

Multi-user Knowledge Space
Sam Carney
Information Systems BSc
2004/2005

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

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

(Signature of student) _____

Summary

The aim of the project was to develop a knowledge space system that could be used by members of a research group to share their comments on papers and documents that they had read. The current method that the problem owner Dr Ruddle used to store his comments was a text file but this had grown in size and become unmanageable.

An initial attempt to do produce a knowledge space system was made by a previous student. The system developed was not being used and the reasons for this had to be identified and a solution to the problem found. After initial investigations into extending this system it was found that there were a lot more changes needed and it was not just a case of simply extending it. Certain aspects of the system had to be redesigned which meant that much of the version 1 implementation had to be changed.

A version 2 system was developed that fulfils all the requirements that were laid out in the requirements analysis section.

The final part of the report looked into how the version 2 system could be enhanced. The feasibility of producing a semantic system and the benefits it could bring were investigated and conclusions were drawn.

Acknowledgements

First of all I would like to thank my FYP tutor Dr. Roy Ruddle who has given much of his time throughout the length of the project giving me guidance and support.

I would also like to thank my project supervisor Eric Atwell who provided useful feedback in the marking of my mid-project report.

My thanks also go to John Hodrien for helping to evaluate the security measures in place on the system and also setting up access to the system at the beginning of the project.

Thanks to my friends of my course who participated in testing the functionality of the system and I would like to give extra thanks to Royce Reagle who provided especially useful feedback.

Thanks to my parents for proof reading the document and supporting me throughout university.

Table of Contents

Chapter 1 – Introduction	1
1.1. Problem definition	1
1.2. Project Aim	2
1.3. Project Objectives.....	2
1.4. Minimum Requirements and Possible Extensions	2
1.5. Deliverables	2
1.6. Project schedule	3
1.7. Relevance of chosen Project.....	4
 Chapter 2 - Planning and Evaluation of Cai's System	 5
2.1. Design Methodology	5
2.1.1. Description of Methodologies	5
2.1.2. Conclusion	6
2.2. Problem with the Requirements	7
2.3. Techniques for Capturing the Requirements	8
2.4. Methods Used	8
2.5. Generic Knowledge Space	10
2.6. Systems Usability	10
2.7 Evaluation	12
2.7.1. Functional Requirements	12
2.7.2. Non-Functional Requirements	14
 Chapter 3 – Design	 16
3.1 Database Design	16.
3.2 Database Design	16
3.2.1. Initial ER Diagram	16
3.2.2. Initial Scheme for the database	17
3.2.3. Normalization	18
3.2.4. Scheme for the database	19
3.2.5. Integrity constraints	19
3.3. Interface Design	19
3.4. Design of Version 2 System	21
3.4.1. The Design Process	21
3.4.2. Design for Navigation within the System	22
 Chapter 4 – Implementation	 24

4.1. Database creation	24
4.1.1. Perpetration of the Database	24
4.1.2. Converting text file into the database	24
4.1.3. Inserting into the Database	25
4.2. Security	26
4.2.1. Security Issues	26
4.2.2. Register Globals Off.....	26
4.2.3. Create Sessions	27
4.2.4. Storing Database Connections	28
4.3. Backup	28
4.4. System functionality	29
4.4.1. Search Engine	29
4.4.2. Add/Update/Remove Papers	31
4.4.3. Add and Delete Users	32
4.5. System Testing	32
4.6. Delivery of System.....	33
4.7. Conclusion	33
5 - System Evaluation	36
5.1 Introduction	36
5.2 Security Evaluation	36
5.2.1 The Nature of Security	36
5.2.2 Method for Evaluating Security	36
5.2.3. Results and Conclusions from Security Evaluations	37
5.3 System Functions Evaluation	38
5.3.1. Introduction to Functions Evaluation	38
5.3.2. Method for Evaluating the Systems Functions.....	38
5.3.3. Results and Conclusions from Systems Functions Evaluation	40
5.4. Issues Raised from Interviews	42
5.5. Comparisons to other Knowledge Space Systems	43
5.6. Overall Conclusion	44
5.7. Possible Further Enhancements to current system	45
6 – Further Enhancements – Semantic Version	46
6.1. Introduction	46
6.2. Advantages of a Semantic System	46

6.3. Technologies for implementing semantic system	47
6.4. High level Systems Architecture	48
6.5. Conclusion.....	49
References	50
Appendix A - Personal reflection on Project	52
Appendix B - Discussion of Project Plan and Project Methodology	54
Appendix C - Cross Referencing sections to project deliverables	57
Appendix D - Heuristic Evaluation Checklist	58
Appendix E - Test Plan	60
Appendix F - Security Checklist	61
Appendix G - Log Form	62
Appendix H - Results from Evaluations	63
Appendix I - User Manual for Version 2 Multi User Knowledge Space	69

Chapter 1 - Introduction

1.1. Problem definition

Dr Roy Ruddle has compiled a list of many research articles and documents that he has read over numerous years. The list compiled by Dr Ruddle's is a review of all of these articles. The purpose of this review is to help him and others find papers within a certain field and help decide whether these papers are worth reading. The current system he uses stores this information as one large text file. There are well over 1000 entries within the text file which mean's gaining information from it is time consuming. All of the entries in the file have a structured format. Some of the fields included in this format are the title of the article, the authors name, keywords associated with the article and the clients comments about them.

This common format of the entries allows the client to manually search the text file for entries but due to the increased size of the file this is now a very inefficient way of operating. If the client wanted to find details about Human-Computer Interaction they would have to look through the file at every entry where it was mentioned. It could appear in the title, keywords or even in the comments that the clients has made. This demonstrates how inefficient the process is and it does not even take into account other subjects being part of Human-Computer Interaction like heuristics that you may want to find as well. The current format of the file also limit's the accessibility of others to it. Dr Ruddle would like to share the content with others within a research group but the only way to do this at present would be to send out copies of his text file. This solution would not really be feasible as it would inherit the earlier issues discussed about viewing information. It would be a very messy solution to allow others to update this file as they would then have to re-send it to everyone else in the research group, which is likely to lead to confusion over other users having the latest copy.

An attempt was made by Zhidan Cai, an undergraduate student to solve the problem by creating a 'knowledge space'. The purpose of a knowledge space is to make data more easily accessible; in this case turning the relatively unprocessed data into more viewable information. The system would do this by allowing users to search for scientific papers by a number of criteria. It is also envisaged that once a knowledge space is implemented it will allow others to have access to the data and permit trusted users to contribute to the catalogue of information.

The system developed by Cai, was not adopted by the end users and the reasons for this are to be explored and a solution to the problem reached.

1.2. Project Aim

The aim of the project is to research current attempts at trying to implement a knowledge space and evaluate these so that a fully functioning system can be produced that can be used by the problem owner.

1.3. Project Objectives

- Evaluation of Cai's system
- Design and Implementation of Version 2 of the system
- Evaluation of version 2
- Production of a semantic version of the system
- Evaluation of this semantic version

1.4. Minimum Requirements and Possible Extensions

- Evaluation of previous system (2003-4 UG project) using various methodologies.
- Perform requirements analysis to produce the requirements of my system.
- Design of the system.
- Implementation and production of the system.
- Evaluation of my system.

The major extension to the project after the minimum requirements are fulfilled would be to build the system using semantic technologies. The requirements would be:

- Implementing a semantic version of the system.
- Evaluation of this system.

1.5. Deliverables

- The project report.
- A fully functioning system.

1.6. Project schedule

Stage	Date	Objective	Milestone
1	30/09/2004	Submit Project Preference form	
2	14/10/2004	Discuss the project idea with supervisor	
3	22/10/2004	Complete Aim and Minimum Requirements form	
4	23/10/2004 – 30/10/2004	Research into the context of the problem	
5	30/10/2004 – 10/11/2004	Evaluation of Cai's system	Evaluation section
6	11/11/2004 – 26/11/2004	Requirements Capture	
7	26/11/2004 – 09/12/2004	Write up of Mid-Project Report	Mid-project Report
8	10/12/2004	Submit Mid-Project Report	
9	13/12/2004 – 15/12/2004	Design of version 2	Design section
10	15/12/2004 – 10/12/2004	Implementation of version 2	System built
11	22/01/2005 – 29/01/2005	Evaluation of version 2	Evaluation section version 2
12	01/02/2005 – 09/02/2005	Background reading on Semantic technologies	
13	10/02/2005 – 14/02/2005	Re-assess Requirements for the system	
14	15/02/2005 – 13/03/2005	Implementation of version 3 of the system	System built
15	11/03/2005	Submit table of Contents and Draft Chapter	Evaluation section version 3
16	18/03/2005	Progress meeting with supervisor and assessor.	

Stage	Date	Objective	Milestone
17	13/04/2005 – 19/04/2005	Evaluation if the system	
18	20/04/2005	Completion of final Report. Proof Reading.	Final Report
19	27/04/2005	Submit Report to CSO	
20	29/04/2005	Submit Report in PDF format	

Table 1.1 Project Schedule

1.7. Relevance of chosen Project

The project was chosen as the writer felt that it was a well rounded project that would cover aspects of nearly all the subjects he had studied. The information systems modules have prepared the writer for gathering requirements so it is a good chance to test the skills learnt. These modules had also highlighted many examples of why projects are not taken up after they are developed and this project provided a good opportunity to examine this. Designing and developing the system will also draw on knowledge gained from Database modules, HCI modules and Software development modules.

Chapter 2 - Planning and Evaluation of Cai's System

2.1. Design Methodology

To increase the chances of implementing a successful solution to the problem an appropriate design methodology must be chosen from the numerous choices available. These methodologies are likely to entail “A collection of procedures, techniques, tools and documentation aids” (Avison, D & Fitzgerald, G, 2003). Most methodologies organise these procedures into phases to aid the developer. The way of implementing these procedures can vary greatly depending on which methodology is followed due to the various objectives they have.

2.1.1. Description of Methodologies

One of the most popular approaches to developing systems is the Systems Development Life Cycle (SLDC) more commonly known as the Waterfall model. There are various descriptions of the life cycle but (Avison, D and Shah, H, 1997) identify the main components.

- **Feasibility study** - Problems of current system, report on viability of solving problem.
- **Systems investigation** - Highlighting the user's requirements for the system and planning how to implement the necessary resources to implement these requirements.
- **Systems analysis** - Detailed analysis of the current system to determine requirements for the new system.
- **Systems design** - Design of the system to highlight the specification's that will fulfil the user's requirements.
- **Implementation** - Production of the system to include testing and training of staff.
- **Review and maintenance** - Evaluation of the system

The system is called the life cycle model as at some stage the review and maintenance section will start to highlight problems that will mean a new system will need to be developed and then the process starts again. (Laudon, K & Laudon, J, 2002) tell us that each stage has “specific milestones” which should be completed before moving onto the next section. This is one of the main strengths of the SDLC approach when implementing a large system; its structured approach provides a rigid formula and documented goals to follow. However this has a trade off in that it is “costly, time consuming and inflexible” (Laudon, K & Laudon, J, 2002). This rigid structure also means that large amounts of work are done before the system is presented to the user, which means the user requirements must be spot on from the beginning. However even if these are done correctly, what

many users say they want and actually need can differ greatly and it's only after they see a solution that they realise this.

Prototyping was developed as a more flexible methodology to counter some of the weaknesses of the SDLC model. It aims to produce tangible products that can be demonstrated to the user so they can obtain feedback on what they are producing “building iterative working methods of a system” (Avison, D and Shah, H, 1997). The intention is that the iterative steps can be performed quickly and cheaply until a prototype is developed that is acceptable to the user and can be used to help build a final system. The extra user involvement is designed to help refine the requirements for the system and lower the chances of project failure, “with prototyping, user acceptance of a system is regarded as far more likely” (Avison, D & Fitzgerald, G, 2003). However caution needs to be exercised to make sure the prototypes are not implemented too soon without thorough analysis and design.

Rapid action development (RAD) is another example of methodology designed to counter the problem of the traditional approaches. It is designed to find solutions in a much shorter time than the traditional approaches by using structured techniques at the expense of other aspects of a project, “Technical excellence is secondary to the speed of delivery” (Maciaszek, L, 2001). RAD is like prototyping in that the requirements are refined over time but the methodologies differ as RAD allows “key parts of development can occur simultaneously” (Laudon, K & Laudon, J, 2002). We are told by (Avison, D & Fitzgerald, G, 2003) that Joint Application Development (JAD) is often used in conjunction with RAD to try and overcome many of the issues usually associated with requirements gathering through a series of structured meetings. JAD is used to try and make decisions quickly with all stakeholders involved rather than conducting interviews separately. Problems with RAD arise when people try to take too many shortcuts at the expense of “systems quality and documentation” (Avison, D & Fitzgerald, G, 2003).

2.1.2. Conclusion

The methodology chosen should reflect the context of the problem. As a previous attempt has been made at solving the problem with a user knowledge space system being developed it may not be appropriate to use the SDLC as there may only be small changes that need to be implemented. Using the SDLC may be a cumbersome approach due to its rigid nature. Prototyping seems like the best methodology to use as it would allow lots of user involvement. The prototyping will be given structure by defining iterative steps that need to be taken which should safeguard against rushing into the development. As there has already been one system developed that has not been implemented it highlights that there may be difficulties in finding the requirements for the system. To a certain extent some techniques could be used from RAD so the requirements can be obtained quickly. However

these must be picked carefully as the project is not that large and there are a limited number of stakeholders so using some techniques like JAD could extend the development rather than shortening it as intended.

Cai developed her system in a systematic way that went through a modified version of the SLDC and did some evaluation for the system. However the current context means version 2 is an extension of this so it is appropriate to carry on at a review level and then develop the system.

Version 2 of the systems will contain the following sections.

Version 2

- Evaluation of Cai's System and Requirements Capture
- Design of new System.
- Implementation and Testing of changes
- System Evaluation.

If the semantic Version 3 of the system is implemented it is likely to follow a more conventional approach as it will be its own system rather than being an extension of a previous system.

As the project does not follow a completely standard approach the sections produced were cross referenced against the marking scheme to indicate what each section was tackling. This can be found in appendix C.

2.2. Problem with the Requirements.

The current system has some good features, but it is not being used by the end users. This could be because the system developed does not match the requirements set out by the user, or the system fulfilled the initial requirements that the user wanted but these requirements were simply insufficient or inappropriate. The distinction between these is important; if the system just needs the implementation of some more requirements originally set out then there are a clear set of goals, but if the system did what the user asked it will be a more difficult task. This situation will be explored by examining the system using various techniques to discover the extent of the problem.

