

Bridge Worldwide -- Software Library

By

Janet Crawford and Dam Luong

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

May 2002

Bridge Worldwide -- Software Library

By

Janet Crawford and Dam Luong

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 2002 Janet Crawford and Dam Luong

The information in this document is proprietary and may not be reproduced or distributed
in whole or in part without the permission of the owner.

Janet Crawford

Date

Dam Luong

Date

Robert Schlemmer, Faculty Advisor

Date

Lawrence G. Gilligan, Department Head

Date

Acknowledgments

I wish to thank my advisor, Professor Robert Schlemmer, for his support throughout my career in the Information Engineering Technology program. I would also like to thank my husband, Shawn Crawford. Without his constant support and understanding, my educational career would not have blossomed. I would like to thank Mr. Jim Elkin of Bridge Integration Company for his continued support throughout the project. Finally, I would like to thank my family for their constant support in my education.

Janet Crawford

In appreciation to my parents and family members, who always supported and encouraged me to overcome the difficulties in a new land. I would like to thank Mr. Jim Elkin, Mike Todd, and all Bridge's employees for their help in the creation of this project. Thanks also to my project partner for all her hard work, and thanks to the teachers that took time from their busy schedules to answer my many questions.

Dam Luong

Table of Contents

Section	Page
Acknowledgments	i
Table of Contents	ii
List of Figures	iii
Abstract	iv
1. Statement of the Problem	1
1.1 Definition of the Need	1
2. Description of the Solution	2
2.1 User Profile	7
2.1.1 Users	7
2.1.2 Administrators	7
2.2 Design Protocols	7
2.2.1 Database	7
2.2.2 Programming	8
2.2.3 Networking	8
2.2.4 Multimedia	8
2.2.5 Interface Design and Navigation	8
2.2.6 Icons and Graphic Symbols	8
2.2.7 Help	8
3. Deliverables	9
4. Design and Development	11
4.1 Budget	11
4.2 Timeline	11
4.2.1 Senior Design I	11
4.2.2 Senior Design II	12
4.2.3 Senior Design III	13
4.3 Software	13
4.4 Hardware	13
5. Proof of Design	14
6. Conclusions and Recommendations	28
Appendix A. ASP Code	29
Appendix B. HTML Code	50
Appendix C. Chat Room Code	60
Appendix D. References	75

List of Figures

Figure 1. Hardware requirements and prices	11
Figure 2. Software requirements and prices	11
Figure 3. Three-Tier architecture description	14
Figure 4. Relationships of database	15
Figure 5. Home page	17
Figure 6. Sending an e-mail	18
Figure 7. FAQ section	19
Figure 8. Entering a chat room	19
Figure 9. Signing into a chat session	20
Figure 10. Chat room	21
Figure 11. Administrator utility and function	22
Figure 12. Editing and updating personal information	24
Figure 13. Search results	24
Figure 14. Changing and updating records	25
Figure 15. Adding a record	25
Figure 16. Deleting records	26
Figure 17. User utility or function	26
Figure 18. Updating personal information	26

Abstract

This Web-based inventory-tracking system has been designed to improve BIC's ability to track and report software and hardware inventory to their employees. In the past, employees have been responsible for tracking their usage of software and their workstation hardware by using one Microsoft Excel spreadsheet. This creates problems with the integrity of the data entered, as well as the data being unreliable, decentralized, and unprotected. In addition, the spreadsheets do not allow management to effectively monitor software installation, hardware replacements or upgrades, or compliance with software licensing agreements. The Web-based technology used in this application is reliable and simplifies the gathering of information.

Inventory Tracking System for Bridge Integration Communication (BIC)

1. Statement of the Problem

1.1 Definition of the Need

Bridge Integration Company (BIC) is primarily a marketing communication consultant for the creation and development of Web sites. BIC currently handles its in-house software and hardware inventory by maintaining several, shared Microsoft Excel worksheets. All employees of BIC have access to this information and can manipulate the data at any time, potentially corrupting the data and leaving it vulnerable to viruses. This presents a significant problem from a management standpoint, as the data is unreliable, decentralized, and unprotected. In addition, the current procedure does not allow management to effectively monitor software installation, hardware replacements or upgrades, or compliance with software licensing agreements.

Pursuant to discussions with Mr. Jim Elkin, Chief Technology Officer for BIC, the current database is insufficient to provide meaningful decision-making information to senior management. We have identified three significant problems with the existing architecture: (1) The internal tracking of software and hardware inventory is not performed by a designated administrator; (2) The data is not centralized; and (3) BIC is unable to calculate actual costs incurred by the company's Information Technology department.

2. Description of the Solution

We propose to implement, in part, a single database that will help resolve these deficiencies. The database's primary owner will be the network administrator. The network administrator will be providing support to users that are requesting a hardware or software change. From the company Intranet, users will be able to complete a form that will be sent to their manager via email. Once these upgrades or changes are approved, the network administrator will be notified of the request. The network administrator will then be responsible for inputting these changes. This will also allow management to have more accurate data about hardware and software usage, will provide a tool for analyzing budget and inventory, and will serve as a blueprint for revised internal controls and procedures.

Since BIC has a dynamic dataset there should be one primary owner who will be responsible for entering changes to the database. We plan on using Microsoft Access 2000 to design the database. By using a Relational Database Management System (RDBMS) such as Microsoft Access, we will be able to design a data-entry form or screen in order to make the individual data-entry task as efficient and accurate as possible. To store BIC's hardware and software inventory, we have chosen to use an RDBMS to manage all of the data in tables. These tables store information about a subject (for example, hardware) and have columns that contain the different kinds of information about the subject (such as what software performs most efficiently for this hardware) and rows that describe all the attributes of a single instance of the subject. Microsoft Access 2000 is a fully functional RDBMS. It will provide all the data definition, data manipulation, and data control features we will need to manage large

volumes of data (1, pp.45-49). A RDBMS is designed to maintain data integrity while permitting updates from one or more user. In addition, RDBMSs have security features, which can be used to identify users by a username and password and selectively allow or restrict access to different portions of the data to the database (2, pg.14). By securing this database and having one administrator, this will create a more efficient, manageable database.

Creating one centralized database to input software and hardware inventory information will assist management with reports and auditing that may need to be performed on a weekly or monthly basis. Microsoft Access 2000 will allow the administrator the opportunity to query on information contained in the database. Management may request these reports to determine the frequency of software upgrades that are taking place, replacing hardware as determined by the lease agreements, as well as particular users that are requesting these changes or additions.

We have chosen to use Microsoft Visual Basic 6.0 (VB6) to create the data-entry forms required to input data into the database. These data-entry forms will also be available on the company Intranet. To create an on-line software/hardware request form, we have chosen VB6 because of the Internet and Intranet enhancements that have been made from the previous versions of VB. VB6 includes new Internet and Intranet features, including the capability to create Dynamic Hypertext Markup Language (DHTML) applications with VB. We will also be using the Component Object Model (COM) to create templates or components for reuse in our project (3, pp.124-126).

The Web site programming will consist of Dynamic Hypertext Markup Language (DHTML). To create efficient, dynamic code on the web site, we have selected to use

Dynamic Hypertext Markup Language (DHTML). While traditional HTML is static, the use of DHTML allows the designer increased flexibility and greater control over the appearance of pages. With DHTML, a web page can react to the user without relying on the server or depending on an embedded program. DHTML keeps any element on the page, at anytime after the page has downloaded, and changing its appearance, content, or location on the page. This enables web pages to act much more like computer programs and interactive CD-ROMs rather than as the static book pages that we have been limited to for so long (4, pp.32-45). For programming efficiency, we have chosen to use Active Server Pages (ASP). Active Server Pages are server-generated pages, which can call other programs to do things like access databases and serve different pages to different browsers. ASP is almost as efficient as writing code directly to the server's application programming interface (5, pg.19).

We have decided on these technologies because of their efficiency, user-friendly interfaces, and their relatively low cost and maintenance. These technologies are easy to upgrade and they work well with minimum hardware requirements.

Our research of the series of Microsoft Excel spreadsheets shows that BIC lost approximately \$10,000.00 worth of software in 2000. The design and content of the database will help not only in preventing software loss but also in protecting against the overuse of the software key provided with licensed software packages.

Our Web site will be available to BIC employees with access to the Internet. The site provides information regarding BIC's hardware and software inventory. Our project was designed using a Web site, a database, and several Visual Basic forms for interfacing with the database.

Our Web page provides an internal location for information regarding BIC's inventory. The home page includes links to BIC's home page, EPMT (for payroll and project management), an e-mail link for additional support, a Frequently Asked Question (FAQ) link, a chat link designed for users and administrators to chat at designated times, a link for new users to register, and login boxes for users and administrators. Depending upon the login profile, members of the site are able to access Department/Team Members, Server Information, Employee List, Software List, History Log, Palm Pilot Information, IP and Sites List, Printer List, Server Space, Personal Information, and Report Administrator. Jim Elkin and the Network Administrator suggested the links that were created on the Web site. The links contain the following fields and information:

- Department/Team - Department ID (AutoNumber) and BIC's department/team names.
- Server Information - Server ID (AutoNumber), Server Name, IP, NOS, CPU Speed, and RAM
- Employee List – Employee ID (AutoNumber), First Name, Last Name, Phone Extension, Cell Phone, Department ID, Logon and Password.
- Software List – Software ID (AutoNumber), Software Number, Software Name, Version, Software Type, Serial Number, and Date Received.
- History Log – Login name, time, and date.
- Palm Pilot – Palm ID (AutoNumber), Serial Number, Employee ID.
- IP and Sites List – Web site ID (AutoNumber), DNS, IP, Server ID, Drive ID, Home Directory, and Status.

- Printer List – Printer ID (AutoNumber), Printer Name, Locations, Path, IP, Server ID.
- Server Space – Drive ID (AutoNumber), Server Name, Drive Letter, Capacity, Unit, Free Space, and Free Unit.
- Personal Information – Password, Password Hint, First Name, Last Name, Cell Number, E-mail Address.

The database has all information for the links. The information that is stored in the database includes tables and records. The database will also provide information to employees by using Report Administrator. Report Administrator is designed to allow employees to create reports based on query information. These reports can be printed and/or saved to the user's machine. Report Administrator will only allow queries against existing information. Users will not be able to save generated reports to the database. The search function was designed so that each member of the site can query on fields contained in the database. Searching will be restricted by access to certain links. Depending upon the member's access, only tables that they have read/write capabilities will appear. For example, an administrator could search any table contained in the database. If the administrator requests information about two servers and their memory, he or she could search for that information.

The Visual Basic forms will only be available to administrators of this site to update existing information. The form will allow the administrator to choose existing tables or records to update and also create new tables. We have created simple forms for the administrators. They will enter the required information, click SAVE, and information will be updated.

2.1 User Profile

The intended users of this site will have a moderate level of information technology experience. We have identified two types of users for this site. The two types are Users and Administrators.

2.1.1 Users

- Users of this site will have access to departments/teams, editing personal information, and requesting new hardware and software.
- Users will have read-only access to some portions of the site.
- Users may be able to run reports requested by their manager(s). This will be established in the user profile (by login identification).
- Users will be responsible for registering for access to the site.
- Users will be responsible for reporting any problems of the site to the administrators.

2.1.2 Administrators

- Administrators will provide all updates to the site.
- Administrators will support, monitor, test and troubleshoot site problems.
- Administrators will provide any needed training for the site.
- Administrators will be responsible for backup, recovery, and hardware maintenance.
- Administrators will provide any updates to the database and create new tables as necessary.
- Administrators will be responsible for handling any e-mail received for user assistance.
- Administrators will be responsible for the chat session(s).

2.2 Design Protocols

We have used all four components of the Information Engineering Technology program. The different components of the IET program and how they will be incorporated into our project are listed below.

2.2.1 Database

Our database was designed using Microsoft Access 2000 and Microsoft SQL 7.0. We created fifteen tables to interact with the Web site. We also imported current records

stored in BIC's Microsoft Excel spreadsheet. We designed the relationships using many-to-many modeling.

2.2.2 Programming

We used ASP pages, JavaScript, and Visual Basic to design the user interface, ODBC accessibility, and interfacing with the database.

2.2.3 Networking

BIC has a hybrid network consisting of UNIX, Windows, and Apple. Our project is capable of providing services to all of these platforms. Our project must encompass the total enterprise. We designed a network that matches BIC's network and software applications, using two Windows 2000 servers.

2.2.4 Multimedia

The design of the Web site was to provide uniformity with BIC's company Web sites. There are no large graphics or animation. The goal of this site was to provide simple user navigation.

2.2.5 Interface Design and Navigation

The interface is based upon BIC's Intranet Web site. We designed text boxes and buttons that would be familiar to employees.

2.2.6 Icons and Graphic Symbols

We are using BIC's color scheme, which is white background, black text, blue symbols and BIC's corporate logo.

2.2.7 Help

Help will be available throughout the site. It will be a link that is clearly marked for the employee's convenience.

3. Deliverables

- The Web site and database reside on our Windows 2000 Web server. All code, databases, and material pertaining to this project will be given to BIC at the end of Senior Design III.
- In addition to the Web server that houses our data, we have another Windows 2000 Web Server for load balancing and backup.
- The user interface is written in ASP and JavaScript, which is easy to update.
- Our Web site uses ASP pages to communicate between the client and database.
- The Web site is designed to run on all browser versions and does not require any specific operating system.
- The Web site is easily accessed from a computer with an Internet connection.
- The Web site's design is based on the company's color scheme, graphic symbols, font, interface design, and navigation.
- Create a FAQ link that will provide information regarding applying for access, changing a password, e-mailing for further assistance, queries, searching, and printing reports.
- Develop a chat site that will enable users to login and chat with an administrator. The chat session will enable Administrators to view the user ID, login time and date, and the user's IP address of their workstation.
- Users of the site will be able to print reports from the search function.
- Administrator and User login. Users of this site will login with their correct profile.

