The Network Sniffer Database and Web Interface

By

Anthony M. Boehm

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Engineering Technology

University of Cincinnati
College of Applied Science

June 2003

The Network Sniffer Database and Web Interface

By

Anthony M. Boehm

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements
for
the Degree of Bachelor of Science
in Information Engineering Technology

© Copyright 2003 Anthony M. Boehm

The authors grant to the Information Engineering Technology Program permission to reproduce and distribute copies of this document in whole or in part.

Anthony M. Boehm — Date: 6/5/03

Prof. Russell McMahon, Faculty Advisor — Date: 6/5/03

James F. Sullivan, Department Head — Date: 5 June, 2003

Acknowledgements

I would like to give special thanks to all those that helped make this project possible. This includes my wife Rebecca and my son Alexander. Thank you for your patience and support. I would also like to thank Alan Lucking for his guidance through some difficult research. Finally, I would like to thank the rest of my family for making every effort to support my successful completion of this project and my degree.

Table of Contents

Section	Page
Acknowledgements	ii
Table of Contents	iii
List of Figures	v
Abstract	vi
1. Introduction	1
2. The Purpose	1
2.1 The Problem	1
2.2 The Solution	3
2.2.1 Allowing the Data to be Queried	3
2.2.2 Expanding the Range of Information	3
2.2.3 The Interface	4
3. User Profiles	4
3.1 Technicians	4
3.2 Administrators	5
4. Design Protocols	5
4.1 Database Design	5
4.2 Web Interface Design	6
4.2.1 Sniffer Data Interface Opening Page	7
4.2.2 Insert Pages	9
4.2.3 Query Pages	10
4.2.4 Security	10
5. Deliverables	10
6. Timeline	11
7. Budget	12
8. Proof of Design	13
8.1 The SQL Server Database	13
8.1.1 Importing the Inventory Database	13
8.1.2 Creating the Sniffer Text File DTS Package	14
8.2 The Web Interface	16
8.2.1 The Sniffer Datagrid Page	16
8.2.2 The Add PC Inventory Page	17
8.2.3 The Add Dell Purchase Page	19
8.2.4 The Add User Page	20

8.2.5 The Software Inventory Page 21
8.2.6 The Database Search Page 22
8.2.6.1 The Search 22
8.2.6.2 The Results 25
8.2.6.3 The Edit Option 26
8.2.6.4 The Delete Option 27
8.2.6.5 Data Validation In The Datagrids 28
8.2.7 The User Login Page 29

9. Testing Procedure 30

10. Conclusions and Recommendations 31
10.1 Conclusions 31
10.2 Recommendations 31

References 34

List of Figures

Figure Number	Page
Figure 1. Database Design, The seven primary tables are labeled.	6
Figure 2. Site Map of the Web Interface	7
Figure 3. Initial Sniffer Data Interface Appearance	8
Figure 4. Input Page Example	9
Figure 5. Sniffer Database Development Timeline	12
Figure 6. Budget Analysis	13
Figure 7. Sniffer Data DTS Package	15
Figure 8. The Sniffer Datagrid	17
Figure 9. The Add New PC to Database page	19
Figure 10. The Dell Purchase Page	20
Figure 11. The New User Page	21
Figure 12. The Software Inventory Page	22
Figure 13. A numeric search of the PC Inventory by CPU speed	23
Figure 14. A non numeric search of the User inventory by Login Name	24
Figure 15. A search based on a fixed criteria sub category for Software Inventory by Software Title	25
Figure 16. The edit options available for the User record from Figure 13	26
Figure 17. The delete option in a Software Inventory search	28
Figure 18. Datagrid regular expression validation control for the User Name field	29
Figure 19. User Login page with incorrect user name and password	30

Abstract

The Network Sniffer Database and Web Interface is a web based solution for the Xetron Information Systems department's information needs. This project was requested by Dave Steele, the Information Systems Department Head. The purpose of the project was to replace the existing PERL script based web site with a more functional ASP.NET base site. The site uses a SQL Server 2000 backend to maintain the database. The Database is a combination of data from a network sniffer program and a PC inventory database. The Sniffer Data is pulled from a text file that the program updates every ten minutes. A Data Transformation Package imports the text file into SQL Server on a matching rotation. The Inventory Database was imported from Access using the DTS tools in SQL Server. The site allows the users to query the database along with editing the data. Visual Studio.NET was used for the site development and VB.NET was the chosen language. The site is simple to navigate and offers a complete data solution for all the departments needs.

The Network Sniffer Database and Web Interface

1. Introduction

Xetron Corporation is a small engineering company owned by Northrop Grumman. The head of Xetron's Information Systems department is Dave Steele[A]. He and Dave Alverson[B], one of the software engineers, have developed an in-house network sniffer program. The output of the program is a text file that is over-written every ten minutes. Mr. Steele assigned me the task of developing a database to store the output of this program so that the information could be queried, sorted and managed. The interface for the database is an internal Web page that allows him to sort, query and modify the records. From my discussions with Mr. Steele, I determined it would need to reference an inventory database that I have created to keep track of the hardware and software that each workstation is using.

2. The Purpose

2.1 The Problem

Dave Steele and Dave Alverson developed their programming skills by developing their own software solutions to problems in the work place. Several years ago they began to develop a network sniffer to track activity on the Xetron Corporate network. This software was developed in C code. Its purpose was to track network packets and to use the information recorded to make assumptions about which workstation sent the packets. They were successful in creating a program that could identify the following characteristics of user workstations based on the packets it sent: the

---

[A] All information supplied about Dave Steele was received though a personal interview conducted on January 18, 2002. This includes the history of the project and his expectations of future development.

[B] All information supplied about Dave Alverson was received though a personal interview conducted on January 23, 2002. This includes information regarding the sniffer text file and its format.

MAC address, the IP address, the computer name, the operating system, the port number on each switch used and the number of days since the last logon. The software gathered all this information and inserted it into a text file. It used tab delimiters to separate each field and carriage returns to separate each record. This text file served as the sole output of the software. Every ten minutes the text file would be replaced with the latest updates. The software was successful in retrieving the information they were interested in.

The next step in their project was to develop a way to view this data. Their solution was to use a PERL script to read the text file and place it on an internal Web page. The PERL script read the text file and inserted the data into an HTML table. Each field had a separate column and each record had a separate row. To make the table manageable, each column had a heading. The table could be sorted by column just by selecting the heading at the top. This would place the records in ascending order based on the column selected. This output made the information readily viewable in any web browser. This is the current status of the network sniffer project.

Since the development of the Web page interface, Mr. Steele has been unable to spare the time to further develop his project. He has remained unsatisfied with the interface that they developed. He wanted to take the interface to the next level by making the information user friendly. This involved allowing the information to be queried so that he does not have to search through hundreds of workstations to find the few he is working with. He also wanted to expand the range of information that he was able reference about each machine. Mr. Steele wanted to see what software and hardware each machine was using for licensing and inventory purposes. Finally he wanted a web

interface that would allow him to conduct queries and produce reports that make the information manageable and increasingly useful.

2.2 The Solution

The solution consisted of three primary parts. The first step was to make the sniffer data workable. The second step was to link the Inventory database. The third step was creating a web interface for interaction with the data.

2.2.1 Allowing the Data to be Queried

For information to be queried it must be organized and stored within a database. The information that the sniffer collects is stored within a text file on a workstation. I proposed to use Data Transformation Service (DTS) in SQL Server 2000 to import the text file into a database. SQL Server 2000 has the ability to import text files using the built-in DTS flat file OLE DB provider. This allowed me to set up a package to interpret the information within the text file and direct the data into a table. Tab delimiters separate the fields in the text file and the records are separated by carriage returns. I used DTS to read each record into a data table so that the information could be reviewed. This process repeats every ten minutes in cycle with the sniffer output. The data table allows the information to be queried and linked to other useful data.

2.2.2 Expanding the Range of Information

A database is only as good as the information it provides to the user; thus, the Sniffer database was linked to a workstation inventory database. The inventory database includes the hardware specifications from each workstation, the name of each workstation and the licensed software on each workstation. The purpose of this link is to make all the information about each workstation available. The expansion of the

database allows Mr. Steele to see how often each computer is used, what applications are being used and which workstations need hardware and software upgrades. The port and switch information also allows him to deduce where each workstation is located. All the information he has requested is at his fingertips.

2.2.3 The Interface

The final stage to this project was creating the user interface. The interface needed to be available to all members of the IS staff, therefore it needed to be accessible from the company Intranet. I proposed using a Visual Basic.Net web application to develop this interface. I used ADO connections to connect to the database. The application uses input boxes to receive restrictions for database queries. The interface will be customized to fit Mr. Steele's specific needs and will allow opportunities for future growth.

3. User Profiles

Xetron's Information Systems group had two types of users: technicians and administrators.

3.1 Technicians

The technicians at Xetron are computer experts. They are responsible for trouble shooting PC and networking issues and training PC users how to effectively use both hardware and software. This group will be using the database to keep track of computer hardware and software distribution and ownership. They will also use the database as a resource to look up information about individual computers.

This group is experienced in navigating through web interfaces. All the technicians have many years of experience and are full time employees. However, this group has limited experience with database software and programming languages.

3.2 Administrators

The Administrators are the head of the Information Systems Department. They will use the database to display reports and conduct queries. This is a small group of two or three people. They are knowledgeable and experienced. They desire a simple and functional web interface. This group has a strong background in databases and programming. This group will be capable of editing the SQL database and Web interface as the groups needs grow.

4. Design Protocols

4.1 Database Design

The Sniffer database includes seven primary tables. The first table is the Sniffer data table. It includes all the data imported from the network sniffer program using the SQL server DTS tool. This is the table that will get automatically replaced every ten minutes. The DTS package recreates the table and its relationships and then repopulates the records. The other six primary tables were imported from the inventory database. The Dell purchase list table includes records of all PCs ordered from Dell. The PC Naming List table includes all hardware and software information that has been collected and acts as the center of the entire database. The User Information table includes information about each user and his/her user account. The Department Information table includes all information about the department structure of Xetron. The Windows Purchase table is a summation of all recorded Microsoft Windows purchases. The Office

Purchase table is a summation of all recorded Microsoft Office purchases. The remaining tables are used to store a static list of all the possible options for the related fields. These tables are used to fill list boxes for the interface and allow the options in the list boxes to be updated by adding records to these tables rather than by modifying code. The database diagram along with all of the database relationships can be seen in **Figure 1**.

