click here to permanently delete the contact from the database

Backup Database

To backup the database, it is recommended that you do this at least twice a week, basically click on the “Backup DB” button on the main menu. A file called “backup.txt” will be produced on your web host. It is imperative that the individual who has access to the web server makes a copy of this file and stores it in a different location.

Restore Database – Only to be executed by the system administrator

To restore the file “backup.txt” to the database click on the “Restore DB” button on the main menu which will take you to the screen seen below. Click on the “browse” button and locate the file named “backup.txt” then select it. When you are content and certain that the correct file to be restored is selected click on the “Restore Backup” button.

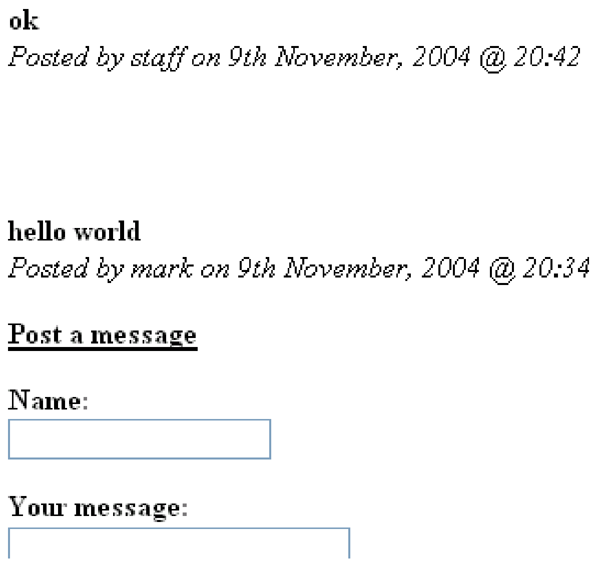


Select a previously saved backup and press the button below to restore it

backup.txt

<u>Message Forums</u>

To make a post on the message forum click on the "Message Forum" button on the main menu and select the forum you require. Enter "Staff" into the "Name" box and the post you wish to send in the "Your message" box and when you are happy with your message click on the "Post Message" button.



ok

Posted by staff on 9th November, 2004 @ 20:42

hello world

Posted by mark on 9th November, 2004 @ 20:34

Web Site Review Checklist

This checklist must be fully completed and analysed before any action is taken or any advice is given by the organisation. The purpose of this checklist is to effectively score a companies current web site in terms of it's effectiveness for marketing and sales. If you think an important factor has been missed off the list please contact Mark Brogan from the IT Department and he will consider your proposal. Please take care when reviewing the web site and make sure you have not missed anything out. Look at the following example of how to complete this check list:

	No				Yes
	0	1	2	3	4
E.g. Does the website have a bold, large, clear, defining title on the home page?					

- If the web site did not have any title whatsoever you would mark "0".
- If the web site had a title but it was small, vague and poorly positioned you would mark either "1" or "2".
- If the web site had a title that was bold, large, clear and defining on the home page you would mark a "4".
- If this questions was not relevant to the web site you were assessing mark "3".

N.B. Some of the questions may have a heavier weighting. See the example below:

	No				Yes
	2	4	6	8	10
E.g. Is the screen layout consistent?					

When you have marked ALL relevant questions, total up the scores (playing close attention to the weighting).

Assessor Name:

Date:
Company Name:
Company URL:

Section 1: The customer has expressed a "need" by visiting the web site. This section relates to whether or not the web site satisfies that "need".

	No				Yes	Comments
	0	1	2	3	4	
(A) Is it apparent from the web site's home page what information the web site contains?						
(B) Is it clear from the home page who the targeted audience (targeted market segment) is?						
(C) Is the information contained on the entire web site suitable for the targeted audience?						
(D) Is the companies name clearly visible on every page?						
(E) Is the web site free from irrelevant pop-ups?						
(F) Does the company's logo appear on each page?						
	0	4	6	8	14	
(G) Go onto www.google.co.uk and click on the "pages from the UK only" box. Now type in the name of the company and click "search". If the companies web site is top of the list in both "sponsored links" and the main list score "14". If the web site does not appear on any page at all score "0".						

