I dedicate this project to my mother, Helen, whose encouragement, support and unconditional love has made it possible for me to successfully complete this program.
ABSTRACT

This software project is the design and implementation of an automated office system for the College of Science and Technology at Texas A&M University-Corpus Christi. The Internet-based pilot application is called the Electronic Document Management System (EDMS) and provides shared information for the personnel in the College of Science and Technology. Documents and records providing grant information, schedule information, and faculty information may be read and edited in this "file-sharing" environment according to user access privileges. An important feature of this system is the Internet-based access control to this centralized repository of shared documents and records.

The EDMS application has been developed primarily using Web-based Distributed Authoring and Versioning (WebDAV), a Web-based protocol to organize and store the shared documents. The Lightweight Directory Access Protocol (LDAP), another Internet-based protocol, has been used to support this document management system by providing user authentication in the access control scheme. The LDAP subsystem works in tandem with .htaccess files, generated by PHP Hypertext Preprocessor (PHP) scripts, to manage the shared documents. The majority of the design and implementation of the deliverable EDMS application involved the successful integration of the various components needed to develop the overall system.
# TABLE OF CONTENTS

ABSTRACT ............................................................................................................................... i
LIST OF FIGURES ..................................................................................................................... iv

1. BACKGROUND AND RATIONALE .................................................................................. 1
   1.1 Client Background ......................................................................................................... 1
   1.2 The Necessity for an EDMS ........................................................................................ 1
   1.3 Project Goals .............................................................................................................. 3

2. ELECTRONIC DOCUMENT MANAGEMENT SYSTEM .................................................. 4
   2.1 Overview ..................................................................................................................... 4
   2.2 Functions of the EDMS .............................................................................................. 7
   2.3 The Desktop User Interface ........................................................................................ 7
       2.3.1 Accessing the Desktop Interface ........................................................................ 7
       2.3.2 Submitting Documents to the EDMS as Course Syllabi .................................... 12
       2.3.3 Modifying Documents from the EDMS as Faculty Handbook ......................... 13
       2.3.4 Editing Documents in Dav-Aware Programs as MS-Excel ................................ 13
   2.4 The Web Interface ...................................................................................................... 14
       2.4.1 Main Index Page .................................................................................................. 15
       2.4.2 The Eleven Web Folder Directories .................................................................. 15
       2.4.3 Indexed Administration Page ............................................................................ 16
       2.4.4 "Edit Directory Permissions" Form ...................................................................... 16
       2.4.5 "Add or Edit User" Form .................................................................................... 18
       2.4.6 "User Administration" Form .............................................................................. 19
       2.4.7 "Personnel Update" Form .................................................................................. 20
       2.4.8 Personnel Records ............................................................................................. 23
       2.4.9 Telephone/Address List Index Page ................................................................... 25
       2.4.10 Grant Information Documents Viewed from the Web ...................................... 26

3. ENVIRONMENT .................................................................................................................. 27
   3.1 The Client-Server Architecture .................................................................................. 27
       3.1.1 The Server Side .................................................................................................... 27
       3.1.2 The Client Side .................................................................................................... 28
   3.2 The Protocols ............................................................................................................... 28

4. SYSTEM DESIGN .............................................................................................................. 30
   4.1 Procedural Overview ................................................................................................... 30
   4.2 Gathering System Requirements ............................................................................... 30
   4.3 Determining the Requirements List .......................................................................... 31
       4.3.1 Application Overview and Requirements Analysis ........................................... 31
       4.3.2 Functional Requirements List .............................................................................. 32
# TABLE OF CONTENTS (CONT.)