2.3. Techniques for Capturing the Requirements

To find the requirements of the system a technique called SQIRO was used. SQIRO is actually a

mixture of five requirements capturing techniques; Sample documents, Questionnaires, Interviews, Research, Observations. This was seen as an appropriate method as the context of the problem lends itself well to being explored in this way. There is also the advantage that the developer of the system has experience using this technique. SQIRO is capable of highlighting both functional and non-functional requirements however it would be beneficial to perform heuristic evaluation to compensate SQIRO as an extra precaution.

The current system and any future system created will have various usability issues that need to be considered. Many of these usability issues are related to the heuristics of the system, for example, if an interface is poorly designed many users are likely to reject the system no matter how good its functionality. When a system is poorly designed it can leave users “frustrated and even angry” (Shneiderman, B & Plaisant, C, 2005). A common way to test the usability of a system is to perform heuristic evaluation “reviewers critique an interface to determine conformance with a short list of design heuristics” (Shneiderman, B & Plaisant, C, 2005). There are various lists of design heuristics that you may follow ranging from the ‘Eight Golden Rules’ developed by Shneiderman to Neilsons set of ‘ten heuristics’. It is only necessary to perform the evaluation against one list as many of the points are very similar. The heuristic design lists are not exhaustive but give a general specification that systems should comply to.

2.4. Methods Used

The major sample document studied was the text file where all the data was stored especially in regard to how each entry was structured. It highlighted that each entry had a common structure to it and also that there were certain criteria within these entries that users were likely to want to search by. Questionnaires were deemed unnecessary as interviews with the problem owner and research into the requirements was of more benefit than questioning potential users who were unfamiliar with the system.

An interview with the problem owner (Dr Ruddle) was conducted to highlight the requirements of the system. During the interview with the problem owner it was noted that the requirements were well documented by previous attempts at the system, the problem owner confirmed that he was happy with these requirements. This meant that many requirements could be taken directly from Cai’s report. Cai’s system was then examined against these stated requirements to see which requirements if any were missing.

The following requirements were missing from the system but noted in Cai's initial requirements.

1. The searches within the system for 'keywords and authors' should be able to perform an 'and' operation.
2. Spelling errors within the system should be recognized.
3. Users can print output to a text file.
4. System administrators can create accounts for the users.
5. System administrators can update/delete papers online.
6. The system should have security controls to protect the data.

In a second follow up interview this set of six requirements were presented to the problem owner.

The purpose of this was to confirm their earlier judgement that they were happy with the documented requirements and they were not using the system as one or more of these were not implemented.

After speaking to the client it was clear that the main issue with the current system was that it was not secure. This was one of the requirements noted by Cai but it had not been implemented. The system had a lack of security controls and the client could not risk the data being altered or deleted by someone using the system or someone deliberately trying to cause damage. The other main issue with the system was that the search function should be able to include the 'and' operator. This would allow the client to search for records using multiple entries, e.g. Research papers written by two authors. These requirements must be implemented before the user would use the system. The client would like accounts to be set up on the system so there should be appropriate functionality to allow for this. The system should also allow administrators to update and delete records online. The client said they would like the last two requirements to be implemented but they are not essential. These requirements would allow the user to be able to output data from the system to a text file and for the system to recognize spelling errors but these requirements were not essential.

The research section of SQIRO consisted of Cai's report and system being examined thoroughly with extra emphasis being put on noticing any issues found with the system and possible improvements to it. After Cai developed her system she produced an evaluation of it. These evaluations were conducted with potential users for the system and as it is designed for multiple users it is important to consider their feedback. Due to time constraints it was decided that rather than conducting more evaluations into the system with potential users who were not familiar with the system it would be more beneficial to examine Cai's findings and cross reference them against the interviews conducted with the problem owner. Cai evaluated her system by observing people using the system and obtaining comments from them; the issues found are summarised in the table below.

	Issues with the System
1	After an entry is made into a search field the majority of users press enter expecting a search to occur but they are brought to a menu page instead.
2	The system can't handle spelling errors.
3	Some users found the search titles ambiguous.
4	Keywords are too specific e.g. Entering Virtual reality instead of VR will get completely different results.
5	The system should allow for an 'and' operator in the search function.
6	The simple search is quite complex and the advanced option is not that much more detailed.
7	Not necessary to have viewing order or results on search page.
8	The menu options should allow users to select more than one option at a time.

Table 2.1 Issues Cai found with her System

2.5. Generic Knowledge Space

The requirements capture detailed what was needed for a solution to the problem but the feasibility of implementing these changes within the current system needed to be explored. There are many different ways to store the information about the research documents, with the text file being an example but some are more appropriate than others. A knowledge management based solution was sought to find the most effective way of storing the data so it can be accessed more efficiently.

The system developed will disseminate information to users from various members of the research group allowing them to share their collective knowledge. It will be more than just a bibliography database that provides just the bibliographical references, the system developed will allow the users to share their thoughts and comments on each particular entry. The intended use of the system is not just to simply present a list of references for users to read, but for users to use the comments about the records to help pick relevant documents. It is about the users passing on their knowledge when they critique the papers to be of use to others. Coakes, Willis and Clarke (2002) state that knowledge management is about "the extraction and conversion of tacit knowledge on an individual and organisational level into explicit knowledge". The proposed system will extract tacit knowledge from the experienced users when they critique papers so that it is accessible to others who may not know which papers are relevant to certain subjects. The various comments and ratings will form the basis of the explicit knowledge.

Knowledge management incorporate "the use of technology to make information relevant and

accessible wherever it may reside” (Brown, JS and Duguid, P, 2000). Appropriate technologies should be used to reflect the context of the problem. These technologies must help with the “systematic processes of finding, selecting, organizing and presenting information” (Brown, JS and Duguid, P, 2000). The two main issues that arise when choosing an appropriate technology will be how to store the data and how to view it? In the current situation a text file deals with both of these but more appropriate technologies could be employed.

Cai decided to implement the Knowledge space by storing the data in MySQL and viewing it using the scripting language PHP. When the problem owner was observed in the context of the current problem to see how he operated and the processes he had it was noted that he used the Linux operating system. A major issue for the problem owner was that the system should be able to operate across different platforms while being maintained on a Linux machine and this is possible with these technologies. Research into these technologies also confirmed that it is possible to implement the extra requirements that were needed using these technologies.

2.6. Systems Usability

Although Cai’s evaluation highlighted many of the non-functional requirements it was felt that heuristic evaluation should also be performed as it would compliment the research already done and give extra credibility to the results and not just rely on one source of data. To help explore these non-functional requirements a heuristic checklist (see Appendix D) was compiled based on Neilsons set of ten heuristics. This checklist was then used by the writer and a final year computing students to separately evaluate the site. After both evaluators used the heuristic checklist the data and comments were analysed. It confirmed many of the issues found by Cai but also highlighted some new problems. These results are summed up in the tables below; the first shows the positive aspects of the system in regard to heuristics that should be maintained and the second table highlights potential heuristic issues that need to be considered.

	Positive Heuristic Aspects with the System
1	Consistent use of menu styles.
2	System displays a message to user’s to keep them updated with actions.
3	Error messages are consistent and constructive rather than critical.
4	Menu organized appropriately.
5	Visual feedback of selecting a menu option provided.
6	System allows users to change an earlier menu choice.
7	Familiar icons used for the search function.

	Positive Heuristic Aspects with the System
8	Lowercase lettering has been used where possible.
9	When a search produces zero results a message is presented.
10	Demonstration of how to use the system given.
11	This 'help' is clear and easy to find.

Table 2.2 Positive Heuristic Aspects with the System

	Negative Heuristic Issues with the System
1	The start-up page for the system requires the user to scroll down.
2	No warning prompts for the user if they try and alter data
3	System does not have functionality to undo actions, I.e. updates
4	The user can't edit existing data to save time on data processing
5	The user can't set session defaults
6	The format of the page does not lead users to important information, too clustered.
7	Too much information and features provided on simple search.

Table 2.3 Negative Heuristic Issues with the System

2.7. Evaluation

After examining the results the following set of requirements were set out for implementation in the version 2 system.

2.7.1. Functional Requirements

The table below shows the functional requirements for version 2 of the system. The requirements are prioritised using the MuSCoW rules, what Must be implemented, what Should be implemented, what Could be implemented and what Won't be implemented. It also highlights the requirements that are already present in the system, and these are shown as 'already implemented' in the table. Version 2 would have to provide all the requirements implemented in Cai's system and at least the requirements that the system Must have.

	Requirement	MuSCoW
1	The searches within the system for 'keywords and authors' should be able to perform an 'and' operation.	Must

	Requirement	MuSCoW
2	System administrators can create accounts for the users.	Should
3	Users can update/delete papers online.	Should
4	Users can print output to a text file.	Should
5	Offer address of online copy of an article if available.	Could
6	Have a database holding the fields as required and in the same format of the text file.	Already Implemented
7	A simple search engine where users can search on the fields' source name, title, author, keywords and year published.	Already Implemented
8	The combination of the fields performs an "and" operation.	Already Implemented
9	An advanced search engine where users can search on the additional field's source detail, summary and comments.	Already Implemented
10	Case does not matter in input fields.	Already Implemented
11	Users do not necessarily input the whole name in the source or title fields.	Already Implemented
12	Users can input more than one author in the author name field, and more than one keyword in the keyword field.	Already Implemented
13	An "or" operation is available for the author name field and keyword field.	Already Implemented
14	Output all source names, author names and keywords existed in the database.	Already Implemented
15	Enable users to get a source name, author name and keyword by clicking on a button.	Already Implemented
16	Users are able to find a source name, author name and keyword easily.	Already Implemented
17	Users can search for a range of year with a beginning and ending, or a year before an ending or after a beginning.	Already Implemented
18	Users inputs are visible together with the papers returned.	Already Implemented

	Requirement	MuSCoW
19	Users can choose the detail of papers to view.	Already Implemented
20	Users can view the full details of papers.	Already Implemented
21	Users can order the papers returned by a variety of ways.	Already Implemented

Table 2.4 Functional Requirements for Version 2

2.7.2. Non-Functional Requirements

The results from the heuristic evaluation and the evaluation by Cai on her own system have been considered and resulted in the following non-requirements for improvements on version 2 of the system. The requirements already implemented in the system are again noted and will have to be included in version 2.

	Requirement	MuSCoW
1	The system should have security controls to protect the data.	Must
2	Users should be prompted when removing/updating data.	Must
3	Implement changes in appearance of site in accordance with heuristic guidelines.	Should
4	Simplify the simple search feature.	Should
5	The system should remember users so they don't have to log in every time.	Could
6	Spelling errors within the system should be recognized.	Could
7	Allow user's to set session defaults.	Won't
8	Fit web site standards and the system can at least be browsed in Internet Explore and Netscape.	Already Implemented
9	Multi input options to improve the system's efficiency and effectiveness.	Already Implemented

	Requirement	MuSCoW
10	Desired response time from input.	Already Implemented
11	A clear output with viewable preceding user input.	Already Implemented

Table 2.5 Non-Functional Requirements for Version 2

3. Design

The major design aspects for this project are the design for the database and the design of the user interface. As there is already a system in place this chapter will have to explore whether the present design of these aspects is appropriate or if they inhibit users from using the system. The major decision that will have to be made will be whether the extensions that will be added will require a complete redesign of these different aspects or if they can be made in conjunction with the design that has already been laid out for the system.

3.1. Database introduction

To produce a system that provides the desired requirements the database will have to be redesigned from version 1. Version 1 does not have provisions for storing details about users and this will be a pivotal part of the security for setting up sessions with security levels. The way that the authors and keywords are stored in the database inhibits the alterations that need to be implemented on the search function so this will also have to be changed.

3.2 Database Design

3.2.1. Initial ER Diagram

To obtain a good solution for the design of the database ER (Entity relationship) modeling was used as it allows you to produce a conceptual data model at a high level of abstraction. Elmasri & Navathe (2000) state that an entity is a “thing in the real world” but this encompasses physical entities like authors and also conceptual entities like papers so they can exist in the same model. These entities will all have “attributes” (Elmasri & Navathe, 2000) that have specific values and these can also be represented within the model. The ER model for the system is produced by collating the information gained in the requirements capture (figure 3.1)

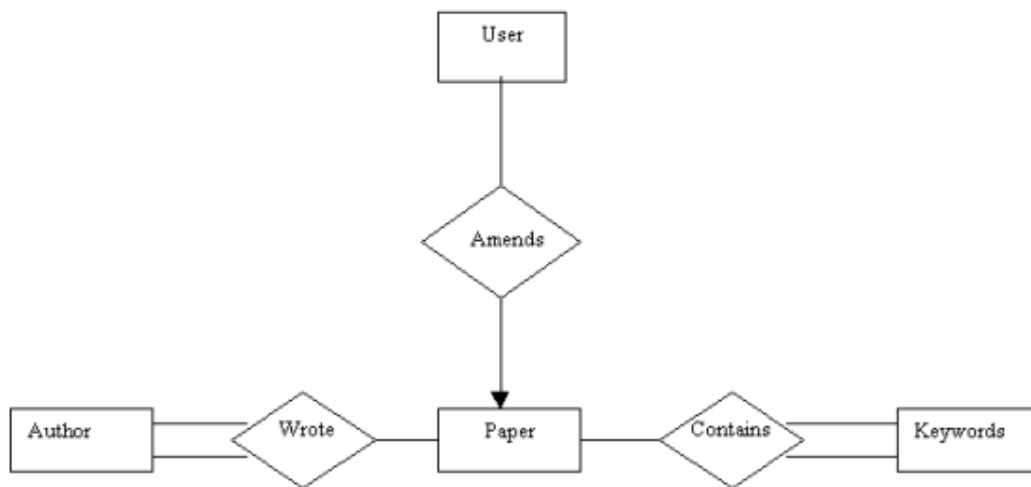


Figure 3.1 - Initial ER Diagram

The ER diagram depicts the scope of the system. It can be seen that there are 4 main entities within the system. These entities have different relationships with each other depending on the cardinality of their relationship which is determined by the “the numeric relationships between occurrences of the entities” (www.DataModel.org, 2005)

Users are able to amend records but papers cannot amend the users so this relationship is one way. It is possible for more than one user to amend records and they are able to amend more than one record so it is many to many relationship.

