

2003

Database development and intranet based image included database management system for ballistic firearm identification system

Toni C. Jiao
Edith Cowan University

Follow this and additional works at: <https://ro.ecu.edu.au/theses>



Part of the [Data Storage Systems Commons](#)

Recommended Citation

Jiao, T. C. (2003). *Database development and intranet based image included database management system for ballistic firearm identification system*. <https://ro.ecu.edu.au/theses/1551>

This Thesis is posted at Research Online.
<https://ro.ecu.edu.au/theses/1551>

Edith Cowan University

Copyright Warning

You may print or download ONE copy of this document for the purpose of your own research or study.

The University does not authorize you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site.

You are reminded of the following:

- Copyright owners are entitled to take legal action against persons who infringe their copyright.
- A reproduction of material that is protected by copyright may be a copyright infringement. Where the reproduction of such material is done without attribution of authorship, with false attribution of authorship or the authorship is treated in a derogatory manner, this may be a breach of the author's moral rights contained in Part IX of the Copyright Act 1968 (Cth).
- Courts have the power to impose a wide range of civil and criminal sanctions for infringement of copyright, infringement of moral rights and other offences under the Copyright Act 1968 (Cth). Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.

USE OF THESIS

The Use of Thesis statement is not included in this version of the thesis.

PUBLICATIONS

Chase, Chaoyingjiao Toni, Li, D. "Web Based Image Database System for Ballistic Firearm Identification", Proceedings of 2001 International Conferences (F) on Info-tech and Info-net, Beijing, China, October 29-November 1, 2001.

Chase, Chaoyingjiao Toni, Li, D. "Intranet Ballistic Image Database", Proceedings of 8th International Conference on Neural Information Processing, Shanghai, China, 14 - 18 November, 2001.

D. Li, C. Jiao, "Performing Firearm Identification Ballistic Database Operation Based on An Intranet", Proceedings of 2002 International Symposium on Distributed Computing and Applications to Business, Engineering and Science, Wuxi, Jiangsu, China, 16 - 20 December 2002

T.C. Jiao, D. Li, "Intranet Ballistic Image Database Optimization", Proceedings of The First International Conference on Optimization Methods and Software, Hangzhou, China 15 - 18 December, 2002

D. Li, T.C. Jiao, "Using SQL Server 2000 for Intranet Based Image Included Database Management System", Special Issue of Annals of Operations Research 2002, The International Conference on Operations Research for Development, Chennai, India, 27 - 30 December, 2002

**Database Development And Intranet Based Image
Included Database Management System For Ballistic
Firearm Identification System**

By

Toni Chaoying Jiao

**A dissertation submitted in partial fulfilment
of the requirements for the Award of**

Master of Science

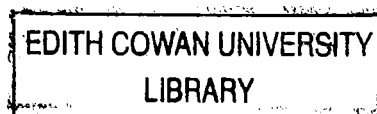
at the

School of Computer and Information Science
Faculty Of Communications, Health And Science
Edith Cowan University
2 Bradford Street, Mount Lawley 6050
Perth, Western Australia

cjiao@student.ecu.edu.au

Supervisor:
Dong Li

6 March, 2003



ABSTRACT

The process of imaging, collecting and searching a cartridge case to identifying its suspected firearm is a time consuming procedure.

Within this study, a cartridge case identification database management system in an Intranet environment is designed and implemented, thus enabling firearm examiners from different forensic laboratories to engage firearm identification without the constraints of time and location. Specifically, the study investigates appropriate database management system for image involved and Intranet secured ballistics firearm identification database.

The results demonstrated that a computerized firearm identification system could be implemented in Intranet with a secure, scalable, performable Intranet database management system.

DECLARATION

I certify that this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any institution of higher education; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person except where due reference is made in the text.

Signature:

Date: 14 March 2003

ACKNOWLEDGEMENTS

I wish to acknowledge my supervisor, guidance, and trust of Dr Dong Li, without your guidance, courage and support I would not be able to complete the study.

I would like to thank Mike Collins for the assistants in preparing the proposal, Dr Judith Rochecouste for the tuition in English writing, Chunnong Zhao for setting up the tools, and Joseph Brodalka, Geoff Lourens, Mark Leslie and all other computer support staff for providing me a workable computer system. I would also like to thank to Professor Tony Watson, Professor Linda Kristjanson and Assoc/ Prof. Wojciech Kuczborski for the understanding and supports.

As a fulltime employee and part-time student, I need to thank Western Power, especially my work supervisor Brian Jones, for the great understanding and support.

This work is dedicated to my parents and my family, especially for my daughter Alison, the most understanding child a parent could respect.

TABLE OF CONTENTS

USE OF THESIS	i
PUBLICATIONS	ii
THESIS TITLE.....	iii
ABSTRACT	iv
DECLARATION.....	v
ACKNOWLEDGEMENTS.....	vi
TABLE OF CONTENTS	vii
TABLE OF FIGURES.....	x
LIST OF TABLES.....	xii
1 INTRODUCTION	1
1.1 Manual Firearm Identification.....	1
1.2 Computerized Firearm Identification – an Alternative.....	5
1.3 The Need For Intranet Firearm Identification Systems	7
2 RESEARCH FRAMEWORK	13
2.1 Background.....	13
2.1.1 Cartridge Case Evidence Imaging	15
2.1.2 Data Collection and Storage	16
2.1.3 Searching Matches.....	19
2.1.4 Coordination	20
2.1.5 Achieving Firearm Identification DBMS with Intranet	21
2.2 Significance of the Project.....	22
2.2.1 Time Spent.....	23
2.2.2 Encourage A More Productive Outcome.....	23
2.3 Purpose of the Study.....	24
2.4 Research Questions To Be Answered	25
2.5 Variables of the Study	25
2.5.1 Controlled Variables.....	26
2.5.2 Independent Variable.....	26
2.5.3 Dependent Variable	27
2.6 Hypothesis	27
2.7 Assumptions and Limitations of the Study.....	27
2.7.1 Small Testing Database	27
2.7.2 Database Tool and Programming Language	28
2.8 Summary.....	30
3 REVIEW OF RELEVANT LITERATURE.....	32
3.1 General Literature.....	32
3.1.1 Description of Ballistics Firearm Identification DBMS.....	32
3.1.2 Database Management System.....	34
3.1.3 Relational Database Development and ORDBMS	36
3.1.4 Object Oriented Database Development	38

3.1.5	Conversion between the RDBMS and ORDBMS	41
3.1.6	Current Trends	42
3.2	Other Literature of Significance to This Study	49
3.2.1	Perceived Advantages of OO/OR over 'Relational' Model	49
3.2.2	Complex Data Handling	51
3.2.3	Multimedia Database Management	52
3.2.4	Database Support for Image Databases	58
3.2.5	Intranet/Web and /Client/Server Mechanism Support.....	59
3.3	Specific Studies Similar to the Current Study	63
3.3.1	Current Limitation of Computerized BFIS.....	64
3.3.2	Advantage of Computerized BFIS	73
3.4	Similar Study	74
3.5	Summary.....	80
4	RESEARCH DESIGN.....	82
4.1	The Issues of Intranet and Database	82
4.2	Theoretical Framework of an IFCIS - CartridgeLink.....	85
4.3	Functions Beyond Scope Of Current Study	92
4.4	System Requirements	93
4.4.1	Web Access	93
4.4.2	Intranet Access	94
4.5	User Interfaces Design.....	96
4.5.1	Logging and Registration Interface	97
4.5.2	Register a New Cartridge Case.....	101
4.5.3	Search Cartridge Case	103
4.6	Database Design	108
4.6.1	Data Model	108
4.6.2	Table List.....	109
4.6.3	Column List (Data Dictionary).....	110
4.6.4	Keys	111
4.6.5	Foreign Keys (References)	112
4.7	Summary.....	113
5	IMPLEMENTATION AND RESULTS	114
5.1	Selection of DBMS.....	114
5.2	Selection of Programming Language	115
5.3	Implementation of the Study	115
5.3.1	MSS2000 Enterprise Manager.....	117
5.3.2	ASP and Database Objects	124
5.4	Testing Output and Results.....	125
5.4.1	Logging Method	125
5.4.2	Uploading Method.....	137
5.4.3	Search Method.....	142
5.5	Performance Tests	144
5.6	Basic Capacity Scalability Testing.....	146
5.7	Summary.....	148
6	CONCLUSION AND FUTURE WORK.....	149

TABLE OF CONTENTS

6.1	Question 1.....	149
6.1.1	Question 1.1.....	149
6.1.2	Question 1.2.....	150
6.1.3	Question 1.3.....	150
6.1.4	Question 1.4.....	150
6.1.5	Question 1.5.....	151
6.1.6	Question 1.6.....	151
6.2	Question 2.....	151
6.3	Question 3.....	153
6.4	Question 4.....	154
6.5	Limitation	155
6.6	Benefits.....	156
6.6.1	Information Availability	156
6.6.2	Save of Time.....	156
6.6.3	Reduce Redundant Data	157
6.6.4	Security.....	157
6.7	Conclusion.....	158
6.8	Future Work.....	159
REFERENCES		162
APPENDIX A: A GLOSSARY OF TERMS USED IN THIS DOCUMENT.....		167
APPENDIX B: ASP PAGES AND SOURCE CODE		169
	Logging.asp	169
	LoggingVerify.asp.....	171
	WelcomeBack.asp	172
	Registration.asp	173
	RegComfirm.asp.....	177
	MainMenu.inc.....	178
	ListCartridge.asp.....	180
	ShowImage.asp.....	182
	Insert.htm.....	183
	ImgLoad.asp	187
	SearchCart.asp	190
	Connect.inc	194
	DisConnect.inc	194
	AuthenticationCheck.inc	194
	ProductionErrorHandler.inc	194
APPENDIX C: STORED PROCEDURES		197

TABLE OF FIGURES

FIGURE 2.1 BEFORE THE METADATA DATABASE IS MOVED TO ANOTHER MACHINE18

FIGURE 2.2 AFTER THE METADATA DATABASE IS MOVED TO ANOTHER MACHINE ..19

FIGURE 3.1 SQL SERVER 2000 ERD FOR TABLES IN A BALLISTIC DATABASE37

FIGURE 3.2 A SAMPLE OF AN OQL STATEMENT.....39

FIGURE 3.3 MSS2000 SECURITY ARCHITECTURE70

FIGURE 3.4 SAMPLE OF FIREBALL – USER MENU76

FIGURE 3.5 PORTION OF FIREBALL CARTRIDGE HEADSTAMP77

FIGURE 4.1 ISOLATED BFISS83

FIGURE 4.2 ON-LINE BFIS84

FIGURE 4.3 CONDUCTING CARTRIDGE IDENTIFICATION IN CURRENT SYSTEM86

FIGURE 4.4 CONDUCTING A CARTRIDGE IDENTIFICATION IN INTRANET SYSTEM.....90

FIGURE 4.5 MICROSOFT INTERNET INFORMATION SERVICE 5.0 (IIS5.0)95

FIGURE 4.6 OPEN INTRANET PAGE.....96

FIGURE 4.7 FUNCTIONAL FLOW CHART.....97

FIGURE 4.8 LOGGING INTO THE CARTRIDGE LINK98

FIGURE 4.9 NEW USER REGISTRATION99

FIGURE 4.10 MENU FOR THE OPTIONS100

FIGURE 4.11 REGISTER A CARTRIDGE CASE INTO THE DATABASE102

FIGURE 4.12 LIST OF EXISTING CARTRIDGES104

FIGURE 4.13 EXTENDED IMAGE INCLUDED INTERFACE.....106

FIGURE 4.14 IMAGE SELECTION BOX107

FIGURE 4.15 DATA MODEL OF CARTRIDGE LINK.....109

FIGURE 4.16 CARTRIDGE LINK DBMS TABLE LIST110

FIGURE 4.17 CARTRIDGE LINK DBMS TABLE COLUMN LIST.....111

FIGURE 4.18 CARTRIDGE LINK DBMS TABLES PRIMARY KEYS112

FIGURE 4.19 CARTRIDGE LINK DBMS TABLES FOREIGN KEYS112

FIGURE 5.1 INTRANET CARTRIDGE LINK’S DBMS *CARTRIDGE LINK*.....117

FIGURE 5.2 MSS2000 DATABASE CARTRIDGE LINK’S DIAGRAM119

FIGURE 5.3 STORED PROCEDURES PAN LISTS THE STORED PROCEDURES PROGRAMS122

FIGURE 5.4 USER LIST PAN AND USER PROPERTIES WINDOW123

FIGURE 5.5 USER LOGGING PROCESS FROM A WINDOWS NT TO THE CARTRIDGE LINK127

FIGURE 5.6 EXTRA LOGGING LAYER REQUESTED BY MIX AUTHENTIC MODE129

FIGURE 5.7 LOGGING ALGORITHMS.....130

FIGURE 5.8 REGISTRATION ALGORITHMS.....131

FIGURE 5.9 SHOWS AN ADMINISTRATOR’S LOGGING.....133

FIGURE 5.10 CONFIRMATION TO ADMINISTRATOR’S LOGGING WITH MAIN MENU ...134

FIGURE 5.11 REJECTED LOGGING.....135

FIGURE 5.12 NEW USER REGISTRATION136

FIGURE 5.13 USERACCOUNT TABLE CONTAINS137

FIGURE 5.14 UPLOADCARTRIDGE ALGORITHM.....138

TABLE OF FIGURES

FIGURE 5.15 UPLOADING IMAGE INTO DATABASE 141

FIGURE 5.16 ALGORITHM OF SEARCH METHOD 143

FIGURE 5.17 RESULT OF SEARCH AND SEARCH RESULTS 144

FIGURE 5.18 TRANSACTION SPEED BASED ON NUMBER OF CPUs 147

FIGURE 5.19 TRANSACTION SPEED BASED ON GB OF MEMORY 148

LIST OF TABLES

TABLE 1 PERCENTAGE OF FIREARM ARMED CRIMES IN 199822

TABLE 2 LIST OF MSS2000 MAJOR COMPONENTS RELEVANT TO THE STUDY30

TABLE 3 CRIME TYPE BY AGE OF POSSESSOR DURING 1ST JAN TO 31ST DEC 200033

TABLE 4 DIFFERENCES BETWEEN FIREBALL AND PROPOSED INTRANET BFIS 78

TABLE 5 COMPARISON TO THE DATABASES TOOL USED BY FIREBALL80

TABLE 6 CONFIGURATIONS SET UP FOR THE SYSTEM.94

TABLE 7 COMPONENTS OF HOME96

TABLE 8 EXPLANATION OF LOGGING PAGE.....98

TABLE 9 EXPLANATION OF USER REGISTRATION PAGE99

TABLE 10 EXPLANATION OF THE MAIN MENU PAGE..... 101

TABLE 11 EXPLANATION OF CARTRIDGE UPLOADING PAGE..... 103

TABLE 12 EXPLANATION OF LISTING EXISTING CARTRIDGES PAGE..... 105

TABLE 13 EXPLANATION OF EXTENDED CARTRIDGE CLASSIFICATIONS AND IMAGE
PAGE 106

TABLE 14 EXPLANATION OF EXTENDED CARTRIDGE CLASSIFICATIONS AND IMAGE
PAGE 108

TABLE 15 EXPLANATION OF DBMS *CARTRIDGELINK* COMPONENTS 118

TABLE 16 USERS AND THE PRIVILEGES 124

TABLE 17 DATABASE OBJECTS..... 125

TABLE 18 TEST CARTRIDGE DATA 140

TABLE 19 PERFORMANCES OF THE METHODS 146

1 INTRODUCTION

Before the invention of the computer, firearm identifications were conducted manually. Each evident submitted to the crime laboratory firearm unit was manually compared either through firearm evidence, projectiles or fired cartridge. These forms of firearm identification were important to allow examiners to specify the firearm used in a crime, as well as to discover the link between the crimes. In this chapter, an overview of manual firearm identification by comparing the marks on the cartridge case breech face is presented. Then the need for a new medium to be used as a tool to conduct an on-going collection, maintenance and comparison of cartridge cases is discussed. Finally the aims of the project are described and the contents of the remaining chapters are outlined.

1.1 Manual Firearm Identification

Law enforcement systems spend large amount of time in identifying firearms when firearm crimes are investigated. There are many forms of firearm identification, one of which is comparing recovered cartridge cases with test-fired cartridge cases to find matched firearms. As Doyle (Doyle, 2002) observes, “the foremost question normally asked in most shooting investigations is whether a firearm from the scene, a gun recovered as the

result of a search warrant, or one taken from the suspect was the gun that fired the bullets and cartridge case that have been recovered.” ATF (America Bureau of Alcohol Tobacco and Firearms of Department of the Treasury - ATF, 2002) agrees with him by further suggesting that “the comparison of ballistic evidence has in the past been a tedious and time consuming process”, because of “evidence recovered at crime scenes or from recovered firearms was compared, piece by piece, to the vast inventory of recovered or test-fired projectiles and casings”. Firearm laboratories, whether large or small, identify firearm cartridges among their collections including both recovered and test-fired cartridge cases. The size of the collection and the recovered evidence scale in a laboratory increase as time pass by. The ATF points out that “bullets and cartridge casing found at one crime scene can also be compared with those found at another in order to link the crimes”. These firearm identifications can be made locally or remotely but the purpose of such activities remains producing a forensic scientific firearm identification result for law enforcement system.

Firearm identification through matching cartridge cases may take several forms. The most basic form is manually comparing cartridge cases by eye-witnessing the marks in headstamp images, as ATF comments, firearm identification requires searching each piece manually even though it dose severe stress and eyestrain on firearms examiners.

Cartridge case comparison is the most direct means of imparting a match to a firearm. Doyle (Doyle, 2002) points out that:

Every firearm has individual characteristics that are as unique to it as fingerprints are to human beings. When a firearm is fired, it transfers these characteristics – in the form of microscopic scratches and dents – to the projectiles and cartridge casings fired in it. The barrel of the firearm marks the projectile traveling through it, and the firearm's breech mechanism marks the ammunition's cartridge casing.

While cartridge cases can be classified by manufacture or marking positions, firearm examiners may feel more comfortable to exam cartridge cases images using their physical observation through a microscope because of an instant feedback is inherent in this type of firearm identification. Doyle says, "if there is a likelihood that ... two cases are connected, the actual evidence is examined by using the conventional forensic comparison microscope." Indeed ATF's gun tracing studies have shown that "many agencies are now using both crime gun tracing and ballistics identification to support firearm investigations". Celens & Demanet (Celens & Demanet, 1989) agree and note, "If there are evidence bullets and cartridge cases, we first investigate the latter" because "generally the investigation procedure for the

identification of cartridge cases is easier and less time consuming, and usually evidence cartridge cases are less deformed than evidence bullets.” This view is also shared by Geradts, Bijhold & Hermesen (Geradts, Bijhold, & Hermesen, 1999) who observe, “often the cartridge case is the most important forensic specimen in the identification of weapons, as bullets are commonly deformed by the impact”.

AFT points out the limitation of manual firearm identification system is “the necessity of searching each piece manually greatly reduced the amount of evidence that could be examined, given manpower and time constraints, and made matches less likely”. Further, they explain that there are two problems for manual systems – the first is the size of the collection that a manual system can handle and the second is the speed that a manual system can precede. Therefore, if manual firearm identifications are to be successful up front of the crimes, all participants of the law system must share the sufficient amount of information for the worst scenario that one crime is linked to another in different states to identify firearms within required time frame. This has become increasingly difficult in our hectic world. Thus, a new medium for firearm identification is needed for ongoing increased evident searching and collection.

1.2 *Computerized Firearm Identification – an Alternative*

Hamby (Hamby, 1999) submits that “The most significant advances” during the last part of nineteen century “include the tremendous growth, popularity, and relatively inexpensive cost of computers” because “its ability to fully utilize the vast potential of computers has allowed science overall, and forensic science more specifically, to take full advantage in development of several useful ‘tools’ for use within the laboratory”. Adding to this observation, Moran (Moran, 2000) suggests that database can be used to retain ballistics cartridge case information including images. Hence, computerized firearm identification system has proven to be an appropriate alternative to overcome the limitations of manpower and time constraints described earlier. According to Tartaro (Tartaro, 2000), computerized firearm identification “intended to speed the process by which ballistics technicians and criminalizes could match up evidence samples from bullets, bullet fragments and spent cartridge cases retrieved at crime sites”.

The introduction of computerized firearm identification system has eliminated both speed and size restrictions imposed on manual systems. Using these systems, examiners may collect as many data as currently available and search for a match as quick as the computer runs. As Doyle points out, the search results that narrowed down by computer systems may then be examined by using the conventional forensic comparison microscope on the

actual evidence. If there is a positive identification, the two cases are then tied together.

Apart from this, computerized firearm identification has additional benefit. In manual system, examiners normally search for the matches by eyesight. This is applicable when the size of the collection is small, but when it comes to a large scale search, especially when eyes are tired after long period of viewing images, it may not work properly as an important match may missed by mistake. More importantly, computerized system offers relational database management system that allows a imaged cartridge case to be classified with its marking details so that later it can be searched by these classifications.

According to Geradts et al (Geradts et al., 1999), small-scale search evident in manual firearm identification systems may be countered productive. He observes that examiners required to accomplish an identification task with short time constraints tend to work faster but with a lower level of quality, such an examiner may skip certain tasks in a development stage to make up for a short time frame. More importantly, he suggests that much of the firearm identification activities would be eliminated in the process, resulting in poor understanding of relationships between linked firearm armed crimes. On the other hand, he observes that when examiners are provided with a well-organized and large scale of data to accomplish a task, they normally produce a higher level of understanding of crime links, because a examiner may search

through a well managed information system to identify the firearm which are required for solving the crimes. Links between cases within a lab may be better explored since there are constant maintenance activities made to the information system; such a form of firearm identification can be achieved through desktop computerized systems.

1.3 The Need For Intranet Firearm Identification Systems

As previously stated, one of the main purposes of firearm identification that cannot be ignored is to explore the link between crimes. To achieve this, it requires the networks between laboratories, particularly when there are suspects of links between the crimes in different states.

With regard to the field of organized crime, Australian Police and intelligence authorities have observed that organized crimes in Australia are “well organized and dangerous” and “there have been many unsolved murders in Melbourne linked to the Mafia”. They add that it is this capability of organizing crimes that making linked crimes more threatening and damaging.

In the state of Alabama in America, the police uses networked computerized firearm identification systems solved 10 crimes either linking cartridges to guns or bullets to guns, which could not be identified before

(Moran, 2000). They propose that the availability of networked firearm identification is important for solving more violent crimes. They stress that interfaces with networked firearm identification “reduce the geographical barrier between states and increase the opportunities identifying firearms”

While interaction and share data are important, current computerized ballistic firearm identification systems in Australia do not facilitate their use in their best forms. Smith, Cross & Variyan (Smith, Cross, & Variyan, 1995) point out that while current computerized firearm identification system is successfully used by police departments around the country, “further research is directed towards an investigation of digital image processing and registration of the complete firearm ‘fingerprint’ and matching techniques” which should be incorporated, Li (Li & Watson, 1998a) added, in an networked computerized firearm identification system.

One of the main differences between desktop and networked computerized firearm identification systems is that desktop system is local information based and networked system is shared information based. Other database management mechanism and facilities available in advance database management systems are also lacking in current system.

In summary, manual firearm identification means that examiners are obliged to search through collected large amount evidence for suspected

match, which may be considered to be time consuming and inaccurate. Computerized firearm identification system allows more data and faster speed, but the current mechanism used in Australian laboratories are mostly isolated to each other and do not incorporate integrated information management system that are evident when identifying firearms in an Intranet system.

This study will incorporate Intranet into a computerized firearm identification system with an integrated database management system. It will establish that use of Intranet integrated information system enhances integrity, security and performance of computerized firearm identification system, by using some of the facilities normally available in database management systems, a set of tools of a database in an desktop system is not presented.

The discussion in chapter one seeks to indicate that a vulnerable point exists in the DBMSs, which may be poised to succeed the performance of a DBMS. Due to this vulnerable point, different approaches may choose, for expediency, to access data in different DBMSs. It is noted that ODBMS, when faced with similar complexity, have adopted a newer approach – OO approach alteration of Relational approach. This method to increase the complexity of design or access and so enhances the performance of data access. The study sets out to answer the question of whether the OO approach may be successfully applied, whilst using maximum available Internet programming resources, to traditional Relational approach.

Chapter two discusses the background of the study and explain the importance of applying Intranet technology to the ballistic firearm identification hence ballistic database. It will also describe variables that will impact upon the research questions. It then builds upon the foundations outlined in the study background (chapter one) and forms the basis for the detailed study design. The choice of using Intranet technology compared to desktop technology is also discussed, together with the reason for choosing the SQL Server 2000 DBMS. This design will be developed, together with its methods and verifiable outcomes.

Chapter three takes the form of a review of the relevant literature. The work of others, illustrated in the firm foundation of text books and augmented by the documented research and experiences of others in papers and articles forms the basis of guidance and justification for the approach taken and substance in this study. Specific reference is made to a set study of the comparison between DBMSs and to the FireBall system developed by Edith Cowan University. It is demonstrated that the underlying presence of an unsuitable DBMS may enable more fundamental problems such as security and impeachment mismatch. The nature of DBMS is explored, together with its application, and justification established for its adaptation to provide a solution to the problem outlined in chapter one.

Chapter four describes at a high level the design and procedure that the study will take, requirement to be met, the instruments that will be employed, and the study's limitations. More over, the findings and results of the experiments and implementation will be presented. A theoretical framework to existing ballistic database is explained, followed by a verification of the proposed system with test criteria. Several snapshots of the program interface are also illustrated.

Chapter five describes the study has been concluded in resulting the founding presented by chapter four. It details the outcomes of the experiment; tackles each research questions through relevant evidence. Further opportunity of improvement is documented.

Chapter six concludes the project. It will include a summary of the purpose of the study; initial aims; the manner in which the study's framework was derived; and the design criteria for the implementation. The study's results will be critically reviewed before finally discussing implications for current practice and future research.

Finally, following by a section where end text references for documents used to support the study, this document concludes with several appendices described below:

Appendix A consists of a glossary of terms used in the document.

Appendix B consists of a dictionary of the database

Appendix C consists of a listing of the program

2 RESEARCH FRAMEWORK

The Intranet firearm ballistic identification systems is designed to allow on-going firearm identifications to be carried out on-line with the focus on improving such activities with on-going collection and administration of ballistic data. The system should fulfil specific requirements and assumptions as required. In order to establish a framework for the design and implementation presented in later chapters, this chapter examines the background, significance and objective as well as the requirement and limitations of the Intranet firearm identification system developed by this study.

2.1 *Background*

It is known that every firearm has individual characteristics that are as unique to it as fingerprints are to human beings (J. S. Doyle, 2002). When a firearm is fired, it transfers these characteristics – in the form of microscopic scratches and dents – to the projectiles and cartridge cases fired in it. The barrel of the firearm marks the projectile traveling through it, and the firearm's breech mechanism marks the ammunition's cartridge case.

This fact creates a great opportunity for law enforcement. When bullets or cartridge cases are found as crime scene, firearms examiners can use the marks for comparison, to determine whether or not the bullets or cases were expelled from a suspect's firearm. If a firearm is recovered at the scene, a test fire of the weapon creates example bullets and cartridge cases for comparison. Bullets and cartridge cases found at one crime scene can also be filed and compared with those found at another in order to link the crimes. The prospect is important for solving organized crimes. A system of these facilities, collected information and identification processes forms a Ballistic Firearm Identification System (BFIS).

The comparison of ballistic evidence, as pointed out by Inbau (Inbau, 1999), has in the past been a tedious and time-consuming process. Evidence recovered at crime scenes or from recovered firearms was compared piece by piece, to the vast inventory of recovered or test-fired projectiles and cases. The necessity of searching each piece manually greatly reduced the amount of evidence that could be examined, given manpower and time constraints, and made matches less likely. Severe stress and eyestrain on firearms examiners slowed the process even more.

In developing the Intranet BFIS, the benefit to law enforcement that ballistic imaging database could provide is recognized. The Intranet BFIS development includes the development and maintenance of a database of

ballistics images from crime guns and its shooting components. The database contains images of carriage case or bullets recovered at crime scenes, as well as cases or bullets from test fires of recovered firearms. As new images are entered, the system searches the existing database and comparisons are made from possible matches. The purpose of these comparisons is to link ballistic evidence from crime scenes, linking one crime scene to another.

Whatever way the firearm identification database system is implemented, it is always contain four main functions as the recovered evidences needs to be imaged; the images and text data need to be collected and stored; the collected images and data need to be searched for possible matches for incoming evidence; and above activities needed to be coordinated.

The following sections explain how each of the function plays apart in providing a firearm identification system.

2.1.1 Cartridge Case Evidence Imaging

Ballistics evidence imaging may take on many forms with different devices and different mediums. Despite traditional cameras, computer aided imaging is widely used as the images can be saved in computers as image files. Mixdorf & Goldsworthy (Mixdorf & Goldsworthy, 1996) state “...

computers have been linked successfully to all imaging modalities. ... with computer-assisted imaging, it is possible to generate clear images”.

In believing ordinary imaging dose not merely involve the capsulation of a fired bullet’s characteristics, Li and A. C. Watson (Li & Watson, 1998a) take an approach in which a evident imaging is progressed with an optical microscope connecting to a desktop PC in which software are running to capture the image from the microscope, enhance the image and saved in to a file. Additionally, this explains a need for a facility that collects and stores these images, which is explained in next section.

2.1.2 Data Collection and Storage

Data collection, including both text information and images, is perhaps the most important activities for a BFIS. According to Handerson (Handerson, 1996) data collection meet the information needs as “sources have been well established and historical data kept in the system”. United Nation (United Nation, 1997) stress the importance of data collection when they state that “a better coordinating the data collection that is necessary for a more complete understanding of the issues affecting firearm regulation”. Therefore, whether people are working as scientific ballistic evident provider or a crime investigator, whether they work on a case happened in Queens land or WA, they can compare recovered evidence with the historical collections

effectively to find suspected links between the crimes. The collection may be achieved through manual means such as a filing system in which papers are divided by cases. The information can then be exchanged by mailing system if it is required to across the boundaries between areas.

Collected data may be stored in various ways. They may be filed into folders case-by-case; they may be extracted into different categories and stored together with other cases information in the same category; or they may be stored in computers file systems or database systems. Songini (Songini, 2002) refers data, collection, storage and database as a facility that “a datum is a raw piece of information that's capable of being moved and stored. ... A database is a collection or aggregation of such data, along with information on how pieces of data relate to one another.”

As Li (Li & Chase, 2001) points out that “images are the most important data in Intranet ballistic database.” They are cartridge images, cartridge’s head stamp images and projectile images. These images often exist as image files in computer filing system. Besser (Besser, 2002) believes that image data in file form dose not merely involve the integrity rules of information because the use of records, tables, queries, reference and securities are also significant for an application “to construct relationship from both the texture information and their individual images.” He says that there is a model of image-involved-application consisting of a set image files and

another database for metadata, however he says that as a result, there are usually problems of missing information, viewing images, linking inter-relations, custodial and translation. An example is given in figure 2.1 and 2.2 that demonstrated the problem of maintaining the image files and their metadata database. Additionally, another explains a need for a different quality of interaction, one that helps to establish a result from a query. This is the function of querying, which is explained in the next section.

Cartridge Case Database

HeadStampId : 21

Markings : 12-SOFIM


CartridgeName : 12 Gauge Shotgun

CalibreName : 72 Calibre

Image Location:

C:\Firearm\CartridgeCase\Headstamp21.bmp

Record 21 of 260



Red Ribbed plastic case walls with six star crimp. "ALIMOW 4 12-70 gauge load" printed on side wall.

<< First

< Prev

Next >

Last >>

Figure 2.1 Before The Metadata Database Is Moved To Another Machine

Cartridge Case Database

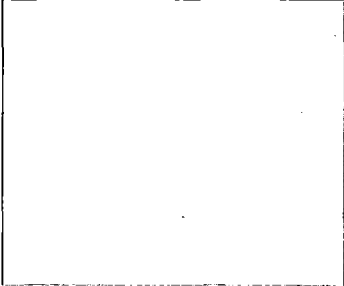
Record 21 of 260	
<p>HeadStampId : 21 Markings : 12-SOFIM CartridgeName : 12 Gauge Shotgun CalibreName : 72 Calibre</p> <p><u>Image Location:</u> <u>C:\Firearm\Cartridge Case\Headstamp21.bmp</u></p>	 <p>Red Ribbed plastic case walls with six star crimp. "ALIMOW 4 12-70 gauge load" printed on side wall.</p>
<div> << First < Prev Next > Last >> </div>	

Figure 2.2 After The Metadata Database Is Moved To Anther Machine

2.1.3 Searching Matches

Hamby (Hamby, 1999) asserts that the collection of ballistics data “provides the examiner with the opportunity to search for possible identifications on fired evidence bullets and cartridge cases in the laboratory”.

In a ballistic laboratory, a firearm examiner normally conducts numerous searches so as to find the suspected matches. When a search is made to a ballistics database, the DBMS “enables the searching by writing the queries” and “ a query transformations performed on the searchers' behalf”. Shapiro (Shapiro, 2001) has further described a query as "a question or an

enquiry into the data in the DBMS". These queries typically made with the specified attributes from recovered evidences. For instance, a question of "is there a category *17Calibre* cartridge manufactured by *Sake* in current collection?" transfers specifications *17Calibre* and *Sake* into a query to search a set of matched cartridges. The examiners reports latest findings and this may result further searching. It is observed the queries and results are not really collaborating with one another when images are saved as files, instead, they are merely collecting text information because of the complexity of the systems.

Thus, the key to querying is not merely to collect text data but to find the matched images provided by the system. It is the images matching part that should be achieved.

2.1.4 Coordination

The coordination is required for the BFIS when people are work together as a network from different location and different department. So far, we know that people have to interact to collect and search in order to share information as well as finding matched cartridges. It is notice that much of these interactions occur in a very insufficient nature when such new work is coordinated by mail or Email systems.

2.1.5 Achieving Firearm Identification DBMS with Intranet

To embrace the collection and searching, a new concept was born to take advantage of the Internet, namely, the Intranet – a term used to describe implementation of Internet technologies within a corporate organization instead of external connection to achieve cooperation in an internal organization.

Australian Personal Computer (1996) states that although traditional network solutions can streamline an organization's communications and enhance collaboration, they also have reputation for complexity that can result in high startup and running costs. Moreover, the editor of Australian Personal Computer (1996, 70) suggests "Intranets, which take the open technology of the Internet and apply it to internal company networks, offer a cheap and relatively simple alternative. "

From the definition of Intranet Firearm Identification Database and its potential advantages discussed in section 1.2, there is an understanding that Intranet Firearm Identification database offers a new way for people to share and interact without regard for time and location. Therefore, if Intranet Firearm Identification database is to be implemented within an Intranet environment, it will allow people within the same law enforcement network to hold an on going querying without the knowledge of the wider community

residing outside the organization. This could bring about numerous benefits not evident in current desktop identification database systems. This benefit is discussed in the next section.