Section 2: This section questions how well the web site allows the customer to find the relevant information that they require.

	No				Yes	Comments
	0	1	2	3	4	

(A) Is the web site in question visually stimulating to the user?					
(B) Do the buttons and/or links on the web site explain themselves clearly?					
(C) Does the web page fit exactly into the width of your computer screen without the need to scroll to the left and to the right?					
(D) Does the site contain a comprehensible “site map” to aid navigation?					
(E) Does the web site contain good links/buttons enabling the customer to go to any section of the web site (i.e. go back to the home page from the catalogue page)?					
(F) Does the web site use images appropriately?					
(G) Are colours used sensibly in a professional manner?					
(H) Is the site searchable through a search box?					
(I) Does the website have a multilingual option?					
	1	2	5	6	7
(J) Is the layout on the site consistent and easy to navigate?					
	0	2	4	6	8
(K) Does each page on the web site contain a “title tag” (title at the top of the web browser) that clearly describes each page?					

<u>Section 3: This section looks into how the web site in question enables the customer to access all possible offers, comparing and contrasting different products/services etc.</u>

	No				Yes	<u>Comments</u>
	1	2	5	6	7	
(A) Are all products, services etc. supplemented by a detailed explanation and prices (with and without VAT, RRP etc.)?						
	0	1	2	3	4	
(B) Is there an email link?						

(C) Is there a 24 hour phone line (“3” if open during office hours)?					
(D) Is there a “help” option on each page?					
(E) Do the images on the site enhance the content information?					
(F) Is the information obviously and clearly separated for different users of the web site?					

<u>Section 4: This section looks into how the web site in question allows the customer to make a decision.</u>

	No 0	1	2	3	Yes 4	<u>Comments</u>
(A) Can the customer compare products or services side-by-side?						
(B) Is there a section on the web site for “frequently asked questions”?						
(C) Is there an option to show what previous customers think of the product or service?						
(D) Is there an online “livehelp” feature where the customer can talk through instant messages to a member of staff?						

<u>Section 5: This section looks into how the web site in question allows the customer to actually make the purchase/action.</u>

	No 2	4	6	8	Yes 10	<u>Comments</u>
(A) Can multiple products be added to a single “basket”, single products deleted and can the customer continue shopping with items in the “basket”?						
	0	1	2	3	4	
(B) Is the registration process simple and contain relevant validation?						

(C) Do the web pages load quickly (2 seconds would be quick)					
	1	2	4	5	6
(D) Is the number of clicks needed to make a purchase/action kept to a minimum (3 clicks to make a purchase would deserve “6”)					

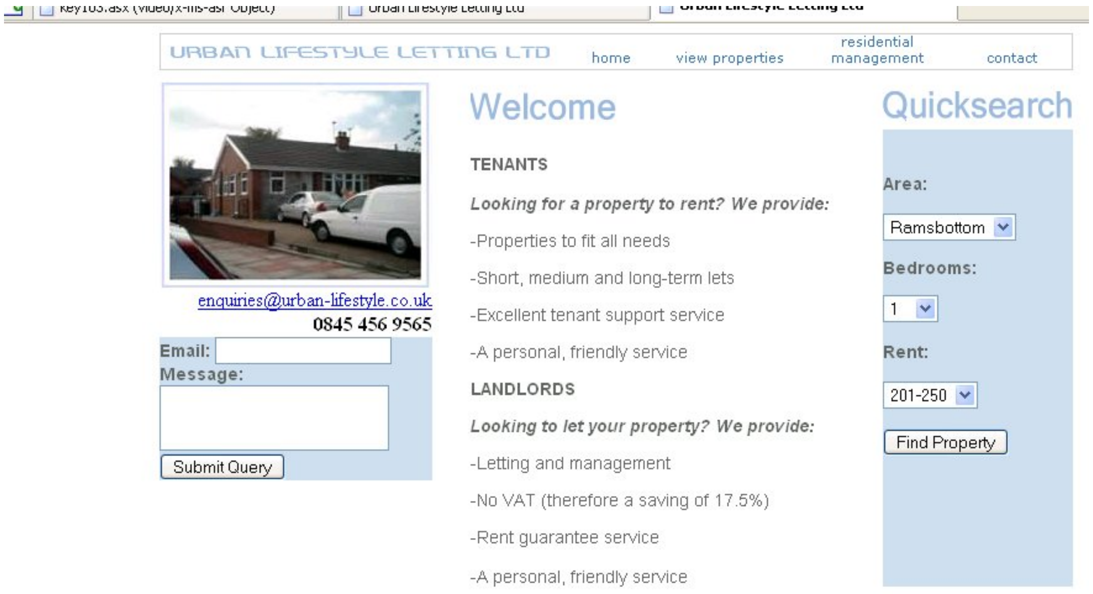
<u>Section 6: This section looks into how the customer can assess the results of the purchase/action taken.</u>