The paper entity can be viewed as the main entity in the system as the author, source and keyword entities all link to it. One paper can only have one set of authors or one set of keywords so the relationship is a one to one relationship.

3.2.2. Initial Scheme for the database

An initial schema was created for the database by using the ER diagram to create the tables. The attributes for these tables were formed by examining the requirements for the system.

User (User_id, First_Name, Last_Name, email, Username, Password, User_Level, Activated)

Paper (Paper_Id, Source_Name, Title, Issue, Year_published, Summary, Comment, RF, DateRead, Location, Quality)

Author(s) (Author_Id, Author_Name)

Keyword(s) (Keyword_Id, Keyword)

3.2.3. Normalization

Once an initial schema is created it should go through the process of normalization to improve the quality of the design. Elmasri & Navathe (2000) state there are 2 desirable consequences that can occur due to the normalization. The process will help minimize redundancy which means that you will store less data in the database so there will be less of a chance of inconsistency between the tables and should also help the speed when searching the records. The source attribute demonstrates how redundancy could enter the database. As noted earlier papers can come the same source therefore one option would be to store all the source records within the papers table. However this means that you will be storing the same data multiple times and as the database grows this could have a significant effect on size of the database and due to this increased size the time it takes to search it. This simple solution to this is to decompose the paper table so that source details are now contained in a separate table and the paper table just contains a source_id.

The author and keyword tables do have an element of redundancy in that there is a chance that the system could store a keyword or authors name more than once. The database will store every unique collection of keywords or authors e.g. Smith and Jones will be stored as a unique author's collection, so the author Smith could also appear in the unique collection of Smith and Bailey. However the database will only store Smith and Jones once even if they write more than one paper. The database is designed in this way to improve the search on the system "Controlled redundancy may be useful for improving the performance of queries" (Elmasri & Navathe, 2000). If each author was only stored once temporary tables joining the collections of authors that worked on particular papers would have to be created. This would mean searching

multiple tables that would slow down queries especially when searching for a collection of authors.

Normalization will also minimize anomalies that occur because of insertion, deletion, and update statements. This new source table demonstrates this; if the name of a source changed this would mean changing all the occurrences of this in the paper table which could easily lead to inconsistencies. Even if every detail was changed to make sure there were no inconsistencies it would be a very laborious task that can be avoided due to a better database design.

3.2.4. Scheme for the database

After the process of normalization was applied a new schema was designed to accommodate the changes.

User (User_id, First_Name, Last_Name, email, Username, Password, User_Level, Activated)

Paper (Paper_Id, Source_id, Title, Issue, Year_published, Summary, Comment, RF, DateRead, Location, Quality, Author_id, Keyword_id, Source_id)

Author(s) (Author_Id, Author_Name)

Keyword(s) (Keyword_Id, Keyword)

Source (Source_Id, Source_Name)

3.2.5. Integrity constraints

Once the schema has been designed Integrity constraints have to be implemented to control the data in a database and make sure it is consistent. They work by placing “restrictions on data that can be specified on a relational database schema” (Elmasri & Navathe 2000). One form of integrity constraint is explicitly defining the data type of the values in the system. In version 2 users have a user level which controls their access rights within the system, for this to work it has to be a specific integer so the user table will only accept integers for this attribute. Another control will be to make sure primary keys do not accept null values, which is known as Entity integrity. If this situation did occur the joins between the tables would not function properly that would have a knock on effect on the system and would disable much of the functionality. Referential integrity concerns the relationships between the different tables; it is designed to preserve relationships between tables when new records are added, updated or deleted. An example of how referential integrity constraints are enforced on the system is that it would not let you change the paper_id of a paper as other tables would then be referencing values that are non-existent.

3.3. Interface Design

The design of the user interface is an important aspect of any system but in many systems it is not given much thought until after the system is built. The interface needs to be carefully designed so that users are able to use the features that the system has to offer. It is “the point of contact that enables an interaction between a human being and a system” (Cadle, J & Yeates, D, 2004). The user interface is a major factor in determining the usability of a system. Ravden and Johnson (1989) describe usability as “The extent to which an end-user is able to carry out required tasks successfully”. The version 2 system could meet all the requirements stated in section 2, but if the usability of the system is poor it may still be rejected by the problem owner. According to Shneiderman (1998) there are five measurable human factors that need to be considered when designing systems to increase their usability.

The time taken for users to learn how to use the system.

The efficiency of the system in regards to how long it takes to complete various tasks.

The rate of errors that occur.

Retention of knowledge on how to use the system.

User satisfaction.

A tutorial is provided on the system to try and decrease the time taken for users to learn how to use the various functions however even this will take time to time to read so will keep learning time high. Version 2 will try to decrease these times by making the features the system offers more consistent with the standard features associated with popular search engines. Developing the system with this in mind should also allow user to retain much of the knowledge that they developed when using the system.

The most frequently performed task on the system will be searching the various records. The new requirements that have to be implemented on version 2 in regards to the search function will make this task more efficient.

The most likely error that will occur on the system will be users inputting incorrect information into the search fields. The system will try to deal with this by providing users with messages when searches result in no papers returned e.g. If a user entered the paper title into the authors field it would return no results, instead a message informing the user that they had searched the authors field and the string that they inputted would be presented.

If users are not happy with a system they will think up all sorts of excuses for not using it so it is

important to conform to their needs. Although user satisfaction will not be completely measured until after the system is developed the 4 factors listed above should increase the usability of the system which is a “key component of overall systems acceptability” (Le Peuple, J & Scane, R, 2003)

3.4. Design of Version 2 System

3.4.1. The Design Process

As version 1 of the system was already implemented this meant that there was a previous design structure for the system. The decision that had to be made was whether the problem owner was happy with this design, happy with aspects of it but wanted some bits changed or whether he wanted this design scrapped and a new design presented. After conducting an interview with the problem owner it was clear that he was happy with the general style of the design and wanted the new features added to the system to be consistent with the previous version.

The problem owner did however raise some general issues concerning certain aspects of the design. The problem owner wanted the interface on the home page to be altered so that all the information fitted on one page rather than having to scroll down to complete his search. He also asked that the system keystroke pattern be changed so that when a user pressed return while on the home page it commenced a search. In its present state pressing return brought up further options that could be put into the search and this didn't conform to the standards associated with most search engines. As well as implementing these changes the results from the heuristic evaluation carried out in section 2.6 had to be examined. This major issue it raised was in regards to consistency between the various pages on the version 1 system so the new system had to address this as well as considering the other potential heuristic improvements.

Due to the version 1 system missing certain requirements version 2 will had to have more web pages on it to accommodate these requirements. These pages that have been added to the system were designed to be consistent with the general design of the rest of the system by using the same style and layout that had been set by the previous system. The changes were developed in a prototype fashion by showing the problem owner some of the new design for aspects like the registration. As it was not a complete redesign of the system and it was just a case of making sure the new implementation changes were consistent with the previous design the writer decided to implement the changes on the system itself. The major advantage of developing it on the system is that changes can be “examined and reviewed by the users and modifications and refinements can be made quickly and easily” (Cadle, J & Yeates, D, 2004). The problem owner confirmed that he was happy with the initial design for the new aspects of the system although he did comment that he thought the home page now looked

clustered. This was due to the writer restructuring the borders and some font sizes for text to make sure the page fitted into one.

The second prototype developed only needed a small amount of adjustments. Some of the text was taken out of the home page that was deemed unnecessary to try and alleviate the problem of users getting confused due to information overload on the page when they use the system.

3.4.2. Design for Navigation within the System

When designing the extensions to the system the flow that users would take had to be considered. A good system will guide the user through the processes they need to perform and this is especially important for new users who are unaccustomed to the system. Cadle & Yeates (2004) state that navigational flow within a system can be improved by making sure that the features it offers are “logically structured and labeled”. Obviously one of first tasks a new user will have to perform will be to register for the system so they can gain access to it therefore the flow of control for this aspect of the system needs to be simple and not confuse the user at an early stage.

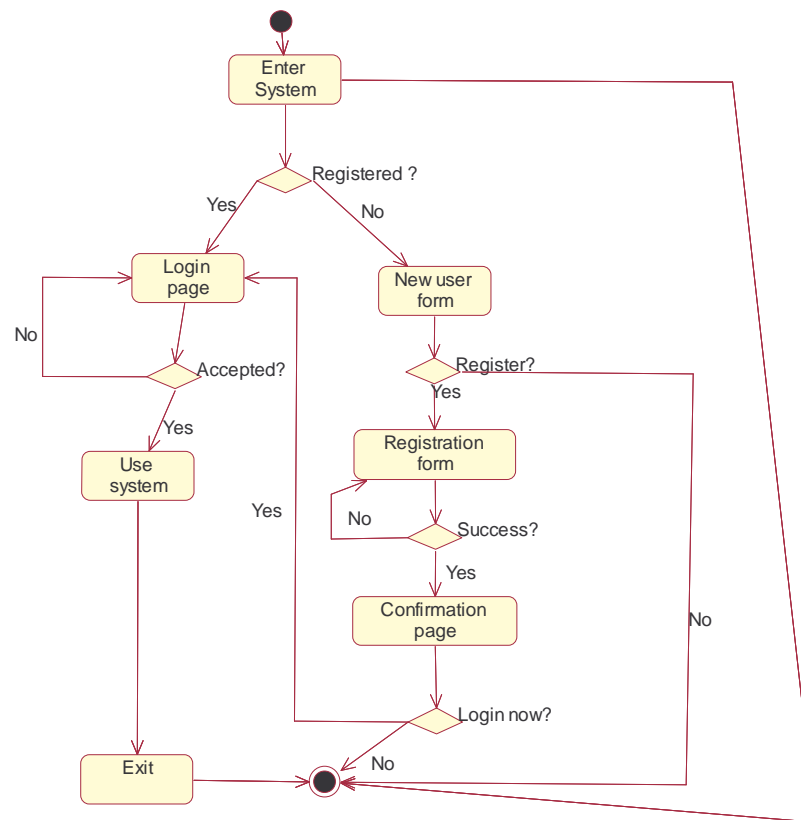


Figure 3.2 - UML activity diagram demonstrating user navigation through the registration process

The activity diagram shows the flow that a new user would have to negotiate through the system. Once a user decides to enter the system there next action will be dependant on whether they are already registered. If they are a new user they will have to register for the system if they want to access it. They will be presented with a registration form and if they fill in all the fields correctly their part of the registration will be complete. The user will then have to get the account activated by the system administrator. Once this has been done they can then login to the system.

4. Implementation

This section details the implementation of Version 2 of the system. It is not an exhaustive list of every modification to the previous system but it highlights the major changes that have been made. The initial creation of database is discussed, and then the security features added to protect the system before finally going on to discuss the major functional requirements that have been added. To conclude the implementation the system had to be tested. To accomplish this task a test plan was created and the results from these tests were examined to see if they had any major implications for the system.

4.1. Database creation

4.1.1. Perpetration of the Database

The first stage of the implementation was to create a new database on the test server so that that when alterations were made it did not affect version 1 of the system. The database originally had the same set up as the database accessed by version 1 of the system but this had to be changed to accommodate changes in the functionality. Also a new version of the text file containing the papers read by Dr Ruddle had to be inserted into the database.

4.1.2. Converting text file into the database

The text file could not be entered into the database in its present form directly so a parser was created in Java that took certain elements from the text file and outputted them to files. The input file has a structured format and an example input is seen in figure 4.1.

```
JN: Proc. ACM I3D
IS:
TI: Semantics of interactive rotations
AU: Pique
PY: 1986
KY: Object Manipulation, Theory
SU: Kinesthetic correspondance between device and manipulation
CO: Kinesthetic correspondance is akin to psychologists' stimulus-response compatibility
RF:
DT: Sep 1998
PH: YES
```

Figure 4.1- An example entry in original source file

The parser worked by pointing it at the original source file and getting it to process this document and output the selected fields e.g. To output all the authors the parser is altered to look for each occurrences of the authors in the text document. The parser reads the document and then an IF statement is used to outputs all occurrences of lines that begin with 'AU:' to a separate file.

The logic in the parser required that there be no errors in the original source but when it was run it was clear this was not the case so the original source file had to be 'cleaned up' to remove any anomalies e.g. missing fields or incorrectly spelt fields. Rather than reading every entry another parser was created to look for occurrences in the documents that did not match the standard fields. The occurrences that were outputted could then just be searched for in the document and corrected. The parser was then used to output selected fields but there could still be errors from a field being entered twice for the same document. To test for this the output was examined in a sort of binary search manor by examining a record in the middle of the output and seeing whether this was correct by comparing it to the original source document. The process was continued until the original source document was free from errors.

4.1.3. Inserting into the Database

This allowed for the easy creation of tables like Authors as the parser could just print out every instance of where this occurs in the text file to another file that just contained the authors of the papers. Data can than be entered into MySQL from a text file by either using 'LOAD DATA INFILE' via the command line prompt or by using a insert feature in phpMyAdmin. An attempt was made to do it via the command line prompt but errors occurred which the writer believes were either due to the text files not being in the correct directory or not having the necessary permissions within MySQL to perform the operation. Consequently due to time limitations it was decided to leave this method and use phpMyAdmin feature which performs the same operation. When creating the database tables, i.e. Author table, an auto_increment function was included within the table so that for every author entry entered into the table it set an Author_id primary key so the table could be joined to the other tables.

The database was accessed via the PHP codes. All of the existing files for version 1 of the system had to be updated so that they accessed the new database rather than the previous version. This is now controlled from one PHP file that is included in the rest of the PHP files so any future amendments would mean altering only one file.

4.2. Security

4.2.1. Security Issues

One of the main problems with the Version 1 system was that it was not secure as there were no features in place to control who had access to the system or what they could do on the system which meant it was extremely vulnerable to malicious attack. The problem owner could not use a system where his source data was so easily compromised therefore security features had to be implemented. However before implementing these it was important to understand the nature of security. Good security will come from understanding that it is a top down process that involves many different people, it is not just about implementing the best current technology, (“to ensure that your security features deliver what they promise, they must be driven from a sensible security policy”, W.S. Whyte, 2001). Good security measures are of extra importance to institutions like universities as they are a likely target of attack from recreational hackers who see universities and especially computing departments as a good challenge. One suggestion to make the system more secure that was considered was to only allow access to the system from IP addresses that came from Leeds University. However this had to be dismissed as the problem owner wanted to be able access the system out of university. This highlighted the need to find a balance for the system between allowing access and implementing the security “estimate the actual amount of material loss versus cost of protection” (W.S. Whyte, 2001).