Administrators will have access to the entire site, including editing rights to the items below. Administrators will also be able to search on existing items in each link.

- Department and Team Information.
- Server Information.
- Employee List.

- Software List.
- History Log.
- Palm Pilot Information.
- IP and Sites List.
- Printers List.
- Server Space.
- Personal Information.

Users will only have read access to the items below. Users will also be able to search on existing items in each link.

- Department Information.
- Employee List.
- Software List

4. Design and Development

4.1 Budget

Hardware Requirements	
Server	\$749.00
Network Cards	40.00
Hub (4-port hub connection)	50.00
Internet Provider (monthly for 9 months/\$39.00)	351.00
<hr/>	
Total Hardware Cost	\$1190.00

Figure 1. Hardware Requirements and prices

Software Requirements	
Windows 2000 Server	\$561.51
Visual Studio	490.00
Firewall (Security Protection)	150.49
Microsoft Office 2000 Professional	450.00
PhotoShop	350.00
Norton AntiVirus	560.95
Total Software Cost	\$2,562.95
<hr/>	
Total	\$3,752.95

Figure 2. Software Requirements and prices.

Prices taken from <http://eshop.msn.com/category.aspx?catid=6>

4.2 Timeline

4.2.1 Senior Design I

September 27 – October 11

1. Initial idea; interview with Jim Elkin and the network administrator at BIC.
2. Research Access 2000 as a front-end and back-end of database.
3. Continue research on ASP and web page design.

October 11 - 25

1. Complete research on Access 2000.
2. Begin reports for statistical use.

October 25 – November 22

1. Progress Report 2
2. First draft of proposal
3. Server set up and software installation.
4. Database lay out.

November 22 – December 6

1. Final proposal and presentation

4.2.2 Senior Design II

January 25 – February 8

1. Progress report and continuing update of ASP files for personal/software info and online report(s)
2. Testing and fix web form errors.
3. Meeting with Jim Elkin, Jeff Webber, and Mike Todd for testing input and color schema. Decisions of what information users can view/edit.

February 9 – 28

1. Redesign user login to display profile.
2. ASP for updating history log for both Admin/User log in.
3. Design and insert the BIC logo.
4. Searching/Sorting Functions (ASP).

January 25 – February 8

1. Progress report and continuing update of ASP files for personal/software info and online reporting.
2. Testing and fix any web form errors.
3. Meeting with Jim Elkin, Jeff Webber, and Mike Todd for testing input and color schema. Decisions of what information users can view/edit.

February 9 – 28

1. Redesign admin login to display profile..
2. ASP for updating history log for both Admin/User log in.

March 1 - 14

1. Adding missing software and current licensing table in the database.
2. FAQs page with the help content.
3. Creating test user for multiple testing environment.
4. Reports and presentation at the end of Senior Design 2.

4.2.3 Senior Design III

March 25 – April 12

1. Implementing Web Site and fix any errors.
2. Checking history log, cleaning unwanted (testing) user in database.
3. Implementing Database with front-end forms.
4. Meeting with BIC for the final addition of information.

April 13 – May 3

1. Testing site.
2. Compiling VB and related files.
3. Proof of Design work on proposal and formatting.

May 4 – May 31

1. Continue documenting project and formatting final document.
2. Final Presentation.

4.3 Software

- Visual Studio
- Microsoft Office 2000 Professional
- Adobe Photoshop
- Norton Antivirus
- Firewall
- Windows 2000 Server

4.4 Hardware

- Server
- Network Cards
- Hub
- Internet Provider

5. Proof of Design

This inventory-tracking project was designed based on the idea of a three-tier architecture. The client computer or graphical user interface (GUI) continuously communicates with server. The server then communicates with the database. The relationship between the GUI, server, and database can be seen below in Figure 3.

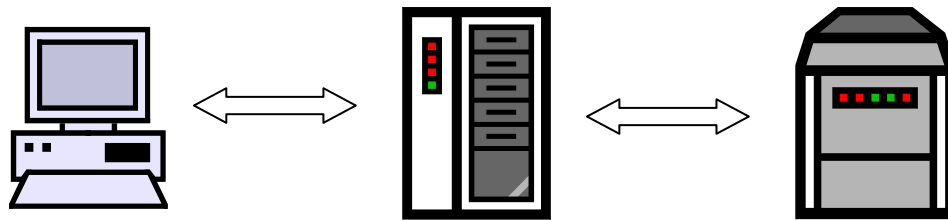


Figure 3. Three-Tier architecture description

The Web site and database reside on our Windows 2000 Web server. All code, databases, and material pertaining to this project will be given to BIC at the end of Senior Design 3. In addition to the Web server that houses our data, we have another Windows 2000 Web Server for load balancing and backup. The user interface is written in ASP and JavaScript, which is easy to update. Our Web site uses ASP pages to communicate between the client and database. The Web site is designed to run on all browser versions and does not require any specific operating system. The Web site is easily accessed from a computer with an Internet connection.

Within the database, there are twelve tables which store the data related to the project. The tables, along with primary and secondary keys and their respective relationships can be viewed in Figure 4.

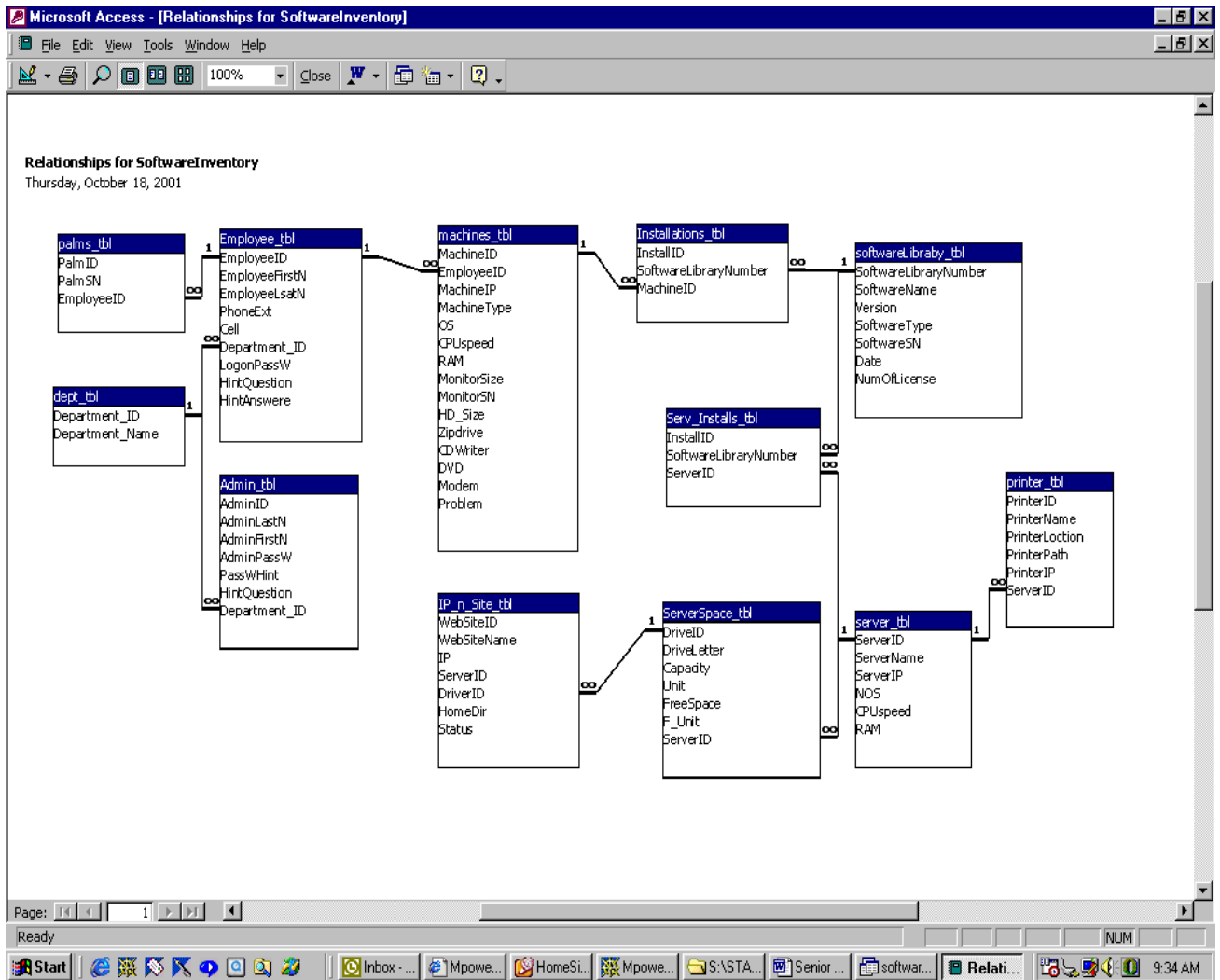


Figure 4. Database Tables

- Palms Table – Stores information about employee Palm Pilots
- Department Table – Stores information about departments within BIC
- Employee Tables – Stores information about users of the site
- Administrator Table – Stores information about administrators of the site
- Machine Table – Stores information about workstations at BIC
- Site and IP Table – Stores information about Internet Protocols within BIC and BIC sites
- Installation Table – Stores information about all hardware and software installations

- Server Installation Table – Stores information about software and hardware installed on servers
- Server Space Table – Stores information about space on BIC servers
- Software Library Table – Stores information about registered software in BIC
- Server Table – Stores information about servers at BIC
- Printer Table – Stores information about printers at BIC

The following screen shots show the interface, layout, and navigation of the site. The Web site's design is based on the company's color scheme, graphic symbols, font, interface design, and navigation. The users of the site will not create their own login accounts; accounts will be created at the discretion of the management staff and the site administrator(s). Upon receiving a login and password, employees will have access to the site. If the user enters an incorrect login identification or password they will be taken to an error page and asked to login again. Figure 5 shows the home page that contains the Administrator login, User login, and an e-mail link for support, a FAQ link, and the Chat link.

Administrator's Login	
UserName	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="Login"/>	

Bridge User's Login	
UserName	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="Login"/>	

Forgot password? Send an [e-mail](#) or call support team 0300, for more information read [FAQs](#)
Or click [here](#) go to admin chatroom.

Figure 5. Home page

Figure 5 also shows other options available to users of the site. First, clicking on [Send E-mail](#) will send e-mail from Microsoft Outlook to the site administrator. You will use this if you have a general question about the site or if you forget your user name or password. Clicking on [FAQs](#) will list frequently asked questions regarding the site and its contents. Clicking on the [chat room](#) will allow you to chat with the site administrator at certain times.

The e-mail link is available for users of the site to send an e-mail message to the support staff. When the user clicks on the link, it automatically opens the default e-mail package installed on the workstation, as shown in Figure 6.

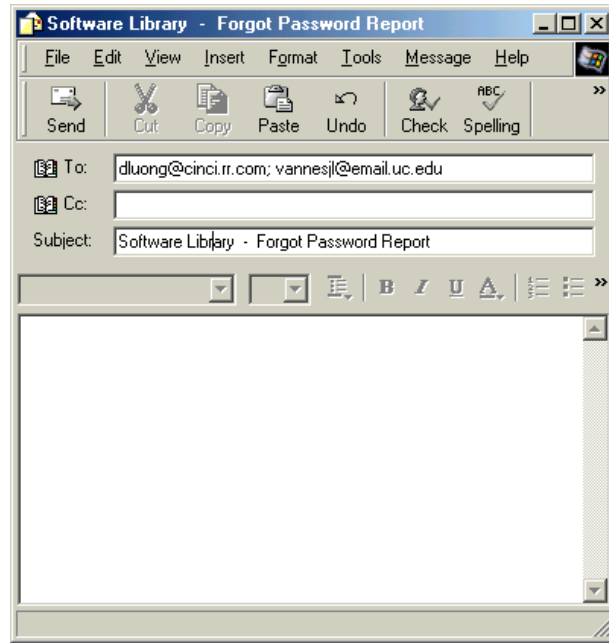


Figure 6. Sending an e-mail

A Frequently Asked Questions, or FAQ, section is available on each page in the site. When a user clicks on FAQ, information regarding applying for access, changing a password, e-mailing for further assistance, performing queries, searches, and printing reports. Figure 7 shows a list of questions contained in the FAQ section. Once users are finished, they may simply close the window.

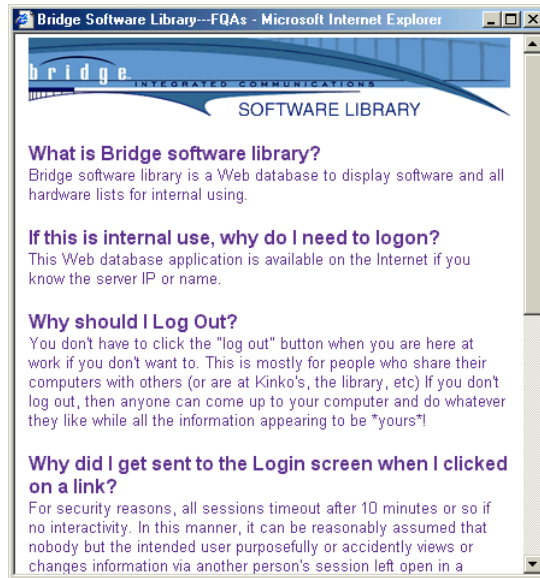


Figure 7. FAQ section

A chat session has also been developed for users. When the user clicks on chat, Figure 8 shows how to enter a chat room session. Users will then click the Enter button to enter the chat session.