4.4 Designing System Operations.................................................................33
   4.4.1 Component Integration ........................................................................33
   4.4.2 Architectural Design for the System .....................................................34
      4.4.2.1 Overview of the Functional Design of the EDMS .............................34
      4.4.2.2 Data Integrity and Security Issues ...............................................36
      4.4.2.3 Data Integrity ..............................................................................38
      4.4.2.4 Security ......................................................................................38
      4.4.2.5 Authorization .............................................................................38
      4.4.2.6 Authentication ............................................................................41
   4.4.3 Interface Design ..................................................................................43
      4.4.3.1 Document Entry ..........................................................................44
      4.4.3.2 Document Retrieval .................................................................45
      4.4.3.3 Data Entry ..................................................................................45
      4.4.3.4 Data Retrieval ............................................................................48
   4.4.4 Database and File Specifications .........................................................49
      4.4.4.1 The Directory Information Tree ....................................................49
      4.4.4.2 File Specifications ....................................................................51
   4.4.5 Design of the PHP Scripts ..................................................................51

5. RESULTS .................................................................................................53
   5.1 Nonfunctional Requirements .................................................................54
      5.1.1 Availability ....................................................................................54
      5.1.2 Maintainability ..............................................................................54
      5.1.3 Data Integrity ...............................................................................55
      5.1.4 Development Cost .........................................................................55
      5.1.5 Delivery Date ...............................................................................56
      5.1.6 Extensibility ..................................................................................56
      5.1.7 Flexibility .....................................................................................56
      5.1.8 Installability ..................................................................................56
      5.1.9 Leveragability/Reuse .....................................................................57
      5.1.10 Operability ..................................................................................57
      5.1.11 Performance ................................................................................57
      5.1.12 Portability ....................................................................................58
      5.1.13 Robustness ..................................................................................59
      5.1.14 Scalability ....................................................................................59
   5.2 Social Impact .........................................................................................60
   5.3 Conclusion ............................................................................................61
   5.4 Future Work .........................................................................................62

REFERENCES ...............................................................................................64
LIST OF FIGURES

Figure 2-1  System overview of the EDMS .................................................................6
Figure 2-2  Web Folders icon displayed from “My Computer” .........................8
Figure 2-3  Adding the EDMS Web Folder ...........................................................9
Figure 2-4  EDMS directories structure ...............................................................10
Figure 2-5  Web user interface map of EDMS pilot ............................................14
Figure 2-6  “Edit Directory Permissions” form .......................................................17
Figure 2-7  “Add or Edit New User” form .............................................................18
Figure 2-8  “User Administration” form ...............................................................19
Figure 2-9  “Personnel Update” form .................................................................21
Figure 2-10 “Faculty Information” section on the “Personnel Update” form ......22
Figure 2-11 “Faculty Records” index has tenure status information ...............23
Figure 2-12 Complete view of faculty member’s record ...................................24
Figure 4-1  Overview of network model for the EDMS pilot .........................35
Figure 4-2  Table of PHP scripts .........................................................................37
Figure 4-3  Example of attribute name-value pair for entry in the DIT ............46
Figure 4-4  Data organization in the EDMS LDAP DIT .....................................50
1. BACKGROUND and RATIONALE

1.1 Client Background

In a similar manner to several colleges on campus, the office suite of the Dean of Science and Technology at Texas A&M University - Corpus Christi was operating inefficiently via a "sneakernetting" system. The main reason for this was that a central repository for shared information did not exist. Each individual working in this office had access to a PC, but no simple-to-use file-sharing environment. Thus, the daily office operations involved the passing of hard copy documents between co-workers, other staff and faculty members in the College of Science and Technology. Data entry of organizational personnel information was a redundant process, and there was little, if any, consistency or standardization of organizational processes. Hence, the capability of the computing environment already in place was barely being tapped. The College of Science and Technology represents a smaller example of some of the document-sharing challenges university-wide.

1.2 The Necessity for an Electronic Document Management System

After conducting a series of preliminary interviews with members of the staff at the Dean's office, many problem areas were discovered. The first problem entailed duplicate data entry. For instance, data was extracted from other data by opening a document on the PC. The same data was then printed in hard copy, passed along to a fellow co-worker, and re-entered into a spreadsheet on a different PC. There existed no intranet for file-sharing or Internet-based document filing system. This sort of
disorganization resulted in extra work, inconvenience, and redundancy of data. The deliverable EDMS alleviates these problems and aids the Dean in her decision-making processes.