4.2.2. Register Globals Off

When PHP reads in variables as inputs i.e. a password; it can do this globally by having the `register_globals()` function turned on which allows the input to take in almost any variable. However this leaves it vulnerable to malicious input as the variables can come from anywhere, which means a user could overwrite the intended variables on the system to send the system variables of their choosing e.g. If a user level of 1 was needed to gain access to a system a user could insert code to say that their user level equals 1 and overwrite their actual user level. They would then be granted access to the system. The `register_globals()` function was turned off and certain alterations in the PHP code were made to accommodate these changes in relation to how inputs were read. In this example the location of the input for the username is specified, as the POST method called ‘username_input’.

```
$username2 = $_POST['username_input'];
```

Attempts to circumnavigate this by inputting variables directly into the URL will fail.

4.2.3. Create Sessions

Sessions were created on the system to enable user authentication so that access to the system could be controlled. The benefits of these sessions are that they are temporary so users who do not enable cookies can still log into the system. Potential users can register to use the system and set up their own details including passwords which are encrypted as a 32 bit hash before it is stored in the database. The hash is created by using the md5() hash function which is part of PHP base set of functions. Storing the passwords using the md5() hash function took longer to implement than expected. The system was able to store the passwords as a hash but it would then not recognize the password when it was entered. There were two reasons for this; originally the users table had been designed only allocating 25 characters for the password so when the passwords were stored they would miss the last 7 characters of the hash. The md5() hash function is an example of one way encryption; it checks the passwords in the database by encrypting the inputted password and then compares the 2 hashes. When the system compared the hashes the one stored in the database was missing 7 characters so they did not match. When this was corrected the passwords would still not validate and it was discovered by the writer that this was due to the register_globals() function being turned off. The inputted password was processed using post methods and this fixed the problem. The example line of PHP code demonstrates how to encrypt a password using the md5() hash function.

```
$password =md5($password);
```

The hashing of passwords on its own does not guarantee that the passwords are reasonably secure as a user may pick a very poor password in the first place that could be guessed. To stop this the pc_password check() function in PHP was included in the script when users register so they pick passwords that are not easy to guess. The password checker stipulates that passwords must be at least 6 characters long, they cannot be a dictionary word or the username of the subject.

The Session_Check() function is included in all the scripts to stop anyone bypassing the authentication on the system. If a user tries to navigate manually to a page by typing in the address and they have not logged in they will be redirected to the login page. This on its own is not enough to stop people entering the system as they could set up accounts and then use the system, therefore an activation level for users was created. Once the users have registered with the system they can only gain access to it once the system administrator changes their activation level is changed from 0 to 1. This can easily be done by the database administrator by editing the users table in phpMyAdmin. This extra level of control is designed to restrict who can join the system. If an invalid attempt is made to login they will be presented with the following page.

You could not be logged in! Either the username and password do not match or you have not validated your membership!
Please try again!

Knowledge Space

Please log into the system.

Note: Entries are Case Sensitive!

Username	<input type="text"/>
Password	<input type="password"/>
	<input type="submit" value="Submit"/>

[New user?](#)

Figure 4.2 - Demonstrates an invalid login attempt

4.2.4 Storing Database Connections

Another issue for the system was that the connections to the database were originally stored in each of the PHP files. Storing the passwords in each of these PHP files is very insecure therefore these were removed and access to the database was achieved by calling the config.php file indirectly which stored the connections.

4.3. Backup

Although security measures have been put on the system measures to backup the system have also been explored. The problem owner also stated that they would like to be able to output the system to a text file as an extra precaution. It was decided the best way to do this was to use the phpMyAdmin backup utility as this would be available online. It allows the administrator to store the contents of the database but it also has the added benefit of storing the commands to restore the tables if they do become corrupted.

4.4 System functionality

4.4.1. Search Engine

One of the main implementation issues with the system was to make the Author and Keywords searches do a Boolean 'and' search when more one value was entered. After initial attempts to implement the required changes using the existing structure of the database and search function failed it became clear that the requirements that needed to be added could not be implemented without making major alterations to the version 1 system. The design of the database had to be changed and this had a knock on effect meaning that all code accessing these databases also had to be altered

which resulted in a significant amount of work. This situation highlighted the benefit of prototyping as the problems with the design of the version 1 system were found at an early stage which then allowed sufficient time to find a solution.

The design of the database for version 1 was inhibitive to producing this Boolean ‘and’ search as all the authors and keywords were stored independently of each other and they then referenced the paper table indirectly. Implementing this Boolean searching on this would have involved creating large temporary tables that would have been very complex procedure. These temporary tables would have had to join all the authors who participated in writing a paper together so that it could then be searched by the system.

The writer was apprehensive about going down a similar path that lead to this Boolean searching not being implemented on Version 1 of the system and decided it would be prudent to attempt to implement it in a different way. To achieve the changes the database was redesigned in chapter 3. The author’s table was altered so that the table now stores all instances of collections of particular authors. This means that an instance of author Card in the Authors table will be stored independently from an instance of say authors Card & Burr. The disadvantage of redesigning the database like this is that it had approximately 100 more records than the version 1 alternative. However this controlled redundancy should provide a more efficient search than if it had been implemented on temporary tables.

MySQL has a feature which allows for FULLTEXT indexing which alters database tables to facilitate Boolean searches on them. The important distinction between how the instances were stored meant that this feature could now be used on the table which was not possible under the previous design. The tables were altered using:

```
ALTER TABLE Table_Name ADD FULLTEXT(Column_1, Column_2);
```

After the tables are indexed you can run matches against every record in the table or just for certain collections of columns. The Match() function performs a natural language search on the collection. The Match() function was implemented within a “where” clause in the PHP search code so that when the results were returned they were sorted by relevance with the most relevant appearing first. This relevance is determined by factors including how closely it matches the input, the amount of documents in the collection and the amount of times the specified input appears in the document. If a user entered the authors Card, English & Burr into the authors field search the scoring system would result in the following outcome.

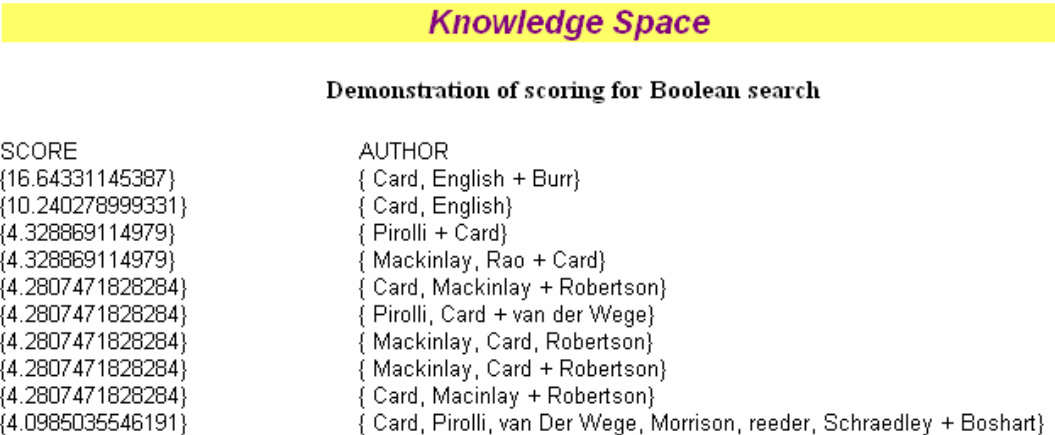


Figure 4.3 - Demonstrates scoring mechanism for Boolean search.

This example demonstrates how the match returns the most relevant document first as it contains all 3 authors. The system does not take into account the order in which keywords are entered, e.g. If the user entered the authors as English, Burr & Card into the search it would still return this record first. The second record returned has a lower score as it contained just 2 of the authors but still has a much higher score than the rest of the records that just contain the author Card. If a user had entered just the authors English & Card the paper that was authored by just them would be returned first, then the paper written in conjunction with Burr and then finally the papers written by Card.

Implementing a new method for searching the author and keyword fields meant that the code for all the remaining search functions had to be re-written. The source name, paper title and year published fields would not work in conjunction with the new author and keyword fields, which meant that more work had to be done just to make sure no requirements were missing in the version 2 system.

4.4.2. Add/Update/Remove Papers

The ability for systems administrators to add/delete papers was not fully implemented in the version 1 system. There were issues in making sure that all records were removed from the system when a record was deleted. There was also a problem regarding who was able to alter the records as it required a user to log in every time an alteration was going to be made to the papers records. Version 1 had a login but the passwords were not encrypted within the database and the accounts had to be created within MySQL itself.

Deleting records from the database was complicated by the need to make sure that when a record from one table is deleted all associate records were also deleted so that the database does not violate

referential integrity constraints. When a paper is deleted from the system a record in the paper table will be removed but it also has to remove all the associated records that are joined to it, in the Author and Keyword tables. They are removed by joining the papers record to the other records in the database that share the same primary key and then they are all deleted. However to make sure referential integrity constraints are not violated the keywords and authors will only be deleted if they are not being referenced by another paper. This is achieved by placing the delete function within an IF statement that checks that paper table to see if any other records are referencing the same author id or keyword id. If another paper was referencing the author and keywords only the original paper record would be deleted.

The problem owner also specified that not all users should have the same rights within the system and that he would like to be able to specify which users had the ability to alter the collection of papers within the database. To achieve this role based access control levels within the system were created to stop unauthorized users from performing certain actions and viewing certain pages. There are 3 access levels on the system. The default level 0 will have the most basic access and will only have permission to view records on the system. Level 1 access allows users to add and update papers as well as viewing them. Finally the highest access level 2 allows all the previous privileges and also allows users to delete records from the database.

A simple if statement was created that checked the user's permissions.

```
if ($_SESSION['user_level'] == 0) {  
    print 'You Do Not have permission to view this page';  
}
```

Changing the user level is similar to changing the activation level; it can only be done by the system administrator by using phpMyAdmin.

4.4.3. Add and Delete Users

A further requirement that version 2 of the system had to include was the ability to add and remove users. The system allows for users to register their details themselves and the only task the system administrator needs to perform is to activate the accounts as discussed in the security measures. It was decided in conjunction with the problem owner that using phpMyAdmin was the best way to remove users as this task will be done rarely and can be completed with relative ease within the program.

4.5. System Testing

The test plan for the system was slit up into two main areas, unit testing and integration testing. The unit testing process tested new features added to see if the specific units worked as intended. Cadle and Yeates (2004) state that unit testing is worthwhile where developers are “unfamiliar with the techniques or standards” and as many of the features implemented went beyond the writers prior knowledge of the technologies it was decided unit testing would be beneficial.

In the integration testing units were indexed into groups and tested to check whether the units worked in conjunction with the related aspects of the system before finally testing whether all the aspects of the system worked in conjunction with each other. A sample test plan is presented in Appendix E For the adding, updating and deleting of papers.

The testing highlighted a number of bugs in the system. The largest problem was that when papers were updated the system would confirm these changes were made but in reality they would not have changed in the database. The SQL commands to update the database were missing a field in a table so the update command was not being inserted; the SQL command was changed to fix the error. The testing also found that some of the PHP pages added to the system at an early date were missing the session checker which meant users could navigate to them without having to login. This was a relatively easy fix and the pages were altered to call the session checker.

4.6. Delivery of System

The system itself is self contained on the test server so there was no actual file transfer to the problem owner as they had full access to this server. However the passwords used to access the database were given to the problem owner and they were then shown how to change the password in MySQL. For the PHP files to work the config.php file also needed to be changed to accommodate this change in password and the problem owner was shown how to make this simple change.

After the difficulties that the writer encountered trying to get ownership of the system, it was decided to produce a user manual (Appendix I). Although this wasn't in the original plan it was decided it would be beneficial in case there were any future developments or alterations that had to be made. The manual lists all the files that are needed to produce the system and gives an overview of how the different file work in relation to each other. The other advantage of producing the user manual is that it will help to allow the system to be portable so that it could be used in other research departments. They may want to make minor alterations to suit their needs so an explanation was given in the

manual by the writer for this eventuality.

4.7. Conclusion

The implementation of version 2 of the system was intended to provide extra functionality that the problem owner believed was missing from the previous system. All of the functional requirements that were agreed with the problem owner (Section 2.7) have been implemented on the system except for requirement 5 (table 2.4) to offer website address to the documents which was not implemented due to time constraints. However this was not a priority requirement, as shown by its “could” be implemented status within the original requirements. The main non-functional requirement that has been implemented is the security features.

A complete list of the functions available on the system is compiled in table 4.1.

	Requirement	Status
1	The searches within the system for ‘keywords and authors’ should be able to perform an ‘and’ operation.	Implemented
2	System administrators can create accounts for the users.	Implemented
3	System administrators can update/delete papers online.	Implemented
4	Users can print output to a text file.	Implemented
5	Have a database holding the fields as required and in the same format of the text file.	Implemented
6	A simple search engine where users can search on the fields source name, title, author, keywords and year published.	Implemented
7	The combination of the fields performs an “and” operation.	Implemented
8	An advanced search engine where users can search on the additional fields source detail, summary and comments.	Implemented
9	Case does not matter in input fields.	Implemented
10	Users do not necessarily input the whole name in the source or title fields.	Implemented

	Requirement	Status
11	Users can input more than one author in the author name field, and more than one keyword in the keyword field.	Implemented
12	An “or” operation is available for the author name field and keyword field.	Implemented
13	Output all source names, author names and keywords existing in the database.	Implemented
14	Enable users to get a source name, author name and keyword by clicking on a button.	Implemented
15	Users are able to find a source name, author name and keyword easily.	Implemented
16	Users can search for a range of year with a beginning and ending, or a year before an ending or after a beginning.	Implemented
19	Users inputs are visible together with the papers returned.	Implemented
18	Users can choose the detail of papers to view.	Implemented
19	Users can view the full details of papers.	Implemented
20	Users can order the papers returned by a variety of ways.	Implemented

Table 4.1 Functional Requirements implemented on Version 2

5. Evaluation

5.1. Introduction

The main alterations to the system have been in regards to what the system can actually do and security for the system. Due to the different nature of these implementation changes it is beneficial to test them in different ways. First the security of the system is evaluated and then the systems functionality and usability is evaluated. The system is then examined against other knowledge space systems to highlight the various comparisons and differences between them.

An overall conclusion is then made from the evidence of the evaluations. Finally some further enhancements are examined to highlight some potential extensions to the project.