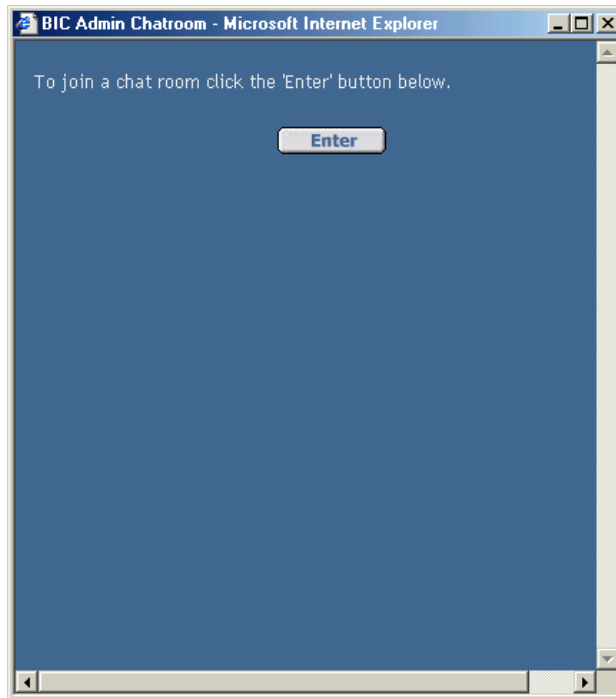


Figure 8. Entering a chat room

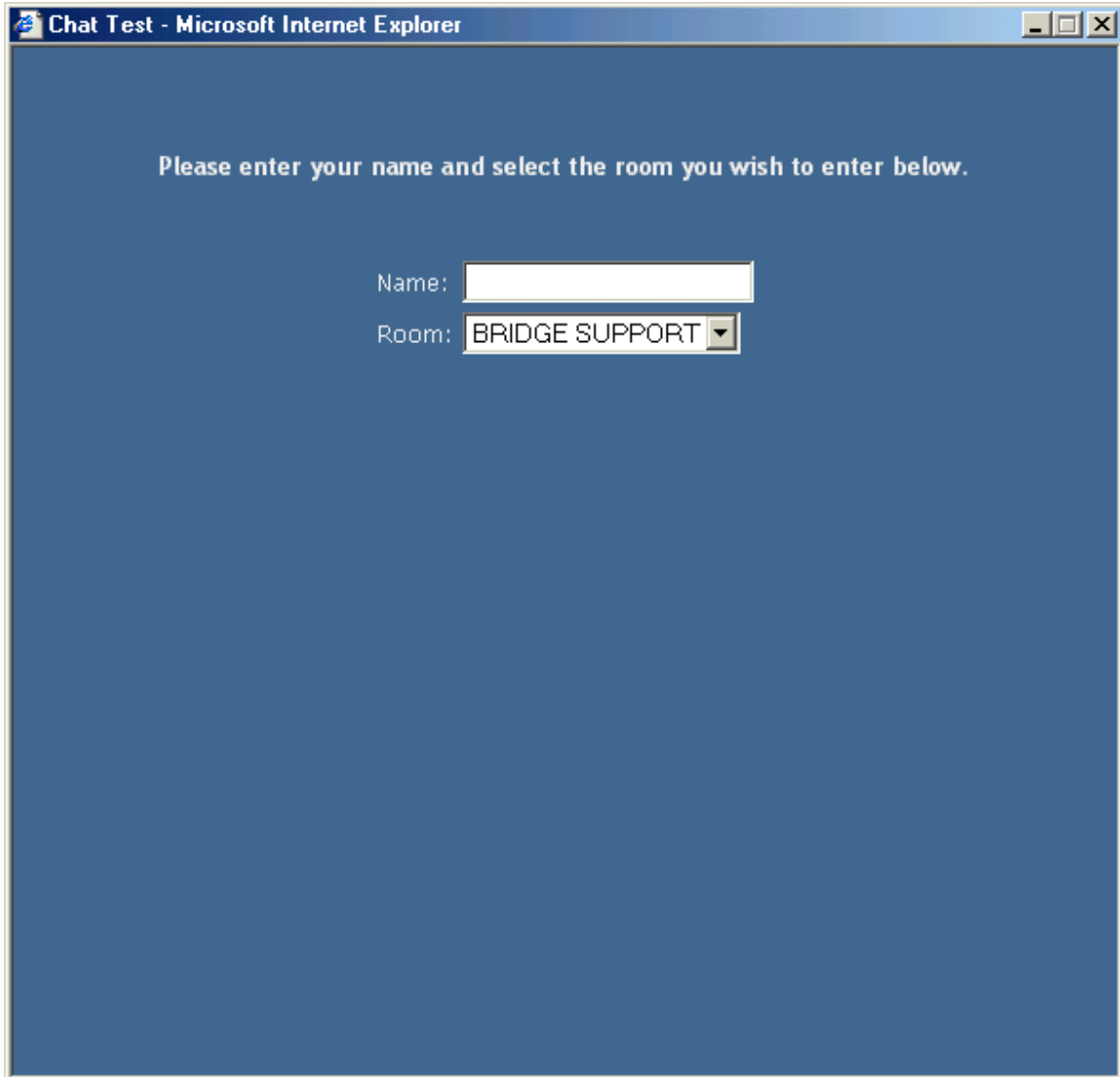


Figure 9. Signing into a chat session

Users can choose any name they would like. A password is not required. By default, users will enter into the Bridge Support chat session. Other sessions may be developed later at the discretion of BIC management. Once the user has logged in, they will be brought to a screen as shown in Figure 10. The left-hand pane shows who is currently signed in and when they have exited. This also displays messages sent. The right-hand pane lists users logged in by user name, IP address, and the time and date that

they signed in. You may also click on an individual to send a private message. A private message will only be displayed on the chosen user's screen. When a new message is typed in, it is sent automatically in ten seconds. This is called the refresh rate. The refresh rate can be changed to 20, 30, or 40 seconds. When you are ready to leave the chat session, click on Logout. You will be brought back to the home page.

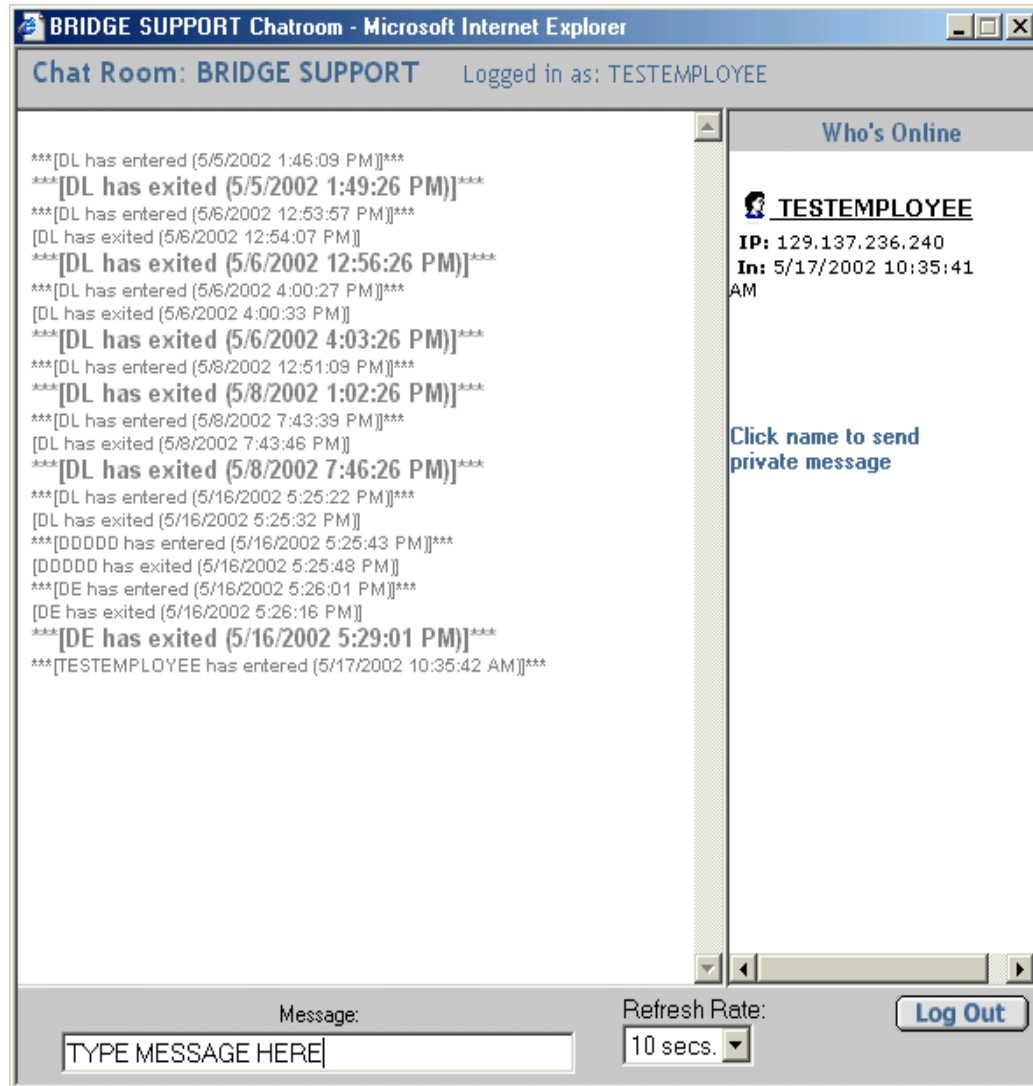


Figure 10. Chat room

You are now ready to login to the inventory site. Once you have logged in successfully, you will be brought to the administrator screen as seen in figure 11. This screen has fourteen links, which are explained in detail on the following pages.

Administrators will have access to the entire site, including editing rights to the items below. Administrators will also be able to search on existing items in each link.

Utility or Function	Description
1. Display Department and Team	View/Update Department and Team Members
2. Display Servers Information	View/Update Servers Information
3. Display Employee List	View/Update Employee List
4. Display Software List	View/Update Software List
5. History Log	View Login history
6. Display Palms Information	View/Update Palms Information
7. Display IP and Sites List	View/Update IP and Sites List
8. Display Printer List	View/Edit Printers Information
9. Edit Reviews Server Space	View/Edit Weekly Server Space Report
10. Edit Personal Info	View/Edit Personal Information

Figure 11. Administrator utility and function

- Display Department and Team.**
This link allows the administrator to view department identification numbers and department names within the company.
- Display Machine Information**
This link allows the administrator to view machine identification, employee identification associated with machine, machine IP address, machine type, operating system installed on machine, CPU speed of machine and RAM installed on machine.
- Display Server Information**
This link allows the administrator to view the server identification, server name, and server IP address, network operating system installed on server, CPU speed of server, and RAM installed on server.
- Display Employee List**
This link allows the administrator to view employee ID, first name, last name, phone extension, cellular phone number, department ID, and login name/password.

- **Display Software List**
This link allows the administrator to view the software ID, software number, software name, software version, software type, software serial number, and date software was last used.
- **Display Current Software Installation**
This link allows the administrator to view installation ID, software number and machine ID in which software is installed on.
- **Display History Log**
This link allows the administrator to view any successful logon to the site. The fields available to view are user name, logon time, logon date, and if user is currently logged in.
- **Display Palms Information**
This link allows the administrator to view Palm Pilot information. The fields include Palm ID, Palm serial number, and employee ID Palm is assigned to.
- **Display IP and Sites List**
This link allows the administrator to view IP and sites within company. The fields include Web site ID, Web site name, IP address of site, the server ID Web site resides on, driver ID, home directory ID, and status of server.
- **Display Printer List**
This link allows the administrator to view all company printers. Fields include printer ID, printer name, printer location, printer path, printer IP, and server ID printer resides on.
- **Edit and Review Server Space**
This link allows the administrator to review server space on all company servers. The fields include Drive ID, server name, drive letter, capacity, unit, free space, and free unit.
- **Edit and Review Admin Information**
This link allows the administrator to view all site administrator information. The fields include admin ID, login name, password, password hint, first name, last name, and cell phone number.
- **Edit Personal Information**
This link allows the administrator to view and update their password, password hint, first name, last name, cell number, and e-mail address. After editing or changing information click the Update record button for changes to take effect. Refer to Figure 12 on updating these fields.

Please update values and press the Update button.

Password	<input type="text"/>
PasswordHint	<input type="text"/>
FirstName	<input type="text"/>
LastName	<input type="text"/>
Cell	<input type="text"/>
E-Mail	<input type="text"/>

Figure 12. Editing and updating personal information

- Edit and Review Employee Information**
 This link allows the administrator to view all employees registered with the site.
- Searching Records**
 Choose the sorting option you wish. You may sort by specific column and/or ascending or descending order. You can also select to view the record by only one column, and display only certain columns to view. For example, when the Administrator clicks on *Display Department and Team*, they will be able to view the Department ID, Department Name, edit the record, delete the record, and add a new record. To view chosen information, click on Display. The results of your search will be displayed, as seen in Figure 13.