	No				Yes	<u>Comments</u>
	0	1	2	3	4	
(A) Is an automated email (autoresponder) sent to the customer as soon as a product or service has been delivered confirming the actions of the customer?						
(B) Can a user add themselves to a mailing list?						
(C) Can a customer login to the web site and check order progress or personal details?						
(D) Can a customer leave feedback?						
(E) Is the customer taken to a confirmation page when a product has been purchased or action has been taken?						
(F) Can the customer review their previous activity on the web site (i.e. bought a specific item on a specific date)						

Total Score:

Screen Captures

The screen below shows the opening page that greets the general public when they visit the company's website.



URBAN LIFESTYLE LETTING LTD | home | view properties | residential management | contact

enquiries@urban-lifestyle.co.uk
0845 456 9565

Email:
Message:
Submit Query

Welcome

TENANTS

Looking for a property to rent? We provide:

-Properties to fit all needs

-Short, medium and long-term lets

-Excellent tenant support service

-A personal, friendly service

LANDLORDS

Looking to let your property? We provide:

-Letting and management

-No VAT (therefore a saving of 17.5%)

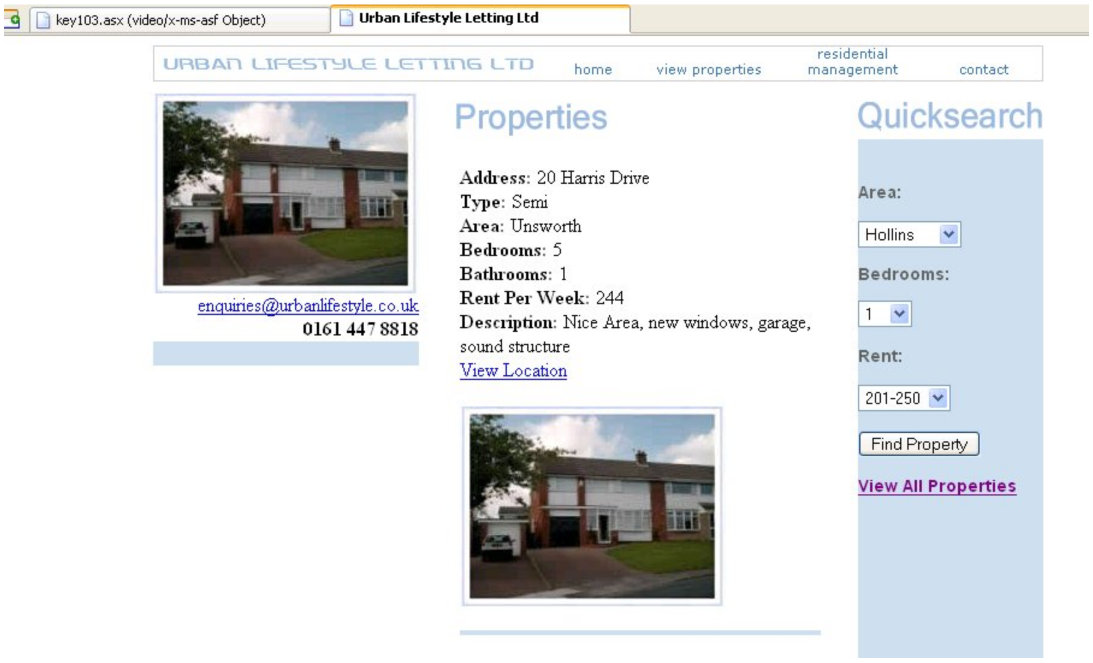
-Rent guarantee service

-A personal, friendly service

Quicksearch

Area: Ramsbottom
Bedrooms: 1
Rent: 201-250
Find Property

This screen shows the property illustrated following a query using the quick search feature.



