Distance Education at the College of Applied Science

by
John Nyland
Michael Olivera
Sean Snider

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
OMI College of Applied Science

March 2000

Distance Education at the College of Applied Science

by
John Nyland
Michael Olivera
Sean Snider

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
OMI College of Applied Science

John Nyland	Date 3/15/00
Michael Olivera	Date 3/15/00
Sean Snider	Date 3/15/00
Dr. Vladimir Uskov, Faculty Project Advisor	Date 3/10/00
Dr. Kim Myers, Department Head	Date 3/14/2000

Acknowledgements

The authors wish to express their sincere gratitude to the faculty of the Information Engineering Technology Program, without who's help the successful completion of this project would not have been possible. We would like to make a special acknowledgement of the contribution made by Dr. Vladimir Uskov, our project advisor. We thank him for his guidance, support, and for giving freely of his time and resources.

Table of Contents

List of Illustrations

Abstract

This report covers distance education at the College of Applied Science, University of Cincinnati, using the technology of Web Based Learning. The purpose of this project was to investigate and demonstrate a methodology by which traditional classroom pedagogy could become web-enabled using the curriculum of the Information Engineering Technology Program within the existing University of Cincinnati infrastructure. The report covers the various technologies which were investigated, and describes what was accepted and adopted to the project, as well as what was rejected and why.

Within the report the rationale for the use, and method of use, of various technologies is given, including SQL Server, WebCT, ADO, HTML, XML, Java Script, Visual Interdev, FrontPage, Visual Basic, Visual Basic Script, NetMeeting, Microsoft Site Server, Microsoft Transaction Server, Active Server Pages, as well as numerous hardware requirements. The structure, flow, and management of the eCAS website is explained, and plans for future expansion of the project are provided.

Distance Education at the College of Applied Science

1. Statement of the Problem

Advances in telecommunication technologies, changing student demographics and the ever-present need of universities to increase and sustain enrollment has renewed interest in distance education. However, distance education differs from traditional classroom instruction in two significant respects: instructors and students are separated by geographic distance and communications between them are mediated by technology. These differences have significant implications for the design and delivery of instruction. Distance education is not just an old class with new technology. Effective design and delivery requires considerable effort by a team of instructors, course designers, technicians, administrators, and students.

Over the past century, distance education has evolved through several generations (Moore & Kearsley, 1996)[16]. In the mid-to late 1800s, correspondence or home study became a legitimate form of education with the development of inexpensive postal services in Europe and across the United States. Instructors sent readings, study guides and other printed materials by mail to students who earned credit for successfully completing specified assignments. This correspondence model spawned the first generation of distance education, and is still the predominant model outside of the United States.

The advent of open universities and the use of broadcast and recorded media spawned the second model of distance education. In the late 1960's and early 1970's educators began to experiment with the use of radios, televisions, audiotapes, and

telephones in combination with study guides and local library resources to provide educational opportunities to students who were isolated by geographic distance.

Advances in interactive and digital technologies formed the basis for the third generation of distance education. Currently, educators are delivering instruction to students in remote locations via interactive television, electronic networks and computer-based multimedia systems. The primary difference between second and third generation distance education is that the third generation allows educators and students to communicate through both synchronous (real-time) and asynchronous audio, video, text and/or graphics.

We are now beginning to see the development of a fourth generation of distance education, which combines the concepts associated with Electronic Performance Support Systems (i.e., tools, information, training, and advisement) with network based delivery systems (Hirumi, 1997)[16]. These programs provide users with just-in-time training, information and support through both synchronous and asynchronous audio, video, text or graphic communications provided by a computer or by a person.

Distance education is becoming important for several reasons. First, as we continue to move into the information age, the exponential accumulation of information and rapid changes in information and technology necessitates on-going education and professional development. Conservative estimates suggest that to remain competitive, individuals will require at least 30 credit hours of instruction every 7 years[22]. That's almost equivalent to a master's degree every 7 years. Second, student demographics continue to change. A much higher percentage of university students are older, have children and work full-time. Convenience is important. Third, to address the changing

student demographics, many institutions of higher education must provide distance education if they are to sustain or increase enrollment. [12]

As the Internet is fast emerging, the World Wide Web (WWW) is becoming an increasingly powerful, global, interactive and dynamic medium for delivering instruction. An increasing number of institutions are using the Web to provide instruction and training. These institutions often offer Web-based courses and are recognizing the fact that the Web is a viable medium of learning and instruction.

There are currently over 1300 different institutions offering various levels of Web based training, education, and degree programs on the Internet. This number is expected to grow geometrically over the next decade. The University of Cincinnati is currently not involved in this flourishing market. Unless we take action our university could be shut out entirely within five years, because by then the dominant web-educations will have taken the majority of the market.

2. Review of the Literature

By allowing for the use of multimedia, the World Wide Web is fast becoming the tool of choice for distance educators, hence the term Web-based Training. This is one of the newest forms of distance education, where communication between and among students, the instructors and possibly others is mediated through the use of various Internet or Intranets resources (e.g., e-mail, listservs, gopher, IRC, WWW). The popularity of Web-Based Training is primarily due to the prevalence of the Internet. Estimates are that over 50 million people in 90 different countries are now connected. [13]

There are basic differences between Informational Web-Sites, Pseudo Web-Based Training, and Systematically Designed Web-Based Training. Informational Web Sites contain text or other media to communicate a message from sender to receiver. Pseudo Web-Based Training contains a series of activities to facilitate learning based on past practices, fads, or personal opinions. In comparison, systematically designed Web-Based Training should contain a series of events to facilitate learning that is based on a combination of practical experience, learning theory and research.

Chute, Alan G., Melody Thompson and Burton Hancock. *The McGraw-Hill Handbook of Distance Learning: A "How to Get Started Guide" for Trainers and Human Resources Professionals.* New York: McGraw-Hill, 1998.

This book is for human resources managers, trainers, and other who want to get started using distance-learning technologies like videoconferencing and online multi-media training modules. It is geared towards minimizing the cost of training everyone in an organization. Practical advice is offered to help get organizations started in distance learning, or how to improve an existing distance learning program. Topics covered

include how to apply distance learning to new product, policy information, job skills, training/retraining, advanced professional education, management development courses, and customer education. In his comprehensive guide, the author addresses the needs and concerns of human resource and training managers who want to help their organizations implement distance learning programs but don't know where to start.

Phillips, Vicky and Cindy Yager. *The Best Distance Learning Graduate Schools 1999 Earning Your Degree Without Leaving Home.* Princeton Review, 1999

The author of this text concentrated on the future of higher education. Her claim is that computers and other technologies have made it easier than ever to make graduate school a reality by bringing the classroom directly into the home. She says about 55 percent of colleges in the United States, and many abroad, offer some type of distance learning degree program. This book shows how to find exactly the program being sought and answers many of the questions one may have along the way, such as: What method of delivery (Internet, videocassettes, etc.) is best for me? Can I get credits for my career experience? Do I need to travel to campus at all? And can I design my own degree? How schools are accredited, and how to avoid rip-offs and diploma mills is also covered. The latest edition also includes an information-packed directory of 195 accredited institutions that offer graduate degrees via distance learning. Each entry includes everything a perspective student need know about prerequisites, admissions, tuition, campus visit requirements, and delivery methods, plus all pertinent addresses (including email). Here are just some of the study areas represented inside:

- Business Administration
- Computers and Information Systems

- Counseling Psychology
- Creative Writing
- Criminal and Correctional Justice
- Education
- Engineering
- Health Care Administration
- Human Resources
- Library Science
- Pharmacy
- Public Health
- Public Administration
- Telecommunications
- Women's Studies

We also visited and analyzed the following sites:

- Colorado Electronic Community College
- Eastern Oregon State University
- Education Network of Maine
- Front Range Community College
- Intel University
- Mind Extension University
- National Technological University
- Northern Arizona University
- Northwest Indian College
- Novell Education
- Open Learning Agency
- Rochester Institute of Technology
- University of Alaska Southeast
- University of California – Los Angeles
- University of North Dakota
- University of Phoenix
- University of Wyoming
- Utah Education Network
- Washington State University
- ZD University

Analysis of Existing Colleges

A review of 18 cases published by The Western Cooperative for Educational Telecommunications, Western Interstate Commission for Higher Education[19] shows that today's organizations choose from a large array of technologies and techniques as they attempt to fulfill their missions and meet the practical constraints faced by their students. In addition to conventional classrooms and printed text a given organization might use video (one-way or two-way; digitally compressed or not; delivered by broadcast, cable TV, satellite, optical fiber, microwave, phone lines, or shipped on tape); voice communication (phone, voice mail, audio conferences and audio cassettes); computers (e-mail; computer conferences; computer graphics and simulations; CD-ROM, computer

chat sessions; via local networks, the Internet, and the World Wide Web); and fax; mail, express services and courier.

3. General Description of Proposed Solution

We intend to provide CAS with a Web based distance learning application. This will involve the installation and configuration of a Web server for the training/education application, a SQL server for enrollment, payments, testing and viewing grades, sample courses (OOP I, OOPII, UNIX), and sample tests (OOP I, OOPII, UNIX).

3.1 User Profile

The typical user for our system will be undergraduate students at the University of Cincinnati's College of Applied Science, Information Engineering Technology Program. Since the courses currently offered on the site have prerequisites, we can assume a level of computer proficiency greater than that of the general public. Given that this is not a true commercial site, but an extension of the university into cyberspace, certain provisions can be stipulated to our users. These stipulations will include the use of Internet Explorer 5.0 and the mandatory installation of NetMeeting on student computers. We foresee no problems with our users being able to extract maximum advantage from this Web site.

3.2 Design Protocols

We have calculated the typical bandwidth needed for conferencing. This means assessing the number of conferences that can be expected to run concurrently, and the

expected number of users in each. From this data the typical bandwidth required has been determined.

We have reviewed the current performance of the college's network. We predicted traffic growth in order to address the question of sufficient scope to add conferencing. If the network is already full, any additional applications will cause problems. We looked at predicted growth of traffic, remembering that the actual growth is likely to be twice that predicted as mandated by design standards. Can our network support conferencing on top of existing and already planned applications? Yes it can. We will attempt to anticipate bottlenecks and consider increasing network capacity where they are likely to occur. Problem areas are likely to include buffers located in bridges, routers and switches, as they are often a major bottleneck in any network. This was clearly demonstrated by experiences in the IET computer lab.

We have addressed the question of capacity being available at the times when it will be needed, by comparing the usage cycle of our network with the anticipated frequency and duration of multimedia conference calls. In addition, while users are in a conference they are unlikely to be generating any other network traffic and conferences are unlikely to coincide with busy network periods, such as the early morning e-mail rush. Therefore, in our assessment of network capacity we considered the increase caused by conference traffic, not the absolute conference traffic.

We needed to determine whether or not conference traffic would cross between sites over WAN links. If so, we may have needed to consider a traffic priority scheme on those links to avoid disruption of either the conference or of business critical applications.

Another important question was: will our subnet structure support the expected conferencing traffic flows? First, we analyzed the likely conference traffic flows. Then we compared our data with the current subnet structure. It was not necessary to upgrade that structure.

We considered the network, end-to-end delay, and the variability of this delay also known as jitter. These problems often cause more difficulty than lack of bandwidth. We considered the maximum acceptable end-to-end delay perspective users would accept. Again, anything with a buffer can add appreciably to that delay.

We looked at the expected conferencing routes across the network to decide on the locations of conference components. The conference server and gateway are placed strategically, considering traffic flows, so they are closest to the sources of maximum usage to avoid creating unnecessary network traffic.

Conference traffic, unlike other types, can be tightly controlled so it cannot break our network unless we let it. We will only allow use to grow at a predetermined rate. We can then measure demand, and use our measurements to justify any increase in budget necessary to augment capacity.

4. Objectives of the Project

We have delivered a comprehensive web based distance learning application installed on a CAS server where students can register, select classes, view courses, interact with professors, and take tests. Students may receive announcements, lecture notes, course handouts, receive assignments and join in discussion groups. Students may also use our video conferencing page to chat, share applications, transfer files, and see their professors. Students will be able to view streaming video, and over 1200 Power Point slides of course material.

Professors undertake many tasks on this site. Using our system they will be able to develop and disseminate course materials such as announcements, lecture notes, course handouts, and assignments. If they choose, professors will be able to videotape lectures and place them on the server for students to view. Professors will be able to take control of a student's remote computer and show them how to perform tasks, such as write functioning code. On this site a professor can reach an ever-increasing number of students with a minimal increase in effort.

On this site, administrators can edit records of students, courses, the courses applicable to students, and edit professors' office hours to coincide with each quarter. The site is fully functioning, and will provide the model for a full production site with little modification.

In the concluding section of this we report we recommend the installation of the Faculty Technology Resource Center. This is already underway. The Faculty Technology Resource Center's function is to support the distance learning activities of the College of Applied Science.

5. Design and Development

The design and development aspect of this project has been ongoing throughout the project. The vast number of considerations, variables, and the acquisition of technical "know how" have occupied the authors' time over the past months. Now that this has been accomplished, we feel confident that we could replicate, or extend, this project with comparative ease.

5.1 Budget

Item	*Cost*
Compaq Prosignia Server	$8,500.00
Visual Studio Enterpise Edition	$700.00
MS Office Server Extensions	$0.00
SQL Server 7.0	$1,399.00
10 Zip Disks	$90.00
WMT CD	$11.95
Total	$10,700.95

Figure 1. List of Hardware Deliverables and Associated Costs.

5.2 Timeline

Time Schedule and Time Considerations

1. What can we accomplish this quarter?
2. Literature analysis
3. Analysis of web sites
4. Analysis of existing systems
5. State of the art features of systems
6. Analysis of appropriate programming languages
7. PowerPoint Presentation of Results

8. How much time for each task?

Literature analysis	3 weeks
Analysis of web sites	2 weeks
Analysis of existing systems	2 weeks
State of the art features of systems	1 week
Analysis of appropriate programming languages	1 week
Power Point presentation of results	1 week
Build initial prototype	10 weeks
Work prototype into working site	8 weeks
Testing and modification of site	2 weeks
Preparation of documentation	3 weeks
Preparation of presentation	2 weeks

5.3 Software

Web CT

Throughout the course of our research, we have been exposed to many software tools. None of the web development tools we considered initially met our needs as well as WebCT. WebCT is in widespread use. In the first 16 months it has been in business vendors have sold over 1200 licenses to institutions in over 33 countries. WebCT has been chosen by many of the largest higher education publishers as their tool for building web-based courses to complement their textbooks. Most notably, Prentice-Hall, Harcourt

Brace, John Wiley and Sons, Addison Wesley, W. W. Norton and Course Technology are using WebCT to create online material that supplements their existing textbooks.[14]

WebCT has a comparatively low cost. It began as a university-based project. The goal of its developers is to charge just enough to provide support, maintenance and continued development. WebCT is free to download, install and to use it to create courses. A license is required before one can use a created course to educate students. WebCT's developers even provide first-time users with free use of WebCT to create one course, hosted on their server, for up to 50 students.

WebCT is a tool built by instructors for instructors. It was built by educators at the University of British Columbia as a tool to aid other educators without a lot of time, resources or technical expertise in construction of sophisticated web-based learning environments. This tool still manages to have a broad and deep feature set. WebCT also allows for web pages to be developed by any authoring tool.

To accommodate the needs of international students, WebCT's student interface is available in many languages, including English, Spanish, French, Dutch and Finnish. This feature will tie in nicely with CAS's international cooperation projects.

The bad news is that the current WebCT 2.0 is only in beta release, and undertesting, became unstable. WebCT continually lost pass words. For these reasons we chose to drop the use of the product late into our production schedule.

MacroMedia Attain

Dreamweaver Attain is Dreamweaver, but with an added set of functionality specific to creating complex interactions for learning. This added functionality provides Web and courseware developers with ways of creating complex interactions that might

otherwise be too difficult or time-consuming to hand-code and track back to a management system.

Dreamweaver Attain's functionality, while impressive, is difficult to learn. The interface of the program is not as familiar as a Microsoft product like InterDev. Given the serious time constraints of the Senior Design process, and the workload of this project, we can not afford the luxury of using this product. As the project develops over time, perhaps within the next year, students may enjoy the opportunity to use this product.

Visual Interdev

Microsoft Visual InterDev 6.0 is the latest version of an integrated web application development system for professional programmers. The new version enables Web teams to design, build, debug, and deploy cross-platform Web applications. Visual InterDev 6.0 also features a new integrated WYSIWYG editor for ASP & Dynamic HTML pages, enhanced database programming tools, and end-to-end debugging facilities for multi-tier applications built with HTML and Script.

FrontPage

Microsoft FrontPage is a quick, effective way to create and manage professional-quality Internet or Intranet sites without programming. It makes it easy for new users and professional Web developers to build and maintain great looking, professional-quality Web sites in no time, and easily create great-looking Web sites. Easy-to-use features let you create professional Web sites without programming. You can create WYSIWYG frames pages and draw HTML tables in the WYSIWYG FrontPage Editor, then drop in what is needed using FrontPage components. Management tools allow the quick building and maintenance of well-organized Web sites.

Visual Basic

More than 50 percent of all professional developers are using the Visual Basic language. Visual Basic 6.0 is a productive tool for creating high-performance enterprise and Web-based applications. Integrated Visual Database Tools and a RAD environment promote productivity while native code compilation provides fast applications. We used the Visual Basic 6.0 integrated Visual Database Tools and Data Environment Designer to design our Microsoft SQL Server database and create reusable data access queries, all without leaving the Visual Basic environment.

Visual Basic Script

Visual Basic Scripting Edition (also known as VBScript) is a subset of the Microsoft Visual Basic language. It is implemented as a fast, portable, lightweight interpreter for use in World Wide Web browsers and other applications that use ActiveX Controls, OLE Automation servers, and Java applets. Visual Basic Scripting Edition is available from Microsoft in two forms: as a ready-to-run (compiled) binary for several platforms, or as source code. When used in Internet Explorer, VBScript is directly comparable to JavaScript (not Java). Like JavaScript, VBScript is a pure interpreter that processes source code embedded directly in the HTML.

NetMeeting Software Development Kit

As the name suggests Netmeeting SDK is Microsoft's answer to personalized applications using Netmeeting's interactive environment. The full capabilities of Netmeeting are described below in the Electronic Commerce section. The kit includes Active X Controls useable in the Visual Basic environments described above. We intend to use this kit to develop a personalized video conferencing environment for users who

wish to confine their calls to on campus locations. Because all on campus computers have static Internet Protocol Addresses, they can be used to call machines directly reducing network traffic and response time. Other possible applications include direct calls from a Web-based course to that course's instructor during office hours.

Electronic Commerce

Electronic Commerce (E-commerce, or EC) is the buying and selling of goods and services on the Internet, especially the World Wide Web. In practice, this term and a new term, "e-business," are often used interchangeably.

Part of the learning experience is the joy of paying for what has been learned. All students are familiar with the cashier's office. The on-line student will need an on-line cashier and registrar's office. We hope to provide this service with a suite of Microsoft products that are, in some cases, already present in Windows NT Server.

Microsoft Site Server 3.0, Commerce Edition

Microsoft Site Server Commerce Edition is the comprehensive Internet Server, optimized for Microsoft Windows NT operating system with Internet Information Server (IIS), which enables businesses to engage and transact with customers and partners online. Site Server Commerce helps businesses engage customers and partners with the creation of cost-effective commerce sites and applications, targeted online advertising and marketing, and personalized promotion. Based on strong integration with Windows NT Server, it allows businesses to transact online with secure and scalable order capture, management, and routing while integrating more easily into existing inventory, accounting, and EDI systems. Site Server Commerce Edition provides tools to help

business managers analyze customer and partner usage data and factor analysis into changes to help maximize the effectiveness of their online business.

Microsoft Transaction Server

Microsoft Transaction Server oversees the components running in an application and adjusts their operation to ensure that they do not fail. Transactions make complex and sophisticated database applications possible. Transactional Web applications work in IIS 4.0 through:

- Developing components.
- Scripting Web applications.
- Using transactions.
- Using the transactional Web server.
- Using IIS 4.0.

Using transactions:

At the most basic level, transactions ensure that data is protected in databases. A transaction monitor watches over what an application does and only updates records in a database if all of the changes the application wanted to make across all databases are successful. Otherwise, no records are changed in any of the databases.

The classic example of this is a bank transaction. In the New World of Internet banking, it is now possible to go to a Web page and transfer money from a savings account to a checking account. This application involves two steps: subtracting money from one account, and adding the same amount to another account. However, this transaction should only happen if both steps are successful. Money should be removed from the savings account only when it is transferred to the checking account.

Making this transaction work, such that the money is deducted from savings if and only if the same amount of money is deposited in checking, requires a transaction

monitor to manage the application. Microsoft Transaction Server (MTS) provides this functionality to applications running on Windows NT Server, including Internet Information Server 4.0 (IIS).

Microsoft SQL Server 7.0

SQL Server is the best database for the Microsoft Windows platform and the relational database management system of choice for a broad spectrum of corporate customers and independent software vendors (ISVs). SQL Server delivers a flexible, powerful platform that scales up to terabyte-size databases and down to small business servers and laptop databases; it fits seamlessly with existing applications and provides the most cost-effective environment for customizing and building new applications tailored to unique business needs.

Developing components:

Network applications that run on the server and access databases can be built two ways: business logic can reside in a particular database, or it can be packaged into components. The advantage of components is that business rules can be generalized to work with any database, and be reused in multiple applications.

Windows NT Server and Internet Information Server make it easy to create components that can talk to almost any database. Open Database Connectivity (ODBC) allows us not only to write a component with business logic that accesses a database once but also reuse it with over 55 different databases running on Windows NT Server, Unix, or IBM mainframe servers.

By putting business logic into components that can work with multiple databases, the components become building blocks for applications. The same components can be

used for both client/server and Web-based applications. Reusing components speeds development time and reduces overhead.

Scripting Web Applications:

Components are ideal for creating Web applications. Because they package together business logic in reusable modules, developing a Web application requires only assembling the proper components, using scripts to tie them together, and designing the HTML for user interaction and displaying the results.

Active Server Pages:

ASP in IIS 4.0 allows the use of scripting languages like Visual Basic Script and Java Script to pass information to components, instruct the components to perform a programmed action, and return results. ASPs support standard HTML or advanced dynamic HTML for the application's interface. This means that applications can be accessed from any Web browser.

In the bank scenario, a user would go to a Web page and enter the amount of the transfer in a form. This form would be submitted to IIS and passed to an ASP file. The ASP file would read the transfer amount from the form, and instruct the bank transfer component to transfer the money from the user's savings account to their checking. The component would then access the savings account database to subtract the transfer, and then the checking account to make the deposit. However, since we only want the savings withdrawal to happen if the checking deposit is successful, the application must use a transaction monitor like MTS. This is easy to do with IIS 4.0.

In order to perform a transaction, the ASP file containing the application script simply needs to include a command at the beginning of the script to declare that a

transaction is beginning. When IIS processes this ASP, this script tells it to pass the request to MTS. Because the transfer component knows nothing about transactions and what is required to make them work, MTS handles everything.

All the application developers need to know is what they want to happen. They don't have to worry about things like threading, context, state, connection pooling, service logic, queues, and shared data. MTS provides all this plumbing that is required for scalable and reliable server applications.

IIS 4.0 makes it easy to write scalable and reliable applications for delivery over the Web. The features of MTS are available to all ASP and ISAPI applications. These applications can easily use transactions that access multiple databases with true two-phase commit. IIS 4.0 frees us to focus on what is really important in an application- the business logic.

IIS 4.0 supports real transactions, not database record locking. While database locking, such as that provided in ODBC, could be used in the bank scenario to hold committing the savings withdrawal until the checking deposit is complete, doing this is a poor model. Record locking consumes precious database resources while holding for the next request. It makes information unavailable to other users, and can potentially freeze an entire database. Record locking does not scale. Serious business applications require the power of a transaction monitor like MTS in IIS 4.0.