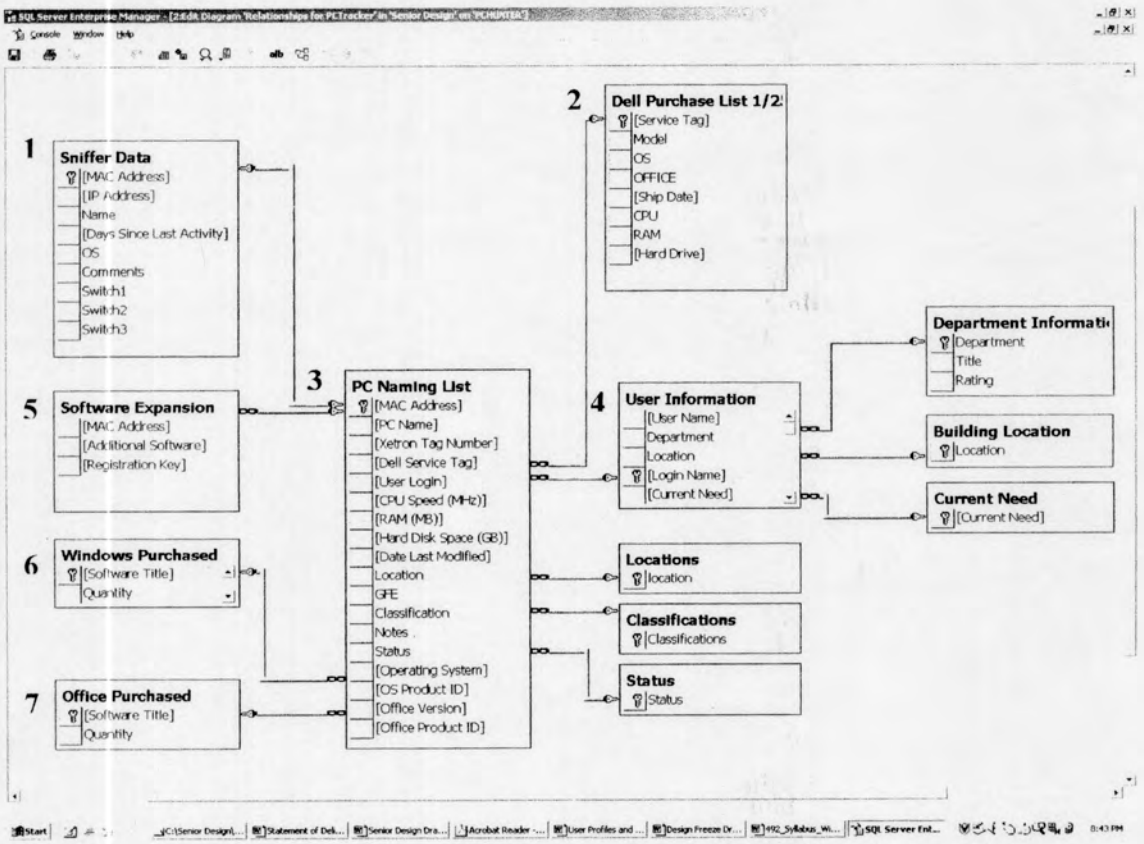


Figure 1. Database Design, The seven primary tables are labeled.

4.2 Web Interface Design

A map of the basic layout of the site can be seen in **Figure 2.**

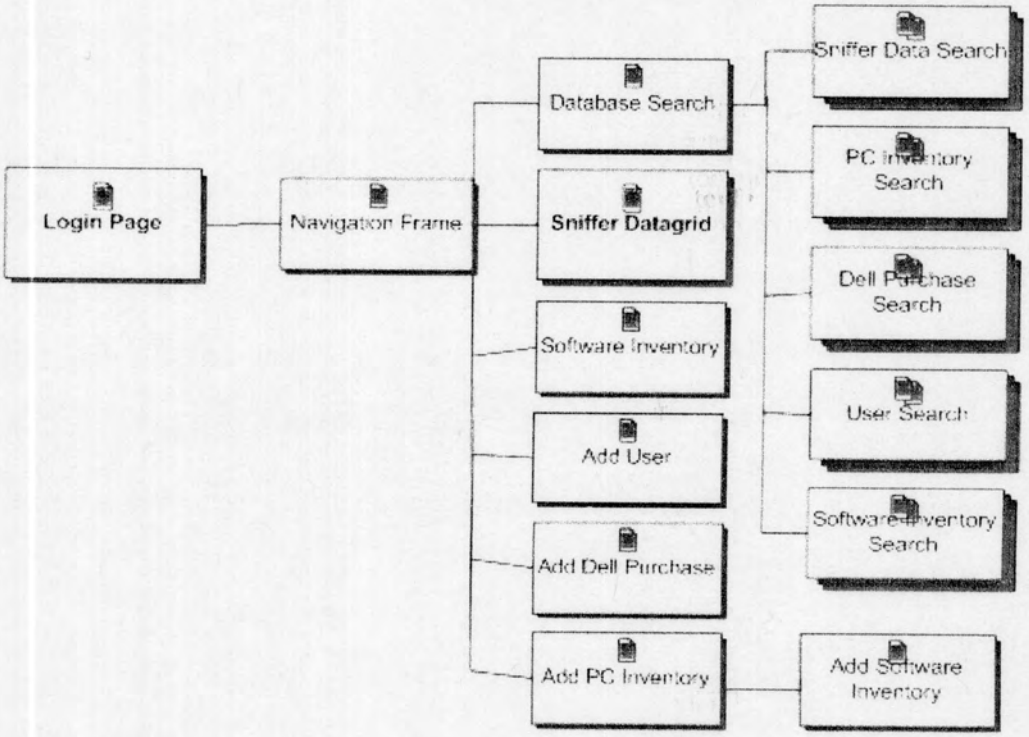
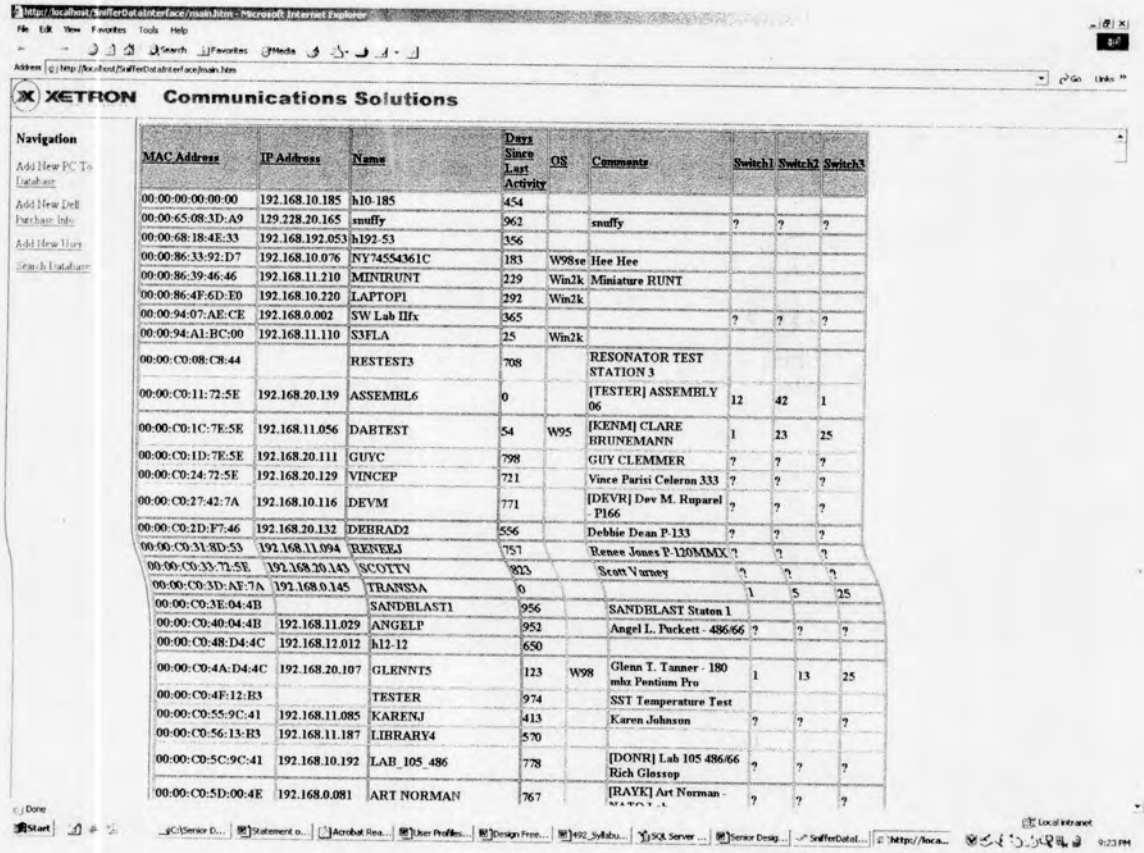


Figure 2. Site Map of the Web Interface

4.2.1 Sniffer Data Interface Opening Page

The Sniffer Data Web Interface is divided into three frames; a Header frame, a Navigation frame and a Main frame. The initial web interface opens as shown in **Figure 3**.



MAC Address	IP Address	Name	Days Since Last Activity	OS	Comments	Switch1	Switch2	Switch3
00:00:00:00:00:00	192.168.10.185	h10-185	454					
00:00:65:08:3D:A9	129.228.20.165	snuffy	962		snuffy	?	?	?
00:00:68:18:4E:33	192.168.192.053	h192-53	356					
00:00:86:33:92:D7	192.168.10.076	NY74554361C	183	W98se	Hee Hee			
00:00:86:39:46:46	192.168.11.210	MINIRUNT	229	Win2k	Miniature RUNT			
00:00:86:4F:6D:E0	192.168.10.220	LAPTOP1	292	Win2k				
00:00:94:07:AE:CE	192.168.0.002	SW Lab IIfx	365			?	?	?
00:00:94:A1:BC:00	192.168.11.110	S3FLA	25	Win2k				
00:00:C0:08:C8:44		RESTEST3	708		RESONATOR TEST STATION 3			
00:00:C0:11:72:5E	192.168.20.139	ASSEMBL6	0		[TESTER] ASSEMBLY 06	12	42	1
00:00:C0:1C:7E:5E	192.168.11.056	DABTEST	54	W95	[KENM] CLARE BRUNEMANN	1	23	25
00:00:C0:1D:7E:5E	192.168.20.111	GUYC	798		GUY CLEMMER	?	?	?
00:00:C0:24:72:5E	192.168.20.129	VINCEP	721		Vince Parisi Celeron 333	?	?	?
00:00:C0:27:42:7A	192.168.10.116	DEVM	771		[DEVR] Dev M. Ruparel - P166	?	?	?
00:00:C0:2D:F7:46	192.168.20.132	DEBRAD2	556		Debbie Dean P-133	?	?	?
00:00:C0:31:8D:53	192.168.11.094	RENEEJ	757		Renee Jones P-120MMX	?	?	?
00:00:C0:33:72:5E	192.168.20.143	SCOTTV	823		Scott Varney	?	?	?
00:00:C0:3D:AF:7A	192.168.0.145	TRANS3A	0			1	5	25
00:00:C0:3E:04:4B		SANDBLAST1	956		SANDBLAST Staton 1			
00:00:C0:40:04:4B	192.168.11.029	ANGELP	952		Angel L. Puckett - 486/66	?	?	?
00:00:C0:48:D4:4C	192.168.12.012	h12-12	650					
00:00:C0:4A:D4:4C	192.168.20.107	GLENNT5	123	W98	Glenn T. Tanner - 180 mhz Pentium Pro	1	13	25
00:00:C0:4F:12:B3		TESTER	974		SST Temperature Test			
00:00:C0:55:9C:41	192.168.11.085	KARENJ	413		Karen Johnson	?	?	?
00:00:C0:56:13:B3	192.168.11.187	LIBRARY4	570					
00:00:C0:5C:9C:41	192.168.10.192	LAB_105_486	778		[DONR] Lab 105 486/66 Rich Glossop	?	?	?
00:00:C0:5D:00:4E	192.168.0.081	ART NORMAN	767		[RAYK] Art Norman -	?	?	?