key103.asx (video/x-ms-asf Object) | Urban Lifestyle Letting Ltd

URBAN LIFESTYLE LETTING LTD | home | view properties | residential management | contact

enquiries@urbanlifestyle.co.uk
0161 447 8818

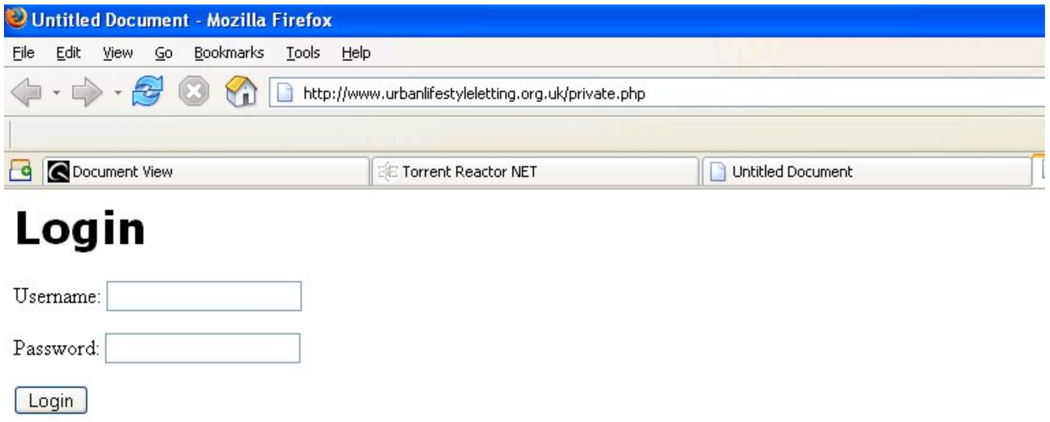
Properties

Address: 20 Harris Drive
Type: Semi
Area: Unsworth
Bedrooms: 5
Bathrooms: 1
Rent Per Week: 244
Description: Nice Area, new windows, garage, sound structure
View Location