IIS 4.0 also provides applications and components with crash protection. Many applications can be running at once, but each in their own memory space, so a misbehaved component will not bring down another application or the entire server.

Components for IIS 4.0 Web applications can be written in almost any programming language, and with almost any tool. IIS 4.0 is language agnostic. In fact, MTS is the first product to provide full transaction support for Java components.

Using IIS 4.0:

Server-side scripting, components, and transactions open up a whole New World for Web development. ASP pages can aggregate data from many different sources and display them together in a Web browser. With Dynamic HTML, the Web page becomes interactive, making it easier for students to analyze the data and take meaningful action. Support for real transactions means that Web applications are secure, reliable, and scalable.

ADO (ActiveX Data Objects)

ADO (ActiveX Data Objects) is an application program interface (API) from Microsoft that lets programmers writing Windows applications get access to relational and nonrelational databases from both Microsoft and other database providers. For example, if one wanted to write a program that would provide users of a Web site with data from an IBM DB2 database or an Oracle database, one could include ADO program statements in an HTML file that one then identified as an Active Server Page (ASP). When a user requested the page from the Web site, the page sent back could include appropriate data from a database, obtained using ADO code.

Like Microsoft's other system interfaces, ADO is an object-oriented interface. It is also part of an overall data access strategy from Microsoft called Universal Data Access (UDA). Microsoft says that rather than try to build a universal database as IBM and Oracle have suggested, why not provide universal *access* to various kinds of existing and

future databases? In order for this to work, Microsoft and other database companies provide a "bridge" program between the database and Microsoft's OLE DB, the low-level interface to databases. OLE DB is the underlying system service that a programmer using ADO is actually using. A feature of ADO, Remote Data Service, supports "data-aware" ActiveX controls in Web pages and efficient client-side caching. As part of ActiveX, ADO is also part of Microsoft's overall Component Object Model (COM), a component-oriented framework for putting programs together.

ADO evolved from an earlier Microsoft data interface, Remote Data Objects (RDO). RDO works with Microsoft's ODBC to access relational databases, but not nonrelational databases such as IBM's ISAM and VSAM.

NetMeeting/ ILS Server

As the leading Internet conferencing solution, Microsoft NetMeeting has become the key building block for vendors of conferencing products and services. NetMeeting 2.1 is the most recent release, which is designed to support new technology featured in Windows 98, including DirectX 5, universal serial bus (USB) video cameras, and the new video device driver model. NetMeeting 2.1 is packaged as part of Windows 98, but is designed to run also as a stand-alone product on Windows 95 and Windows NT 4.0 (with Service Pack 3) operating systems. Connecting to other NetMeeting users is made easy with the Microsoft Internet Locator Server (ILS), enabling participants to call each other from a dynamic directory within NetMeeting or from a Web page. While connected on the Internet or corporate Intranet, participants can communicate with both audio and video, work together on virtually any 32-bit Windows-based application, exchange or mark-up graphics on an electronic whiteboard, transfer files, or use the text-based chat

program. The ILS enables users to locate each other for conferencing. Users can view the ILS directory from within NetMeeting or a Web page and review a list of people currently running NetMeeting. They can choose to connect to one or more of the listed users or select another user by typing the user's location information. NetMeeting can detect whether a server is available and automatically attempt to log on in the background, without user intervention. If a user disconnects and then logs on again later, NetMeeting automatically logs on to the specified ILS. However, since this system is doing point-to-point videoconferencing, ILS will not be needed for this project.

5.4 Hardware Components

(2) Servers. One each for Databases, and Courses:

- Pentium III 550 or faster
- 512MB Ram or more
- 54 Gigabytes or more of hard drive storage

Item

Compaq Prosignia Server
Visual Studio Enterpise Edition
MS Office Server Extensions
SQL Server 7.0
10 Zip Disks
WMT CD

Figure 2. List of Hardware

A SQL sever is already in place at the College of Applied Science, Information Engineering Technology Program. Had we needed to purchase one for this project, the

cost would have been similar to that of the Compaq Prosignia Server, and would have raised the total cost of our hardware to $19,000.95 .

6. Proof of Design

The proof of our design lies in our site's full functionality and completion of our project objectives. In the following sections the main Web pages are shown, along with an explanation of their workings, and thusly will attest to the soundness of our project.

We have, with this project, helped CAS to meet the distance education needs of its students now and in the foreseeable future. This is being achieved through the use of well-designed web pages that allow the student to download syllabi, notes and assignments, and interact with professors during office hours via Microsoft Netmeeting's functions. We have worked to add capabilities that will allow the student to view lectures via live video, take quizzes and tests, and of course, pay for courses.

To see the coding done to accomplish this, please refer to the Code Appendix.

6.1 Default:

Our home page provides an entry point to the college site. After the title of the page, the most dominant feature is the XML navigation bar. The navigation bar provides links to the Application, Login, Video Conferencing, and Feedback pages. On the navigation bar there is a link to the browser detection page. Users' browsers are detected, and the link to the necessary Netscape plug-in is provided.

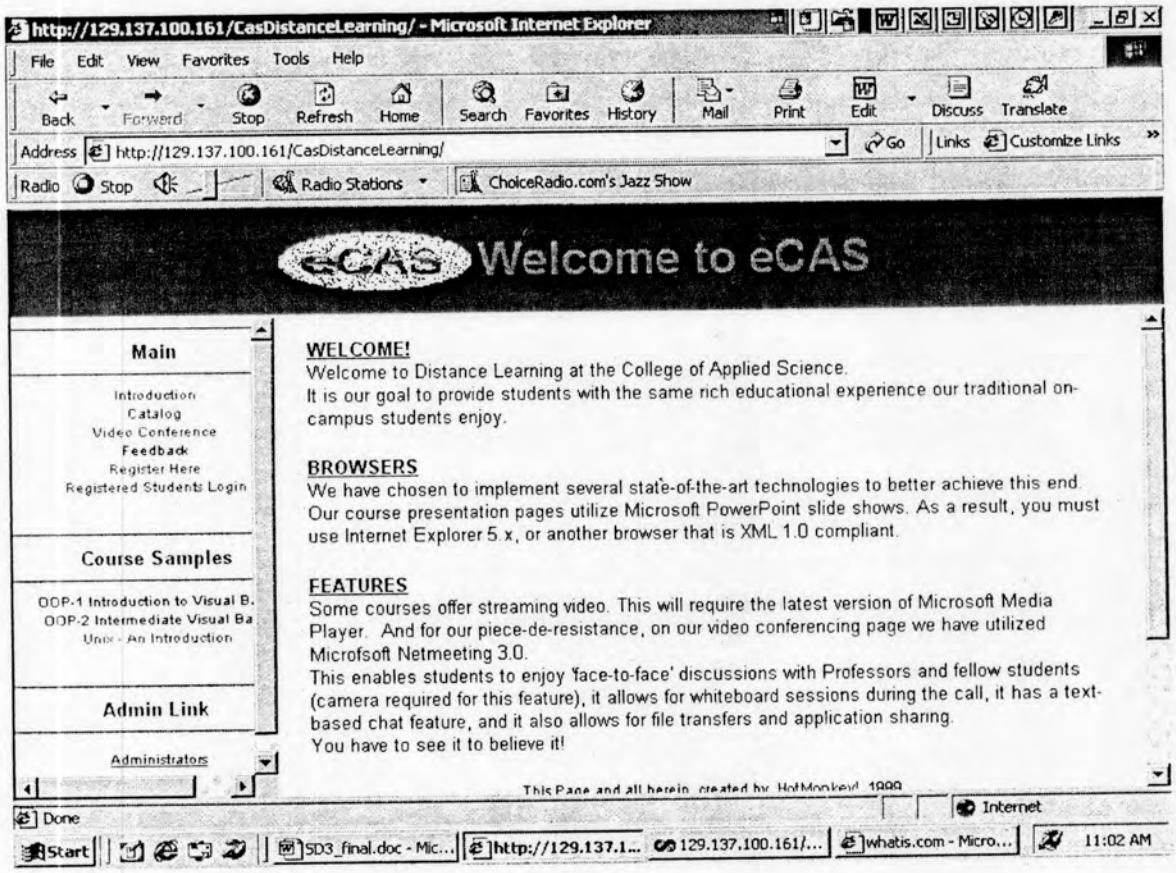


Figure 3. This Shows the Default Page.

6.2 Application:

This section requires two pages. The first is a visible HTML page with fields for Name, Address, E-mail, Password, etc.. The variables are then passed to a ASP page on the web server that handles the writing to the SQL database. Some of the information is checked against the database prior to posting to ensure non-duplication of data. The Application table in the SQL 7.0 database handles eCAS registration data. In the future we hope to integrate this system with the University of Cincinnati system.

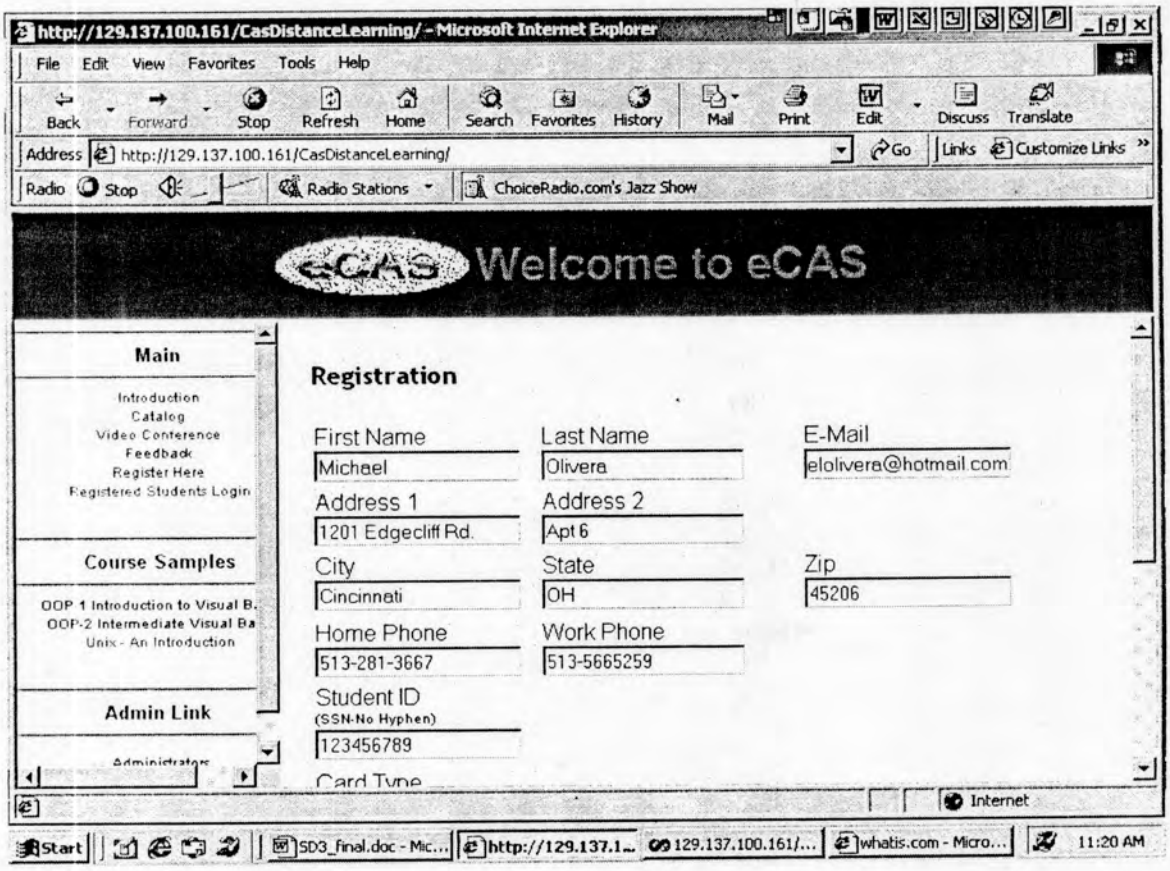


Figure 4. This Shows the Registration Page.

This information is then sent back to the student's browser for verification (next page).

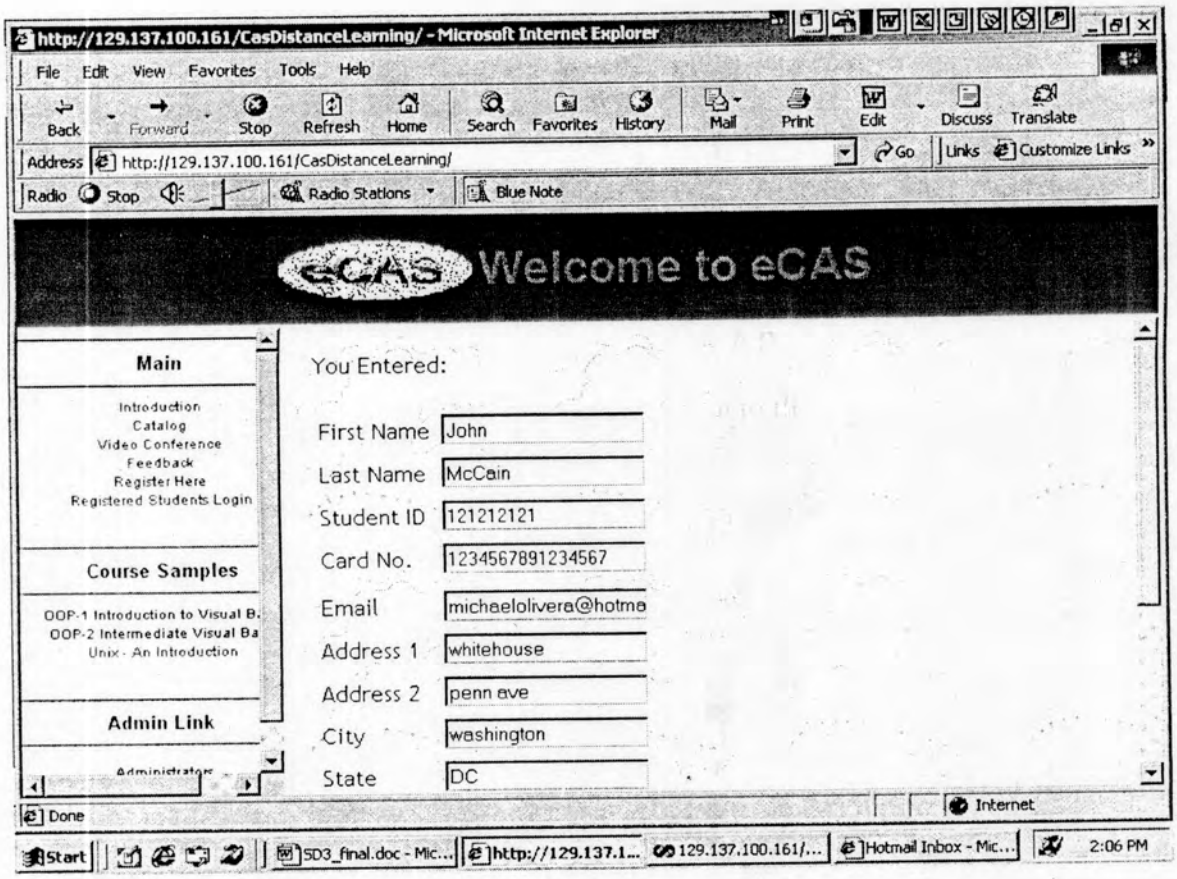


Figure 5. This shows the Registration page, part 2.

After the student clicks "OK" this information is written to the database, and the student is sent to the Login page.

6.3 Login

The login process is a two-step process. The first illustration shows the initial page:

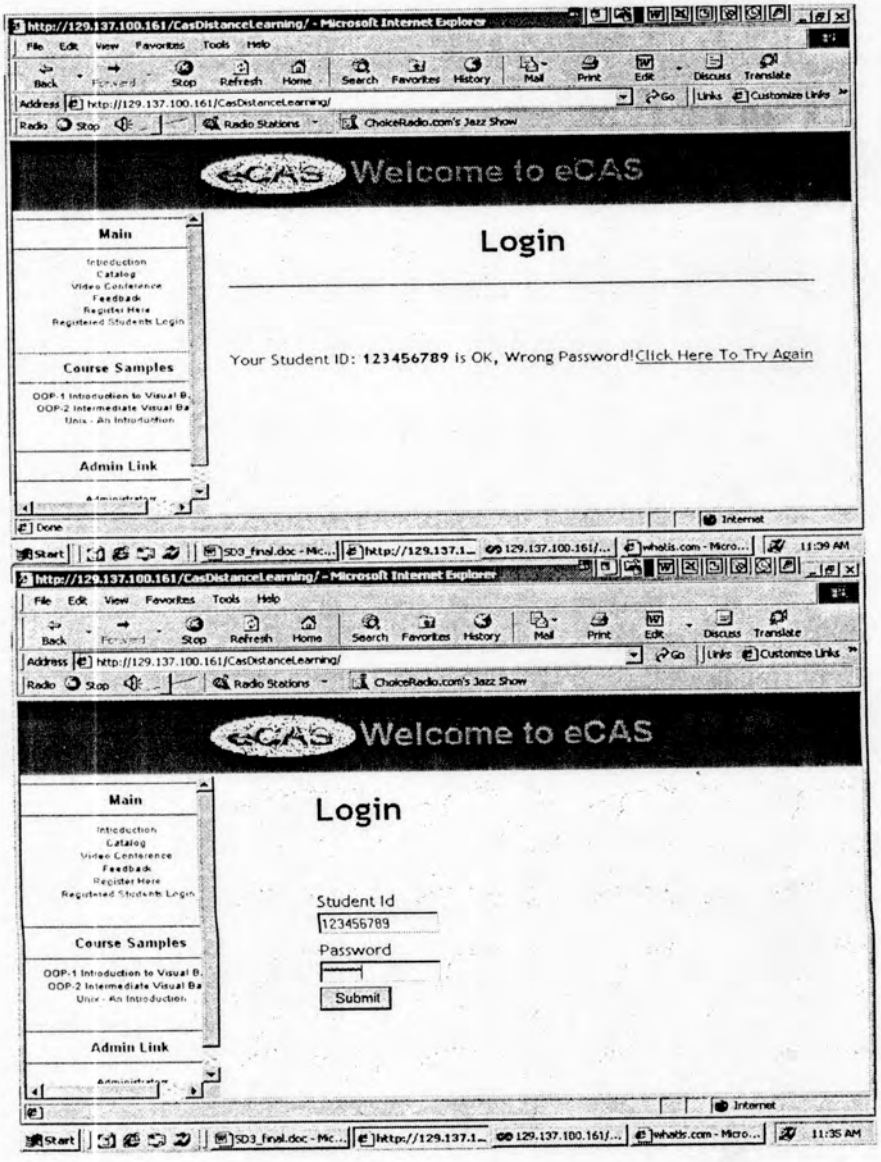


Figure 6. This shows the Login pages.

When a problem is caught the student is directed to try the login again.

Once successful, the student sees the following page:

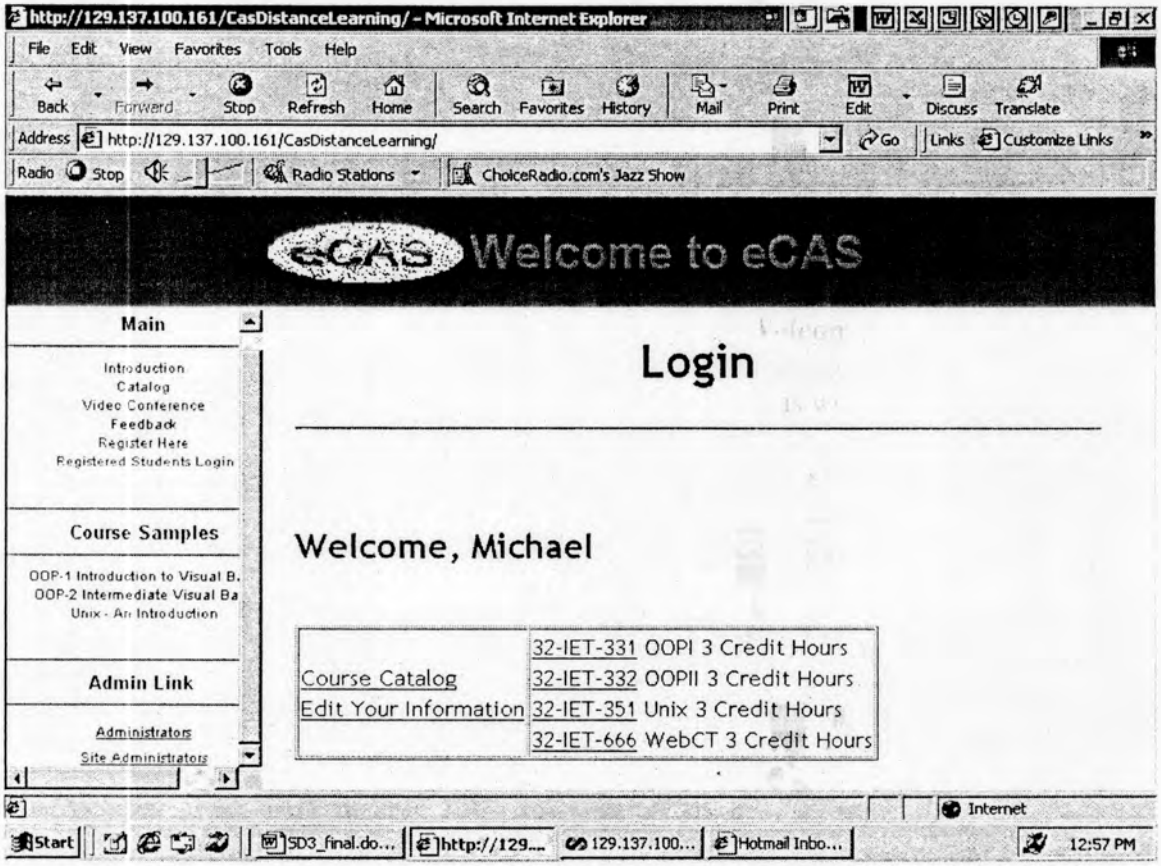


Figure 7. This shows the Login Welcome page

The student sees the courses he or she is enrolled in, as well as a link to edit their personal information incase of an address change. The student should then click on the link for the course they wish to work on.

The following example uses the UNIX course.

6.4 Courses

In the course section the students are able to view courses, get notes and assignments, and of course take tests.