Figure 3. Initial Sniffer Data Interface Appearance

A data grid displays the contents of the Sniffer Data table. The data grid operates in the same manner as the existing Sniffer web page that it replaced. The page reads the sniffer data into a table that can be sorted by columns simply by clicking on the heading. The purpose of using this page is to give the users the sense that none of the original functionality is lost with this new interface. Instead they should feel that the new interface builds upon the old interface.

4.2.2 Insert Pages

The insert pages all maintain a consistent feel. They all contain three columns; a column of labels, a column of input/list boxes and a column of validation controls. The list boxes are filled using SQL commands that pull the list of options from corresponding tables in the database. The validation controls are a frontline defense against improper input from the user. A submit button at the bottom of each page forces the page to further authenticate the input and will then pass it on to the database. **Figure 4** is an example of the insert page format.

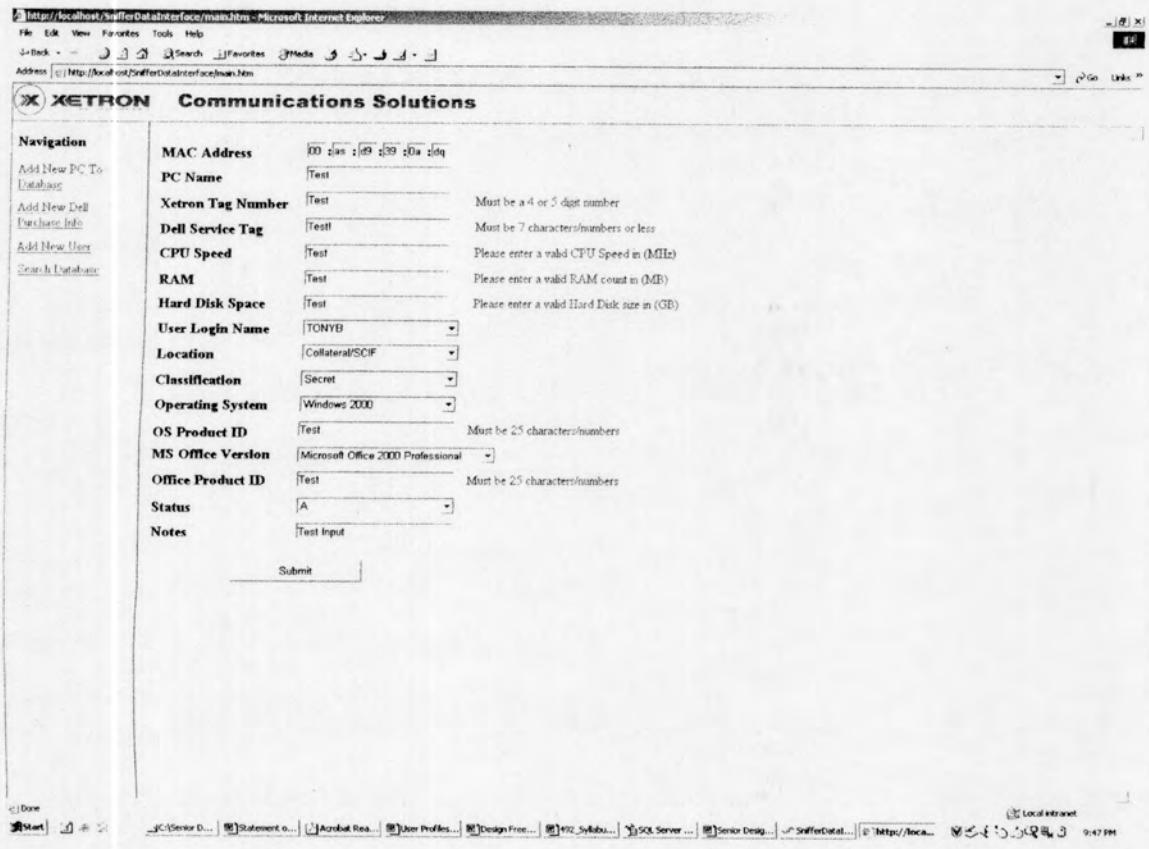


Figure 4. Input Page Example

4.2.3 Query Pages

The Query page is set up in an advanced search format. The user specifies fields to search and criteria to compare to. The results are displayed using data grids. The datagrids allow the user to sort the results by column headings and include edit and delete links to allow the user to edit records and save the changes to the database. The query page uses a series of list boxes to narrow the range of search options until the user has narrowed the search to one or a combination of fields. The users will then be able to enter text to compare against the database. The results of the search page act as an editing interface for the records that are returned.

4.2.4 Security

The Sniffer Data Interface requires a user login to access the site. The login is integrated into the main page. If a user tries to access another page without logging in first, they are redirected to the main page

5. Deliverables

5.1 A web-based interface to access a combination of data from a Network Sniffer Program and a Computer Inventory Database

- **5.1.1** The data from the network sniffer and the inventory database will be imported into the database using DTS
- **5.1.2** The Sniffer Data will be updated every ten minutes through a scheduled DTS package.

5.2 The user interface will be created using Visual Studio.Net.

- **5.2.1** The coding will be done using Visual Basic.Net and ASP.Net.

5.3 The data will be stored in a SQL database using SQL Server 2000.

5.4 The Web Interface will be compatible with Microsoft Internet Explorer 5.5 and 6.0.

5.5 The Web Interface will allow the users to insert, update and delete records in the database.

5.5.1 Validation controls will be used to ensure clean data is being sent to the database.

5.5.2 There will be insert pages for the User, PC, Software Inventory and Dell Purchase tables.

5.5.3 Edits and Deletions will be possible through the search pages.

5.6 The home page will initially load the Sniffer Data table, which will operate the same as the current display tool.

5.6.1 It will allow them to sort by column

5.7 A search page will be included to allow the user to specify criteria and conduct searches of the database.

5.7.1 There will be Software, PC, User and Dell Purchase search options.

5.8 The Web Interface will use a user login to restrict use to proper personnel.

5.9 The Web Interface will be simple and functional. It will not be flashy or incorporate large amounts of graphics.

5.10 The Navigation will be simple and straightforward. Everything will be reachable from the main page and logical links will be incorporated on similar pages

6. Timeline

The timeline for this project spanned 17 months, starting January 2002 and ending June 2003. The timeline is outlined in **Figure 5**.

ID	Task Name	Start	Finish	Duration
1	Research	1/18/2002	4/1/2002	10.4w
2	Write Proposal	2/11/2002	3/4/2002	3.2w
3	Present Proposal	3/4/2002	3/4/2002	2w
4	Develop Test Sniffer Database	8/1/2002	11/1/2002	13.4w
5	Develop Connection To Inventory Database	11/1/2002	11/29/2002	4.2w
6	Develop Test DTS Project	9/2/2002	10/3/2002	4.8w
7	Develop Test Web Interface	1/15/2003	2/14/2003	4.6w
8	Develop Prototype	1/15/2003	3/13/2003	8.4w
9	Write Report For Senior Design 2	1/30/2003	3/13/2003	6.2w
10	Present Prototype For Senior Design 2	3/13/2003	3/13/2003	2w
11	Refine and Develop Final Sniffer Database, DTS and Inventory Database Connection	3/13/2003	4/30/2003	7w
12	Refine Final Web Interface	3/13/2003	4/30/2003	7w
13	Finalize and Test Project	4/30/2003	5/30/2003	4.6w
14	Prepare Final Report	5/15/2003	6/2/2003	2.6w
15	Present Final Project	6/2/2003	6/2/2003	2w

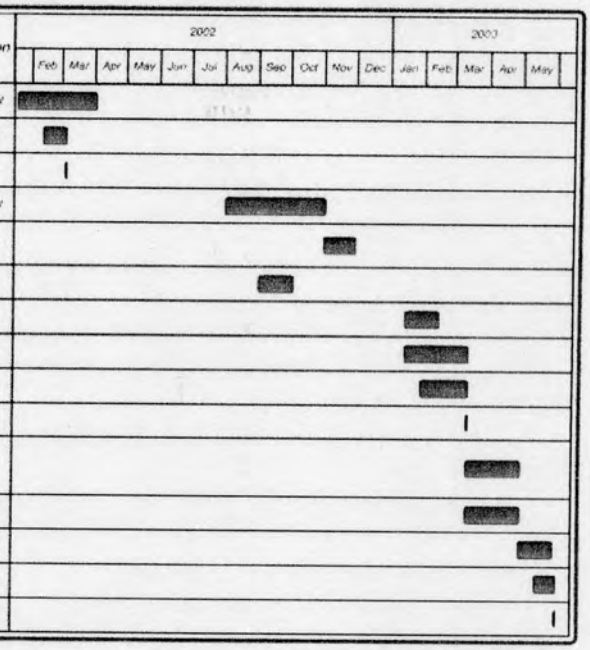


Figure 5. Sniffer Database Development Timeline

The time distribution is based on an average of eight hours of dedicated work each week. The total labor hours for this project are based on an estimated eight hours a week over seventy weeks. The timeline includes project research and development thru three co-op quarters in addition to Senior Design I and II. The completed project was presented in June 2003.