5.2. Evaluation of Security

5.2.1. The Nature of Security

Testing the security of the system needed careful consideration as the nature of security means that it's not just evaluating one piece of technology that may have been implemented. The different methods and security features implemented had to be examined to see how they work to protect the system. The security of the systems then has to be examined as a whole as it is only as good as its most vulnerable aspect.

5.2.2. Method for Evaluating Security

The system was shown to a PhD student who had expertise and experience in areas of security to gain a second opinion on it. He was asked to examine the system itself and was also shown the database and the code used to develop the system to see if he could spot any flaws in it. His comments were recorded in an interview and his views were considered to see if any changes needed to be made to the system.

The security was also evaluated in relation to reference material ranging from general security issues on password protection to issues specific to the technologies used. The advice given from these resources was collated to give general guidelines for security, (see Appendix F). The writer then checked these guidelines against the system in a step by step approach. A judgment then had to be made as to whether the security implementations were sufficient for each point on the guidelines. Ultimately evaluating the security measures of the system was a judgement of whether the security was sufficient rather than being able to say at the end of the evaluations that it was 100 percent secure

“A fundamental principle of software security is that no system is perfectly secure” (B. McCarty, 2001).

5.2.3. Results and Conclusions from Security Evaluations

Once the system had been shown to the PhD student it became clear that he had two main concerns about the security of the system. The first concern that he raised was in relation to the error reporting that had been implemented on the system. If an error occurred on the system it would indicate to the user what the nature of the problem was, however some of the error reports were leaking information about the directory structure of the system. If the MySQL server went down the following information could be seen by users.

```
Warning: MySQL Connection Failed: Access denied for user: 'carney@localhost'  
(Using password: YES) in /home/linux_b/carney/public_html/login.php on line 77
```

Error reports like this could provide information about your directory structure that malicious users may use for their advantage to compromise the system. His advice was to remove as much of the error reporting as possible. The system was updated so that error reports were only displayed in certain circumstances e.g. missing a field on a form at registration. The error messages left on the system are now specific and do not leak unnecessary information.

The other issue that he had with the system was how it processed inputs from the user specifically in regards to how data was inputted into MySQL. He noticed in one of the login forms that data was inserted directly into MySQL. It is possible for a user to alter inputs in the HTML which can manipulate SQL queries; this is known as an SQL injection attack. The PHP codes were altered to include the addslashes() function. This disturbs user inputs and stops them being able to initiate unauthorized processes.

After the system was shown to the PhD student the security checklist (see Appendix F) was used to double check the security of the system by compiling all the major security issues into one document. The writer then considered each point to decide whether it was relevant and if there were enough security measures in place to tackle the problem. It was concluded that all the points in the checklist were relevant to the security of this system and also that sufficient measures had been put in place to protect against them. This meant that for every vulnerability raised on the checklist there had been a counter measure implemented on the system to deal with it. The conclusion of the writer is that adequate security measures have now been installed on the system to protect against most common levels of attack. As with all systems there is still a chance of the system being compromised but the

chances of this have been significantly reduced due to the measures put in place.

5.3. Systems Functions Evaluation

5.3.1. Introduction to Functions Evaluation

Testing the system in relation to how it performed involved getting a range of people to accomplish a set of tasks using the system. The tasks were specifically picked to test some of the requirements that had been implemented on the system.

5.3.2. Method for Evaluating the Systems Functions

The system was tested in a similar way to previous observational tests carried out on it to see if the problems encountered in earlier implementations were still evident in the new system. The evaluations were intended to complement the evaluations already done so the main considerations were to test the main implementation changes to the system. Six final year undergraduate students and one postgraduate within the School of Computing were asked to accomplish the following tasks.

1. Attempt to register and then log onto system.
2. Find the paper written by “Card, English, and Burr”
3. Use the system to recommend the 5 best papers reviewed on the topic “Workbench” using the range of comments and scores compiled for each entry to help evaluate your choice.

The users were given a brief overview of what the system entails, but they were not shown how to use the system. They were directed to it and asked to accomplish the tasks. The results of these tasks were recorded by observations made by the writer, notes from the users and by implementing tracking on the site that kept a record of the user’s actions. The writer observed half the test subjects using the system and made notes using a simple pen and paper method. The writer recorded what tasks they were undertaking, whether they were able to accomplish the task, how long it took and general notes on any difficulties or unexpected behaviour that was observed.

The writer is aware that some users might have felt nervous being observed in this way which may prompt them to make errors resulting in data that may reflect more on the evaluation technique than the system. To guard against this some test subjects were asked to accomplish the tasks on their own. These tests were recorded using the diary method, where users are asked to record their activities on a log form (see Appendix G) as they used the system. They were asked to record the nature of the activities, how long each activity took and whether their goal was accomplished. After they

completed the log forms a short interview was conducted so that the writer could get them to elaborate on descriptions and to guard against data on the log forms being misinterpreted. When these tasks were completed the users will be encouraged to 'explore' the system. The intention behind this was that it may highlight issues that the writer had not considered in regards to earlier tasks set.

The tracking highlighted the path the users had taken through the site. The tracking was important as the writer would struggle to note all actions taken by the users and it acted as a useful backup. The tracking gave an accurate record of user's actions i.e. what pages they have visited and what actions they have performed on these pages. This information could then be collated with the other data for analysis that gave an impression of the usability of the site.

The interviews were intended to draw attention to any general issues they may have had regarding the systems functionality and usability. As the user's only had a limited opportunity to use to the system the interviews were conducted while the users still had access to the system so they could refer back to it before formulating an answer. The writer noted down the main points of the users answers but the interview was also recorded so that no points highlighted by the users were lost. Users were asked the following questions:-

With regards to the usability of the system were there any features implemented/missing that would discourage you from using the system?

Do you feel the system design is appropriate? Should some design features be altered?

Did you find any areas of the system that inhibited your ability to find the documents in the tasks?

Are there any improvements you feel should be made to the system?

Do you feel the system would be beneficial to use if you were researching documents?

Once these tests were completed the data gathered demonstrated how successful the system is in completing the tasks set out. They resulted in Yes/No answers to whether the tasks had been accomplished successfully. If all users were able to complete the tests it would show that the systems functionality works as these tasks are general spread of the changes to the system. However a note of caution is that these tasks obviously needed to be completed in a reasonable time for the systems functionality to be useful.

The data gathered from each user shows how long each activity took. The data was compiled to see if any activities held the users up to see if there was a generic problem with the system that needed be fixed, i.e. learning to use the system could have held users up and may indicate that a better help tutorial is needed. As an extra test these times were gauged against a base set of times; these base sets were how long it took the writer to perform the tasks. The users did not have previous experience of

the system and it was interesting to consider the difference in times with the writer who obviously knew the system well.

The differences in the time taken for certain users to accomplish a task and the way in which they have performed the tasks had to be explored. The tracking was used to show how many users accomplished the task in the same way. The main aspect in analysing data was to consider why these processes may have differed and did it result due to ambiguity in the system. If all the test subjects had completed tasks in a different way then although it would demonstrate flexibility for the system it may highlight potential problems in that users are confused by it.

General issues observed by the writer or noted by the users were considered on an individual basis using the methods above to analyse why they occurred. The writer found that the interview resulted in more general data but it was useful for gathering user's perceptions of the system. The questions chosen were open questions to increase the chances of generating ideas or issues that the writer may not have considered. The answers highlighted the reasons for earlier problems in accomplishing the tasks. The answers were also compiled and considered as a whole; to see if most users commented on problems in relation to the same aspect of the system.

5.3.3. Results and Conclusions from Systems Functions Evaluation

Once the evaluations were complete it was clear that the system offered the necessary functionality to complete the tasks as all users managed to do this. However some noticeable disparities in times taken to accomplish the tasks highlighted some minor issues within the system. The complete list of evaluation results can be found in Appendix H. The average time to complete task 1 was 82 seconds and all users completed this task with relative ease

Implementing a search with a Boolean 'and' function was one of the main requirements for version 2 of the system and the fact that all users managed to complete task 2 using it shows that this requirement was fulfilled. Task 2 had a slightly longer average time of 96 seconds compared to task 1 which does not seem that large but when you compare this to the base time set by the writer of 20 seconds there is quite a large difference. This gap in times was caused by users having to study which fields to enter the author's names into. Once users gain experience of the system they will understand which fields to use and the time taken to perform these sorts of operation is likely to fall sharply.

Task 3 took by far the longest with an average time of 281 seconds which was unsurprising considering the more complex nature of the task although this should not be used to mask some problems in the system that were found at this stage. All users including the writer had difficulties

navigating to the previous page and on many occasions users were presented with a no page display. The postgraduate student tried clicking other available links and got stuck in a loop where he was unsure where to navigate too which had a large impact on the time it took him to complete the task.

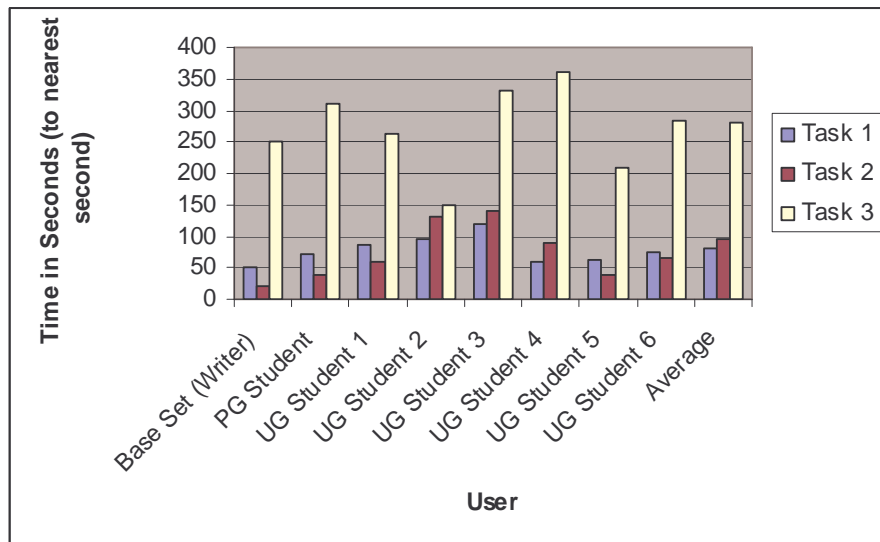


Figure 5.1 - Graph showing the time taken to complete the tasks.

The nature of task 3 meant that there was no definitive answer and it was more of judgement of what 5 papers would be best, but as there were only certain criteria that they could be judged on you would expect to see some similar results. There were 8 available papers to find for the topic “workbench” and all users found them although most picked different combinations for their 5 papers best. 4 of the papers had the best quality rating of 3 and 7 of the users choose to use these for their recommendations and then the last choice was usually a judgment of whether the user felt a more relevant comment on a paper might be more important than the year published or vice-versa. Only the second undergraduate student to be evaluated got very contrasting results but it is likely that these were due to the fact they were the quickest to complete the task by some margin which indicates they may have rushed the task. Figure 5.2 highlights the consistency of results showing that 5 papers were clearly recommended more. This demonstrates that the extra information that the system provides for distinguishing between records is logical and understood by most users.

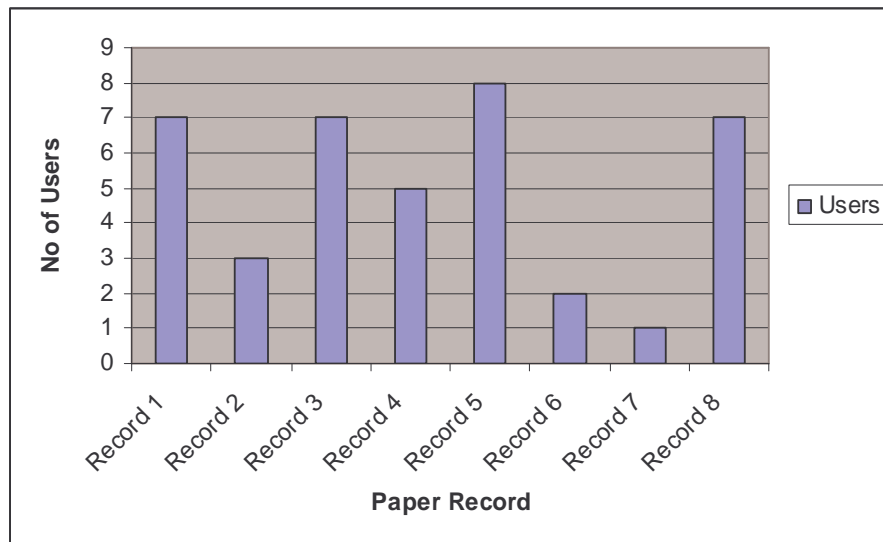


Figure 5.2 Graph showing the papers chosen from task 3.

5.4. Issues Raised from Interviews

The results of the interviews highlighted problems with the usability of the system and most users went on to describe some potential improvements that they felt may cure issues that they had. Every user commented on the problems that they had in returning to the previous page when the page display was lost and most said they would have liked to have had a page back button displayed on the system.

Some of the users were confused by the home page of the system and commented that there were too many options available and the extra information confused them. Various opinions were given by the users on how the home page should be improved ranging from removing some text information into the tutorial to just having one search field on the main page i.e. Google. The general consensus was that it should be simplified in some way. One user commented that the option of controlling the order of results was a big distraction on the home page. They thought this option should be removed and that allowing users to alter the results after they had seen them was sufficient and kept in conventions set by most search engine.

A couple of the users commented that they expected to be able to gain full details of the papers by clicking the paper title and thought this would be a better option than having to click a separate full details link that they felt was not immediately obvious.

The PhD student thought the advanced search was very similar to the normal search. He commented that he thought the normal search was more like an advanced search and maybe an interesting feature

to add would be a simplified search which allowed you to search all tables from just one field.

5.5. Comparisons to other Knowledge Space Systems

Citeseer is a well known knowledge based system that contains scientific literature. The system is quite similar to the system developed by the writer. Probably the biggest difference with the Citeseer system is that this is designed to be used by thousands of people whereas the writers system is more focused on a tight knit research group of trusted users. In terms of how Citeseer works it offers a single search box which will allow you to search through all the documents that they have. This is big difference between their system and the writers in that they provide one generic search box whereas the writers system refines the search a lot more so that it is possible to search on just one field. When Citeseer finds documents it display the most relevant documents first, although how this relevance is calculated is not entirely clear. The writers system will return the most relevant documents in relation to the users keyword input and then they have the option of altering the returned documents by which ever field they consider most relevant to them. The writers system has a potential advantage here as documents that are relevant to one person may not be relevant to another. Allowing them to choose which field is most important allows them a lot more freedom. One potential advantage Citeseer has is that it shows related documents to the ones that users search for. On the writers system it highlights related keywords and then the users would have to manually search these themselves.