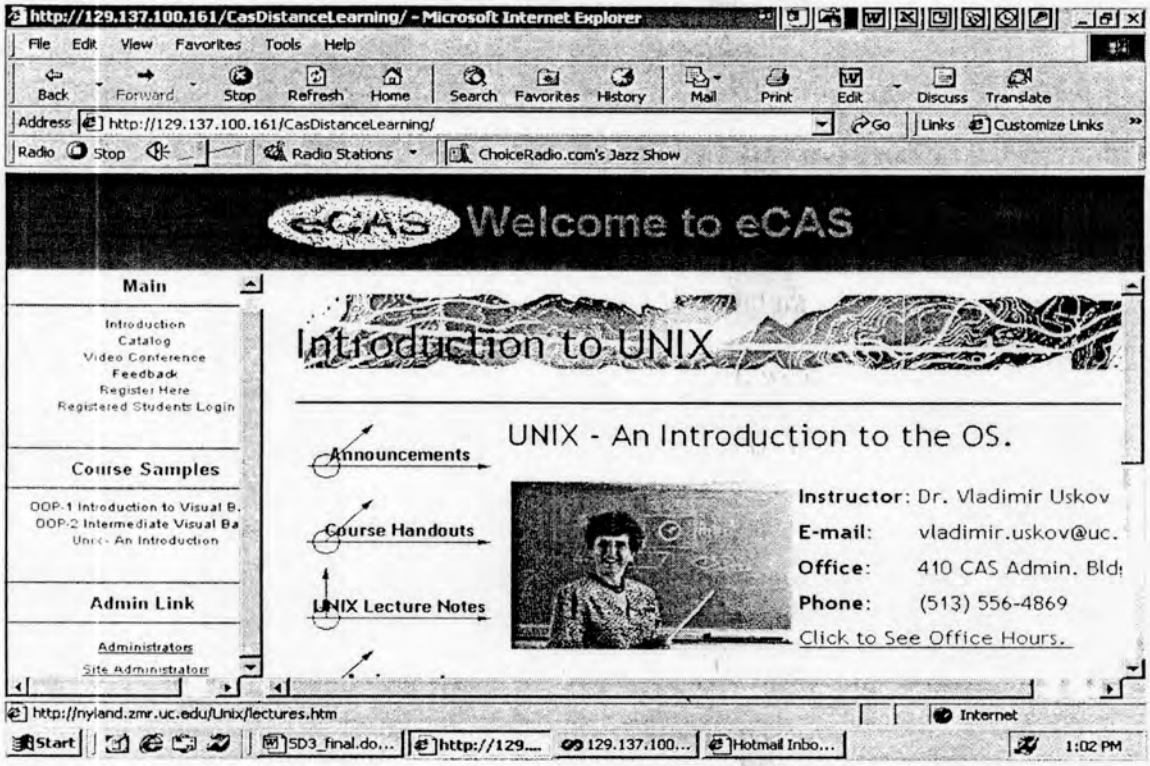


Figure 8. This shows the Course page

FrontPage 2000 is an excellent tool for creating and managing a course Web site. With the ClassWeb template downloaded from Microsoft, we created a complete course Web site. The Quizzing Engine Add-in for FrontPage 2000 provides a simple way for instructors to create tests consisting of multiple choice and true/false questions. These tests are automatically graded. Depending on how the test was configured, students will be able to review their progress and/or retake the test multiple times. Tests can pull questions from a bank of questions. Test questions and answers can contain anything that is supported in HTML. Using FrontPage 2000, professors can create test questions and answers that include various media types.

Sample Course Notes

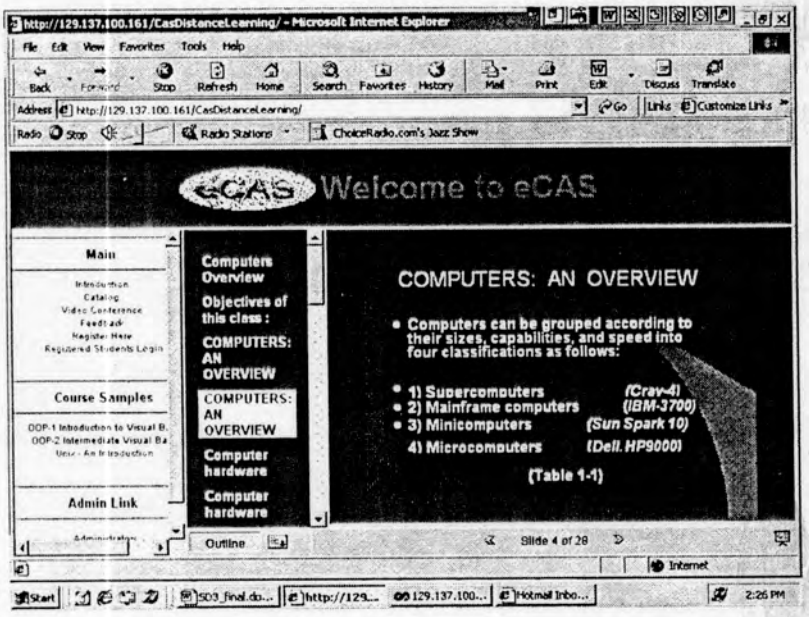


Figure 9. With Navbar Frame

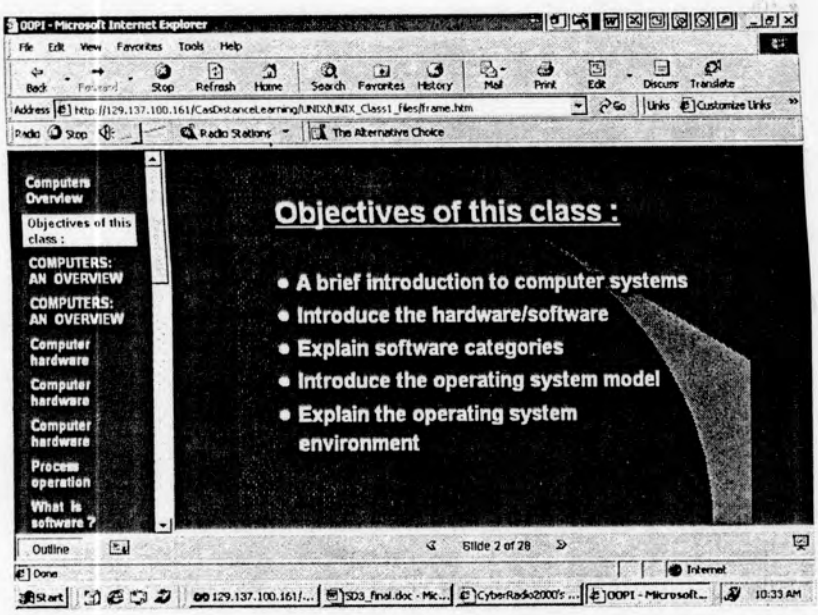


Figure 10. With no Navbar Frame.

6.5 Course Catalog

This page uses a session variable (stuID) to track the user from page to page as is done on other pages. The registered student comes to this page and the system displays the student ID, and a list of available courses. The student enters the course ID into a text box and then clicks Submit to add the class. Once finished adding classes, the student clicks the Done button and is then redirected to the Login page.

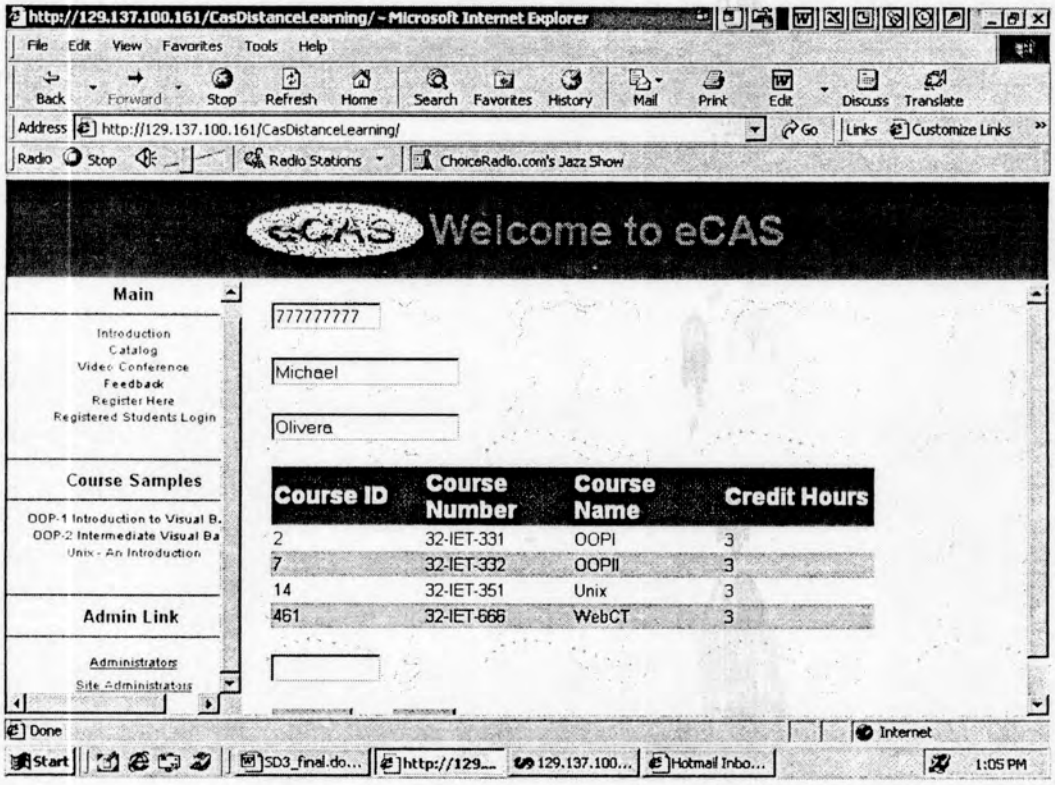


Figure 11. Course Catalog Page

6.6 Feedback:

This is an ASP page tied to the Feedback table in the SQL Server 7.0 database. On this page users are able to input their feedback about the site, along with their e-mail address for further contact. This page was included as an effort towards continually improving the site.

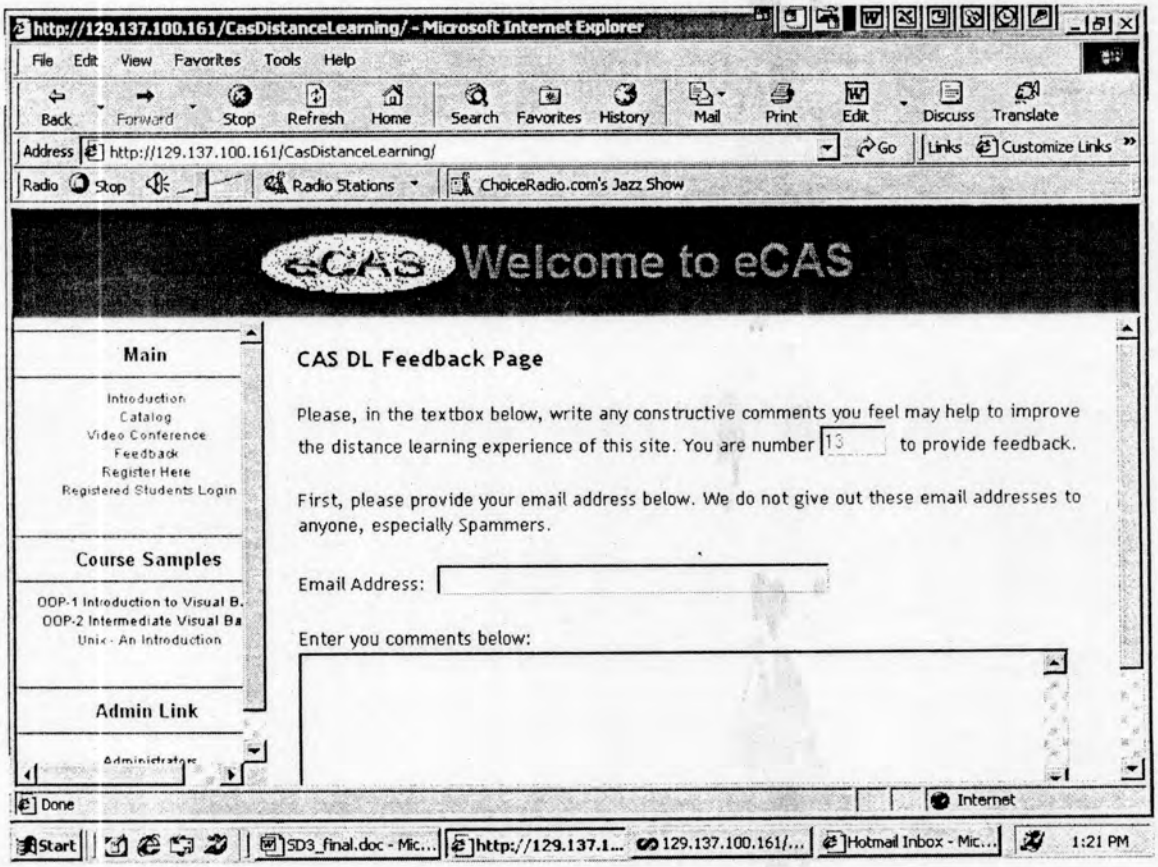


Figure 12. This Shows the Cashier Page

6.7 Cashier:

The functionality is that when students select a course or courses each will be added to a simple a SQL recordset. The students name and ID appear on the Cashier page along with the available courses. The Cashier totals the cart items and the student is then billed via the credit card information included in the electronic Application. The final version of the cashier page will resemble the one pictured below.

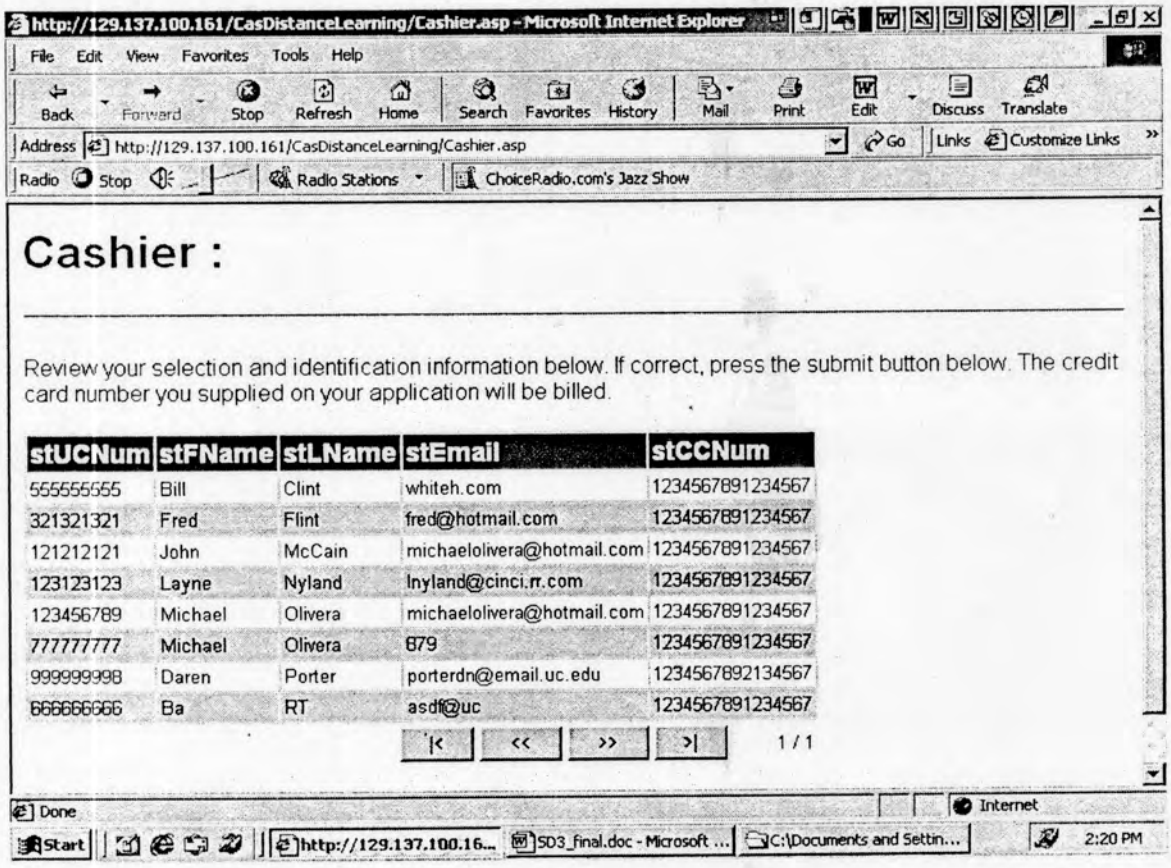


Figure 13. This shows the Cashier Page

6.8 Administration

As explained earlier in this report, this site features administrative functions. The page below is the link page to the various administrative pages. Each link is to an ASP page that allows for moving through and editing the underlying database table. Access to these links will be restricted to those with the proper access authority.

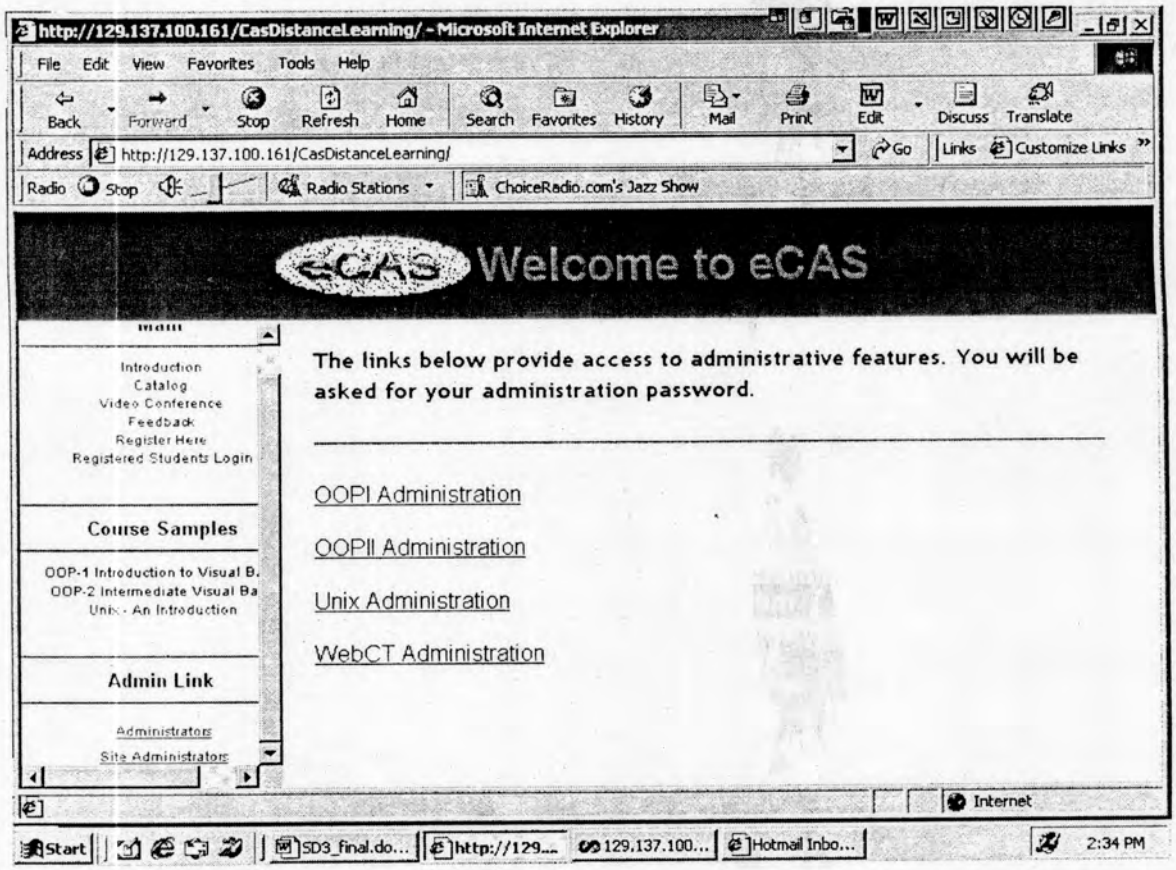


Figure 14. This shows the Site Administration Page

Below is a diagram of how navigation through our site is accomplished using XML technology.

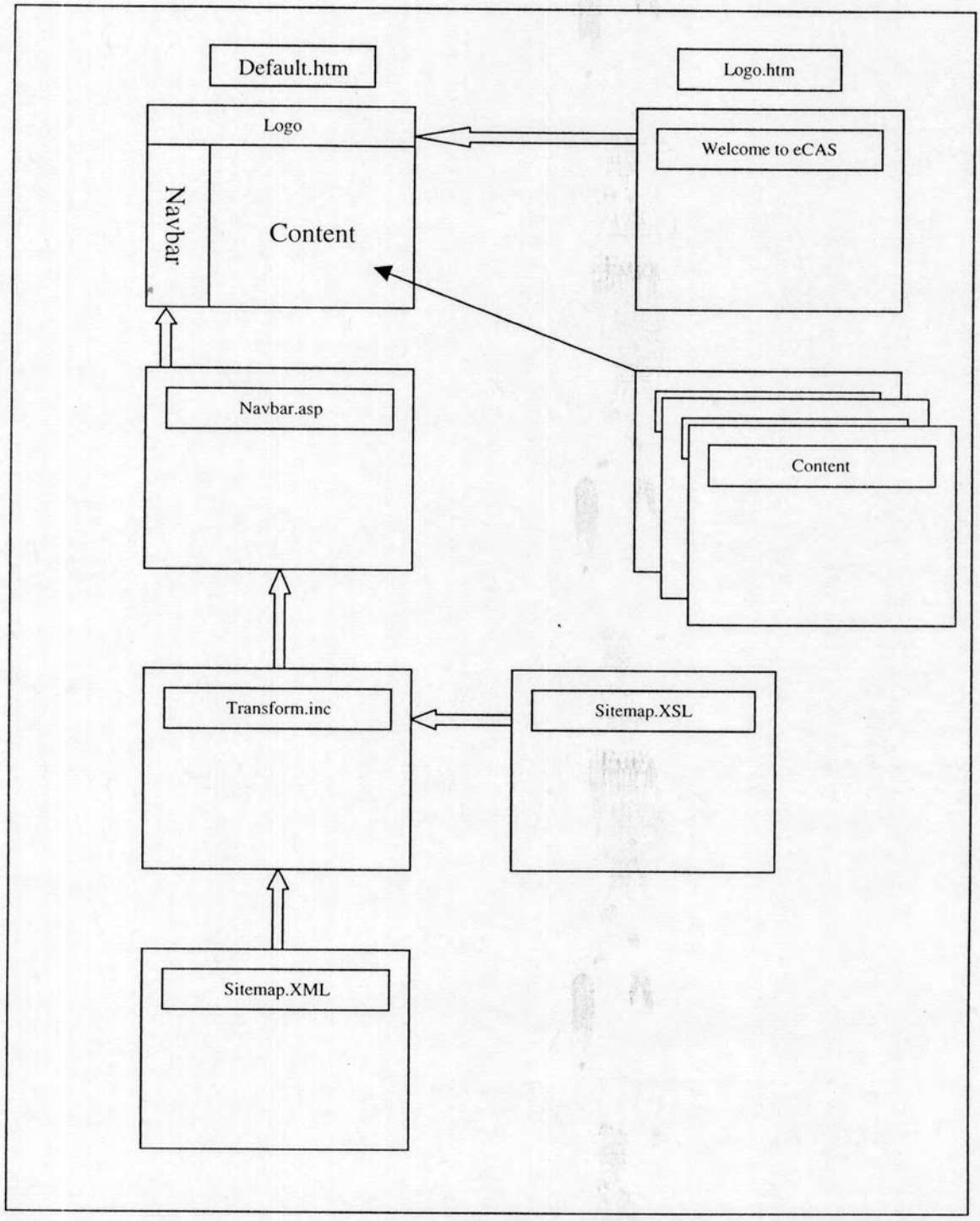


Figure 15. This shows the XML Functionality

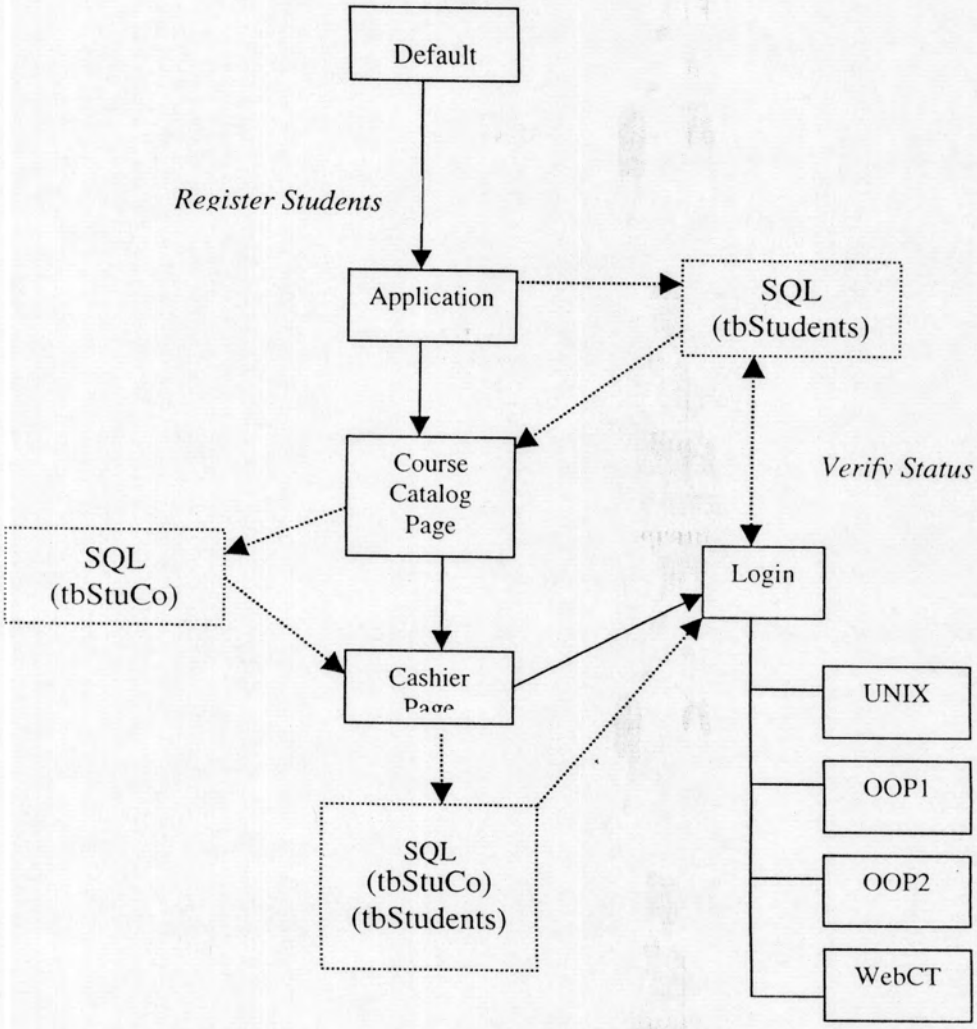


Figure 16. eCAS Web Site Administrative Flow Diagram

7. Conclusions and Recommendations

7.1 Conclusions:

WebCT 2.0 Beta is currently unstable. The 'myWebCT' global ID 'loses' passwords and will not allow Professors to login to their courses. The Course Web/Score! combination add in for FrontPage 2000 has proven to be an excellent tool for our purposes, as evidenced by the fully functioning Web site.