7. Budget

The budget requirements for this project involve software, hardware and labor hours. The software requirements include the full costs for SQL Server 2000, Visual Studio .Net and Microsoft Office 2000. However, the actual cost to Xetron is lower because the software has been purchased for previous projects. The hardware used for development was a Dell Inspiron 8200 laptop. I chose a laptop for portability, which allowed me to transport the product to and from home and school for development and

demonstration purposes. The computer hardware is sufficient enough to perform as a database and web server. The labor costs are the most significant portion of the budget and are based on the hours estimated in the project timeline. The reference materials are an estimate to cover materials that needed to be purchased during the project. **Figure 6** shows a complete list of all expenditures.

Microsoft SQL Server 2000 Standard Edition	$1,489.00 (Amazon.com)
Visual Studio .Net Professional	$999.99 (Amazon.com)
Reference Material	$200.00
Labor Hours	$12.50 / hour
Estimated Labor Hours (Based on timeline in section 4)	560
Hardware	$1,900.00 (Dell.com)
Total Estimated Project Budget	$11,588.99

Figure 6. Budget Analysis

8. Proof of Design

8.1 The SQL Server Database

The database solution I used was SQL Server 2000. The design of the database can be seen in **Figure 1**.

8.1.1 Importing the Inventory Database

The inventory database was imported from Access using the DTS Import/Export Wizard. I created a local copy of the Access file and then ran the wizard using this copy.

However, I had to make some formatting changes before I imported the database. In the sniffer text file, the MAC address is stored using semicolons between the numeric pairs. In the inventory database dashes were used to divide the pairs. After conducting research on manipulating data through DTS, I decided it best to convert the dashes to semicolons within Access before importing the data to SQL Server. I did this using the find and replace all command in the edit menu of access. This was done to all tables that included the MAC address column. Once my data was formatted correctly I ran the import wizard and created the tables within my Senior Design database in SQL Server. Once the tables were imported I began the task of recreating the table relationships. The relationships were not created by the Wizard. During this process I discovered several duplications in the data that needed to be corrected before the primary keys and relationships could be created.

8.1.2 Creating the Sniffer Text File DTS Package

The Sniffer text file is imported through a scheduled DTS package. The package was designed to run every ten minutes to retrieve the updated file. As seen in **Figure 7**, the package has a connection to the Sniffer text file and a connection to the SQL Database.

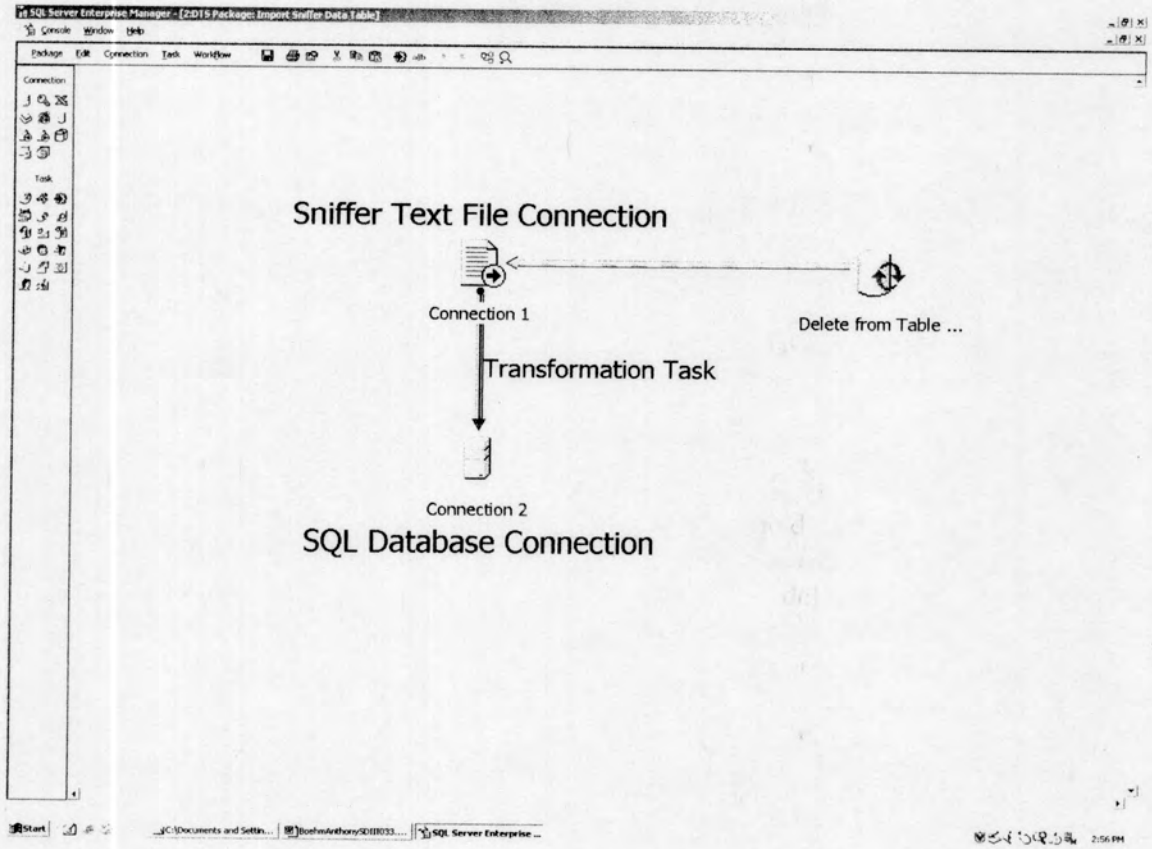


Figure 7. Sniffer Data DTS Package

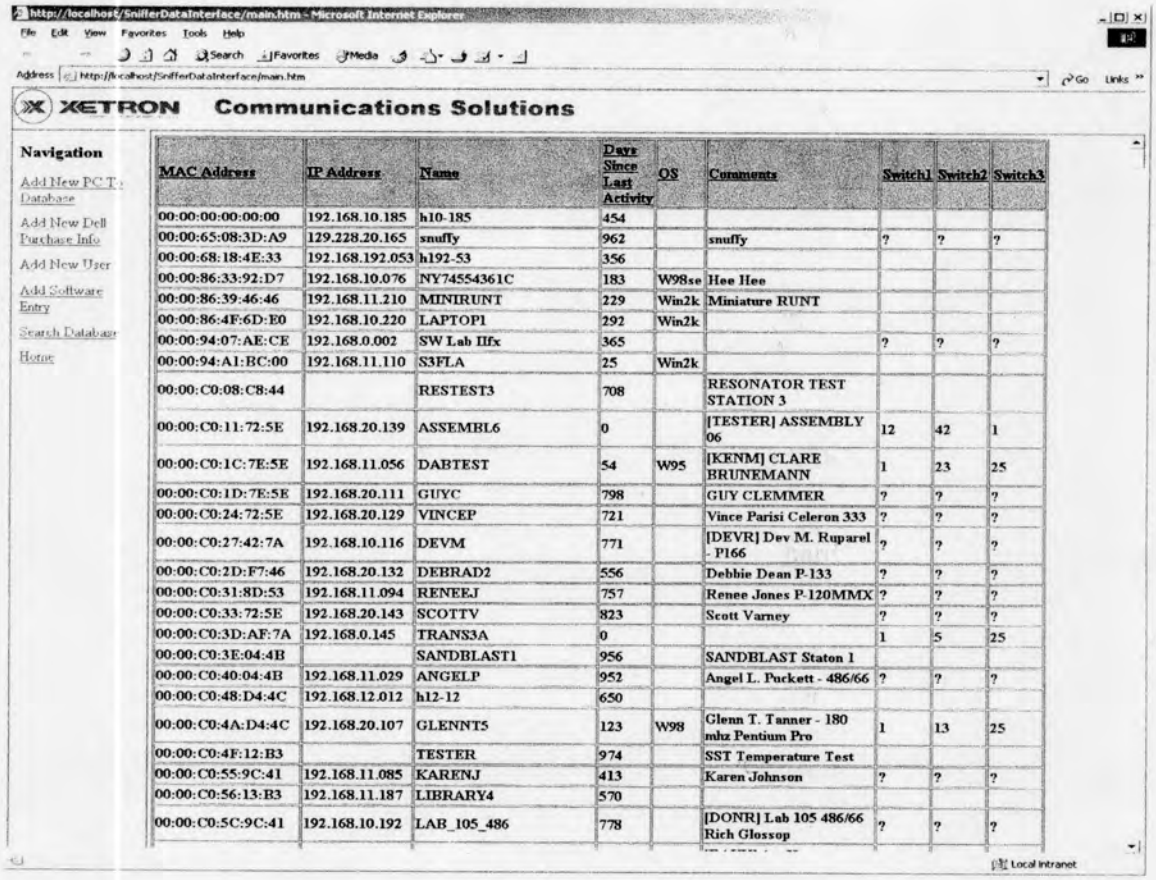
The first task it processes is deleting all the records in the Sniffer data table. Once this has completed successfully, the workflow continues on to the data connections and the data transformations. In the data transformation task, the tab delimited file is passed into columns. This task recreates the records in the Sniffer data table. This package was tested by importing three sample files from the Sniffer program that were saved over the span of two weeks.

8.2 The Web Interface

The Web Interface is designed to be easy to navigate and is very simple graphically. A plain white background is used along with blue and black text and boarders. Occasionally red text is used to grab the user's attention to important details. The simple layout was selected to allow the user's attention to remain focused on the data and not be distracted. The only graphic on the site is the ever present Xetron company logo across the header of all the pages. The simplicity of the page layout is designed to reflect the simplistic layout of the page I am replacing with this interface.

8.2.1 The Sniffer Datagrid Page

The Sniffer Datagrid page is the direct replacement of the PERL script page that was being used to interface with the Sniffer data. The page consists of a datagrid control that displays the data from the Sniffer data table. The datagrid is bound to a dataset and is configured to auto generate the columns and column headers. This is the most simplistic form of a datagrid. The column headers act as sort buttons for the datagrid. The "onsort" command is activated whenever a column header is selected and causes the dataset to rebind, sorted in ascending order by the column which was selected. This page is a functionally identical replacement to the PERL page that it replaced. This page acts as the site home page and can be reached by selecting the *Home* link on the navigation bar. **Figure 8** shows an example of the Sniffer Datagrid.