Xerox have developed a system called the knowledge pump that “provides users with personalised recommendations for things to read” (www.Xerox.com, 2005). This system works by users setting preferences on which subjects and documents are relevant to them but these preferences are updated as they use the system “Profiler agents track and map each user's interests” (www.Xerox.com, 2005). If the research knowledge space was extended to incorporate a number of research groups implementing technology that helped recommend documents could be a valuable asset to the system.

A Bibliographic database could be considered similar to the system that has been developed as they are used to provide references to documents. However when you compare the knowledge space system to bibliography databases you can see that the knowledge space system developed is a lot more interactive and is not just a case of displaying bibliographic references. An example is the ICAME Bibliography system that just lists all their references to papers and does not offer any comments about these. The writers system offers the chance to search and then order documents by different keywords but in the ICAME system there is no option to do this, it simply lists the references alphabetically on one web page. Trying to pick out relevant data from the system could be compared to retrieving information from Dr. Ruddle original text file. It is a very time consuming process that almost makes it unusable to first time users.

5.6. Overall Conclusion

The final test was conducted using the data gained from the earlier tests to see if they for filled the requirements stated in the requirements analysis. The data gained from the evaluations was cross referenced against the requirements to gain simple yes or no answers to whether these requirements had been implemented. All of the requirements specified under MuSCoW rules as ‘must’ be implemented and ‘should’ be implemented are in place on version 2 of the system. Some of the ‘could’ be implemented requirements have also been implemented on version 2 of the system. This means that the system is capable of performing all the requirements that it was expected to.

One of the most common reasons for project failure is due to the requirements changing so it was important to demonstrate the system to the problem owner. After this demonstration the problem owner confirmed that the system fulfilled the requirements that he wanted and that his requirements had not changed. A full list of the implemented requirements can be found in table 4.1.

5.7. Possible Further Enhancements to current system

The evaluations have highlighted a number of extensions that could be made to version 2 of the system and also major extensions that would require a new system to be developed. The evaluations highlighted that the main problems with the version 2 system were in regards to its usability rather than the functionality that it offered. The writer felt that many of the suggestions made in the interviews for improvements were valid so some of these improvements were implemented. The summary of potential improvements is listed in table 5.1.

No.	Potential improvement	Potential Improvement Implemented?
1	Redesign home page so that the ordering or results is removed.	Yes
2	Prompt user to type in password twice.	Yes
3	Add page return button so that page display is not lost when going back to previous page.	Yes
4	Provide a single search option that searches all tables.	No
5	Present paper titles as hyperlinks to bring up full details.	No
6	Provide links to the papers.	No

No.	Potential improvement	Potential Improvement Implemented?
7	Implement recommendations for users so when new papers are reviewed a user is notified if they meet there personal criteria.	No
8	Implement Symantec layer	No

Table 5.1 Potentail Further Enhancements

6. Further Enhancements - Semantic Version

The major extension to the project to improve the knowledge space would be to add a semantic layer to the system and the feasibility of doing this was explored. This chapter will give an introduction to semantic technologies and the benefits they can bring, specifically in regards to how these technologies could be used to implement an improvement on the current system. The possible technologies that could be used to implement a semantic system will be examined and the architecture for the semantic system will be presented. Finally a conclusion will be given to whether a semantic system would be feasible and whether it would be beneficial to produce such a system.

6. 1. Introduction

The knowledge space system that has been developed relies in part on Web technologies to help share the information that is contained within the original source file. The ability to distribute information in various situations, i.e. within a research group has played a large part in the World Wide Web's success. "Information retrieval technology has been central to the success of the Web" (Finn et al, 2005).

In its present state most of this information can only be processed by humans but the Semantic web is trying to change this and by better incorporating computers they hope to improve information retrieval and processing "information is given well-defined meaning, better enabling computers and people to work in cooperation" (Berners-Lee, Hendler and Lassila, 2001). The Semantic web will automate tasks by processing information from various sources "software agents roaming from page to page can readily carry out sophisticated tasks for users" (Berners-Lee, Hendler and Lassila, 2001).

6.2. Advantages of a Semantic System

Semantic technologies are an exciting new area of interest but it would be foolish to implement them if they did not offer improvements over the version 2 system, so these potential benefits had to be considered. In the present situation these semantic technologies could be used to improve the information retrieval of the search engine by exploiting the relationships between source information and the keywords. The objective would be to have the keywords linked in such a way that when a user searched a keyword or a combination of keywords the system would recognise and process these keywords. The semantic system could then return related keywords that share relationships. A big advantage that a semantic version of the system would have is that it would help to bridge the

knowledge gap between experienced users and novice users. Less experienced users are unlikely to understand the relationships between certain keywords. When the users search the systems records it is likely that they may miss large chunks of records as they don't realize the relationship between certain keywords. As new relationships are formed in the semantic enabled system it would be able to adapt and process these new relationships between keywords to present results that might have been deemed irrelevant in non semantic versions.

One of the major intended uses of the system is to assist users in picking whether to read certain papers. On Version 2 of the system users judge which paper to read by processing the various comments and ratings entered for each record. A major benefit that a semantic system could have is that it could do this processing for the user when a search is done. The system envisaged would then present the records in order of importance. A semantic system could dramatically lower the time taken to achieve tasks like task 3 in the evaluations. Users had to recommend 5 papers for a given topic and the average time was 281 seconds for this task. Most of the users time was spent processing how good a paper was so a system that did this automatically would have obvious benefits. A semantic system could also allow for the expansion of the system over various research groups and take advantage of their combined knowledge.

6.3. Technologies for implementing semantic system

Two of the main technologies used to develop the semantic web are Extensible Markup Language (XML) and the Resource Description Framework (RDF). RDF's were developed for the semantic web to counter the problems that would be presented by just using XML but it should not be viewed as an alternative to XML. It is designed to build on XML and Universal Resource Identifier's (URI) technologies. RDF's were designed to be more constraining than XML. The goal is to set standards for defining data so you can't have the same data defined in different ways, which is possible in XML.

XML can be used to structure documents but it does not goes as far as defining what this structure means, this is done in the RDF "Meaning is expressed by RDF, which encodes it in sets of triples" (Berners-Lee, Hendler and Lassila, 2001). The reason data is encoded as a set of triples is that these represent a set of URI's that store the object, attribute and value names (www.HP.com, 2005) which gives a way of defining elements on the semantic web, these triples can be written in XML. Defining them as triples gives elements a standard "URIs ensure that concepts are not just words in a document but are tied to a unique definition that everyone can find on the Web" (Berners-Lee, Hendler and Lassila, 2001). The original source file compiled by Dr

Ruddle would obviously have to be altered so that it represented entries like keywords in this RDF format. The most efficient way to do this would be to use software to parse the document into RDF format. Most development languages have tools built to do this but it would be advisable to use a language like PHP as it would also allow you to insert the RDF's straight into the database.

RDF's provide a standard way for defining elements but this does not stop people from defining the same elements with different URI's. For the semantic web to build relationships between concepts it needs to be able to recognize that the same concept may be defined in different ways and this is achieved by building ontologies. These ontologies that are built are rules that define the relationship between different concepts. The ontologies could be developed specifically for the closed research group system but it may be advisable to develop them in the Web Ontology language (OWL) in case the system was expanded in the future. OWL allows data to be distributed across many applications and is being developed as the standard for web ontology languages. The semantic system developed would process the ontologies written in the OWL, it is "designed for use by applications who need to process the content of information" Cardoso and Sheth (2004).

The ontologies written in OWL would still have to be processed to create a set of rules for the keywords that demonstrated how they were linked and the relationships between them. There are various technologies that can be used to help index the entries into ontologies. The two main technologies are Jena that was developed in conjunction with HP which is an open source solution that allows you to export the RDF documents into database via an Application programming interface (API). An alternative to this is Sesame, which is more concerned with the storage of the ontologies but has slightly less support on the API side. Either of these technologies could be used to produce the indexes for the semantic system as they both allow the stored ontologies to be queried.

Developing the ontologies by using RDF's allows them to be read by the semantic web but on its own it would not be classed as the semantic web. These technologies are used to present the data into a machine readable form so that that can then be accessed by semantic web agents. These web agents could be used to allow data to be shared between different research groups which could help produce more advanced ontologies that could in turn help produce better searches within the system.

6.4. High level Systems Architecture

A High level architecture for the envisaged system is presented in figure 6.1.

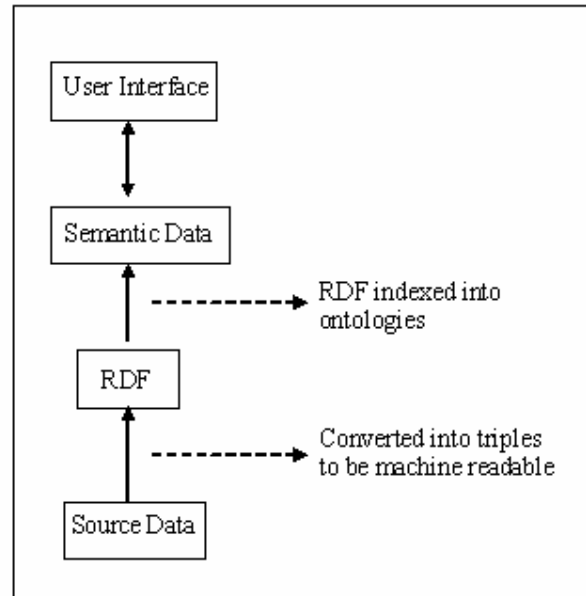


Figure 6.1 - High level for Semantic System

The source data in the text document would have to be altered so it is written in the basic semantic web language. Once the data is in RFD form it has to be processed into ontologies so that the relationships between the various entries can be understood by the application. These indexed entries are then stored in a database and it is this database that will be searched by the application which the user will access via the user interface.

6.5. Conclusion

Developing a semantic version of the system would be a complex task and due to time constraints it was not possible for the writer to develop it. It is the writer's conclusion that a semantic system would offer potential improvements that would not be possible by extending the functionality of the current system using the implemented technologies. The major area of improvement that the semantic system could offer is to process search results. This semantic system is likely to be quicker to use but should also give a better scope of results as it will process the relationships between certain keywords. One note of caution is that any system developed should adhere to the standards laid out at these early stages of the semantic webs development otherwise the system developed may not be compatible with future advances.

References

Avison, D & Fitzgerald, G, (2003), Information system development: Methodologies, techniques and tools, Higher Education.

Avison, D and Shah, H, (1997), The Information Systems Development Life Cycle: A First Course in Information Systems, McGraw-Hill.

Berners-Lee, Tim, Hendler, James and Lassila, Ora (May 2001), The Semantic Web [Online]
URL < <http://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21>> [Accessed 01-04-2005].

Brown, JS and Duguid, P, (2000), The Social Life of Information, Harvard Business School.

Cadle, J & Yeates, D, (2004), Project Management for Information Systems, pp 88, Pearson.

Cardoso, Jorge & Sheth, Amit, (2004), Introduction to Semantic Web Services and Web Process Composition in Semantic Web Services and Web Process Composition (2005), pp 1-13, Springer.

Citeseer, [Online] URL < <http://citeseer.ist.psu.edu/cs> > [Accessed 21-04-2005].

Coakes, E, Willis, D & Clarke, S, (2002) Knowledge management in the Socio Technical World: The Graffiti Continues, pp1-3, Springer.

DataModel.org, (2005), Data Model Cardinality, [Online]
URL < <http://www.datamodel.org/DataModelCardinality.html> > [Accessed 06-01-2005].

Elmasri & Navathe, (2000), Fundamentals of Database Systems, pp 15, 45-80, 202, Addison Wesley.

Finn, Tim, Mayfeild, James, Fink Clay, Anupam, Joshi, and Cost, Scott, (2005), Information Retrieval and the Semantic Web [Online]
URL < <http://ebiquity.umbc.edu/v2.1/paper/html/id/185/>> [Accessed 02-04-2005].

HP (2005), Introduction to Semantic Web Technologies [Online]
URL < <http://www.hpl.hp.com/semweb/sw-technology.htm> > [Accessed 01-04-2005].

ICAME [Online] URL < <http://helmer.aksis.uib.no/icame/icame-bib3.htm> > [Accessed 21-04-2005].

Laudon, K & Laudon, J, (2002), Management information systems: Managing the Digital Firm, Prentice Hall.

Le Peuple, J & Scane, R, (2003), User Interface Design, pp 15, Crucial.

McCarty, B, (2001), PHP 4: A Beginner's Guide, pp 506, Osborne/McGraw-Hill.

Maciaszek, L, (2001), Requirements Analysis and System Design: Developing Information Systems with UML, Addison-Wesley.

Ravden, S. & Johnson, G. (1989) Evaluating Usability of Human-Computer Interfaces: a practical method. Ellis Horwood Limited.

Shneiderman, B & Plaisant, C, (2005), Designing the user interface, pp 15, 60, 142, Addison Wesley.

W3 (2004), Web Ontology Language [Online]

URL < <http://www.w3.org/2004/OWL/> > [Accessed 01-04-2005].

White, W, (2001), Enabling eBusiness: Integrating Technologies, Architectures and Applications, pp222-225, John Wiley & Sons, Ltd.

Xerox, (2005), Knowledge Pump [Online]

URL < <http://www.xrce.xerox.com/programs/kc/knowledgepump.html> > [Accessed 13-04-2005].

Appendix A - Personal reflection on Project

In many situations when you are presented with new challenges you will almost certainly feel a degree of apprehension at some point and this was no different for me in regards to the project. I am studying Information Systems at University and I was asked why I chose to do a project that involved coding as many students in my course avoid this style of project. At the time of choosing the project I felt that it would be a challenge worth undertaking where at the end I would have a finished product that I could say I was involved in developing. The coding was the most difficult aspect of the project and at times I had doubts to whether I would get all the functions for the system completed. However I do feel the rest of the sections within the methodology went quite well and producing a product does give you well defined goals and stages to aim for.