7.2 Recommendation: Use of Secure Sockets Layer

Due to the limitations of the public web server and the cost of acquiring certification for it, SSL technology was not employed on this model. The authors wish to strongly recommend the use of SSL technology on the final production version of this site. HTTP (Hyper Text Transfer Protocol) is unsecured and HTTPS is secured. HTTPS (Secure Hypertext Transfer Protocol) is a Web protocol developed by Netscape that encrypts and decrypts user page requests as well as the pages that are returned by the Web server. HTTPS is really just the use of Netscape's Secure Socket Layer (SSL) as a sublayer under its regular HTTP application layer. (HTTPS uses port 443 instead of HTTP port 80 in its interactions with the lower layer, TCP/IP.) SSL uses a 40-bit key size for the RC4 stream encryption algorithm, which is considered an adequate degree of encryption for commercial exchange. SSL manages the security of message transmissions in a network. Netscape's idea is that the programming for keeping your messages confidential ought to be contained in a program layer between an application (such as your Web browser or HTTP) and the Internet's TCP/IP layers. The "sockets" part of the term refers to the sockets method of passing data back and forth between a client and a server program in a network or between program layers in the same computer.

Netscape's SSL uses the public-and-private key encryption system from RSA, which also includes the use of a digital certificate. To set up SSL on IIS do this:

- Generate a key pair file and a request file.
- Contact VeriSign and request a certificate for your security keys.
- Install the certificate you receive.
- Start the MSMC and activate SSL for user authentication.
- Activate SSL for directories that require it.

7.3 Recommendation: Faculty Technology Center

The mission of the proposed Faculty Technology Center will be to support faculty and other educational organizations at the College of Applied Science with up to date, dependable, and operational electronic media, alert to the opportunity to improve both our equipment and service. Our goal is to provide learners with a high quality, web-based, distance-learning experience

Hardware Infrastructure

Earlier approaches to distributed education constrained delivery in significant ways. Extension operations required the co-presence of teacher and student as well as a classroom in a remote location where students could assemble. Televised instruction eliminated the need for co-presence, but introduced problems of asynchrony. Students in televised courses were often unable to ask questions of teachers during the course of a lesson, if at all. Except in expensive, specially equipped facilities that enabled live, two-way broadcasts, distance learning via television was largely a passive activity.

The dramatic emergence of the Internet and, more recently, the World Wide Web, have suddenly created a way of transcending many of the problems associated with

earlier forms of distance education. Web technologies are capable of delivering text, data, images, audio, and video in an integrated and coordinated manner. Moreover, "chat" rooms that allow nearly synchronous e-mail exchanges, electronic whiteboards that allow people to co-produce drawings at a distance, and fledgling imaging technologies that bring two-way, live video to a computer screen (such as CUseeme and Microsoft Corporation's Netmeeting) open the possibility of real-time exchanges, not only between students and teachers who are separated in space but among students who are themselves distributed. Finally, because computers are prevalent in the workplace, computer-based training (CBT) and Web-based training (WBT) can provide multiple users with access to training materials and to each other without anyone leaving their desks.

Unfortunately, the rapid advance of technology has made the standardization of hardware nearly impossible. Resources covering distance-learning applications talk only in the broadest possible terms. The few resources that discuss the actual hardware are typically two years old. In Internet years this is roughly age 17, and obviously obsolete. However, there are hardware components that are common to all web-based learning applications (see Appendix D for our current specifications). These include:

- Internet Connection
- Firewall/Proxy Server
- Web Server
- File Server
- Database Server
- Instructor Workstations

When these recommendations are implemented, the site will be ready to go into production.

Appendix A : Microsoft Netmeeting Manual

1 Conferencing in Distance Education

The need to include conferencing capabilities in the eCAS Distance Education web sight is apparent. Only through conferencing, particularly audio-video enabled, can the successful learning environment of the classroom be closely simulated over a distance. Microsoft's Netmeeting application has the superior video-conferencing interface and application development kit at this time. Both the Netmeeting application and the Netmeeting Software Development Kit are absolutely free.

2 Netmeeting Overview

Netmeeting is a versatile, extremely powerful tool, which includes functionality for audio-video conferencing, chat, whiteboard, and application/resource sharing. Newer Microsoft Operating Systems such as Windows 98 and Windows 2000 include Netmeeting. For older Operating Systems, Windows 95 or Window NT 3.0 and above, Netmeeting is freely distributed. The necessary software is available for download at www.microsoft.com/windows/netmeeting. The Netmeeting SDK complete with examples and how-to's for embedding the Netmeeting ActiveX object in an application is also available and free to download at the same site.

A video conferencing session is easily transformed into a virtual classroom. Instructors or fellow student can provide step by step instructions, answer questions, give troubleshooting assistance, or even take over a student's desktop and show them first hand how to do something, provided they have the express permission of the student. Netmeeting also includes for file sharing functionality.

3 Conferencing

Once you are ready with a Windows operating system and a relatively new version of Internet Explorer direct your browser to the eCAS homepage, currently located at http://129.137.100.161/casdistancelearning. Click the *video conferencing* link located on the left navigation bar of the page. The Video Conferencing page is shown below in Fig. 3.1. This page is designed to help students contact the instructors of their distance classes. If the page you see does not resemble fig. 3.1 then consult section 8.0, Troubleshooting. If the current user has not used Netmeeting before the first time user wizard execute and Netmeeting will require a few simple details such as your email address.

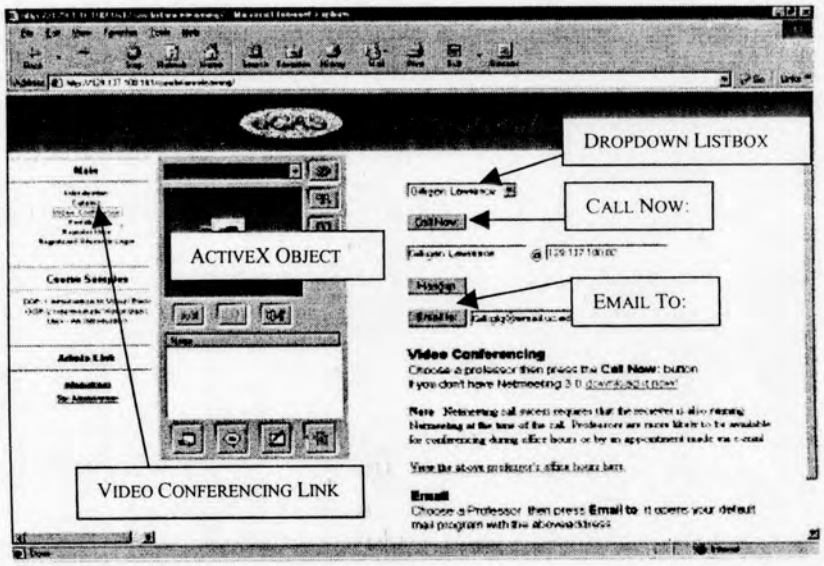


Fig. 3.1

Notice The top Dropdown Listbox, which currently begins with the default value 'Gilligan Lawrence'. Use this box to select the professor or consultant you wish to contact. The page will query our database and populate the other text boxes on the page with the proper information. Obviously not all instructors/consultants are available all the time. We recommend that students set up conferences through preliminary correspondence.

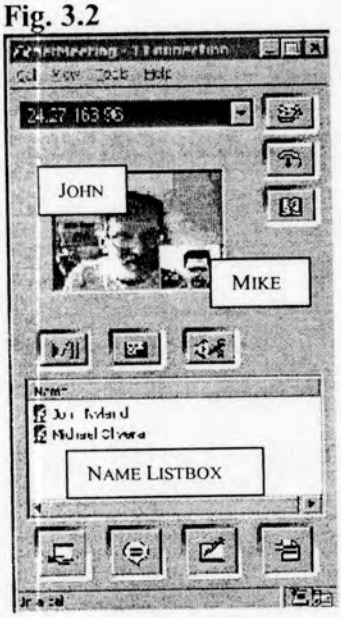
Clicking the *Email to*: button will open your default mail program.

Browser compatible mail programs will also open a new message directed to the consultant/instructor selected. This is a convenient way to contact instructors and set up conferences.

Ok you have an appointment, then click the Call Now: button. Our Netmeeting object dials the internet protocol address located in the textbox to the right of the @ symbol on the page. Alternatively, advanced users may wish to make the call using the Netmeeting interface directly. To do so, click the small phone icon shown in the upper right hand corner of the Netmeeting ActiveX control shown in figure 3.2. Place the IP you wish to dial in the proper listbox on the corresponding pop-up form and press ok. The person on the receiving end of the call, assuming they have Netmeeting, will see a pop-up dialog box on their screen with the origin of the call displayed and an accept or decline button. With any luck the receiver will accept your call.

Fig. 3.2 shows a videoconference in session. What more do you need to know in order to hold a conference? Nothing really. If video/audio hardware is available then real time face-to-face interaction begins immediately. Netmeeting will also work fine in the absence of either video or audio/recording equipment by eliminating unavailable channels. Video play/pause, local video echo and audio recording levels can be adjusted using the three buttons below the video shown in fig. 3.2. In each case the procedure is self evident once initiated.

Fig. 3.2



4 Chat

The Chat feature of Netmeeting serves primarily as a back up channel when audio communication is inappropriate or unavailable. Notice the four buttons below the name listbox shown if Fig. 3.2. The second button from the left is used to initiate a chat session. When pressed a Chat dialogue box, analogous to the one shown in Fig. 4 below, is opened. When in a chat session type the message you wish to send in the *Message*: listbox. Then select the persons you would like to see the message using the *Send To:* drop down list box. Finally, press the send button. The send button is located to the right of the message list box. It has a picture of a white dialogue bubble, with blue sample text, on it. The output list box, complete with scrollbar, displays messages and the name of the person responsible for them as the chat session proceeds.

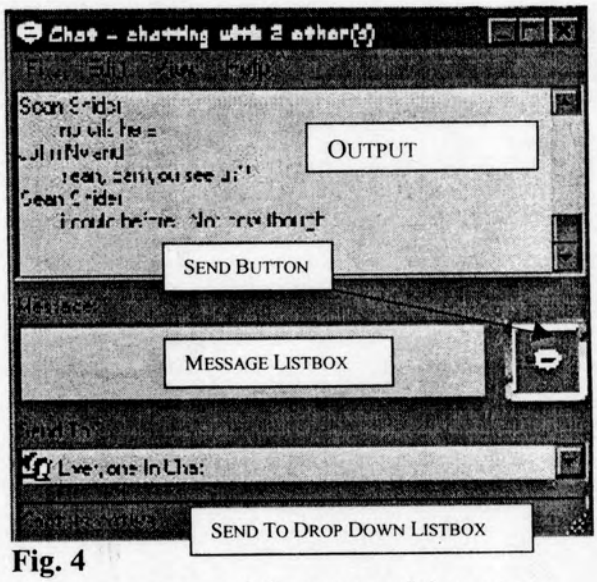


Fig. 4

5 Whiteboard

The process of using Netmeeting's Whiteboard is very similar to using it's chat features. Next to the button that initiates a chat session is the one that initiates a

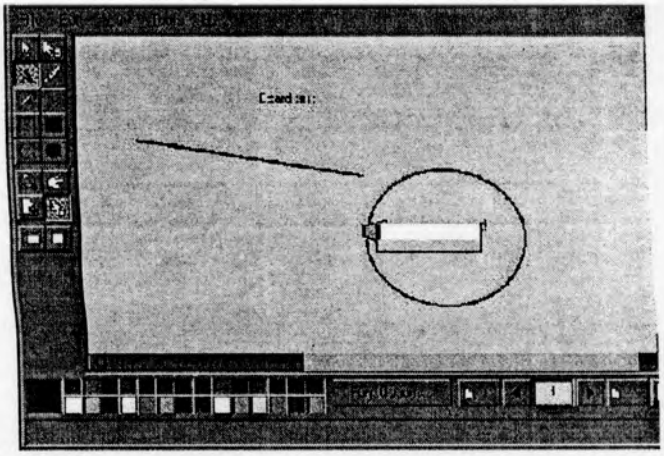


Fig 5

Whiteboard session. In Fig. 3.2 this button is found below the name listbox and is the third button from the left. Clicking this button executes the embedded whiteboard application shown in Fig. 4. Once activated the whiteboard is a hands on program reminiscent of early Microsoft Paint programs. Here session participants can share real time drawings in order to better express the idea's they would like to communicate.

6 File Transfer

Netmeeting has a built in file transfer protocol interface that allows conference participants to send and receive files amongst each other. The transfer files button is located below the name listbox shown if Fig. 3.2 and to the far right. The icon resembles a text-covered piece of stationary with its top right hand corner dog-eared. Clicking this button will bring up the File Transfer dialogue box shown if Fig. 6.1. Let your mouse hover over he button below the *file* menu options. The tool tip 'add files' should appear. Click the add files button. The Select Files to Send dialogue box shown if Fig. 6.2 will open.

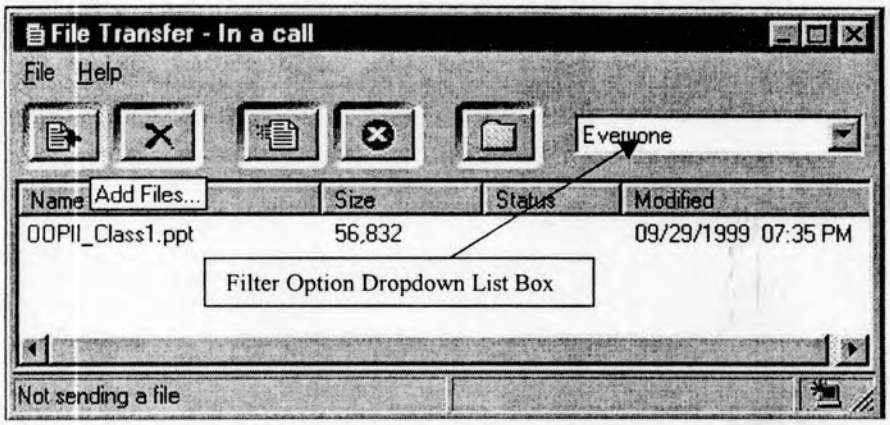


Fig. 6.1

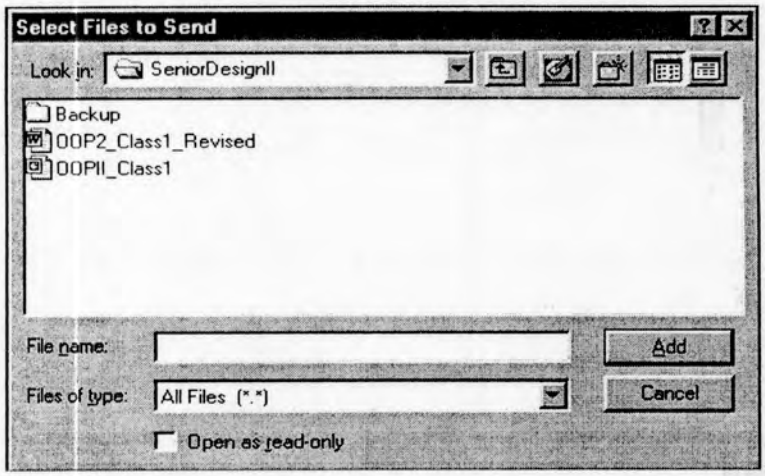


Fig. 6.2

Windows users should be familiar with the file list box interface. Browse to the files you wish to send, select them and press the add button. The second button is the remove files button. Select a file from the list that you no longer wish to send and press this button to remove it. The third button is the send files button. Click this button and the files you have selected will be sent. The last button has a folder icon. Click it to view files that have been sent to you during the current Netmeeting session. The dropdown listbox to the right of this button is a filtering option that becomes useful when you do not want everyone in the current conference to receive the files you are going to send. When this is the case be sure the person you do want to receive the files is selected in this box before you press the send button.

7 **Application Sharing**

The Application Sharing button is located to the left of the Chat button in Fig. 3.2. It is the first of the buttons below the name listbox. Click this button to bring up the dialogue box shown in Fig. 7 to the right. Select desktop in the listbox as shown. Then click the *Share* button. The person you are conferencing with will find that a dialogue box appears on their screen asking if they will allow you to take control of their desktop. Once you have control of another person's desktop a window will open showing the remote desktop as it appears at the time. The maximum resolution of this window is 800 by 600 so desktops with a higher resolution setting will not be able to maximize this window and have it take up their whole screens. Controlling the remote desktop is exactly the same as sitting in front of the computer with the exception of special keystrokes such as control + alt + delete. These exceptions are handled with the control menu located at the top of the shared application window. Click the drop down menu. Select the desired keystroke from the menu and send it.

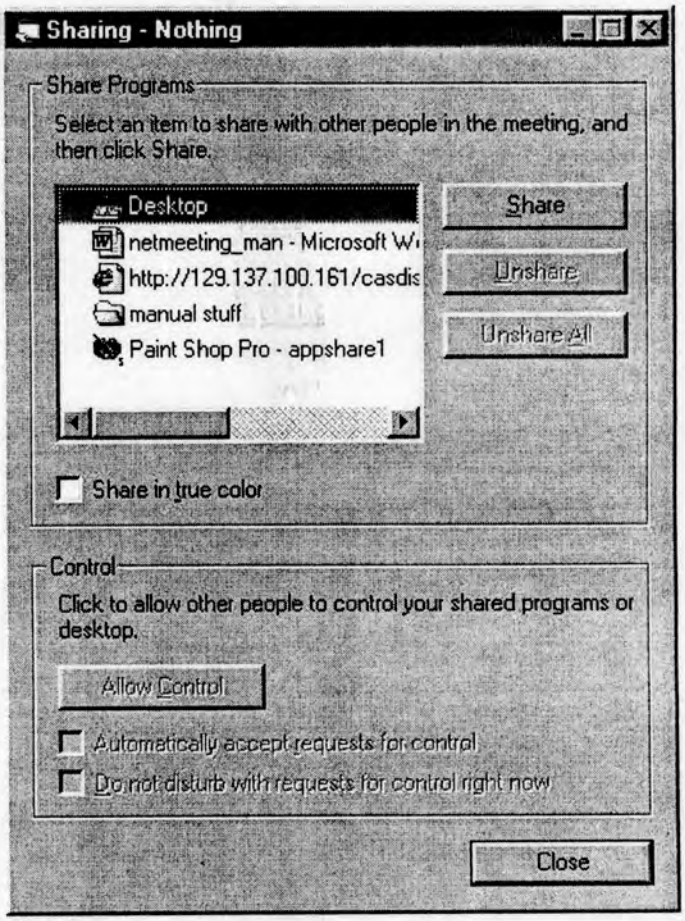


Fig. 7

8 Troubleshooting

The eCAS Distance Learning web site, like the Information Engineering Education program it is designed to mirror, is heavily tied the Microsoft operating systems and the Internet Explorer browser that it was developed within. Also, the inclusion of ASP code requires that Internet Explorer be set to pull a page from the server each time it is refreshed rather than displaying a cached version. To be certain that your browser is configured properly for ASP pages open Internet Explorer now. Click the *tools* menu and select *Internet Options.* On the Internet Options dialogue box is a *Temporary Internet Files* frame and within that frame is a *Settings*... button. Click it. Select the *Every visit to the page* option and click the *OK* buttons. Other hardware/software errors and conflicts are beyond the scope of this manual. Consult your system administrator.

Appendix B: eCAS Website Manual

Purpose of Site

eCAS is a web site. The eCAS site is designed to provide a complete web-based distance learning solution for the College of Applied Science. In the current edition, a student may fill out an application, register, and pay for classes. They may also interact with fellow students, faculty, and the web site administrator.