2.2 Significance of the Project

As firearm armed crimes are increasingly involved in armed crimes, effective firearm identification becomes significant in law system. Table 1 shows armed robbery by location, type of weapon used and type of victim in Australia (Mouzos & Carcach, 2001).

	Residential Location		Street/ footpath		Open spaces/park s		Banks/ Credit union/ building societies		chemists/ pharmacies		Service stations		Other retail locations	
	Ind	Org	Ind	Org	Ind	Org	Ind	Org	Ind	Org	Ind	Org	Ind	Org
Firearm	24.4	25.4	11.6	10.9	11.3	4.4	71.8	52.9	20	22.9	14.8	17.3	27.9	19.9
Knife/sharp instrument	17.3	34.2	42.8	49.4	16.9	54.3	9	16.1	68	35.4	60.6	43.6	19	40.3
Syringe	0.2	1.3	2.5	2.7	0	1.9	1.1	1.4	4	13.7	6.1	6.1	2	5.7
Blunt instrument	5.7	11.7	3.9	3.5	1.9	6.6	0.5	2.1	0	3.9	7.4	5.3	2.6	7
Other weapon	52.4	17.7	38.9	15.6	69.9	16.6	17.6	11.4	8	8.3	9.9	11.3	48.2	17.7
Imitation weapon/ threats	0	9.7	0.3	17.9	0	16.2	0	16.1	0	15.8	1.2	16.4	0.3	9.4

Table 1 Percentage of Firearm Armed Crimes in 1998

2.2.1 Time Spent

A great deal of time is spent collecting, searching, and exchanging ballistics information. Much information is constrained either by being geographically distant or by limitations of organization. In the current world, increasing linked (organized) crimes require greater effort for solving and judging the crimes.

2.2.2 Encourage A More Productive Outcome

To encourage a more productive outcome from identifying and matching firearms, a common platform is needed that is readily available to everyone involved for solving the crime. Internet technology offers such flexibility. It is not only far-reaching but also easily accessible. That means that the Intranet contains not only the same services evident in Internet technology but has the added function of being able to restrict the access of that information for internal use only. Therefore, the Internet is a suitable medium for an firearm identification database system to be built upon.

Having established a suitable technology, a tool to use the technology is required. Such a tool is the SQL Server 2000 (Microsoft, 2000b) and its competitive programming components. Developed by Microsoft, it is designed to be objective, administrative, secure, and client/server architecture across

different platforms. More importantly, SQL Server 2000 (SQL2000) can be programmed to work with Internet Web browsers or as a stand-alone database.

SQL Server is a fairly new enterprise DBMS compare with some other mainstream database tools such as DB2 at the time of this writing, and has mostly been implemented for smaller scale tasks and stand-alone databases. A significant part of this project is also to determine if SQL Server 2000 can be used to develop more complex applications. This study subjects the SQL Server 2000 database management tool to the rigors of developing a complex application.

2.3 *Purpose of the Study*

This study investigates the possibility of development an Intranet Firearm Cartridge Identification System that uses a selected DBMS to conduct on-going data collection, date inquiry and data administration. By developing such a information system in an Intranet environment using a set of tools provided by Microsoft, a common medium is available for firearm examiners to contribute their recovery of the ballistic information without area and time constraints, allowing each link of the law enforcement system to become more involved and more aware of the situation in firearm identification during the investigations.

2.4 *Research Questions To Be Answered*

The focus of the study is in the following area:

- What are the problems associated with intraneting firearm identification system IFCIS? OR: What are the problems associated with implementing Intranet firearm identification system?
- What are some possible logging solutions to securing an intelligent database like ballistic database?
- How do we convert image files into image data type fields? E.g. how do we upload cartridge case image files into cartridge case records?
- How can we provide sufficient performance to an Intranet based image? E.g. how can we provide IFCIS DBMS performance?

2.5 *Variables of the Study*

The variables used to gain the experimental outcomes include controlled variables, independent variable and dependent variable. Each category will be described as follow.

2.5.1 Controlled Variables

Controlled variables are data that are kept the same throughout study experiments:

- Cartridge case details;
- Cartridge case images; and
- User Accounts.

2.5.2 Independent Variable

The independent variable of the study is the firearm identification system that can be switched between following forms:

- manual;
- desktop computerized; and
- Intranet computerized.

The form of firearm identification is purposely changed and tested against to the same scenario and data provided as controlled variables.

2.5.3 Dependent Variable

The dependent variable of the study is the measurement of changed sufficiency observed because of change of the firearm identification method.

2.6 Hypothesis

An Intranet based, appropriate DBMS and programming language incorporated ballistic firearm identification system will provide scalability, security, and performance to firearm identification.

2.7 Assumptions and Limitations of the Study

Since the project deals mainly with the development of database and programs, the result is a subjective one. Some of the main drawbacks of the study include:

2.7.1 Small Testing Database

As the main purpose of this study is to prototype an Intranet based, information secured, image data retrievable and performable ballistic database

system, it will not provide details of the capacity and speed of the system.

However, a reliable testing result will be quoted as reference in chapter 5.

The project was tested on two type of ballistics images and eighteen entrances of each type, providing indicative results only.

2.7.2 Database Tool and Programming Language

The SQL Server 2000 DBMS and ASP Internet programming language is the software for the development of the proposed system. SQL2000 is employed because “it offers significantly enhanced Web features with which to construct database solutions” (Microsoft, 2001). As Shapiro (Shapiro, 2001) explain, “SQL Server 2000 is design to ... be deployed into Internet and Intranet environments ... , your programs are deal with a set of database objects – there are no data type incompatibilities across hardware and software architectures”.” Kauffman (Kauffman, 2000) emphasizes that the advantage to using SQL2000 DBMS is that “the strength of SQL Server in storing binary large objects (BLOB) data. You can access them as a field of a record without concerning their file location, file type ...”

Another important aspect of SQL2000, substantiated by Microsoft (Microsoft, 2000b), is that it supports the TCP/IP network protocol. Microsoft (2001) stresses that “SQL Server 2000 applications can be standalone or

client/server applications that ride over TCP/IP, the most popular network protocol on the planet.” Furthermore, Kauffman (1997) observes the SQL2000 “has evolved from a tool for supporting Microsoft programming languages to support OLE DB and ADO objects that forms major part of database components in ASP language.” Therefore, SQL2000 makes the task of developing a client/server system in an Intranet environment much easier with its rich set of database objects.

Nevertheless, at the commencement of this project, SQL Server 2000 was just released. Much of the other functionality of SQL Server’s later versions was not available, leading to a less efficient method for storing ballistics information. Although this did not lead to problems with the program, it did slow the system down during the experiment.

Feature	Ability
Extendable OO Components	<ul style="list-style-type: none">• User-Defined data type used by extended SQL 92• User-Defined function by extended SQL 92• XML Integration of Relational Data• Object Oriented Meta Data Services• ADO, OLE, ODBC Object for Application development
Internet Integration	<ul style="list-style-type: none">• IIS Virtual Directory Management for SQL Server• The ability to access SQL Server using HTTP• The ability to retrieve and write XML data• Enhancements to the Microsoft SQL Server 2000 OLE• DB provider (SQLOLEDB) that allow XML documents <p>to be set as command text and to return result sets as a stream</p>
BLOB / Image Management	<p>SQL Server 2000 supports a new text in row table option that specifies that small text, ntext, and image values be placed directly in the data row instead of in a separate page. This reduces the amount of space used to store small text, ntext, and image data values, and reduces the amount of disk I/O needed to process these values.</p>

Table 2 List Of MSS2000 Major Components Relevant To The Study

2.8 *Summary*

To identify firearm with recovered evidences, firearm examiners need to collect, query as well as coordinate their activities together to achieve a

common goal to match suspected firearms. Therefore, co-operation by all parties involved must be achieved.

In traditional firearm identifications where examiners analysis images pictures by pictures to match the evidences and exchange information by mails for linked cases, both size of available data and speed of searching the matches were limited by the constrain of manpower. Desktop system improved the situation but had geographical barrier. Intranet ballistic database system eliminates that constraint and enables a more prolonged and thorough matches.

While Intranet ballistics database system offers advantages, the majority of systems available today in Australia are mostly desktop-based together with insufficient database management system. The images data in particular need to be managed properly so that the relationships between text and images can been administrated in the way that a proper database should be administrated and referenced. Fail to do so could lead results in ambiguity and misunderstanding. The fundamental aim of this study is to prototype an Intranet firearm identification system which can effectively identify firearms via matching up cartridge cases over an Intranet interface and Intranet DBMS. This application is to be prototyped in a manner which is reliable in terms of the DBMS selected to equip vast amount of data including images.

3 REVIEW OF RELEVANT LITERATURE

The aim of this chapter is to discuss the usefulness and limitation of DBMSs in general. The main focus is on specific studies undertaken in this area and the different approaches taken thus far to develop different kinds of applications including computerised ballistics firearm identification database systems.

3.1 *General Literature*

3.1.1 Description of Ballistics Firearm Identification DBMS

The firearm identification laboratories exist in criminological investigations divisions and play important roles in the legal system. Table 3 shows firearm offenses are most crimes in Albuquerque of New Mexico in USA (ATF, 2002). The main mission of the laboratory, as described by Papaglannis (Papaglannis, 2000), is “the rendering of assistance to the mission of the Police and Judicial Authorities as well as to all Law Enforcement Authorities, through the rendering of scientific and technical services” which are nationally recognized, scientifically substantiated and judicially accepted. A firearm laboratory mainly acts as a productive center for the rendering of prompt and reliable ballistic information services. Therefore, firearm laboratories constitute ballistic databases.

Crime Type	Total	Juvenile (ages 17 & younger)	Youth (ages 18-24)	Adult (ages 25 & older)	Age Unknown
Number of Trace Requests					
Firearm Offenses	477	30	77	318	52
Homicide	30	1	6	14	9
Kidnapping	1	0	0	1	0
Robbery	9	1	4	1	3
Sex Crimes	3	0	0	2	1
Assaults/Threats	88	4	26	47	11
Burglary/Theft/Fraud	22	1	7	9	5
Vice Crimes	2	0	1	1	0
Narcotics	90	7	19	49	15
Other Crime Types	178	2	30	61	85
Total	900	46	170	503	181

Table 3 Crime Type by Age of Possessor During 1st Jan to 31st Dec 2000

When a recovered cartridge submitted to the laboratory, it becomes a ballistic datum representing a raw piece of information that's capable of being moved and stored. A collection of such data along with information on how pieces of data relate to one another make up a ballistic database. A ballistic database is organized into tables with records - one record per item, such as a cartridge - that are themselves divided into several fields, with each field containing information about a specific aspect or attribute of the cartridge include manufacture data, calibre, marking, composition type and heads tamp image.

In theory, a ballistic database doesn't even require a computer, but it certainly makes its use a lot more scalable and efficient, examined by Butterfield (Butterfield, 2001) with real cases. "A firearm and cartridge collection folder is certainly a database, but searching cartridge entries by classifications such as manufacturer or calibre requires flipping through each page, not to mention searching a cartridge case headstamp image similar to recovered evidence", he comments.

Computer ballistic DBMS, by using modern database technologies from vendors like Microsoft Corp., Oracle Corp. or IBM, act as the underlying vault and

retrieval technology. In addition to storing ballistics images and text data, ballistics DBMS handles security and assessment controls.

Ballistics DBMSs can be implemented by different methods. A relational DBMS stores data in tables and then joins or combines those tables across common fields. An object-oriented DBMS encapsulates both data and business logic. In between, an object-relational DBMS allows users to create their own data types and provides objects for programming facilities. We will review each of method and compare the differences.

3.1.2 Database Management System

Date (Date, 1993) describes Database Management Systems (DBMSs) as computerised systems for maintaining information for future availability. Contemporary DBMSs originate from the system proposed by Codd (Codd, 1990) which incorporated tables of data stored so as to minimise or eliminate redundancies and anomalies occurring when deleting/editing the data or inserting further data. Codd's proposition also included the concept of relations existing between the tables to facilitate navigation of the contained data. Codd's precision of data draws from the mathematical definition of a relation that he regards, simply, as a set with special properties. These are, namely, that all of its elements are tuples of similar type; and that the set is unordered, thereby removing any dependency upon ordering of the data when accessing the data. When stored in table form, Codd suggests that a relation, R , when conceived as a table (stored in rows and columns), has the following properties:

each row represents a tuple of R;
the ordering of rows is immaterial; and
all rows are distinct from one another in content.

Furthermore, Codd introduces the naming of relations and, particularly, of columns for the following reasons:

the column name implies the meaning of the column of
stored data;

naming removes the need for a sense of "nextness", or
relative positioning of the columns; and

naming distinguishes each column from its underlying
domain, i.e. indicating that a column is, in fact, a particular
use of a domain.

Most current systems are based on two particular data models, delivering two common DBMSs, namely the relational model and the object-oriented model. These two models form the basis for Relational Database Management System (RDBMS) and Object-Oriented Database Management System (OODBMS) respectively (Ramakrishnan, 1998). Object extended RDBMS (ORDBMS), a module that starts from RDBMS and includes many object features, is often mentioned as part of

RDBMS. Both relational and object models will be discussed further in next two subsections.

3.1.3 Relational Database Development and ORDBMS

Codd's proposed system provides us with the framework of a Relational Model (RM). The principal aim of relational database design is to normalise an input relation schema together with a set of data dependencies into an appropriate normal form. The motivation behind the various normal forms is to remove problems caused by update anomalies and redundancy problems. As a result of normalisation we obtain a collection of relation schemas. A RDBMS creates tables in the application database and maintains the data in the schema data dictionary by executing the SQL statements in the scripts that the schema generates.

There is, however, dispute over whether the relational model as originally described has ever been implemented fully (Kroenke, 1992). Furthermore, it is claimed that with the implementation of the formal model and expansion of one central idea, the domain, there would be a reduced need to abandon relational databases.

The commercial implementations of relational databases that followed Codd's principles changed the nature of management of stored data at the time and, as a consequence, RDBMSs have dominated the industry for more than two decades.

When using the relational methodology, the data flow and entity relationship diagrams (ERD) are produced from the design phase. As described by Anstey

(Anstey, 1999), “these tools, particularly the ERD, were adequate in the relational model because entity types were simple and the set of relation types correlated to the mapping of normalised data structures.” Many major database vendors together with their database system offer ERD. Figure 3.1 shows an ERD of the entities cartridges’ head stamps, manufacture, calibre and composition type in database FireBall instituted in MS Access (Li & Watson, 1998a).

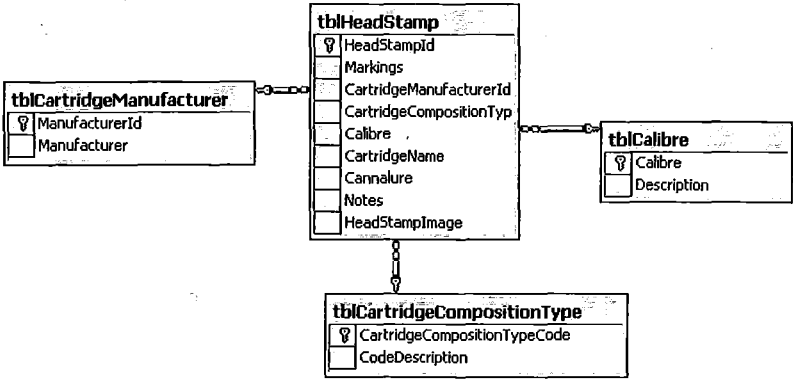


Figure 3.1 SQL Server 2000 ERD for Tables in a Ballistic Database

Quinlan (Quinlan, 2001) has analysed three major RDBMSs (Oracle9I, IBM UDB and MSS2000) and found some advantages of MSS2000 over others. In order to become a high-end database server in the way that other major vendors have in the Unix and mainframe world, Microsoft has come a long way for its performance, scale, and deliver high availability.

To deliver higher performance and scalability, MSS2000 scales to 32 processors in a symmetric multiprocessor environment and also operates as a shared-nothing clustering implementation. MSS2000 has a better benchmark for e-business systems in recent tests.

MSS2000 ensures high availability through features Fail-over clustering that supports business continuity. Fail-over clustering can operate in active-passive mode, where the passive server waits for a fail over, or in active-active mode, where both servers are online and one can fail over to the other. For disaster recovery, log-shipping features can ship transaction logs to a remote site where they are subsequently applied. High availability is of course dependent on the underlying Windows platform's stability. MSS2000 also provides access to external data not via gateways but rather with Microsoft Data Access Components, which is included in many Microsoft products.

By including object components into its new versions, like Oracle and DB2 did, SQL Server acts like an ORDBMS and allows users to build its applications with data objects such as ActiveX Data Objects (ADO) including Command object, Connection object, Record object, Recordset Object and Stream object.

The weakness of MSS2000 is that it has not been as quick to implement Java in the database as IBM and Oracle did.

3.1.4 Object Oriented Database Development

Object-oriented databases, resulting from an integration of object-oriented programming capabilities with database capabilities, have stimulated interest within the database community. OO programming represents a disciplined programming style that incorporates four OO software-engineering principles: abstraction,

encapsulation, inheritance and polymorphism. The addition of a rich set of type constructors and a persistent storage mechanism to an OO language produces a database system that is conceptually in advance of the relational database.

Fundamental to the OO approach is the concept of an object model and, until recently, a variety of research activities have resulted in a diversity of techniques where the absence of a “standard” model was considered to be an impediment.

In an attempt to remedy the situation of diverse standards and development, the Object Database Management Group (ODMG), part of the Object Management Group (OMG, an industrial consortium established to promote object technology, has proposed an industry standard for object databases. Currently, ODMG represents a convergence of the various object database techniques, and incorporates all major facilities from the existing object-oriented database products. ODMG has provided SQL equivalence for the OODBMS, i.e. the OQL (Object Query Language), which supports their data model. ODMG’s white page, ODMG 2.0, states that, “the ODMG specification allows SQL querying by basing its OQL on SQL-92.” The example of a query provided inspects all children of all “Persons,” but can restrict the result to the people living on Main Street and having at least two children, who do not live in the same city as their parents, as described by Figure 3.2 (Cattell, 1997).

```

Select c.address
From Persons p, p.children c
Where p.address.street = "Main Street" and
count(p.children) >= 2 and
c.address.city != p.address.city

```

Figure 3.2 A sample of an OQL Statement

This OQL statement navigates from the Person class using the child reference to another instance of the Person class and then to the Address and City classes.

To be able to access an unquiet object, Object Identifier (OID) is introduced for OODB modelling. Dendrinis (Dendrinis, 1997) describes this important element of OODB modelling where each database object is given a unique object identifier (OID) corresponding to the unique real-world object it represents by the underlying OODB system. This is the opposite to an RDBMS implementation, where a real-world object may possess different keys in different relations. Relational DBMSs typically support value-based access to persistent data (Srinivasan, 1997), i.e., if an application needs to access a particular row in a database, it has to query the database using the name of the relation that the row is in and a primary key value that is equal to the value of the primary key value of a row in the table. This form of access to persistent data alone is inadequate in an object-oriented application, since objects might actually have identical values but be different objects. This is because object-oriented applications support non-first-normal-form values where an object can contain another object (e.g., two employees might own the same make, model, and year of a car but each respective car object might not be shared between the two employees, resulting in an identical valued car object in each employee object). OIDs might also be needed for direct access to an object in a database.

Further features are encapsulation, inheritance and persistence. The structure of an object in an OO database may be arbitrarily complex by including the information concerning a specific object. This is in contrast to traditional relational databases where the information about an object is often scattered over many

relations. Such inclusion of all information, known as encapsulation, concerns not only the data but also operations performed on it. Persistence is that feature of an OODBMS, which deals with the transience of programming language objects that persist solely during a program's life. An OODBMS provides capabilities whereby objects may be created to exist permanently and, further, which may be shared by any program written in an OO programming language and extended with accessing operations provided by the OODBMS. Current implementations of such languages are O2 (by O2 Technology, Inc.) and ObjectStore (by Object Design, Inc.).

3.1.5 Conversion between the RDBMS and ORDBMS

Relational to Object

To advance the new DBMS technologies, some relational databases have been re-engineered into Object databases. For example, the U.S. Naval Research Laboratory has developed a relational database system call NEONS (Naval Environmental Operational Forecasting System) manage huge quantities of geophysical data. Its Object-oriented version is being developed to accommodate data previously managed in an RDBMS environment. This development involves mapping the relational schema into an object-oriented (OO) schema, and, as pointed out by Ramanathan & Koduri (Ramanathan, 1995), the OO schema should subsume the relational schema in such a way that there is no loss of data information. This also changes the way a user looks at the data because data organization differs in a manner of changing relational tuples into objects and relationships - from a relational to an object-oriented form.

The transformation of relational schemas into object-oriented schemas is further evidenced in an algorithm developed by Meng, Kameda and Chang (Meng, 1995). In their case, multiple databases in various forms utilise an OO front-end. To present users with an object-oriented view, the schema of each relational database is transformed to an equivalent OO schema.

Object to Relational

For those who need to transfer object to relational databases for specific purposes such as retaining the relational front-end, an opposite approach may be used by re-engineering an OO model into an RDBMS schema. In this approach, an application sometimes combines an OO development method with RDBMS implementation. The benefit here is the separation of design concerns from implementation. Blaha, Premerlani and Shen (Blaha, 1994) provide such example. In the example, an Object-model compiler automates mapping rules and converts the object model into SQL code, which can then be used to generate relational tables. Blaha found that this approach frees application experts from the details of database structure during early development and lets them customise tables for a particular RDBMS. The aim of this type of approach is to remain a current RDBMS to the users, but apply an OO technique internally for database modelling. The benefits of the approach includes less training is requested by the users.

3.1.6 Current Trends

An advantage of these current trends in database management is that of increasing the value of stored data by capturing more of its semantics. Where the

semantic behaviour of data is stored in a database, then that behaviour becomes a resource that may be shared across many applications. Such data semantics include domain-specific knowledge, such as how to find the area of a polygon, and organisational policies for inventory management. Representing these semantics within the database (in the form of user-defined functions, constraints, and triggers) reduces the need for redundant coding in individual database applications. This, Chamberlin (Chamberlin, 1997) observes, makes applications easier to develop and eliminates inconsistencies by having object features such as inheritance and polymorphism, to providing better protection for the integrity of the database.

Darwen and Date (Darwen, 1995) have challenged the industry to turn away from today's relational database products and to begin developing databases conforming to the formal Relational Model proposed by Code in 1969. Darwen and Date describe OO features that "are orthogonal to the Relational Model, and therefore that the Relational Model needs no extension, no correction, no subsumption, and, above all, no perversion", in order for OO features to be accommodated in some database language that could represent the foundation we seek. In order to meet the storage requirements of modern data, Darwen and Date propose to extend the notion of the formal relational model by. Crigler and Orooji (Orooji, 1999) propose that such an extended model would be much more efficient for the vast majority of applications that are now being moved to object-oriented databases. Crigler and Orooji observe three trends with regard of modern databases:

OODBMSs

Object databases were originally conceived as a way to provide persistent storage for object-oriented programming languages such as C++. Interestingly, information from Object Design, the maker of ObjectStore, indicated that “full join”, a feature of Codd’s original relational model, was being added to Object Store’s supported features.

From an example of managing a set of spectrally analysed signals based on OODBMS, Dendrinis (Dendrinis, 1997), explains that the OO approach in DBMS first appeared in OO programming languages by encapsulating data and operations within the same module. Dendrinis suggests this approach has become the “heart” the widely used OO languages, such as Object Pascal / Delphi, C++ and Visual Basic etc. The OODBMS examples, described by Dendrinis, O2 and ObjectStore, enable created objects to persist and be shared by any program written in an OO programming language and extended with OODBMS accessing operations.

ORDBMS

“Object-relational databases attempt to merge the notions of objects and relations, generally providing support for a dialect of SQL, but treating attributes, tuples and even relations as objects.” (Orooji, 1999)

There are many examples of ORDBMS and the methodology for their implementation is commonly available from major DBMS vendors. Presently, Oracle8 (Anstey, 1999) is providing object-oriented features while its underlying databases are still relational. Its data-modelling tool includes an UML-like branch and its new object extension touches upon several subject areas listed below:

- Semantics
- Abstract data types
- Methods
- Pointers
- Support of unstructured data
- Support for complex objects
- Support of client-side objects
- Object views

Other advertised features that will be introduced with the next version of Oracle are inheritance and extensibility. It is said that Oracle8's inheritance features "will provide for subclass objects to inherit or assimilate the data structures and behaviours of the class above it in the hierarchy". The extensibility will also allow "users to create software cartridges consisting of objects, methods operators, and aggregates defined by the user". The Oracle8 server "will subsequently treat these cartridges as native data types"(Anstey, 1999).

Object-Oriented Access to Relational Databases

"Object-Oriented access to relational databases is an effort to provide an object-oriented programming facility for the treatment of data still housed in relational databases" (Orooji, 1999). This case is typically seen when people use object databases to isolate that part of their IT systems where high-speed performance with complex data is required. Other processing, such as back office analysis, will be done using an RDBMS. Often the object database data is transferred to the relational

database during quieter processing times. Barry (Barry, 1998) calls this “database diversity” and sees it as “using the right tool for the right purpose”. Barry uses the Chicago Stock Exchange as an example of this database diversity. The company keeps two days of trading: namely, that of yesterday and that of today, in the Versant OODBMS. All the trading activity occurs using Versant, which is a fault-tolerant system preventing the loss of any trade transactions. After each day’s trading, the data in Versant is transferred to an Oracle relational database. The Oracle database is used for historical analysis and reporting. However, lately, there has been limited time to load Oracle at the end of the day and breaking apart the object data and mapping it to the relational database is too slow. Therefore, the stock exchange is seeking alternatives such as moving some data to Oracle during the trading day.

In the above examples, we can see a trend towards Object-Relational. As an object extension of RDBMS, ORDBMS differs from OODBMS mainly, as described by Barry (Barry, 1998), in the use and storage of the data. However, ORDBMS products do require the mapping of the object model on to a relational model. Such mapping requires additional development and can sometimes cause a significant performance bottleneck.

McFadden (McFadden, 1999) found Object-Relational technology arrears to be preferable to other database technologies in terms of adding value to a business. For example, the web databases application iWebDB uses Object-Relational approach to achieve easier web database administration (Loeser, 1999).

3 REVIEW OF RELEVANT LITERATURE

The major RDBMS vendors have countered this emerging technology by providing extended ORDBMS. The major relational database vendors name their OO extension modules of the DB server as Datablade (Informix), Extender (IBM), or Cartridge (Oracle).

According to McFadden, substantial growth in the Web has profoundly changed our view of applications. New applications are needed to give people both inside and outside an organization access to data. Developers want these new applications to give users the flexibility and dynamism that they have come to associate with the Web. One practical way to create these new data-rich applications is to use the power of object-relational technology, upon which the Web is predominantly built.

OO accesses to relational DBMSs are common nowadays, especially when various types of database systems are employed. Since different types of database systems often use different data models and query languages, the co-existence of both Relational and Object-Oriented approaches makes it difficult for users who are familiar with one type of system but need to access data stored in a different system. One solution is to construct a front-end system using a single data model together with a single global query language on top of different types of existing systems. Currently, many organizations are interested in heterogeneous database systems using an OO system for their front-end system (Meng, 1995). Applications that combine an OO development method with an RDBMS implementation may benefit from the use of an approach that separates design concerns from those of implementation. Meng predicts that the coexistence of both OO and Relational approaches will continue for

at least the next decade, although the two are likely to service different application niches. "RDBMS will probably stay in business and high-reliability applications, while OODBMS will be used more in computer-aided engineering, design, and manufacturing".

Relational approach

Cummer (Cummer, 1998) observes that the flexibility of data types offered by an Object-oriented approach is not necessary for most business usage. Cummer states that "while OODBMSs promises to extend multimedia functionality to databases through an almost unlimited variety of data types that may be stored and retrieved, some industry watchers are not so sure those various data types are being used, or even that there is a need to use them. RDBMS fits all of their needs and SQL-Based RDBMS is more mature and easier to use."

RDBMS are also believed to "offer the advantages of: a firm theoretical foundation, mature technology that has governing standards, and the ability to work with a declarative language." OODBMS have their own advantages of rapid navigation of data structures, support of a greater number of data types (for example, audio and video), and the ability to integrate cleanly with at least one programming language." Different data merits a different approach, with some being suited to a relational approach while other data is more suited to an OO model. For example, Ramakrishna believes that the RDBMS data model, having fewer features to master, simplifies optimisation of queries, thereby yielding more efficient execution. Ramakrishna anticipates that the additional features in an ORDBMS will mean that it

will “not be as efficient as the relational model” and that “the complexities of database design will become ever greater with the advent of ORDBMS.”

According to Blaha et al, new technologies bring new ways of dealing with RDBMS. The object-oriented paradigm, for example, is useful in designing such systems because object models are “expressive, concise, easy to develop, and less prone to the update anomalies that have plagued attribute-based database-design techniques”. Likewise, RDBMS are a viable implementation vehicle for object models because the technology is mature and many commercial packages are available.

3.2 *Other Literature of Significance to This Study*

3.2.1 Perceived Advantages of OO/OR over ‘Relational’ Model

While both relational and object-oriented data modelling are commonly available, the object-oriented model appears to overcome several shortcomings of the relational model especially with scientific databases, which are expansive and complex. Another significant strengths of OO model are the flexibility of data organization that facilitates efficient query processing. This flexibility is achieved because object identifiers, effectively pointers, are used for direct referencing. This differs from the relational case where the referencing between different objects is done solely by values (e.g., foreign keys) and therefore joining is required in order to navigate.

3 REVIEW OF RELEVANT LITERATURE

Complex data such as scientific data, poses a special challenge to DBMS. While the strength of the relational approach is its full provision of SQL, its weakness is cumbersome modelling of complex scientific objects (for example, genomic objects) due to normalisation. The relational modelling of information ordering is cumbersome results.

When one attempts to model the ordering information into tabular forms, the retrieval of constituents in the sequence requires repeated joins due to the nature of chained referencing. This deficiency stems from a lack of support for aggregated data types in the relational data model.

The relational approach has also proven to be insufficient, for several non-conventional applications such as CAD, CASE, office automation, multimedia databases and knowledge databases. In all these cases, the flat structure of the model makes the representation of complex and structured data difficult (Nanni, 1992). This is because not all persistent data is stored in the database. In fact, some data are still kept in files with specific file format, for example, engineering applications, CAD tools and electronic documents. This type of applications use files structures not only because of their complex nature, but also because the performance of relational systems is poor when handling complex data structures using traversal operations (Ebert, 1999).

Dendrinis (Dendrinis, 1997) also agrees that an OODBMS schema is preferable to the RDBMS alternative in terms of linkage to the actual objects, property inheritance and data and method encapsulation. With OODBMS, the data are

protected through their encapsulation within the objects and the operations are hidden from users, who see only the necessary interface. An OODBMS can be easily extended or modified by the addition of a new spectral analysis method or by changing an existing one. Thus, modularity and expendability are two other very important advantages of the object schema.

3.2.2 Complex Data Handling

What might be considered complex data here are data with complex structures/representation (scientific database, CAD), data with complex type (multimedia), and data which requests massive storage and fast retrieval.

One of the major strengths of the object-oriented approach found by Shin in his implementation of a genetic database with both OO and relational approaches is its highly flexible data modelling power which offers an elegant way of representing complex scientific objects. In his comparison, two criteria are used. First one is how to describe what is called the IS-A-PART-OF relationship. For example, the illustration of *item A* “*is a sub-object of*” the *item B* (or the item A “*is a sub-object of*” the negative item B). The second one is how to represent ordering information, such as that illustrated by “a portion of ECOILVGE is followed by a portion of ECOILVGMED which in turn is followed by a portion of ECOILVGE”. In object-oriented databases, the IS-A-PART-OF relationship and ordering information can be described in a straightforward way by using reference pointers and aggregate data types such as set, list, and array. Shin explains that this mainly stems from the aggregate data types that eliminate the need for normalisation. The use of direct references with object identifiers (OID) further facilitates structural resemblance.

Bear (Baer, 1999) found that OODBMSs outperform RDBMSs at handling complex relationships in data. He claims that a RDBMS is usually a good choice when relationships among different types of data are fairly fixed and well known and that an OODBMS can handle the changes based on a mix of product storage and transportation attributes.

Wang, Sheu & Cotman (Wang, 1998) have also found that a limitation of the RDBM system is data retrieval. First, in a relational database system such as MS Access and SQL Server, information has to be stored as tables (relations). This, Wang et al claim, is not a natural representation for complex objects such as patient cases which are associated with initial patient intake, his/her medical history, physical exams, neuron-psychological tests, neuron-pathologic exams, reports, laboratory results, neuron-imaging, diagnosis, and treatments. Second, if the information is stored in a relational database which uses the SQL query language, it is limited in scope in many situations (i.e. some queries cannot be expressed in SQL).

Finally, for map-based information, software developers are considering the OODBM as a newer generation of technology for the solution since the RDBMS will not contain pictures (Ferris, 1998). However, MSS2000 has come up the solution to allow images to be contained in its databases.

3.2.3 Multimedia Database Management

3.2.3.1 Characteristics of MDBMS

Based on his observation of Visual Information Systems, Santini (Santini 2001) finds that image databases have a group of characteristics that makes image databases into an autonomous discipline.

Similarity. Image databases do not match symbolic representations of the meaning of images, but rely on a more loosely defined notion of similarity between images.

Generality. There is a great interest in methods that can be applied to a large class of images, despite the methods offered by restricted domain databases.

Interaction. By and large, image databases are not autonomous systems, but work in strict interaction with a user sitting in front of them.

Data complexity. This refers to the sheer complexity of providing and managing an adequate representation of the image data.

3.2.3.2 Usage of MDBMS

Further to his observation above, Santini made a loose list of fields in which it is possible to envision an application of image databases and content-based image retrieval:

- Entertainment (or recreation);
- Document preparation (inserting images in a document);
- Learning (distance learning);
- Graphics and advertising;
- Trademark search.
- Medicine
- Security and law enforcement

3.2.3.3 Development of MDBMS

Shih (Shih 2002) reviews search and retrieval methodologies of image databases in his edited book *Distributed Multimedia Database – Technologies and Applications*. Shih treats MDBM technologies as extensions to data management technologies of indexing and query processing to effectively support efficient content-based retrieval in DBMSs.