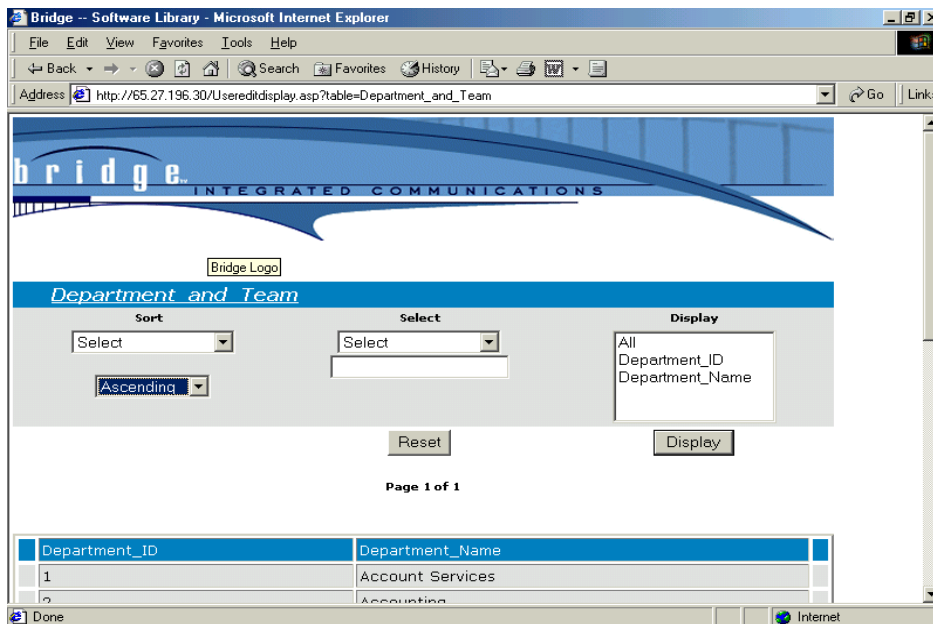


Figure 13. Search results

- **Editing Records**

Editing a record allows the administrator the option of changing existing information. Click on Edit, which can be found to the left of each record. Figure 14 illustrates changing the value of a record and updating those changes.

Please update values and press the Update button.

Department_ID	<input type="text" value="1"/>
Department_Name	<input type="text" value="Account Services"/>
<input type="button" value="Update record"/>	

Figure 14. Changing and updating records

- **Adding Records**

Adding records will be at the discretion of the administrator. The administrator will be able to add records to the site. Figure 15 shows how to add a record. It is also important to note that depending upon the type of record you are adding, you must fill out all fields.

Department_Name	<input type="text"/>
<input type="button" value="Add a"/>	

Figure 15. Adding a record

- **Deleting Records**

To delete a record, you must check the box of the specific record to be deleted. Once the box has been checked, the administrator must click on Delete Selected Record. The administrator also has the option of selecting all records, unselecting all records, deleting all records, and deleting selected records. Figure 16 illustrates these items.

	PalmID	PalmSN	EmployeeID	Delete
Edit	1	5555555-55555-55555	10	<input type="checkbox"/>

Unselect All

Delete All Records

Select all

Delete Selected Records

Figure 16. Deleting records

Users of this site will also login with a valid user ID and password. Once you have logged in successfully, you will be brought to the user screen as seen in figure 17. This screen has four options, which are explained in detail on the following pages.

Welcome to Bridge	
Utility or Function	Description
1. Edit Personal Info	View/Edit Personal Information
2. Display Employee List	View Employee List
3. Display Department	View Department Information
4. Display Software List	View Software List

Figure 17. User utility or function

- Edit Personal Information**
 This link allows the user to view and update their password, password hint, first name, last name, cell number, and e-mail address. Refer to figure 18 on updating these fields.

Please update values and press the Update button.

Password	
PasswordHint	
FirstName	
LastName	
Cell	
E-Mail	

Figure 18. Updating personal information

- Display Employee List**

This link allows the administrator to view employee ID, first name, last name, phone extension, cellular phone number and department ID.

- **Display Department**

This link allows the user to view department identification numbers and department names within the company.

- **Display Software List**

This link allows the user to view the software ID, software number, software name, software version, software type, software serial number, and date software was last used.

6. Conclusions and Recommendations

We chose to do an inventory tracking system based on the need we saw at BIC. The project helped us develop necessary, marketable skills in the field of Information Engineering Technology. We also developed team relation skills that are necessary in any career.

Even though this project is complete and fully functional, it may require much administration on BIC's part. For example, the addition of users by the administrator may become time-consuming. It is feasible, however, to create a "new user?" option on the site that will allow users to sign up immediately upon visiting the site. The current BIC administration recommended that the administrators create new user profiles. We developed this site based on scalability. If BIC chooses to export the databases into SQL server, it will be quite easy to do so.

Overall, this project was challenging and made us both realize the importance of teamwork and communication. The technical aspect of creating this project also challenged us. Based on the completion of this project, Janet Crawford has become interested in pursuing a career in project management. Dam Luong, on the other hand, is interested in pursuing a career in programming. The knowledge that we gained throughout this project helped to determine our future career goals and well as improve our overall skills in Information Engineering Technology.

Appendix A. ASP Code

A.1 ASP Database Setup (invent\$config.asp)

```
<%
' Database Setup
const xDatabase="softwareInvent" 'Database name
const xdblocation="databas"      'location of database
const xdatabasetype=""          ' database type; Drive, Default Access, can be ODBC,
SQLServer
const xssl=""                   ' secure site location

' These const will be used in the whole application
const xbuttonImage=""          ' if not null used as default button
const xbuttonText="View"      ' text for view buttons
const xLCID=""                 ' use Local System identifier. Use system default
const xfont="<font face=""verdana,arial,Helvetica,sans-serif"" size=3 color='#BB0000'>"
const xLogonrequired="No"     ' User must login
const xOtherDB=""             ' Use separate Other Database
const xSearchDB=""           ' search term recording database
const xPromptForErrorLogin="Yes" 'Displaying Error to login
const xSQLUser=""             ' SQL Server user
const xSQLPwd=""              ' SQL Server password
const xSQLServer=""           ' SQL Server name or IP address
%>
```

A.2 ASP for common routines (invent\$db.asp)

```
<%Response.buffer=TRUE%>
<!-- #include file="invent$config.asp" -->
<!-- #include file="invent$language.asp" -->
<!-- #include file="invent$colors.asp" -->
<!-- #include file="inventvbs.asp" -->
<!-- #include file="inventsess.asp" -->
<%
const xAccessOLE="No"          ' Yes uses Jet Ole 4.0 for Access
*****
' Almost all common routines are here and many of the database access
' routines. This routine is included in all other Inv. routines
' Senior Design Project Dam Luong & Janet Crawford
*****
```

```

***** Common variables
dim database
Dim dbname      'unqualified database name
Dim SError
Dim con
Dim rsorder
'for record paging
Dim mypagesize
Dim maxpages

' Users Data
Dim strUserId
Dim strFirstname
Dim strLastname
Dim strAddress
Dim strCity
Dim strState
Dim strPostcode
Dim strPhone
Dim strWorkphone
Dim strMobilephone
Dim strFax
Dim strEmail

' Database Access
Dim SQL
Dim objRS
Dim rsprod
dim mypage
dim maxrecs
Dim DESCRIPTION
Dim databasecnt

%>
<%

***** Open database
Sub InventOpenDatabase (connection)
'Sess ("db")= needs to be set to access file name or ODBC connection
dim databasetype
databasetype=ucase(xdatabasetype)
ShopInit
if xLCID<>"" then
    Session.LCID=xLCID      ' set user supplied LCID
end if

```

```

CheckValidLogin
If databasetype="" or databasetype="DRIVE" then
    ProcessAccessOpen connection
    exit sub
end if
if databasetype="ODBC" then
    database= GetSess("db")
    ProcessODBC connection
    exit sub
end if
if databasetype="SQLSERVER" then
    ProcessSQLServer connection
    exit sub
end if
end sub

*****
' Open Access Database
Sub ProcessAccessOpen(connection)
dim dblocation
dim strconn
dim database
database=GetSess("db") & ".mdb"      ' database name
dblocation=GetSess("dblocation")' location
If dblocation<>"" then
    database = GetSess("dblocation") & "\" & database
end if
if ucase(xdatabasetype)="DRIVE" Then
    If xAccessOle<>"Yes" then
        strconn = "DRIVER=Microsoft Access Driver (*.mdb);DBQ=" & database
    else
        strconn = "provider=microsoft.jet.oledb.4.0;persist security info=false;data source=" &
database
    end if

else
    If xAccessole<>"Yes" then
        strconn = "DRIVER=Microsoft Access Driver (*.mdb);DBQ=" &
Server.MapPath(database)
    else
        strconn = "provider=microsoft.jet.oledb.4.0;persist security info=false;data source=" &
Server.MapPath(database)
    end if
end if

if xdebug="Yes" then

```

```

    debugwrite strconn
end if
Set connection = Server.CreateObject("ADODB.Connection")
on error resume next
connection.open strConn
SetSess "dbc", connection
If connection.errors.count > 0 then
    SetSess "Openerror", "Open Messages<br>" & connection.errors(0).description & "
<br>" & GetSess("dbc")
else
    SetSess "Openerror", ""
end if
End Sub
*****
*****
Sub ProcessODBC (connection)
on error resume next
dim strconn
Set connection = Server.CreateObject("ADODB.Connection")
strconn=GetSess("db")          ' xdatabase = ODBC connection
connection.open strConn
SetSess "dbc", connection
If connection.errors.count > 0 then
    SetSess "Openerror", "Open Messages<br>" & connection.errors(0).description & "
<br>" & GetSess("dbc")
else
    SetSess "Openerror", ""
end if
end sub
*****
' Open SQL Server
Sub ProcessSqlServer(connection)
Set connection = Server.CreateObject("ADODB.Connection")
Dim varServerIP, varUserName, varPassword, varDataBaseName
varServerIP = xSQLServer
varUserName = xSQLUser
varPassword = xSQLPwd
varDataBaseName = GetSess("db")
Connection.Open "DRIVER={SQL Server}; Server=" & varServerIP & "; Database=" &
varDataBaseName & "; UID=" & varUserName & "; PWD=" & varPassword'
If connection.errors.count > 0 then
    SetSess "Openerror", "Open Messages<br>" & connection.errors(0).description & "
<br>" & GetSess("dbc")
else
    SetSess "Openerror", ""
end if

```

```

end sub
'*****
Sub inventCloseDatabase (connection)
on error resume next
connection.close
set connection=nothing
End sub
'

Sub Header_PAGE
%>
<!-- #include file="header.htm" -->
<%
end sub
Sub PageTrailer
%>
<!-- #include file="trailer.htm" -->
<%
end sub
Sub AdminPageTrailer
%>
<!-- #include file="trailer.htm" -->
<%
end sub

Sub UsersPageTrailer
%>
<!-- #include file="userstrailer.htm" -->
<%
end sub

Sub AdminPageHeader
%>
<!-- #include file="header.htm" -->
<%
end sub

'Gets options from database
' ***** Get Contact Session Data
Sub GetCustomerSessionData()
If GetSess("Lastname")<> "" then
    strUserID = GetSess("UserID")
    strFirstname = GetSess("Firstname")
    strLastname = GetSess("Lastname")
    strAddress = GetSess("Address")
    strCity = GetSess("City")
    strState = GetSess("State")

```

```

        strPostCode = GetSess("PostCode")
        strCompany = GetSess("Company")
        strPhone = GetSess("Phone")
        strWorkphone = GetSess("Workphone")
        strMobilephone = GetSess("Mobilephone")
        strFax = GetSess("Fax")
        strEmail = GetSess("Email")
    exit sub
else
    If xSaveUsersDetails="Yes" then
        RestoreUsersDetailsCookie
    end if
end if
End sub
'

Sub CheckError (dbc, rc)
rc=0
If err.number > 0 Then
rc=4
'Response.write "msg=" & msg & "<br>"
Response.write("VBScript Errors Occurred:" & "<P>")
Response.write("Error Number=" & err.number & "<P>")
Response.write("Error Descr.=" & err.description & "<P>")
Response.write("Help Context=" & err.helpcontext & "<P>")
Response.write("Help Path=" & err.helppath & "<P>")
Response.write("Native Error=" & err.nativeerror & "<P>")
Response.write("Source=" & err.source & "<P>")
Response.write("SQLState=" & err.sqlstate & "<P>")
End If
If dbc.errors.count> 1 then
    rc=4
    Response.write "count=" & dbc.errors.count
    For counter= 0 to dbc.errors.count
        Response.write("Error #" & dbc.errors(counter).number & "<P>")
        Response.write("Error desc. -> " & dbc.errors(counter).description & "<P>")
    next
End If
end sub
'*****
'*****
' Logic to determine if we should reinitialize everything
Sub UserInit()
if GetSess("INIT") <> "INIT" or GetSess("db")<>xDatabase then
'    debugwrite "setting up session db= " & GetSess("db")
    SetupSession
end if