Requirements

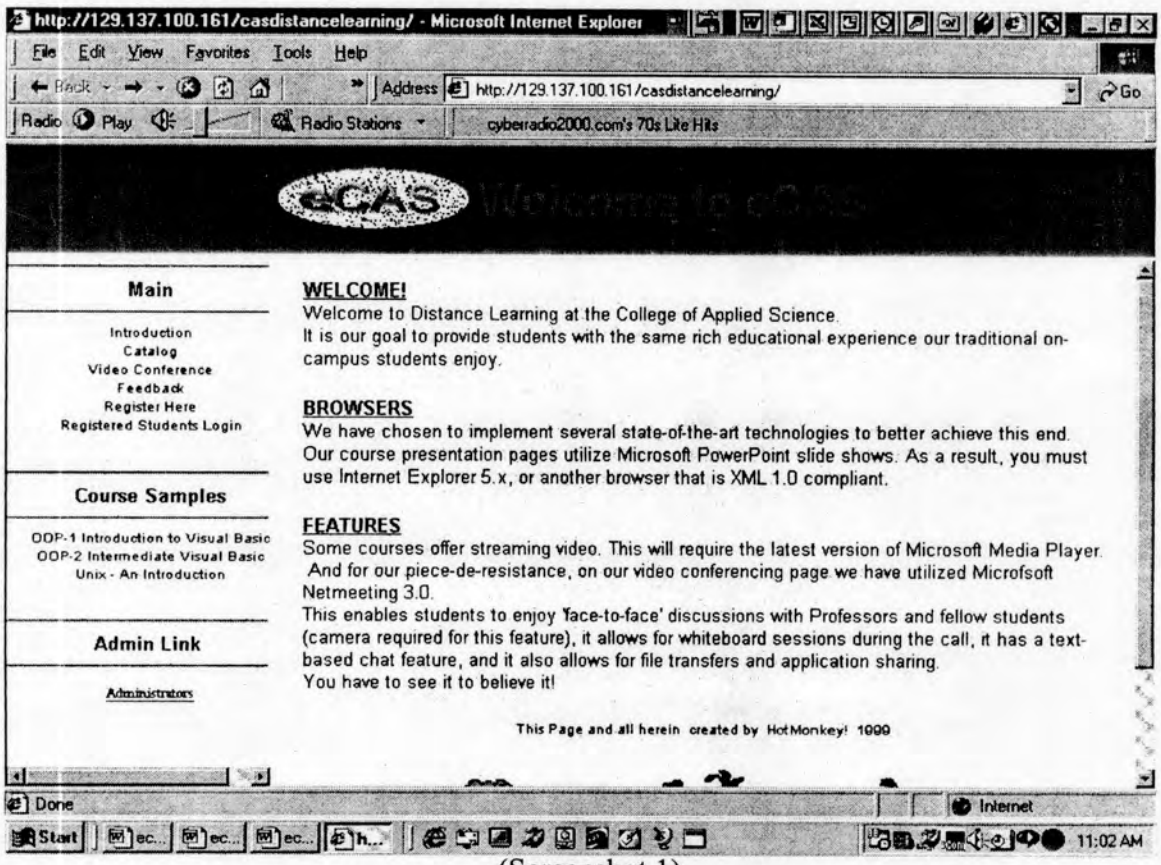
Distance learning at eCAS takes advantage of the Internet and computer conferencing along with other multimedia technologies to deliver the classroom to your computer. In order to participate, students must have full access to the Internet and a computer with multimedia capabilities (see Appendix D for our current specifications). System requirements include:

- Windows 98, NT 4.0, or 2000
- Internet Explorer 5.x (or better)
- Pentium II processor (or better)
- Full-duplex sound card
- Microphone
- Speakers
- Camera (for video conferencing)
- A valid email address

Site Overview

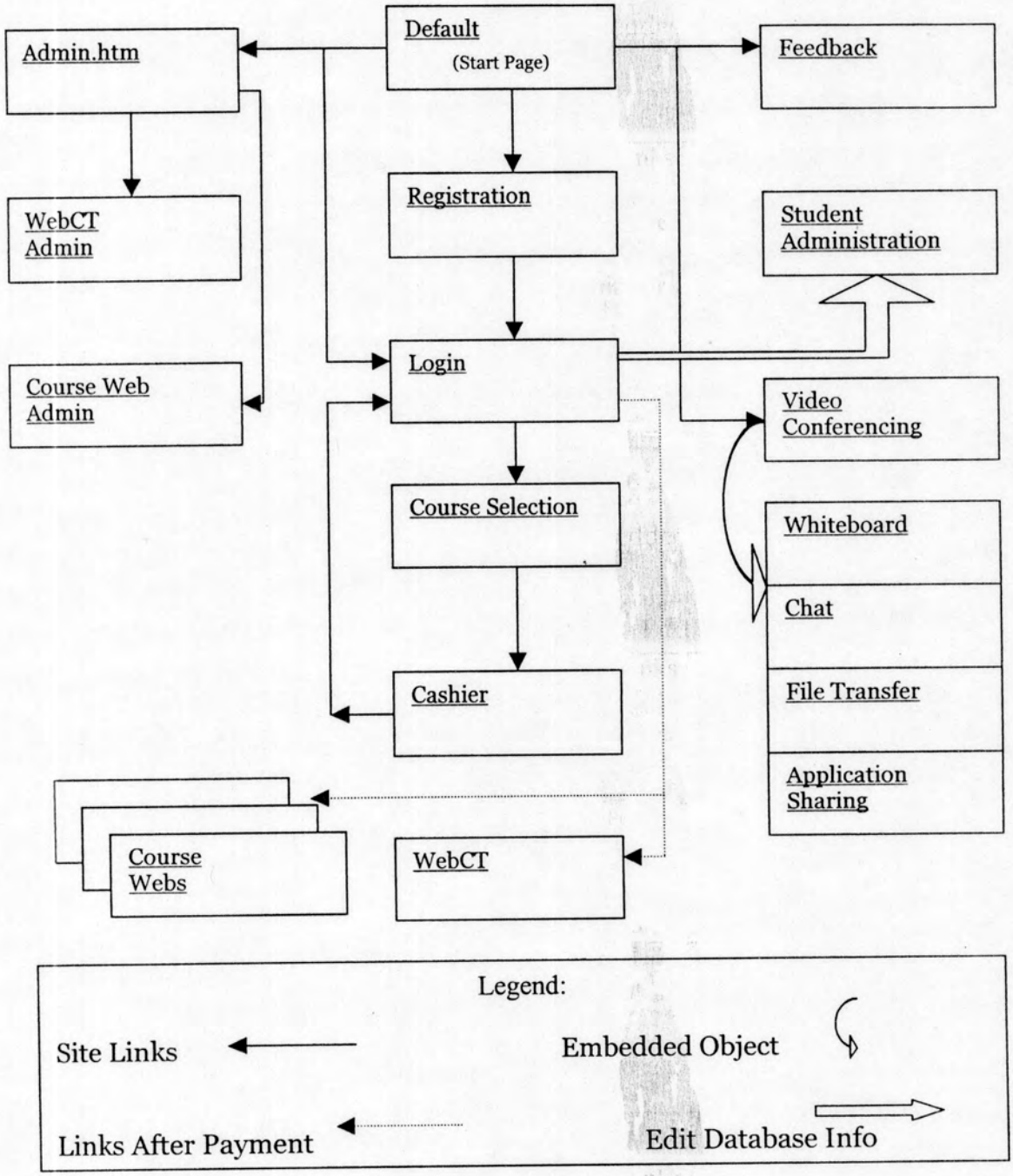
eCAS was built with the current editions of various Microsoft products. These include Visual Interdev, SQL Server and FrontPage 2000. As a result, Internet Explorer 5.0 or newer is required to view this site.

The Default page (Screenshot 1) shown below is the first page visible to the student. From here, site navigation is controlled by the links on the left hand side of the screen. A student can choose to register, login, use Netmeeting, and provide feedback to the site administrator. The site diagram on the following page provides an overview of the rest of the structure.



(Screenshot 1)

eCAS – 2000
Site Diagram



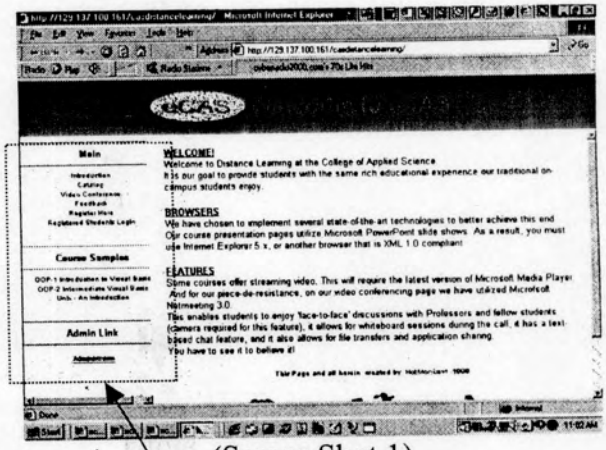
Navigating the Pages

The student begins at the Default screen. From here the first time visitor/student will begin the registration process. After registration and course selection, the student should carefully read each syllabus, and immediately e-mail the course Professor about any questions or concerns.

Default

The navigation bar currently contains three divisions. *Main* will always be present.

SELECT → Registration



(Screen Shot 1)

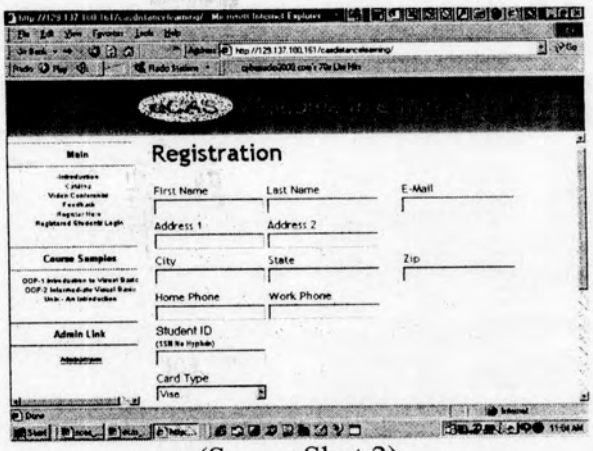
Navigation

Note:

The other components of the navigation structure are designed to change and evolve over the life of the site. For example, *Course Samples* may be replaced with links relevant to the various academic departments represented on this site. The *Admin Link* will not be accessible after March 9, 2000.

Registration

Fill in each box (Address 2 and Work Phone are optional). Your student ID is usually your Social Security Number. Scroll to the bottom of the screen and choose **OK** when finished, **RESET** to start over.



(Screen Shot 2)

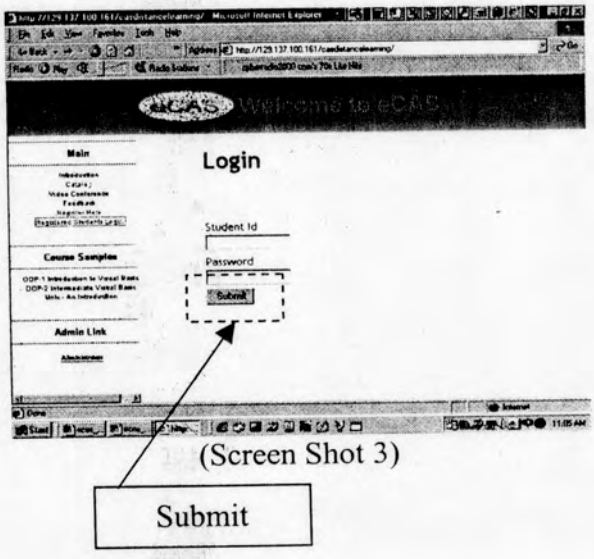
Registration Errors

Address 2 and Work Phone are the only optional fields. The system will not allow you to continue until all others fields are complete. In addition, a simple validation routine is performed on the credit card information prior to submission.

Login

Fill in your ID and Password.

Press **Submit.**



(Screen Shot 3)

Login Errors

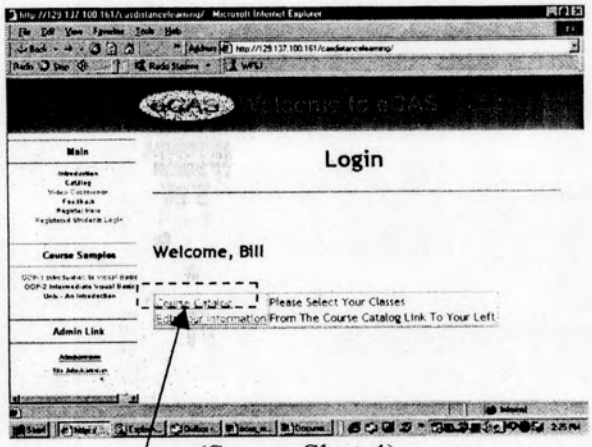
There are only two errors possible on this screen. You will be asked to correct either your user ID or password. If you have forgotten your password or if the error persists DO NOT repeat the application process. Email the webmaster, John Nyland, at jnyland@hotmail.com with a description of your problem, and it will be resolved.

Login

First time visitors choose

Course Catalog

Others may choose to edit their student information.



(Screen Shot 4)

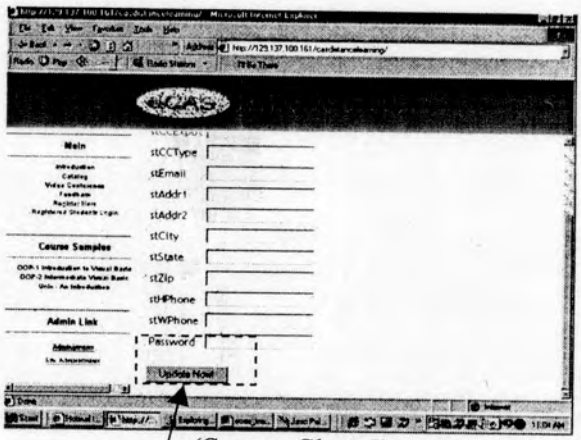
Note:

Your student information is sensitive and important information. You should review it often to check for completeness and accuracy. Students are responsible for all information submitted.

Student Administration

Select the field you wish to change.

Press **Update Now** when finished.



(Screen Shot 5)

Update Now

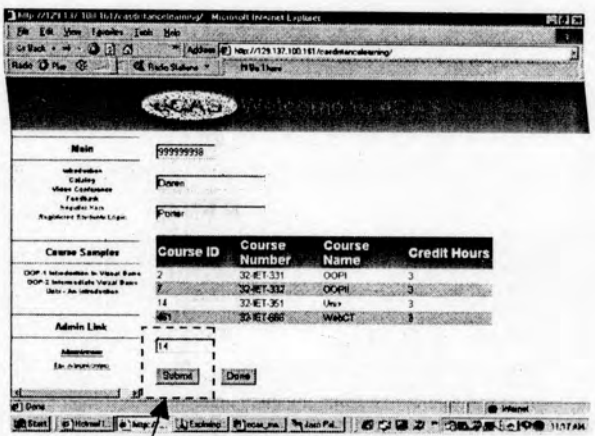
Course Selection

There are currently two different varieties of course presentation software. By selecting OOP I, OOPII, or UNIX, the student will be utilizing courses developed in Microsoft FrontPage 2000. WebCT is a stand-alone application. If the student chooses WebCT, the student must email the webmaster to receive their WebCT password.

Select Your Courses

Enter the course ID number and press **Submit** for each course you wish to take.

Press **Done** when finished.



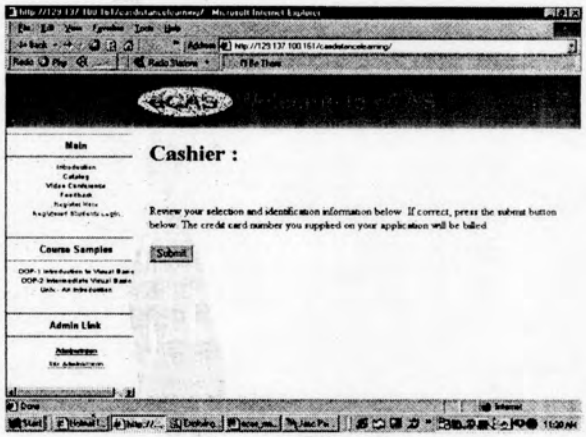
(Screen Shot 6)

Press Submit for each course you wish to take

Cashier

You are now taken to the Cashier Page. Here you will be able to review your selections and their costs. You may choose to Submit your application, Cancel your application, or Change your selections.

Press **Submit** when finished. You are now returned to the Login screen. Enter your ID and password to view the links to the course you selected.



(Screen Shot 7)

After completing the login process, you will see a screen like this. Choose the course you wish to view.

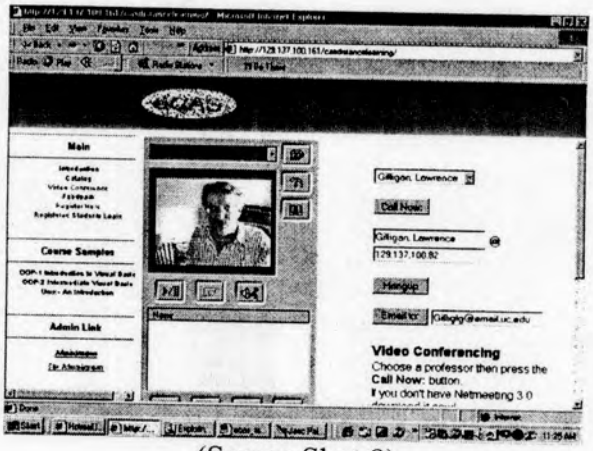
Prior to viewing your course, you should familiarize yourself with the Video Conferencing page.

(Screen Shot 8)

Netmeeting Page

Students and faculty may use this page to hold a videoconference, chat, transfer files, and share a whiteboard or applications.

Choose your Professor and press **Call To**.



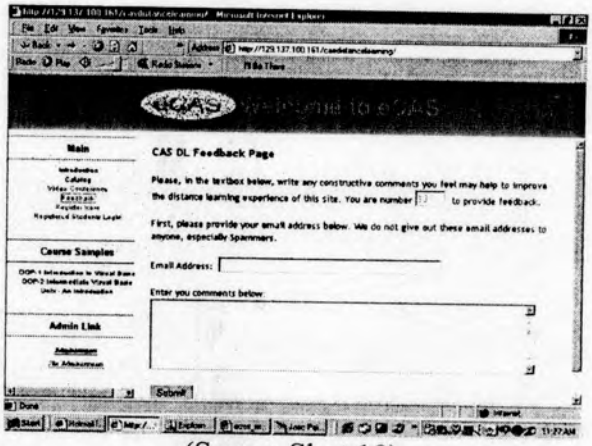
(Screen Shot 9)

Note:

NetMeeting is Microsoft's solution for multi-party voice, video, and application sharing over intranets and the Internet. It is a robust, all-purpose conferencing application. The student should refer to the Netmeeting manual for complete instructions in the installation, configuration, and use of this tool.

Feedback

Students should take a moment to share their thoughts and feelings with the web master. This site is designed to be flexible, and to support ever-changing course requirements and teaching styles.

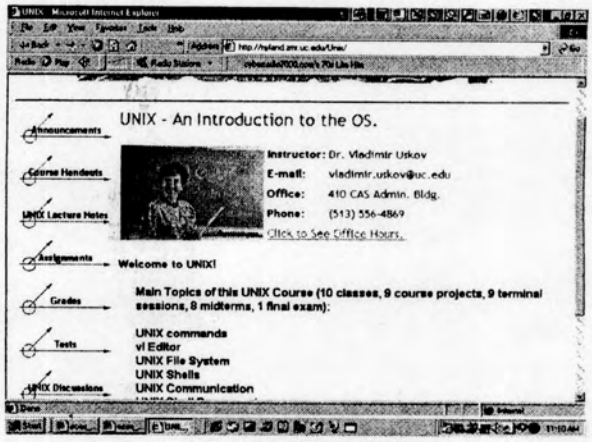


(Screen Shot 10)

Taking a Course

Professors all have their own ideas about the best way to present their particular course materials. However, every course will have several common features. There will be links provided to download syllabi, handouts, lecture notes, and assignments. Some courses will offer streaming video presentations. Some may offer PowerPoint slides with and without attending video lectures, and some courses may have threaded discussion pages.

This is a sample course with links to various course materials.



(Screen Shot 11)

This is the WebCT Welcome page. Please see the WebCT documentation for further instructions.

(Screen Shot 12)

Troubleshooting

Course Problems
E-mail your Professor a description of your problem or concern.

Registration
E-mail the web master at jnyland@hotmail.com

Video Conferencing Questions
Review the Netmeeting documentation.
If you don't see the Netmeeting interface on the video conferencing page, download and install a copy from www.microsoft.com/netmeeting

Appendix C: Code for Main Pages

Login Page (login.asp)

```
<%@ Language=VBScript%>
<SCRIPT id=DebugDirectives runat=server language=javascript>
// Set these to true to enable debugging or tracing
@set @debug=false
@set @trace=false
</SCRIPT>
<% ' VI 6.0 Scripting Object Model Enabled %>
<!--#include file="_ScriptLibrary/pm.asp"-->
<% if StartPageProcessing() Then Response.End() %>
<FORM name=thisForm METHOD=post>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<basefont color=black face="Trebuchet MS">
<html>
<head>
<!-- add some include files --><!--#include file="adovbs.inc"--><!--#include
file="adostuff_JFN.inc"-->
<SCRIPT LANGUAGE=vbscript RUNAT="server">
dim rst
dim strStuid
dim strPW
dim strSQL
dim strSQLrq
dim objConn
dim strSQLDSNLess
dim strFName
dim strCoNum
dim strCoName
dim strCoCred
'stuff vars to find the student info
strStuid = request.form("txtStuid")
strPW= request.form("txtPW")
strSQL= "SELECT * FROM tbStudents WHERE (stUCNum = " & "'" & strStuid & "')"
strSQLrq= "SELECT tbStudents.stUCNum, tbStudents.stFName, tbStudents.stLName,
tbStudents.stCCNum, tbStudents.stCCType, tbStudents.stCCExpDt, tbStudents.stEmail,
tbStudents.stAddr1, tbStudents.stAddr2, tbStudents.stCity, tbStudents.stState,
tbStudents.stZip, tbStudents.stHPhone, tbStudents.stWPhone, tbStudents.Password,
tbStuCou.stUCNum AS Expr1, tbStuCou.coID, tbStuCou.scGrade, tbCourses.coID AS
Expr2, tbCourses.coNum, tbCourses.coName, tbCourses.coCredHour, tbCourses.coCost
FROM tbStudents INNER JOIN tbStuCou ON tbStudents.stUCNum =
tbStuCou.stUCNum INNER JOIN tbCourses ON tbStuCou.coID = tbCourses.coID
WHERE (tbStudents.stUCNum = " & "'" & strStuid & "')"
'create connection
```

```
objConn.ConnectionString=strSQLDSNLess
objConn.CursorLocation=adUseClient
objConn.Open


'password check
Set rst=objConn.execute (strSQL)
If strStuid= "" or strPW= "" then
        Response.write "You did not enter a user ID or Password. Try again."
        Response.write "<a
href=http://129.137.100.161/CASDistanceLearning/login.htm>Click Here To Try
Again</a>"
Else
        if rst.eof then
                Response.write "Your Student ID is not in the System!"
                Response.Write "<BR>"
                Response.write "<a
href=http://129.137.100.161/CASDistanceLearning/login.htm>Click Here To Try
Again</a>"
        Else
                If rst ("Password") = strPW Then
                        strFName = rst ("stFName")
                        rst.movefirst 'set the recordset for the loop
                        'rst.movelast
                        Response.write "<FONT><H2>Welcome, " & rst ("stFName") &
"</H2>"
                        Response.write "<br>"
                        Response.Write "<TABLE border=1 cellPadding=1 cellSpacing=1
><TR><TD>"
                        Response.Write "<A
href=http://129.137.100.161/CASDistanceLearning/CourseCatalog.asp>Course
Catalog</A>"
                        Response.write "<br>"
                        Response.Write "<A
href=http://129.137.100.161/CASDistanceLearning/studAdmin.asp>Edit Your
Information</A>"
                        Response.Write "</TD><TD>"
                        rst.close 'we know the student has filled out the application
                        Set rst=objConn.execute (strSQLrq)'this string will return the
courses
                        if rst.EOF = false then
                                Do while rst.EOF = false 'write out registered course links
                                        strCoNum = rst ("coNum")
                                        strCoName = rst ("coName")
                                        strCoCred = rst ("coCredHour")
                                        if rst ("coName") = "WebCT" then
```

```
                                        Response.Write "<A
HREF=http://nyland.zmr.uc.edu:8900/>"& strCoNum & "</A>"
                                        Response.Write " "
                                        Response.Write strCoName
                                        Response.Write " "
                                        Response.Write strCoCred & " Credit
Hours"
                                        Response.Write "<BR>"

                                else
                                        if rst ("coName") <> "WebCT" then
                                                Response.Write "<A
HREF=http://nyland.zmr.uc.edu/" & strCoName & ">"& strCoNum & "</A>"
                                                Response.Write " "
                                                Response.Write strCoName
                                                Response.Write " "
                                                Response.Write strCoCred & "
Credit Hours"
                                                Response.Write "<BR>"
                                        end if
                                end if
                                rst.movenext
                        Loop
                else
                Response.Write "Please Select Your Classes <BR> From The
Course Catalog Link To Your Left"
                end if
                Response.Write "</TD></TR></TABLE>"
                Session ("User") = strStuid

        Else
                Response.write "Your Student ID: <strong>" & strStuid &
"</strong> is OK, Wrong Password!"
                Response.write "<a
href=http://129.137.100.161/CASDistanceLearning/login.htm>Click Here To Try
Again</a>"
        End if
    End if
End if
rst.close
</SCRIPT>
<META name=VI60_defaultClientScript content=VBScript>
<title>CAS Distance Learning - Login</title>
</head>
<body bgColor=white  bgProperties=fixed
background=images/Background_GIFS/sandston-14.gif>
```

```
<CENTER>
<H1><FONT color=black face="Trebuchet MS">Login</FONT></H1></CENTER>
<p>
<P>
<P>
<hr>
<P> </P>
</body>
<% ' VI 6.0 Scripting Object Model Enabled %>
<% EndPageProcessing() %>
</FORM>
</html>
```