In Shih's view of a distributed multimedia database, it involves network technology, distributed control, security, and multimedia computing in which covered the issues of multimedia database (MDBMS) development including firstly the fundamental requirements from the perspective of its functionality, secondly the architecture of a MDBMS from the view of a software system, and finally various approaches to developing a MDBMS.

Requirements of a MDBMS

“Multimedia objects are different from traditional text or numerical documents in that multimedia objects usually require a large amount of memory and disk storage,” specified by Castelli (Castelli and Bergman 2002), therefore a MDBMS should be able to provide the following basic functions:

- Handles image, voice, and other multimedia data type;
- Handles a large number of multimedia objects;
- Provides a high-performance and cost-effective storage management scheme; and
- Supports database function, such as insert, delete, search and update.

Moreover, following issues should be taken into consideration:

- Composition and decomposition of multimedia objects;
- Operations of multimedia objects with media synchronization;
- Persistence object;
- Content-based multimedia information retrieval;
- Concurrent access and locking mechanisms for distributed computing;
- Security;
- Consistency, referential integrity, and error recovery;
- Long transitions and nested transitions; and
- Indexing and clustering.

Architecture of a MDBMS

Castelli outlines a MDBMS' architecture into three layers:

- The interface layer;
- The object composition layer; and
- The storage layer.

The interface layer needs to deal with object browsing, query processing, and the interaction of object composition/decomposition. The object composition layer works in conjunction with the interface layer to manage objects. The last layer, the storage management layer, includes two performance-related issues: clustering and indexing. Clustering means to organize multimedia information physically on a hard disk. Indexing means that a fast locating mechanism is essential to find the physical address of a multimedia object.

MDBMS Development Approaches

Castelli summarised the strategies for storing multimedia resources into four basic approaches:

- Rely on a regular file system;
- Use a traditional database management system with the support of an object-oriented interface;

3 REVIEW OF RELEVANT LITERATURE

- Use an object-oriented database management system, with user interface support; or
- Design the database from scratch, based on object-oriented concepts.

It is observed (Shih 2002) that the first approach relies on the users to manage multimedia resources by themselves, therefore there is no support for reuse of presentations; the second approach relies on a relational DBMS and its table-based organization become inefficient. Shih explains the nature of multimedia presentations makes it easier to organize a presentation using an object-oriented methodology. As Shih points out, the reusability (in both knowledge and software components) is one of the most important factors in improving the efficacy of a multimedia database; an object-oriented approach of development of MDBMS is advance than a RDBMS or even an ORDBMS.

3.2.4 Database Support for Image Databases

Indeed, the complexity and characteristics of image data and its distribution like ballistic firearm data create challenges in developing a functional Intranet BFIS.

So far, the development of ballistic firearm cartridge case identifying has considered that the collected images may be use to compare with a newly recovered cartridge case image to hope a match (link) can be found amount the collection. How to hand the images directly affect the efficiency of the identification.

Traditionally, associated with a metadata database, the images are stored and maintained separately as image files. The problems of this method of handling image data, as Besser describes (Besser, 2002), “affect the longevity of the images by images separating form their metadata, inaccessibility of software needed to view an image, and inability to even decode the file format of an image.”

Kauffman (Kauffman, 2000) argues that storing images in databases as a field “may consume a lot of resources” as firstly the data may be any length with only a pointer saved in the database, secondly “the data must be read in

chunks rather than all at once” and finally the head of a image needs special treatment before it can be displayed.

Major DBMS vendors have been developing techniques in handling image data and the concept of BLOB (Binary Large Objects) is used for image objects. Among them, Microsoft SQL Server 2000 (Microsoft, 2000b) supports a new **text in row** table option that specifies that small **text**, **ntext**, and **image** values be placed directly in the data row instead of in a separate page. This reduces the amount of space used to store small **text**, **ntext**, and **image** data values, and reduces the amount of disk I/O needed to process these values.

3.2.5 Intranet/Web and /Client/Server Mechanism Support

3.2.5.1 Intranet/Web Client/Server DBMS

Intranet DBMSs are commonplace in today's business environment and Simsion (Simsion, 1993) has found that the integration of data and transaction management achieved with an ORDBMS produces the following advantages over RDBMS in Intranet environment:

- reduction of network traffic;
- better performance in terms of speed.

3 REVIEW OF RELEVANT LITERATURE

From a wider perspective, the Intranet represents a framework where such advantages may become significant. Anstey (Anstey, 1999) supports this view as he describes the transition from RDBMS to OODBMS via ORDBMS as being a major technology merger in the industry, wherein database technology must advance to accommodate the demands of an object-oriented world, such as World Wide Web and Intranet.

Enright (Enright, 1998) views OODBMS with reference to Jasmine (CA) as used “either for a very fast object store for an intranet or Intranet application. The last has been discussed in previous topic. The first one is also observed by Anonymous (Anonymous, 1998) that the OODBMS allows accessing information more efficiently from remote of mobile sites and encourages greater use of the Web.

The detail of implementing a Web Database with an ORDBMS has been described by Loeser and Ritter who shows how the Web Database application iWebDB is implemented by using Informix’s IDS/UDO. Loeser and Ritter note that Web database developers can extend database systems using object user-defined functions (UDF) which can be executed within the server. In addition, by using object user-defined types (UDT), an index structure can be created and external data can still be accessed via SQL. While integrated functionality, such as C-Based UDF, works, Loeser and Ritter note

that problems can arise from limited modelling capabilities. However, this is a problem of all (commercially) available ORDBMSs, because there is no commonly accepted definition of the term “Object-relational”. Consequently, there is no system providing all object-relational features with full functionality. For instance, apart from Illustrate (no longer distributed), none of the available systems offers a reference type. As an alternative, people employed traditional relational primary/foreign key associations. Therefore, an OODBMS could be the solution of these problems.

To address the requirements of commercial Web publishing and enable the further expansion of Web technology into new domains of distributed document processing, the World Wide Web Consortium has developed an eXtensible Markup Language (XML) for applications that require functionality beyond the current Hypertext Markup Language (HTML) (Bosak, 1997). According to Barry, many web sites are currently using OODBMSs in anticipation of growth in XML use. As Bosak (REF) points out, XML is basically an object model with which OODBMSs may be expected to perform in superior fashion to RDBMSs or ORDBMS. In addition, the cost of development reduces because XML does not need to be translated into some other storage model, as it maps directly onto an object model.

To DBMSs themselves, the advent of client/server mechanism enables more robust systems to be designed and built. Despite providing a dramatic improvement in what users may accomplish, moving work from the server to the client introduces a set of fresh problems, including those of network bottlenecks, hardware conflicts, high maintenance costs and added complexity of system design. Together with the demands of client/server architecture, those of the World Wide Web and the Intranet may increase demands on a database to a point beyond which relational databases can support. It is anticipated that database technology needs to be advanced from the Relational Model to accommodate the demands of what is rapidly becoming an OO environment. In practice, a system may be decomposed into groups of concurrent objects, such as client objects and server objects, whereby communication and synchronisation are achieved by explicit message exchanges between the client and server objects. The messages are passed according to a predetermined protocol obeyed by the communicating objects. At a systems level, OIDs allow the implementation of references whereby one object may access another in order to pass the necessary message according to the agreed protocol. Since a database server object automatically handles many of the difficult and complex data processing tasks, including congruency control (locking) and transaction processing, an application's responsibilities are to issue requests (messages) for data and to handle returning data.

3.2.5.2 Complexities of Programming

Barry explains that one of the major benefits of using OODBMS is in Intranet application development. He views the use of OODBMSs allows developers writing less codes. He uses an example, when people are using an OO programming language such as Java or C++ to access the data, the programs do not have to translate into a database sub-language such as SQL, ODBC, or JDBC. Instead, this part of the function will be written in the programming language. Hence the systems are more consistent and request less code.

Kauffman, however, observes that using some facilities provided by RDBMS or ORDBMS products can reduce the complexity of programming. He found that, for instant, instead of using embedded SQL statement within the programs, the SQL statements and procedures could be pre-built and stored in a DBMS itself. MSS2000 has this facility called Stored Procedures.

3.3 *Specific Studies Similar to the Current Study*

This section describes factors such as the benefits and limitations of current ballistic DBMSs, including the need for Intranet and Client/Server DBMS.

3.3.1 Current Limitation of Computerized BFIS

Research into the usefulness of computerized BFISs has been conducted since the late 1970s (Hamby, 1999) as the result of rapidly growth, popularity, and relatively inexpensive cost of computers. There have been some successful developments into the area in USA, Canada and Australia: Drugfire (FBI, 2002), IBIS (IBIS, 2002), NIBIN (ATF, 2002) and FireBall (Smith et al., 1995).

A common theme among the literature is that most of the computerized BFIS are developed and applied overseas, mainly USA, Canada, France and Turkey, with the exception of FireBall, which is developed as a joint project between Edith Cowan University (ECU) and National Institute of Forensic Science (NIFS). Aim to develop an Australian self own BFIS, FireBall project was first created by Dr Smith and his team (Smith, 1995). It established the methodology in identifying firearms through identifying their cartridges. The project was than further developed into an desktop application named Fireball that catches cartridge images into a multimedia database sited in MS Access and to be searched/matched with existing records by the system (Watson, 2002). Despite the great success in applying the system in various Australian Police departments, FireBall is a desktop database application using Microsoft

application using Microsoft Access (MSA) as its DBMS that has some limitations that will be discussed in the following section.

3.3.1.1 Issues Desktop Only

The National Institute of Forensic Science (NIFS, 2002) states the purposes for Australasian Police Ministers Council to establish NIFS in 1991 is to provide a solution to the problem that “interstate isolation was one of the major weaknesses of forensic science in Australia.” The mission of NIFS was

To be an integral part of and a support base for the forensic science community, by working in partnership with all the elements of the community for the advancement of forensic science.

McFadden (McFadden, Joffer, & Prescott, 1999) observe that substantial growth in the Intranet has profoundly changed people’s view of applications. New applications are needed to give people both inside and outside an organization access to data.

However, desktop BFIS does not provide a shareable Intranet database for interstate laboratories. When a state laboratory examines a firearm and its components with a desktop BFIS, the possible criminal link between the states

may be delimited as the databases are isolated between the laboratories. Even though the email or mail system can be used to exchange information for suspected link, the potential or hidden links are not observed, the operation of mailing slows down the process, and importantly the collections of the ballistic information are restricted by only having local firearm identification information.

3.3.1.2 Lack of Scalability and Performance

Since the Scientific Crime Detection Laboratory at Northwestern University was bought by the City of Chicago and formed Chicago Police Department Crime Laboratory in 1938 (Hamby, 1999), forensic firearm identification laboratories have collected numerous evidences and using the linked and source-rich data collections solved many cases. By late 1989, the FBI Laboratory Division has implemented DRUGFIRE, an electronic database and computer network that was designed to digitally capture photographs of fired bullets and cartridge casings for collation both within the laboratory and those laboratories that also have the equipment. Since then with many forensic laboratories operate DRUGFIRE equipment with the result that numerous unknown suspect cases have been linked. The scale of data collection was huge. Besides of evidence information and images, a General Rifling Characteristic (GRC) file contains over 18,000 rifling characteristic measurements.

3 REVIEW OF RELEVANT LITERATURE

Hamby proves that a networked BFIS needs large covered information. This is due to the long durations of solving historical crimes, the collection of the historical information, the vast amount of specifications for the identification, and the large size of image information (Papaglannis, 2000). Together with the consideration of running a BFIS over the Intranet, the DBMS deployed needs to have the sufficient capacity to support the system.

Kauffman note that MSA “was originally developed as straightforward desktop database application ...was not designed with the Internet in mind and its not scaleable”. It is not only do not provide enough space for increasingly large amount of data and images, but do not allow enough users to access the database at the sometime. Another advantage that MSS2000 over MAS is that stored procedures are used not only to reduce the complexity of the programming, but to improve DBMS performance.

Another problem inherent in desktop MAS database BFIS, as Clearform (Clearform, 2002) point out, is its performance. This is because “Microsoft Access was designed with the primary goal of providing ease of use for inexperienced and experienced database users/developers. To accomplish this goal, Microsoft had to use a one-file system and embed many of its ease of use tools into the actual MDB file, data tables, and form objects. In fact, all of the objects and tables reside within one file format. This type of structure results in Microsoft Access performing sluggishly in high production

environments”. This view is evidenced by Microsoft (Microsoft, 2000a) as they point out that MSA takes longer to run because it does not have the ability to run queries in parallel (using multiple native threads within a single process to handle user requests) and to minimize additional memory requirements when more users are added.

3.3.1.3 Absent of Server/Client Mechanism

A Computerized desktop BFIS using MSA as its database tool does not have the Server/Client setup for the system. Kauffman expresses views as described “MSA is (only) a file-based DBMS (not a client-server application)”. He then, in opposite, described the MMS2000 “was developed with the Internet in mind ... it is a database server, rather than a stand-alone database (like MSA). It can be accessed through virtually any kind of network connection, including TCP/IP. ... It can be administered remotely. An ODBC System DSN to access a SQL Server database doesn’t have to reside on the same LAN as the database itself. It can reside anywhere that has access to the SQL Server via TCP/IP (any where in the world that has a connection to the Internet). This is because SQL server is a client-server DBMS, designed as a literal ‘database server’, just as Internet Information Server (IIS) is a ‘web server’. It can be accessed via TCP/IP, as the SQL Server can have its own permanent IP address on the Internet.”

3.3.1.4 Difficulty of handling Images

Computerized BFIS such as Fireball allows users to incorporate images. MSA stores images as OLE objects. However there appear to have difficulty in handling image data. Because of MSA stores a header that includes information about image type and device to be use to display the image. According to Kauffman, “stripping out the header information in an Access OLE field is one big headache”.

From discussion in section 1.3, image data handling is understood to be as important as text data handling. In development of the National Integrated Ballistic Information Network (NIBIN), ATF recognized the benefit to law enforcement that ballistic imaging and analysis could provide.

3.3.1.5 Improvement needed for Security

Several studies into the use of MSA have indicated a lack of security. Among the literature, Clearform (Clearform, 2002) suggest that MSA cannot be properly administrated in database rules based Internet environment. Microsoft (Microsoft, 2000b) goes further by promoting MSS2000 has improved security over MSA as it integrate with the Windows NT operating system security to provide a single log on to the network and the database. “This makes it much easier for us to administer complex security schemes.”

“An MSS database on a server”, it continuous,” is also better protected because unauthorized users can’t get to the database file directly but must access the server first.” Figure 3.3, reproduced from Microsoft MSS2000 document (Microsoft, 2000c), shows architecture of a MSS2000 security system. It is based on users and groups of users and illustrates how users and local and global groups in Windows NT can map to security accounts in MSS2000, and how MSS2000 can handle security accounts independently of the accounts in Windows NT.

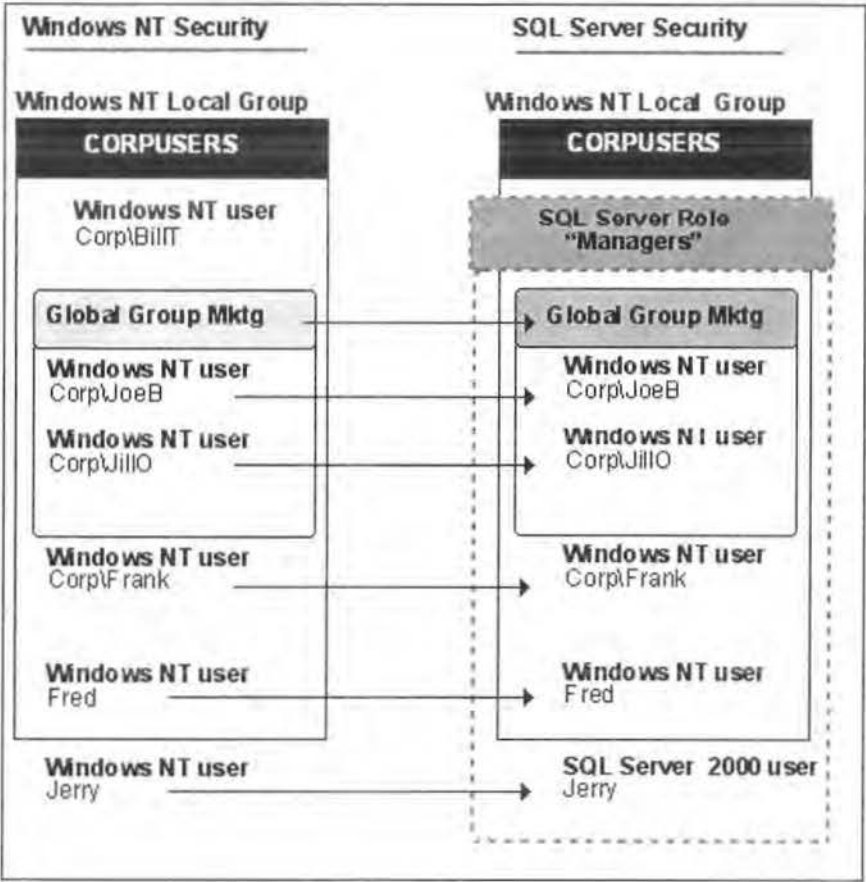


Figure 3.3 MSS2000 Security Architecture

The scenario of the diagram can be explained as following. The *CORPUSERS* local group contains two users and a global group, *Mktg*, which also contains two users. MSS2000 allows Windows NT local and global groups to be used directly to organize its user accounts. Additionally, the Windows NT users *Fred* and *Jerry*, not part of a Windows NT group, can be added to an instance of MSS2000 either directly as a Windows NT user (*Fred* for example), or as a MSS2000 user (*Jerry*).

MSS2000 extends this model further with the use of roles. Roles are groups of users organized for administrative purposes, like Windows NT groups, but are created in MSS2000 when an equivalent Windows NT group does not exist. For example, the *Managers* role contains the Windows NT *Mktg* global group and the Windows NT 4.0 users *Frank* and *Fred*.

MSS2000 also provides security at the application level through the use of individual database application roles. This ability allows the database administrator specify the privilege as detail as the accessibility of a column. To simplify the study, this part of the facility has not been implemented into the DBMS.

3.3.1.6 Needs for Availability, Recoverability and Reliability

Date (Date, 1993) assert “data in a database is subject to a variety of possible threats (both deliberate and accidental). For example, a disk might be

physically damaged or destroyed, or sensitive data might be exposed to an unauthorised user. The system - more precisely, the base DBMS - therefore DBMS has to provide an appropriate set of controls to protect the database against such threats.” The controls Date points out can be measured by availability, recoverability and reliability.

Shapiro (Shapiro, 2001) observed that, when comparing MAS and MSS2000, using MSS2000 as the DBMS for an Intranet system improves the availability of the system. This is because MSS2000 allows dynamic backup, either incremental or complete, of the database while it's in use. Consequently, the system does not have to force users to exit the database to back up data. This means that an Intranet BFIS based on MSS2000 can be running up to 24 hours a day, seven days a week.

In another study which compares MSA and MSS2000, Clearform (Clearform, 2002) found that in a case of system failure (such as an operating system crash or power outage), MSS has an automatic recovery mechanism that recovers a database to the last state of consistency in minutes, with no database administrator intervention. This recoverability is important to the Intranet BFIS as it can be up and running again right away after a system failure.

In its own comparison with MSA, Microsoft (Microsoft, 2000b) states that in MSS2000, each transaction is secured in the worse case of system failure and in the middle of complex updates by more than one user. MSS2000 treats all database changes inside a transaction as a single unit of work. By definition, either an entire transaction is completed safely and all resulting changes are reflected in the database, or the transaction is rolled back—and all changes to the database are undone. This advantage of MSS2000 gives the Intranet BFIS the reliability it requested.

3.3.2 Advantage of Computerized BFIS

According to ABC (Australia Broadcasting Corporation Television, 2001), “With Fireball police carry out almost 20 times more cross-checks than with the old system. Ballistic examination is reduced from days to just minutes. The data can also be sent across the Internet (by Email). And, even better, the system costs only about \$30,000 - a seventh of the cost of the equivalent FBI systems. ...Australia must have some sort of system and given that the competitor systems are of the order of two, three, four million dollars, you multiply that by the number of States. So the savings to Australia would probably start in the order of 8 million and blow out to maybe 12 or 14 million depending on how many systems you used across the country”. This view supports that of Snr Constable Peter Lawrence (Australia Broadcasting Corporation Television, 2001), who state that he will continuing to build his

database, confident that Fireball will continue to "assist him with his enquiries."

The successfulness of fireball is supported by evidence of solving some difficulty crimes and winning *Western Australia Information Technology and Telecommunications Award for Product of the Year* in 1999.

3.4 *Similar Study*

This section will examine studies undertaken by other researchers, especially Smith et al (Smith, 1995) and Li & Watson's {Li, 1998a #58} work which justifies the use of DBMS.

Computerized BFIS have enabled the use of image expressions, markings and other types of ballistic specifications, for which studies have been conducted. One such study is that of Smith et al, who developed a project called FireBall, which later was further developed, by Li and Watson.

3.4.1.1 FireBall

Fireball is a ballistics identification program developed by ECU researchers and it was adopted by several Australian police services and is

being trailed by police services in other countries. ABC (Australia Broadcasting Corporation Television, 2001) suggests that “Australia must have some sort of (computerised BFIS) system and given that the competitor systems are of the order of two, three, four million dollars, you multiply that by the number of States”. This comment is based on the fact that Australia has not had its own computerized BFIS until FireBall has been created in 1999 by group researchers from Edith Cowan University (ECU). The system has filled in the gaps in the fields of forensic science and technology in Australia.

The FireBall provides a MSA relational database of ballistic information where spent cartridge cases and bullets information and images are stored for later identifications. It is observed that with Fireball firearm examiners carry out almost 20 times more crosschecks than with the old system. The examination time is reduced from days to just minutes. The data can also be inquired and exchanged interstates by Email system. Figure 3.4 provides the main menu of Fireball.

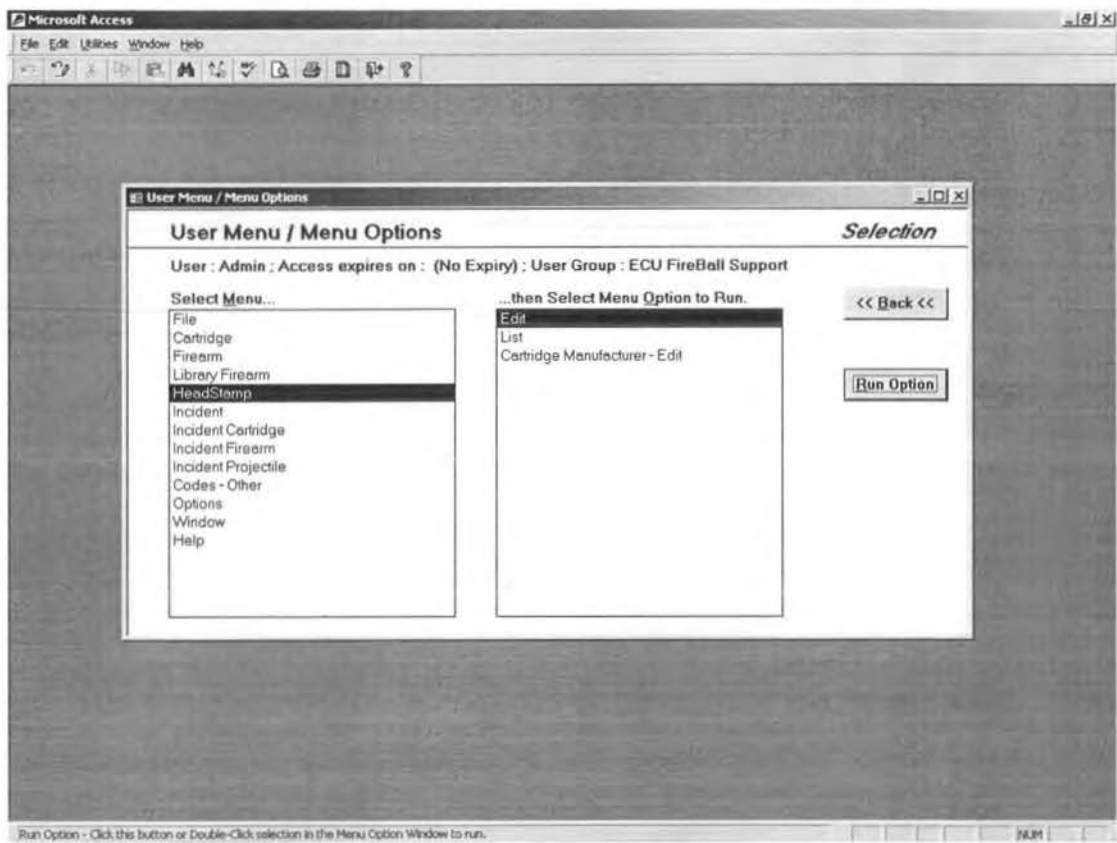


Figure 3.4 Sample of Fireball – User Menu

The Fireball displays images of each cartridge cases stored in its database with its classification information. Figure 3.5 provides a snapshot of the FireBall cartridge image.

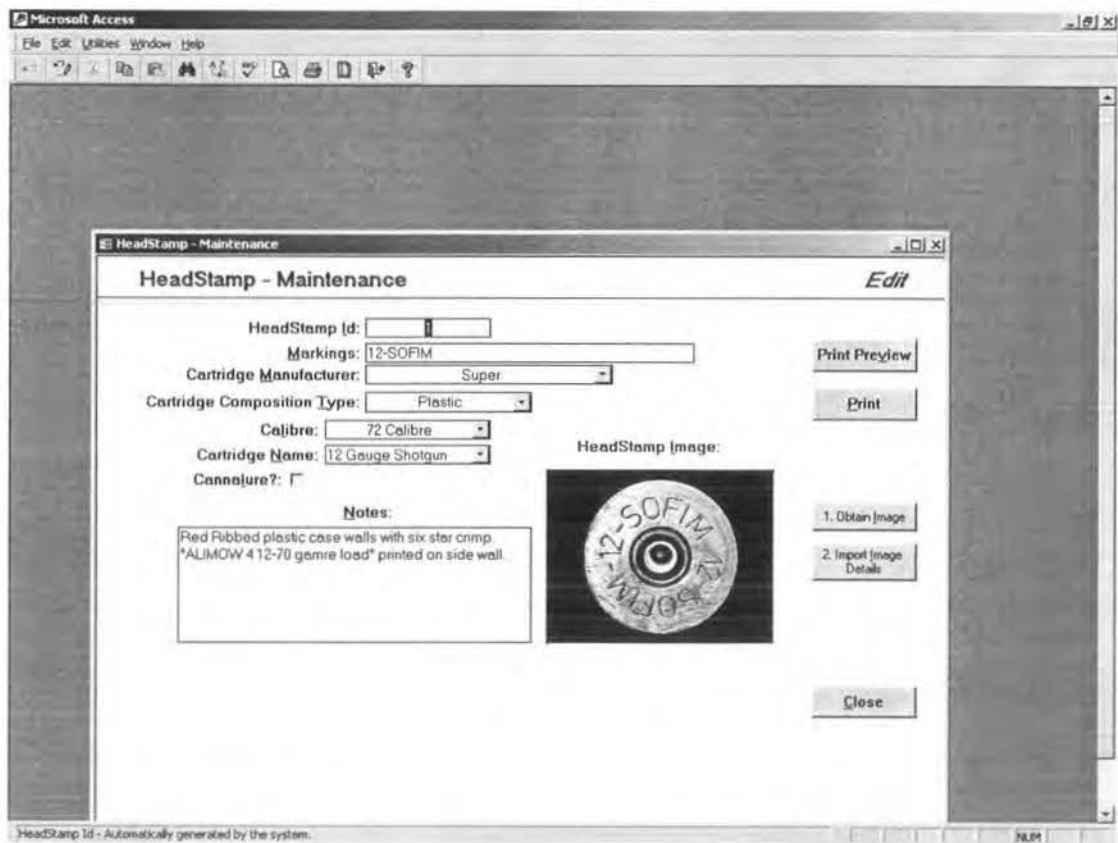


Figure 3.5 Portion of Fireball Cartridge Headstamp

The Fireball screen in Figure 3.5 shows a typical screen among usages of a system showing a headstamp image and its specifications.

The fireball attempts to use images as a source of information. The database contains marking, manufacture, calibre, composition-type and cannalure classifications. This provides a user with information on which cartridge of a calibre is searching and matching. Thus, the program provides a graphical interface for a firearm examiner, using emphasized images instead

of the original cartridges. The fireball appears to be the only attempt that uses image representation in a BFIS system in Australia.

In their study, Li et al (Li, 1998) use MSA secure the user logging in. However, they maintain the same access authentic methods for each user of the system. This means that although the user might have the *write* authority, they have no protections of the rest of the users who have only the *read* authority.

Table 4 listed the main differences between FireBall and proposed Intranet BFIS.

	FireBall	Intranet BFIS
Database	MSAccess	SQL2000
Programmming Language	MS Access Visual Basic	Active Server Page with VB Script
Application Platform	Desktop	Intranet
Image Data Type	OLE	Image
Communication	Email	Intranet

Table 4 Differences between FireBall and proposed Intranet BFIS

3.4.1.2 DBMS

There is a considerable amount of literature on the study of selecting appropriated DBMS for required applications, where the emphasis is: on the use of RDBMS, as evident in the work of Date (Date, 1995) and Crigler (Crigler & Orooji, 1999); or in the use of OODB, as described by

3 REVIEW OF RELEVANT LITERATURE

Chamberlin (Chamberlin, 1998) and Dendrinis (Dendrinis, 1997). There have been very few studies into the use of RDBMS together with Intranet ASP programming with the objects. Kauffman (Kauffman, 2000) explores the usefulness and different approaches taken in building Internet applications using a RDBMS like MSS2000 together with a set of database objects provided to ASP. Kauffman shows that a high level of performance may be achieved with a combination of MSS2000 and ASP. On the other hand, a lower level of administrability is delivered when a MSA is used as an Intranet DBMS. On the basis of this finding, MSS2000 are used for DBMS in the current study.

Table 5 is a summary of differences between the databases used by FireBall and proposed Intranet BFIS – former one uses Microsoft Access and later one uses SQL Server 2000.

3 REVIEW OF RELEVANT LITERATURE

Requirement	SQL Server	Microsoft Access (Jet)
Scalability		
	<ul style="list-style-type: none">• Symmetric Multi Processing (SMP) support	<ul style="list-style-type: none">• No SMP support
	<ul style="list-style-type: none">• Virtually unlimited number of concurrent users	<ul style="list-style-type: none">• Maximum of 255 users
	<ul style="list-style-type: none">• Terabyte levels of data	<ul style="list-style-type: none">• 2 GB of data
	<ul style="list-style-type: none">• Transaction logging	<ul style="list-style-type: none">• No transaction logging
Business Critical		
	<ul style="list-style-type: none">• 7 days of 24 hours (7X24) support and Quick Fix Engineering (QFE)	<ul style="list-style-type: none">• No 7X24 support
	<ul style="list-style-type: none">• Point-in-time recovery	<ul style="list-style-type: none">• Recoverable to last backup
	<ul style="list-style-type: none">• Guaranteed transaction integrity	<ul style="list-style-type: none">• No transaction logging
	<ul style="list-style-type: none">• Built-in fault tolerance	<ul style="list-style-type: none">• No built-in fault tolerance
	<ul style="list-style-type: none">• Security integrated with Windows NT	<ul style="list-style-type: none">• No integrated security with Windows NT
Rapid Application Prototyping	<ul style="list-style-type: none">• Access is UI for both engines and offers WYSIWIG database tools and built-in forms generation.	

Table 5 Comparison to the databases tool used by FireBall

3.5 Summary

The discussion in this chapter has sought to establish that, stemming from the recognition that a suitable DBMS will be selected, in which a proper database modelling method need to be applied so that the data can be correctly represented and accessed. The discussion has further sought to establish that the proven OODBMSs, now entering more general use, may appropriately be applied to the specified systems, affording better solutions and performance. Hopping integration may be generated from a combination of both OO and

3 REVIEW OF RELEVANT LITERATURE

relational approaches, so as to minimise disadvantages attempts based upon the coexistent of OODBMS and RDBMS. While there are a considerable number of studies available which investigate computerised BFIS, only the Fireball is available to be evaluated. The images of the cartridges are categorized in ways that create the search criteria that firearm examiners are looking at a list of images when engaging in a matching process. However, the database tool used by Fireball does not provide the scalability, accessibility and reliability required for an Intranet intended BFIS. There is no way of using database objects directly from the MSA database to ASP Intranet programming language. Thus, the program lacks a means of networking and reducing the possibility to link the proposed crimes. Few studies exist that examine the use of different type of DBMS such as RDBMS and ODBMS except the experiments employing ORDBMS demonstrate that having database objects enhance ASP Intranet programming and engage them in a integrated system. When compared to MSA, MSS provide a Server/Client structured Intranet work environment that may be operated efficiently and appropriately in BFIS, for this reason, this study employs MSS2000 instead of MSA.

4 RESEARCH DESIGN

4.1 *The Issues of Intranet and Database*

Within section 1.2 it has been established that desktop BFIS are the only computerized BFIS available in Australia and it is a much more efficient form of firearm identification compared to its original systems because of enhanced images and categorised characters of recovered evidences are collected and available when we search the proposed matches. As ABC (Australia Broadcasting Corporation Television, 2001) comments, “with Fireball police carry out almost 20 times more cross-checks than with the old system. Ballistic examination is reduced from days to just minutes.” However, section 1.1 also indicates that identifying firearms in a desktop system has its drawbacks. One such drawback is missing links between cases occurred in different states.

There is no doubt that identifying the linked cases plays an important role in law enforcement. According to Hamby (Hamby, 1999), linkage is important and is a limiter. That is, in most series situations, the links found between the cases identifies most powerful crimes that threat people lives. In the law enforcement world, forensic evidences is one of the best weapon that we can use to fight with the crimes, therefore we have to maximize every

effort and ensure that these efforts become fruitful. It follows, therefore, that network is a factor when conducting firearm identifications. ATF states that in isolated firearm identification systems, the work of the law enforcement is restricted by locations of the laboratory when examiners work together for a proposed linked crime, and time consuming when they are making enquiries and exchanging information which may result in a loss of potential link between the crimes and some effort has to be put in rebuilding those links when the next suspect arise. This observation is illustrated in Figure 4.1.