The second problem area involved little, if any, digital access between the Dean's office and the staff and faculty in the College of Science and Technology, other than a telephone system. Organizing the data from the Dean's office suite and presenting it in an Internet-based document filing system easily accessible by all staff and faculty reduces the inefficiency in the daily operations internal to the college.

The primary requirement of the client was that faculty contact and personal information (e.g., name, telephone number, e-mail addresses, academic degree, full-time or adjunct status, past and current classes taught, past and current syllabi, tenure-track status, curricula vitae, current publications, seminars attended, training sessions) be made available from some database system. Then, personnel could access this information according to security levels.

Another concern involved placing the faculty handbook on-line. Similar to several other document management processes, such as the course syllabi process, the old faculty handbook document management process entailed the creation of the document in a different format from the distribution of the document. For example, the faculty handbook was generated in electronic format, only to be distributed in hard-copy format. It soon became apparent in the design phase of this project that it
would be more convenient to keep documents in electronic format, and directly place the electronically created documents in a central on-line location for easy accessibility.

Further meetings with the client revealed a highly anticipated need to convert from the archaic "paper-shuffling" office system to a more advanced time-saving system. It was determined that human time was an expensive commodity that was being wasted due to the old inefficient and outdated system. It was also determined that information should be entered only once into the system and not removed from the system for modifications or updates.

1.3 Project Goals

The main goal of this project was to provide an automated office system that would enable the staff and administration at the Dean's office and the staff and faculty of the College of Science and Technology to operate more efficiently via the use of an intranet system. A more encompassing goal was to pilot a system that could reveal the possibilities of what could be done for the entire campus. The architecture of the EDMS is generic so that it may be implemented campus-wide. It was designed to allow for expandability, in the hopes that future graduate students would be able to readily use it for future projects. This system was piloted with a committee minutes information system, a course syllabi information system, a grants information system, a personnel-records information system, a policy information system, and a schedule information system for the College of Science and Technology.
2. ELECTRONIC DOCUMENT MANAGEMENT SYSTEM

2.1 Overview

The primary goal of this automated office system was to provide a simple-to-use tool for electronically centralizing and organizing document storage. The EDMS allows users to share stored files via the World Wide Web, and to manipulate these same files via the desktop. The EDMS also prevents general users, such as students, from accessing confidential information and from modifying or deleting the publicly accessed shared files.

The overall architecture of the EDMS consists of a large umbrella structure to accommodate the various sub-systems. The pilot for the EDMS provides users with documents such as academic calendars, committee minutes, course syllabi, the Faculty Handbook, grant records such as project establishment notices and summaries of proposal budgets, personnel records including faculty curricula vitae and other related documents. The pilot was constructed in a fashion to allow for expandability within the structure so that other documents can be added after the completion of this project should the need arise.
Figure 2-1 is a system overview showing the general flow of documents and records in the EDMS system. The following list helps define the flow in the diagram:

1. The personnel users in the three departments in the College of Science and Technology enter all requested documents into their computers via a word-processor of their choice. These electronic documents are preferably saved to a local drive as either a Portable Document Format (PDF) or a HyperText Markup Language (HTML) file.

2. The personnel users in the three departments may also enter requested documents via the Hewlett Packard HP-9100 scanner.

3. These electronic documents may be dragged and dropped into the appropriate Microsoft Web Folder accessible from the desktop to enter the document into the EDMS.

4. The Microsoft Web Folder automatically uploads these electronic documents to the Web-based Distributed Authoring and Versioning (WebDAV) server where the organized documents will be stored.

5. These documents are made immediately available at this project’s Web site for general public viewing, depending on the user’s access rights.

6. The reverse process occurs for retrieving documents from the WebDAV server, if the user determines that modifications are necessary or if complete deletion of the file is necessary. If the program is “WebDAV-aware”, such as Microsoft Excel, the document may be opened up directly from the Web.