Application Page (application.asp)

```
<%@ Language=VBScript %>
<SCRIPT id=DebugDirectives runat=server language=javascript>
// Set these to true to enable debugging or tracing
@set @debug=false
@set @trace=false
</SCRIPT>

<% ' VI 6.0 Scripting Object Model Enabled %>
<!--#include file="_ScriptLibrary/pm.asp"-->
<% if StartPageProcessing() Then Response.End() %>
<FORM name=thisForm METHOD=post>
<HTML>
<HEAD>
<META name=VI60_defaultClientScript content=VBScript>

<%

dim strFName, strLName, strAdd1, strAdd2, strCity, strState, strZip, strEmail, strPhone
dim strStuid
dim strPW, strPW1, strPW2
dim strCCNum, strddlCCD, strExpDate
dim newID

Recordset1.moveLast
with Request
        strPW1 = .Form("txtPW2")
        strFName = .Form("txtFName")
        strLName = .Form("txtLName")
        strAdd1 = .Form("txtAdd1")
        strAdd2 = .Form("txtAdd2")
        strCity = .Form("txtCity")
        strState = .Form("txtState")
        strZip = .Form("txtZip")
        strEmail = .Form("txtEmail")
        strPhone = .Form("txtHPhone")
        strStuid = .Form("txtStuid")
        strCCNum = .Form("txtCCNum")
        strExpDate = .Form("txtExpDate")
end with
Recordset1.addRecord

txtAdd1.value = strAdd1
txtAdd2.value = strAdd2
txtCity.value = strCity
```

```
txtState.value = strState
txtZip.value = strZip
txtEmail.value = strEmail
txtStuid.value = strStuid
txtPWD.value = strPW1
txtCCNum.value = strCCNum
txtFName.value = strFName
txtLName.value = strLName
txtHPhone.value = strPhone



Sub btnSubmit_onclick()
Recordset1.updateRecord
Response.Redirect "login.htm"
End Sub

%>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">

<LINK REL="stylesheet" TYPE="text/css" HREF="_Themes/topo/THEME.CSS"
VI6.0THEME="Topo">
<LINK REL="stylesheet" TYPE="text/css" HREF="_Themes/topo/GRAPH0.CSS"
VI6.0THEME="Topo">
<LINK REL="stylesheet" TYPE="text/css" HREF="_Themes/topo/COLOR0.CSS"
VI6.0THEME="Topo">
<LINK REL="stylesheet" TYPE="text/css" HREF="_Themes/topo/CUSTOM.CSS"
VI6.0THEME="Topo">

</HEAD>
<BODY>

<P>
<!--METADATA TYPE="DesignerControl" startspan
<OBJECT classid="clsid:9CF5D7C2-EC10-11D0-9862-0000F8027CA0" id=Recordset1
style="LEFT: 0px; TOP: 0px">
        <PARAM NAME="ExtentX" VALUE="12198">
        <PARAM NAME="ExtentY" VALUE="2090">
        <PARAM NAME="State"
VALUE="(TCConn=\qconSnider\q,TCDBObject=\qTables\q,TCDBObjectName=\qtbSt
udents\q,TCControlID_Unmatched=\qRecordset1\q,TCPPConn=\qconSnider\q,RCDBOb
ject=\qRCDBObject\q,TCPPDBObject=\qTables\q,TCPPDBObjectName=\qtbStudents\q
,TCCursorType=\q3\s-\sStatic\q,TCCursorLocation=\q3\s-\sUse\sclient-
side\scursors\q,TCLockType=\q3\s-
\sOptimistic\q,TCCacheSize_Unmatched=\q10\q,TCCommTimeout_Unmatched=\q10\q,
CCPrepared=0,CCAllRecords=1,TCNRecords_Unmatched=\q10\q,TCODBCSyntax_Un
```

```
matched=\q\q,TCHTargetPlatform=\q\q,TCHTargetBrowser_Unmatched=\qServer\s(AS
P)\q,TCTargetPlatform=\qInherit\sfrom\spage\q,RCCache=\qRCBookPage\q,CCOpen=1,
GCParameters=(Rows=0))">

                        </OBJECT>
-->
<!--#INCLUDE FILE="_ScriptLibrary/Recordset.ASP"-->
<SCRIPT LANGUAGE="JavaScript" RUNAT="server">
function _initRecordset1()
{
        var DBConn = Server.CreateObject('ADODB.Connection');
        DBConn.ConnectionTimeout = Application('conSnider_ConnectionTimeout');
        DBConn.CommandTimeout = Application('conSnider_CommandTimeout');
        DBConn.CursorLocation = Application('conSnider_CursorLocation');
        DBConn.Open(Application('conSnider_ConnectionString'),
Application('conSnider_RuntimeUserName'),
Application('conSnider_RuntimePassword'));
        var cmdTmp = Server.CreateObject('ADODB.Command');
        var rsTmp = Server.CreateObject('ADODB.Recordset');
        cmdTmp.ActiveConnection = DBConn;
        rsTmp.Source = cmdTmp;
        cmdTmp.CommandType = 2;
        cmdTmp.CommandTimeout = 10;
        cmdTmp.CommandText = '"tbStudents"';
        rsTmp.CacheSize = 10;
        rsTmp.CursorType = 3;
        rsTmp.CursorLocation = 3;
        rsTmp.LockType = 3;
        Recordset1.setRecordSource(rsTmp);
        Recordset1.open();
        if (thisPage.getState('pb_Recordset1') != null)
                Recordset1.setBookmark(thisPage.getState('pb_Recordset1'));
}
function _Recordset1_ctor()
{
        CreateRecordset('Recordset1', _initRecordset1, null);
}
function _Recordset1_dtor()
{
        Recordset1._preserveState();
        thisPage.setState('pb_Recordset1', Recordset1.getBookmark());
}
</SCRIPT>

<!--METADATA TYPE="DesignerControl" endspan-->
</P>
```

```
<P>You Entered:</P>
<P>
</P>
<P>

</P>

<P>
<TABLE cellPadding=2 cellSpacing=2 cols=2>
 <TR>
  <TD>First Name
  <TD>
   <!--METADATA TYPE="DesignerControl" startspan
<OBJECT classid="clsid:B5F0E469-DC5F-11D0-9846-0000F8027CA0" id=txtFName
    style="LEFT: 0px; TOP: 0px">
        <PARAM NAME="_ExtentX" VALUE="3175">
        <PARAM NAME="_ExtentY" VALUE="503">
        <PARAM NAME="id" VALUE="txtFName">
        <PARAM NAME="ControlType" VALUE="0">
        <PARAM NAME="Lines" VALUE="3">
        <PARAM NAME="DataSource" VALUE="Recordset1">
        <PARAM NAME="DataField" VALUE="stFName">
        <PARAM NAME="Enabled" VALUE="-1">
        <PARAM NAME="Visible" VALUE="-1">
        <PARAM NAME="MaxChars" VALUE="30">
        <PARAM NAME="DisplayWidth" VALUE="20">
        <PARAM NAME="Platform" VALUE="0">
        <PARAM NAME="LocalPath" VALUE="">


         </OBJECT>
-->
<!--#INCLUDE FILE="_ScriptLibrary/TextBox.ASP"-->
<SCRIPT LANGUAGE=JavaScript RUNAT=Server>
function _inittxtFName()
{
        txtFName.setStyle(TXT_TEXTBOX);
        txtFName.setDataSource(Recordset1);
        txtFName.setDataField('stFName');
        txtFName.setMaxLength(30);
        txtFName.setColumnCount(20);
}
function _txtFName_ctor()
{
        CreateTextbox('txtFName', _inittxtFName, null);
}
```

```
</script>
<% txtFName.display %>

<!--METADATA TYPE="DesignerControl" endspan-->
</TD>
 <TR>
  <TD>Last Name
  <TD>
   <!--METADATA TYPE="DesignerControl" startspan
<OBJECT classid="clsid:B5F0E469-DC5F-11D0-9846-0000F8027CA0" id=txtLName
   style="LEFT: 0px; TOP: 0px">
       <PARAM NAME="_ExtentX" VALUE="3175">
       <PARAM NAME="_ExtentY" VALUE="503">
       <PARAM NAME="id" VALUE="txtLName">
       <PARAM NAME="ControlType" VALUE="0">
       <PARAM NAME="Lines" VALUE="3">
       <PARAM NAME="DataSource" VALUE="Recordset1">
       <PARAM NAME="DataField" VALUE="stLName">
       <PARAM NAME="Enabled" VALUE="-1">
       <PARAM NAME="Visible" VALUE="-1">
       <PARAM NAME="MaxChars" VALUE="30">
       <PARAM NAME="DisplayWidth" VALUE="20">
       <PARAM NAME="Platform" VALUE="0">
       <PARAM NAME="LocalPath" VALUE="">


        </OBJECT>
-->
<SCRIPT LANGUAGE=JavaScript RUNAT=Server>
function _inittxtLName()
{
        txtLName.setStyle(TXT_TEXTBOX);
        txtLName.setDataSource(Recordset1);
        txtLName.setDataField('stLName');
        txtLName.setMaxLength(30);
        txtLName.setColumnCount(20);
}
function _txtLName_ctor()
{
        CreateTextbox('txtLName', _inittxtLName, null);
}
</script>
<% txtLName.display %>

<!--METADATA TYPE="DesignerControl" endspan-->
</TD>
```

```
 <TR>
        <TD>Student ID 
        <TD>
    <!--METADATA TYPE="DesignerControl" startspan
<OBJECT classid="clsid:B5F0E469-DC5F-11D0-9846-0000F8027CA0" id=txtStuid
    style="LEFT: 0px; TOP: 0px">
        <PARAM NAME="_ExtentX" VALUE="3175">
        <PARAM NAME="_ExtentY" VALUE="503">
        <PARAM NAME="id" VALUE="txtStuid">
        <PARAM NAME="ControlType" VALUE="0">
        <PARAM NAME="Lines" VALUE="3">
        <PARAM NAME="DataSource" VALUE="Recordset1">
        <PARAM NAME="DataField" VALUE="stUCNum">
        <PARAM NAME="Enabled" VALUE="-1">
        <PARAM NAME="Visible" VALUE="-1">
        <PARAM NAME="MaxChars" VALUE="9">
        <PARAM NAME="DisplayWidth" VALUE="20">
        <PARAM NAME="Platform" VALUE="0">
        <PARAM NAME="LocalPath" VALUE="">


         </OBJECT>
-->
<SCRIPT LANGUAGE=JavaScript RUNAT=Server>
function _inittxtStuid()
{
        txtStuid.setStyle(TXT_TEXTBOX);
        txtStuid.setDataSource(Recordset1);
        txtStuid.setDataField('stUCNum');
        txtStuid.setMaxLength(9);
        txtStuid.setColumnCount(20);
}
function _txtStuid_ctor()
{
        CreateTextbox('txtStuid', _inittxtStuid, null);
}
</script>
<% txtStuid.display %>

<!--METADATA TYPE="DesignerControl" endspan-->
</TD>

 <TR>
  <TD>
  Card
  No.
```

```
  <TD>
    <!--METADATA TYPE="DesignerControl" startspan
<OBJECT classid="clsid:B5F0E469-DC5F-11D0-9846-0000F8027CA0" id=txtCCNum
    style="LEFT: 0px; TOP: 0px">
        <PARAM NAME="_ExtentX" VALUE="3175">
        <PARAM NAME="_ExtentY" VALUE="503">
        <PARAM NAME="id" VALUE="txtCCNum">
        <PARAM NAME="ControlType" VALUE="0">
        <PARAM NAME="Lines" VALUE="3">
        <PARAM NAME="DataSource" VALUE="Recordset1">
        <PARAM NAME="DataField" VALUE="stCCNum">
        <PARAM NAME="Enabled" VALUE="-1">
        <PARAM NAME="Visible" VALUE="-1">
        <PARAM NAME="MaxChars" VALUE="16">
        <PARAM NAME="DisplayWidth" VALUE="20">
        <PARAM NAME="Platform" VALUE="0">
        <PARAM NAME="LocalPath" VALUE="">


</OBJECT>
-->
<SCRIPT LANGUAGE=JavaScript RUNAT=Server>
function _inittxtCCNum()
{
        txtCCNum.setStyle(TXT_TEXTBOX);
        txtCCNum.setDataSource(Recordset1);
        txtCCNum.setDataField('stCCNum');
        txtCCNum.setMaxLength(16);
        txtCCNum.setColumnCount(20);
}
function _txtCCNum_ctor()
{
        CreateTextbox('txtCCNum', _inittxtCCNum, null);
}
</script>
<% txtCCNum.display %>

<!--METADATA TYPE="DesignerControl" endspan-->
</TD>
 <TR>

</TD>
 <TR>
  <TD>Email
  <TD>
    <!--METADATA TYPE="DesignerControl" startspan
```

```
<OBJECT classid="clsid:B5F0E469-DC5F-11D0-9846-0000F8027CA0" id=txtEmail
    style="LEFT: 0px; TOP: 0px">
        <PARAM NAME="_ExtentX" VALUE="3175">
        <PARAM NAME="_ExtentY" VALUE="503">
        <PARAM NAME="id" VALUE="txtEmail">
        <PARAM NAME="ControlType" VALUE="0">
        <PARAM NAME="Lines" VALUE="3">
        <PARAM NAME="DataSource" VALUE="Recordset1">
        <PARAM NAME="DataField" VALUE="stEmail">
        <PARAM NAME="Enabled" VALUE="-1">
        <PARAM NAME="Visible" VALUE="-1">
        <PARAM NAME="MaxChars" VALUE="40">
        <PARAM NAME="DisplayWidth" VALUE="20">
        <PARAM NAME="Platform" VALUE="0">
        <PARAM NAME="LocalPath" VALUE="">


         </OBJECT>
-->
<SCRIPT LANGUAGE=JavaScript RUNAT=Server>
function _inittxtEmail()
{
        txtEmail.setStyle(TXT_TEXTBOX);
        txtEmail.setDataSource(Recordset1);
        txtEmail.setDataField('stEmail');
        txtEmail.setMaxLength(40);
        txtEmail.setColumnCount(20);
}
function _txtEmail_ctor()
{
        CreateTextbox('txtEmail', _inittxtEmail, null);
}
</script>
<% txtEmail.display %>

<!--METADATA TYPE="DesignerControl" endspan-->
</TD>
 <TR>
  <TD>Address 1
  <TD>
   <!--METADATA TYPE="DesignerControl" startspan
<OBJECT classid="clsid:B5F0E469-DC5F-11D0-9846-0000F8027CA0" id=txtAdd1
    style="LEFT: 0px; TOP: 0px">
        <PARAM NAME="_ExtentX" VALUE="3175">
        <PARAM NAME="_ExtentY" VALUE="503">
        <PARAM NAME="id" VALUE="txtAdd1">
```

```
        <PARAM NAME="ControlType" VALUE="0">
        <PARAM NAME="Lines" VALUE="3">
        <PARAM NAME="DataSource" VALUE="Recordset1">
        <PARAM NAME="DataField" VALUE="stAddr1">
        <PARAM NAME="Enabled" VALUE="-1">
        <PARAM NAME="Visible" VALUE="-1">
        <PARAM NAME="MaxChars" VALUE="69">
        <PARAM NAME="DisplayWidth" VALUE="20">
        <PARAM NAME="Platform" VALUE="0">
        <PARAM NAME="LocalPath" VALUE="">


         </OBJECT>
-->
<SCRIPT LANGUAGE=JavaScript RUNAT=Server>
function _inittxtAdd1()
{
        txtAdd1.setStyle(TXT_TEXTBOX);
        txtAdd1.setDataSource(Recordset1);
        txtAdd1.setDataField('stAddr1');
        txtAdd1.setMaxLength(69);
        txtAdd1.setColumnCount(20);
}
function _txtAdd1_ctor()
{
        CreateTextbox('txtAdd1', _inittxtAdd1, null);
}
</script>
<% txtAdd1.display %>

<!--METADATA TYPE="DesignerControl" endspan-->
</TD>
 <TR>
  <TD>Address 2
  <TD>
   <!--METADATA TYPE="DesignerControl" startspan
<OBJECT classid="clsid:B5F0E469-DC5F-11D0-9846-0000F8027CA0" id=txtAdd2
    style="LEFT: 0px; TOP: 0px">
        <PARAM NAME="_ExtentX" VALUE="3175">
        <PARAM NAME="_ExtentY" VALUE="503">
        <PARAM NAME="id" VALUE="txtAdd2">
        <PARAM NAME="ControlType" VALUE="0">
        <PARAM NAME="Lines" VALUE="3">
        <PARAM NAME="DataSource" VALUE="Recordset1">
        <PARAM NAME="DataField" VALUE="stAddr2">
        <PARAM NAME="Enabled" VALUE="-1">
```

```
        <PARAM NAME="Visible" VALUE="-1">
        <PARAM NAME="MaxChars" VALUE="69">
        <PARAM NAME="DisplayWidth" VALUE="20">
        <PARAM NAME="Platform" VALUE="0">
        <PARAM NAME="LocalPath" VALUE="">


         </OBJECT>
-->
<SCRIPT LANGUAGE=JavaScript RUNAT=Server>
function _inittxtAdd2()
{
        txtAdd2.setStyle(TXT_TEXTBOX);
        txtAdd2.setDataSource(Recordset1);
        txtAdd2.setDataField('stAddr2');
        txtAdd2.setMaxLength(69);
        txtAdd2.setColumnCount(20);
}
function _txtAdd2_ctor()
{
        CreateTextbox('txtAdd2', _inittxtAdd2, null);
}
</script>
<% txtAdd2.display %>

<!--METADATA TYPE="DesignerControl" endspan-->
</TD>
 <TR>
  <TD>City
  <TD>
   <!--METADATA TYPE="DesignerControl" startspan
<OBJECT classid="clsid:B5F0E469-DC5F-11D0-9846-0000F8027CA0" id=txtCity
    style="LEFT: 0px; TOP: 0px">
        <PARAM NAME="_ExtentX" VALUE="3175">
        <PARAM NAME="_ExtentY" VALUE="503">
        <PARAM NAME="id" VALUE="txtCity">
        <PARAM NAME="ControlType" VALUE="0">
        <PARAM NAME="Lines" VALUE="3">
        <PARAM NAME="DataSource" VALUE="Recordset1">
        <PARAM NAME="DataField" VALUE="stCity">
        <PARAM NAME="Enabled" VALUE="-1">
        <PARAM NAME="Visible" VALUE="-1">
        <PARAM NAME="MaxChars" VALUE="30">
        <PARAM NAME="DisplayWidth" VALUE="20">
        <PARAM NAME="Platform" VALUE="0">
        <PARAM NAME="LocalPath" VALUE="">
```

```
        </OBJECT>
-->
<SCRIPT LANGUAGE=JavaScript RUNAT=Server>
function _inittxtCity()
{
        txtCity.setStyle(TXT_TEXTBOX);
        txtCity.setDataSource(Recordset1);
        txtCity.setDataField('stCity');
        txtCity.setMaxLength(30);
        txtCity.setColumnCount(20);
}
function _txtCity_ctor()
{
        CreateTextbox('txtCity', _inittxtCity, null);
}
</script>
<% txtCity.display %>

<!--METADATA TYPE="DesignerControl" endspan-->
</TD>
 <TR>
  <TD>State
  <TD>
   <!--METADATA TYPE="DesignerControl" startspan
<OBJECT classid="clsid:B5F0E469-DC5F-11D0-9846-0000F8027CA0" id=txtState
    style="LEFT: 0px; TOP: 0px">
        <PARAM NAME="_ExtentX" VALUE="3175">
        <PARAM NAME="_ExtentY" VALUE="503">
        <PARAM NAME="id" VALUE="txtState">
        <PARAM NAME="ControlType" VALUE="0">
        <PARAM NAME="Lines" VALUE="3">
        <PARAM NAME="DataSource" VALUE="Recordset1">
        <PARAM NAME="DataField" VALUE="stState">
        <PARAM NAME="Enabled" VALUE="-1">
        <PARAM NAME="Visible" VALUE="-1">
        <PARAM NAME="MaxChars" VALUE="2">
        <PARAM NAME="DisplayWidth" VALUE="20">
        <PARAM NAME="Platform" VALUE="0">
        <PARAM NAME="LocalPath" VALUE="">


         </OBJECT>
-->
<SCRIPT LANGUAGE=JavaScript RUNAT=Server>
```

```
function _inittxtState()
{
        txtState.setStyle(TXT_TEXTBOX);
        txtState.setDataSource(Recordset1);
        txtState.setDataField('stState');
        txtState.setMaxLength(2);
        txtState.setColumnCount(20);
}
function _txtState_ctor()
{
        CreateTextbox('txtState', _inittxtState, null);
}
</script>
<% txtState.display %>

<!--METADATA TYPE="DesignerControl" endspan-->
</TD>
 <TR>
  <TD>Zip
  <TD>
   <!--METADATA TYPE="DesignerControl" startspan
<OBJECT classid="clsid:B5F0E469-DC5F-11D0-9846-0000F8027CA0" id=txtZip
    style="LEFT: 0px; TOP: 0px">
        <PARAM NAME="_ExtentX" VALUE="3175">
        <PARAM NAME="_ExtentY" VALUE="503">
        <PARAM NAME="id" VALUE="txtZip">
        <PARAM NAME="ControlType" VALUE="0">
        <PARAM NAME="Lines" VALUE="3">
        <PARAM NAME="DataSource" VALUE="Recordset1">
        <PARAM NAME="DataField" VALUE="stZip">
        <PARAM NAME="Enabled" VALUE="-1">
        <PARAM NAME="Visible" VALUE="-1">
        <PARAM NAME="MaxChars" VALUE="10">
        <PARAM NAME="DisplayWidth" VALUE="20">
        <PARAM NAME="Platform" VALUE="0">
        <PARAM NAME="LocalPath" VALUE="">


         </OBJECT>
-->
<SCRIPT LANGUAGE=JavaScript RUNAT=Server>
function _inittxtZip()
{
        txtZip.setStyle(TXT_TEXTBOX);
        txtZip.setDataSource(Recordset1);
        txtZip.setDataField('stZip');
```

```
        txtZip.setMaxLength(10);
        txtZip.setColumnCount(20);
}
function _txtZip_ctor()
{
        CreateTextbox('txtZip', _inittxtZip, null);
}
</script>
<% txtZip.display %>

<!--METADATA TYPE="DesignerControl" endspan-->
</TD>
 <TR>
  <TD>Phone
  <TD>
   <!--METADATA TYPE="DesignerControl" startspan
<OBJECT classid="clsid:B5F0E469-DC5F-11D0-9846-0000F8027CA0" id=txtHPhone
    style="LEFT: 0px; TOP: 0px">
        <PARAM NAME="_ExtentX" VALUE="3175">
        <PARAM NAME="_ExtentY" VALUE="503">
        <PARAM NAME="id" VALUE="txtHPhone">
        <PARAM NAME="ControlType" VALUE="0">
        <PARAM NAME="Lines" VALUE="3">
        <PARAM NAME="DataSource" VALUE="Recordset1">
        <PARAM NAME="DataField" VALUE="stHPhone">
        <PARAM NAME="Enabled" VALUE="-1">
        <PARAM NAME="Visible" VALUE="-1">
        <PARAM NAME="MaxChars" VALUE="16">
        <PARAM NAME="DisplayWidth" VALUE="20">
        <PARAM NAME="Platform" VALUE="0">
        <PARAM NAME="LocalPath" VALUE="">


         </OBJECT>
 -->
 <SCRIPT LANGUAGE=JavaScript RUNAT=Server>
 function _inittxtHPhone()
 {
        txtHPhone.setStyle(TXT_TEXTBOX);
        txtHPhone.setDataSource(Recordset1);
        txtHPhone.setDataField('stHPhone');
        txtHPhone.setMaxLength(16);
        txtHPhone.setColumnCount(20);
 }
 function _txtHPhone_ctor()
 {
```

```
        CreateTextbox('txtHPhone', _inittxtHPhone, null);
}
</script>
<% txtHPhone.display %>

<!--METADATA TYPE="DesignerControl" endspan-->
</TD>

 <TR>
  <TD>Password
  <TD>
   <!--METADATA TYPE="DesignerControl" startspan
<OBJECT classid="clsid:B5F0E469-DC5F-11D0-9846-0000F8027CA0" height=19
    id=txtPWD style="HEIGHT: 19px; LEFT: 0px; TOP: 0px; WIDTH: 120px"
    width=120>
        <PARAM NAME="_ExtentX" VALUE="3175">
        <PARAM NAME="_ExtentY" VALUE="503">
        <PARAM NAME="id" VALUE="txtPWD">
        <PARAM NAME="ControlType" VALUE="0">
        <PARAM NAME="Lines" VALUE="3">
        <PARAM NAME="DataSource" VALUE="Recordset1">
        <PARAM NAME="DataField" VALUE="Password">
        <PARAM NAME="Enabled" VALUE="-1">
        <PARAM NAME="Visible" VALUE="-1">
        <PARAM NAME="MaxChars" VALUE="20">
        <PARAM NAME="DisplayWidth" VALUE="20">
        <PARAM NAME="Platform" VALUE="0">
        <PARAM NAME="LocalPath" VALUE="">


         </OBJECT>
-->
<SCRIPT LANGUAGE=JavaScript RUNAT=Server>
function _inittxtPWD()
{
        txtPWD.setStyle(TXT_TEXTBOX);
        txtPWD.setDataSource(Recordset1);
        txtPWD.setDataField('Password');
        txtPWD.setMaxLength(20);
        txtPWD.setColumnCount(20);
}
function _txtPWD_ctor()
{
        CreateTextbox('txtPWD', _inittxtPWD, null);
}
</script>
```

```
<% txtPWD.display %>

<!--METADATA TYPE="DesignerControl" endspan-->
</TD></TR></TABLE></P>
<P> </P>
<P>
<!--METADATA TYPE="DesignerControl" startspan
<OBJECT classid="clsid:B6FC3A14-F837-11D0-9CC8-006008058731" height=27
id=btnSubmit style="HEIGHT: 27px; LEFT: 0px; TOP: 0px; WIDTH: 63px" width=63>
        <PARAM NAME="_ExtentX" VALUE="1667">
        <PARAM NAME="_ExtentY" VALUE="714">
        <PARAM NAME="id" VALUE="btnSubmit">
        <PARAM NAME="Caption" VALUE="Submit">
        <PARAM NAME="Image" VALUE="">
        <PARAM NAME="AltText" VALUE="">
        <PARAM NAME="Visible" VALUE="-1">
        <PARAM NAME="Platform" VALUE="0">
        <PARAM NAME="LocalPath" VALUE="">

                                                        </OBJECT>
-->
<!--#INCLUDE FILE="_ScriptLibrary/Button.ASP"-->
<SCRIPT LANGUAGE=JavaScript RUNAT=Server>
function _initbtnSubmit()
{
        btnSubmit.value = 'Submit';
        btnSubmit.setStyle(0);
}
function _btnSubmit_ctor()
{
        CreateButton('btnSubmit', _initbtnSubmit, null);
}
</script>
<% btnSubmit.display %>

<!--METADATA TYPE="DesignerControl" endspan-->
</P>

</BODY>
<% ' VI 6.0 Scripting Object Model Enabled %>
<% EndPageProcessing() %>
</FORM>
</HTML>
```