```

```
end sub
```

```
Sub SetupSession()
```

```
' sets database for this session use "mydatabase" not "mydatabase.mdb"
```

```
' location is relative path to database such as ..\..\data
```

```
SetSess "db", ""
```

```
SetSess "dblocation", xdblocation
```

```
SetSess "ssl", xssl
```

```
SetSess "home", xhome
```

```
SetSess "$email", xemail
```

```
SetSess "$emailname", xemailname
```

```
SetSess "$emailsubject", xemailsubject
```

```
SetSess "$emailsystem", xemailsystem
```

```
SetSess "InventAdmin", ""
```

```
SetSess "inventAdminDB", ""
```

```
SetSess "InventUser", ""
```

```
SetSess "inventUserDB", ""
```

```
SetSess "GiftID", ""
```

```
If GetSess("Login") <> "Force" then
```

```
    SetSess "Login", ""
```

```
end if
```

```
' locate database on string
```

```
'GetDb
```

```
database= GetSess("db")
```

```
if database="" then
```

```
    SetSess "db", xDatabase ' default value
```

```
end if
```

```
'response.write "Session initialized database = " & GetSess("db") & "<br>"
```

```
SetSessionTimeout
```

```
SetSess "INIT", "INIT"
```

```
end sub
```

```
'
```

```
Sub GetDB()
```

```
'used to set database name from form or query string
```

```
' sets database for this session from query string or form
```

```
dim database
```

```
database=Request("db")
```

```
if database <> "" then
```

```
    SetSess "db", database
```

```
    exit sub
```

```
end if
```

```
'response.redirect "Invent.asp?msg=" & Server.URLEncode("Unable to locate database name")
```

```
end sub
```

```
sub PageNavBar(sql)
```

```

dim scriptname,counterstart,pad,counterend,counter,ref,mysql
mysql=sql
SetSess "sqlquery",sql
    ,
    pad=""
    scriptname=request.servervariables("script_name")
    response.write PageNavTable & PageNavRow
    response.write PageNavColumn & PageNavFont
    if (mypage mod 10) = 0 then
        counterstart = mypage - 9
    else
        counterstart = mypage - (mypage mod 10) + 1
    end if
    counterend = counterstart + 9
    response.write "maxpages=" & maxpages & " pagesize=" & mypagesize &
"myapge=" & mypage
    response.write scriptname & " " & mysql
    if counterend > maxpages then counterend = maxpages
    if counterstart <> 1 then
        ref="<a href=" & scriptname
        ref=ref & "?page=" & 1
    response.write ref
    ref=ref & "&pagesize=" & mypagesize
    ref=ref & "&sqlQuery=" & server.URLEncode(mySQL)
    ref=ref & ">First</a>&nbsp;:&nbsp;"
    Response.Write ref
        ref="<a href=" & scriptname
        ref=ref & "?page=" & (counterstart - 1)
    response.write ref
    ref=ref & "&pagesize=" & mypagesize
    ref=ref & "&sqlQuery=" & server.URLEncode(mySQL)
    ref=ref & ">Previous</a>&nbsp;"
    Response.Write ref
    end if
    Response.Write "["
    for counter=counterstart to counterend
        If counter>=10 then
            pad=""
        end if
        if cstr(counter) <> mypage then
            ref="<a href=" & scriptname
            ref=ref & "?page=" & counter
        response.write ref
        ref=ref & "&pagesize=" & mypagesize
        ref=ref & "&sqlQuery=" & server.URLEncode(mySQL)
        ref=ref & ">" & pad & counter & "</a>"
        else
            ref="<b>" & pad & counter & "</b>"
        end if
    next counter
end for
end sub

```

```

    response.write ref
    if counter <> counterend then response.write " "
next
Response.Write "]"
if counterend <> maxpages then
    ref="&nbsp;<a href=" & scriptname
    ref=ref & "?page=" & (counterend + 1)
'   ref=ref & "&pagesize=" & mypagesize
'   ref=ref & "&sqlQuery=" & server.URLEncode(mySQL)
    ref=ref & ">Next</a>"
    Response.Write ref

    ref="&nbsp;:&nbsp;<a href=" & scriptname
    ref=ref & "?page=" & maxpages
'   ref=ref & "&pagesize=" & mypagesize
'   ref=ref & "&sqlQuery=" & server.URLEncode(mySQL)
    ref=ref & ">Last</a>"
    Response.Write ref
end if
response.write "<br>" & PageNavFontEnd
response.write PageNavTableEnd

end sub

Sub GenerateSelectV (iFieldnames,ifieldvalues,currentvalue,selectname,count, firstfield)
dim i
' Generates Select with values
%>
<select size=1 name="<%=selectname%>">
<%
If firstfield<>"" then
    response.write "<option>" & firstfield & "</option>"
end if
for i = 0 to count-1
    If currentvalue= ifieldvalues(i) then
        response.write "<option value=" & ifieldvalues(i) & " selected>" & iFieldnames(i) &
"</option>"
    else
        response.write "<option value=" & ifieldvalues(i) & ">" & iFieldnames(i) &
"</option>"
    end if
next
%>
</select>
<%
end sub

```

Sub GenerateSelectNV (iFieldnames,currentvalue,selectname, count,firstfield)

' Generates select with no values

dim i

%>

<select size=1 name="<%=selectname%>">

<%

If firstfield<>"" then

 response.write "<option>" & firstfield & "</option>"

end if

for i = 0 to count-1

 If currentvalue= ifieldnames(i) then

 response.write "<option Selected>" & ifieldnames(i) & "</option>"

 else

 response.write "<option>" & ifieldnames(i) & "</option>"

 end if

next

%>

</select>

<%

end sub

Sub GenerateSelectMULT (iFieldnames,fieldcount,currentvalues,currentvaluecount,selectname,firstfield)

' Generates select with no values

%>

<select MULTIPLE size=5 name="<%=selectname%>">

<%

Dim Currentvalue

Dim j

Dim found

Currentvalue=""

If firstfield<>"" then

 response.write "<option>" & firstfield & "</option>"

end if

,

for i = 0 to fieldcount-1

 If currentvaluecount> 0 then

 Found="FALSE"

 j=0

 Do WHILE found="FALSE" and j < currentvaluecount

 if iFieldnames(i)= Currentvalues(j) then

 currentvalue=ifieldnames(i)

 found="TRUE"

 end if

```

        j=j+1
    Loop
end if
If currentvalue= ifieldnames(i) then
    response.write "<option Selected>" & ifieldnames(i) & "</option>"
else
    response.write "<option>" & ifieldnames(i) & "</option>"
end if
next
%>
</select>
<%
end sub
Sub DebugWrite (msg)
response.write msg & "<br>"
end sub
'
Sub DiagnosticOpen (connection,database,databasetype)
SetSess "db",database
databasetype=ucase(databasetype)
If databasetype="" then
    ProcessAccessOpen connection
    exit sub
end if
if databasetype="ODBC" then
    database= GetSess("db")
    ProcessODBC connection
    exit sub
end if
if databasetype="SQLSERVER" then
    ProcessSQLServer connection
    exit sub
end if
ProcessAccessOpen connection
end sub

' all admin now must use standard open
Sub OpenDB (ByRef con, d)
InventOpenDatabase con
End Sub
%>
<%
'-----
Function GetAccess(user, con)
    SQL = "SELECT * FROM Admin WHERE LoginName = " & user & ""
    Set objRec = con.Execute(SQL)

```

```

        GetAccess = objRec("AccessNumber")
        objRec.Close
    End Function

Function GetAccess2(user, con) 'Bridge members login
    SQL = "SELECT * FROM Employees WHERE LoginName = '" & user & "'"
    Set objRec = con.Execute(SQL)
    GetAccess2 = objRec("AccessNumber")
    objRec.Close
End Function

'-----
Sub inventCheckAdmin
If GetSess("InventAdmin")="" Then
    If XshowAdmin<>"Yes" then
        response.redirect "shoperror.asp?msg=" & Server.URLEncode(LangAdminUnauth)
    else
        response.redirect xadminpage
    end if
end if
If Getsess("inventAdminDB")<>xdatabase then
    If XshowAdmin<>"Yes" then
        response.redirect "shoperror.asp?msg=" & Server.URLEncode(LangAdminUnauth)
    else
        response.redirect xadminpage
    end if
end if
End Sub

'-----

Sub inventCheckUser
If GetSess("InventUser")="" Then
    If xShowUsers<>"Yes" then
        response.redirect "shoperror.asp?msg=" & Server.URLEncode(LangAdminUnauth)
    else
        response.redirect xUserPage
    end if
end if
If Getsess("inventUserDB")<>xdatabase then
    If xShowUsers<>"Yes" then
        response.redirect "shoperror.asp?msg=" & Server.URLEncode(LangAdminUnauth)
    else
        response.redirect xUserpage
    end if
end if
end if

```

End Sub

```
'--- login user activity: History log
Sub LogUser(user,io, dbc)          'admin log
on error resume next
    login = "INSERT INTO History_Log (fldUserName,fldTime,fldDate,fldInOut)
VALUES('" & user & _
    "','" & Time & "','" & Date & "','" & io & "')"
    dbc.Execute(login)
    SetSess "inventAdminDB",xdatabase
```

End Sub

```
Sub LogUser2(user,io, dbc)        'Users log
on error resume next
    login = "INSERT INTO History_Log (fldUserName,fldTime,fldDate,fldInOut)
VALUES('" & user & _
    "','" & Time & "','" & Date & "','" & io & "')"
    dbc.Execute(login)
    SetSess "inventUserDB",xdatabase
```

End Sub

```
'--- Build Access control table for user
Sub BuildAccess(id)

Dim objrec3, objrec2, ba, temp
temp = "SELECT * FROM tblAccess ORDER BY fldAuto"
Set objRec3 = con.Execute(temp)
While Not objRec3.EOF
    ba = "SELECT * FROM tblUser WHERE fldAuto = " & id
    Set objRec2 = con.Execute(ba)
    If Not objRec2.EOF then
        If Instr(objRec2("fldAccess"),objRec3("fldAuto"))<>0 Then
            response.write "<input type=Checkbox name=Access value=" &
objRec3("fldAuto") & " checked><font face=verdana size=2 color=navy>" &
objRec3("fldName") & "</font><br>"
        Else
            response.write "<input type=Checkbox name=Access value=" &
objRec3("fldAuto") & "><font face=verdana size=2 color=navy>" &
objRec3("fldName") & "</font><br>"
        End If
    Else

```

```

        response.write "<input type=Checkbox name=Access value=" &
objRec3("fldAuto") & "><font face=verdana size=2 color=navy>" &
objRec3("fldName") & "</font><br>"
    End If
    objrec2.close
    set objrec2=nothing
objRec3.MoveNext
Wend
objrec3.close
set objrec3=nothing
End Sub

' --- convert access number into description
Function ConvertAccess(list)
dim objrec2, ac, temp, name
on error resume next
    ac = Split(list, ",")
    For each item in ac
        temp = "SELECT * FROM tblAccess WHERE fldAuto = " & cINT(item)
        Set objRec2 = con.Execute(temp)
        If Not objRec2.EOF then
            name = name & objRec2("fldName") & ", "
        End If
    Next
    If name <> "" Then
        ConvertAccess = Left(name,Len(name)-2)
    End If
    objrec2.close
    set objrec2=nothing
End Function
Sub GetDataBaseTables (tables, tablecount,con)
'set array tables with names of tables in database
dim table
dim i
dim tblName
redim tables(50)
Set table = con.OpenSchema (20)
i=0
While Not table.EOF
    tblName= table("Table_Name")
    If Left(tblName,4) <> "MSys" AND Left(tblName,3) <> "sys" AND Left(tblName,4)
<> "RTbl" Then
        Tables(i)=tblName
        i=i+1
    end if
    table.MoveNext

```

```

Wend

Dim othertables(50),othercount,j
If xothertables<>"" then
    parserecord xothertables,othertables,othercount,""
    for j=0 to othercount-1
        tables(i)=Othertables(j)
        i=i+1
    next
end if
tablecount=i
end sub
*****

'Open databases other than standard
Sub OpenOtherDatabase (connection, indb, inlocation, intype)
if xLCID<>"" then
    Session.LCID=xLCID      ' set user supplied LCID
end if
dim databasetype
SaveSessionDB
SetSess "db",indb
SetSess "Dblocation",inlocation
databasetype=ucase(intype)
If databasetype="" then
    ProcessAccessOpen connection
    RestoreSessionDB
    exit sub
end if
if databasetype="ODBC" then
    database= GetSess("db")
    ProcessODBC connection
    RestoreSessionDB
    exit sub
end if
if databasetype="SQLSERVER" then
    ProcessSQLServer connection
    RestoreSessionDB
    exit sub
end if
end sub

Sub SaveSessionDb
SetSess "Olddb",GetSess("db")
SetSess "Olddblocation", GetSess("dblocation")
end sub
Sub RestoreSessionDB

```

```

SetSess "db",GetSess("Olddb")
SetSess "dblocation",GetSess("Olddblocation")
end sub
' Use to open other database
,

Sub inventOpenOtherDB (connection, database)

dim mytable
If database="" then
    inventOpendatabase connection
    exit sub
end if
OpenOtherdatabase connection,database,xdblocation,xdatabasetype
end sub

Sub CheckValidLogin
' everyone must login then make sure they have
If xlogonrequired="Yes" then
    If GetSess("Login")="" then
        response.redirect "shoploginforce.asp"
    end if
end if
end sub

Sub LanguageSet (language)
Setsess "Language",language
LanguageDefine Language
end sub
,

Sub FindOtherDatabase (database, table)
' look in othertables and otherdatabases to find database match the table
dim othertables(50),otherdatabase(50), j, dbcount, tablecount
database=""
If xotherdatabases<>"" and xothertables<>"" then
    parserecord xothertables,othertables,tablecount,""
    parserecord xotherdatabases,otherdatabase,dbcount,""
    for j=0 to tablecount-1
        if table=ucase(Othertables(j)) then
            database=otherdatabase(j)
        end if
    next
end if
end sub

Sub OpenRecordSet (conn, irs, isql)
Set irs=conn.execute(isql)