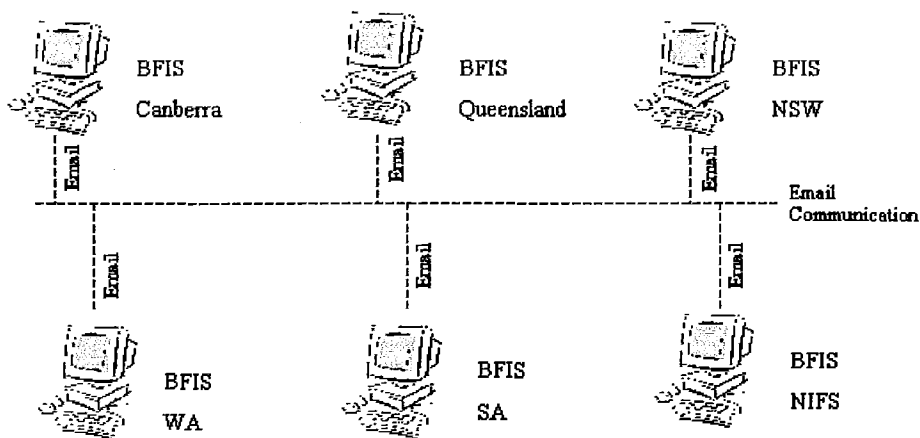


Figure 4.1 Isolated BFISs

A database capability restriction is more problematic when vast amount and complicated data are available to be stored in the ballistic firearm identification database, because restricted information eliminated the possibility for the examiners to discover possible linkage between the crimes.

Examiners need to make sure the database is properly administrated and protected, the database contains sufficient amount of historical data so that the proposed match is included in the database. On the other hand, Kauffman (Kauffman, 2000) suggests that Intranet DBMSs use MSS2000 provides great advantages. There are no scalability, security and reliability concerns for users to use MSS2000 to stored complex data over the Intranet. More importantly, users of the system do not have to physically sending enquiries over the mail or Email systems. Hence, ATF claims that firearm examiners can identify firearms with the system more effectively since they avoid sending enquiries and images to other part of law enforcement system to find the possible links. Figure 4.2 illustrates the use of an Intranet BFIS.

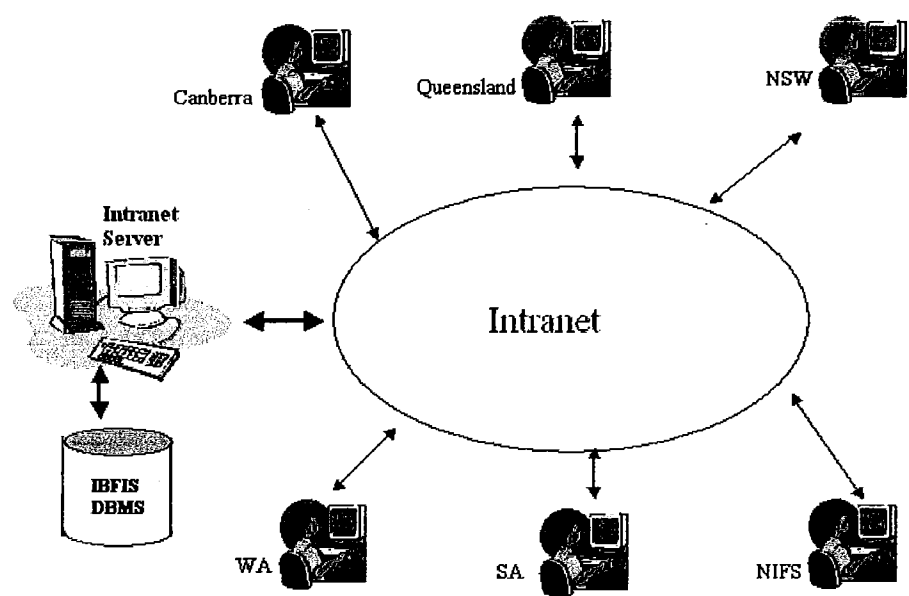


Figure 4.2 On-line BFIS

4.2 *Theoretical Framework of an IFCIS - CartridgeLink*

The previous section shows how isolations and insufficient DBMS affect firearm identifications, especially for identifying linked crims. Within this section, example 1 shows that existing methods of firearm identification differ from Intranet based firearm identification. Both manual and desktop firearm identification suffer from a linkage constraint evident in isolation and data availability, but the desktop system allows electronic identification process because the identification can be conducted by searching computerised categories and images. However, such searches are generally conducted in local laboratory database, which means that users must still using physical methods such as mails and Emails to make the enquiries for suspected linkage between the crims. The scenario in example 2 attempts to demonstrate how the same activities shown in example 1 are carried out when Intranet BFIS is employed.

Example 1 –Using existing framework for cartridge case identification

Investigations leading up to firearm identifications may involve numerous activities. Depending on what kind of the evidences is recovered from the crime scene, firearm identification may be through bullet or cartridge case identification. In this paper, we assume that cartridge cases are used for the identification, as this is more likely to happen in the cases. Using existing

methods for carrying out manual and computerized firearm identifications via cartridge cases, activities may be categorized into three basic groups (J. S. Doyle, 2002). Base on the diagram on manual or desktop BFIS shown in figure 4.1, the diagram may be expanded further to include an example. The expanded diagram is illustrated in Figure 4.3.

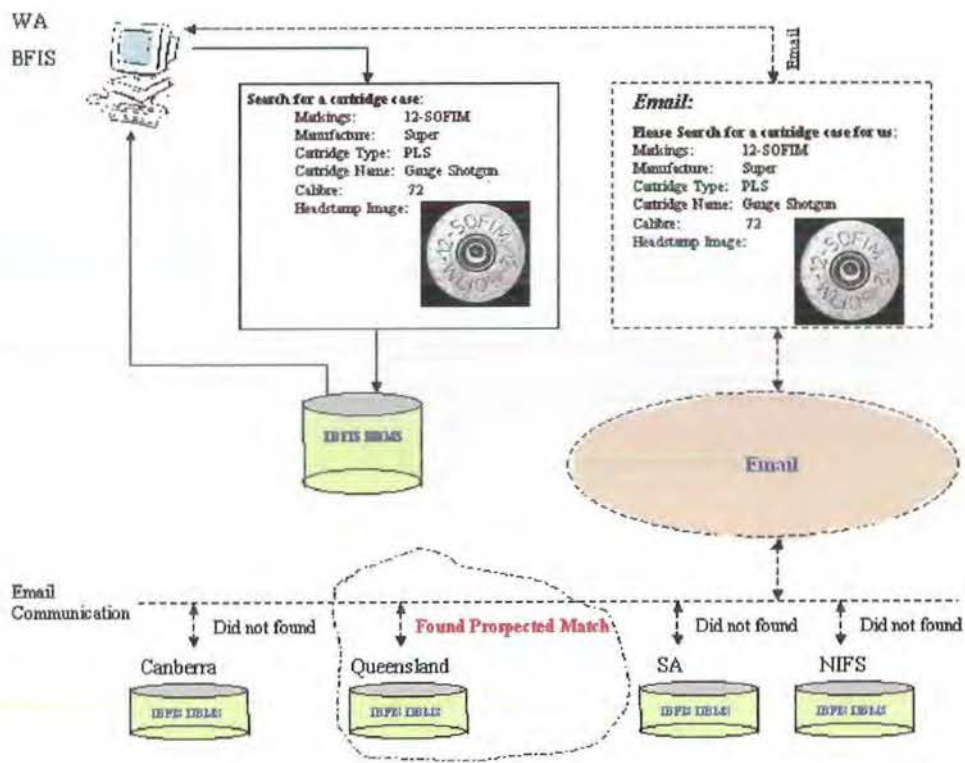


Figure 4.3 Conducting cartridge identification in current system

Scenario

Snr Constable Peter Lawrence, a firearm examiner of WA police forensic laboratory firearm examiner, receives a cartridge case recovered from

a recent crime. He is asked to identify the firearm that could have fired this cartridge case and produce a legal report for the Court.

Activity 1 Gain the access

Peter will need to be able to access both local and external forensic laboratories to find the suspected firearm wherever it might be collected in the system.

Activity 2- Imaging

Before Peter starts to identify the cartridge case, there are several activities that must be carried out including imaging and classifying.

Imaging:

During this phase, Peter uses an optical microscope to image the cartridge case. The enhanced image may be filed or saved into a desktop computer system. It is assumed that there is a collection of cartridge case images available produced from recovered firearms in past history.

Classifying

There is a set of standard classifications used in the laboratory. Peter will classify the cartridge case image according to this standard. The

classifications including marking, manufacture, composition-type, calibre and cartridge name.

Activity 2 - Searching

Peter now starts to search the possible match among the existing firearm cartridge casing collections. Both the image and classifications may be used for the search. He may or may not find a match from current local collection in WA. In order to demonstrate how the system works, we assume that Peter did not find the match among the recovered-firearm-cartridge-cases collection in WA.

Activity 4 - Enquiring

To search the possible matches national wide, Peter will now need assistants from other states' laboratories. He sends mails or Emails with the cartridge case image attached to them to all other laboratories at the same time. After searching their local data collections, other states' laboratories reply Peter with their findings. Whilst other states did not find any suspects, Queensland police has found several suspects and sends the information and images to Peter. With this information, Peter repeats the identification process he had carry out at first place. Queensland's suspected firearms should be either rejected or agreed to. For simplicity sake, assume that the one of the

firearms cartridge case is matched. Peter will complete his work and submit the final report.

It can be seen that the whole process of networking is quite lengthy and time-consuming. From sending other states the mail or Emails to receiving the responses, the same operations are repeated in different laboratories and much time and resources are consumed.

The whole scenario above assumes that the reproduction of cartridge case images based on recovered firearms is available in each laboratory. Thus, if any of these assumptions are not true, the whole process may take much longer.

Example 2 using computerised BFIS in an intranet environment

Using the same scenario as before, a likely outcome of using an Intranet firearm identification system on an Internet based environment is demonstrated in Figure 4.4.

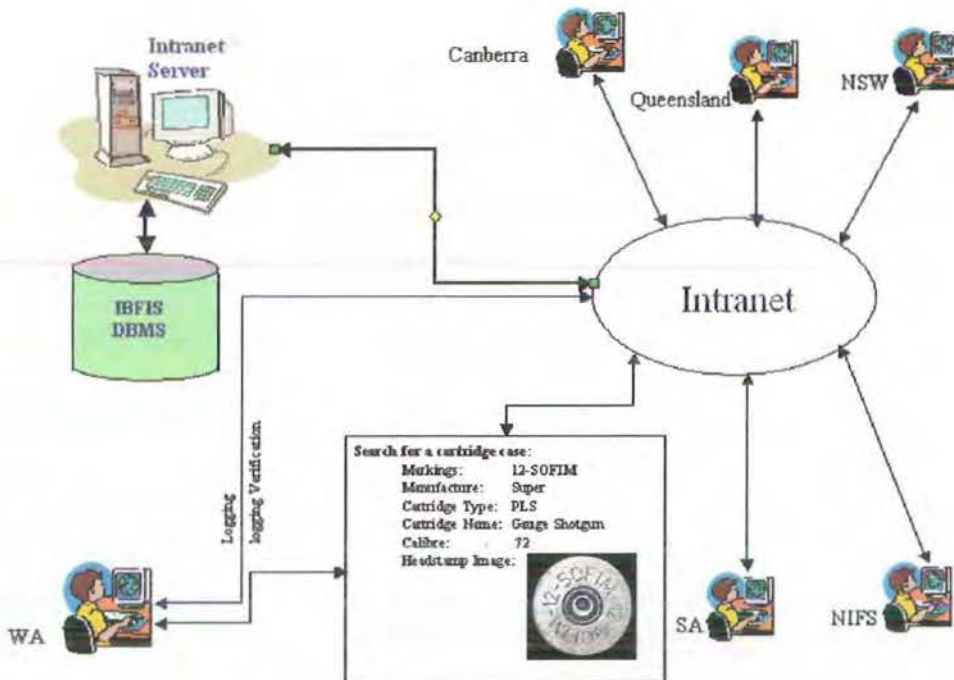


Figure 4.4 Conducting a Cartridge Identification in Intranet system

Activity 1 – Logging on to the CartridgeLink

This activity is similar to the previous one. Peter will need to log in to the system by providing his user name and password. If he has not yet had a user account with the system, he will need to registry him self over the system by providing his own username and password. The database administrator will set up a user account for Peter and notify him with the Email. Peter now can log into the system and he is given the privilege to update the database with new information and images.

Activity 2- Imaging/Uploading

The activities that Peter has to carry out remain the same as previous Activity 1. The only difference is that instead of filing or saving the image and classifications, he may now uploading the image and its classifications into the Intranet BFIS DBMS, as Peter is recognised with the privilege to update the database with new information and images.

Activity 3 - Searching

Activity 4 shown in **Figure 4.4** has now been eliminated because all such activities have been combined into activity 3. In an identifying session, there are continual searches for the proposed match. This is consistent with earlier suggestion made by BATF that firearm identification in on-line environment dose not disturb continuously processes.

Since it is conducted in an Intranet environment, all other state laboratories may also join in data collections without exchanging the mails or emails. The reproduction of cartridge case image for the same firearm only needed to be made for once. This also means that all the participants of the firearm identification obtain the privilege to access the database from different

locations such as Court or crime scene. The whole system will work more effectively.

It may be seen from these examples that by implementing BFIS on the Intranet, cartridges case images may be raised and searched without many of the constraints imposed by isolations and database, leading to more productive and effective outcomes from identifications. Additionally, introduction of the technology described will not reduce but increase participation rate of those laboratories involve. BATF demonstrates that the use of technology does not prevent users of computerised firearm identification from carrying out the same amount of interactions as they would have in manual and computerised BFISs.

4.3 Functions Beyond Scope Of Current Study

Having established that advantages of Intranet based BFIS using appropriate DBMS, the next step is to explain the design for the proposed system. The system is not designed to fulfil the complete requirements of an Intranet BFIS. Instead, the proposed system is simplified so as to satisfy the research questions outlined in Section 2.4. Some of the functions that are available in current desktop system such as capturing images or bullet identifying are ignored in this study.

4.4 *System Requirements*

In order to allow firearm examiners national wide to share data safely in a closed environment without the constraint of isolation and database, several steps must be satisfied. This section provides a brief explanation of the required steps and outlines necessary software and hardware requirements.

4.4.1 **Web Access**

To overcome isolation and database capacity restrictions, there must be a database server where users of the system may gain access in their own location, such as that provided by the Internet. In recent years, access to the Internet has increased significantly. One reason for this has been the use of the word-wide-web browser made popular by Internet browsers such as Netscape (Netscape, 2002) and Microsoft Internet Explorer (IExplorer) (Microsoft, 2002). The program takes advantage of the popularity of IExplorer, thus it is built for the configuration listed in Table 6.

Item	Value
OS Name	Microsoft Windows 2000 Professional
Internet Explorer	V6.0
IIS (Internet Information Service)	5.0
SQL Server 2000	Server

Internet Access	Promised
System Type	X86-based PC
Processor	X86 Family 6 Model 8 Stepping 6 GenuineIntel ~870 Mhz
Total Physical Memory	261,600 KB
Available Physical Memory	59, 368 KB
Network Adapter Type	Ethernet 802.3
Hard Disk	19.13 GB

Table 6 Configurations set up for the system.

At the time of this study, another popular Internet browser, Netscape 7.0 developed by Netscape, dose not fully support the ASP programming language, thus it is not used.

4.4.2 Intranet Access

Security usage of the database may be carried out among users of the system constructed for this prototype by creating an Intranet environment. Within a closed environment, Intranet pages may be accessed, employing Internet technology, only is people granted permission.

To develop an Intranet environment, a personal computer must be set up as a web server. There are several web servers currently available, but IIS 5.0 provided by Microsoft, is used because it is readily availability and

competitive to rest of components used for the system. Figure 4.5 illustrates how the Intranet CartridgeLink is managed by IIS5.0.

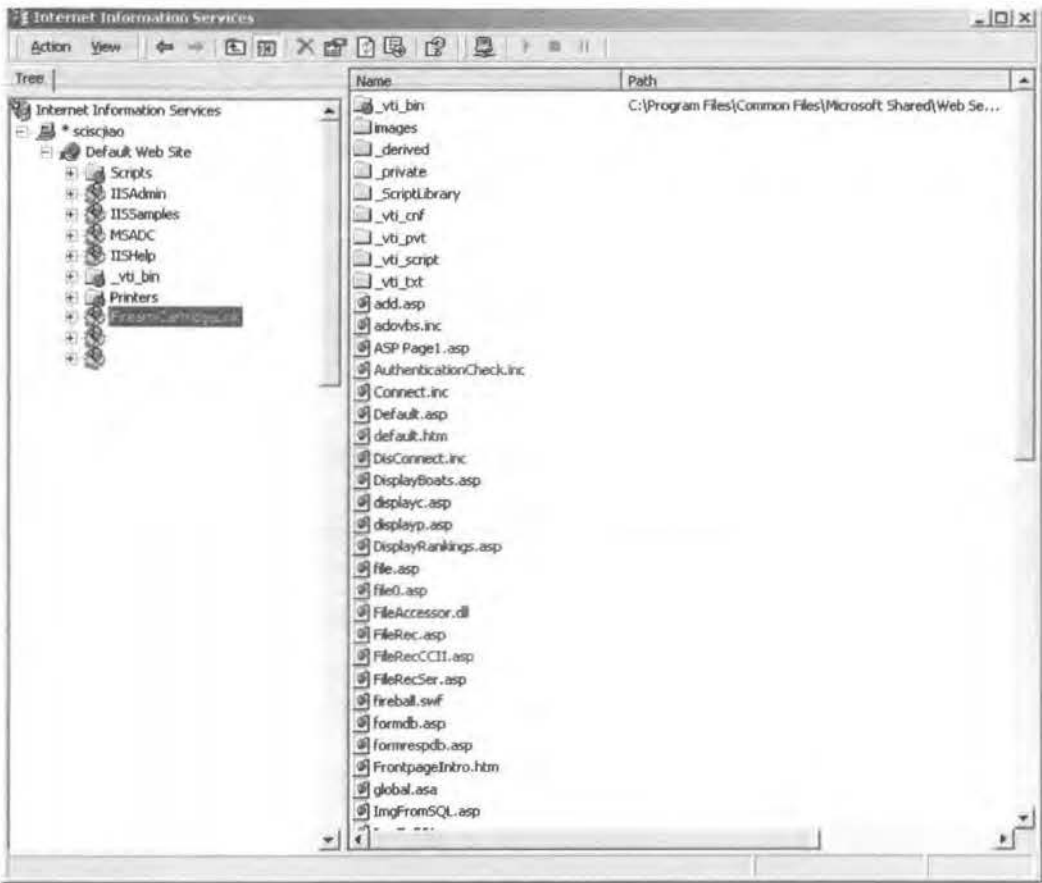


Figure 4.5 Microsoft Internet Information Service 5.0 (IIS5.0)

A home page shown in Figure 4.6 is established on the server as a starting point for users to access the system. All components used in the Home page are explained in Table 4.2.

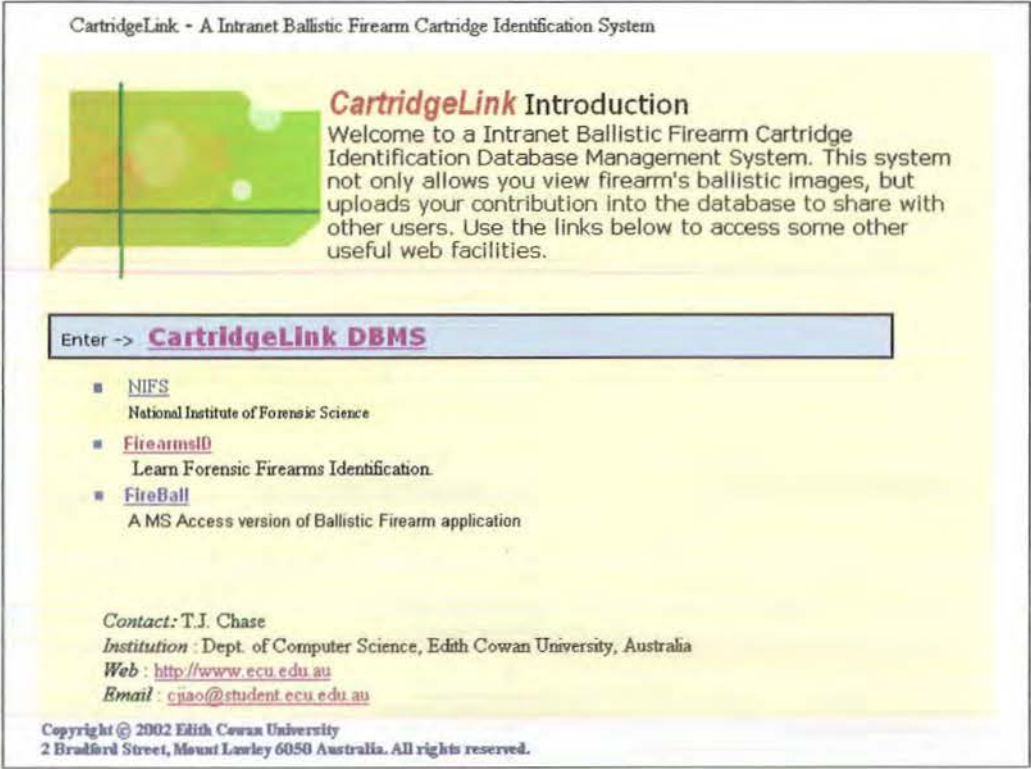


Figure 4.6 Open Intranet page

Component	Explanation
Intranet Title	The name of the Intranet
Introduction	A Brief description about the Intranet
CartridgeLink DBMS	Link to the entrant of the CartridgeLink Intranet
NIFS, FirearmsID ...	Link to other related and useful forensic web sites
Contact	For users to send Emails to the provider
Web	For users to view ECU web site

Table 7 Components of Home

4.5 User Interfaces Design

The flow char shown in Figure 4.7 can demonstrate the functions and operations of the system.

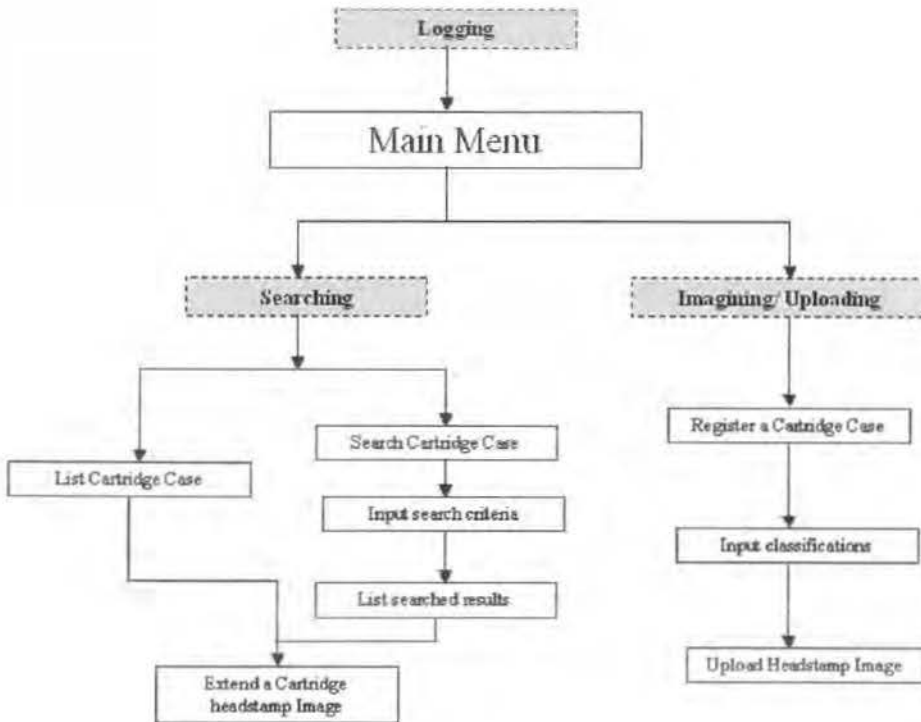


Figure 4.7 Functional Flow Chart

4.5.1 Logging and Registration Interface

When user enters the CartridgeLink DBMS, a logging page prompts for the user to enter his username and password for verification. Figure 4.8 illustrates the authentication process and Table 4.3 explains the components.

Ballistic CartridgeLink Intranet

Enter login details and click submit.

Username

Password

Submit

If you don't have a username and password you can

If you don't have a login ID and password you can [register here](#).

Figure 4.8 Logging into the CartridgeLink

Component	Explanation
Username	Prompt users to enter they assigned Intranet username
Password	Prompt users to enter they assigned Intranet password
Submit	Click the button to allow the system evaluates the logging
Register here	Link to a registration form for new users

Table 8 Explanation of Logging Page

For other users that have not yet gain their user accounts, they will need to register themselves by using “register here” option provide in the screen and waiting a Email to notify he if his registration is accepted. Figure 4.9 shows a user-interface for a new user’s registration and table 4.4 explains each of the components.

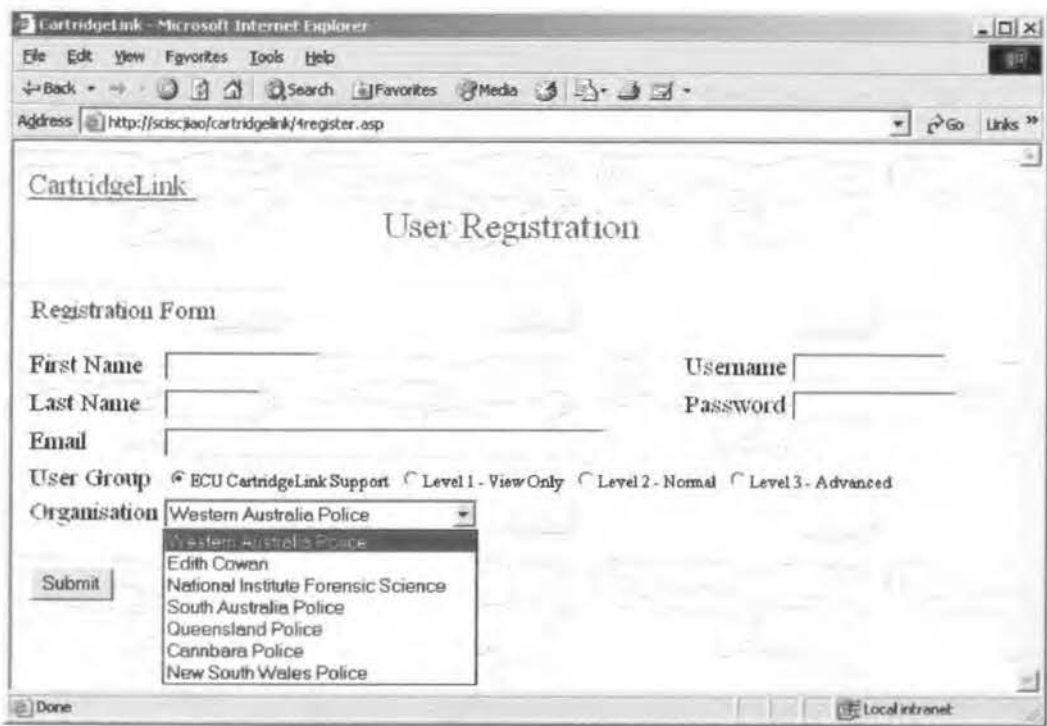


Figure 4.9 New User Registration

Component	Explanation
CartridgeLink Home Highlight	Return back to the Home page
Registration Form	Where users enter their personal/organization info
Submit	Notify the system for the submission

Table 9 Explanation of user registration page

When access is granted, the user is free to explore the database based on his user account privileges. A user may have the privileges to read and write the whole database, or he may only allow reading some certain data. It is database administrators’ responsibility to set up and maintain the user account for their laboratory.

Once users pass the evaluation of the logging, they may access the demonstration system by choosing an option designed to each user of the program, as shown in Figure 4.10. Table 4.5 explains the components in the page.

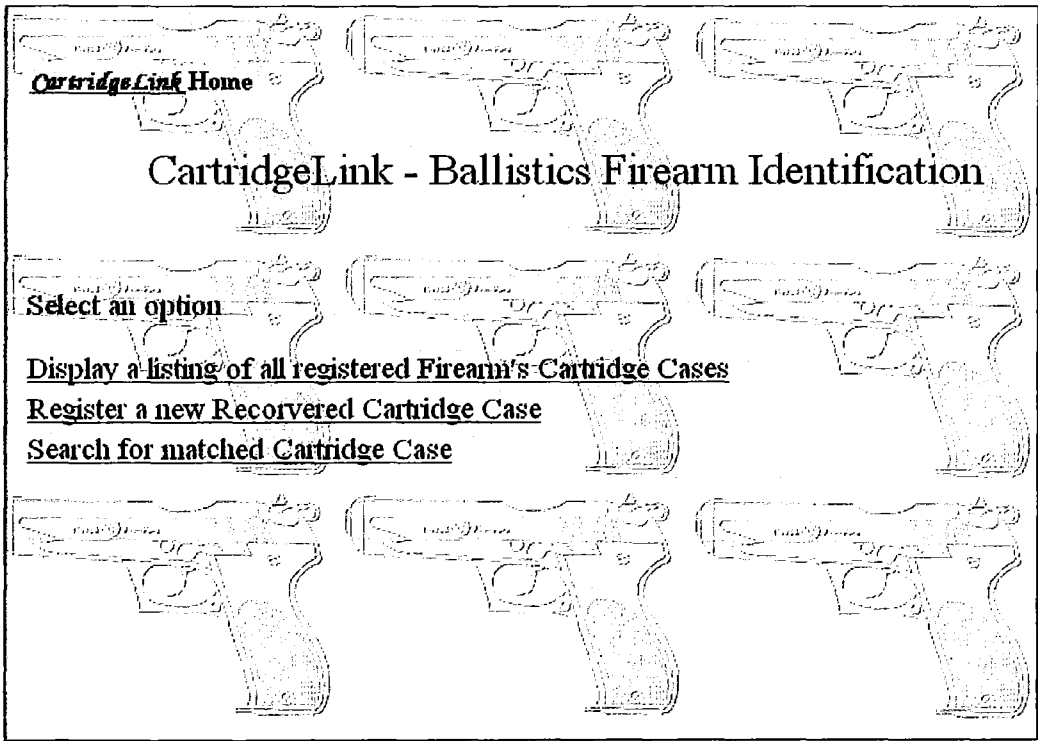


Figure 4.10 Menu for the options

Component	Explanation
CartridgeLink Home Highlight	Return back to the Home page
Option Display a listing ...	For user to overview existing cartridges in the database
Option Register a Cartridge ...	Link to the page where a cartridge data can be added into the database
Option Search a cartridge ...	Link to the page where a search criteria can be built for the search

Table 10 Explanation of the Main Menu page

4.5.2 Register a New Cartridge Case

Once the register cartridge case option is selected, the user may respond to a registration form by entering a set of classifications associated with the cartridge using the interface Figure 4.11. The cartridge case headstamp image can be uploaded into the form by giving the location of the image file. Since the program is only a prototype, a mechanism is not implemented to automatically detect the matched cartridge cases. Thus, users are asked to undertake this task. Table 4.6 explains the components in the page.

CartridgeLink - Inserts Cartridge Casing into Database - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites Media Print Mail

Address http://sciscjao/firearmCartridgeLink/insert.htm Go Links

Inserting Cartridge Casing Data into Database

[Show the List of Cartridges](#) [Return To Main](#)

Enter Cartridge Case Classifications

Calibre

Manufacture

Composition Type

Provided By

Cartridge Name

Enter Headstamp Markings, Image and Notes

Markings

Notes

File Browse...

Image Submit

Local intranet

Figure 4.11 Register a cartridge case into the database

Component	Explanation
<u>Show the list of the cartridges</u>	Click to show the list of existing cartridges
<u>Return To Main</u>	Click to return to the main menu
Enter Cartridge Classifications section	Select a value from each drop down list
Cartridge Name box	For users to provided a name
Enter Headstamp Info section	Enter the markings, Notes
Browse Button	Click the button to browse the location of the image file
Upload Button	Click to upload the image into the database and saves the associated classifications in the same record.
Image box	The uploaded image is displayed

Table 11 Explanation of cartridge uploading page

4.5.3 Search Cartridge Case

Searching cartridge case can be made by either over viewing all existing records or entering searching criteria for a specific group (or a single) record.

4.5.3.1 Listing Registered Cartridge Cases

Figure 4.12 shows the interface designed for a user when they selected the first option – “Display a listing of all registered Firearm's Cartridge Cases” or when they click the option “show the list of the images” described in pervious section 4.5.2. The interactions take place on this screen and users view images of collected cartridge cases. It resembles a listing where each of

4 RESEARCH DESIGN

cartridge case's classifications are shown and the image of its headstamp just one click away to be displayed. Through cartridge listing, users may read classifications and, in an additional screen, display the image by extending highlighted cartridge ID. Table 4.7 explains various components of the cartridge listing shown in Figure 4.12.

Showing Headstamp Image from the Database
[Insert Image](#) [Return To Main](#)

Headstamp ID	Markings	Manufacturer	Composition Type	Cartridge Name	Calibre
29	12-SOFIM	Super	Plastic	12 Gauge Shotgun	72
29	WINCHESTER 12 GA	Winchester	Plastic	12 Gauge Shotgun	72
27	WINCHESTER 12 GA	Winchester	Plastic	12 Gauge Shotgun	72
26	WINCHESTER 12 GA	Winchester	Plastic	12 Gauge Shotgun	72
25	12-SOFIM	Super	Plastic	12 Gauge Shotgun	72
24	12-SOFIM	Super	Plastic	12 Gauge Shotgun	72
23	WINCHESTER 12 GA	Winchester	Plastic	12 Gauge Shotgun	72
22	12-SOFIM	Super	Plastic	12 Gauge Shotgun	72
21	WINCHESTER 12 GA	Winchester	Plastic	12 Gauge Shotgun	72
20	12-SOFIM	Super	Plastic	12 Gauge Shotgun	72
19	12-SOFIM	Super	Plastic	12 Gauge Shotgun	72
18	WINCHESTER 12 GA	Winchester	Plastic	12 Gauge Shotgun	72
17	12-SOFIM	Super	Plastic	12 Gauge Shotgun	72
10	WINCHESTER 12 GA	Winchester	Plastic	12 Gauge Shotgun	72
9	12-SOFIM	Super	Plastic	12 Gauge Shotgun	72
8	WINCHESTER 12 GA	Winchester	Plastic	12 Gauge Shotgun	72
7	12-SOFIM	Super	Plastic	12 Gauge Shotgun	72
6	WINCHESTER 12 GA	Winchester	Plastic	12 Gauge Shotgun	72
5	12-SOFIM	Super	Plastic	12 Gauge Shotgun	72
4	WINCHESTER 12 GA	Winchester	Plastic	12 Gauge Shotgun	72
3	12-SOFIM	Super	Plastic	12 Gauge Shotgun	72
Number of records : 21, Table Name : CartridgeHeadStamp					

Figure 4.12 List of Existing Cartridges

Component	Explanation
CartridgeLink Home Highlight	Return back to the Home page
Insert Image Highlight	Click to Insert a image
Return to Main Highlight	Click to return to the main menu
Listing Area	List of existing collection of cartridge cases
Highlighted Headstamp IDs	Click to explore the headstamp images

Table 12 Explanation of Listing Existing Cartridges Page

To begin the listing, the users must first create a collection of the cartridge cases records. In this case, it is assumed that a collection is already established and waits to be updated and searched. The option of expend the image of a cartridge is available for each cartridge and the result of using the option is shown in figure 4.13. The user clicks the Cartridge ID and the image is shown with its classifications.