Application Page (application.htm)

```
<%@ Language=VBScript%>
<script id="DebugDirectives" runat="server" language="javascript">
// Set these to true to enable debugging or tracing
@set @debug=false
@set @trace=false
</script>

<html>
<title></title>
<head>
<meta name="VI60_DTCScriptingPlatform" content="Server (ASP)">
<meta name="VI60_defaultClientScript" content="VBScript">

<SCRIPT ID=clientEventHandlersVBS LANGUAGE=vbscript>
<!--text box validation
        Option Explicit
```

```
Dim strMsgBoxTitle
Dim bValidOrder
dim strFName, strLName, strAdd1, strAdd2, strCity, strState, strZip,
strEmail, strPhone
dim strStuid
dim strPW, strPW1, strPW2
dim strCCNum, strddlCCD
dim strExpDate
dim curYear
dim curMonth
dim cdMonth, cdYear

Sub Window_OnLoad
    strMsgBoxTitle = "eCAS"
End Sub


Sub cmdOK_OnClick
    bValidOrder = True

    with window.frmLogin
        strFName = .txtFName.value
        strLName = .txtLName.value
        strAdd1 = .txtAdd1.value
        strAdd2 = .txtAdd2.value
        strCity = .txtCity.value
        strState = .txtState.value
        strZip = .txtZip.value
        strEmail = .txtEmail.value
        strPhone = .txtHPhone.value
        strStuid = .txtStuid.value
        strPW1 = .txtPW1.value
        strPW2 = .txtPW2.value
        strCCNum = .txtCCNum.value
        strddlCCD = .ddlCCD.value
        strExpDate = .txtExpDate.value
    end with

    Call CheckSpecified(strFName,    "Please specify a first
name.")
    Call CheckSpecified(strLName,    "Please specify a last name.")
    Call CheckSpecified(strAdd1,     "Please specify an address.")
    Call CheckSpecified(strCity,  "Please specify a city.")
    Call CheckSpecified(strState,    "Please specify a state.")
    Call CheckSpecified(strZip,   "Please specify a zip code.")
```

```
                Call CheckSpecified(strExpDate, "Please specify a date.")
                Call CheckSpecified(strEmail,   "Please specify an Email
Address.")
                Call CheckSpecified(strPhone,  "Please specify a phone number.")
                Call CheckStudentID(strStuid,   "Please specify your Student ID.")
                Call CheckPassword
                Call ValidateCreditCard

                If bValidOrder=true Then
                        'MsgBox "Thank you for registering at eCAS!", 0,
strMsgBoxTitle

                        'Actually send the order -- via a hidden submit button.
                        window.frmLogin.cmdSubmit.click
                End If
        End Sub


        Sub ValidateCreditCard
                curYear = 0
                curYear = right((Year(date)),2)
                curMonth = Month(date)
                cdMonth = left(strExpDate,2)
                cdYear = right(strExpDate,4)

                If strCCNum = "" And bValidOrder Then
                        MsgBox "This is not a valid card number!", 0,
strMsgBoxTitle
                        bValidOrder = False
                Else
                        if not isnumeric(strCCNum) or (len(strCCNum)<16 or
len(strCCNum)>16) then
                                MsgBox "This is not a valid card number!", 0,
strMsgBoxTitle
                                bValidOrder = False
                        end if
                End If


                if curYear < 50 then 'Y2K me baby!
                        curYear = "20" & curYear
                else
                        curYear = "19" & curYear
                end if
```

```
                if cdYear < curYear then
                        MsgBox "Your Card is Expired!", 0, strMsgBoxTitle
                        bValidOrder = False
                        else
                if (cdMonth) < curMonth then 'this is a bitch and is not working
                                MsgBox "Your Card is Expired!", 0,
strMsgBoxTitle
                                bValidOrder = False
                        end if
                end if
        End Sub

        Sub CheckPassword
                If strPW1 = "" And bValidOrder Then
                        MsgBox "You didn't enter a password!", 0, strMsgBoxTitle
                        bValidOrder = False
                else
                        If strPW1 <> strPW2 And bValidOrder then
                                MsgBox "Your passwords don't match!", 0,
strMsgBoxTitle
                                bValidOrder = False
                        End If
                end if
                If (len(strPW1)< 4) And bValidOrder then
                        MsgBox "Passwords Need to be at least 4 characters!", 0,
strMsgBoxTitle
                        bValidOrder = False
                End If
        End Sub
        Sub CheckSpecified(ByVal strFieldValue, ByVal strMsg)
                If strFieldValue = "" And bValidOrder Then
                        MsgBox strMsg, 0, strMsgBoxTitle
                        bValidOrder = False
                End If
        End Sub

        Sub CheckStudentID(ByVal strFieldValue, ByVal strMsg)
                If strFieldValue = "" And bValidOrder Then
                        MsgBox strMsg, 0, strMsgBoxTitle
                        bValidOrder = False
                Else
                        if not isnumeric(strFieldValue) or (len(strFieldValue)<9 or
len(strFieldValue)>9) then
                                MsgBox strMsg, 0, strMsgBoxTitle
                                bValidOrder = False
                        end if
```

```
                End If
            End Sub
        -->
        </SCRIPT>
        </head>
        <body background=images/Background_GIFS/sandston-14.gif>
        <form action="Application.asp" method="post" name="frmLogin">

        <h1><FONT face="Trebuchet MS"
        size=4>Registration</FONT> </h1>

        <table border="0" cellPadding="1" cellSpacing="1" id="TABLE1"
style="HEIGHT: 347px; WIDTH: 391px" width="75%">

          <tr>
           <td>First Name<br><input id="txtFName" name="txtFName" tabIndex=1>
</td>
           <td>Last Name<br><input id="text8" name="txtLName" tabIndex=2> </td>
                <td>E-Mail<BR><INPUT id=text2 name=txtEmail
            tabIndex=3> </td></tr>
          <tr>
           <td>Address 1<br><input id="text7" name="txtAdd1" tabIndex=4> </td>
           <td>Address 2<br><input id="text6" name="txtAdd2" tabIndex=5> </td></tr>
          <tr>
           <td>City<br><input id="text5" name="txtCity" tabIndex=6> </td>
           <td><p>State<br><!--<input id="text4" name="txtState" tabIndex=7
maxLength=2 style="HEIGHT: 22px; TEXT-TRANSFORM: uppercase; WIDTH: 57px"
width="2">-->
           <select name="txtState" size="1">
               <option selected value=""></option>
               <option value="AL">Alabama</option>
               <option value="AK">Alaska</option>
               <option value="AZ">Arizona</option>
               <option value="AR">Arkansas</option>
               <option value="CA">California</option>
               <option value="CO">Colorado</option>
               <option value="CT">Connecticut</option>
               <option value="DE">Delaware</option>
               <option value="DC">District of Columbia</option>
               <option value="FL">Florida</option>
               <option value="GA">Georgia</option>
               <option value="HI">Hawaii</option>
               <option value="ID">Idaho</option>
               <option value="IL">Illinois</option>
               <option value="IN">Indiana</option>
```

```
                <option value="IA">Iowa</option>
                <option value="KS">Kansas</option>
                <option value="KY">Kentucky</option>
                <option value="LA">Louisiana</option>
                <option value="ME">Maine</option>
                <option value="MD">Maryland</option>
                <option value="MA">Massachusetts</option>
                <option value="MI">Michigan</option>
                <option value="MN">Minnesota</option>
                <option value="MS">Mississippi</option>
                <option value="MO">Missouri</option>
                <option value="MT">Montana</option>
                <option value="NE">Nebraska</option>
                <option value="NV">Nevada</option>
                <option value="NH">New Hampshire</option>
                <option value="NJ">New Jersey</option>
                <option value="NM">New Mexico</option>
                <option value="NY">New York</option>
                <option value="NC">North Carolina</option>
                <option value="ND">North Dakota</option>
                <option value="OH">Ohio</option>
                <option value="OK">Oklahoma</option>
                <option value="OR">Oregon</option>
                <option value="PA">Pennsylvania</option>
                <option value="RI">Rhode Island</option>
                <option value="SC">South Carolina</option>
                <option value="SD">South Dakota</option>
                <option value="TN">Tennessee</option>
                <option value="TX">Texas</option>
                <option value="UT">Utah</option>
                <option value="VT">Vermont</option>
                <option value="VA">Virginia</option>
                <option value="WA">Washington</option>
                <option value="WV">West Virginia</option>
                <option value="WI">Wisconsin</option>
                <option value="WY">Wyoming</option>
              </select></p>
            </td>
                <td>Zip<br><input id="text3" name="txtZip" tabIndex=8></td></tr>
          <tr>

            <td> </td></tr>
          <tr>
            <td>
              <p>Home Phone<br><input id="txtHPhone" name="txtHPhone"
tabIndex=9></p> </td>
```

```
            <td>Work Phone      <br><input
id="txtWPhone" name="txtWPhone" tabIndex=10></td></tr>
          <tr>
           <td noWrap vAlign="top">
            <p style="FONT-FAMILY: sans-serif" align=left>Student
            ID<BR>
            <FONT size=1>(SSN-No Hyphen)  <br></FONT>
            <FONT size=1>
            <input id="txtStuid" name="txtStuid" tabIndex=11
            style="LEFT: 110px; TOP: 5px"></FONT></p> </td>
           <td noWrap vAlign="top">
            <p tabIndex=0> </p></td></tr>

              <tr><td></td></tr>
              <tr>
               <td>
            <P>Card Type<BR><SELECT id=ddlCCD name=ddlCCD size=1
            style="HEIGHT: 24px; VERTICAL-ALIGN: middle; WIDTH: 154px"
tabIndex=12>
             <OPTION selected value=Visa>Visa</OPTION> <OPTION
             value=MasterCard>MasterCard</OPTION> <OPTION
             value=Discover>Discover</OPTION> <OPTION
             value=""
            > </OPTION></SELECT></P></td>
          <tr>
              <TD>
           <P style="VERTICAL-ALIGN: bottom">Credit Card
           Number<BR><INPUT id=txtCCNum name=txtCCNum style="HEIGHT:
22px; WIDTH: 154px" tabIndex=13></P></TD>
           <td>
           <P align=left><FONT size=3>Expiration Date
           </FONT><FONT size=1>(mm/yyyy)</FONT><BR><INPUT id=txtExpDate
name=txtExpDate style="HEIGHT: 22px; LEFT: 1px; TOP: 46px; VISIBILITY: visible;
WIDTH: 103px" tabIndex=14>
           </P></td><td>
           <P></P></td></TD>
              </tr>
              <tr><td></td><td></td>      <td>
           <P> </P></td></tr>
              <tr>
                    <td>Password<BR><INPUT id=txtPW1 name=txtPW1
          tabIndex=15 type=password></td>
                    <td>Confirm Password<BR><INPUT id=txtPW2 name=txtPW2
          tabIndex=16 type=password style="HEIGHT: 22px; WIDTH: 181px"></td>
              </tr></table>
       <P></P>
```

```
<P>
<TABLE border=0 cellPadding=1 cellSpacing=1 width="75%">

 <TR>
  <TD><INPUT id=cmdOK name=cmdOK type=button value=OK
tabIndex=17></TD>
  <TD></TD>
  <TD><INPUT id=reset name=reset style="HEIGHT: 24px; WIDTH: 81px"
tabIndex=18 type=reset value=Reset></TD></TR></TABLE></P>
   <p> </p>
<P><input id="cmdSubmit" name="cmdSubmit" style="HEIGHT: 24px; LEFT:
51px; TOP: 583px; VISIBILITY: hidden; WIDTH: 46px" type="submit" value="Submit"
tabIndex=0>       
</P>
 </form>
</body>
</html>
```