```

```
end sub
```

```
Sub CloseRecordSet (irs)
```

```
irs.close
```

```
set irs=nothing
```

```
end sub
```

```
Sub Adjustdate (newdate)
```

```
end sub
```

```
Sub AdjustTime (newtime)
```

```
end sub
```

```
%>
```

A.3 ASP to format display (invent\$colors.asp)

```
%
```

```
'
```

```
*****
```

```
' Used to modify tables and fonts, search fields,
```

```
' product fields
```

```
*****
```

```
' Forms Users
```

```
const TableDef="<center><table cellpadding=1 cellspacing=1>"
```

```
Const TableDefEnd="</table></center>"
```

```
Const TableRow="<tr bgcolor=#C4CEE5>"
```

```
const TableRowEnd="</tr>"
```

```
const TableColumn="<td><font face='verdana,arial' size=2 color=#0000ff>"
```

```
const TableColumnEnd="</font></td>"
```

```
Const xTableRowFont="<font face='verdana,arial' size=2 color=#000000>"
```

```
Const xTableRowFontEnd="</font>"
```

```
Const xTableRowColor="#C4CEE5"
```

```
const TableDefLogin="<table cellpadding=2 cellspacing=0 width='30%'>"
```

```
' Reports reports
```

```
const ReportTableDef="<table width=701 border=2 align=center cellpadding=2  
cellspacing=2>"
```

```
const ReportHeadRow="<tr bgcolor='#0080C0'>"
```

```
Const ReportRowEnd="</tr>"
```

```
const ReportHeadColumn="<td><font face='Verdana, Arial' size=2 color='#FFFFFF'>"
```

```
const ReportHeadColumnEnd="</font></td>"
```

```
const ReportDetailRow="<tr bgcolor='#E5E5E5'>"
```

```
const ReportDetailColumn="<td valign='top'><font face='Verdana, Arial' size=2>"
```

```
const reportDetailColumnEnd="</font></td>"
```

```

const ReportDetailRowX="<tr bgcolor='#CCCCCC'>"
const ReportTableEnd="</table>"
const ReportInfoFont="<font face='Arial, Verdana' size=3><b>"
const ReportInfoFontEnd="</font></b>"

' Categories
Const xSortCategories="catdescription" ' category sort
const CatMaxColumns=3 ' columns to display
const CatHeader="<b><p align=center><font size='5' color='#990033'>"
'const CatHeader="<b><p align=center><font size='5' color='#CC0000'>"
const CatRow="<TR bgColor='#FFFFFF'>"
Const CatColumn="<TD valign=top><p align=center><b><font face='arial,verdana'
size=5 color='#BB000'>"
Const CatTable="<table border=10 cellpadding=4 width='60%' cellspacing=4
bordercolor='#BB000'>"
Const CatColumnEnd="</font></b></p></td>"
'

' User
const CustOutsideTableDef="<table cellpadding=2 cellspacing=2><tr><td valign=top>"
'

'FormatSoftware Details
Const FOrderTable="<table border='1' cellpadding='2' cellspacing='0' width='80%'>"
Const FOrderRow="<tr>"
Const FOrderRowEnd="</tr>"
Const FOrderHeaderColumn="<TD align='left' colspan=2 bgColor='darkblue'><font
face='Verdana, Arial, Helvetica' size=3 color='#FFFFFF'><STRONG>"
Const FOrderHeaderColumnEnd="</STRONG></font></TD>"
Const FOrderFieldRow="<tr>"
' user details
Const FOrderFieldLeft="<TD align='left' width='30%' bgcolor='#FFFFFF'><font
face='Verdana, Arial'>"
Const FOrderFieldLeftEnd="</font></TD>"
Const FOrderFieldRight="<TD align='left' width='70%'><b><font face='Verdana,
Arial'>"
Const FOrderFieldRightEnd="</font></b></TD></TR>"

' company details
Const FOrderCompany="<font face='Verdana, Arial' size=2>"
const FOrderCompanyEnd="</font>"
Const FOrderComment="<font face='Verdana, Arial'>"
Const ForderCommentend="</font>"
'

const SearchDisplayCategories="Yes"
Const SearchKeywordTable="<table cellpadding=1 cellspacing=1 width='70%'>"
Const SearchHeaderRow="<tr bgcolor='#0080C0'><td colspan=2><font face='verdana'
size='3' color='#FFFFFF'><b>"

```

```

Const SearchKeywordRow="<tr bgcolor=#E5E5E5><td>"
Const SearchCatTable="<table border=0 cellpadding=1 cellspacing=1 width='70%'>"
Const SearchCatHeaderLeft="<tr bgcolor=#E5E5E5><td></td><td valign=top><b>"
Const SearchCatHeaderRight="<td valign=top><b>"
Const SearchCatRowStart="<tr bgcolor=#E5E5E5>"
Const SearchCatRowEnd="</tr>"
const SearchCatColumnStart="<td><font face='verdana,arial' size=3 color='#000000'>"
const SearchCatColumnEnd="</font></td>"
const SearchSubcatColumnStart="<td><font face='verdana,arial' size=3
color='#000000'>"
const SearchSubCatColumnEnd="</font></td>"
'
'InventError
const ErrorFontStart="<p align=center><font face=Verdana size=4
color='#BB0000'><b>"
const ErrorFontEnd="</b></font><br><br>"

Sub SetupProductFields (ProdFields, ProdHeaders)
ProdFields=Array("cname","cdescription","quantity","cprice")
ProdHeaders=Array(LangProductName,LangProductDescription,LangProductQuantity,L
angProductPrice)
end sub
'

Sub SetupSearchFields (SearchFields)
SearchFields=Array("cname","cdescription","ccode", "mfg")
end sub
'

' EditDisplay
const EditSpecialColumn="<td align=center><font face='verdana' size=1
color='#FFFFFF'>"
const AddGeneralTable="<table border=1 cellpadding=1 cellspacing=1
width=""90%"">"
const EditSpecialColumnEnd="</font></td>"
const EditLinkColumn="<td valign=top><font face='verdana' size=1>"
Const EditLinkColumnEnd="</font></td>"
const EditFieldSpecial="<td bgcolor=#E5E5E5><center>"

' Navigation
const PageNavTable="<table align=center>"
const PageNavRow="<tr>"
const PageNavColumn="<td>"
const PageNavFont="<font size='2' color='black' face='Verdana, Arial'>"
const PageNavFontEnd="</font>"
const PageNavTableEnd="</td></tr></table>"
' Admin
const CustAdminTable="<table border='0' cellpadding='1' cellspacing='1' width='60%'>"

```

```

Const CustAdminHeaderRow="<tr bgcolor='#0080C0'><td colspan=2'>"
const CustAdminHeaderFont="<p align='center'><b><font face='Verdana'
color='#FFFFFF'>"
const CUSTAdminHeaderEnd="</font></b></td></tr>"
const CustadminRow="<tr bgcolor='#E5E5E5'>"
Const CustAdminNumColumn="<td align='right' width='5%'><font size=1
face=verdana>"
const CustAdminNumEnd="</font></td>"
const CustAdminColumn="<td width='80%'><p><b>"
const CustAdminFont="<font size=2 face=verdana>"
const CustAdminColumnEnd="</font></b></td></tr>"
'

Sub CreateCustRow (caption, fieldname, fieldvalue, required)
Dim aster
If required="Yes" then
    aster="* "
else
    aster=" "
end if
Response.write tablerow & tablecolumn
Response.write aster & Caption & TablecolumnEnd
Response.write tablecolumn
%>
<input size=25 name="<%=fieldname%>" value="<%=fieldvalue%>">
<%
Response.write tablecolumnend & tableRowend
end sub
'

Sub FormatEditRow (caption,fieldname,fieldvalue)
dim capdisplay
capdisplay=caption
if capdisplay="" then
    capdisplay=fieldname
end if
Response.Write TableRow
Response.write TableColumn & capdisplay & TableColumnEnd
Response.write TableColumn & "<input size=40 name=" & fieldname & " value=" &
Chr(34) & fieldvalue & Chr(34) & ">"
Response.write TableColumnEnd
Response.write TableRowEnd
end sub

Sub FormatEditRow2 (caption,fieldname,fieldvalue)
dim capdisplay
capdisplay=caption
if capdisplay="" then

```

```
        capdisplay=fieldname
    end if
    Response.Write TableRow
    Response.write TableColumn & capdisplay & TableColumnEnd
    Response.write TableColumn & "<input type='password' size=40 name=" & fieldname &
    " value=" & Chr(34) & fieldvalue & Chr(34) & ">"
    Response.write tableColumnEnd
    Response.write TableRowEnd
end sub
```

```
Sub FormatEditRow3 (caption,fieldname,fieldvalue)
    dim capdisplay
    capdisplay=caption
    if capdisplay="" then
        capdisplay=fieldname
    end if
    Response.Write TableRow
    Response.write TableColumn & "" & TableColumnEnd
    Response.write TableColumn & "<input type='hidden' size=38 name=" & fieldname & "
    value=" & Chr(34) & fieldvalue & Chr(34) & ">"
    Response.write tableColumnEnd
    Response.write TableRowEnd
end sub
```

```
%>
```

Appendix B. HTML Code

B.1 Bridge Software Library Web Front-End Code

B.1.1 Admin.asp

```
<!--#include file="aspInclude/invent$db.asp"-->
<%
*****
' Senior design project, Fall 2001
' Dam Luong & Janet Crawford
' Force Admin & User logon -----> going to AdminLogged.asp
' If not logon, -----> going to Admin.asp to force logon with an error message
'
*****
SetSess "InventAdmin", ""
SetSess "INIT", ""
Dim myconn
Dim rs
Dim username, userpassword
msg=""
If xlogonrequired="Yes" then
    SetSess "Login", "Force"
end if
InventOpenDataBase myconn
If GetSess("Login")="Force" then
    SetSess "Login", ""
end if
'on error resume next
If Request("Submit")<>"" Then
    username=request("Username")
    userpassword=request("password")
    username=replace(username, "", "")
    userpassword=replace(userpassword, "", "")
    SQL = "SELECT * FROM Admin WHERE LoginName=" & username & " AND
Password=" & userpassword & ""
        Set rs = myconn.Execute(SQL)

    if err.number>0 then
        msg="database Open error<br>" & GetSess("Openererror")
    else
        If Not rs.EOF Then
            SetSess "InventAdmin" ,rs("LoginName")

            if isnull(rs("Admintype")) then
                SetSess "admintype", "SUPER"
```



```

</TR>
<TD WIDTH=30 ALIGN=LEFT VALIGN=Middle><font face="Arial, Helvetica"
SIZE=2><B><%=LangAdminPassword%></B></FONT></TD>
<TD ALIGN=LEFT VALIGN=TOP><font face="Arial, Helvetica"><INPUT
TYPE=PASSWORD NAME="Password"></font></td></TR>
<tr>
<td COLSPAN=2 ALIGN=CENTER><font face="Arial, Helvetica"><INPUT
TYPE=SUBMIT VALUE="<%=LangAdminLogin%>" name="Submit"></font></TD>
</tr> </form>
</TABLE>

```

```

</TD>
</TR>
<TR>
<td width=50><br></td>
<TD>
<TABLE WIDTH=50 BORDER=1 CELLPADDING=3 CELLSPACING=0
align="right" bordercolordark="#333399" bordercolorlight="#666699"><form
action="<%=xUserPage%>" method="post" name="LoginForm2">
<TR>
<TD BGCOLOR="#0080C0" COLSPAN=2 ALIGN=LEFT VALIGN=TOP>
<font face="Arial, Helvetica" SIZE=2
color=white><B><%=LangAdmin04%></B></FONT></TD>
</TR>
<TR>
<TD WIDTH=30 ALIGN=LEFT VALIGN=Middle><font face="Arial, Helvetica"
SIZE=2><B><%=LangAdminUserName%></B></FONT></TD>
<TD ALIGN=LEFT VALIGN=TOP>
<font face="Arial, Helvetica"><INPUT TYPE=TEXT NAME="UserName"
VALUE="<%=Request("UserName") %>"></font></TD>
</TR>
<TR>
<TD WIDTH=30 ALIGN=LEFT VALIGN=Middle><font face="Arial, Helvetica"
SIZE=2><B><%=LangAdminPassword%></B></FONT></TD>
<TD ALIGN=LEFT VALIGN=TOP><font face="Arial, Helvetica"><INPUT
TYPE=PASSWORD NAME="Password"></font></td></TR>
<tr>
<td COLSPAN=2 ALIGN=CENTER><font face="Arial, Helvetica"><INPUT
TYPE=SUBMIT VALUE="<%=LangAdminLogin%>" name="Submit"></font></TD>
</tr></form>
</TABLE>