Looking back on the project, the most difficult task I have had to deal with was trying to get ownership of the scope of the project. By this I mean the inherent difficulties in trying to re-engineer a previous project that I had no involvement in. Some of the PHP files within the project had nearly 5000 lines of code in them and although there were some comments in the files it was sparse and I found it quite vague at best. I will be the first to admit I under-estimated the difficulties in implementing simple changes within the system. Changes to one file would result in changes to other files that were completely unexpected. With hindsight I would never have tried to fix certain fragments of code that were not fully functioning. I think the implementation stage of the project could have been finished much quicker if I had decided at an early stage to scrap certain bits of code and just to start from scratch for some aspects. A lot of time was wasted trying to understand what certain variables did but as I had no experience of this sort of project I knew no better at the time.

The true test of whether you have learnt any lessons from projects like this is if you would tackle it in the same way again if you were faced with the similar problem and I would definitely say no to this. My advice would be that unless you can completely understand what certain aspects of code are; do not use them as they have a habit of effecting other sections within the system at a later stage. Holding on to sections may seem like the easier option as you don't want to redo sections but as I found trying to tweak aspects of something you didn't create can prove very difficult. It would have been hypocritical of me to complain about these problems and then not try to develop a solution for future system administrators who may need to alter the system. To rectify this problem a user manual for the system was created (see Appendix H) which listed all of the files and their uses. I have also tried to improve the comments within the files and have strived to make the files that I have created as simple as possible to understand.

I have tried to be as professional and polite as possible when dealing with others who have helped

with my project. I took the view early on that I did not want to be in the position where people helping me with my project were put out or delayed because I had missed deadlines due to a poor work ethic and just thinking I could go through the motions. When undertaking the evaluations I knew lots of the people helping me would be busy with their own deadlines and so I wanted to streamline the evaluations as much as possible. One example of where I tried to achieve this was by using a Dictaphone in interviews so the users did not have to wait for me to write down their answers.

Time management is obviously an important factor of any project and it was no different for me. At times I did not stick to deadlines that I set myself but as I set these quite early on it allowed for a certain 'slippage'. Leaving too much to the last minute allows no margin for error and results in more problems in the long term i.e. if you only finish implementation during Easter you will struggle to get people to test it as most students will have gone home. Telling people to start early if they want to do well is obvious but it really is true. Help from tutors and assessors is available to students at an early stage but if they do very little for the mid project report or draft chapter there is only so much advice that they can give. These deadlines should be approached like final deadlines and that way you will receive much more valuable comments back which should help guide the rest of the project.

This project is likely to be developed further next year and I would advise the student who undertakes the project to thoroughly research the previous systems to get a context for the problem. If they are new to semantic technologies their approach to tackling the problem should be carefully considered. Developing the system with a prototyping approach would allow them to get some experience in the technologies before completing a large body of design that may end up not being appropriate to the problem. They should also be aware that the implementation of the system will be a sizeable task that will probably take longer than they might imagine which was something I definitely learnt during this project.

Appendix B

Discussion of Project Plan and Project Methodology

Project Schedule

The original project schedule (Table 1.1) had to be revised after it became clear that the implementation of version 2 was going to take a lot longer than originally envisaged (the reasons for this are documented in the implementation section). After this assessment was made a new plan was created so that the project did not just float along and lose productivity. The new plan allocated more time to the design and development of the system to reflect the changing nature of the project compared to the original plan. In the plan some tasks were carried out at the same time as other tasks as it was not be possible to complete certain tasks like systems evaluation over the Easter break.

Stage	Date	Objective	Milestone
1	30/09/2004	Submit Project Preference form	
2	14/10/2004	Discuss the project idea with supervisor	
3	22/10/2004	Complete Aim and Minimum Requirements form	
4	23/10/2004 – 30/10/2004	Research into the context of the problem	
5	30/10/2004 – 10/11/2004	Evaluation of Cai's system	Evaluation section
6	11/11/2004 – 26/11/2004	Requirements Capture	Finalized requirements for version 2 system
7	26/11/2004 – 09/12/2004	Write up of Mid-Project Report	Mid-project Report
8	10/12/2004	Submit Mid-Project Report	
9	13/12/2004 – 22/12/2004	Research into how to implement the system	
10	03/01/2005 - 10/01/2005	Design of version 2 system	Design section for the project.
11	22/01/2005 – 25/02/2005	Implementation of version 2	System built.
12	25/02/2005 – 01/03/2005	Write up Implementation section of report	Implementation section for project.

Stage	Date	Objective	Milestone
13	01/03/2005 – 09/03/2005	Propose evaluation measures to test implementation of system.	Plan for evaluations.
14	09/03/2005 – 17/03/2005	Carry out evaluation tests on system	
15	11/03/2005	Submit table of Contents and Draft Chapter (Implementation Section)	Implementation section and table of contents.
16	12/03/2005 – 28/03/2005	Background reading on Semantic technologies	
17	18/03/2005	Progress meeting with supervisor and assessor.	
18	22/03/2005 - 01/04/2005	Write up of Evaluation section.	Evaluation section for the project.
19	02/04/2005 - 07/04/2005	Implement further enhancements suggested in evaluations.	Completed version 2 system.
20	08/04/2005 – 16/04/2005	Further Enhancements Section - Background reading on Semantic technologies. Conclusions on the potential advantages and the viability of semantic version 3 system	Further Enhancements section for project.
21	17/04/2005 - 26/04/2005	Completion of final Report. Project reflection & Proof Reading	Final Report.
22	27/04/2005	Submit Report to CSO	
23	29/04/2005	Submit Report in PDF format	

Project Methodology

The original plan for developing the version 2 system is presented below.

Evaluation of Cai's System and Requirements Capture.

Design of new System.

Implementation and Testing of changes.

System Evaluation.

The project progressed in the same way as the original plan; however the design and implementation sections were extended significantly. This meant that the major extension to the project, the Version 3 system could not be implemented due to time constraints. The project was developed by prototyping that was loosely based on the SDLC to give it a more rigid structure. This prototyping worked well because it found out at a relatively early stage that extra time was going to have to be spent on the design and implementation sections. This allowed the project plan to be revised and resulted in the completion of this more accurate plan.

The final processes involved in the completion of the project were:

- Evaluation of Cai's System and Requirements Capture.

- Design of new System.

- Implementation and Testing of changes.

- System Evaluation.

- Implementation of some Further Enhancements

- Background Reading - Development of Further Enhancements section

Appendix C

Cross Referencing sections to project deliverables

As the project does not follow a completely standard approach the sections produced were cross referenced against the marking scheme to indicate what each section was tackling.

Section 1 -	Project Management Understanding the problem
Section 2 -	Project Management Understanding the problem
Section 3 -	Delivery of solution
Section 4 -	Delivery of solution
Section 5 -	Evaluation of solution
Section 6 -	Understanding the problem Further Enhancements
Appendix A -	Reflect upon the project experience
Appendix B -	Project management.
Appendix C -	-
Appendix D -	Understanding the problem Design
Appendix E -	Delivery of solution
Appendix F -	Evaluation of solution
Appendix G -	Evaluation of solution
Appendix H -	Evaluation of solution
Appendix I -	Further Enhancements

Appendix D

Heuristic Evaluation Checklist

From the list of Nielsen's ten heuristics a checklist was compiled, which will be analysed and the data interpreted so a view on the usability of the site can be given.

1. Visibility of system status

- 1.1 Does the system display messages telling the user what has happened after an action has been performed?
- 1.2 Is it clear when certain data is selected? I.e. highlighted icon.
- 1.3 Is there system feedback when an operation is performed?
- 1.4 If multiple menu options can be selected is there visual feedback of this?

2. Match between system and the real world

- 2.1 Are icons familiar
- 2.2 Are menu's used, organized appropriately
- 2.3 Are menu options organized appropriately
- 2.4 Are the tasks described to user's familiar

3. User control and freedom

- 3.1 Are users prompted to confirm actions that have severe consequences. I.e. delete record
- 3.2 Does the system allow the user to undo there actions?
- 3.3 Can users reduce data entry time by editing existing data?
- 3.4 Can users change an earlier menu choice?
- 3.5 Can the user set the own session defaults?

4. Consistency and standards

- 4.1 Is the system consistent, I.e. menus
- 4.2 Do the searches follow the normal conventions?, I.e. pressing return starts search
- 4.3 Is the font between 9-12 point font for the main bodies of text?
- 4.4 Is reverse block avoided for main bodies of text?
- 4.5 Do keys with similar actions look similar?
- 4.6 Are error messages consistent?

5. Error prevention

- 5.1 Does the system warn people if they are about to remove/update data?
- 5.2 Are these prompts unambiguous ?
- 5.3 Are the prompts brief?
- 5.4 Is there a sound to signal an error?
- 5.5 Are the error prompts constructive rather than imply criticism of user
- 5.6 Do the error messages inform the user of the severity of the problem, I.e. Minor/Major

6. Recognition rather than recall

- 6.1 Have prompts been formatted to allow for easy scanning of the page I.e white space around them.
- 6.2 Is the format of the page designed to lead users to important information
- 6.3 Have items been organized into appropriate areas and headings used to distinguish between these areas?

7. Flexibility and efficiency of use

- 7.1 Has lowercase lettering been used where possible?
- 7.2 Will the system support different levels of users? I.e. Novice to expert.
- 7.3 Are there shortcuts provided to select menu options?
- 7.4 Does the system allow users to find next/find previous options when an entry is being viewed?
- 7.5 Does the system remember users so they don't have log on every time?

8. Aesthetic and minimalist design

- 8.1 Is there a good brightness and colour contrast between background colours and the features on the site
- 8.2 Is only information that is needed displayed on the screen?
- 8.3 Are the different features available easily distinguished?
- 8.4 Are menu titles brief?
- 8.5 Is each low level menu choice assigned to only one higher level menu?

9. Error recovery

- 9.1 Are error messages presented to the user?
- 9.2 If they do not have sufficient access rights to alter data is this stated?
- 9.3 If searches produce zero results is any message presented to help user?

10. Help and documentation

- 10.1 Is information provided to give a demonstration on how to use the system?
- 10.2 Is this help clear and easy to find?
- 10.3 In advanced features of the system is there sufficient information to explain them?
- 10.4 Is there to an option to see what variables exists within a specific search field?

Appendix E**Test Plan**

Sample test plan for the adding, updating and deleting of papers.

Function Tested	Test Performed	Expected Outcome	Success
Add Paper	Add paper to system filling in all fields	The paper and all relevant fields should be added to the database.	Yes
	Add paper filling in just the title and author name	The paper and all relevant fields should be added to the database.	Yes
	Add paper when user level is set to 0	Refuse user as there user level is not high enough.	Yes
	Add paper when user level is set to 1	Add paper	Yes
	Add paper when user level is set to 2	Add paper	Yes
Delete Paper	Delete paper when user level is set to 0	Refuse user as there user level is not high enough.	Yes
	Delete paper when user level is set to 1	Refuse user as there user level is not high enough.	Yes
	Delete paper when user level is set to 2	Allow paper to be deleted	Yes
Update Paper	Update a paper on the system	The paper and all relevant fields should be updated to the database.	No
	Update a paper on the system when user level is set to 0	Refuse user as there user level is not high enough.	Yes
	Update a paper on the system when user level is set to 1	The paper and all relevant fields should be updated to the database.	No
	Update a paper on the system when user level is set to 2	The paper and all relevant fields should be updated to the database.	No

Appendix F**Security Checklist**

This checklist was compiled from the writers own knowledge on security, various advice from websites and from various security and PHP related literature. It was used to see

No	Security Issue	Features Implemented?
1	Authenticate users who use the system by implementing password protection.	Yes
2	Encryption of passwords in the database.	Yes
3	Remove database connections and passwords from files	Yes
4	Authorizing users to have specific privileges	Yes
5	Register Globals() Switched off	Yes
6	Check Password Strength	Yes
7	Error handling	Yes
8	Avoid False Uploads	Yes

Appendix G**Log Form**

Name of participant:

Date of test:

Level of study:

Please attempt the following tasks and then answer the questions:

Access the system at <http://www>

Task: 1. Attempt to register and then log onto system.

Were you able to complete the task? YES/NO

Time taken to complete the task?

Describe the main steps taken to complete the task?

Did you encounter any difficulties in completing the task?

Any further comments ?

Task: 2. Find the paper written by “Card, English, and Burr”

Were you able to complete the task? YES/NO

Time taken to complete the task?

Describe the main steps taken to complete the task?

Did you encounter any difficulties in completing the task?

Any further comments ?

Task: 3. Find papers on the topic “WWW” and view the results in descending order.

Were you able to complete the task? YES/NO

Time taken to complete the task?

Describe the main steps taken to complete the task?

Did you encounter any difficulties in completing the task?

Any further comments ?

Appendix H

Results from Evaluations

User Writer (for base set of results)

Task 1

Completed	Time	Route Taken	Issues
Yes	50 sec	Clicked link to register new user	None

Task 2

Completed	Time	Route Taken	Issues
Yes	20 sec	Searched author field list	None

Task 3

Completed	Time	Route Taken	Issues
Yes - Chose papers 1, 3, 4, 5, 8.	4min 10 sec	Searched keyword field, examined full details. Used the quality rating to judge paper and then used the summary to see if workbench was paper mentioned and took the most recent paper as there was a choice of two.	Problem in returning to previous page

User PG Student 1

Task 1

Completed	Time	Route Taken	Issues
Yes	1min 11sec	Clicked link to register new user	None

Task 2

Completed	Time	Route Taken	Issues
Yes	38 sec	Searched Author field	None

Task 3

Completed	Time	Route Taken	Issues
Yes - Chose papers 1, 5, 6, 7,8.	5 min 10 sec	Searched keyword field, examined full details. Used the comments quality rating and personal knowledge to judge papers.	Unsure how to return to previous page. Clicked on the advanced search and he got stuck in a loop trying to find original results page.

User UG Student 1

Task 1

Completed	Time	Route Taken	Issues
Yes	1min 26 sec	Clicked link to register new user	None

Task 2

Completed	Time	Route Taken	Issues
Yes	59 sec	Originally searched wrong field so clicked back when given wrong results.	Confused by different options available to him. Clicked back and he lost page display

Task 3

Completed	Time	Route Taken	Issues
Yes - Chose papers 1, 3, 5, 6,8.	4min 22 sec	Searched keyword field, examined full details. Used the quality rating to judge papers.	Kept trying to click the paper title as he thought this would display details on paper. Again unsure on how to return to previous page.