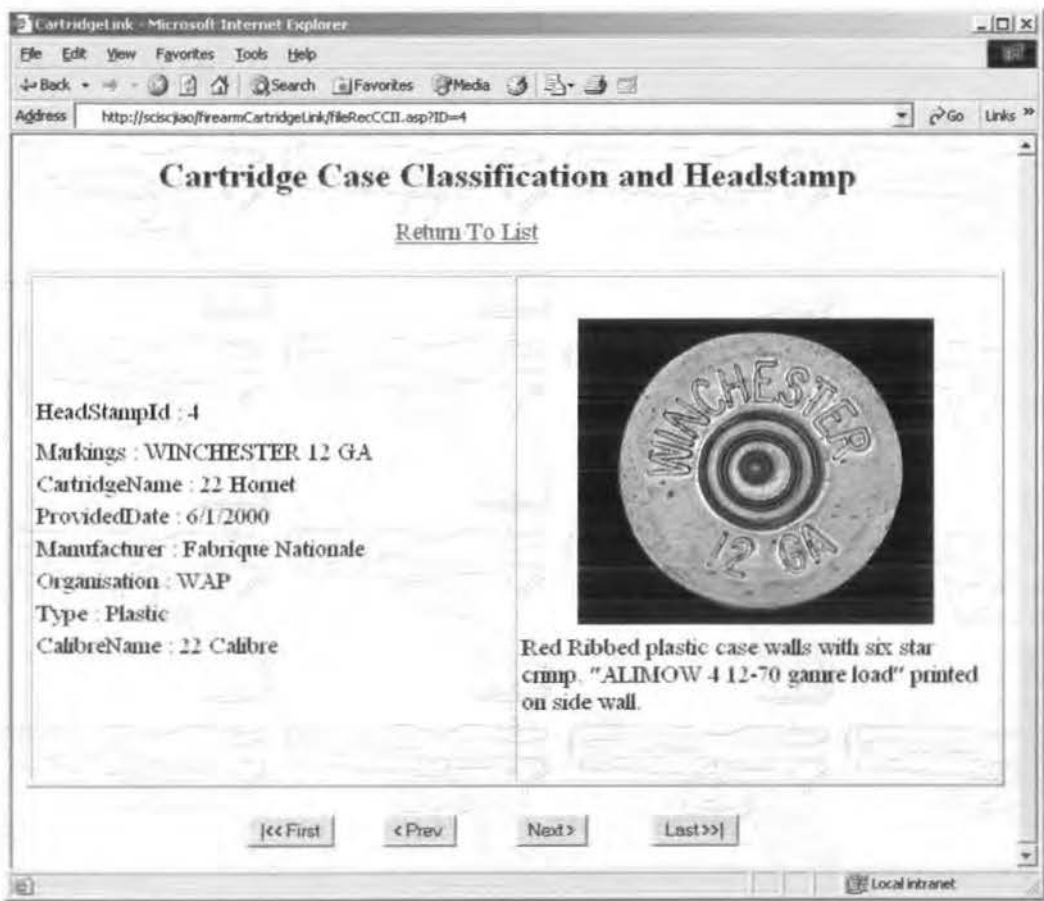


Figure 4.13 Extended Image included interface

Component	Explanation
Return To List	Return back to the original list
Left hand side box	Classifications of selected cartridge
Right hand side box top part	Cartridge case headstamp image
Right hand side box bottom part	Notes or remarks
Buttons at the bottom	Navigation through the collection

Table 13 Explanation of Extended Cartridge classifications and image page

4.5.3.2 Search Cartridge Cases with criteria

When *Search cartridge case* option is selected, the user should be prompted for a set of searching criteria using the interface show in figure 4.14. A set of the cartridge case met the criteria should be listed with their images available to be explored. All components used in the search function are explained in Table 4.

CartridgeLink - Search a cartridge

Enter Search details and click submit

Calibre

17 Calibre

Manufacture

Aust Defence Industries

Composition Type

Brass

Provided By

Western Australia Police

Search

Search Result

Headstamp ID	Markings	Cartridge Name	Notes
10	WINCHESTER 12 GA	12 Gauge Shotgun	Low brass head, black colour etruded ribbed plastic husk, shotsize marked on side wall, model "GAMELOAD" marked on side wall. Components assembled in Australia by BALLANTYNE.
8	WINCHESTER 12 GA	12 Gauge Shotgun	Check
6	WINCHESTER 12 GA	12 Gauge Shotgun	Small brass head, blue colour draw ribbed plastic husk.
4	WINCHESTER 12 GA	12 Gauge Shotgun	ICI, 'CIVIC' Model .22 long rifle, standard velocity, lead non coated round nose projectile fitted, brass cartridge case, one cannalure 0.56" from base.

Number of records : 4, Table Name : CartridgeHeadStamp

Figure 4.14 Image selection box

Component	Explanation
<u>Return To Main</u>	Return back to the main menu
Top part	Enter searching criteria
Bottom part	Shown search results
<u>Headstamp ID</u>	Click to explore headstamp image
Footer	Shows the number of matches

Table 14 Explanation of Extended Cartridge classifications and image page

4.6 Database Design

The following section describes the database design and various components involve in the study. A trial version of PowerDesigner is used to illustrate the design.

4.6.1 Data Model

Figure 4.15 is a physical data model diagram produced from PowerDesigner that shows CartridgeLink DBMS entities, links, keys and references.

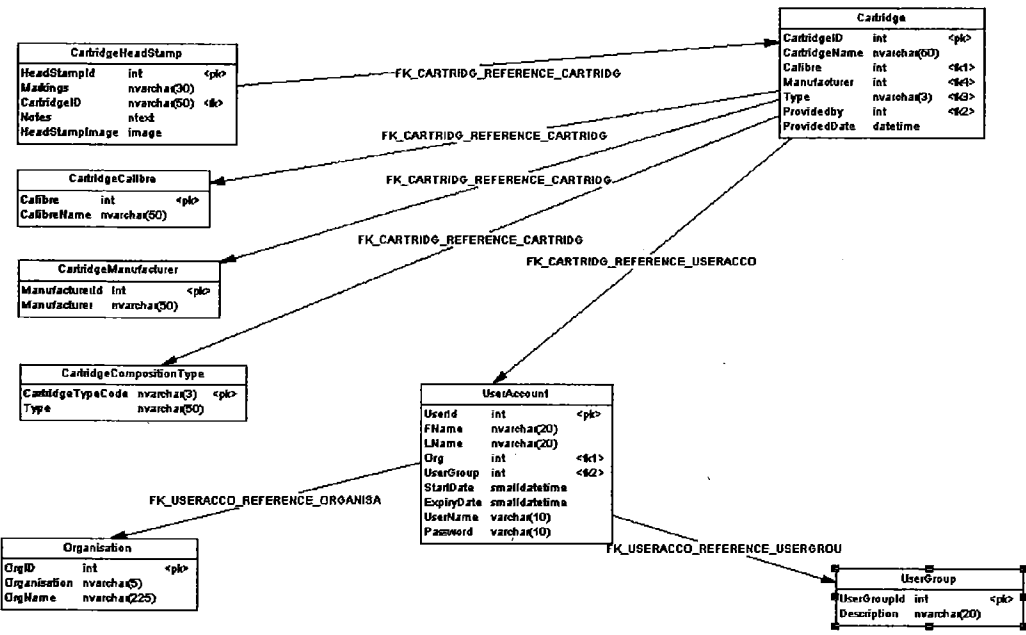


Figure 4.15 Data Model of CartridgeLink

4.6.2 Table List

The ERD shown above can be further broke down as a list of tables shown in Figure 4.16.

	Name	Code	Parent	Comment	G	DB
1	Cartridge	Cartridge	Physical Da		<input checked="" type="checkbox"/>	Microso
2	CartridgeCalibre	CartridgeCalibre	Physical Da		<input checked="" type="checkbox"/>	Microso
3	CartridgeCompositionType	CartridgeCompositionType	Physical Da		<input checked="" type="checkbox"/>	Microso
4	CartridgeHeadStamp	CartridgeHeadStamp	Physical Da		<input checked="" type="checkbox"/>	Microso
5	CartridgeManufacturer	CartridgeManufacturer	Physical Da		<input checked="" type="checkbox"/>	Microso
6	Organisation	Organisation	Physical Da		<input checked="" type="checkbox"/>	Microso
7	UserAccount	UserAccount	Physical Da		<input checked="" type="checkbox"/>	Microso
8	UserGroup	UserGroup	Physical Da		<input checked="" type="checkbox"/>	Microso
					<input type="checkbox"/>	
					<input type="checkbox"/>	
					<input type="checkbox"/>	
					<input type="checkbox"/>	

Figure 4.16 CartridgeLink DBMS Table List

4.6.3 Column List (Data Dictionary)

The fields (columns) contained in each table can be documented as a dictionary for the database. Figure 4.17 shows the column list of the database.

List of Columns

	Name	Code	Type	Comment	Data Type	P	F	M	D	Do
1	Calibre	Calibre	Cartridge	Calibre ID - Linked to table Cartridge	int					< N
2	CartridgeID	CartridgeID	Cartridge	Cartridge ID - Automatically generated by the system	int					< N
3	CartridgeName	CartridgeName	Cartridge	Cartridge Name	nvarchar(50)					< N
4	Manufacturer	Manufacturer	Cartridge	Manufacturer ID - Linked to table CartridgeManufacturer	int					< N
5	ProvidedBy	ProvidedBy	Cartridge	UserID - Linked to table UserAccount	int					< N
6	ProvidedDate	ProvidedDate	Cartridge	Date that the cartridge was provided	datetime					< N
7	Type	Type	Cartridge	Cartridge Composition Type Code	nvarchar(3)					< N
8	Calibre	Calibre	CartridgeCalibre	Calibre of Projectile	int					< N
9	CalibreName	CalibreName	CartridgeCalibre	Description of calibre of Projectile	nvarchar(50)					< N
10	CartridgeTypeCode	CartridgeTypeCode	CartridgeCompositionType	Cartridge Composition Type Code	nvarchar(3)					< N
11	Type	Type	CartridgeCompositionType	Description of Cartridge Composition Type Code	nvarchar(50)					< N
12	CartridgeID	CartridgeID	CartridgeHeadStamp	Cartridge ID - Linked to table Cartridge	nvarchar(50)					< N
13	HeadStampId	HeadStampId	CartridgeHeadStamp	HeadStamp Id - Automatically generated by the system	int					< N
14	HeadStampImage	HeadStampImage	CartridgeHeadStamp	Graphic Image of Cartridge Case HeadStamp	image					< N
15	Markings	Markings	CartridgeHeadStamp	Markings - Description of Unique HeadStamp Marking	nvarchar(30)					< N
16	Notes	Notes	CartridgeHeadStamp	Notes and remarks	ntext					< N
17	Manufacturer	Manufacturer	CartridgeManufacturer	Cartridge Manufacturer Name	nvarchar(50)					< N
18	ManufacturerId	ManufacturerId	CartridgeManufacturer	Cartridge Manufacturer ID - Automatically generated by int	int					< N
19	Organisation	Organisation	Organisation	Organisation code	nvarchar(5)					< N
20	OrgID	OrgID	Organisation	Organisation ID - Automatically generated by the system	int					< N
21	OrgName	OrgName	Organisation	Organisation Name	nvarchar(225)					< N
22	ExpiryDate	ExpiryDate	UserAccount	User account expiry date	smalldatetime					< N
23	FName	FName	UserAccount	User's first name	nvarchar(20)					< N
24	LName	LName	UserAccount	User's last name	nvarchar(20)					< N
25	Org	Org	UserAccount	Organisation ID - Link to table Organisation	int					< N
26	Password	Password	UserAccount	User password	varchar(10)					< N
27	StartDate	StartDate	UserAccount	User start date	smalldatetime					< N
28	UserGroup	UserGroup	UserAccount	User Group ID - Link to table user group	int					< N
29	Userid	Userid	UserAccount	User ID - Automatically generated by the system	int					< N
30	UserName	UserName	UserAccount	User Name	varchar(10)					< N
31	Description	Description	UserGroup	User Group name	nvarchar(20)					< N
32	UserGroupid	UserGroupid	UserGroup	User Group ID - Automatically generated by the system int	int					< N

Figure 4.17 CartridgeLink DBMS Table Column List

4.6.4 Keys

To maintain the integrity and performance of the DBMS, each table has been given a primary key that is unique to the table and use by other tables as foreign key. Figure 4.18 shows primary keys used by main tables.



The screenshot shows a window titled "List of Keys" with a table listing primary keys. The table has columns for an index, Name, P (Primary), and Table. The data is as follows:

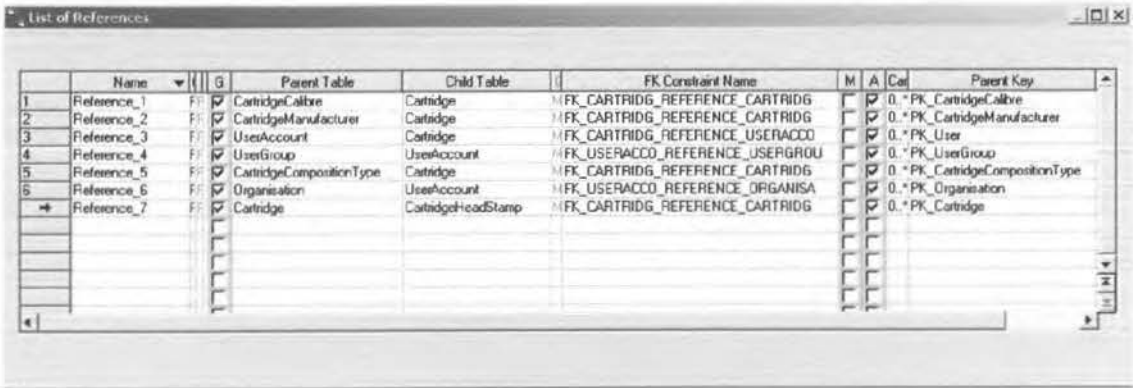
	Name	P	Table
→	PK_Cartridge	<input checked="" type="checkbox"/>	Cartridge
2	PK_CartridgeCalibre	<input checked="" type="checkbox"/>	CartridgeCalibre
3	PK_CartridgeCompositionType	<input checked="" type="checkbox"/>	CartridgeCompositionType
4	PK_CartridgeHeadStamp	<input checked="" type="checkbox"/>	CartridgeHeadStamp
5	PK_CartridgeManufacturer	<input checked="" type="checkbox"/>	CartridgeManufacturer
6	PK_Organisation	<input checked="" type="checkbox"/>	Organisation
7	PK_User	<input checked="" type="checkbox"/>	UserAccount
8	PK_UserGroup	<input checked="" type="checkbox"/>	UserGroup

Figure 4.18 CartridgeLink DBMS Tables Primary Keys

4.6.5 Foreign Keys (References)

Each table uses foreign key(s) as references to the linked the table(s).

Figure 4.19 lists the foreign keys and the tables use them.



The screenshot shows a window titled "List of References" with a table listing foreign keys. The table has columns for Name, G, Parent Table, Child Table, FK Constraint Name, M, A, C, and Parent Key. The data is as follows:

Name	G	Parent Table	Child Table	FK Constraint Name	M	A	C	Parent Key
Reference_1	FF	CartridgeCalibre	Cartridge	MFK_CARTRIDG_REFERENCE_CARTRIDG	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0 * PK_CartridgeCalibre
Reference_2	FF	CartridgeManufacturer	Cartridge	MFK_CARTRIDG_REFERENCE_CARTRIDG	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0 * PK_CartridgeManufacturer
Reference_3	FF	UserAccount	Cartridge	MFK_CARTRIDG_REFERENCE_USERACCO	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0 * PK_User
Reference_4	FF	UserGroup	UserAccount	MFK_USERACCO_REFERENCE_USERGROU	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0 * PK_UserGroup
Reference_5	FF	CartridgeCompositionType	Cartridge	MFK_CARTRIDG_REFERENCE_CARTRIDG	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0 * PK_CartridgeCompositionType
Reference_6	FF	Organisation	UserAccount	MFK_USERACCO_REFERENCE_ORGANISA	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0 * PK_Organisation
→ Reference_7	FF	Cartridge	CartridgeHeadStamp	MFK_CARTRIDG_REFERENCE_CARTRIDG	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0 * PK_Cartridge

Figure 4.19 CartridgeLink DBMS Tables Foreign Keys

4.7 *Summary*

Following on from chapter three, which discussed the limitations of current computerized ballistics firearm identification systems, within this chapter, components are described that overcomes those limitations. Additionally, research questions posed in chapter two have been explained and the prototype is been designed. The next chapter will present the implementations and results of the prototyped system.

5 IMPLEMENTATION AND RESULTS

Based on the description from the previous chapters, that our fundamental aim is to identifying cartridge case over an Intranet interface and Intranet DBMS, in order to find the matches to firearms. This concealment is to be accomplished in a manner which is reliable in terms of the DBMS selected to equip vast amount of data including images. Design decisions are required for the selection of appropriate DBMS tools to perform the specific components of the implementation, and for the choice of a suitable programming tools which are adequately integrated to produce a prototype system illustrating the scenarios previously identified.

5.1 *Selection of DBMS*

For reasons of availability, scalability, reliability, economy, compatibility (to the programming language used in this project – see next paragraph) and its feature-similarity to typical image data storage, as the review made in Chapter 3, a proprietary DBMS SQL Server 2000 (Microsoft, 2000b) was employed. Usage of the evaluation system's hardware was constrained for the study's purpose, to those facilities normally found within an Internet Server. The constraints precluded the use of clients.

5.2 *Selection of Programming Language*

Similar reason prompted the initial use of the VB scripts and HTML encapsulated by ASP and HTML pages for the Intranet's programming, as the vehicle for user interface generation. The ASP, however, proved to be incomplete in its support for Netscape browser. The limitation caused by competitions between Netscape and Microsoft and Netscape supports Java as its HTML generator. Notably, Microsoft Internet Explore supports both Java and ASP script as a marketing strategy and it is more popular to the developers. This is one of the reasons that Microsoft Internet Explorer is used as the platform.

Timing measurements were made using *SQL Profiler* a facility provided together with MSS2000 that can monitor the performance of an instance of SQL Server.

5.3 *Implementation of the Study*

The design discussed in chapter 4 suggests three scenarios to be implemented:

- The authorisation of the valid user name and password, with the feedback informs the user if the logging information is correct,

what privilege is associated to the logging if the it is valued – this is labelled the logging method;

- The uploading of the filed cartridge case images, encapsulated within classified information package– this is labelled the upload method;
- The searching of the specific cartridge with a set of known classifications – this is labelled the search method; (a full list of the cartridge cases is giving for overview existing records)

It may be seen that variations and combinations of the above scenarios may be employed to gain further improvement. However, for the purpose of the study, implementation has been limited to these scenarios.

The implementation concerning each of these will now be presented in turn, with every one being discussed in terms of:

- Its requisite implementation DBMS in the MSS2000
- Its requisite implementation code in the ASP language;
- The test data chosen to demonstrate facilities offered by the technique; and
- The resultant production given by the application of the scenario's DBMS with its Intranet ability.

Timing measurements yielding a measure of CPU resource usage, observed for each scenario will also be presented for each methods in later section 5.5. Annotated source code listing may be found in Appendix B.

5.3.1 MSS2000 Enterprise Manager

Before implementing three methods described above, we need to implement designed DBMS from Chapter 4 into MSS2000. Figure 5.1 shows implemented DBMS CartridgeLink and Table 5.1 explains the various components used for this project.

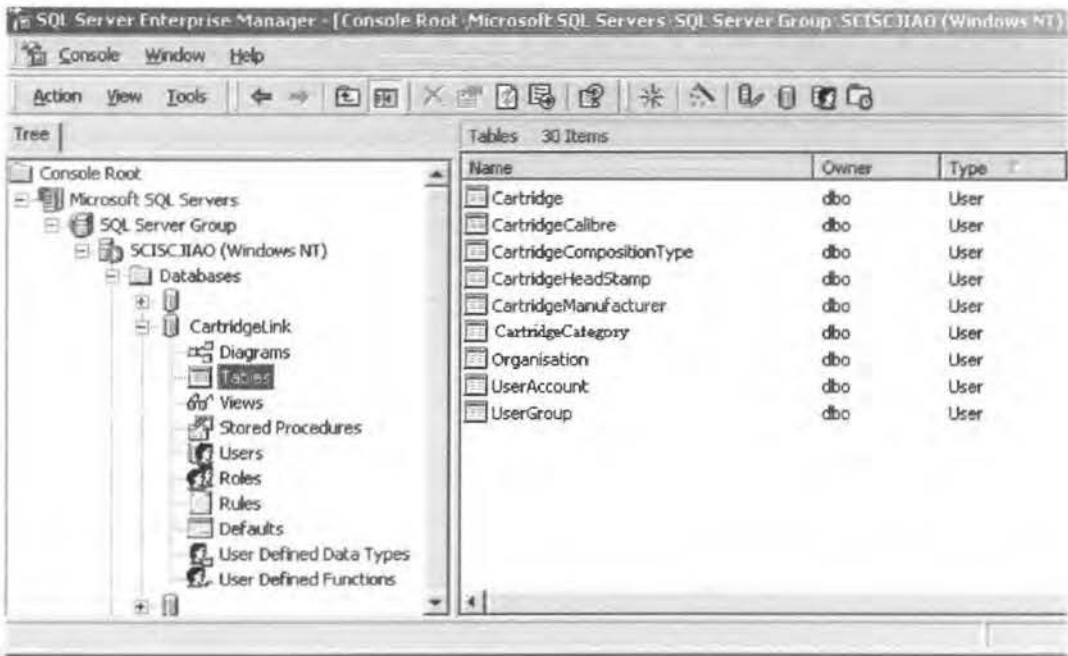



Figure 5.1 Intranet CartridgeLink’s DBMS *CartridgeLink*

Component	Explanation
Left hand part	A tree structure shows the components under the CartridgeLink
Diagram (see Figure 5.2)	The Diagram pane presents the graphic display of the tables we have selected from the data connection. It also shows join relationships among them
Tables	List the tables contained in the database
Stored Procedures (see Figure 5.3)	List the stored procedures built for the application
Users (see Figure 5.4)	List the users who were administrated with privilege that allows to access the database

Table 15 Explanation of DBMS *CartridgeLink* components

5.3.1.1 **Diagram**

The diagram CartridgeLink (Figure 5.2) shows the tables and their fields, the keys, and the joins between tables. It also defines the type of joins between the tables. A  defines a one-to-many join.

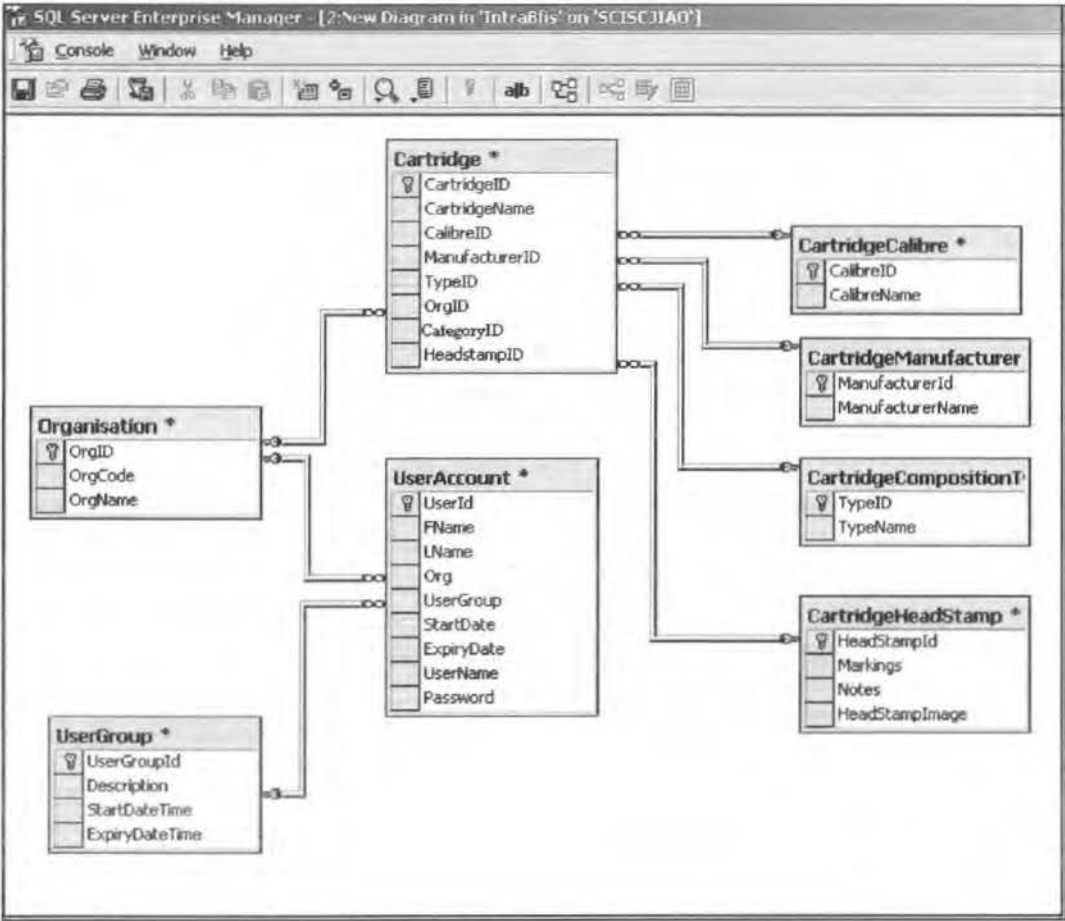


Figure 5.2 MSS2000 database CartridgeLink's Diagram

5.3.1.2 Tables

As a prototype we are only using 9 tables to support the basic methods of the application. Each table's definition is given as following:

Cartridge

A collection of cartridge records.

Each record contains a cartridge's name and a set of Ids that leads to the details of its manufacture, calibre, category, composition type, headstamp image, and organization that provides the record.

CartridgeCalibre

A collection of the possible calibre classifications.

CartridgeCategory

Two categories, *Lab* and *Crm* are used to separate the cartridges reproduced by laboratories and the cartridges recovered from crime scenes.

CartridgeCompositionType

A collection of the possible composition types.

CartridgeHeadstamp

A collection of cartridge's

headstamp images. One cartridge case would be imaged more than once to emphasis different marks.

CartridgeManufacture

A collection of possible cartridge manufactures

Organisation

A collection of organizations that could provide cartridge data for the system

UserAccount

A collection of the users who have the privileges to access the application (note: it is different from the *users* from Figure 5.4 shown later)

UserGroup

A collection of a user group that horizontally define the level of usage for a particular user.

5.3.1.3 **Stored Procedures**

Stored Procedures were used to pre-build the SQL Statements with passed parameters. This technique was used to reduce the complexity of the programming and improves the performance of the system as described in Chapter 3 section 3.3.1.2 Compare with embed SQL statement.

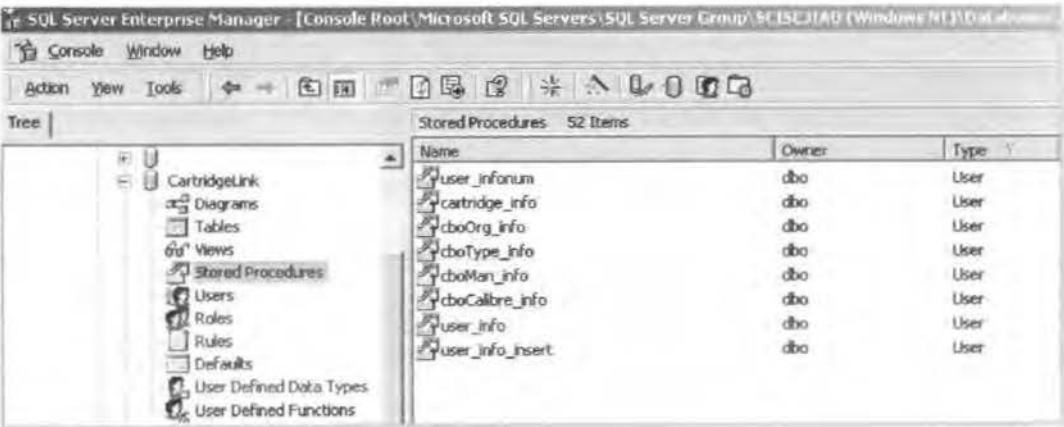


Figure 5.3 Stored Procedures Pan lists the stored procedures programs

The usage of the stored procedures will be mentioned in following sections in context to the programming implementation and the contents of the stored procedures will be given as APPENDIX C users configurations.

5.3.1.4 **Users**

The Users Pan shown in Figure 5.4 explores the database’s users and the property of User Administrators. The current users of the database and their privileges of the activities are shown in Table 5.2.

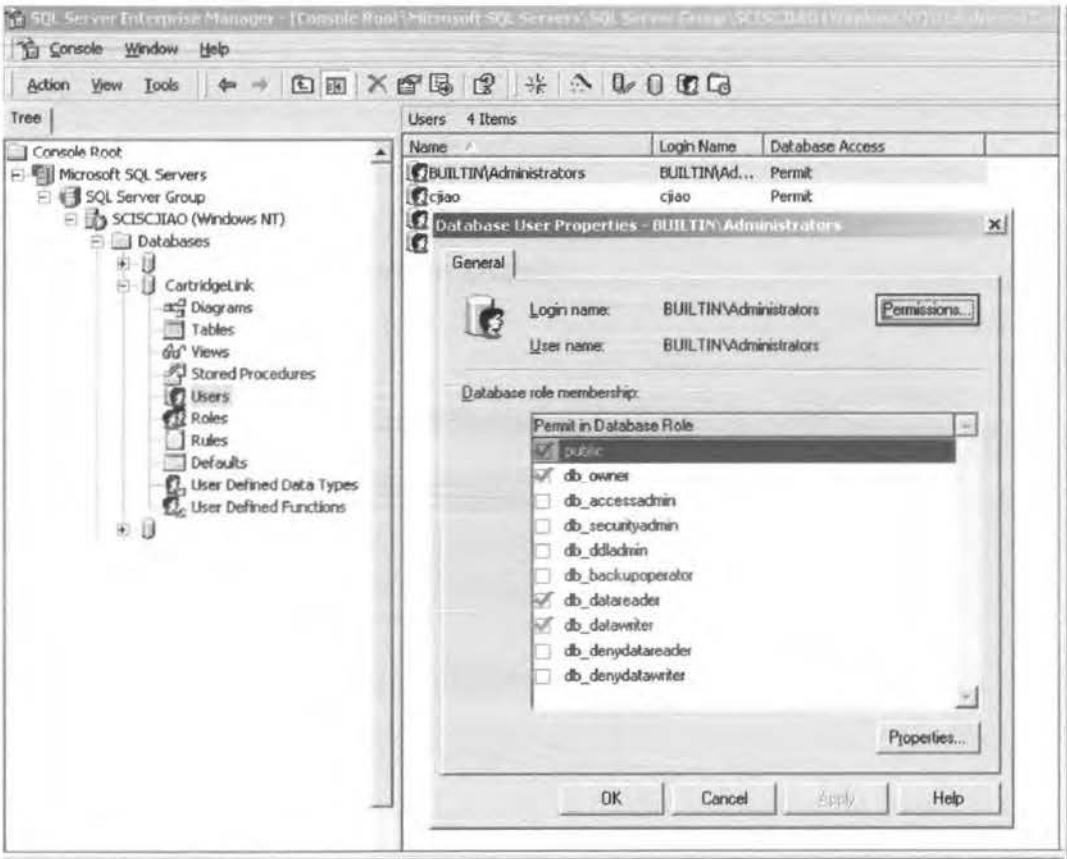


Figure 5.4 User List Pan and User Properties Window

User account	Activity
Administrators	All database access
cjiao	Create databases
dbo	Full access to cartridge data
guest	Read-only access to cartridge information

Table 16 Users and the Privileges

5.3.2 ASP and Database Objects

CartridgeLink uses ASP pages to access MSS2000's objects. These objects are provided by SQLOLEDB allowing ADO to access MSS2000, and supports objects listed in Table 14.

Table 13 lists the database objects the MSS2000 provided to ASP pages via VB script.

SQLOLEDB Objects	Description
Command Object	Command object is used to query a database and return records in a Recordset object, to execute a bulk operation, or to manipulate the structure of a database
Connection Object	A Connection object equivalent to an actual network connection to the server.
Recordset Object	Recordset objects is used to manipulate data from the provider
Record Object	Similar to the recordset, but for one row (record) usage
Stream Object	Stream object can be used to manipulate fields or records containing binary streams

Table 17 Database Objects

5.4 *Testing Output and Results*

This section focuses on the algorithms and outputs from the prototyped methods listed below:

- Logging into CartridgeLink
- Upload Cartridge into database
- Search cartridge from database

5.4.1 **Logging Method**

5.4.1.1 Method and algorithm

The logging security designed in chapter 4 is implemented in different layers of the system: namely,

- Windows user logging (supported by local system administrator)
- Intranet user logging support;
- DBMS user logging support; and
- Application user logging support.

In keeping with Intranet technology, the security that the DBMS has implemented for this study is affected by both logging into the Windows and the Intranet. Synchronisation is necessary between the three layers of the securities constructed for this application. Figure 5.5 shows how a user logging on to the Intranet CartridgeLink from the right beginning of a Window NT station.