```

```

</TD>
</TR>

```



```

InventOpenDatabase myconn
FormatHeader
list = GetAccess(GetSess("inventAdmin"), myconn)
ac = Split(list, ",")
For each item in ac
    SQL = "SELECT * FROM tblAccess WHERE fldAuto = " & cINT(item)
        Set objRec = myconn.Execute(SQL)

        FormatRow objRec
    Next
inventCloseDatabase myconn
FormatTrailer
end sub
Sub FormatHeader
%>
<table border="0" cellpadding="1" cellspacing="1" width="701" bordercolor="#008080"
align=center>
    <tr bgcolor="#0080C0">
        <td colspan=2 width="45%">
            <p align="center"><b><font face="Verdana"
color="#FFFFFF"><%=LangAdminFunction%></font></b></td>
            <td width="60%">
                <p align="center"><b><font face="Verdana"
color="#FFFFFF"><%=LangAdminDescription%></font></b></td>
        </tr>
    <%
end sub
Sub FormatRow (objRec)
dim name,url,comment
name = objRec("fldName")
url = objRec("fldURL")
Comment=objREC("fldComment")
NumProc=Numproc+1
%>

<tr bgcolor="#E5E5E5">
    <td align="right" width="5%"><font size=1 face=verdana><%=NumProc%>.</td>
    <td width="40%"><p><b><a href="<%=url%>"><font size=2
face=verdana><NOBR><%=name%></NOBR></font></a></b></td>
    <td width="55%"><font size=2 face=verdana><%=comment%></font></td></tr>
<%
End Sub

Sub FormatTrailer
%>
</table>

```

```
<%  
end sub
```

```
%>
```

B.1.3 Header.htm (Header of every page)

```
<HTML>  
<script language=JavaScript type="Text/JavaScript">  
<!-- Hide script from old browsers  
  
// function to open new window  
function openWin(url,windowname) {  
    popwin =  
    open(url,windowname,'toolbar=no,location=no,status=no,menubar=no,scrollbars=yes,res  
izable=yes,width=422,height=455');  
    popwin.focus();  
}  
  
// Stop hiding script from old browsers -->  
</script>  
<HEAD>  
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;CHARSET=iso-8859-  
1">  
<META NAME="keywords" CONTENT="Bridge, marketing, marketing  
communications, communications, adverstising, education, Worldwide Partners,  
American Association, Advertising Agencies, communications experts, Endo-Surgery  
Institute, brand identity, professional education, brochures, promotional video, video,  
interactive, CD-ROM, healthcare, diffusion marketing, commercials, TV commercials,  
television commercials, print campaign">  
<META NAME="description" CONTENT="Bridge. Strategies that help businesses  
thrive in an e-commerce world.">  
  
<TITLE>Bridge -- Software Library</TITLE>  
  
<Body marginheight="0" marginwidth="0" bgcolor="#ffffff" topmargin="0"  
leftmargin="0">  
<table width=701 align=left>  
    <tr><td>  
        <table width="701" cellpadding="0" cellspacing="0" border="0" align=center>  
  
            <tr>  
  
                <td width="264"><br><br></td>
```



```

if dbtable="" then
    AdminPageHeader
    Response.write LangEditSelectFail & "<br>"
    AdMinPageTrailer

else
    EditOpenDatabase conn, database,dbtable
    If sAction = "" Then
        AdminPageHeader
        GenerateForm
        AdminPageTrailer
    Else
        AdminPageHeader
        UpdateRecord
        GenerateForm
        AdminPageTrailer
    end if
    inventclosedatabase conn
end if
*****

Sub GetInputValues
' ID, allows editing a record
which=request.querystring("which")
idfield=request.querystring("idfield")
dbtable=request.querystring("table")
database=request.querystring("database")
ValidateTable
End Sub
'

Sub ValidateTable
*****
'See if user has access to this table
Dim UserTables, i
dim tablecount
if XRestrictAdminTables<>"Yes" then exit sub
UserTables=GetSess("UserTables")
If Isnull(UserTables) then
    exit sub
end if
if UserTables="" then
    exit Sub
else
    UserTables=split(GetSess("UserTables"),",",-1,1)
end if
tablecount=ubound(UserTables)
for i = 0 to tablecount

```

```

    if ucase(dbtable)=ucase(Ustables(i)) then
        exit sub
    end if
next
dbtable=""
end sub

```

```

Sub GenerateForm
dim sqltemp
sqltemp="select * from " & dbtable
sqltemp=sqltemp & " where " & idfield & "=" & which
'Debugwrite sqltemp
set rstemp=conn.execute(sqltemp)
DisplayForm
rstemp.close
set rstemp=nothing
end Sub

```

```

Sub DisplayForm()
dim keyvalue, howmanyfields, i, fieldname, fieldvalue
keyvalue=rstemp(idfield).value
howmanyfields=rstemp.fields.count -1
%>
<form name="editform"
action="editrecord.asp?database=<%=database%>&table=<%=dbtable%>&idfield=<%=
idfield%>&which=<%=which%>" method="POST">
<%
                Response.Write(xfont & LangEdit02 & "</font><br>")
                Response.Write(xfont & sError & "</font><br>")
                Response.Write TableDef
    for i=0 to howmanyfields
        fieldname = rstemp(i).name
        fieldvalue = rstemp(i).value
        FormatEditRow "",fieldname,fieldvalue
    next
    Response.Write(TableDefEnd)
    Response.Write("<input type=submit name=action value=\"" & LangEditUpdateRecord
& "\"">")
    Response.Write("</form>")
    response.write "<hr>"
end sub

```

,

```

Sub UpdateRecord

```

```

dim sqltemp
sqltemp="select * from " & dbtable
sqltemp= sqltemp & " where " & idfield & "=" & which
Set rstemp = Server.CreateObject("ADODB.Recordset")
rstemp.open sqltemp, conn, 1, 3
'rstemp.open sqltemp, conn, adOpenKeyset, adLockOptimistic
GenerateUpdateSQL
rstemp.close
set rstemp=nothing
sError= sError & "<br><b>" & LangEdit03 & "</b>"
end sub
' ***** general Sql
Sub GenerateUpdateSQL()
Dim howmanyfields
dim fieldname, fieldvalue, fieldtype
dim i
howmanyfields=rstemp.fields.count -1
rstemp.update
for i=1 to howmanyfields
    fieldname = rstemp(i).name
    fieldtype=rstemp(i).type
    fieldvalue = request.form(fieldname)
    EUpdatefield fieldname,fieldvalue
next
rstemp.update
end sub
Sub EUpdateField (fieldname, fieldvalue)
on error resume next
'Debugwrite fieldname & "value=" & fieldvalue
if fieldvalue="" then
    rstemp(Fieldname)=NULL
    exit sub
end if

if ucase(fieldvalue)="NULL" then
    rstemp(Fieldname)=NULL
else
    rstemp(Fieldname)=fieldvalue
end if
end sub

%>

```

Appendix C. Chat Room Code

C.1 Access To Bridge Support Chat Room Code

C.1.1 default.asp (default file of chat room)

```
<html>
<head>
  <Title>BIC Admin Chatroom</title>

<SCRIPT LANGUAGE="JavaScript">

  function enter(){
    chatWin =
window.open("main.asp","chatWin","toolbar=no,location=no,directories=no,status=no,m
enubar=no,scrollbars=no,resizable=no,WIDTH=580,HEIGHT=540");
    chatWin.location.href = "main.asp"
    if(navigator.appVersion.indexOf("4") != -1){
      chatWin.moveTo(0,0);
    }
  }

  function closeWin(){
    chatWin.close();
  }
</SCRIPT>
</head>

<body Bgcolor="#476895" link=#000000 vlink=#000000>
<table width=422><tr><td><font size=2 face="Trebuchet MS, ms sans serif,
Verdana,Arial" size=2 color=#f0f0f0>
To join a chat room click the 'Enter' button below.</font>
<br><br>
</td></tr>
<tr><td align=center>
<form onSubmit="enter()"><input type=image src="images/enter.gif" border=0
value="Enter ASP Chat" ></form>
</td></tr>
<tr></tr>
</table>
```

C.1.2 main.asp

```

<% Response.Redirect "main2.asp" %>

<html>

<head>
<title>ASP Chat</title>
<script language="JavaScript">

    function closeWin(){
        self.close();
    }

</script>
</head>
<frameset Marginwidth="0" Marginheight="0" Framespacing="0" Border="0"
Frameborder="0"
rows="*,94">
    <frame Name="main-body" Src="main2.asp" Scrolling="auto" Marginwidth="0"
Marginheight="0"
    Framespacing="0" Border="0" Frameborder="NO">
    <frame Name="bottom" Src="bottom.asp" Marginwidth="0"
    Marginheight="0" Framespacing="0" Border="0" Frameborder="NO" scrolling=no>
</frameset>

</html>

```

C.1.3 main2.asp

```

<% Application("filebase") = Server.MapPath("files") & "\" %>

<html>
<head>
    <title>Chat Test</title>
    <script language="JavaScript">
        function verify(form){
            if(form.Name.value == ""){
                alert("Please enter your name!");
            }else{
            }
        }
    </script>
</head>
<body Bgcolor="#476895" TEXT=#f0f0f0>
<%=Request.QueryString("ErrorMessage")%>
<BR>

```

```

<center>
<CENTER>
<!--img src="images/chat.gif"//-->
<br>
<font face="Trebuchet MS, ms sans serif, Verdana,Arial" size=2
color=#f0f0f0><b>Please enter your name and select the room you wish to enter
below.</b>
</center>
<BR>
<BR> <center>
  <center>
    <form name="form_topics" action="chatroom.asp" method="post"
onSubmit="verify(this)">
    <table>

      <tr>
        <td>
          <font face="Trebuchet MS, ms sans serif, Verdana,Arial" size=2
color=#f0f0f0>Name:
        </td>
        <td>
          <%
            If Not IsNull(Session("Name")) Then
              Response.Write "<input type='text' name='Name' value='" &
Session("Name") & "' maxlength=12 maxsize=12>"
            Else
              Response.Write "<input type='text' name='Name' maxlength=12
maxsize=12>"
            End If
          %>
        </td>
      </tr><tr>
        <td>
          <font size=2 face=verdana color=#f0f0f0>Room:
        </td>
        <td>
          <select name="Topic">

            <%
              Topics = Application("TopicNames")
              If IsArray(Topics) Then
                For I=1 To UBound(Topics)
                  Name = Topics(I)
                  Response.Write "<option value='" & Name & "'>" & Name
                Next
              End If
            %>
          </td>
        </td>
      </tr>
    </table>
  </center>
</center>

```

```

        %>

        </select>
    </td>
</tr><tr>
    <td colspan=2 align=right><br>
        <input type="image" src="images/logon.gif" border=0>
    </td>
</tr>
</table>
</form>
</center>
</body>
</html>

```

C.1.4 chatroom.asp

```

<%
    If Request("Name") = "" OR Request("Topic") = "" Then
        Response.Redirect("default.asp?0")
    Else
        Topic = FilterTopic(Request("Topic"))
        Session("Topic") = Topic
        Session("Name") = Request("Name")
    End If

    Name = Request("Name")
    Topic = Request("Topic")

    Const BadName = "<font face=verdana size=+1><b>You selected a name that is already
in use. Please select a different username.</b></font><br><br>"

    If Not IsEmpty(Application("Users")) Then
        If IsArray(Application("Users")) Then
            uLen = UBound(Application("Users"))
            For I = 0 To uLen
                If Application("Users")(I) = Name Then
                    Name = Name & "_1"
                    Session("Name") = Name
                    'Response.Redirect("main2.asp?ErrorMessage=" &
Server.URLEncode(BadName))
                Exit For
            End If
        Next
        Users = Application("Users")

```

```

    UsersGroup = Application("UsersGroup")
    NumOfUsers = UBound(Users)+1
    ReDim Preserve Users(NumOfUsers)
    ReDim Preserve UsersGroup(NumOfUsers)
    'Response.AppendToLog "Number of users when " & Name & " entered: " &
NumOfUsers
    Users(NumOfUsers-1) = Name
    UsersGroup(NumOfUsers-1) = Topic
    Application("Users") = Users
    Application("UsersGroup") = UsersGroup

    UsersIP = Application("UsersIP")
    UsersIdleTime = Application("UsersIdleTime")
    UsersTimeOn = Application("UsersTimeOn")
    ReDim Preserve UsersIP(NumOfUsers)
    ReDim Preserve UsersIdleTime(NumOfUsers)
    ReDim Preserve UsersTimeOn(NumOfUsers)
    UsersTimeOn(NumOfUsers-1) = Now
    UsersIdleTime(NumOfUsers-1) = 0
    UsersIP(NumOfUsers-1) = Request.ServerVariables("REMOTE_ADDR")
    Application("UsersIP") = UsersIP
    Application("UsersTimeOn") = UsersTimeOn
    Application("UsersIdleTime") = UsersIdleTime
Else
    'Response.AppendToLog "Number of users when " & Name & " entered: " & "0
(Not Array)"
    ReDim Users(1)
    ReDim UsersGroup(1)
    Users(0) = Name
    UsersGroup(0) = Topic
    Application("Users") = Users
    Application("UsersGroup") = UsersGroup

    Redim UsersIP(1)
    Redim UsersIdleTime(1)
    Redim UsersTimeOn(1)
    UsersTimeOn(0) = Now
    UsersIdleTime(0) = 0
    UsersIP(0) = Request.ServerVariables("REMOTE_ADDR")
    Application("UsersIP") = UsersIP
    Application("UsersTimeOn") = UsersTimeOn
    Application("UsersIdleTime") = UsersIdleTime

End If
Else