MAC Address	IP Address	Name	Days Since Last Activity	OS	Comments	Switch1	Switch2	Switch3
00:00:00:00:00:00	192.168.10.185	h10-185	454					
00:00:65:08:3D:A9	129.228.20.165	snuffy	962		snuffy	?	?	?
00:00:68:18:4E:33	192.168.192.053	h192-53	356					
00:00:86:33:92:D7	192.168.10.076	NY74554361C	183	W98se	Hee Hee			
00:00:86:39:46:46	192.168.11.210	MINIRUNT	229	Win2k	Miniature RUNT			
00:00:86:4F:6D:E0	192.168.10.220	LAPTOP1	292	Win2k				
00:00:94:07:AE:CE	192.168.0.002	SW Lab IIfx	365			?	?	?
00:00:94:A1:BC:00	192.168.11.110	S3FLA	25	Win2k				
00:00:C0:08:C8:44		RESTEST3	708		RESONATOR TEST STATION 3			
00:00:C0:11:72:5E	192.168.20.139	ASSEMBL6	0		[TESTER] ASSEMBLY 06	12	42	1
00:00:C0:1C:7E:5E	192.168.11.056	DABTEST	54	W95	[KENM] CLARE BRUNEMANN	1	23	25
00:00:C0:1D:7E:5E	192.168.20.111	GUYC	798		GUY CLEMMER	?	?	?
00:00:C0:24:72:5E	192.168.20.129	VINCEP	721		Vince Parisi Celeron 333	?	?	?
00:00:C0:27:42:7A	192.168.10.116	DEVM	771		[DEVR] Dev M. Ruparel - P166	?	?	?
00:00:C0:2D:F7:46	192.168.20.132	DEBRAD2	556		Debbie Dean P-133	?	?	?
00:00:C0:31:8D:53	192.168.11.094	RENEEJ	757		Renee Jones P-120MMX	?	?	?
00:00:C0:33:72:5E	192.168.20.143	SCOTTV	823		Scott Varney	?	?	?
00:00:C0:3D:AF:7A	192.168.0.145	TRANS3A	0			1	5	25
00:00:C0:3E:04:4B		SANDBLAST1	956		SANDBLAST Staton 1			
00:00:C0:40:04:4B	192.168.11.029	ANGELP	952		Angel L. Puckett - 486/66	?	?	?
00:00:C0:48:D4:4C	192.168.12.012	h12-12	650					
00:00:C0:4A:D4:4C	192.168.20.107	GLENNT5	123	W98	Glenn T. Tanner - 180 mhz Pentium Pro	1	13	25
00:00:C0:4F:12:B3		TESTER	974		SST Temperature Test			
00:00:C0:55:9C:41	192.168.11.085	KARENJ	413		Karen Johnson	?	?	?
00:00:C0:56:13:B3	192.168.11.187	LIBRARY4	570					
00:00:C0:5C:9C:41	192.168.10.192	LAB_105_486	778		[DONR] Lab 105 486/66 Rich Glossop	?	?	?

Figure 8. The Sniffer Datagrid

8.2.2 The Add PC Inventory Page

The Add PC Inventory page allows the users to insert hardware and software information for a new computer. When the page initially loads, it has a series of text boxes and dropdown list boxes used to input the required hardware and software data. The dropdown list boxes are populated through SQL queries that access the option tables in the database. The MAC address field is a series of six text boxes for the six sets of number/characters. A Java script automatically moves the cursor from one box to the next to allow the user to enter the MAC address without entering the required semicolons. Each text box is linked to a Regular Expression Validation control that

informs the user if inappropriate information is entered. Required Field Validation controls are linked to the MAC address fields. These validation controls display errors in red text next to the controls they are linked to. The Submit button below the notes field runs a final check of the validation controls and then runs a SQL command to insert the new record into the database. Once the record has been successfully entered, all the input fields are locked and a set of optional software input fields are displayed at the bottom of the page. A dropdown list box supplies the list of software that can be inventoried and a textbox is located below it for the registration key. The *Add Software Title* button will insert the software into the software table linked to the MAC address of the PC inventory that was just performed. When this option is selected, the title is added to a datagrid below the buttons and the software title and registration key are reset for the next entry. The user can enter any number of software titles at this point without having to leave this page. The other options on this page are the New PC Inventory button which will reload the page for a new PC inventory or the Return to Main Page button which takes the user back to the home page. An example of a new PC inventory can be seen in **Figure 9**.

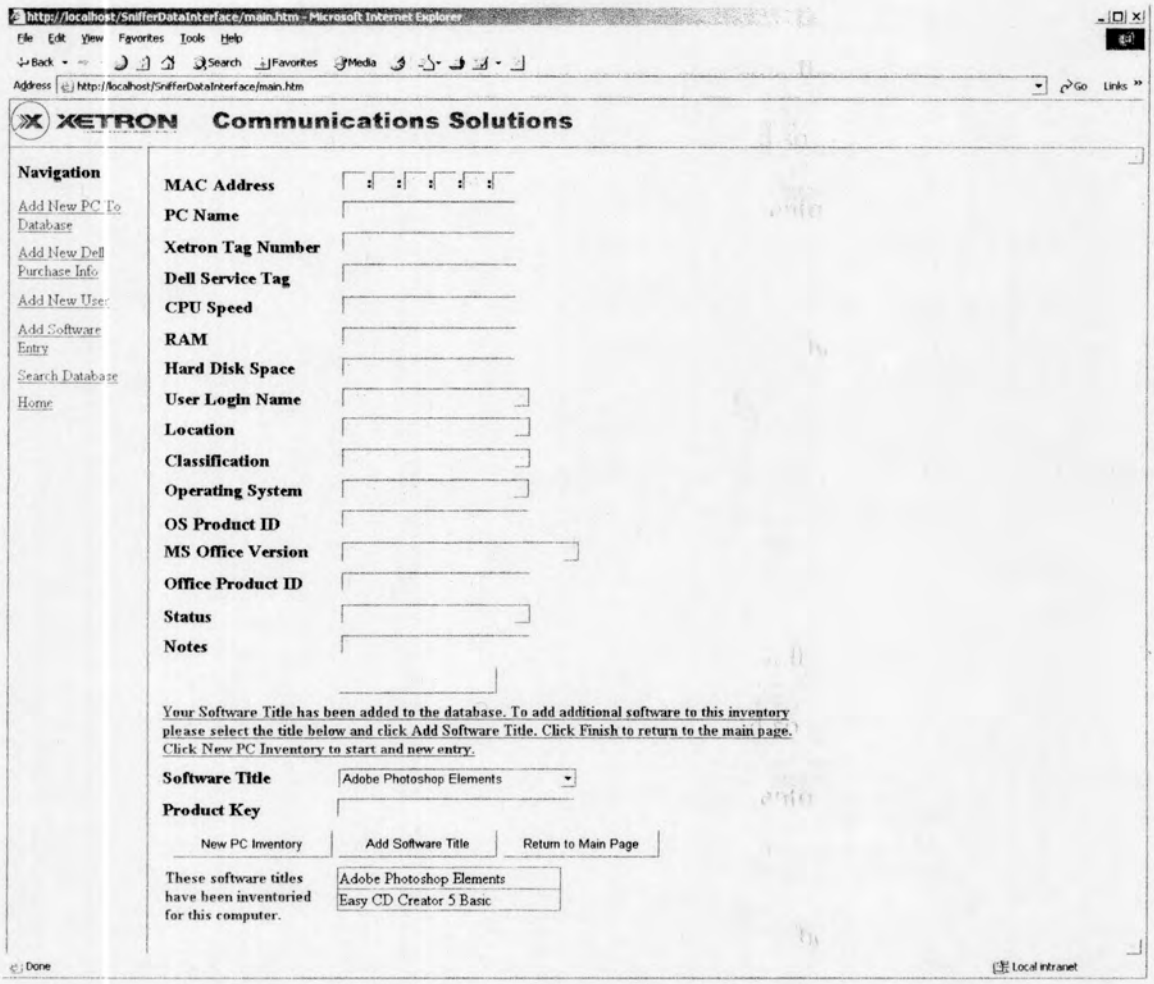


Figure 9. The Add New PC to Database page

8.2.3 The Add Dell Purchase Page

The Add Dell Purchase page allows the users to input the specifications of a new Dell purchase. The specifications include the hardware and software along with the Dell service tag. The layout of this page is similar to the PC Inventory page. It has a series of text boxes and dropdown list boxes. The dropdown list boxes are populated from the database and the textboxes are linked to validation controls. The *Submit* button will insert the new record and lock the fields. The user will have the options to enter another

Dell purchase or return to the home page. The Dell Purchase page can be seen in **Figure 10**.

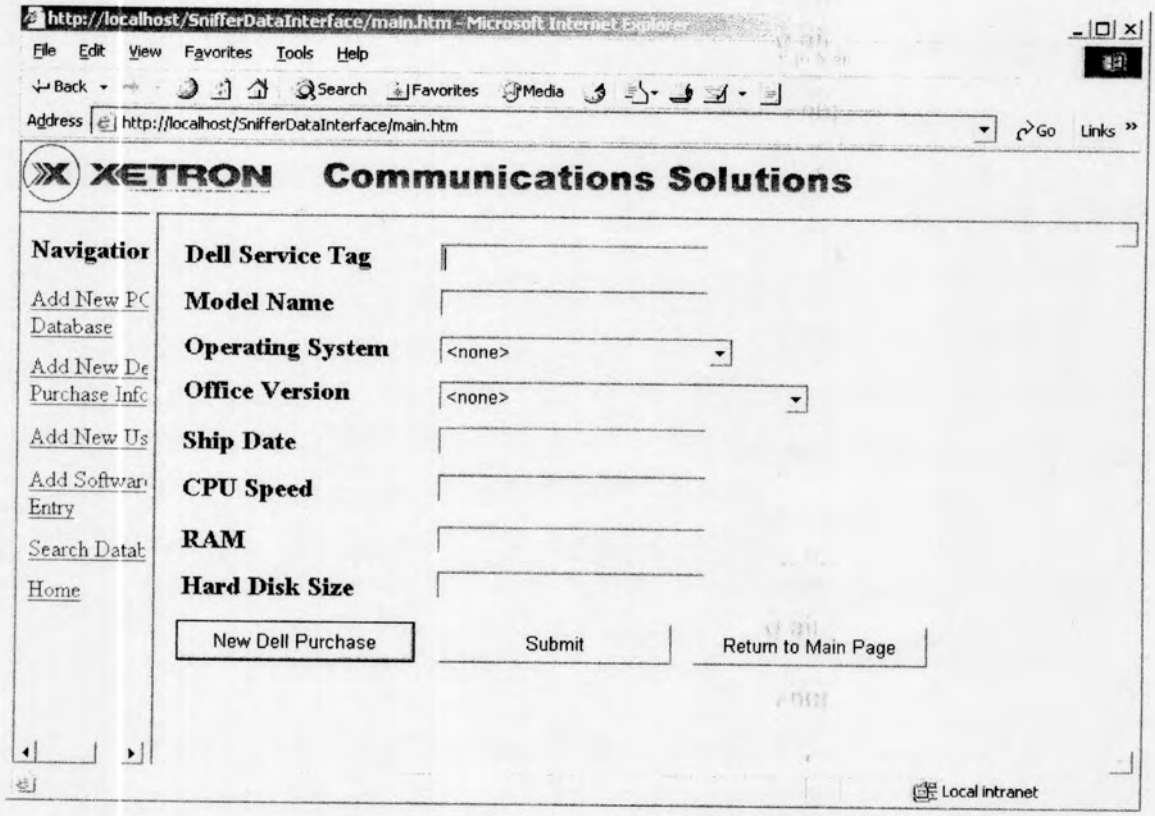


Figure 10. The Dell Purchase Page

8.2.4 The New User Page

The Add User page is designed similarly for the previous two pages. The page allows the users to insert a new user account into the database. This includes the users full name, login name, department, location and PC need. Again it has a series of text boxes and dropdown list boxes. The dropdown list boxes are populated from the database and the textboxes are linked to validation controls. The *Submit* button will insert the new record and lock the fields. The user will have the options to enter another

User or return to the home page. An example of the New User page can be seen in **Figure 11**.