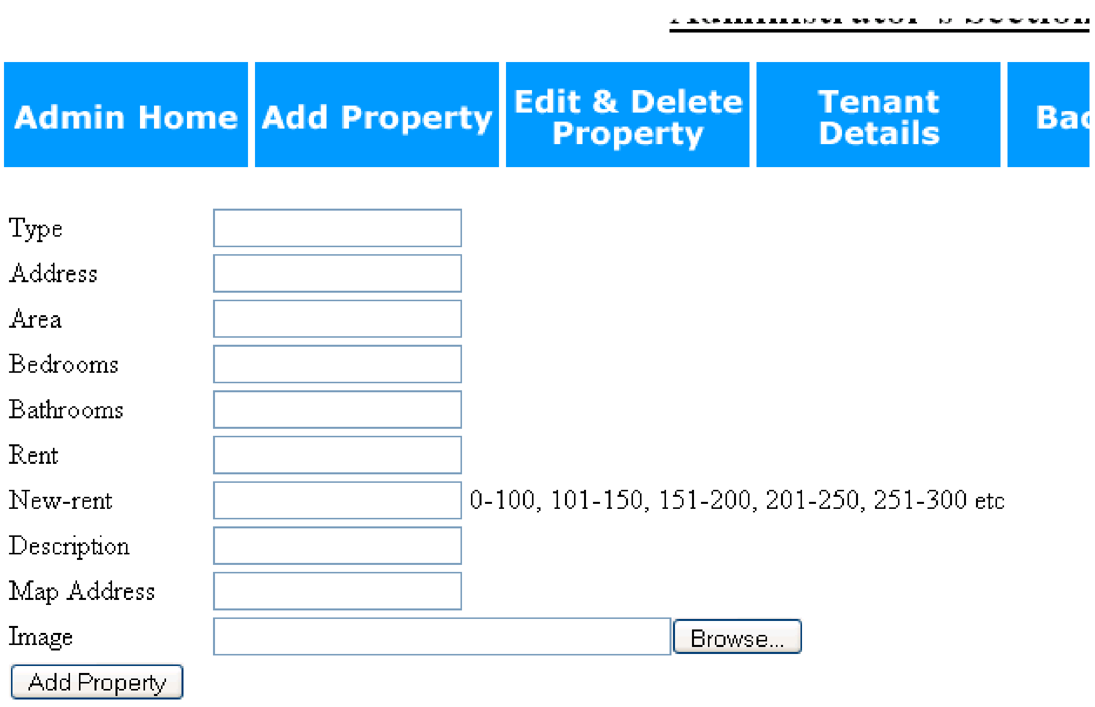
Quicksearch

Area: Hollins
Bedrooms: 1
Rent: 201-250
Find Property
View All Properties

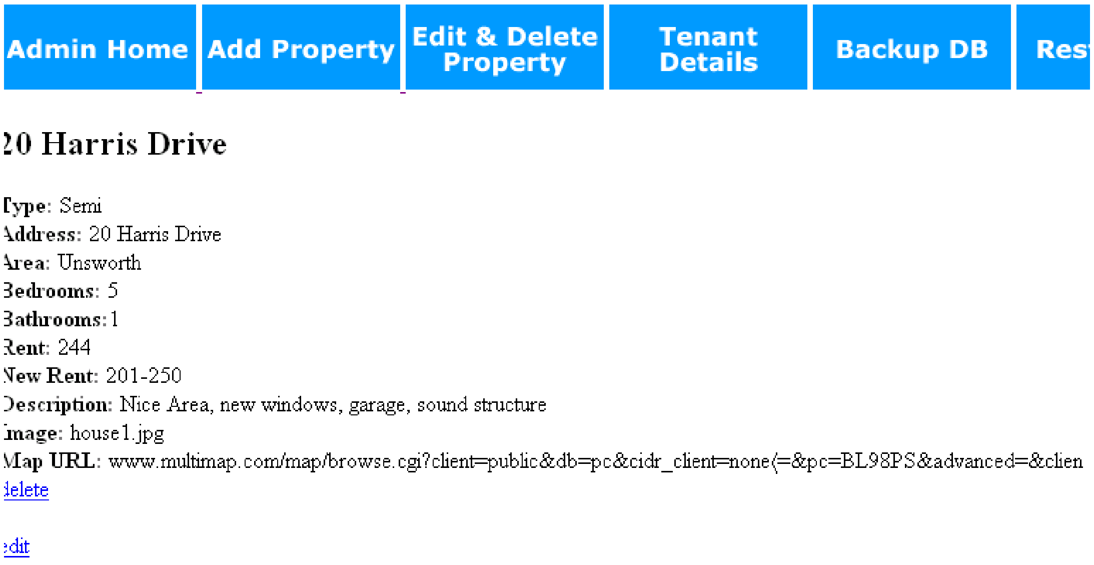
The screen below shows the login script that the user must use in order to access the private pages of the web system.


Untitled Document - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.urbanlifestyleletting.org.uk/private.php
Document View
Torrent Reactor NET
Untitled Document
Login
Username:
Password:
Login

The web page below shows the page that is used to enter a property to the system.



The following page shows a properties details that have been illustrated on a page after being searched for by the end user. The property can be edited and deleted from this page.