Course Catalog (CourseCatalog.asp)

```
<%@ Language=VBScript %>
<script id="DebugDirectives" runat="server" language="javascript">
// Set these to true to enable debugging or tracing
@set @debug=false
@set @trace=false
</script>


<% ' VI 6.0 Scripting Object Model Enabled %>
<!--#include file="_ScriptLibrary/pm.asp"-->
<% if StartPageProcessing() Then Response.End() %>
<FORM name=thisForm METHOD=post>
<form name="thisForm" METHOD="post">
<form name="thisForm" METHOD="post">
<html>
```

```
<head>
<meta NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">

<%

dim strStuid
dim strSQL

strStuid = Session ("User")
strSQL = "SELECT * From tbStudents WHERE stUCNum = '" & strStuid & "'"
rstStud.setSQLText(strSQL)
rstStud.open
rstStuCo.addRecord
txtStuid.value = strStuid
%>

<LINK REL="stylesheet" TYPE="text/css"
HREF="_Themes/topo/THEME.CSS" VI6.0THEME="Topo">
<LINK REL="stylesheet" TYPE="text/css"
HREF="_Themes/topo/GRAPH0.CSS" VI6.0THEME="Topo">
<LINK REL="stylesheet" TYPE="text/css"
HREF="_Themes/topo/COLOR0.CSS" VI6.0THEME="Topo">
<LINK REL="stylesheet" TYPE="text/css"
HREF="_Themes/topo/CUSTOM.CSS" VI6.0THEME="Topo"><SCRIPT
ID=serverEventHandlersVBS LANGUAGE=vbscript RUNAT=Server>

Sub btnSubmit_onclick()

        rstStuCo.updateRecord
        rstStuCo.addRecord
        txtStuid.value = strStuid
End Sub


Sub btnDone_onclick()
        Response.Redirect "cashier.asp"
End Sub

</SCRIPT>
</head>
<body>
<P align=left>
<!--METADATA TYPE="DesignerControl" startspan
<OBJECT classid="clsid:9CF5D7C2-EC10-11D0-9862-0000F8027CA0"
id=rstStud
style="LEFT: 0px; TOP: 0px">
```

```
                <PARAM NAME="ExtentX" VALUE="12197">
                <PARAM NAME="ExtentY" VALUE="2090">
                <PARAM NAME="State"
VALUE="(TCConn=\qconSnider\q,TCDBObject_Unmatched=\qSQL\sStatement\q,TCD
BObjectName_Unmatched=\qSELECT*\sFROM\stbStudents\s\q,TCControlID_Unmatch
ed=\qrstStud\q,TCPPConn=\qconSnider\q,TCPPDBObject=\qDE\sCommands\q,TCPPD
BObjectName=\qtbStudents\q,RCDBObject=\qRCSQLStatement\q,TCSQLStatement_U
nmatched=\qSELECT*\sFROM\stbStudents\s\q,TCCursorType=\q3\s-
\sStatic\q,TCCursorLocation=\q3\s-\sUse\sclient-side\scursors\q,TCLockType=\q3\s-
\sOptimistic\q,TCCacheSize_Unmatched=\q100\q,TCCommTimeout_Unmatched=\q30\
q,CCPrepared=0,CCAllRecords=1,TCNRecords_Unmatched=\q10\q,TCODBCSyntax_U
nmatched=\q\q,TCHTargetPlatform=\q\q,TCHTargetBrowser_Unmatched=\qServer\s(A
SP)\q,TCTargetPlatform=\qServer\s(ASP)\q,RCCache=\qRCBookPage\q,CCOpen=0,GC
Parameters=(Rows=0))">

                        </OBJECT>
        -->
        <!--#INCLUDE FILE="_ScriptLibrary/Recordset.ASP"-->
        <SCRIPT LANGUAGE="JavaScript" RUNAT="server">
        function _initrstStud()
        {
                var DBConn = Server.CreateObject('ADODB.Connection');
                DBConn.ConnectionTimeout =
Application('conSnider_ConnectionTimeout');
                DBConn.CommandTimeout =
Application('conSnider_CommandTimeout');
                DBConn.CursorLocation = Application('conSnider_CursorLocation');
                DBConn.Open(Application('conSnider_ConnectionString'),
Application('conSnider_RuntimeUserName'),
Application('conSnider_RuntimePassword'));
                var cmdTmp = Server.CreateObject('ADODB.Command');
                var rsTmp = Server.CreateObject('ADODB.Recordset');
                cmdTmp.ActiveConnection = DBConn;
                rsTmp.Source = cmdTmp;
                cmdTmp.CommandType = 1;
                cmdTmp.CommandTimeout = 30;
                cmdTmp.CommandText = 'SELECT* FROM tbStudents ';
                rsTmp.CacheSize = 100;
                rsTmp.CursorType = 3;
                rsTmp.CursorLocation = 3;
                rsTmp.LockType = 3;
                rstStud.setRecordSource(rsTmp);
                if (thisPage.getState('pb_rstStud') != null)
                        rstStud.setBookmark(thisPage.getState('pb_rstStud'));
        }
        function _rstStud_ctor()
```

```
{
        CreateRecordset('rstStud', _initrstStud, null);
}
function _rstStud_dtor()
{
        rstStud._preserveState();
        thisPage.setState('pb_rstStud', rstStud.getBookmark());
}
</SCRIPT>

<!--METADATA TYPE="DesignerControl" endspan-->
</P>
<P align=left>
<!--METADATA TYPE="DesignerControl" startspan
<OBJECT classid="clsid:9CF5D7C2-EC10-11D0-9862-0000F8027CA0"
id=rstCourses
style="LEFT: 0px; TOP: 0px">
        <PARAM NAME="ExtentX" VALUE="12197">
        <PARAM NAME="ExtentY" VALUE="2090">
        <PARAM NAME="State"
VALUE="(TCConn=\qconSnider\q,TCDBObject_Unmatched=\qSQL\sStatement\q,TCD
BObjectName_Unmatched=\qSELECT\scoNum,\scoName,\scoSumFall,\scoDistance,\sc
oID,\scoCost,\scoCredHour\sFROM\stbCourses\sWHERE\s(coDistance\s=\s'1')\q,TCCo
ntrolID_Unmatched=\qrstCourses\q,TCPPConn=\qconSnider\q,TCPPDBObject=\qDE\s
Commands\q,TCPPDBObjectName=\qtbCourses\q,RCDBObject=\qRCSQLStatement\q,
TCSQLStatement_Unmatched=\qSELECT\scoNum,\scoName,\scoSumFall,\scoDistance
,\scoID,\scoCost,\scoCredHour\sFROM\stbCourses\sWHERE\s(coDistance\s=\s'1')\q,TC
CursorType=\q3\s-\sStatic\q,TCCursorLocation=\q3\s-\sUse\sclient-
side\scursors\q,TCLockType=\q3\s-
\sOptimistic\q,TCCacheSize_Unmatched=\q100\q,TCCommTimeout_Unmatched=\q30\
q,CCPrepared=0,CCAllRecords=1,TCNRecords_Unmatched=\q10\q,TCODBCSyntax_U
nmatched=\q\q,TCHTargetPlatform=\q\q,TCHTargetBrowser_Unmatched=\qServer\s(A
SP)\q,TCTargetPlatform=\qInherit\sfrom\spage\q,RCCache=\qRCBookPage\q,CCOpen=
1,GCParameters=(Rows=0))">

                        </OBJECT>
-->
<SCRIPT LANGUAGE="JavaScript" RUNAT="server">
function _initrstCourses()
{
        var DBConn = Server.CreateObject('ADODB.Connection');
        DBConn.ConnectionTimeout =
Application('conSnider_ConnectionTimeout');
        DBConn.CommandTimeout =
Application('conSnider_CommandTimeout');
        DBConn.CursorLocation = Application('conSnider_CursorLocation');
```

```
                DBConn.Open(Application('conSnider_ConnectionString'),
Application('conSnider_RuntimeUserName'),
Application('conSnider_RuntimePassword'));
                var cmdTmp = Server.CreateObject('ADODB.Command');
                var rsTmp = Server.CreateObject('ADODB.Recordset');
                cmdTmp.ActiveConnection = DBConn;
                rsTmp.Source = cmdTmp;
                cmdTmp.CommandType = 1;
                cmdTmp.CommandTimeout = 30;
                cmdTmp.CommandText = 'SELECT coNum, coName, coSumFall,
coDistance, coID, coCost, coCredHour FROM tbCourses WHERE (coDistance = \'1\')';
                rsTmp.CacheSize = 100;
                rsTmp.CursorType = 3;
                rsTmp.CursorLocation = 3;
                rsTmp.LockType = 3;
                rstCourses.setRecordSource(rsTmp);
                rstCourses.open();
                if (thisPage.getState('pb_rstCourses') != null)
                        rstCourses.setBookmark(thisPage.getState('pb_rstCourses'));
        }
        function _rstCourses_ctor()
        {
                CreateRecordset('rstCourses', _initrstCourses, null);
        }
        function _rstCourses_dtor()
        {
                rstCourses._preserveState();
                thisPage.setState('pb_rstCourses', rstCourses.getBookmark());
        }
        </SCRIPT>

        <!--METADATA TYPE="DesignerControl" endspan-->
        </P>
        <P align=left>
        <!--METADATA TYPE="DesignerControl" startspan
        <OBJECT classid="clsid:9CF5D7C2-EC10-11D0-9862-0000F8027CA0"
id=rstStuCo
        style="LEFT: 0px; TOP: 0px">
                <PARAM NAME="ExtentX" VALUE="12197">
                <PARAM NAME="ExtentY" VALUE="2090">
                <PARAM NAME="State"
VALUE="(TCConn=\qconSnider\q,TCDBObject=\qDE\sCommands\q,TCDBObjectNa
me=\qtbStuCou\q,TCControlID_Unmatched=\qrstStuCo\q,TCPPConn=\qconSnider\q,R
CDBObject=\qRCDBObject\q,TCPPDBObject=\qDE\sCommands\q,TCPPDBObjectNa
me=\qtbStuCou\q,TCCursorType=\q3\s-\sStatic\q,TCCursorLocation=\q3\s-
\sUse\sclient-side\scursors\q,TCLockType=\q3\s-
```

```
\sOptimistic\q,TCCacheSize_Unmatched=\q100\q,TCCommTimeout_Unmatched=\q30\
q,CCPrepared=0,CCAllRecords=1,TCNRecords_Unmatched=\q10\q,TCODBCSyntax_U
nmatched=\q\q,TCHTargetPlatform=\q\q,TCHTargetBrowser_Unmatched=\qServer\s(A
SP)\q,TCTargetPlatform=\qInherit\sfrom\spage\q,RCCache=\qRCBookPage\q,CCOpen=
1,GCParameters=(Rows=0))">

                                </OBJECT>
        -->
        <SCRIPT LANGUAGE="JavaScript" RUNAT="server">
        function _initrstStuCo()
        {
                thisPage.createDE();
                var rsTmp = DE.Recordsets('tbStuCou');
                rstStuCo.setRecordSource(rsTmp);
                rstStuCo.open();
                if (thisPage.getState('pb_rstStuCo') != null)
                        rstStuCo.setBookmark(thisPage.getState('pb_rstStuCo'));
        }
        function _rstStuCo_ctor()
        {
                CreateRecordset('rstStuCo', _initrstStuCo, null);
        }
        function _rstStuCo_dtor()
        {
                rstStuCo._preserveState();
                thisPage.setState('pb_rstStuCo', rstStuCo.getBookmark());
        }
        </SCRIPT>

        <!--METADATA TYPE="DesignerControl" endspan-->
        </P>
        <P align=left>
        <!--METADATA TYPE="DesignerControl" startspan
        <OBJECT classid="clsid:B5F0E469-DC5F-11D0-9846-0000F8027CA0"
height=19 id=txtStuid
        style="HEIGHT: 19px; LEFT: 0px; TOP: 0px; WIDTH: 60px" width=60>
                <PARAM NAME="_ExtentX" VALUE="1588">
                <PARAM NAME="_ExtentY" VALUE="503">
                <PARAM NAME="id" VALUE="txtStuid">
                <PARAM NAME="ControlType" VALUE="0">
                <PARAM NAME="Lines" VALUE="3">
                <PARAM NAME="DataSource" VALUE="rstStuCo">
                <PARAM NAME="DataField" VALUE="stUCNum">
                <PARAM NAME="Enabled" VALUE="-1">
                <PARAM NAME="Visible" VALUE="-1">
                <PARAM NAME="MaxChars" VALUE="9">
```

```
            <PARAM NAME="DisplayWidth" VALUE="10">
            <PARAM NAME="Platform" VALUE="0">
            <PARAM NAME="LocalPath" VALUE="">


             </OBJECT>
        -->
        <!--#INCLUDE FILE="_ScriptLibrary/TextBox.ASP"-->
        <SCRIPT LANGUAGE=JavaScript RUNAT=Server>
        function _inittxtStuid()
        {
                txtStuid.setStyle(TXT_TEXTBOX);
                txtStuid.setDataSource(rstStuCo);
                txtStuid.setDataField('stUCNum');
                txtStuid.setMaxLength(9);
                txtStuid.setColumnCount(10);
        }
        function _txtStuid_ctor()
        {
                CreateTextbox('txtStuid', _inittxtStuid, null);
        }
        </script>
        <% txtStuid.display %>

        <!--METADATA TYPE="DesignerControl" endspan-->
        </P>
        <P align=left>
        <!--METADATA TYPE="DesignerControl" startspan
        <OBJECT classid="clsid:B5F0E469-DC5F-11D0-9846-0000F8027CA0"
id=txtFName
        style="LEFT: 0px; TOP: 0px">
                <PARAM NAME="_ExtentX" VALUE="3175">
                <PARAM NAME="_ExtentY" VALUE="503">
                <PARAM NAME="id" VALUE="txtFName">
                <PARAM NAME="ControlType" VALUE="0">
                <PARAM NAME="Lines" VALUE="3">
                <PARAM NAME="DataSource" VALUE="rstStud">
                <PARAM NAME="DataField" VALUE="stFName">
                <PARAM NAME="Enabled" VALUE="-1">
                <PARAM NAME="Visible" VALUE="-1">
                <PARAM NAME="MaxChars" VALUE="20">
                <PARAM NAME="DisplayWidth" VALUE="20">
                <PARAM NAME="Platform" VALUE="0">
                <PARAM NAME="LocalPath" VALUE="">
```

```
            </OBJECT>
        -->
        <SCRIPT LANGUAGE=JavaScript RUNAT=Server>
        function _inittxtFName()
        {
                txtFName.setStyle(TXT_TEXTBOX);
                txtFName.setDataSource(rstStud);
                txtFName.setDataField('stFName');
                txtFName.setMaxLength(20);
                txtFName.setColumnCount(20);
        }
        function _txtFName_ctor()
        {
                CreateTextbox('txtFName', _inittxtFName, null);
        }
        </script>
        <% txtFName.display %>

        <!--METADATA TYPE="DesignerControl" endspan-->
        </P>
        <P align=left>
        <!--METADATA TYPE="DesignerControl" startspan
        <OBJECT classid="clsid:B5F0E469-DC5F-11D0-9846-0000F8027CA0"
id=txtLName
        style="LEFT: 0px; TOP: 0px">
                <PARAM NAME="_ExtentX" VALUE="3175">
                <PARAM NAME="_ExtentY" VALUE="503">
                <PARAM NAME="id" VALUE="txtLName">
                <PARAM NAME="ControlType" VALUE="0">
                <PARAM NAME="Lines" VALUE="3">
                <PARAM NAME="DataSource" VALUE="rstStud">
                <PARAM NAME="DataField" VALUE="stLName">
                <PARAM NAME="Enabled" VALUE="-1">
                <PARAM NAME="Visible" VALUE="-1">
                <PARAM NAME="MaxChars" VALUE="20">
                <PARAM NAME="DisplayWidth" VALUE="20">
                <PARAM NAME="Platform" VALUE="0">
                <PARAM NAME="LocalPath" VALUE="">


                </OBJECT>
        -->
        <SCRIPT LANGUAGE=JavaScript RUNAT=Server>
        function _inittxtLName()
        {
```

```
            txtLName.setStyle(TXT_TEXTBOX);
            txtLName.setDataSource(rstStud);
            txtLName.setDataField('stLName');
            txtLName.setMaxLength(20);
            txtLName.setColumnCount(20);
        }
        function _txtLName_ctor()
        {
            CreateTextbox('txtLName', _inittxtLName, null);
        }
        </script>
        <% txtLName.display %>

        <!--METADATA TYPE="DesignerControl" endspan-->
        </P>

        <P align=left>
        <!--METADATA TYPE="DesignerControl" startspan
        <OBJECT classid="clsid:277FC3F2-E90F-11D0-B767-0000F81E081D"
height=137 id=Grid1
        style="HEIGHT: 135px; LEFT: 0px; TOP: 0px; WIDTH: 462px" width=462>
            <PARAM NAME="_ExtentX" VALUE="12224">
            <PARAM NAME="_ExtentY" VALUE="3572">
            <PARAM NAME="DataConnection" VALUE="">
            <PARAM NAME="SourceType" VALUE="">
            <PARAM NAME="Recordset" VALUE="rstCourses">
            <PARAM NAME="CtrlName" VALUE="Grid1">
            <PARAM NAME="UseAdvancedOnly" VALUE="0">
            <PARAM NAME="AdvAddToStyles" VALUE="-1">
            <PARAM NAME="AdvTableTag" VALUE="">
            <PARAM NAME="AdvHeaderRowTag" VALUE="">
            <PARAM NAME="AdvHeaderCellTag" VALUE="">
            <PARAM NAME="AdvDetailRowTag" VALUE="">
            <PARAM NAME="AdvDetailCellTag" VALUE="">
            <PARAM NAME="ScriptLanguage" VALUE="1">
            <PARAM NAME="ScriptingPlatform" VALUE="0">
            <PARAM NAME="EnableRowNav" VALUE="0">
            <PARAM NAME="HiliteColor" VALUE="">
            <PARAM NAME="RecNavBarHasNextButton" VALUE="-1">
            <PARAM NAME="RecNavBarHasPrevButton" VALUE="-1">
            <PARAM NAME="RecNavBarNextText" VALUE="  >  ">
            <PARAM NAME="RecNavBarPrevText" VALUE="  <  ">
            <PARAM NAME="ColumnsNames"
VALUE="'coID","coNum","coName","coCredHour"'>
            <PARAM NAME="columnIndex" VALUE="0,1,2,3">
            <PARAM NAME="displayWidth" VALUE="68,68,68,68">
```

```
<PARAM NAME="Coltype" VALUE="1,1,1,1">
<PARAM NAME="formated" VALUE="0,0,0,0">
<PARAM NAME="DisplayName" VALUE='"Course ID","Course
Number","Course Name","Credit Hours"'>
<PARAM NAME="DetailAlignment" VALUE=",,,">
<PARAM NAME="HeaderAlignment" VALUE=",,,">
<PARAM NAME="DetailBackColor" VALUE=",,,">
<PARAM NAME="HeaderBackColor" VALUE=",,,">
<PARAM NAME="HeaderFont" VALUE=",,,">
<PARAM NAME="HeaderFontColor" VALUE=",,,">
<PARAM NAME="HeaderFontSize" VALUE=",,,">
<PARAM NAME="HeaderFontStyle" VALUE=",,,">
<PARAM NAME="DetailFont" VALUE=",,,">
<PARAM NAME="DetailFontColor" VALUE=",,,">
<PARAM NAME="DetailFontSize" VALUE=",,,">
<PARAM NAME="DetailFontStyle" VALUE=",,,">
<PARAM NAME="ColumnCount" VALUE="4">
<PARAM NAME="CurStyle" VALUE="Basic Navy">
<PARAM NAME="TitleFont" VALUE="Arial">
<PARAM NAME="titleFontSize" VALUE="4">
<PARAM NAME="TitleFontColor" VALUE="16777215">
<PARAM NAME="TitleBackColor" VALUE="13416">
<PARAM NAME="TitleFontStyle" VALUE="1">
<PARAM NAME="TitleAlignment" VALUE="0">
<PARAM NAME="RowFont" VALUE="Arial">
<PARAM NAME="RowFontColor" VALUE="0">
<PARAM NAME="RowFontStyle" VALUE="0">
<PARAM NAME="RowFontSize" VALUE="2">
<PARAM NAME="RowBackColor" VALUE="16777215">
<PARAM NAME="RowAlignment" VALUE="0">
<PARAM NAME="HighlightColor3D" VALUE="12632256">
<PARAM NAME="ShadowColor3D" VALUE="268435455">
<PARAM NAME="PageSize" VALUE="5">
<PARAM NAME="MoveFirstCaption" VALUE="  |<  ">
<PARAM NAME="MoveLastCaption" VALUE="  >|  ">
<PARAM NAME="MovePrevCaption" VALUE="  <<  ">
<PARAM NAME="MoveNextCaption" VALUE="  >>  ">
<PARAM NAME="BorderSize" VALUE="0">
<PARAM NAME="BorderColor" VALUE="13421772">
<PARAM NAME="GridBackColor" VALUE="8421504">
<PARAM NAME="AltRowBckgnd" VALUE="12632256">
<PARAM NAME="CellSpacing" VALUE="0">
<PARAM NAME="WidthSelectionMode" VALUE="1">
<PARAM NAME="GridWidth" VALUE="462">
<PARAM NAME="EnablePaging" VALUE="0">
<PARAM NAME="ShowStatus" VALUE="-1">
```

```
        <PARAM NAME="StyleValue" VALUE="455657">
        <PARAM NAME="LocalPath" VALUE="">




    </OBJECT>
    -->
    <!--#INCLUDE FILE="_ScriptLibrary/Button.ASP"-->
    <!--#INCLUDE FILE="_ScriptLibrary/RSNavBar.ASP"-->
    <!--#INCLUDE FILE="_ScriptLibrary/DataGrid.ASP"-->
    <SCRIPT LANGUAGE=JavaScript RUNAT=Server>
    function _initGrid1()
    {
    Grid1.pageSize = 0;
    Grid1.setDataSource(rstCourses);
    Grid1.tableAttributes = ' cellpadding=2  cellspacing=0 bordercolorlight=Silver
bgcolor=Gray border=0 cols=4 rules=ROWS WIDTH=462';
    Grid1.headerAttributes = '   bgcolor=#003468 align=Left';
    Grid1.headerWidth[0] = ' WIDTH=68';
    Grid1.headerWidth[1] = ' WIDTH=68';
    Grid1.headerWidth[2] = ' WIDTH=68';
    Grid1.headerWidth[3] = ' WIDTH=68';
    Grid1.headerFormat = '<Font face="Arial" size=4 color=White> <b>';
    Grid1.colHeader[0] = '\'Course ID\";
    Grid1.colHeader[1] = '\'Course Number\";
    Grid1.colHeader[2] = '\'Course Name\";
    Grid1.colHeader[3] = '\'Credit Hours\";
    Grid1.rowAttributes[0] = '  bgcolor = White align=Left bordercolorlight=Silver';
    Grid1.rowAttributes[1] = '  bgcolor = Silver align=Left bordercolorlight=Silver';
    Grid1.rowFormat[0] = ' <Font face="Arial" size=2 color=Black >';
    Grid1.colAttributes[0] = '  WIDTH=68';
    Grid1.colFormat[0] = '<Font Size=2 Face="Arial" Color=Black >';
    Grid1.colData[0] = 'rstCourses.fields.getValue(\'coID\')';
    Grid1.colAttributes[1] = '  WIDTH=68';
    Grid1.colFormat[1] = '<Font Size=2 Face="Arial" Color=Black >';
    Grid1.colData[1] = 'rstCourses.fields.getValue(\'coNum\')';
    Grid1.colAttributes[2] = '  WIDTH=68';
    Grid1.colFormat[2] = '<Font Size=2 Face="Arial" Color=Black >';
    Grid1.colData[2] = 'rstCourses.fields.getValue(\'coName\')';
    Grid1.colAttributes[3] = '  WIDTH=68';
    Grid1.colFormat[3] = '<Font Size=2 Face="Arial" Color=Black >';
    Grid1.colData[3] = 'rstCourses.fields.getValue(\'coCredHour\')';
```

```
}
function _Grid1_ctor()
{
        CreateDataGrid('Grid1',_initGrid1);
}
</SCRIPT>

<%      Grid1.display %>


<!--METADATA TYPE="DesignerControl" endspan-->
</P>

<P align=left>
</P>

<P align=left>
<!--METADATA TYPE="DesignerControl" startspan
<OBJECT classid="clsid:B5F0E469-DC5F-11D0-9846-0000F8027CA0"
height=19 id=txtcoID
style="HEIGHT: 19px; LEFT: 0px; TOP: 0px; WIDTH: 60px" width=60>
        <PARAM NAME="_ExtentX" VALUE="1588">
        <PARAM NAME="_ExtentY" VALUE="503">
        <PARAM NAME="id" VALUE="txtcoID">
        <PARAM NAME="ControlType" VALUE="0">
        <PARAM NAME="Lines" VALUE="3">
        <PARAM NAME="DataSource" VALUE="rstStuCo">
        <PARAM NAME="DataField" VALUE="coID">
        <PARAM NAME="Enabled" VALUE="-1">
        <PARAM NAME="Visible" VALUE="-1">
        <PARAM NAME="MaxChars" VALUE="3">
        <PARAM NAME="DisplayWidth" VALUE="10">
        <PARAM NAME="Platform" VALUE="0">
        <PARAM NAME="LocalPath" VALUE="">


        </OBJECT>
-->
<SCRIPT LANGUAGE=JavaScript RUNAT=Server>
function _inittxtcoID()
{
        txtcoID.setStyle(TXT_TEXTBOX);
        txtcoID.setDataSource(rstStuCo);
        txtcoID.setDataField('coID');
        txtcoID.setMaxLength(3);
        txtcoID.setColumnCount(10);
```

```
}
function _txtcoID_ctor()
{
        CreateTextbox('txtcoID', _inittxtcoID, null);
}
</script>
<% txtcoID.display %>

<!--METADATA TYPE="DesignerControl" endspan-->
</P>

<P align=left>
<!--METADATA TYPE="DesignerControl" startspan
<OBJECT classid="clsid:B6FC3A14-F837-11D0-9CC8-006008058731"
height=27
id=btnSubmit style="HEIGHT: 27px; LEFT: 0px; TOP: 0px; WIDTH: 63px"
width=63>
        <PARAM NAME="_ExtentX" VALUE="1667">
        <PARAM NAME="_ExtentY" VALUE="714">
        <PARAM NAME="id" VALUE="btnSubmit">
        <PARAM NAME="Caption" VALUE="Submit">
        <PARAM NAME="Image" VALUE="">
        <PARAM NAME="AltText" VALUE="">
        <PARAM NAME="Visible" VALUE="-1">
        <PARAM NAME="Platform" VALUE="0">
        <PARAM NAME="LocalPath" VALUE="">
                                                        </OBJECT>
-->
<SCRIPT LANGUAGE=JavaScript RUNAT=Server>
function _initbtnSubmit()
{
        btnSubmit.value = 'Submit';
        btnSubmit.setStyle(0);
}
function _btnSubmit_ctor()
{
        CreateButton('btnSubmit', _initbtnSubmit, null);
}
</script>
<% btnSubmit.display %>

<!--METADATA TYPE="DesignerControl" endspan-->
    
<!--METADATA TYPE="DesignerControl" startspan
```

```
        <OBJECT classid="clsid:B6FC3A14-F837-11D0-9CC8-006008058731"
height=27 id=btnDone
        style="HEIGHT: 27px; LEFT: 0px; TOP: 0px; WIDTH: 53px" width=53>
                <PARAM NAME="_ExtentX" VALUE="1402">
                <PARAM NAME="_ExtentY" VALUE="714">
                <PARAM NAME="id" VALUE="btnDone">
                <PARAM NAME="Caption" VALUE="Done">
                <PARAM NAME="Image" VALUE="">
                <PARAM NAME="AltText" VALUE="">
                <PARAM NAME="Visible" VALUE="-1">
                <PARAM NAME="Platform" VALUE="0">
                <PARAM NAME="LocalPath" VALUE="">

                                                                </OBJECT>

        -->
        <SCRIPT LANGUAGE=JavaScript RUNAT=Server>
        function _initbtnDone()
        {
                btnDone.value = 'Done';
                btnDone.setStyle(0);
        }
        function _btnDone_ctor()
        {
                CreateButton('btnDone', _initbtnDone, null);
        }
        </script>
        <% btnDone.display %>

        <!--METADATA TYPE="DesignerControl" endspan-->
        </P>
        </body>
        <% ' VI 6.0 Scripting Object Model Enabled %>
        <% EndPageProcessing() %>
        </FORM>
        </html>
```

References

1. Willis, Barry. "Distance Education: An Overview." http://www.uidaho.edu/evo/dist1.htm . October 1995.

2. Macromedia. "Macromedia Flash and Shockwave Optimized." http://macromedia.com/macromedia/proom/pr/1999/intel.html_. February 17, 1999

3. Karelis, Charles. "Education Technology and Cost Control: Four Models." http://ed.gov/offices/OPE/FIPSE/LAAP/reading.html_. February 1999.

4. Microsoft. "TechNet for Education" http://www.microsoft.com/education/technet/default.asp. January 2000.

5. Microsoft. "Windows NT Server 4.0." http://www.microsoft.com/ntserver. January 2000

6. Microsoft. "Windows Netmeeting" http://www.microsoft.com/netmeeting. January 2000

7. Rush, Laura. "The Need For Speed." http://ecommerce.internet.com/opinions/article/0,1281,3551_315861,00.html March 6, 2000.

8. MyWay.com. "E-commerce Portal Web-Page" http://ecommerce.myway.com/features/home/default.asp, 2000

9. WebCT.com. "The E-Learning Hub". http://www.webct.com , 2000

10. Whatis.com. "E-Commerce." http://whatis.com/ecommerc.htm, 2000

11. National Center for Education Statistics. "Distance Education in Higher Education Institutions" http://nces.ed.gov/pubs98/distance/index.html, October 1997

12. CONAHEC. "Consortium for North American Higher Education Collaboration" http://conahec.org/english. October 1999.

13. Hirumi, A.. *Virtual Information, Training and Support Center for Distance Educators: A Concept Paper*. Manuscript from the Instructional Technology Program at the University of Houston--Clear Lake. 1997

14. Hirumi, A., and Bermudez, A.. *Interactivity, Distance Education, & Instructional Systems Design Converge on the Super Information Highway.* Journal of Research on Computing in Education, 24(1), 1-16. 1996

15. Holmberg, B. . *Theory and Practice of Distance Education.* London: Routledge. 1989

16. Moore, M. G., & Kearsley, G.. *DistanceEeducation: A SystemsView.* Belmont, CA. Wadsworth Publishing Co. 1996

17. Garrison, D. R.. *A Cognitive Constructivist View of Distance Education: An Analysis of Teaching -Learning assumptions.* Distance Education, 14(2), 199-211. 1993

Project Expert Resources

Dr. Vladimir Uskov, programming

Dr. Ashaf Saad, networking

Dr. Sam Geonetta, web page design, education considerations

Mr. Robert Schlemmer, databases

Mr. Mike Ames, hardware

Mr. Paul Schwab, local network

Mr. Jason Stroh, local network