Figure 11. The New User Page

8.2.5 The Software Inventory Page

The Software Inventory page allows the user to add additional software titles to a PC inventory. The layout of this page is similar to the layout of the software portions of the PC Inventory page. When the page first loads a dropdown list box loaded with the available MAC Addresses. By selecting a MAC Address the remaining controls are displayed. The PC name and the software titles that have been entered will be returned. The PC name is displayed in a label and the software titles are listed in a datagrid. A software title is added by selecting the *Add Software Title* command. When this is done, the title is added to the list and inserted into the database and the software title dropdown list is reset for the next selection. The *Return to Main Page* button can be selected at any time to return to the Home page. An example of the Software Inventory page can be seen in **Figure 12**.

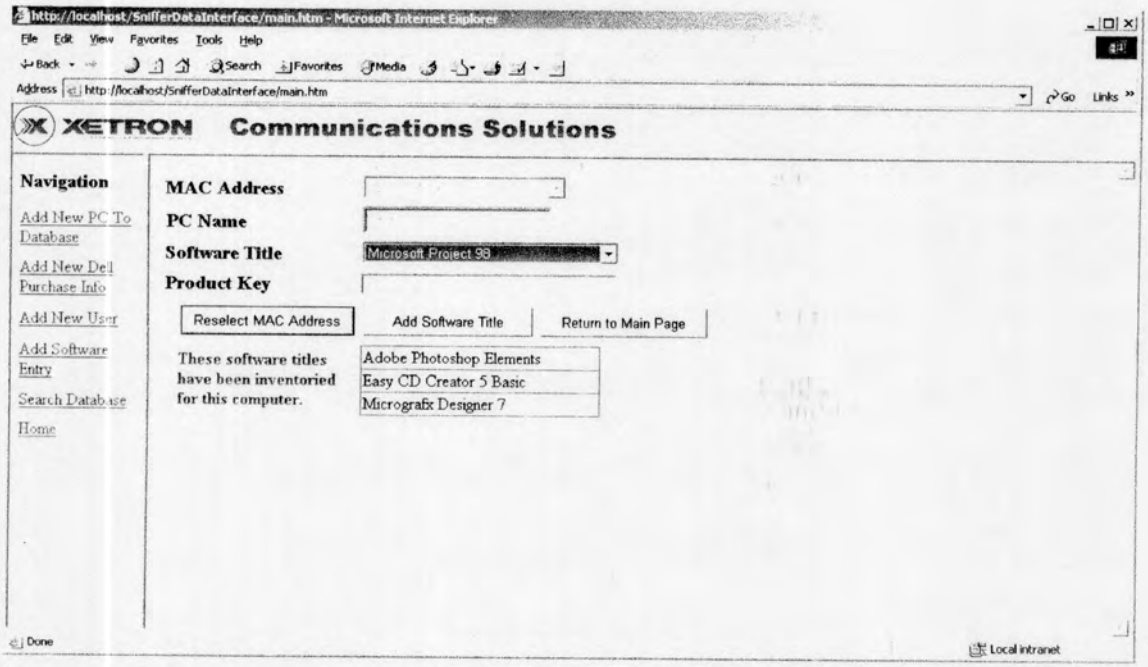


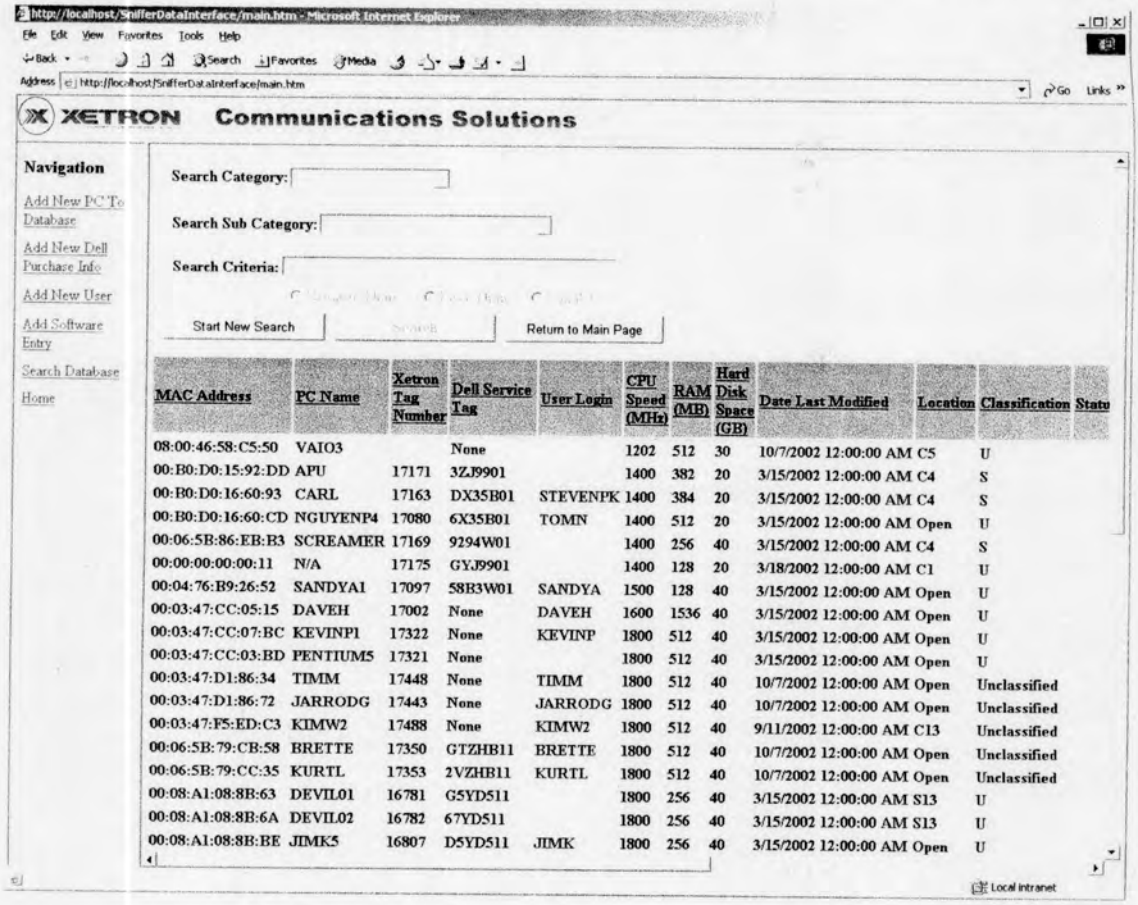
Figure 12. The Software Inventory Page

8.2.6 The Database Search Page

8.2.6.1 The Search

The Search page allows the users to query the Sniffer and Inventory data. When the page loads, a dropdown list box appears with the available categories. There are five categories to choose from; User, PC Inventory, Dell Purchase, Software Inventory and Sniffer Data. These act as main categories to search by. Once a category is chosen a second dropdown list box appears and the first is locked. The second list box contains the optional sub categories to choose from. This list of categories is loaded based on the main category. When a sub category is selected there are three possibilities. If the sub category is a numeric topic, a text box will appear along with a radio button list. The user can enter the numeric value into the text box and then select either greater than, less than

or equal to on the radio button list. These two inputs complete the search query that will be passed to the database. In **Figure 13** a numeric search has been sent to the database.



MAC Address	PC Name	Xetron Tag Number	Dell Service Tag	User Login	CPU Speed (MHz)	RAM (MB)	Hard Disk Space (GB)	Date Last Modified	Location	Classification	Statu
08:00:46:58:C5:50	VAIO3		None		1202	512	30	10/7/2002 12:00:00 AM	C5	U	
00:B0:D0:15:92:DD	APU	17171	3ZJ9901		1400	382	20	3/15/2002 12:00:00 AM	C4	S	
00:B0:D0:16:60:93	CARL	17163	DX35B01	STEVENPK	1400	384	20	3/15/2002 12:00:00 AM	C4	S	
00:B0:D0:16:60:CD	NGUYENP4	17080	6X35B01	TOMN	1400	512	20	3/15/2002 12:00:00 AM	Open	U	
00:06:5B:86:EB:B3	SCREAMER	17169	9294W01		1400	256	40	3/15/2002 12:00:00 AM	C4	S	
00:00:00:00:00:11	N/A	17175	GYJ9901		1400	128	20	3/18/2002 12:00:00 AM	C1	U	
00:04:76:B9:26:52	SANDYA1	17097	58B3W01	SANDYA	1500	128	40	3/15/2002 12:00:00 AM	Open	U	
00:03:47:CC:05:15	DAVEH	17002	None	DAVEH	1600	1536	40	3/15/2002 12:00:00 AM	Open	U	
00:03:47:CC:07:BC	KEVINP1	17322	None	KEVINP	1800	512	40	3/15/2002 12:00:00 AM	Open	U	
00:03:47:CC:03:BD	PENTIUM5	17321	None		1800	512	40	3/15/2002 12:00:00 AM	Open	U	
00:03:47:D1:86:34	TIMM	17448	None	TIMM	1800	512	40	10/7/2002 12:00:00 AM	Open	Unclassified	
00:03:47:D1:86:72	JARRODG	17443	None	JARRODG	1800	512	40	10/7/2002 12:00:00 AM	Open	Unclassified	
00:03:47:F5:ED:C3	KIMW2	17488	None	KIMW2	1800	512	40	9/11/2002 12:00:00 AM	C13	Unclassified	
00:06:5B:79:CB:58	BRETTE	17350	GTZHB11	BRETTE	1800	512	40	10/7/2002 12:00:00 AM	Open	Unclassified	
00:06:5B:79:CC:35	KURTL	17353	2VZHB11	KURTL	1800	512	40	10/7/2002 12:00:00 AM	Open	Unclassified	
00:08:A1:08:8B:63	DEVIL01	16781	G5YD511		1800	256	40	3/15/2002 12:00:00 AM	S13	U	
00:08:A1:08:8B:6A	DEVIL02	16782	67YD511		1800	256	40	3/15/2002 12:00:00 AM	S13	U	
00:08:A1:08:8B:BE	JIMK5	16807	D5YD511	JIMK	1800	256	40	3/15/2002 12:00:00 AM	Open	U	

Figure 13. A numeric search of the PC Inventory by CPU speed

If the sub category is not numeric and there is not a fixed list of options for the search criteria, a text box appears without the radio button list. In this case the value of the radio button is assumed to be "equal to." In **Figure 14** a search is performed for non numeric criteria.