7. Sub-folders containing old documents may be dragged and dropped into the appropriate archive Web folder.
Figure 2-1. System overview of the EDMS.
2.2 Functions of the Electronic Document-Management System

Specifically, after authorization of the user, this system allows the user to:

- enter and organize shared documents in the EDMS,
- access, retrieve, review, modify, and delete shared documents in protected directories in the EDMS,
- quickly archive documents in the EDMS,
- add users and their information to the EDMS, and
- retrieve and update the user information stored in the EDMS.

2.3 The Desktop User Interface

As mentioned previously, there are two different user interfaces. A desktop interface and an Internet interface may be accessed from any computer station with Internet access. However, an authentication window will appear on the screen when attempts are made to access protected directories.

2.3.1 Accessing the Desktop Interface

To access the EDMS from the Windows desktop, the user simply clicks on the "My Computer" icon or "Windows Explorer" icon, and then on the "Web Folder" icon. An example of this process is illustrated in Figure 2-2.
Next, the user adds an EDMS Web Folder if one is not already in place, by clicking on the "Add Web Folder" wizard. A pop-up message window appears and asks the user to type the location to be added (see Figure 2-3).

Figure 2-2. Web Folders icon displayed from “My Computer.”
As of this writing, the location to be added for the EDMS is
"http://sci.tamucc.edu:800/edms/davhome." Another pop-up message window appears
by clicking the "Next" button. The user can name this Web Folder by
typing in "EDMS" and clicking the "Finish" button. A new Web Folder icon should
appear on the screen referring to the EDMS directory. All sub-directories accessible to
the user can be seen by clicking on this EDMS directory icon.

Figure 2-3. Adding the EDMS Web Folder.
The desktop interface is composed of folder icons representing directories which are used to handle the entry, retrieval, modification, deletion and archival of the shared documents as seen below in Figure 2-4. Similar to the appearance and behavior of a typical Windows 95-style folder documents are dragged-and-dropped into and out of the appropriate Web folder. However, files must first be copied to a local drive before applications can use them, unless the application is “WebDAV-aware”, in which case the file may be opened directly from the application.

Figure 2-4. EDMS directories structure.
This means that documents can be created using any word-processor or word-editing program of choice, saved as PDF or HTML documents on a local drive and then dropped in the Web Folder icon. Similar to File Transfer Protocol (FTP) or other file servers, permissions have been set in order to deal with security issues that arise from being able to write as well as read shared files on the Web.

The EDMS deliverable application has eleven main directories to house information for the College of Science and Technology: the “Academic Calendar” directory, the “Committee Minutes” directory, the “Course Syllabi” directory, the “Curricula Vitae” directory, the “Faculty Handbook” directory, the “Grant Information” directory, the “HELP” directory, the “SCI” directory and the three departmental directories, “CAMS”, “PALS” and “SONHS.”

The last three directories are private directories for each department in the College of Science and Technology and are intended to be used to post the minutes of private departmental meetings and other private information pertinent to that department. The “SCI” directory is for the use of the administration personnel from the Dean’s office suite to post policies and pertinent information to all departments in the college. The rest of the directories store the documents referred to in the directory name.
2.3.2 Submitting Documents to the EDMS such as Course Syllabi

Following the current campus registration guide, the “Course Syllabi” directory is subdivided by year, semester, department and departmental division. This layout allows for simple archiving of the course syllabi. To submit a course syllabus to the EDMS, as with other documents, the user takes the following steps:

1. The faculty member prepares the course syllabus document.

2. The faculty member saves the course syllabus document as a PDF or HTML file on the local drive. The Microsoft 98 Word editor has a convenient “save as HTML” selection from the “File” drop-down menu.

3. The faculty member closes the newly saved PDF or HTML document.

4. The faculty member opens the appropriate Web Folder from the desktop currently labeled “Course Syllabi”.

5. The faculty member locates the appropriate sub-folder.

6. The faculty member locates the newly saved document in a separate window on the local drive. For example, Windows Explorer may be used to locate the document by opening a separate Windows Explorer window.