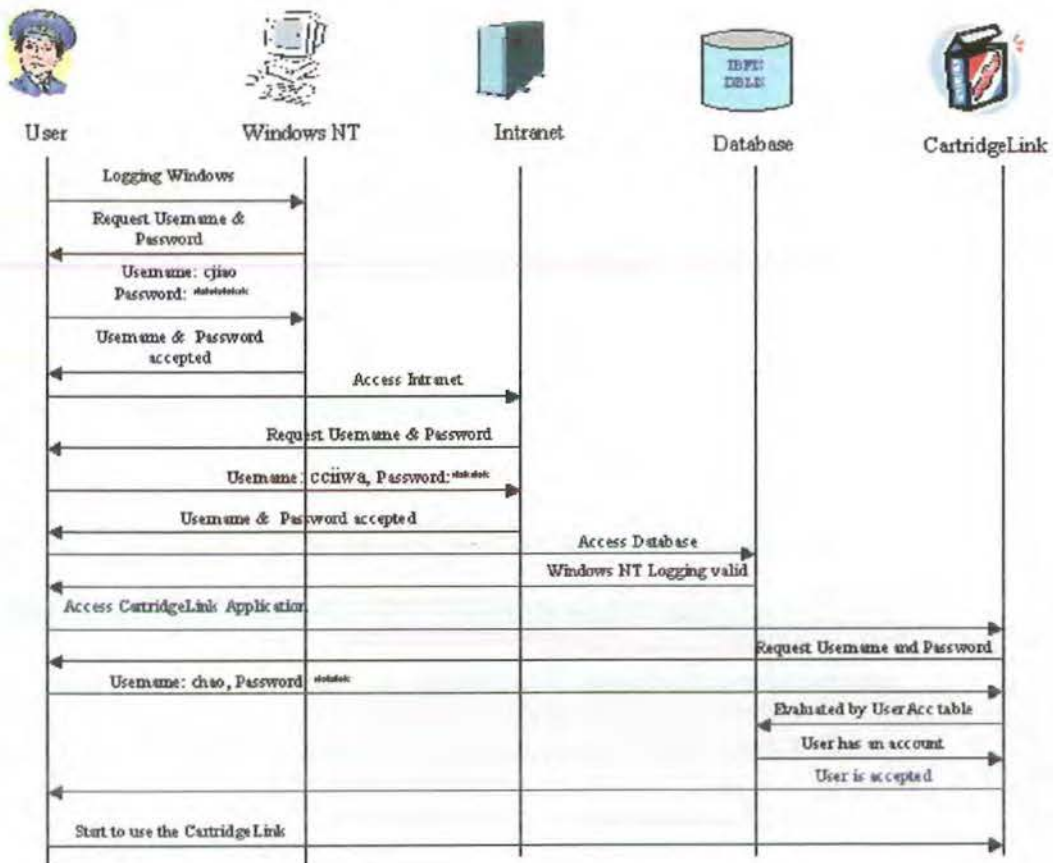


Figure 5.5 User logging process from a Windows NT to the CartridgeLink

Once the four party logging securities have been set up and synchronised, while each successive pass of the layers is sampled with reference to its security evaluation, the given privilege will be fetched from the DBMS and the user can perform tasks accordingly. Clearly, for this technique to work, the DBMS will need to have the ability to accept various authentic methods:

- Window authentic – default the users windows logging as his database logging
- SQL authentic – also named mix mode authentic, user will need to have a separate logging for the database
- Application authentic – after a user is accepted to use the database, his application logging will be evaluated according to table UserAccount described in Section 5.4.1.2.

It is clear that the SQL authentic method introduced complexity by asking an extra logging. However it dose provide better security by preventing somebody else using the machine to access the system after the machine was logged on. This method of authentication is also useful when multiple users share the same machine but do not share the same privilege usage of the CartridgeLink. Figure 5.6 shows a Mix mode (SQL Authentic) incorporated logging process.

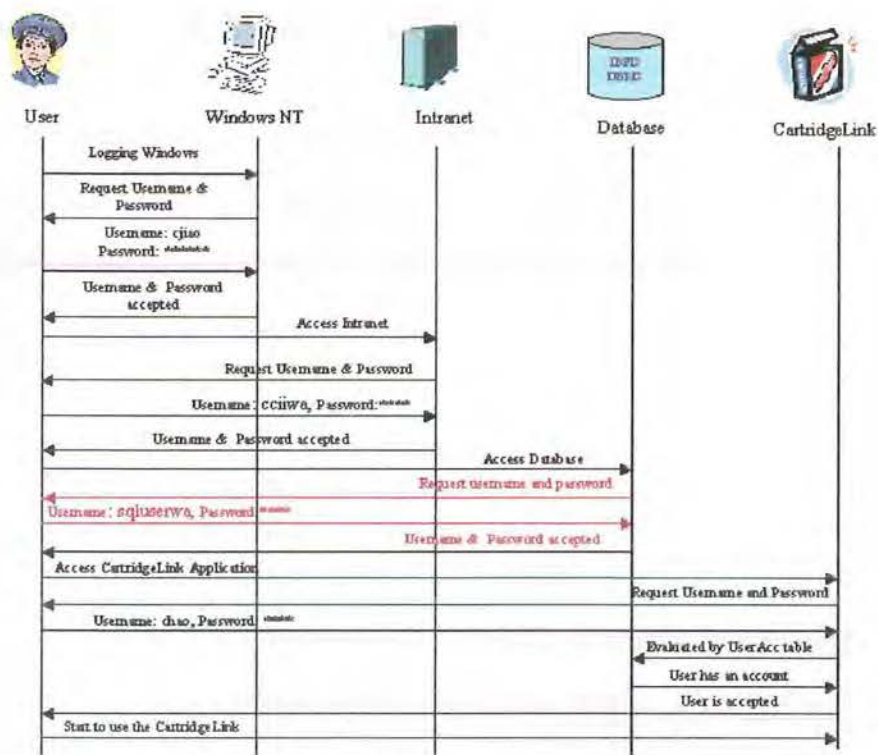


Figure 5.6 Extra Logging Layer Requested by Mix Authentic Mode

To protect the legal usage to the application, an Application Authentic layer is built and shown in both Figure 5.5 and 5.6. For the purpose of this study, this type of authentic is attractive as its logging process may be more complicated and a registration process is incurred. Importantly, such complexity may be seen in a real system: thus, permitting the assessment of a database and then an Intranet application with different modes of authentication can be understood. Put is another way, concurrent with the assessment of an Intranet database; the DBMS may usefully be implemented in securing the information contained in the database.

The basic logging implementation in this study, for convenience, was implemented using Windows authentic mode. For the reader's convenience, the algorithm for application authentic appears in Figure 5.7.

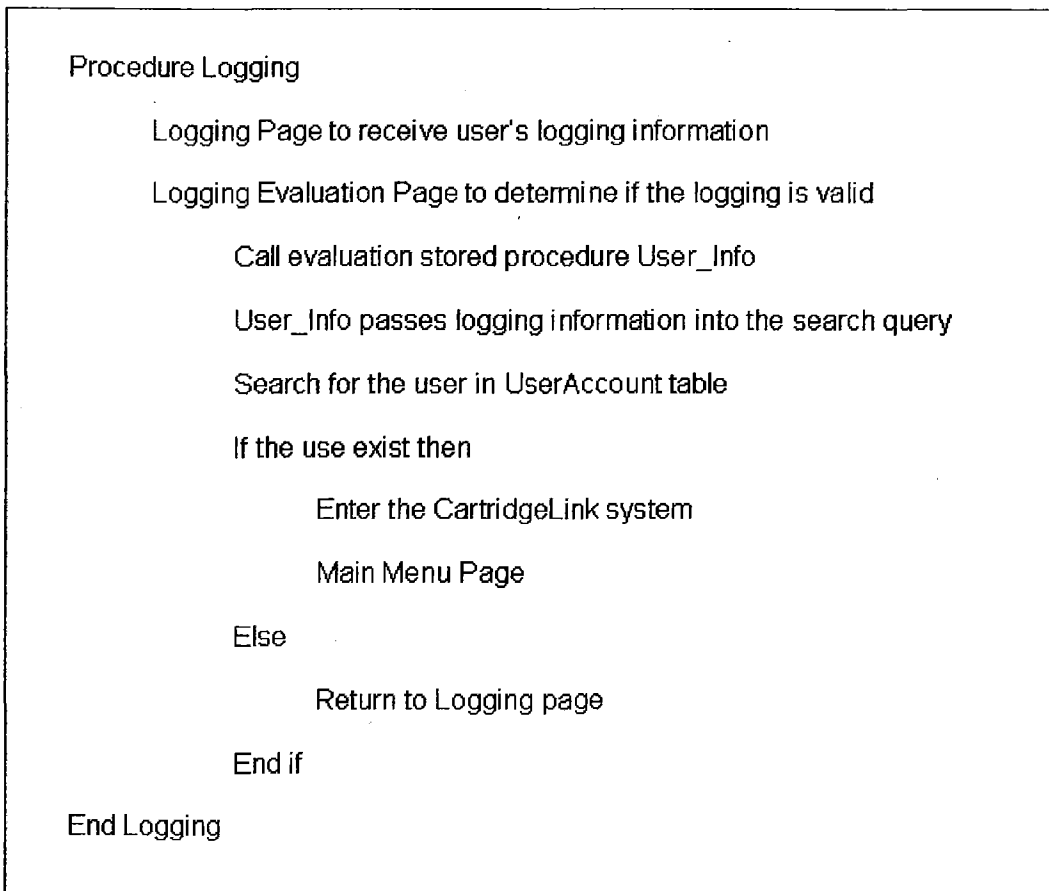


Figure 5.7 Logging Algorithms

As we designed in Chapter 4, new users should gain the access to the application by filling a registration form over the Intranet. Figure 5.8 shows the implement algorithm for the registration.

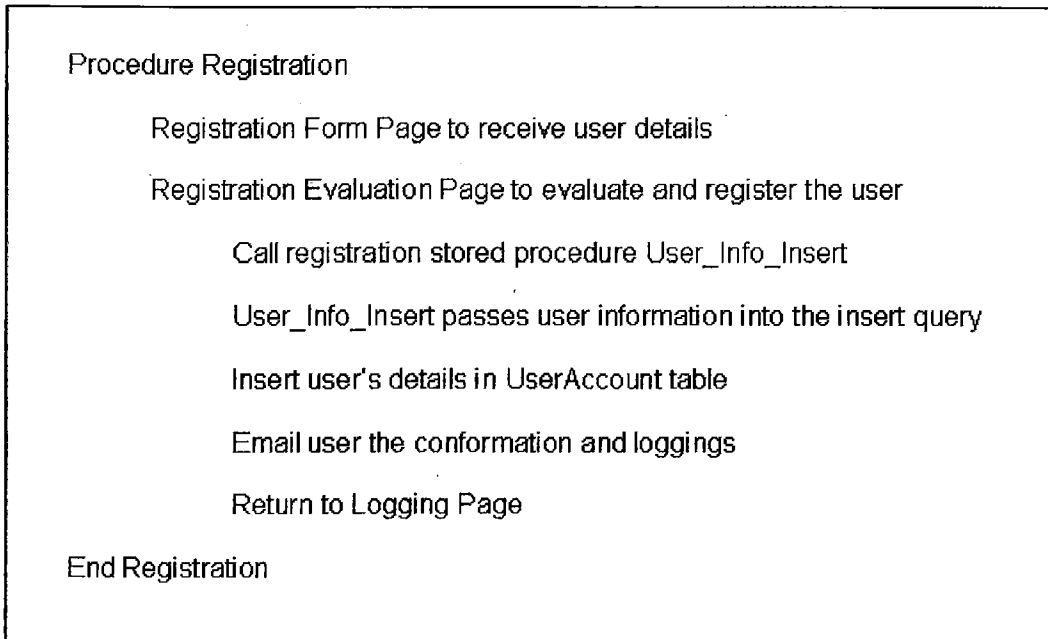


Figure 5.8 Registration Algorithms

The results of the implementation of the logging method meet the requirement of the design shown in Chapter 4 Figure 4.7, 4.8 and 4.9.

5.4.1.2 Test data and resultant productions

Two different sets of test data are presented, together with their output from the implementation for logging on the CartridgeLink system. In each case the DBMS security accounts has been used to evaluate first weather the user is authorized to access the database and, second then the *UserAccount* table is used to evaluate weather the user is authorized to user the application.

Testing data 1:

Username: *administrators*

Password: *cartlink*

Figure 5.9 and 5.10 shows testing data 1 - the logging and the confirmation by offering the main menu of the application.



Figure 5.9 Shows an administrator's logging

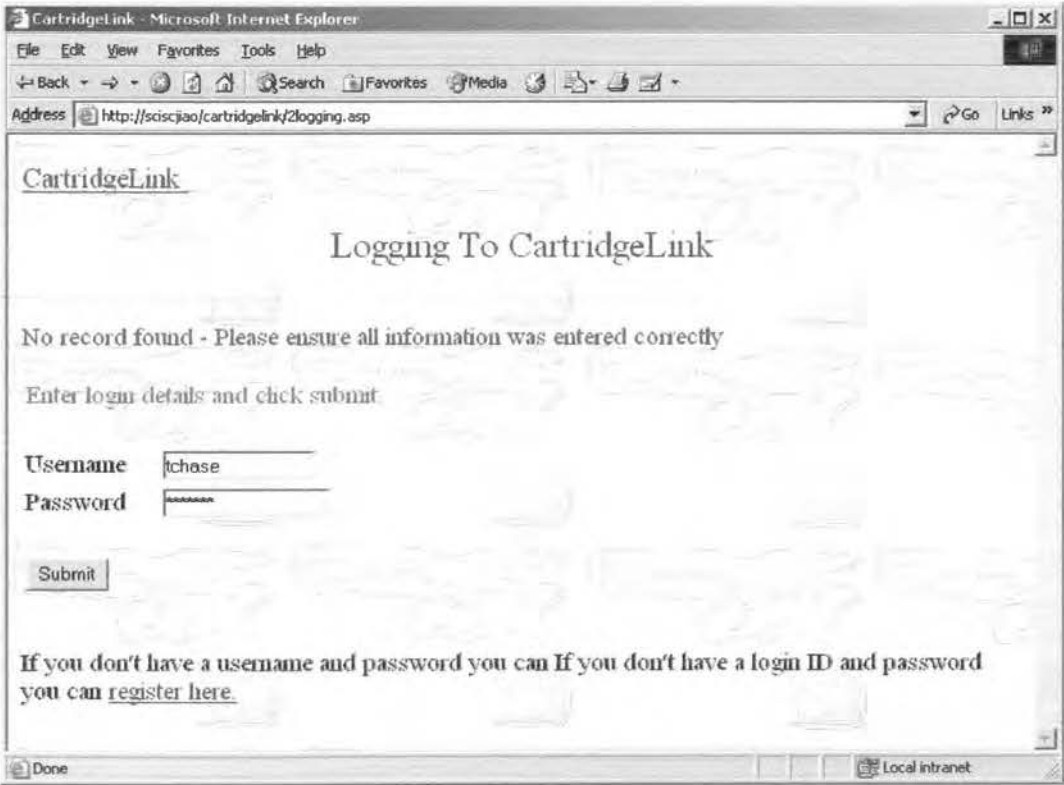


Figure 5.11 Rejected logging

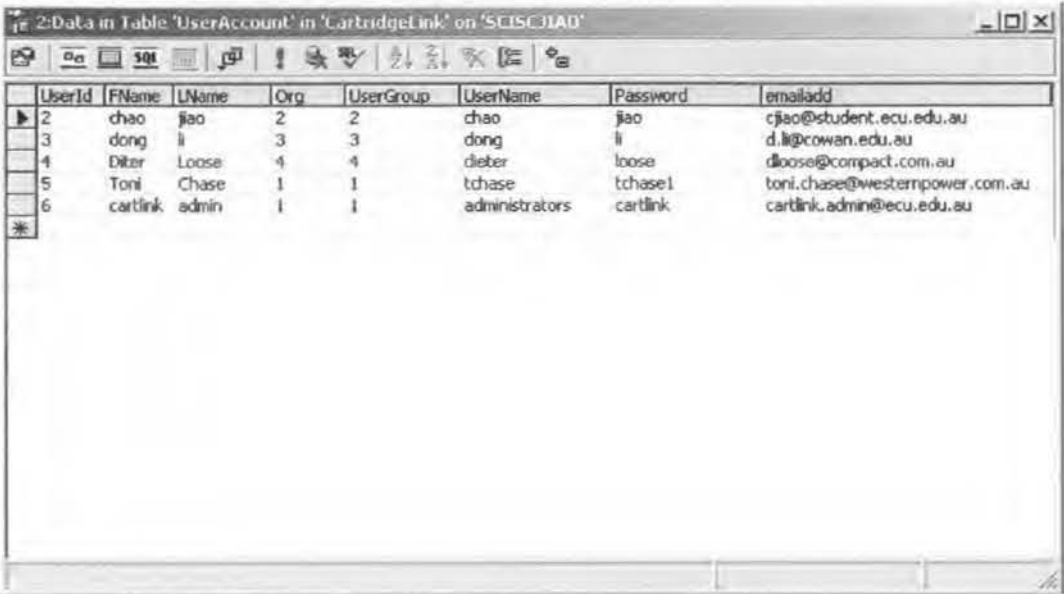
The screenshot shows a Microsoft Internet Explorer window titled "CartridgeLink - Microsoft Internet Explorer". The address bar displays "http://sciscjao/cartridgelink/register.asp". The page content includes the "CartridgeLink" logo and the heading "User Registration". Below this is a "Registration Form" with the following fields and options:

- First Name:** Text input field containing "Toni".
- Last Name:** Text input field containing "Chase".
- Email:** Text input field containing "cjiao@studentecu.edu.au".
- Username:** Text input field containing "cjiao".
- Password:** Text input field with masked characters (dots).
- User Group:** Radio button selection with three options: "ECU CartridgeLink Support" (selected), "Level 1 - View Only", and "Level 2 - Normal".
- Level 3 - Advanced:** Radio button option.
- Organisation:** Dropdown menu showing "Edith Cowan".
- Submit:** A button to submit the registration form.

The status bar at the bottom of the browser window shows "Done" and "Local intranet".

Figure 5.12 New user registration

Figure 5.13 shows the current records in the table UserAccount.



UserId	FName	LName	Org	UserGroup	UserName	Password	emailadd
2	chao	jiao	2	2	chao	jiao	cjiao@student.ecu.edu.au
3	dong	li	3	3	dong	li	d.li@cowan.edu.au
4	Diter	Loose	4	4	dieber	loose	dloose@compact.com.au
5	Toni	Chase	1	1	tchase	tchase1	toni.chase@westernpower.com.au
6	cartlink	admin	1	1	administrators	cartlink	cartlink.admin@ecu.edu.au

Figure 5.13 UserAccount table contains

5.4.2 Uploading Method

5.4.2.1 Algorithm

Recall for chapter two, relying upon a file-oriented image processing of Besser’s (Besser, 2002) “longevity, inaccessibility and inability”, that it may be possible to shorten a image’s life with the contribution of “disappearing information, viewing problem, scrambling problem, inter-relation problem, custodial problem, and translation problem.” Example was given, in chapter three, showing the cartridge headstamp image is missing from rest of its data contained in a database. To reduce and avoid this situation, an implementation is presented whereby image made from a

cartridge is uploaded into the database together with its classifications.

Figure 5.14 shows the algorithm used to upload a cartridge into the CartridgeLink database.

Procedure UploadCartridge

InputACartridgeCase Page prompt user inputs

OpenRecordsetProcedure open table Cartridge as a recordset

Add a new record

Receive inputs of cartridge text details

Receive image file directory by browse its file directory

Receive image by Uploading Image into HeadstampImage field

Update recordset

Close recordset

Options

View cartridge case list

Return to Main Menu

End UploadCartridge

Figure 5.14 UploadCartridge Algorithm

5.4.2.2 Test data and resultant productions

Test data for the upload method are listed in Table 5.3 and the results are shown in Figure 5.15.

5 IMPLEMENTATION AND RESULTS

CartridgeName	CalibreName	Manufacturer	Type	OrgName	CategoryDesc	Markings	Notes	ImageFileDir
22 Long Rifle	22 Calibre	<NULL>	Brass	Edith Cowan	Crime scene recovered cartridge case	7	DQ for notes to be input for review.	c:\cartlink\headstamp\fbhs1.bmp
17 Bee	17 Calibre	Beikal	Brass	Western Australia Police	Laboratory re-produced cartridge case	Test 2 by DQ	Check	c:\cartlink\headstamp\fbhs2.bmp
25 Auto	25 Calibre	Bertram	Plastic	South Australia Police	Laboratory re-produced cartridge case	45	Note.	c:\cartlink\headstamp\fbhs3.bmp
256 Winchester Magnum	25 Calibre	CBC	Brass	National Institute Forensic Science	Crime scene recovered cartridge case	1 C 1	ICI, 'CIVIC' Model .22 long rifle, standard velocity, lead non coated round nose projectile fitted, brass cartridge case, one cannellure 0.56" from base.	c:\cartlink\headstamp\fbhs4.bmp
405 Winchester	40 Calibre	CCI	Plastic	Edith Cowan	Crime scene recovered cartridge case	CHEDDITE - 12 +	Small brass head, blue colour draw ribbed plastic husk.	c:\cartlink\headstamp\fbhs5.bmp
7.35mm Italian	30 Calibre	Bertram	Plastic	Western Australia Police	Laboratory re-produced cartridge case	FIOCCHI 12 ITALY 12	Low brass head, black colour extruded ribbed plastic husk, shotsize marked on side wall, model "GAMELOAD" marked on side wall. Components assembled in Australia by BALLANTYNE.	c:\cartlink\headstamp\fbhs6.bmp
12 Gauge Shotgun	72 Calibre	Super	Plastic	Western Australia Police	Crime scene recovered cartridge case	12-SOFIM	Red Ribbed plastic case walls with six star crimp.	c:\cartlink\headstamp\fbhs7.bmp

Table 18 Test Cartridge Data

Inserting Cartridge Casing Data into Database

[Show the List of Cartridges](#) [Return To Main](#)

Enter Cartridge Case Classifications

Calibre	72	Category	Crm
Manufacture	Super	Cartridge Name	20 Gauge Shotgun
Composition Type	PLS	Provided By	Western Australia Police

Enter Headstamp Markings, Image and Notes

Markings 12-SOFIM |

Notes Red Ribbed plastic case walls with six star crimp. |

File C:\My Pictures\Imagel.gif | Browse... |


Image Submit |

Figure 5.15 Uploading image into database

5.4.3 Search Method

5.4.3.1 Algorithm

After collecting sufficient amount of the cartridge data, the permitted users can now require the system to perform the searches for specific cartridge(s). The searched parameters were carried by stored procedure *cartridge_info* back to the database. The program then queries the database, and brings back searched results. The searched cartridge may be a cartridge recovered from a crime scene or produced from a recovered firearm depending on which category of *crm* and *lab* was specified. The search result may be a single cartridge or a list of cartridges; in whichever case the cartridge headstamp may be viewed by click the highlighted *HeadstampIDs*. Figure 5.16 is the algorithm of the method and Figure 5.17 show a tested SearchCriteria page and its result.

Procedure SearchCartridge

SearchCriteriaPage to receive searched cartridge details

SearchResultPage to searches the requested cartridge or cartridges

Call stored procedure cartridge_Info

Cartridge _Info passes search criteria into the search query

Cartridge _Info brings back a cartridge or a list of cartridges

Click each highlighted *HeadstampID* to view the images

GoTo SearchCriteriaPage or press *Return Main* button

End SearchCartridge

Figure 5.16 Algorithm of Search Method

5.4.3.2 Result

CartridgeLink - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print Mail

Address <http://scischiag/firearmcartridgelink/Search.asp> Go Links >>

CartridgeLink - Search a cartridge

Enter Search details and click submit.

Calibre

Manufacture

Composition Type

Provided By

Category

Search Result

Headstamp ID	Markings	Cartridge Name	Notes
10	WINCHESTER 12 GA	12 Gauge Shotgun	Low brass head, black colour etruded ribbed plastic husk, shotsize marked on side wall, model 'GAMELOAD' marked on side wall. Components assembled in Australia by BALLANTYNE.
10	WINCHESTER 12 GA	12 Gauge Shotgun	Check
10	WINCHESTER 12 GA	12 Gauge Shotgun	Small brass head, blue colour draw ribbed plastic husk.
4	WINCHESTER 12 GA	12 Gauge Shotgun	ICI, 'CIVIC' Model .22 long rifle, standard velocity, lead non coated round nose projectile fitted, brass cartridge case, one cannalure 0.55" from base.

Number of records : 4, Table Name : CartridgeHeadStamp

Local intranet

Figure 5.17 Result of Search and Search Results

5.5 Performance Tests

Observations for performances of above three methods are recorded in Table 5.4. The times were measured using SQL Profiler.

5 IMPLEMENTATION AND RESULTS

Even Class	Duration	Text Data	Application Name	Start Time	SPID
RPC:Completed	10	exec dbo.user_info 'brain', 'jones', 'WAP'	Microsoft(R) Windows (R) 2000 Operating System	2002-09-22 23:33:56.973	57
RPC:Completed	10	exec dbo.dt_verstamp006	MS SQLEM	2002-09-22 23:34:35.170	51
RPC:Completed	10	exec dbo.user_info 'brain', 'jones', 'WAP'	Microsoft(R) Windows (R) 2000 Operating System	2002-09-22 23:39:48.753	57
RPC:Completed	10	exec group_info	Microsoft(R) Windows (R) 2000 Operating System	2002-09-22 23:39:51.720	58
RPC:Completed	10	exec dbo.user_info 'brian', 'jones', 'wap'	Microsoft(R) Windows (R) 2000 Operating System	2002-09-22 23:43:11.397	58
RPC:Completed	10	exec dbo.user_info 'brian', 'jones', 'WAP'	Microsoft(R) Windows (R) 2000 Operating System	2002-09-22 23:43:23.513	58
RPC:Completed	10	exec dbo.user_info 'brain', 'jones', 'WAP'	Microsoft(R) Windows (R) 2000 Operating System	2002-09-22 23:43:55.820	58
RPC:Completed	30	exec sp_MShelpcolumns N'dbo.People', NULL	MS SQLEM	2002-09-22 23:40:47.910	51
RPC:Completed	40	exec sp_MShelpcolumns N'dbo.Org', NULL	MS SQLEM	2002-09-22 23:35:14.567	51
RPC:Completed	40	exec group_info	Microsoft(R) Windows (R) 2000 Operating System	2002-09-22 23:36:48.323	55
RPC:Completed	50	exec sp_tables N'People', N'dbo',	MS SQLEM	2002-09-22 23:34:35.230	51

		NULL, NULL			
RPC:Completed	60	exec user_info_insert 'brain', 'jones', '1/1/67', 1, 'WAP', 1, 'wa'	Microsoft(R) Windows (R) 2000 Operating System	2002-09-22 23:42:38.910	58
RPC:Completed	70	exec group_info	Microsoft(R) Windows (R) 2000 Operating System	2002-09-22 23:32:02.570	58
RPC:Completed	100	exec dbo.user_info 'brain', 'jones', 'wap'	Microsoft(R) Windows (R) 2000 Operating System	2002-09-22 23:31:55.770	57
RPC:Completed	590	exec sp_MShelpcolumns N'dbo.People', NULL	MS SQLEM	2002-09-22 23:34:35.300	51

Table 19 Performances of the Methods

5.6 Basic Capacity Scalability Testing

As we have mentioned in Chapter 2, due to the limitation of a small testing database, the capacity and performance of the real system is not available in this instant. However, an existing testing result provided by the vendor is quoted as a reference for the audience who are interested in the respects.

Microsoft describes that MSS “running on Microsoft Windows 2000 servers handles the largest mission-critical applications” (Xiong, 2001). In the same white paper, which contains a joint effort made by Microsoft and Dell, the testing results demonstrate the scalability of SQL Server 2000 is as good

as terabytes and the transaction speed is as high as 1000 GB per second.

Running SQL Server 2000 on a Dell enterprise eight-way server can support multiple-terabyte databases and handle heavy workload and administrative tasks, producing the best results. It is said that SQL Server 2000 can maximizes your return on investment in symmetric multiprocessing (SMP) systems, allowing users to add processors, memory, and disks to build large, centrally managed enterprise servers. Figure 5.18 shows transaction speed based on difference number of CPUs used in the system and figure 5.19 shows it based on difference GB of memory used in the system.

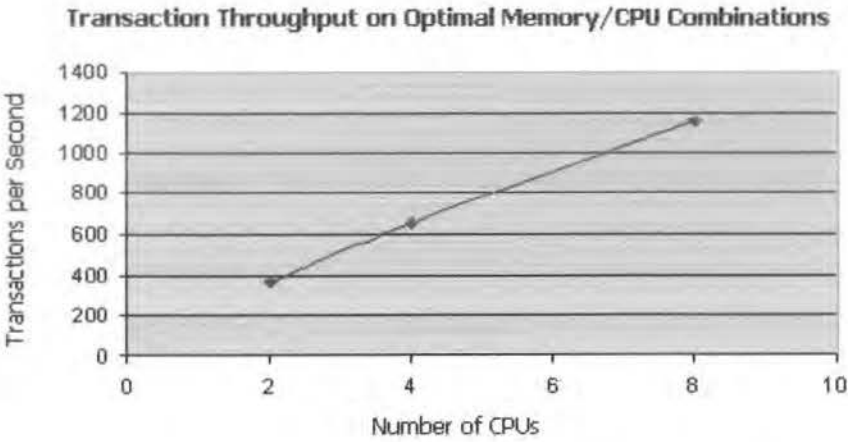


Figure 5.18 Transaction Speed based on Number of CPUs

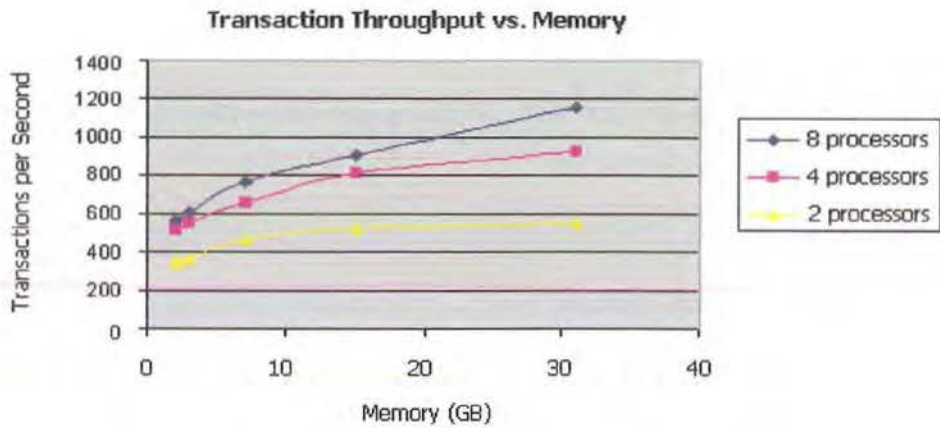


Figure 5.19 Transaction Speed based on GB of Memory

It is clear that the more investment made to the hardware, the better performance received by the users.

5.7 Summary

This chapter focuses on the implementations and outputs from the prototyped methods of the system. The database design in the previous chapter has been implemented. The three scenarios developed in the previous chapter have each been presented in terms of implementation, respective productions of algorithm and appropriate test data. The performances of the methods have been sampled and presented in SQL Profile testing form.

6 CONCLUSION AND FUTURE WORK

This chapter identifies the questions raised by the thesis and provides a discussion to each question. A discussion is given for other important topics. This is then followed by a conclusion and possible future work.

6.1 *Question 1*

What are the problems associated with intraneting firearm identification system CartridgeLink? OR: What are the problems associated with implementing CartridgeLink Intranet firearm identification system?

There are many different problems associated with implementing Intranet firearm identification system CartridgeLink. Below is a sample of some of the problems encountered and overcome during this project:

6.1.1 **Question 1.1**

What are main tools/software required to develop an Intranet firearm identification application as CartridgeLink?

Answer: It is found that the tools required for the requirement include IIS (Internet Information Service), Intranet Server programming language and a DBMS.

6.1.2 Question 1.2

What type of DBMS is suitable for managing Intranet ballistics database with cartridge case images?

Answer: A range of DBMS types/modules can be used for implementing Intranet ballistics database with images. The most effective one found was a combination of the OODBMS and RDBMS.

6.1.3 Question 1.3

How do you access ORDBMS with ASP Intranet programming language?

Answer: Through a set of ORDBMS objects.

6.1.4 Question 1.4

What kind of image storage method is suitable for Intranet ballistics firearm identification application like CartridgeLink?

Answer: Two major methods can be used for store/collecting images.

The effective one found was a stored the images together with their classification/metadata in the database.

6.1.5 Question 1.5

How do you implement images in the database for the Intranet application with only a field in a record such as a cartridge case record? How can you improve the applications overall performance?

NOTE: This question is answered by section 6.3

Question 3 and section 6.4 Question 4.

6.1.6 Question 1.6

What are some possible solutions for securing the database and the system?

NOTE: This question is answered by section 6.2
Question 2.

6.2 Question 2

What are some possible logging solutions to securing an intelligent database like ballistic database?

Answer: A ballistics database can often be insecure as it collects forensic information that is important to law enforcement. Implementing user accounts and passwords for each user or user group can secure the database and thereby give a reliable data resource. This can be effective when the system is only preventing undesired users from accessing its database.

Most mainstream DBMSs provide a security system that allows administrators to specify and assign different levels of privileges to different users and user groups. This is a useful facility to protect valuable resources against unauthorized access to the database.

Providing different layers of logging can also help to reduce the risk of unauthorized access to the application. For example, this project created an application security layer, on top of the security layers for Intranet and database, that allows individual logging of the layer to be accepted or rejected depending on the corresponding record in the database. The logging interfaces allow the user to specify himself to the windows, the Intranet, the DBMS and last the application. Providing different levels of logging is a useful tool for information security. Ideally the DBMS would use a detailed privileges configuration methodology in which each individual is specifically given privileges for read or write, for which table and which piece of information. This would improve both the reliability and security of the system. User

groups can be produced for each of the state police laboratories and Courts separately. The design of the prototype was to allow the user to be administrated in specifications. Although not yet implemented, the prototype was to include a feature to allow the administrator to configure a security chart based on the discussions with all parties. This feature would be useful when the system goes live.

6.3 *Question 3*

How do we convert image files into image data type fields? E.g. how do we upload cartridge case image files into cartridge case records?

Answer: This was addressed by the thesis by first including a field called “HeadstampImage” into “CartridgeHeadstamp” table with data type “image”. The table’s “text in row” option was then turned on and set to “7000 bytes” by “sp_tableoption” system stored procedures. This field was assigned with a single pieces image binary data or a binary tree depending on if the image’s size was less or more then “7000” bytes.

The process of uploading an image (file form) from the local machine into the database server (field form) is as follows:

- Active the function - select uploading image option from the main menu;
- Input associate classifications - enter the classifications in prompted columns;
- Indicate where the image file is located and what type of image file it is -Specify the location of the image file in prompted column;
- Using a sub procedure “upl” and its method “SaveAsBlob” to save the image as a “BLOB” data into the “CartridgeHeadstamp” field - press the “Upload” Button to transfer the image file into database.

6.4 *Question 4*

How can we improve an Intranet base image database’s performance?