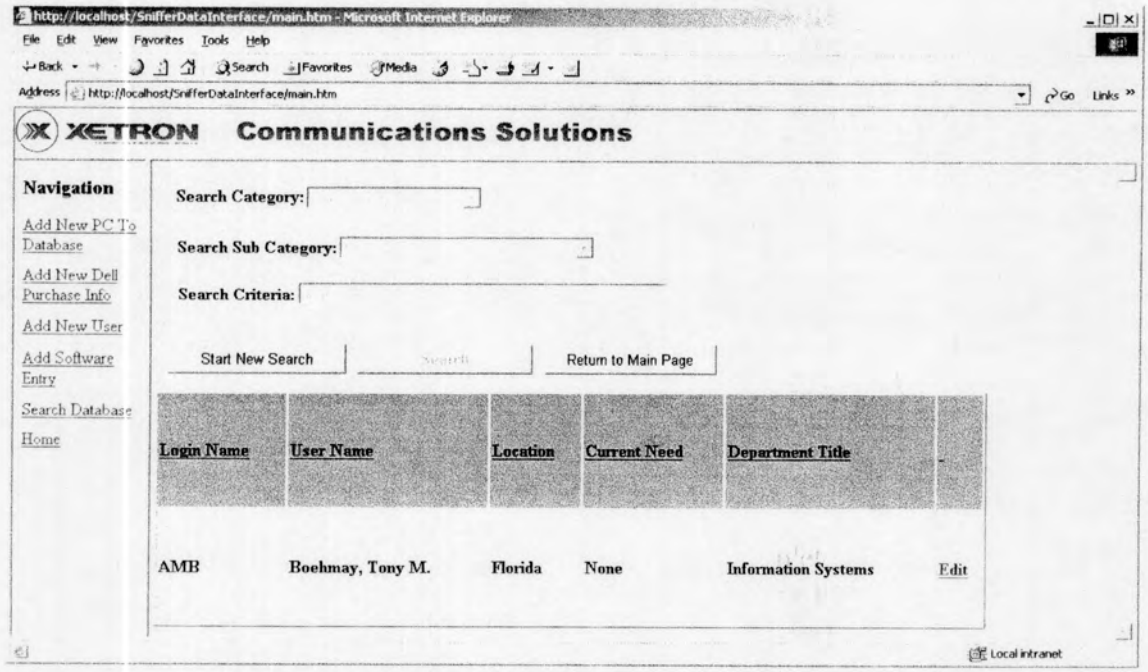


Figure 14. A non numeric search of the User inventory by Login Name

If a subcategory is selected that permits a fixed list of criteria, a dropdown list box is displayed and loaded with the available options. A case statement determines which options are loaded. This interface is the preferred option because it eliminates typographical errors and guarantees the user that a valid criterion has been passed to the database. The user can select a criterion from the list to complete the search. In **Figure 15** a fixed list subcategory has been selected.

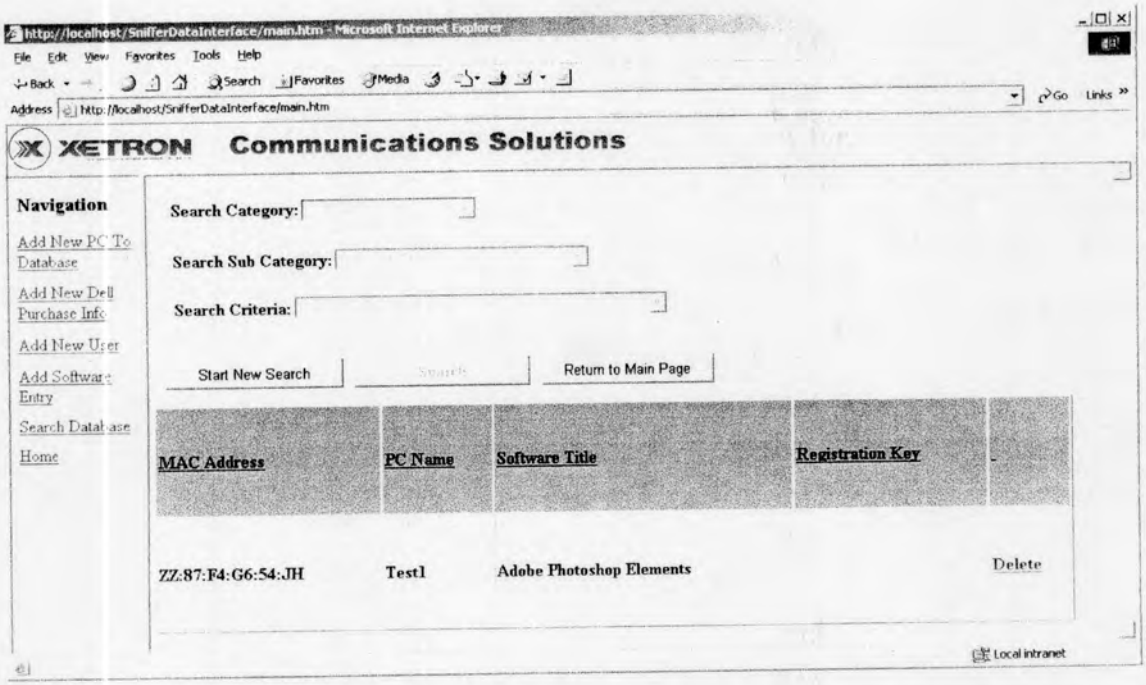


Figure 15. A search based on a fixed criteria sub category for Software Inventory by Software Title

Once the query has been sent to the database, the criteria options for the search are locked and new options are displayed along with the results. Navigation buttons are displayed to allow the user to return to the Home page and to begin a new search. The results of the query are displayed below these navigation buttons.

8.2.6.2 The Results

The results of the Search page are displayed in a datagrid at the bottom of the page. There is a unique datagrid for each of the five main categories. The datagrids on this page are more complex than those from previous pages. In addition to the ability to sort the datagrids by selecting the column heading, these datagrids provide the user the ability to update and delete records that have been returned. An additional column is added to each grid with links to edit and or delete data in the viewable records. The added functionality requires the use of templates for the datagrid controls. With a

template in place, the datagrid no longer auto generates columns. Each column is defined. These definitions include template columns, bound columns, edit templates, item templates and item style. This is how the datagrid knows how to react when the edit option is selected.

8.2.6.3 The Edit Option

The edit template determines how the selected record will appear when the user selects the *Edit* link in the datagrid. When an edit command is selected a combination of text boxes and dropdown list boxes appear within the record on the datagrid. An example of this can be seen in **Figure 16**.

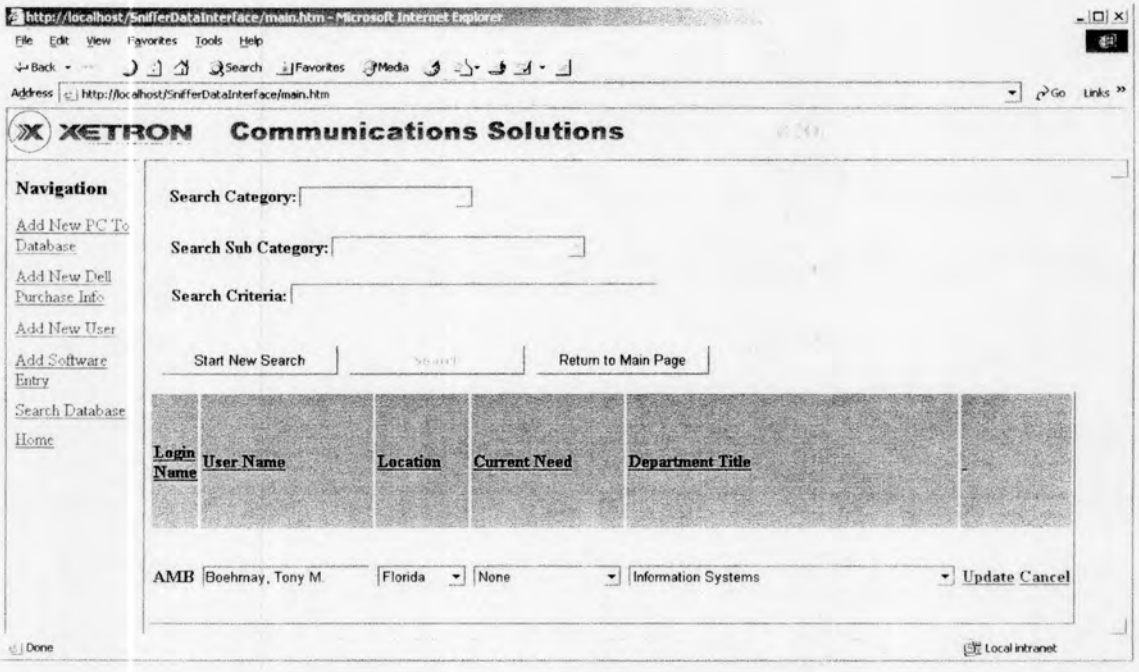


Figure 16. The edit options available for the User record from Figure 14

In the above example, the edit option has been selected and a series of text and dropdown list boxes have been displayed to allow the user to change the values in each field. In the right hand column, the Edit command has been replaced by Update and Cancel options.

The update command sends a SQL statement to the database to change the effected record, reflecting the changes made in the datagrid. The datagrid is than rebound to the data source to confirm that the changes have been made in the database and the datagrid returns to read only mode. The *Cancel* command returns the datagrid to read only mode without sending any updates to the database.

8.2.6.4 The Delete Option

When conducting a search of the software inventory the user has the option to delete records rather than edit them. In this case there is a delete link column. When this link is selected a SQL delete command is sent to the database to delete the selected record. Before the delete command is completed, a confirm/cancel message box is displayed to give the user a chance to change their mind or to catch a mistake before it happens. An example of this option can be seen in **Figure 17**.