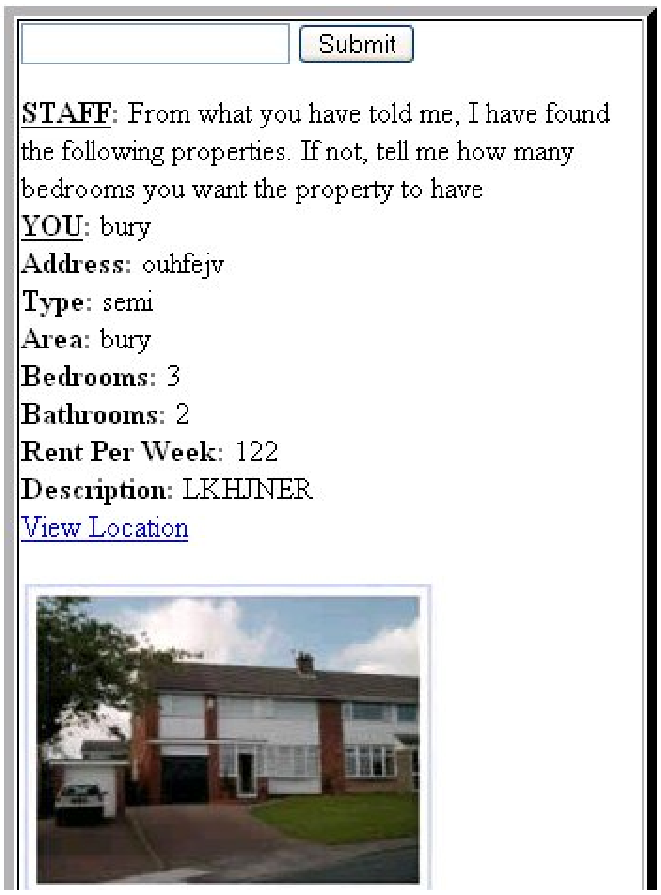


0 Harris Drive

ype: Semi
ddress: 20 Harris Drive
rea: Unsworth
edrooms: 5
athrooms: 1
ent: 244
ew Rent: 201-250
escription: Nice Area, new windows, garage, sound structure
mage: house1.jpg
ap URL: www.multimap.com/map/browse.cgi?client=public&db=pc&cidr_client=none⟨=&pc=BL98PS&advanced=&clien
elete

dit

The screen below shows the prototype of the automated live help feature that stores a user's responses and outputs possible properties they may be interested in.



STAFF: From what you have told me, I have found the following properties. If not, tell me how many bedrooms you want the property to have
YOU: bury
Address: ouhfejv
Type: semi
Area: bury
Bedrooms: 3
Bathrooms: 2
Rent Per Week: 122
Description: LKHJNER
View Location

Code Examples

Switch Statement Used to Inform the User That an Error Has Occurred

```
  switch ($_FILES['backup']['error']) {
        case 0:
          break;
        case 1:
          $msg .= '<b>The uploaded file is too big to process</b><br />'.CR;
          break;
        case 2:
          $msg .= '<b>The uploaded file is too big to process</b><br />'.CR;
          break;
        case 3:
          $msg .= '<b>The uploaded file was only partially received</b><br />'.CR;
          break;
        case 4:
          $msg .= '<b>No file was uploaded</b><br />'.CR;
          break;
        case 5:
          $msg .= '<b>The uploaded file is empty</b><br />'.CR;
          break;
      }
```

Initialising a Session Variable by Creating a Cookie

```
if($admin)
                setcookie("admin", $id, 0);


        if($login)   {
                setcookie("auth", $id, 0);
                //header("Location: index.php");
                //exit;
                }
        else
                {
                setcookie("auth", "no", 0);
                //header("Location: index.php");
                //exit;
                }
```

Sendmail Method of Emailing Forms

```
        <FORM method=post action="sendmail.php">
        Email: <INPUT name="email" type="text"><br>
        Message:<br>
        <TEXTAREA name="message">
        </textarea><br>
        <input type=submit>
        </FORM>
```

```
<?
mail( "root@urbanlifestyleletting.org.uk", "Feedback Form Results",
$message, "From: $email", "-froot. urbanlifestyleletting.org.uk “);
header( "Location: http:// www.urbanlifestyleletting.org.uk /thankyou.htm" );
?>
```