E.g. how can we improve CartridgeLink’s database’s performance?

This was addressed by the thesis by first stored the cartridge case images into the database instead of the files, as addressed in 6.3 Question 3. To further improve the performance of the Intranet cartridge case identification database, besides indexing and some other regular good DBMS practices, a set of stored procedures was pre-built in the database server.

To sequences of tasks using a stored procedure can be listed as below:

- Build the stored procedure with the parameter variables, SQL statements and conditions in the MSS2000 by saving it;
- Program it as part of a single execution plan on the server by providing the parameters in the same sequences as they are listed in variables and conditions in the stored procedure
- When the CartridgeLink is used by a client, the ASP programs pass user-entered parameters to the server and then the stored procedure to conduct the results;
- The results do not have to be returned to the client to have the conditional logic applied and all of the work is done on the server;

6.5 *Limitation*

The limitation set by choosing traditional DBMS with the support of an object-oriented interface. As it was described in 3.2.3.3, MDBMS development approaches, it is observed that pure object-oriented approaches would provide better reusability and effectiveness.

The impact of this limitation with saving a large size image (over the limitation as one piece binary data) into database seemed to affect the performance. For example, if a user transferred a 5 MB image into the

database, the pointers for binary trees will be introduced automatically. A solution to this problem would be to develop a program that converts the user inputted image into a standard size which can be fitted in the field as one pieces. If its large size version is needed, it can be saved in a separated table with its ID link back to the original cartridge table.

6.6 *Benefits*

6.6.1 Information Availability

Intranet firearm identification database like CartridgeLink improve the law enforcement by improving the chances to identify firearms over a possible national wide DBMS. The system required no geometric or organizational limitation within the law system. It is available over an Intranet provided for laboratories. The database was administrated and updated over the net. The search for a suspected firearm via its cartridge case can be made from large scale of data contributed by other states. The potential link between crimes may be discovered effectively hence enhances law enforcement.

6.6.2 Save of Time

During the original firearm identification process, all of external enquired cartridge information was made out on mail or email systems. Received information would need to treat before the comparisons. This is very time consuming. With a Intranet Firearm Identification Cartridge Database and application, once a user gain the access, the needed images and data are automatically available for the identification. This streamlines the process and saves time.

6.6.3 Reduce Redundant Data

Redundant data not only wastes time and resources, but also causes confusions and misunderstandings. By using an integrated Intranet database, the same firearm fires with the same bullet need only to be produced only. This production can than be used by other laboratories is it is needed. All we need to do is make sure this piece of information is correct and updated. This provides data integrity for the users.

6.6.4 Security

Compare with current system's database security, this project provides a integrated security facility between OS, Intranet, Database administration and Application Usage Authorization. When it is implemented into the live system

with other securities components such as firewall, the Intranet firearm identification system would be a secured system over the web.

6.7 *Conclusion*

Computerized Ballistics Firearm identification is a topic that is increasingly interested, as it is comparably new to Australian computerized forensic science. Currently there is a computerized firearm identification system used by some police departments and supports local state laboratories in identifying firearms with local DBMSs. This study has demonstrated the design and implementation of an Intranet firearm cartridge identification system that is capable of operating an Intranet based image included DBMS. The prospect of the study provides the possibility to link forensic laboratories with an Intranet DBMS and share data resource to allow effective firearm identifications, especially linked crimes identifications.

Other points of interest include:

- The ability to providing an Intranet database server for the Intranet users was proven. For example, in this project cartridge case records were inputted and saved into the database and late the database is searched for giving criteria.

- All image files produced were translated into database quickly and correctly and all the enquiries were responded quickly and correctly.
- All images generated from the database produced a high quality output.
- All images produced required NO external location information. This was an important problem to solve as was addressed.
- All the user entries were evaluated correctly in all security layers.
- The prototype demonstrated that creating an Intranet firearm identification system is a highly effective way of working as a law enforcement network.
- It is possible to hide excess information through various techniques such as collapsing diagram components.
- It is possible to querying the database with SQL XML statement and stored procedures using URL box in the browser. This is particular useful for the database administrators.
- The majority of the design in this thesis was implemented through the prototype.
- Initial thesis questions were answered.

6.8 *Future Work*

The prototype was successful in using CartridgeLink DBMS from Intranet. The data collection and search for the cartridge cases were performed quickly. After using the prototype it is clear that this is a far more effective way of identifying cartridges than traditional methods. As this was only a prototype future work could be done to increase the capability, functionality and security of the prototype. Some of this improvement could included:

- Development of completed database to include all necessary information, such as Firearm, Firearm's Cartridge Casing etc, to enhance the identification. This would further increase the capability of the system.
- Automation of producing cartridge case images base on collected firearms and collected brands of bullets. This would be an extremely effective way of generating master cartridge case images over the traditional approach of firing the bullets and producing images from fired bullets, which would be time consuming and inconvenient, especially when both firearms and bullets are not physically available at the site. This would further increase the functionality of the system.
- Automatically narrow down a list of suspected cartridge casing when a newly recovered cartridge case is been uploading and

classifying into the system. This would allow the users to quickly receive the short list of cartridge casing without having to search the database separately. This would improve the effectiveness of the system

- Enhance the security of the system applying other type of security measures, such as encoding/decoding, besides the passwords.
- Develop a completed user account registration procedure so that the registration made by users can be properly evaluated, administrated and notified. This would further improve the security of the system.
- Develop a transformation procedure for desktop users to transfer the whole or part of database into their Fireball database. This would allow the field or isolated users to regularly refreshing their local Fireball database with latest information.
- Evaluated the prototype by implementing it with other DBMS and programming tools, such as Oracle DBMS with Java. This would provide options for further development.

REFERENCES

- America Bureau of Alcohol Tobacco and Firearms of Department of the Treasury (ATF). (2002). The Missing Link: Ballistics Technology That Helps Solve Crimes. *ATFNews*, August 2002(28).
- Anstey, D. (1999). *Oracle 8*. Albany, NY: Coriolis Group Books.
- ATF, B. O. A., Tobacco And Firearms. (2002). Crime Gun Trace Reports 2000.
- Australia Broadcasting Corporation Television, A. (2001). *Fireball* [Television]. Ian Watson, Naomi Lumsdaine. Retrieved 22th March, 2001, from the World Wide Web:
<http://www.abc.net.au/quantum/s259929.htm>
- Besser, H. (2002). *Image Metadata: Important Recent Activities*. UCLA School of Education & Information. Retrieved 01/07/02, from the World Wide Web:
- Butterfield, F. (2001). Gun Control Laws Concerning The Mentally Ill Are Faulted. *The New York Times*, pp. 1.30.
- Celens, E., & Demanet, F. (1989). *Basic Principles Of Weapon Identification Techniques*. Paper presented at the 11th International Symposium on Ballistics, Brussels Congress Centre, Brussels, Belgium.
- Chamberlin, D. (1998). *A Complete Guide to DB2 Universal Database*: Morgan Kaufmann Publishers.

- Clearform, C. S. I. (Artist). (2002). *Microsoft Access Database Design*
[Software].
- Codd, E. F. (1990). *The Relational Model for Database Management* (2 ed.).
Reading, Massachusetts: Addison-Wesley Publishing Company.
- Crigler, J. B., & Orooji, A. (1999, October 25-26, 1999). *ORR: Object-
Relational Rapprochement*. Paper presented at the Twenty-Third
Annual International Computer Software and Applications
Conference, Phoenix, Arizona.
- Castelli, V. and L. D. Bergman, Eds. (2002). *Image Database*. New York,
Wiley.
- Date, C. J. (1993). *Database Systems*: Addison-Wesley Publishing Company.
- Date, C. J. (1995). *An Introduction to Database Systems* (6 ed. Vol. 1).
Reading, Massachusetts: Addison-Wesley Publishing.
- Dendrinios, M. (1997, August 1997). *A Case Study for the Management of a
Set of Spectrally Analysed Signals
Based on OODBMS*. Paper presented at the International Database
Engineering & Applications Symposium, Montreal, Quebec.
- Doyle, J. E. A. G., Division of Law Enforcement Services. (2002). *Firearm
Evidence* [Internet]. Retrieved, from the World Wide Web:
<http://www.doj.state.wi.us/dles/crimelabs/Firearms.asp>
- Doyle, J. S. (2002). *Fundamental of Firearms ID*. Retrieved May 2001, from
the World Wide Web: <http://www.firearmsid.com>

- FBI. (2002, 13 Sep 2002). *FBI History - A time line*. Retrieved, from the World Wide Web:
<http://www.fbi.gov/libref/historic/history/historicdates.htm>
- Geradts, Z., Bijhold, J., & Hermsen, R. (1999). *Pattern recognition in a database of cartridge cases*. Paper presented at the SPIE.
- Hamby, J. E. (1999). The History of Firearm and Toolmark Identification. *Association of Firearm and Tool Mark Examiners Journal*, 31(3).
- Handerson, D. (1996). Add to the warehouse with OLAP tools. *Computing Canada*.
- IBIS. (2002). *Integrated Ballistics Identification System*. Retrieved, from the World Wide Web:
- Inbau, F. E. (1999). Firearms identification--"Ballistics". *Journal of Criminal Law & Criminology*, 49(4), 1293-1314.
- Kauffman, J. (2000). *ASP Database*. Birmingham: Wrox Press Ltd.
- Kroenke, d. M. (1992). *Database Processing*. New York: Macmillan Publishing.
- Li, D. G., & Chase, T. (2001). *Intranet Ballistic Image Database*. Paper presented at the ICII2001, Beijing.
- Li, D. G., & Watson, A. C. (1998a, 3-5 November 1998). *Ballistics firearms identifcaiton based on images of cartridge case and projectile*. Paper presented at the SPIE - The International Society for Optical Engineering, Boston, Massachusetts.

- McFadden, F. R., Joffer, J. A., & Prescott, M. B. (1999). *Modern Database Management*. Reading, Massachusetts: Addison-Wesley.
- Microsoft. (2000a). *Microsoft Access*. Retrieved, from the World Wide Web:
www.microsoft.com.au
- Microsoft. (2000b). SQL Server 2000 (pp. 303-336). Redmond, Washington.
- Microsoft. (2000c). *SQL Server Books On Line*. Microsoft. Retrieved, from
 the World Wide Web:
- Microsoft. (2001). *SQL Server 2000 Web-Enabled Features* [Internet].
 Retrieved, from the World Wide Web: www.microsoft.com
- Microsoft. (2002). Internet Explorer (pp. 303-336). Redmond, Washington.
- Mixdorf, M. A., & Goldsworthy, R. E. (1996). A history of computers and
 computerized imaging. *Radiologic Technology*, 85.
- Moran, W. E. (2000). *DRUGFIRE and IBIS*. Retrieved, from the World Wide
 Web: http://www.adfs.state.al.us/adfs/articles/d_fire.htm
- Mouzos, J., & Carcach, C. (2001). *Weapon Involvement in Armed Robbery*
 (Vol. 38). Canberra: Australia Institute of Criminology.
- Netscape. (2002). *Netscape Products*. Retrieved, from the World Wide Web:
<http://wp.netscape.com/comprod/>
- NIFS. (2002). *What is Forensic Scienc*. Retrieved, from the World Wide Web:
www.nifs.com.au/F_S_A/FSA_frame.html?about.asp&1
- Papaglannis, G. (2000). *Firearms - Ballistics Examinations in Hellas*
Firearms and Toolmarks Laboratory. Paper presented at the

Regional Symposium on Criminalistics, Istanbul University, Institute of Forensic Sciences, Turkey.

Ramakrishnan, R. (1998). *Database Management Systems*. Boston: WCB/McGraw-Hill.

Santini, S. (2001). *Exploratory Image Databases: Content-Based Retrieval*. San Diego, Academic Press.

Shapiro, J. R. (2001). SQL Server 2000: The Complete Reference (pp. 760-789, 133). Berkeley: Osborne.

Shih, T. K., Ed. (2002). *Distributed Multimedia Databases: Techniques & Applications*. Distributed Multimedia Databases. Hershey, Idea Group Publishing

Smith, C. L., Cross, J. M., & Variyan, G. J. (1995). Fireball: An Interactive Database for the Forensic Ballistics Identification of Firearms. *Research Report, 1995*(October).

Songini, M. L. (2002). Collections of data. *Computerworld*, 36(16), 46.

Tartaro, J. P. (2000). Ballistic Fingerprinting' Offers No Quick-Fix Solution. *The New Gun Week*, 37(1734).

United Nation. (1997). *Economic and Social Council*. Paper presented at the 36th Plenary Meeting.

APPENDIX A: A GLOSSARY OF TERMS USED IN THIS DOCUMENT

Extensible Markup Language	XML is the emerging Internet standard for data. XML is a set of tags that can be used to define the structure of a hypertext document. XML documents can be easily processed by the Hypertext Markup Language, which is the most important language for displaying Web pages.
Firearm Identification	The identification of fired bullets, cartridge cases or other ammunition components as having been fired from a specific firearm. commonly known as ballistics.
Home Page	The primary Web page for an individual or organization.
IIS5.0	Internet Information Services 5.0 (IIS) is the Windows 2000 Web service that makes it easy to publish information on your intranet for the Internet.
Impeachment Mismatch	The identification of fired bullets, cartridge cases or other ammunition components as having been fired from a specific firearm.
Intranet	Interconnected computers and networks within an organization.
Stored Procedure	A stored procedure is a group of Transact-SQL statements compiled into a single execution plan.

Structured Query Language

To work with data in a database, you have to use a set of commands and statements (language) defined by the DBMS software. Several different languages can be used with relational databases; the most common is SQL. The American National Standards Institute (ANSI) and the International Standards Organization (ISO) define software standards, including standards for the SQL language.

Content-based Retrieval

Users search the multimedia repository providing information about the actual contents of the image, audio, or video clip. A content-based search engine translates this information in some way as to query the database and retrieve the candidates that are more likely to satisfy the users' requests

BLOB

Binary Large Objects. Multimedia objects are binary large objects mostly (It is common that a video clip occupies more than 100 MB of disk storage). In SQL Server 2000, BLOB data includes data types of image, binary and varbinary.

APPENDIX B: ASP PAGES AND SOURCE CODE

Logging.asp

```

<!-------!>
<!-- This program is to let attempted users to enter their -----!>
<!-- loggings details or to register as a new user to the -----!>
<!-- application. If the program finds a valid cookie, it runs--!>
<!-- WelcomBack.asp to skip prompting logging screen, ---!>
<!-- otherwise if the LOGIN option is selected, program --!>
<!-- LoggingVerify.asp will be run to verify the username --!>
<!-- and password. If the REGISTRATION option is -----!>
<!-- selected, program Registration.asp will be run. -----!>
<!-------!>

<%
'--- Check to see if a cookie exists for this user ---
If Len(Request.Cookies("CartridgeLink")("Username")) > 0 Then
    '--- Cookie exists, authenticate the user for CartridgeLink ---
    Session("Authenticated") = True
    '--- Redirect the browser to the welcome back page ---
    Response.Redirect "WelcomeBack.asp"
End If
%>

<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
<TITLE>CartridgeLink</TITLE>
</HEAD>
<BODY background=images/bg.bmp>
<div align=left>
    <big><big><font color=navy><U><FONT
color=#8b0000><FONT size=4>CartridgeLink</FONT>&nbsp;</FONT></U>
</font></big></big>
</div>
<DIV align=center><BIG><BIG><FONT
color=navy></FONT></BIG></BIG>&nbsp;</DIV>
<DIV align=center><BIG><BIG><FONT color=navy>Logging To
CartridgeLink</FONT></BIG></BIG></DIV> <br><br>

<%
'--- Check for an error message which indicates the previous login attempt failed ---
If Len(Session("ErrorMessage")) > 0 Then

```

```

        Response.Write "<font color=red>" & Session("ErrorMessage") &
"</font><br>".
End If
%>

```

```

<!--Display login form-->
<form action=loggingverify.asp method=post name=frmDefault>
<table style="LEFT: 17px; TOP: 94px"><tr><td colspan=2><font color=teal>Enter
login details and click submit.</font></td></tr><tr><td>&nbsp;</td></tr>
    <tr>
        <td>Username</td>
        <td><input name=txtFirstName size=15 ></td>
    </tr>
    <tr>
        <td>Password</td>
        <td><INPUT id=password1
style="WIDTH: 124px; HEIGHT: 22px" type=password size=16
name=txtPassword></td>
    </tr>
    <tr>
        <td>&nbsp;</td>
    </tr>
<tr><td><input type=submit name=btnSubmit value=Submit></td></tr>
</table></form>

<br>
If you don't have a username and password you can
<A href="registrate.asp" >register here.</A>
</BODY>
</HTML>

```

LoggingVarify.asp

```

<!-------!>
<!-- This program verifies entered username and password --!>
<!-- against UserAccounts in the database CartridgeLink.----!>
<!-- If the login details are correct, the MainMenu.inc will --!>
<!-- be prompted for further operation; otherwise the --!>
<!-- Logging.asp will be run to prompted correct loggings. ---!>
<!-------!>

<!-- #include file="adovbs.inc" -->
<!-- #include file="Connect.inc" -->
<%
'--- Check for database errors ---
Call CheckForErrors(objConn)

'--- Verify user information in the database ---
Set objConn = Server.CreateObject("ADODB.Connection")
objConn.Open "driver={SQL Server};server=sciscjiao;database=cartridgelink","sa",
""

'--- Create the recordset object, set the SQL string and parameters and open the
recordset ---

Set objRS = Server.CreateObject("ADODB.Recordset")
strSQL = "dbo.user_info (" & CStr(Request.Form("txtUserName")) & _
        "," & CStr(Request.Form("txtPassword")) & ")"

objConn.Execute strSQL

objRS.Open strSQL, objConn, adOpenForwardOnly, , adCmdStoredProc

'--- Check for database errors ---
Call CheckForErrors(objConn)

'--- Check for empty recordset which indicates user information was not found ---
If objRS.EOF or objRS.BOF Then
    Session("ErrorMessage") = "No user found - Please ensure all information
                                was entered correctly"
    Response.Redirect "logging.asp"
Else
    Session("ErrorMessage") = Empty
End If
%>

<!-- #include file="Disconnect.inc" -->

```



```

<%
'--- Save the user information to a cookie ---
Response.Cookies("CartridgeLink")("UserName") = Request.Form("txtUserName")
Response.Cookies("CartridgeLink")("Password") = Request.Form("txtPassword")

'--- Set the expiration date of the cookie to the last day of the current year ---
Response.Cookies("CartridgeLink").Expires = "December 31, " & Year(Now)

'--- Authenticate the user for other web pages
Session("Authenticated") = True
%>

<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
<TITLE>CartridgeLink</TITLE>
</HEAD>
<BODY><!--Display the page data-->
<div align=center>
    <big><big><font color=navy>Ballistic CartridgeLink Intranet</font><FONT
color=#000000 size=3> </FONT></big></big>
</div>
<br><br><!-- #include file="MainMenu.inc" -->

</BODY>
</HTML>

```

WelcomeBack.asp

```

<!-------!>
<!-- This program catches entered username and password --!>
<!-- into cookies to remember the user to save repeat enter!>
<!-- the login details. The MainMenu.inc will be prompted --!>
<!-- for further operation. --!>
<!-------!>

<!-- #include file="AuthenticationCheck.inc" -->
<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
<TITLE>CartridgeLink</TITLE>
</HEAD>
<BODY background=images/bg.bmp><!--Display the page data-->
<div align=center>
    <big><big><font color=navy>CartridgeLink</font></big></big>
</div>

```

```

<br><br>
    <font color=teal>Welcome Back
    <%=Request.Cookies("CartridgeLink")("Username")%>&nbsp;
    <%=Request.Cookies("CartridgeLink")("password")%></font>
<br><br><!-- #include file="MainMenu.inc" -->

</BODY>
</HTML>

```

Registration.asp

```

<!-------!>
<!-- This program provides a registration form for attempted -----!>
<!-- registrations. The registered details will be caught by the --!>
<!-- database and RegComfirm.asp will be run to confirm -----!>
<!-- the registration. --!>
<!-------!>

<!-- #include file="advbbs.inc" -->
<!-- #include file="ProductionErrorHandler.inc" -->
<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
<TITLE>CartridgeLink</TITLE>
</HEAD>
<BODY background=images/bg.bmp><!--Display the page data-->
<div align=left>
    <big><big><font color=navy><U><FONT
color=darkred><FONT
size=4>CartridgeLink</FONT>&nbsp;</FONT></U>
    &nbsp;</font></big></big>
</div>
<div align=center>
    <big><big><font color=navy>
    </font></big></big>
</div>
<div align=center>
    <big><big><font color=navy> User Registration&nbsp;</font><FONT
color=#000000 size=3>
</FONT></big></big>
</div><br>

<form action=RegComfirm.asp method=post name=frmRegister>
<table>
<TBODY>
<tr>
<td height=50 colspan=2><font color=navy> Registration&nbsp;Form

```

```

        </font></td>
</tr>
<tr>
<td>First Name</td>
<td><input name=txtFirstName size=15 ></td>
<td width=50></td>
<td>Username</td>
<td><input name=txtLastName size=15 ></td>
</tr>
<tr>
<td>Last Name</td>
<td><input name=txtUsername size=8 ></td>
<td width=50></td>
<td>Password</td>
<td><INPUT id=password1
        style="WIDTH: 121px; HEIGHT: 22px" type=password size=15
        name=txtPassword></td>
</tr>
<tr>
<td>Email</td>
<td><input name=txtEmail size=50></td>
</tr>
<tr>
<td>User Group</td>
<td colspan=4>
        <input type=radio name=optClass value=1 checked><FONT size=1>
            ECU CartridgeLink Support</FONT>
        <input type=radio name=optClass value=2><FONT size=1>
            Level 1 - View Only</FONT>
        <input type=radio name=optClass value=3><FONT size=1>
            Level 2 - Normal</FONT>
        <input type=radio name=optClass value=3><FONT size=1>
            Level 3 - Advanced</FONT>
</td>
</tr>
<tr>
<td>Organisation</td>
<td colspan=3><select name=cboOrgs></select></td>
</tr>
<tr>
<td>&nbsp;</td>
</tr>
<tr>
<td><input type=button name=btnSubmit value=Submit></td>
</tr>
</form><!-- #include file="Connect.inc" -->

<%
'--- Check for database errors

```

Call CheckForErrors(objConn)

'--- Create the recordset object, set the SQL string and open the recordset

```
Set objConn = Server.CreateObject("ADODB.Connection")
Set objRS = Server.CreateObject("ADODB.Recordset")
strSQL = "group_info"
    objRS.Open strSQL,"driver={SQL Server};server=sciscjiao;uid=sa;pwd=;
    database=CartridgeLink", adOpenForwardOnly, , adCmdStoredProc
Call CheckForErrors(objConn)
%>
```

<script language=vbscript>

Sub Window_OnLoad()

<%

Do While Not objRS.EOF

%>

```
    Set objOption = document.createElement("OPTION")
    objOption.text = "<%=objRS("orgname")%>"
    objOption.value = "<%=objRS("organisation")%>"
    document.all.cboOrgs.add objOption
```

<%

objRS.MoveNext

Loop

%>

<!-- #include file="Disconnect.inc" -->

Set objOption = Nothing

End Sub

Sub btnSubmit_OnClick()

'--- Verify all fields have been entered

If Len(frmRegister.txtFirstName.value) = 0 Then

Alert "You must enter a first name"

frmRegister.txtFirstName.focus

Exit Sub

ElseIf Len(frmRegister.txtLastName.value) = 0 Then

Alert "You must enter a last name"

frmRegister.txtLastName.focus

Exit Sub

ElseIf Len(frmRegister.txtUsername.value) = 0 Then

Alert "You must enter a state abbreviation"

frmRegister.txtUsername.focus

Exit Sub

ElseIf Len(frmRegister.txtPassword.value) = 0 Then

Alert "You must enter your password"

frmRegister.txtPassword.focus

Exit Sub

```
ElseIf frmRegister.cboOrgs.selectedindex = -1 Then
    Alert "You must select the club to which you belong"
    frmRegister.cboOrgs.focus
    Exit Sub
End If

'--- Submit the form
Call frmRegister.submit()
End Sub
</script>
</TBODY></TABLE>

</BODY>
</HTML>
```

RegConfirm.asp

```

<!-------!>
<!-- This program confirms the registration and echo back ----!>
<!-- the user name and password. Once the registration is --!>
<!-- confirmed, the main menu is prompted for further --!>
<!-- operations ----!>
<!-------!>

<%
'--- Check for database errors
Call CheckForErrors(objConn)

Set objConn = Server.CreateObject("ADODB.Connection")
objConn.Open "driver={SQL Server};server=sciscjiao;database=CartridgeLink","sa",
""

'--- Set the parameters for the insert stored procedure
strSQL = "user_info_insert ('" & CStr(Request.Form("txtFirstName")) & _
        "','" & CStr(Request.Form("txtLastName")) & _
        "','" & CStr(Request.Form("txtUsername")) & _
        "','" & CStr(Request.Form("txtPassword")) & _
        "','" & Request.Form("cboOrgs") & _
        "','" & Request.Form("optClass") & "'"
'--- Execute the stored procedure to insert the person
objConn.Execute strSQL,,adCmdStoredProc

'--- Check for database errors
Call CheckForErrors(objConn)
%>

<!-- #include file="Disconnect.inc" -->
<%
'--- Save the user information to a cookie
Response.Cookies("Cartridge")("UserName") = Request.Form("txtFirstName")
Response.Cookies("Cartridge")("Password") = Request.Form("txtPassword")

'--- Set the expiration date of the cookie to the last day of the current year
Response.Cookies("Cartridge").Expires = "December 31, " & Year(Now)

'--- Authenticate the user for other web pages
Session("Authenticated") = True
%>

<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
<TITLE>CartridgeLink</TITLE>

```

```

</HEAD>
<BODY><!--Display the page data-->
<div align=center><FONT color=#000080 size=5>CartridgeLink</FONT>
<br><br>
    <font color=teal>Registration Successful, Your Password is
<%=Request.Form("txtPassword")%></font>
<br><br><!-- #include file="MenuOptions.inc" -->
</div>

</BODY>
</HTML>

```

MainMenu.inc

```

<!-------!>
<!-- This program shows main menu of the application. -----!>
<!-- 1. list registered cartridge cases, runs ListCartridge.asp -!>
<!-- 2. Insert a new cartridge case, runs Insert.asp --!>
<!-- 3. Search for a cartridge case, runs Search.asp --!>
<!-------!>

<table>
    <tr>
        <td height=50>Select an option</td>
    </tr>
    <tr>
        <td><a href="ListCartridge.asp"
            onmouseover="window.status='Display Firearms Cartridge Cases'"
            onmouseout="window.status=''">
            Display registered Firearm's Cartridge Cases</font></td>
        </tr>
        <tr>
            <td><a href="insert.htm"
                onmouseover="window.status='Register a New Recovered Cartridge
Case'"
                onmouseout="window.status=''">
                Register a new Recorvered Cartridge Case</font></td>
            </tr>
            <tr>
                <td><a href="search.asp"
                    onmouseover="window.status='Search for matched Cartridge Case '"
                    onmouseout="window.status=''">
                    Search for matched Cartridge Case</font></td>
                </tr>
</table>

```


ListCartridge.asp

```

<!-------!>
<!-- This program lists all registered cartridge case records-!>
<!-- The headstamp image id is highlighted for further    --!>
<!-- displaying image of that specific cartridge by running --!>
<!-- ShowImage.asp. Other Options of insert a new        --!>
<!-- cartridge or return to the main menu are available.  --!>
<!-------!>

<%
Response.Buffer = True
Dim connStr
connStr = "driver={SQL
Server};server=sciscjiao;uid=sa;pwd=;database=CartridgeLink"
%>

<html>
<head>
    <title>CartridgeLink</title>
    <style>
        body, input, td { font-family:verdana,arial; font-size:10pt; }
    </style>
</head>
<body>
    <p align="center">
        <b>List of Cartridges from the Database</b><br>
        <A href="insert.htm">Insert Cartridge</a>
        <A href="mainmenu.htm">Return To Main</A>
    </p>
    <table width="700" border="1" align="center">

<%
'--- Connection Object
Dim con
dim ssl
Dim recCount
Set con = Server.CreateObject("ADODB.Connection")

Dim rs
Set rs = Server.CreateObject("ADODB.Recordset")
ssl= "select [CartridgeID], [Markings], [Manufacturer],
[CartridgeCompositionType], [Description], [CartridgeName], [Calibre],
[HeadstampImage] from Cartridge order by [CartridgeID] desc"
rs.Open ssl,connStr
If Not rs.EOF Then

```

```

recCount = rs.RecordCount
Response.Write "<tr><td align=""center""><i>"
Response.Write "Headstamp ID<i>" & "</td><td align=""center""><i>"
    Response.Write "Markings<i>" & "</td>"
Response.Write "<td align=""center""><i>"
Response.Write "Manufacturer</i>" & "</td><td>"
Response.Write "</td><td align=""center""><i>"
Response.Write "Composition Type</i>" & "</td><td align=""center""><i>"
Response.Write "Cartridge Name<i>" & "</td><td align=""center""><i>"
Response.Write "Calibre<i>" & "</td><td>"
Response.Write "</td></tr>"
Dim rsID
reccount = 0
While Not rs.EOF
    If Len(CartridgeID) < 1 Then
        rsID = rs("CartridgeID")
    End If
    Response.Write "<tr><td>"
Response.Write "<a href=""ShowImage.asp?ID=" & rs("CartridgeID") &
"""">"
    Response.Write rs("CartridgeID") & "</td><td>"
    Response.Write rs("Markings") & "</td><td>"
    Response.Write rs("Manufacturer") & "</td><td>"
    Response.Write rs("Description") & "</td><td>"
    Response.Write rs("CartridgeName") & "</td><td>"
    Response.Write rs("Calibre") & "</td><td>"
    Response.Write "</td></tr>"
    reccount=reccount+1
    rs.MoveNext
Wend
Else
    Response.Write "No Record Found"
End If

Response.Write "<tr><td colspan=""8"" align=""center""><i>"
Response.Write "Number of records : " & recCount
Response.Write ", Table Name : Cartridge</i><br>"
Response.Write "</td></tr>"
rs.Close
Set rs = Nothing
Set con = Nothing
%>
</table>
</body>
</html>

```

ShowImage.asp

```

<!-------!>
<!-- This program shows the headstamp image of a selected!>
<!-- cartridge case and its classifications. By select return --!>
<!--button, user returns back to the original list. --!>
<!-------!>

<%
    '--- Retrieves binary files from the database
    Response.Expires = 0
    Response.Buffer = True
    Response.Clear
    Dim ID
    ID = Request("ID")

    Dim connStr
        connStr = "PROVIDER=MSDASQL;" & _
            "DRIVER={SQL Server};" & _
            "SERVER=sciscjiao;DATABASE=CartridgeLink"
    Dim rs
        Set rs = Server.CreateObject("ADODB.Recordset")
rs.Open "select [HeadstampID],[HeadstampImage] from Headstamps where
HeadstampID = " & ID, connStr, 2, 4
        If Not rs.EOF Then
            Response.ContentType = "Image/gif"
            Response.BinaryWrite rs("Headstampimage")
        End If
        rs.Close
        Set rs = Nothing
%>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" content="text/html; charset=iso-8859-1">
<TITLE>Headstamp Display Results</TITLE>
</HEAD>
<BODY BGCOLOR="#ffffff">
    <p align="center">
        <A href="ListCartridge.asp">To return to the list click here</A>
    </p>
</body>
</html>

```

Insert.htm

```

<!-------!>
<!-- This program allows user to insert a new cartridge case-!>
<!-- with its classifications and image. Options of return  --!>
<!--main and show registered cartridges are available  --!>
<!-------!>

<html>
<head>
    <title>CartridgeLink - Inserts Cartridge Casing into Database</title>
    <style>
        body, input { font-family:verdana,arial; font-size:10pt; }
    </style>
</head>
<body background=images/bg.bmp>
    <p align="center">
        <b>Inserting&nbsp;Cartridge Casing Data into Database</b><br>
        <A href="ListCartridge.asp">Show the List of
Cartridges</A>&nbsp;
        <A href="mainmenu.inc">Return To Main</A>
    </p>

    <table border="0" align="center" background=""  >
    <tr>
<FORM action=ImgLoad.asp method=post encType=multipart/form-data>

        <td>

            <TR>
                <TD colSpan=2><FONT color=teal><STRONG>Enter Cartridge Case
Classifications</STRONG>
                </FONT></TD></TR>
            <TR>
                <TD>Calibre<SELECT name=cboCalibre></SELECT>
                Manufacture<SELECT style="WIDTH: 225px"
name=cboMan></SELECT></TD>
            </TR>
            <TR>
                <TD>Composition Type<SELECT name=cboType></SELECT>
                Provided By<SELECT name=cboOrg></SELECT></TD>
            </TR>
            <TR>
                <TD>Cartridge Name<INPUT id=text1 name=text1> </TD>
            </TR>
            <TR>
                <TD>

```

```

<P><FONT color=#008080><STRONG>Enter
Headstamp;Markings, Image and Notes</STRONG></FONT>
</P>
<P>Markings:<INPUT name=txtMarkings></P></TD></TR>
<TR>
<TD>Notes:<INPUT name=txtNotes></TD></TR>

<TR>
<TD>File: <INPUT name=file></TD></TR>

<INPUT id=text2 name=text2><INPUT id=submit1 type=submit value=Submit
name=submit1></td></TR>

</FORM></tr></table><!-- #include file="Connect.inc" -->

<%
'--- Check for database errors
Call CheckForErrors(objConn)

'--- Preparing Comb Box values
Set objRSCalibre = Server.CreateObject("ADODB.Recordset")
strSQLCalibre = "cboCalibre_info"
objRSCalibre.Open strsqlCalibre,"driver={SQL
Server};server=sciscjiao;uid=sa;pwd=;database=IntraBFIS",
adOpenForwardOnly, , adCmdStoredProc

Set objRSMan = Server.CreateObject("ADODB.Recordset")
strSQLMan = "cboMan_info"
objRSMan.Open strsqlMan,"driver={SQL
Server};server=sciscjiao;uid=sa;pwd=;database=IntraBFIS",
adOpenForwardOnly, , adCmdStoredProc

Set objRSType = Server.CreateObject("ADODB.Recordset")
strSQLType = "cboType_info"
objRSType.Open strsqlType,"driver={SQL
Server};server=sciscjiao;uid=sa;pwd=;database=IntraBFIS",
adOpenForwardOnly, , adCmdStoredProc

Set objRSOrg = Server.CreateObject("ADODB.Recordset")
strSQLOrg = "cboOrg_info"
objRSOrg.Open strsqlOrg,"driver={SQL
Server};server=sciscjiao;uid=sa;pwd=;database=IntraBFIS",
adOpenForwardOnly, , adCmdStoredProc

'--- Check for database errors
Call CheckForErrors(objConn)
%>

<SCRIPT language=vbscript>