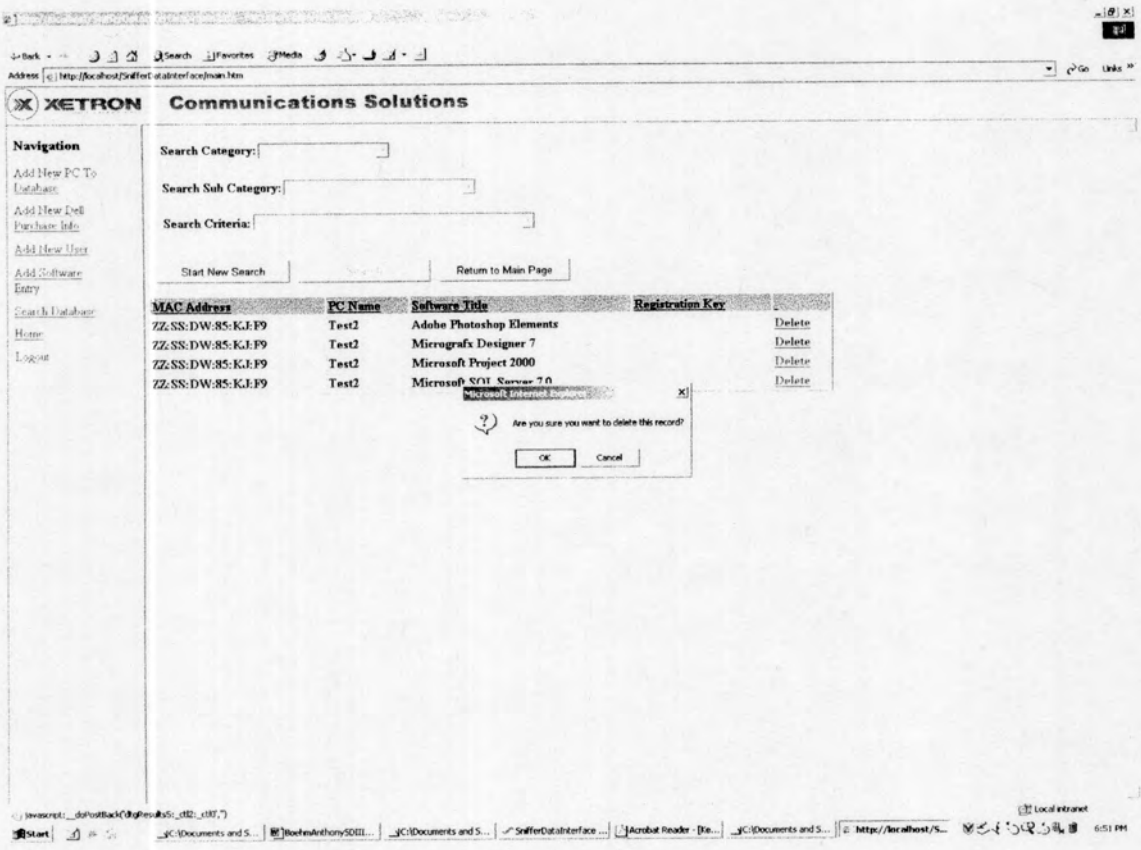


Figure 17. The delete option in a Software Inventory search

8.2.6.5 Data Validation in the Datagrids

An additional feature of datagrid templates is the ability to include data validation controls in the edit item template. Each text box is linked to a regular expression validation control that ensures that the characters that are entered and the format they are in are consistent with the data integrity. In **Figure 18** there is an example of a validation control in a datagrid.

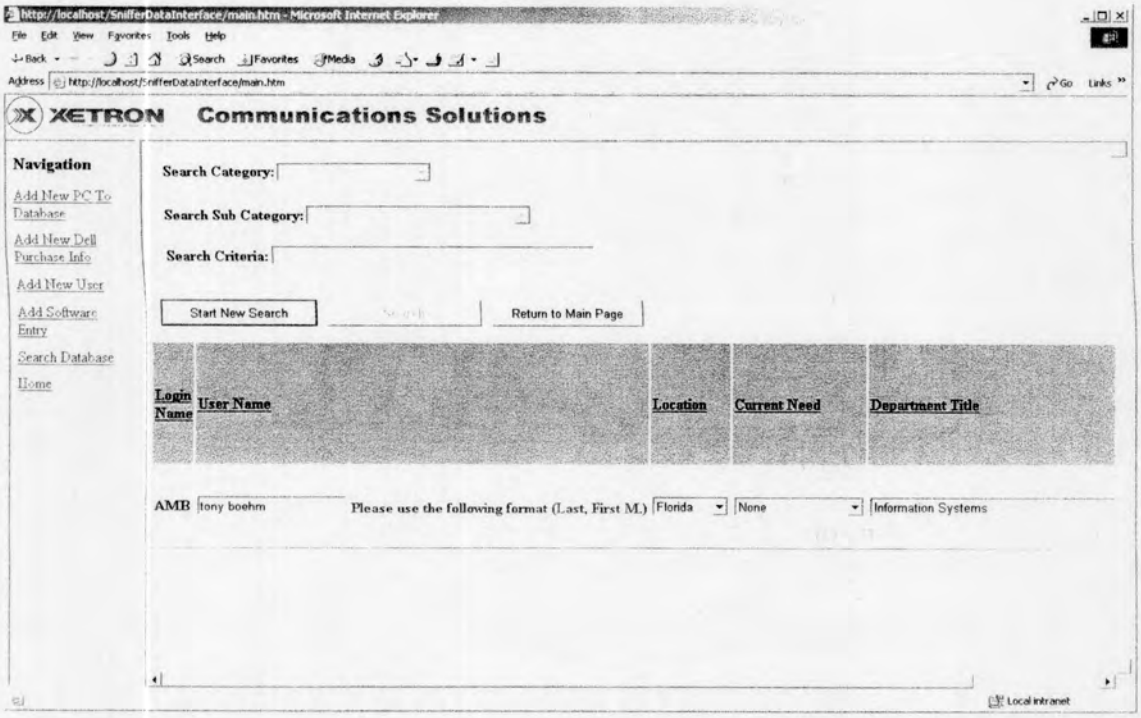


Figure 18. Datagrid regular expression validation control for the User Name field

8.2.7 The User Login Page

The User Login page requires the user to input a login and password before allowing the user access to the data the site provides. When the site loads, the login page is locked into the main frame. The user is not allowed to move to any other areas of the site until he/she provides a correct login. When a correct login is entered, a session variable records a "true" value. Every page checks the value of this variable when loading. If the variable is not "true" then the user is forced back to the login page. This prevents a user from circumnavigating the login page. If the user enters an invalid user name or password, an error message is displayed informing the user to try again. The user can log out of the site by clicking the *Logout* link in the navigation bar. When a user leaves the site by either logging out or closing Internet Explorer, the session variable is

closed and the user is forced to login before accessing additional pages. The login page can be seen in **Figure 19.**

Figure 19. User Login page with incorrect user name and password

9. Testing Procedure

The testing procedure involved several different stages. Alpha testing was conducted by myself after the completion of each page. This testing was conducted on the local host server in the development environment. When I was satisfied with the results of my testing I opened testing up to several different groups.

For group testing I modified the user environment. I setup a local area network at my home with three machines in addition to the development server. Test groups evaluated the web site in groups of three, accessing the site across the network. These tests permitted me to observe how the site handled multiple users. It also allowed me to see what type of performance hits was encountered when the database was accessed across the network. The group tests pointed out several errors that my individual testing

had not caught. The errors were corrected and additional testing was performed. After each round of testing, corrections were made and consideration was given to any suggestions the users offered.

The testing process confirmed that the site was functional and user friendly. Each user was given the opportunity to navigate the site without any initial instructions. Most users were able to enter, search, edit and delete records without any help. Observations, along with feedback I received through the questionnaires I provided, confirmed the site achieved the goal of being user friendly.

10. Conclusions and Recommendations

10.1 Conclusions

The project was developed to meet the information need of Xetron's Information Systems department. The internal website provides access to a combination of sniffer and inventory data. The interface was developed using ASP.NET and VB.NET while the database was designed using SQL Server 2000. The data was imported using SQL Server's Data Transformation Services. The project was completed over the course of sixteen months. The budget of $11,588.99 would have covered all development including estimated labor hours. The project meets all design freeze deliverables. Testing was done to ensure functionality, and areas for future development were established.

10.2 Recommendation

While developing my project I came across several challenges. Most of these challenges involved the use of ASP.NET and the changes upgrading to VB.NET. The ASP.NET datagrid control is a very functional tool. It offers a wide range of data display

and manipulation options, but it is also very challenging. I found it difficult to find useful resources that explained how to implement different functions. I recommend gaining a strong understanding of the functionality of the datagrid template and the options that it can enable. The only way I found to enable the editing and validation controls in the datagrid was through the templates. Without these template controls the datagrid is nothing more that an advanced table.

I recommend that when using SQL Server DTS to import Access databases, be sure to perform all needed data scrubbing in Access first. SQL Server complicates instances where you need to make real world data compatible with the rules setup for the database. Also be prepared to recreate all relationships within SQL Server. These are not imported and, as I found out the hard way, can be very time consuming to rebuild.

VB.NET is more powerful than its predecessor. However, it does take time to master. Visual Studio.NET makes typing code much easier. Unfortunately, if you have to work in the HTML view, which I did often, it tends to throw away formatting changes. I noticed that when I returned to a saved web form, all the HTML code was run together in several continuous lines rather than formatted nicely as I had saved it. This may be something Microsoft will fix with a later patch.

Overall the learning curve on ASP.NET and VB.NET was not too steep. I do warn that as always, first edition development books are full of bugs and it may be difficult to find solutions for some of these. The "or" control for the regular expression validation controls was very frustrating. I was unable to get these to work the way they were shown in the book and I had difficulties finding documentation of the topic online.

Overall I recommend the products I used and wish success to all those who use them in the future.

References

1. Alverson, Dave. Senior Engineer, Xetron. First personal interview. January 23, 2002

2. Case, Richard and Billy Hollis and Narkiewicz, Jan D. *Professional VB.Net.* Birmingham, UK: Wrox Press, Inc. 2001

3. Chaffin, Mark and Knight, Brian and Robinson, Todd. *Professional SQL Server 2000 DTS (Data Transformation Service)*. Birmingham, UK: Wrox Press, Inc 2000.

4. Larsen, Diane and Garden, Euan. *Data Transformation Services (DTS) in Microsoft SQL Server 2000.* http://msdn.microsoft.com/library/default.asp?URL=/library/techart/dts_overview.htm, January 28, 2002

5. Payne, Chris. *Sams Teach Yourself ASP.NET in 21 Days.* Indianapolis, Indiana: Sams Publishing 2002

6. Steele, Dave. Department Head, Information Systems, Xetron. First interview. January 18, 2002