User UG Student 2

Task 1

Completed	Time	Route Taken	Issues
Yes	1min 37sec	Clicked link to register new user	Took a little bit of time to recognize the new user link.

Task 2

Completed	Time	Route Taken	Issues
Yes	2min 12 sec	Used Author field	Was unsure of which field to search, so took time to make decision

Task 3

Completed	Time	Route Taken	Issues
Yes - Chose papers 1, 2, 3, 4, 5.	2min 30 sec	Searched keyword field, and took the first 5 papers returned.	Kept trying to click the paper title as he thought this would display details on paper. Again unsure on how to return to previous page.

User UG Student 3

Task 1

Completed	Time	Route Taken	Issues
Yes	2 min	Clicked link to register new user	Took a little bit of time to recognize the new user link.

Task 2

Completed	Time	Route Taken	Issues
Yes	2min 20 sec	Used Author field	Read details on page then chose correct field

Task 3

Completed	Time	Route Taken	Issues
Yes - Chose papers 2, 3, 4, 5, 8	5 min 30 sec	Searched keyword field, and took the most recently published papers and then used quality ratings to judge remaining.	Return page displayed lost.

User UG Student 4

Task 1

Completed	Time	Route Taken	Issues
Yes	1min 0 sec	Clicked link to register new user	None

Task 2

Completed	Time	Route Taken	Issues
Yes	1min 30 sec	Searched author field list	Slightly confused which slowed decision down.

Task 3

Completed	Time	Route Taken	Issues
Yes - Chose papers 1, 3, 4, 5, 8.	6min 0 sec	Searched keyword field, examined full details. Used the quality rating to judge paper and then used the summary to see if workbench was paper mentioned and took the most recent paper.	Problem in returning to previous page

User UG Student 5

Task 1

Completed	Time	Route Taken	Issues
Yes	1min 03 sec	Clicked link to register new user	None

Task 2

Completed	Time	Route Taken	Issues
Yes	40 sec	Searched author field list	None

Task 3

Completed	Time	Route Taken	Issues
Yes - Chose papers 1, 2, 3, 5, 8.	3min 30 sec	Searched keyword field, examined full details. Used the quality rating and then picked extra one.	Problem in returning to previous page

User UG Student 6

Task 1

Completed	Time	Route Taken	Issues
Yes	1min 14 sec	Clicked link to register new user	None

Task 2

Completed	Time	Route Taken	Issues
Yes	1min 5 sec	Searched author field list	None

Task 3

Completed	Time	Route Taken	Issues
Yes - Chose papers 1, 3, 4, 5, 8.	4min 45 sec	Searched keyword field, examined full details. Used the quality rating and then picked a paper that mentioned workbench in the comments.	Problem in returning to previous page

Appendix I

User Manual for Version 2 Multi User Knowledge Space

Table of Contents

1 Introduction

2 System Functions Overview

3 Controlling Users Access

3.1 Authorizing User/Controlling user access levels.

3.2 Removing Users

4 Managing the Database

4.1 Creating Back-up's for Database Tables

4.2 Restoring Database Tables

5 File Description

5.1 File list – Including descriptions

5.2 Parser Description

1. Introduction

The manual is designed to help future system administrators run the system. It demonstrates the steps involved in controlling user access on the system. It also shows how to backup and then restore database tables. The final section of the manual is designed as documentation for the version 2 system. It lists the files used to develop the system and a description of what these files do.

2. System Functions Overview

Tutorial for the functions that are provided on the system is found at : -

<http://test.iri.leeds.ac.uk/internal/carney/example.html>

3 - Controlling Users Access

Users will register their details using the system, however some of the security controls implemented mean that certain details can only be changed by editing their details from within the database. Access to the database is controlled by phpMyAdmin via

<http://test.iri.leeds.ac.uk/mysql/iri02/>

The phpMyAdmin system will prompt the administrator to enter their username and password to enter the system (these will be provided to system administrator at hand-over)

3.1 - Authorizing User/Controlling user access levels.

For security reasons new users can only gain access to the system once their authorization level has been changed from 0 to 1. This can be altered in easy steps.

Step 1 – Select the user table from the database.

The screenshot shows the phpMyAdmin interface for the 'carney' database. The 'users' table is selected, and its structure is displayed. The table has the following fields:

Field	Type	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> userid	int(25)		No		auto_increment	
<input type="checkbox"/> first_name	varchar(25)		No			
<input type="checkbox"/> last_name	varchar(25)		No			
<input type="checkbox"/> email_address	varchar(100)		No			
<input type="checkbox"/> username	varchar(25)		No			
<input type="checkbox"/> PASSWORD	varchar(50)		No			
<input type="checkbox"/> user_level	enum('0', '1', '2', '3')		No	0		
<input type="checkbox"/> activated	enum('0', '1')		No	0		

Below the table structure, there are options to 'Check All' or 'Uncheck All' for the selected fields. The 'Print view' and 'Propose table structure' buttons are also visible.

The 'Indexes' section shows a PRIMARY index on the 'userid' field. The 'Space usage' section shows the table size and index size. The 'Row Statistics' section shows the number of rows, row length, row size, and next autoindex.

The 'Run SQL query/queries on database carney' section contains the query: `SELECT * FROM 'users' WHERE 1`. The 'Fields' list on the right shows the available fields for the query.

Step 2 - Select a User to Edit

Click this tab to edit details about a particular user

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

Sort by key: None Go

	userid	first_name	last_name	email_address	username	PASSWORD	user_level	activated
<input type="checkbox"/>	23	test	subject	test@test.co.uk	test	098f6bcd4621d373cade4e832627b4f6	0	0
<input type="checkbox"/>	17	sam	sam	sam	sam	332532dcfaa1cbf61e2a266bd723612c	1	1
<input type="checkbox"/>	22	Royce	Neagle	royce@comp.leeds.ac.uk	rolls	c8245d318b3f2b78171997681b3a4e85	0	1
<input type="checkbox"/>	21	sammy	sammy	sammy	sammy	4385695633f8c6c8ab52592092cecf04	0	1

Check All / Uncheck All With selected:

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

Step 3 - Edit their details

Field	Type	Function	Null	Value
userid	int(25)			23
first_name	varchar(25)			test
last_name	varchar(25)			subject
email_address	varchar(100)			test@test.co.uk
username	varchar(25)			test
PASSWORD	varchar(50)			098f6bcd4621d373cade4e832627b4f6
user_level	enum	--		0
activated	enum	--		<input checked="" type="radio"/> 0 <input type="radio"/> 1

Change level of access

Change from 0 to 1 to authorize user

There are 3 user levels on the system. The default level 0 will have the most basic access and will only have permission to view records on the system. Level 1 access allows users to add and update papers as well as viewing them. Finally the highest access level 2 allows all the previous privileges and also allows users to delete records from the database.

	userid	first_name	last_name	email_address	username	PASSWORD	user_level	activated
	23	test	subject	test@test.co.uk	test	098f6bcd4621d373cade4e832627b4f6	2	1
	17	sam	sam	sam	sam	332532dcfaa1cbf61e2a266bd723612c	1	1
	22	Royce	Neagle	royce@comp.leeds.ac.uk	rolls	c8245d318b3f2b78171997681b3a4e85	0	1
	21	sammy	sammy	sammy	sammy	4385695633f8c6c8ab52592092cecf04	0	1

Check All / Uncheck All With selected:

This illustrates how the test subjects details have now been altered to allow them to use the system with full access (level 2)

3.2 – Deleting Users

Click this tab to delete a user from the system.

Show : 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Sort by key: None Go

	userid	first_name	last_name	email_address	username	PASSWORD	user_level	activated
<input type="checkbox"/>	23	test	subject	test@test.co.uk	test	098f6bcd4621d373cade4e832627b4f6	0	0
<input type="checkbox"/>	17	sam	sam	sam	sam	332532dcfaa1cbf61e2a266bd723612c	1	1
<input type="checkbox"/>	22	Royce	Neagle	royce@comp.leeds.ac.uk	rolls	c8245d318b3f2b78171997681b3a4e85	0	1
<input type="checkbox"/>	21	sammy	sammy	sammy	sammy	4385695633f8c6c8ab52592092cecf04	0	1

Check All / Uncheck All With selected:

Show : 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

4 - Managing the Database

The system administrator should take backups of the system after major new additions have been made. This section describes the process involved in backing up and then restoring records.

4.1 - Creating Back-up's for Database Tables

phpMyAdmin can create a back up for the whole database in one go however for databases with a large amount of records it is safer to back-up tables separately.

- 1- Select a table to back up and export it.

Click tab to
export table

The screenshot shows the phpMyAdmin interface for a database named 'carney' on a 'localhost' server. The table 'PAPER1' is selected. The 'Export' tab is highlighted, indicated by a red arrow and the text 'Click tab to export table'. The interface includes a toolbar with buttons for Structure, Browse, SQL, Search, Insert, Export, Operations, Empty, and Drop. Below the toolbar is a table listing the fields of 'PAPER1' with their types, attributes, and actions.

Field	Type	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> PaperId	int(5)		No		auto_increment	
<input type="checkbox"/> Title	text		Yes	NULL		
<input type="checkbox"/> Issue	varchar(100)		Yes	NULL		
<input type="checkbox"/> YearPublished	varchar(20)		Yes	NULL		
<input type="checkbox"/> Summary	text		Yes	NULL		
<input type="checkbox"/> Comment	text		Yes	NULL		
<input type="checkbox"/> RF	text		Yes	NULL		
<input type="checkbox"/> DateRead	varchar(10)		Yes	NULL		
<input type="checkbox"/> Location	varchar(50)		Yes	NULL		
<input type="checkbox"/> Quality	char(2)		Yes	NULL		

At the bottom, there are links for 'Check All' and 'Uncheck All', and a section for 'With selected:' containing icons for various actions.

Step 2 – Save back up. Enter a location to save the file and choose a file name to save it under. If no file compression is selected it will output the commands to the screen. Choose the “gzipped” compression and click save.

View dump (schema) of table

Export

- ☒ SQL
- ☐ LaTeX
- ☐ CSV for MS Excel
- ☐ CSV
- ☐ XML

SQL options

Add custom comment into header (\n splits lines):

☐ Enclose export in a transaction
☐ Disable foreign key checks

☒ **Structure:**

- ☐ Add DROP TABLE
- ☐ Add IF NOT EXISTS
- ☒ Add AUTO_INCREMENT value
- ☒ Enclose table and field names with backquotes

Add into comments

☐ Creation/Update/Check dates

☒ **Data:**

- ☐ Complete inserts
- ☐ Extended inserts
- ☐ Use delayed inserts
- ☐ Use ignore inserts
- ☒ Use hexadecimal for binary fields

Export type:

Dump row(s) starting at record # .

☐ **Save as file**

File name template: (☒ remember template)*

Compression

☒ None ☐ "zipped" ☐ "gzipped" ☐ "bzipped"

Select this option

4.2 Restoring Database Tables

Step 1 – Make sure not database tables are selected and click the SQL tab to insert tables.

Click this tab to get
the insert page

The screenshot shows a database management interface. At the top, it says 'Server: localhost' and 'Database: carney'. Below this is a row of tabs: 'Structure', 'SQL', 'Export', 'Search', 'Query', and 'Operations'. The 'SQL' tab is selected and highlighted in red. A red arrow points from the text 'Click this tab to get the insert page' to the 'SQL' tab. Below the tabs is a section titled 'Run SQL query/queries on database carney:'. It contains a large text area for entering queries. Below the text area is a checkbox labeled 'Show this query here again' and a 'Go' button. Below this is a section titled 'Or' followed by 'Location of the textfile:'. It contains a text input field, a 'Browse...' button, and '(Max: 2,048KB)'. Below this is a 'Compression:' section with four radio buttons: 'Autodetect' (selected), 'None', 'gzipped', and 'bzipped'. At the bottom right of this section is another 'Go' button. A red arrow points from the text 'Once you have located the backup and selected the compression selected 'Go' to insert it into the database.' to this 'Go' button.

Once you have located the backup and selected the compression selected 'Go' to insert it into the database.

5.1 File list – Including descriptions

The table lists all the PHP files on the system and where they are located. A description of what each file does on the system is given. Finally details are given on how these files link to the other files in the system so that administrators will know that changing this file will affect these related files.

File Name	Location	Description of file	Related files
config.php	Home directory (carney)	Holds the database connections out of PHP files that are stored in the directory viewable from the web. All files call the config file. Also controls the details for sessions to redirect users to login if sessions are not created.	All files.
home1.php	../public_html	The home page for the system. Users can search the documents from this page. It also provides links for adding papers, an advanced search and the tutorial for the system.	1_result.php 1_addPaper.php 1_advanced.php example.html
1_result.php	../public_html	Displays the results of the search. Also holds the SQL commands to get the details of authors, keywords, or source details from the lists provided.	1_fullDetails.php home1.php 1_advanced.php
3_test.php	../public_html	Contains the html code for the forms for adding/updating papers. Also provides confirmation of paper details before details can be deleted from the database.	1_delete.php 1_addPaper.php home1.php 1_advanced.php
1_fullDetails.php	../public_html	Displays the all of the details of a chosen record.	3_test.php home1.php 1_advanced.php
1_delete.php	../public_html	Contains the SQL commands to delete a paper from the database.	home1.php 1_advanced.php
1_addPaper.php	../public_html	Contains the SQL commands to insert a new paper into the database.	home1.php 1_advanced.php
1_update.php	../public_html	Contains the SQL commands to update a paper in the database	home1.php 1_advanced.php
example.html	../public_html	Contains tutorials for using the system	author.php keyword.php source.php
author.php	../public_html	Lists the authors available on the system	
keyword.php	../public_html	Lists the keywords available on the system	
source.php	../public_html	Lists the source details for papers on the system	
1_advanced.php	../public_html	Provides the html for the advanced search options.	1_result.php 1_addPaper.php home1.php

File Name	Location	Description of file	Related files
			example.html
1_login_form.php	../public_html	Allows user to login or be redirected to registration form	1_checkuser.php 1_join_form.php
1_checkuser.php	../public_html	SQL statements to validate a user's information.	1_result.php 1_addPaper.php 1_advanced.php example.html
1_join_form.php	../public_html	Registration form for new users.	1_register.php
1_register.php	../public_html	Inserts registration details into MySQL.	1_login_form.php

5.2 Parser Description

The parser is contained within the home directory of the writer. The parser was written in Java, it works by pointing it at the original source file and getting it to process this document and output the selected fields. Further descriptions of each parser can be found in the documentation of the code.