```

```

Sub Window_OnLoad()
<%
    Do While Not objRSCalibre.EOF
%>
        Set objOptionC = document.createElement("OPTION")
        objOptionC.text = "<%=objRSCalibre("CalibreName")%>"
        objOptionC.value = "<%=objRSCalibre("CalibreID")%>"
        document.all.cboCalibre.add objOptionC
<%
        objRSCalibre.MoveNext
    Loop
%>

<%
    Do While Not objRSMan.EOF
%>
        Set objOptionMan = document.createElement("OPTION")
        objOptionMan.text = "<%=objRSMan("ManufacturerName")%>"
        objOptionMan.value = "<%=objRSMan("ManufacturerId")%>"
        document.all.cboMan.add objOptionMan
<%
        objRSMan.MoveNext
    Loop
%>

<%
    Do While Not objRSType.EOF
%>
        Set objOptionType = document.createElement("OPTION")
        objOptionType.text = "<%=objRSType("TypeName")%>"
        objOptionType.value = "<%=objRSType("TypeID")%>"
        document.all.cboType.add objOptionType
<%
        objRSType.MoveNext
    Loop
%>

<%
    Do While Not objRSOrg.EOF
%>
        Set objOptionOrg = document.createElement("OPTION")
        objOptionOrg.text = "<%=objRSOrg("OrgName")%>"
        objOptionOrg.value = "<%=objRSOrg("OrgID")%>"
        document.all.cboOrg.add objOptionOrg
<%
        objRSOrg.MoveNext
    Loop
%>

```

```
<!-- #include file="Disconnect.inc" -->
```

```
Set objOption = Nothing  
End Sub
```

```
Sub btnSubmit_OnClick()  
    Call frmSearchCart.submit()
```

```
End Sub  
</SCRIPT>  
</TABLE>  
</body>  
</html>
```

ImgLoad.asp

```

<!-------!>
<!-- This program uploads inserted cartridge case together -!>
<!-- with its image into the database. Options of return --!>
<!--main and show registrated cartridges are available --!>
<!-------!>

<%@ LANGUAGE="VBSCRIPT" %>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" content="text/html; charset=iso-8859-1">
<TITLE>Simple File Upload To Database Results</TITLE>
</HEAD>
<BODY BGCOLOR="#ffffff">
<%

Set upl = Server.CreateObject("SoftArtisans.FileUp")
upl.Path = server.mappath("/SAFileUpSamples/temp")

%>
<% if upl.IsEmpty Then %>
The file that you uploaded was empty.
<% ElseIf upl.ContentDisposition <> "form-data" Then %>
Your upload did not succeed.
<%Else %>
<P>The file was successfully transmitted by the user.</P>
    <p align="center">
        <A href="ListCartridge.asp">To see inserted data click here</A>
    </p>
<%

dim txtMarkingsInput
dim txtManufacturerInput
dim txtCartridgeCompositionTypeInput
dim txtDescriptionInput
dim txtCartridgeNameInput
dim txtCannalureInput
dim txtCalibreInput
dim txtNotesInput
dim fileInput

txtmarkingsinput = upl.Form ("txtMarkings")
txtManufacturerInput=upl.Form ("txtManufacturer")
txtCartridgeCompositionTypeInput= upl.Form ("txtCartridgeCompositionType")

```



```

txtDescriptionInput= upl.Form ("txtDescription")
txtCartridgeNameInput= upl.Form ("txtCartridgeName")
txtCannalureInput= upl.Form ("txtCannalure")
txtCalibreInput= upl.Form ("txtCalibre")
txtNotesInput= upl.Form ("txtNotes")
'fileInput= upl.Form ("file")

```

```

Set cnBlob = Server.CreateObject("ADODB.Connection")
set rsBlob = Server.CreateObject("adodb.recordset")
rsBlob.Open "cartridgeHeadstamp", "driver={SQL
Server};server=sciscjiao;uid=sa;pwd=;database=cartridgeLink", 2, 3
rsBlob.AddNew
upl.SaveAsBlob rsBlob.Fields("HeadStampImage")
rsblob.Fields("Markings") =txtMarkingsInput
rsblob.fields("Notes")= txtNotesInput
rsBlob.Update
rsBlob.Close
Set rsBlob = Nothing
%>

```

```

<P>&nbsp;</P>
<FONT SIZE="-1"><CENTER>
<TABLE WIDTH="80%" BORDER="1" CELSPACING="2"
CELLPADDING="0" HEIGHT="206">
<TR>
<TD COLSPAN="2"><P><CENTER>Information About The Uploaded
File</CENTER></TD></TR>
<TR>
<TD WIDTH="30%" HEIGHT="27" ALIGN="RIGHT"
VALIGN="TOP">&nbsp;<User's filename</TD>
<TD WIDTH="70%"><%=upl.UserFilename%>&nbsp;</TD></TR>
<TR>
<TD WIDTH="30%" HEIGHT="27" ALIGN="RIGHT" VALIGN="TOP">Size in
bytes</TD>
<TD WIDTH="70%"><%=upl.TotalBytes%>&nbsp;</TD></TR>
<TR>
<TD WIDTH="30%" HEIGHT="27" ALIGN="RIGHT" VALIGN="TOP">Content
Type</TD>
<TD WIDTH="70%"><%=upl.ContentType%>&nbsp;</TD></TR>
<TR>
<TD WIDTH="30%" HEIGHT="27" ALIGN="RIGHT" VALIGN="TOP">Content
Disposition</TD>
<TD WIDTH="70%"><%=upl.ContentDisposition%>&nbsp;</TD></TR>
<TR>
<TD WIDTH="30%" HEIGHT="27" ALIGN="RIGHT" VALIGN="TOP">MIME
Version</TD>
<TD WIDTH="70%"><%=upl.MimeVersion%>&nbsp;</TD></TR>
<TR>

```

```
<TD WIDTH="30%" HEIGHT="27" ALIGN="RIGHT" VALIGN="TOP">Content
Transfer Encoding</TD>
<TD WIDTH="70%"><%=upl.ContentTransferEncoding%>&nbsp;</TD></TR>
</TABLE>
</FONT></CENTER>
<% End If %>
</BODY>
</HTML>
```

SearchCart.asp

```

<!-------!>
<!-- This program receives required cartridge information -!>
<!-- and search through the database. The search results are-!>
<!--listed and can be extended to view images --!>
<!-------!>

<!-- #include file="adovbs.inc" -->
<!-- #include file="ProductionErrorHandler.inc" -->

<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
<TITLE>CartridgeLink</TITLE>
</HEAD>
<BODY background=images/bg.bmp><!--Display the page data-->
<div align=center>
    <big><big><font color=navy>
CartridgeLink -&nbsp;  Search a cartridge</font></big></big>&nbsp;  
</div>
<br>
<TABLE>

<TR>
<TD colSpan=2><FONT color=teal>Enter Search details and click
submit.</FONT></TD></TR>
<TR>
<TD>&nbsp;  </TD></TR>
<TR>
<TD>Calibre</TD>
<TD colSpan=3><SELECT style="WIDTH: 185px"
name=cboCalibre></SELECT></TD></TR>
<TR>
<TD>Manufacture</TD>
<TD colSpan=3><SELECT style="WIDTH: 184px"
name=cboMan></SELECT></TD></TR>
<TR>
<TD>Composition Type</TD>
<TD colSpan=3><SELECT style="WIDTH: 182px"
name=cboType></SELECT></TD></TR>
<TR>
<TD>Provided By</TD>
<TD colSpan=3><SELECT style="WIDTH: 182px"
name=cboOrg></SELECT></TD></TR>
<TR>
<TD>&nbsp;  </TD></TR>
<TR>

```

```
<TD><INPUT style="LEFT: 13px; TOP: 270px" type=submit value=Search
name=btnSubmit></TD></TR><!-- #include file="Connect.inc" -->
```

```
<%
'Check for database errors
Call CheckForErrors(objConn)

Set objRSCalibre = Server.CreateObject("ADODB.Recordset")
strSQLCalibre = "cboCalibre_info"
objRSCalibre.Open strSQLCalibre,"driver={SQL
Server};server=sciscjiao;uid=sa;pwd=;database=IntraBFIS", adOpenForwardOnly, ,
adCmdStoredProc

Set objRSMan = Server.CreateObject("ADODB.Recordset")
strSQLMan = "cboMan_info"
objRSMan.Open strSQLMan,"driver={SQL
Server};server=sciscjiao;uid=sa;pwd=;database=IntraBFIS", adOpenForwardOnly, ,
adCmdStoredProc

Set objRSType = Server.CreateObject("ADODB.Recordset")
strSQLType = "cboType_info"
objRSType.Open strSQLType,"driver={SQL
Server};server=sciscjiao;uid=sa;pwd=;database=IntraBFIS", adOpenForwardOnly, ,
adCmdStoredProc

Set objRSOrg = Server.CreateObject("ADODB.Recordset")
strSQLOrg = "cboOrg_info"
objRSOrg.Open strSQLOrg,"driver={SQL
Server};server=sciscjiao;uid=sa;pwd=;database=IntraBFIS", adOpenForwardOnly, ,
adCmdStoredProc

'Check for database errors
Call CheckForErrors(objConn)
%>

<SCRIPT language=vbscript>
Sub Window_OnLoad()
<%
    Do While Not objRSCalibre.EOF
%>
        Set objOptionC = document.createElement("OPTION")
        objOptionC.text = "<%=objRSCalibre("CalibreName")%>"
        objOptionC.value = "<%=objRSCalibre("CalibreID")%>"
        document.all.cboCalibre.add objOptionC

<%
        objRSCalibre.MoveNext
    Loop
%>
```

```

<%
    Do While Not objRSMan.EOF
%>
        Set objOptionMan = document.createElement("OPTION")
        objOptionMan.text = "<%=objRSMan("ManufacturerName")%>"
        objOptionMan.value = "<%=objRSMan("ManufacturerId")%>"
        document.all.cboMan.add objOptionMan

<%
        objRSMan.MoveNext
    Loop
%>

<%
    Do While Not objRSType.EOF
%>
        Set objOptionType = document.createElement("OPTION")
        objOptionType.text = "<%=objRSType("TypeName")%>"
        objOptionType.value = "<%=objRSType("TypeID")%>"
        document.all.cboType.add objOptionType

<%
        objRSType.MoveNext
    Loop
%>

<%
    Do While Not objRSOrg.EOF
%>
        Set objOptionOrg = document.createElement("OPTION")
        objOptionOrg.text = "<%=objRSOrg("OrgName")%>"
        objOptionOrg.value = "<%=objRSOrg("OrgID")%>"
        document.all.cboOrg.add objOptionOrg

<%
        objRSOrg.MoveNext
    Loop
%>

    <!-- #include file="Disconnect.inc" -->

    Set objOption = Nothing
End Sub

Sub btnSubmit_OnClick()
    'Verify all fields have been entered

    'If we get to this point all is OK, submit the form
    Call frmSearchCart.submit()
End Sub
</SCRIPT>
</TABLE>

```

<HR>
Search Result

</BODY>
</HTML>

Connect.inc

```

<%
'Instruct VBScript to ignore the error and continue
'with the next line of code
On Error Resume Next

'Create and open the database object
Set objConn = Server.CreateObject("ADODB.Connection")
objConn.Open Application("ConnectionString")
%>

```

Disconnect.inc

```

<%
'Close and dereference database objects
If IsObject(objRS) Then
    objRS.Close
    Set objRS = Nothing
End If
objConn.Close
Set objConn = Nothing
%>

```

AuthenticationCheck.inc

```

<%
'Authentication check
If Session("Authenticated") <> True Then
    Session("ErrorMessage") = "You Have not properly logged in."
    Response.Redirect "2Logging.asp"
End If
%>

```

ProductionErrorHandler.inc

```

<script language=vbscript runat=server>
Function CheckForErrors(objConnection)
    'Declare variables

```

```

Dim blnDisplayErrMsg

If objConnection.Errors.Count > 0 Then

    'Create the FileSystemObject and open the error log
    Set objFile = Server.CreateObject("Scripting.FileSystemObject")
    Set objLog = objFile.OpenTextFile( _
        Server.MapPath("ProductionErrorLog.txt"),8,True)

    'Check for an open error from VBScript
    If Err.Number > 0 Then
        Response.Write "Error opening log file<P>"
        Response.Write "Error Number: " & Err.Number & _
            ", Error Description: " & Err.Description
    End If

    'Create an error object to access the ADO errors collection
    Set objErr = Server.CreateObject("ADODB.Error")

    'Log all errors to the error log
    For Each objErr In objConnection.Errors
        If objErr.Number = 0 Then
            blnDisplayErrMsg = False
        Else
            objLog.WriteLine(objErr.Number & "|" & _
                objErr.Description & "|" & objErr.Source & _
                "|" & _
                objErr.SQLState & "|" &
objErr.NativeError)
            blnDisplayErrMsg = True
        End If
    Next

    'Close the log file and dereference all objects
    objLog.Close
    Set objLog = Nothing
    Set objFile = Nothing
    Set objErr = Nothing

    If blnDisplayErrMsg Then
        'Display a graceful message to the user
        Response.Write "An unforeseen error has occurred and
processing " & _
            "must be stopped. You can try your request again
later "

        'Halt Execution
        Response.End
    End If
End If

```


End Function
</script>

APPENDIX C: STORED PROCEDURES

```

cartridge_info      CREATE PROC cartridge_info @cboCalibre int,
                    @cboMan int, @cboType varchar(3), @cboOrg int

                    AS
                    SELECT  dbo.Cartridge.*,
                    dbo.CartridgeCalibre.CalibreName AS Expr1,
                    dbo.CartridgeManufacturer.ManufacturerName AS
                    Expr2,
                    dbo.CartridgeCompositionType.TypeName AS Expr3,
                    dbo.Organisation.OrgName AS Expr4,
                    dbo.CartridgeHeadStamp.Markings AS Expr5,
                    dbo.CartridgeHeadStamp.Notes AS Expr6,
                    dbo.CartridgeHeadStamp.HeadStampImage AS Expr7

                    FROM      dbo.Cartridge LEFT OUTER JOIN
                                dbo.CartridgeHeadStamp ON
                    dbo.Cartridge.HeadstampID =
                    dbo.CartridgeHeadStamp.HeadStampId LEFT OUTER
                    JOIN
                                dbo.Organisation ON dbo.Cartridge.OrgID
                    = dbo.Organisation.OrgID LEFT OUTER JOIN
                                dbo.CartridgeCompositionType ON
                    dbo.Cartridge.TypeID =
                    dbo.CartridgeCompositionType.TypeID LEFT OUTER
                    JOIN
                                dbo.CartridgeManufacturer ON
                    dbo.Cartridge.ManufacturerID =
                    dbo.CartridgeManufacturer.ManufacturerId LEFT
                    OUTER JOIN
                                dbo.CartridgeCalibre ON
                    dbo.Cartridge.CalibreID =
                    dbo.CartridgeCalibre.CalibreID
                    WHERE(dbo.Cartridge.CalibreID = @cboCalibre)
                    AND (dbo.Cartridge.ManufacturerID = @cboMan)
                    AND (dbo.Cartridge.TypeID = @cboType)
                    and (dbo.Cartridge.OrgID = @cboOrg)
                    GO

```

```

cartridgetab_info      CREATE PROC cartridgetab_info @cboCalibre int,
                        @cboMan int, @cboType varchar(3), @cboOrg int
                        AS
                        SELECT  CalibreID, ManufacturerID,TypeID, OrgID,
                        CartridgeID, CartridgeName, HeadstampID
                        FROM      dbo.Cartridge
                        WHERE(CalibreID = @cboCalibre)
                        AND (ManufacturerID = @cboMan)
                        AND (TypeID = @cboType)
                        and (OrgID = @cboOrg)
                        GO

CategoryInfo           CREATE PROCEDURE CategoryInfo
                        AS
                        SELECT CategoryName, Description
                        FROM    Categories
                        FOR XML AUTO
                        GO

cboCalibre_info        CREATE PROC cboCalibre_info as
                        SELECT  Calibre, CalibreName
                        FROM      dbo.CartridgeCalibre
                        GO

cboOrg_info            CREATE PROC cboOrg_info as
                        SELECT  OrgID, OrgCode, OrgName
                        FROM      dbo.Organisation
                        GO

cboType_Info          CREATE PROC cboType_info as
                        SELECT  TypeID, TypeName
                        FROM      dbo.CartridgeCompositionType
                        GO

Group_Info            CREATE PROC group_info as
                        SELECT Organisation,orgname
                        FROM dbo.organisation
                        GO

```

HeadstampInfo	<pre> CREATE PROCEDURE HeadStampInfo AS SELECT HeadStampID, Markings FROM tblHeadStamp FOR XML AUTO GO </pre>
user_info	<pre> CREATE PROC user_info @username varchar(10), @Password varchar(10) AS SELECT UserName, [Password] FROM dbo.useraccount WHERE (username = @username) AND ([password] = @password) GO </pre>
user_info_insert	<pre> CREATE PROC user_info_insert @firstname varchar(20), @lastname varchar(20), @Username varchar(10), @password nvarchar(10), @userorg integer, @usergroup integer as INSERT INTO Useraccount (fname, lname, username, [password], org, usergroup) values (@firstname, @lastname, @username, @password, @userorg, @usergroup) GO </pre>

Paper Published by

The International Conference on Operations Research for Development,
Chennai, India, 27 - 30 December, 2002

Ballistic Image Firearm Identification Database Management System On An Intranet Using SQL Server 2000

D. Li and C. Jiao

*School of Computer and Information Science, Edith Cowan University
2 Bradford Street, Mount Lawley 6050, Australia
Email: d.li@ecu.edu.au*

Abstract: The process of identifying a firearm by using the evidence recovered from a crime is a time consuming one. Additionally, geographic and organization separations compound the amount of effort required to solved linked crimes. This paper introduce a computerized ballistic firearm identification system in an Intranet environment is designed and implemented, thus enabling firearm examiners in the community to share and grow the ballistic information without the dual constraints of location and paper transformation. Specifically, the study investigates the process involved to select the database management system to which it can be integrated into this Intranet system. The results demonstrate that an integrated Intranet ballistic firearm identification system helps firearm examiners effectively therefore helps to improve the law enforcement.

✓ **Key words:** Forensic science, DBMS, Intranet, Image Storage, Security

1. INTRODUCTION

1.1. General Information

Forensic science plays an important role in law enforcement and firearm identification is one of the respects of the area. After forensic technology combines with computer technology, the firearm identification can be processed through not only a forensic laboratory but also a ballistic database that collects information and images about firearms. This study is set up to prototype an Intranet firearm cartridge identification system based on an image included database management system (DBMS).

1.2. Background

Every firearm has individual characteristics that are as unique to it as fingerprints are to human beings [1]. When a firearm is fired, it transfers these characteristics – in the form of microscopic scratches and dents – to the projectiles and cartridge casings fired in it. This fact creates a great opportunity for law enforcement. When bullets or cartridge casings are found as crime scene, firearms examiners can use the marks for comparison, to determine whether or not the bullets or casings were expelled from a suspect's firearm. If a firearm is recovered at the scene, a test fire of the weapon creates example bullets and cartridge casings for comparison. Bullets and cartridge casings found at one crime scene can also be compared with those found at another in order to link the crimes. The prospect is important for solving organized crimes.

However the comparison of ballistic evidence, as pointed out by Inbau [2], has in the past been a tedious and time-consuming process. Evidence recovered at crime scenes or from recovered firearms was compared piece by piece, to the vast inventory of recovered or test-fired projectiles and casings.

There were desktop firearm identification systems [3] used by individual police departments and the identification performance was improved a great deal [4]. However identifying firearms in a desktop system with isolated databases may miss the links between linked crimes occurred in different states [5]. Therefore this study introduces both Intranet and database server technologies into a firearm cartridge identification system named CartridgeLink.

1.3. Project Overview

The fundamental aim of the CartridgeLink is to effectively identifying firearms via matching up cartridge cases over an Intranet interface and Intranet DBMS. This application is to be prototyped in a manner that is reliable in terms of the DBMS selected to equip vast amount of data including images.

1.4. Hypothesis

The hypothesis of the prototype is set to prove that A appropriate selected DBMS and programming language will allow the integration between an Intranet application and its backend DBMS providing scalability, security, and performance.

1.5. Statement of the Problems

The problems set out to be answered by the study are described as following

- a) What are the problems associated with prototyping and implementing an Intranet firearm identification system?
- b) What are some possible simple solutions to secure an intelligent database like ballistic firearm identification database?
- c) How do we store images in the system?
- d) How can we improve an Intranet base image database's performance?

2. DESIGN AND IMPLEMENTATION

Following on from previous introduction, which discussed the limitations of current computerised ballistics firearm identification systems, within this chapter, components are described that overcome those limitations. Additionally, stated problems posed have been explained and the prototype is been designed.

2.1. System Requirements

In order to allow firearm examiners national wide to share data safely in a closed environment without the constraint of isolation and database, several steps must be satisfied. This section provides a brief explanation of the required steps and outlines necessary software and hardware requirements.

Software used for the prototype

Software tools used for the prototype include:

- SQL Server 2000 [6] as DBMS;
- Internet Information Service 5.0 [7] as web server;
- Internet Explorer [8] ;
- Intranet programming tools; and
- PowerDesigner [9] as data modeling tool.

Hardware configuration

To overcome isolation and database capacity restrictions, there must be a database server where users of the system may gain access in their own location, such as that provided by the Intranet. The system is built for the configuration listed in Table 1.

Item	Value
OS Name	Microsoft Windows 2000 Professional
Internet Explorer	V6.0
IIS (Internet Information Service)	5.0
SQL Server 2000	Server
Internet Access	Promised
System Type	X86-based PC
Processor	X86 Family 6 Model 8 Stepping 6 GenuineIntel ~870 Mhz
Total Physical Memory	261,600 KB
Available Physical Memory	59,368 KB
Network Adapter Type	Ethernet 802.3
Hard Disk	19.13 GB

Table 1 - Configurations set up for the system

Intranet access

Security usage of the database may be carried out among users of the system constructed for this prototype by creating an Intranet environment. Within a closed environment, Intranet pages may be accessed, employing Internet technology, only is people granted permission.

2.2 User Interfaces Design

The flow chart shown in Figure 3 can demonstrate the procedures of the system.

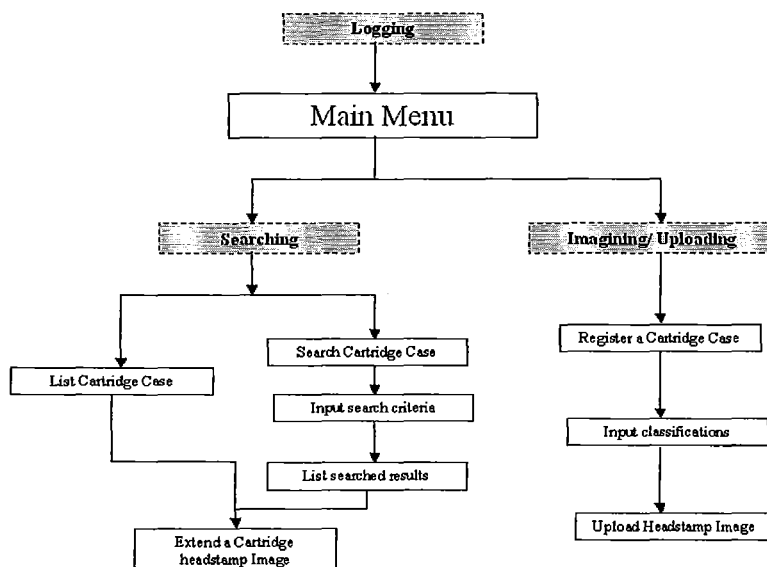


Figure 1 - Procedure Flow Chart

2.3 Database Design

Figure 4 shows the data module designed for prototyped system (A trial version of PowerDesigner is used to illustrate the design).

3. IMPLEMENTATION AND RESULTS

Based on the above design that our fundamental aim is to identifying cartridge case over an Intranet interface and Intranet DBMS, in order to find the matches to firearms. This attempt is to be accomplished in a manner that is reliable in terms of the DBMS selected to equip vast amount of data including images. Design decisions are required for the selection of appropriate DBMS tools to perform the specific components of the implementation, and for the choice of suitable programming tools that are adequately integrated to produce a prototype system performing cartridge identifications.

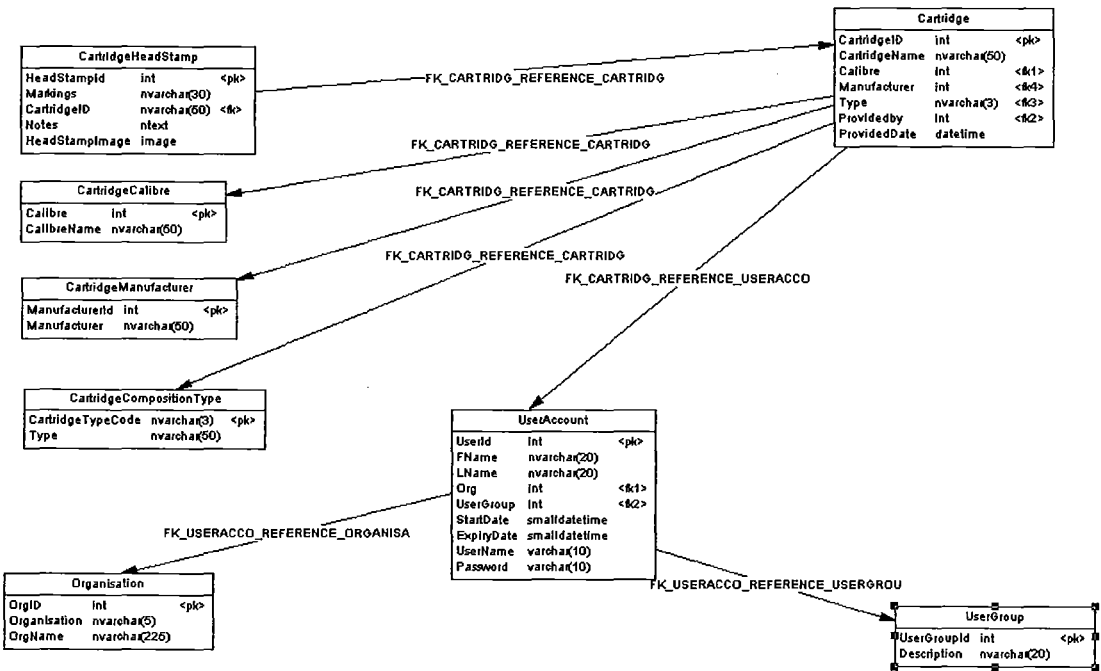


Figure 2 - Data Model of CartridgeLink

3.1 Selection of DBMS

For reasons of availability, scalability, reliability, economy, compatibility (to the programming language used in this project – see next paragraph) and its feature-similarity to typical image data storage, a proprietary DBMS SQL Server 2000 was employed. Usage of the evaluation system’s hardware was constrained for the study’s purpose, to those facilities normally found within an Internet Server. The constraints precluded the use of clients.

3.2 Selection of Programming Language

Similar reason prompted the initial use of the VB scripts and HTML encapsulated by ASP and HTM pages for the Intranet’s programming, as the vehicle for user interface generation. The ASP, however, proved to be incomplete in its support for Netscape browser. The limitation caused by competitions between Netscape and Microsoft and Netscape supports Java as its HTML generator. Notably, Microsoft Internet Explore supports both Java and ASP script as a marketing strategy and it is more popular to the developers. This is one of the reasons that Microsoft Internet Explorer is used as the platform.

3.3 Samples of Outcome

Figure 3 and 4 shows two samples of outcomes from the prototype.

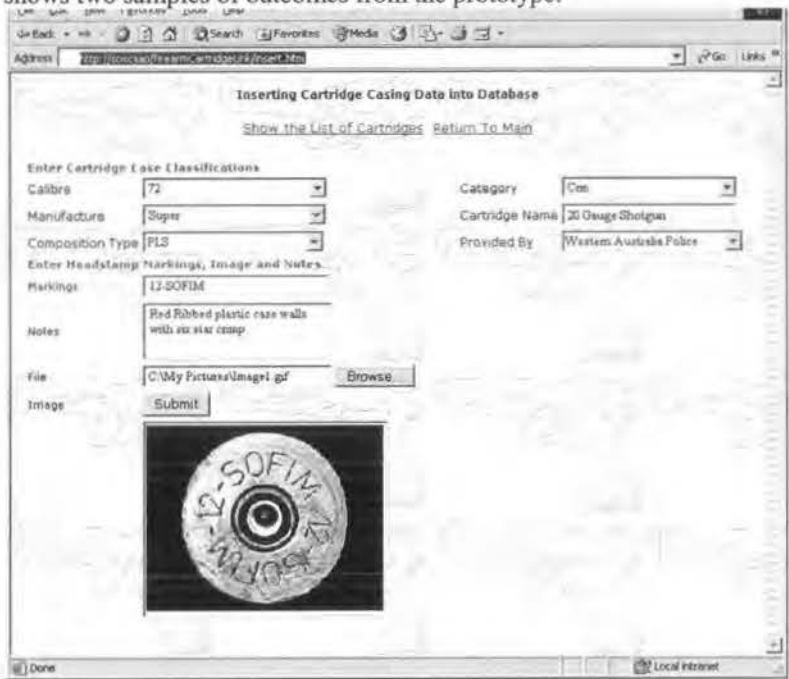


Figure 3 - Uploading image into database



Figure 4 - Result of Search and Search Results

4. CONCLUSION AND FUTURE WORK

This section answers previously stated questions and provides a discussion to each question. A discussion is given for other important topics. This is then followed by a conclusion and possible future work.

4.1 Question a) What are the problems associated with prototyping an Intranet firearm identification system?

There are many different problems associated with implementing Intranet firearm identification system. Below is a sample of some of the problems encountered and overcome during this project:

a1) What are main tools/software required to develop an Intranet firearm identification application as CartridgeLink?

Answer: It is found that the tools required for the requirement include IIS (Internet Information Service), Intranet Server programming language and a DBMS.

a2) What type of DBMS is suitable for managing Intranet ballistics database with cartridge case images?

Answer: A range of DBMS types/modules can be used for implementing Intranet ballistics database with images. The most effective one found was ORDBMS, a combination of the OODBMS and RDBMS.

a3) How do you access ORDBMS with ASP Intranet programming language?

Answer: Through a set of SQL Server 2000 objects via VB script in ASP.

a4) What kind of image storage method is suitable for Intranet ballistics firearm identification application like CartridgeLink?

Answer: Two major methods can be used for store/collecting images. The effective one found was a stored the images together with their classification/metadata in the database.

a5) How do you implement images in the database for the Intranet application with only a field in a record such as a cartridge case record? How can you improve the applications overall performance?

NOTE: This question is answered by Question 3 and Question 4.

4.2. Question b) What are some possible logging solutions to securing an intelligent database like ballistic database?

Answer: Security is essential to a ballistics database. The security tools used including

- SQL Server 2000's Security Architecture with Window or Mix authentic mode;
- A Security Plan - identifies which users can see which data and perform which activities in the database;
- Security Accounts and Permissions
- Establishment of application security user account that protects the application from illegal users; and
- Intranet and database server security environments.

4.3 Question c) How to store images in the system?

Answer: Two major methods may be used for store/collecting images. The prototype stores the images together with their classification/metadata in the database.

4.4 Question d) How can we improve an Intranet based image database's performance?

This was addressed by the prototype by storing cartridge case images into the database, using stored procedures instead of embedded SQL statements and indexing.

4.5 Benefits

Information Availability

Intranet firearm identification database like CartridgeLink improve the law enforcement by improving the chances to identify firearms over a possible national wide DBMS. The system required no geometric or organizational limitation within the law system. It is available over an Intranet provided for laboratories. The database was administrated and updated over the net. The search for a suspected firearm via its cartridge case can be made from large scale of data contributed by other states. The potential link between crimes may be discovered effectively hence enhances law enforcement.

Save of Time

During the original firearm identification process, all of external enquired cartridge information was made out on mail or email systems. Received information would need to treat before the comparisons. This is very time consuming. With a Intranet Firearm Identification Cartridge Database and application, once a user gain the access, the needed images and data are automatically available for the identification. This streamlines the process and saves time.

Reduce Redundant Data

Redundant data not only wastes time and resources, but also causes confusions and misunderstandings. By using an integrated Intranet database, the same firearm fires with the same bullet need only to be produced only. This production can than be used by other laboratories is it is needed. All we need to do is make sure this piece of information is correct and updated. This provides data integrity for the users.

Security

Compare with current system's database security, this project provides a integrated security facility between OS, Intranet, Database administration and Application Usage Authorization. When it is implemented into the live system with other securities components such as firewall, the Intranet firearm identification system would be a secured system over the web.

URL Querying

It is possible to querying the database with SQL XML statement and stored procedures using URL box in the browser. This is particular useful for the database administrators.

CONCLUSION

Computerized Ballistics Firearm identification is a topic that is increasingly interested, as it is comparably new to Australian computerized forensic science. Currently there is a computerized firearm identification system used by some police departments that can effectively support local state laboratories in identifying firearms with local DBMSs. This thesis demonstrated the implementation of a cartridge identification system that is capable of operated over an Intranet. It automatically links laboratories working together based on the cartridge case database server to allow the firearm to be identified quickly and effectively.

REFERENCES

- [1] J. S. Doyle, "Fundamental of Firearms ID," 2002.
- [2] F. E. Inbau, "Firearms identification--"Ballistics"," *Journal of Criminal Law & Criminology*, vol. 49, pp. 1293-1314, 1999.
- [3] D. G. Li and A. C. Watson, "Ballistics firearms identification based on images of cartridge case and projectile," presented at SPIE - The International Society for Optical Engineering, Boston, Massachusetts, 1998a.
- [4] A. Australia Broadcasting Corporation Television, "Fireball," vol. 2001: Ian Watson, Naomi Lumsdaine, 2001.
- [5] Department of the Treasury Bureau of Alcohol Tobacco and Firearms USA, "The Missing Link: Ballistics Technology That Helps Solve Crimes," 2002.
- [6] Microsoft, "SQL Server 2000." Redmond, Washington, 2000, pp. 303-336.
- [7] Microsoft, "Product Documentation for Internet Information Services 5.0," Microsoft, 2002.
- [8] Microsoft, "Internet Explorer." Redmond, Washington, 2002, pp. 303-336.
- [9] I. Sybase, "SYBASE SELECTED BY NCR TO IMPROVE CUSTOMER SERVICE AT THE UNITED STATES POSTAL SERVICE," 1999.