```

```

'Initialize Users Variable
Dim NewUsers()
Redim NewUsers(1)
Dim NewUsersGroup()
Redim NewUsersGroup(1)
NewUsers(0) = Name
NewUsersGroup(0) = Topic
Application("Users") = NewUsers
Application("UsersGroup") = NewUsersGroup

Dim NewUsersIP()
Redim NewUsersIP(1)
Dim NewUsersIdleTime()
Redim NewUsersIdleTime(1)
Dim NewUsersTimeOn()
Redim NewUsersTimeOn(1)
NewUsersTimeOn(0) = Now
NewUsersIdleTime(0) = 0
NewUsersIP(0) = Request.ServerVariables("REMOTE_ADDR")
Application("UsersIP") = NewUsersIP
Application("UsersTimeOn") = NewUsersTimeOn
Application("UsersIdleTime") = NewUsersIdleTime
End If
Session("Enter") = True

Function FilterTopic(Topic)
    Topic = Replace(Topic, ":", ".")
    Topic = Replace(Topic, "/", ".")
    Topic = Replace(Topic, "\", ".")
    FilterTopic = Topic
End Function

%>

<HTML>
<HEAD>
<TITLE><%=Topic%> Chatroom</TITLE>

</HEAD>
<frameset Framespacing="0" Border="0" Frameborder="0" rows="30,*, 50">
    <frame name="chattop" scrolling="no" marginwidth=8 marginheight=0 noresize
src="chattop.asp?Topic=<%=Server.URLEncode(Topic)%>" >
    <frameset marginwidth="0" merginheight="0" framespacing="0" Border="1"
frameborder="1" cols="*,176">

```



```

Else
    Session("RefreshRate") = Session("RefreshRate")
End If

If Session("RefreshRate") = "" Then
    Session("RefreshRate") = 10000
End If
%>

<HTML>
<head>
<SCRIPT LANGUAGE="JavaScript">
<!--

function update(){
    document.location.href="chat.asp"
}

function toBottom(){
    if(document.all){
        window.scroll(0, document.body.clientHeight);
    }else{
        window.scroll(0,20000)
    }
}

setTimeout("update()",<%=Session("RefreshRate")%>);
//-->

</SCRIPT>
</head>
<BODY BGCOLOR="#ffffff" onLoad="toBottom()">
<br>
<FONT SIZE=2 FACE="Verdana,Arial">
<%
    Set fileObject = Server.CreateObject("Scripting.FileSystemObject")
    'Map appropriate topic file
    textFile = Server.MapPath("files/" & Replace(Topic, "+", "_") & ".txt")
    If Not fileObject.FileExists(textfile) Then
        Set inStream = fileObject.OpenTextFile(textFile,8,True,FALSE)
        inStream.WriteLine " "
    End If

```

```

Set inStream = fileObject.OpenTextFile(textFile,1,TRUE,FALSE)

file = inStream.ReadAll
Set inStream = Nothing

If Len(file) > 5000 Then
    'archive the old file
    'archive name is topic_date put under files/archives
    archiveFile = Server.MapPath("files/" & Replace(Topic, "+", "_") & "_" &
Server.URLEncode(Now) & ".txt")
    Set inStream = fileObject.CreateTextFile(archiveFile,True)
    inStream.Write file
    inStream.Close
    Set inStream = Nothing

    Set inStream = fileObject.CreateTextFile(textFile,True)
    inStream.Write " "
    Set inStream = Nothing
    file = " "
End If

If IsNull(Session("Enter")) Or Session("Enter") = True Then
    file = file & "<FONT SIZE=1 FACE='Vedana,Arial' Color=#808080><b>***[" &
name & " has entered (" & Now & ")]***</b></FONT><br>"
    Session("Enter") = False
End If

If Request.Form("Add") = "TRUE" Then
    file = file & "<FONT SIZE=1 Color=#000000 FACE='Verdana,Arial'><b>" &
name & "</b>:</FONT>" & Request("Msg") & "<br>"
End If

Set outStream = fileObject.CreateTextFile(textFile,True)
outStream.WriteLine(file)
Set outStream = Nothing

Response.Write "<font color=#000000>" & file & ""

PrivRcpt = Application("PrivateRcpt")
Match = 0
If IsArray(PrivRcpt) Then
    PrivateMsgs = Application("PrivateMsgs")
    NumOfRcpt = UBound(PrivRcpt)
    ReDim Preserve UpdatedPrivRcpt(1)

```

```

ReDim Preserve UpdatedPrivMsg(1)
For I = 0 To NumOfRcpt
    If Name = PrivRcpt(I) Then
        Response.Write "<FONT color=#ff0000 SIZE=1
FACE='Verdana,Arial'><b>*** " & PrivateMsgs(I) & " ***</b></font>"
    Else
        ReDim Preserve UpdatedPrivRcpt(Match)
        ReDim Preserve UpdatedPrivMsg(Match)
        UpdatedPrivRcpt(Match) = PrivRcpt(I)
        UpdatedPrivMsg(Match) = PrivateMsgs(I)
        Match = Match + 1
    End If
Next
Application("PrivateRcpt") = UpdatedPrivRcpt
Application("PrivateMsgs") = UpdatedPrivMsg
End If

%>
</FONT>
<br>
<br>

<br>

</BODY>
</HTML>

```

C.1.7 user.asp

```

<html>
<head>
<SCRIPT LANGUAGE="JavaScript">
<!--

    setTimeout("update()",15000);

    function update(){
        document.location.href="users.asp"
    }

    function privateWin(user, from){
        url = "privatemsg.asp?user=" + user + "&from=" + from;
        privatemsg =
window.open(url,"privatemsg","toolbar=no,location=no,directories=no,status=no,menub
ar=no,scrollbars=no,resizable=no,WIDTH=400,HEIGHT=200");

```

```

    }
  //-->

</SCRIPT>
</head>
<body topmargin=0 leftmargin=0 marginwidth=0 marginheight=0 BGCOLOR="#ffffff"
TEXT=#000000 LINK=#000000 VLINK=#000000>
<table width=186 border=0 cellspacing=0><tr><td bgcolor=#cccccc width=100%
align=center>
<font face="Trebuchet MS, ms sans serif, Verdana,Arial" size=2
color=#476895><b>Who's Online</b></font>
</td></tr></table>
<br>
<font face='verdana,arial' size=-1 0>
<%

    Topic = Session("Topic")

    Application.Lock

    If Not IsEmpty(Application.Contents("Users")) Then
        Users = Application("Users")
        Group = Application("UsersGroup")
        NumOfUsers = UBound(Users)
        For I = 0 To NumOfUsers
            If Topic = Group(I) Then
                Response.Write ""
                Response.Write "&nbsp;<a href=""javascript:privateWin(' " &
Server.URLEncode(Users(I)) & ", " & Server.URLEncode(Session("Name")) &
""'"><img align=absmiddle src='images/lips.gif' border=0> <b>" & Users(I) &
"</b></a><br>"
                Response.Write "&nbsp;<small><b>IP:</b> " &
Application("UsersIP")(I) & "<br>"
                Response.Write "&nbsp;<b>In:</b> " &
Application("UsersTimeOn")(I) & "</small><br><br>"
            End If
        Next
    End If

    Application.Unlock

%>

<br><br><br>

```

```
<font face="ms sans serif" size=1 color=#476895><b>Click name to send<br>private message </b></font>
```

```
</body>  
</html>
```

C.1.8 user.asp

```
<HTML>  
<HEAD>
```

```
<script language="javascript">
```

```
function handle(form){  
    var text = form.visMsg.value  
    if(text == "") return false;  
    form.Msg.value = text;  
    form.RefreshRate.value =  
document.forms[1].inteval[document.forms[1].inteval.selectedIndex].value  
    form.visMsg.value="";  
    form.visMsg.focus();  
    return true;  
}
```

```
function exitChat(){  
    parent.frames.location.href='exit.asp';  
    return false;  
}
```

```
</script>
```

```
</HEAD>
```

```
<BODY BGCOLOR="cccccc" TEXT=#000000  
onLoad="document.message.visMsg.focus()">
```

```
<TABLE><TR><TD ALIGN=CENTER VALIGN=TOP>  
<FORM NAME="message" TARGET="chat" ACTION="chat.asp" METHOD="POST"  
onSubmit="return handle(this)">  
<font face="ms sans serif" size=1>Message:</font>  
<input type="text" size=40 name="visMsg">  
<input type="hidden" Name="Msg" value="">  
<input type="hidden" Name="RefreshRate">  
<input type="hidden" Name="Add" value="TRUE">  
</TD>  
</form>  
<form onSubmit="return exitChat()">
```

```

<td valign=top>
<font face='ms sans serif' size=2 color=#000000>
Refresh Rate: <select name="intreval">
<option value="10000">10 secs.
<option value="20000">20 secs.
<option value="30000">30 secs.
<option value="45000">45 secs.
</select>
</td><td valign=bottom>
<INPUT type="image" border=0 src="images/logoff.gif">
</FORM>

</TD></TR></TABLE>
</CENTER>
</BODY>
</HTML>

```

C.1.9 exit.asp

```

<%
    Topic = Session("Topic")
    Name = Session("Name")

    Application.Lock

    Users = Application("Users")
    UsersIP = Application("UsersIP")
    UsersTimeOn = Application("UsersTimeOn")
    UsersIdleTime = Application("UsersIdleTime")

    Match = 1
    If IsArray(Users) Then
        ' Response.Write Name & "<br>IsArray=True<br>"
        UsersGroup = Application("UsersGroup")
        NumOfUsers = UBound(Users)
        If NumOfUsers = 1 Then
            ' Response.Write "UBound(Users)=1<br>"
            Application("Users") = ""
            Application("UsersGroup") = ""

            Application("UsersIP") = ""
            Application("UsersIdleTime") = ""
            Application("UsersTimeOn") = ""

```

```

Else
    ReDim UpdatedUsers(1)
    ReDim UpdatedUsersGroup(1)

    ReDim UpdatedUsersIP(1)
    ReDim UpdatedUsersTimeOn(1)
    ReDim UpdatedUsersIdleTime(1)
    For I = 0 To NumOfUsers-1
        ' Response.Write "At " & I & " User is:" & Users(I) &
"<br>"
        If Name <> Users(I) Then
            ReDim Preserve UpdatedUsers(Match)
            ReDim Preserve UpdatedUsersGroup(Match)
            UpdatedUsers(Match-1) = Users(I)
            UpdatedUsersGroup(Match-1) = UsersGroup(I)

            ReDim Preserve UpdatedUsersIP(Match)
            ReDim Preserve UpdatedUsersTimeOn(Match)
            ReDim Preserve UpdatedUsersIdleTime(Match)
            UpdatedUsersIP(Match-1) = UsersIP(I)
            UpdatedUsersTimeOn(Match-1) = UsersTimeOn(I)
            UpdatedUsersIdleTime(Match-1) = UsersIdleTime(I)

            Match = Match + 1
        End If
    Next
    Application("Users") = UpdatedUsers
    Application("UsersGroup") = UpdatedUsersGroup

    Application("UsersIP") = UpdatedUsersIP
    Application("UsersTimeOn") = UpdatedUsersTimeOn
    Application("UsersIdleTime") = UpdatedUsersIdleTime

    End If
End If

Application.UnLock

Set fileObject = Server.CreateObject("Scripting.FileSystemObject")
textFile = Server.MapPath("files/" & Replace(Topic, "+", "_") & ".txt")
Set inStream = fileObject.OpenTextFile(textFile,1,TRUE,FALSE)
file = inStream.ReadAll
Set inStream = Nothing

```

```
file = file & "<FONT SIZE=1 FACE='Vedana,Arial' Color=#808080><b>[" & Name  
& " has exited (" & Now & ")]</b></FONT><br>"
```

```
Set outStream = fileObject.CreateTextFile(textFile,True)
```

```
outStream.WriteLine(file)
```

```
outStream.Close
```

```
Set outStream = Nothing
```

```
'#####
```

```
'Should just call Session.Abandon
```

```
'#####
```

```
'Session.Abandon
```

```
%>
```

```
<html>
```

```
<head>
```

```
<script language="javascript">
```

```
function closeWin(){
```

```
if(opener){
```

```
self.close();
```

```
}else{
```

```
//customize this to go where you want
```

```
window.document.location.href="default.asp";
```

```
}
```

```
}
```

```
</script>
```

```
</head>
```

```
<body onLoad="closeWin()">
```

```
</body>
```

```
</html>
```

Appendix D. References

1. Ashenfelter, John Paul. "Designing Good Databases". [Http://www.webreview.com](http://www.webreview.com). March 26, 1999.
2. Cambra, Troy. "Developing a Visual Basic Component for IIS/MTS". *Microsoft Internet Developers*. February 2001.
3. Campbell, Bruce and Rick Darnell. *Teach Yourself Dynamic HTML in a Week*. Indianapolis: SamsNet 1997.
4. Hillier, Scot and Daniel Mezick. *Programming Active Server Pages*. Washington: Microsoft Press, 1997.
5. MacFarlane, Bryan and Donovan Smith. "Write COM Components for ASP in Visual Basic". *Microsoft Internet Developers*. September 1999.
6. Rofail, Ash and Tony Martin. *Building N-Tier Applications with COM and Visual Basic 6.0*. Canada: John Wiley and Sons, Inc., 1999.
7. Schurman, Eric M. and William J. Pardi. *Dynamic HTML Interaction, Second Edition*. Washington: Microsoft Press 1999.
8. Spenik, Mark, Andrew Indovina, David Jung, and Pierre Boutquin. *Teach Yourself Visual Basic 6*. Indianapolis, IN: Sams Publishing. 2000.
9. Thau, Dave. *The Book of JavaScript*. San Francisco, CA: William Pollock 2000.
10. Viescas, John. *Running Microsoft Access 2000*. Redmond, Washington: Microsoft Press 1999.
11. Weissinger, A. Kemton. *ASP in a Nutshell, Second Edition*. Sebastopol, CA: O'Reilly & Associates, Inc 1999.
12. Wynkoop, Stephen. *Using SQL Server 7.0*. Indianapolis, IN: Que 1999.