7. The faculty member selects the closed document with the mouse and drags it into the appropriate Web Folder. This process is identical to moving files in and out of regular folders on the desktop. The only difference is the use of a Web Folder for making the drop instead of a folder on the local hard drive.
2.3.3 Modifying Documents from the EDMS such as the Faculty Handbook

The “Academic Calendar” and “Faculty Handbook” are examples of directories that already contain documents that may need to be modified. To modify the academic calendar document stored in the “Academic Calendar” directory or one of the Faculty Handbook documents stored in the “Faculty Handbook” directory, the Executive Secretary at the Dean's office can do the following:

1. Select the appropriate directory from the Web folder desktop interface with the mouse.

2. Drag and drop the desired document to a local folder on the desktop.

3. Open the desired document in the word-processor of choice.

4. Edit the document.

5. Save changes to the document as a PDF or HTML file.


7. Drag and drop the documents (word processor and PDF or HTML copy) back into the directory from which they have been retrieved.

2.3.4 Editing Documents in Data-Aware Programs such as Microsoft Excel

Spreadsheets such as the “Summary of Grant Budget.xls” are found in the “Summary Proposal Budget” Web Folder directory. This Web Folder is a subfolder of the Grant Information directory and may be opened directly from the Web interface. Such documents are initially created in Microsoft Excel’s program and saved as an HTML file. A “save as HTML” option is conveniently available from the “File” drop down menu located in the upper left hand corner. Once saved, the document is automatically
stored in its original location. Thus, no drag and drop is necessary after modifying documents in the Microsoft Excel program.

2.4 The Web Interface

The Web interface is used to handle the convenient viewing of the stored documents from any computer station and any location (on-campus or off-campus). It is also used to enter, update and retrieve personnel information in a directory. Additionally, the administrative assistant to the Dean uses the Web interface to set the permissions on the EDMS directories mentioned in section 2.3.1. Figure 2-5, the functional organization of the EDMS pilot, shows the navigation through the Web site from the user’s perspective.

Figure 2-5. Web user interface map of EDMS pilot.
2.4.1 Main Index Page

This is the main Web page of the EDMS application. For the pilot of this particular project, this main page represents the College of Science and Technology. The indexed selection on the left-hand side of the page is available on all subsequent pages to quickly guide the user to the correct destination. The user may select any of the eleven Web Folder directories listed in section 2.3.1 (except for the Curricula Vitae directory). The index pages for “Faculty & Staff,” and “Administration” and the "Telephone/Address Directory" may also be selected from this menu navigation bar.

2.4.2 The Eleven Web Folder Directories

The eleven Web Folder directories (refer to section 2.3.1) share the same format. These directories and sub-directories have a folder icon to the left-hand side of the linked directory name. Clicking on these directories takes the user to any sub-directories and to any documents stored in the directories. The documents have a document icon on the left-hand side of the linked document name.

In addition to the eleven Web Folder directories, there is an indexed administration page. Selecting this page provides the administrative user with various administrative functions such as setting access rights on these eleven directories.
2.4.3 The Indexed Administration Page

The indexed administration page offers the user a menu of two major options:

1. Personnel Records - contains all of the information on the users of the EDMS. Administrative staff may access these personnel records.

2. Top Level Administrators - allows the administrative assistant to the Dean to perform various administrative duties on the system. A new user may be added to the system. Once a new user has been added to the EDMS, the new user needs to be added to one of two user groups for the EDMS: administrative staff or faculty. Users in the top-level administrators group may only be added to this group by the system administrator. The administrative assistant to the Dean may also set and reset permissions on all of the Web Folder directories. Faculty curricula vitae are stored in the designated "Curricula Vitae" Web Folder directory.

Only authorized users are able to gain access to the personnel records and the top-level administrative functions. It is recommended that options to administer the system be made available only to the administrative assistant to the Dean.

2.4.4 The “Edit Directory Permissions” Form

The “Edit Directory Permissions” form is displayed when “Set/Reset Permissions” is selected from the indexed "Administration" Page. This form allows the administrative assistant user to set permissions on the Web Folder directories (refer to Figure 2-6). The directory to have permissions set is selected from the drop-down menu. After a directory selection is made from this menu, the user checks at least one group to
designate write permissions on this directory, and then at least one group to have read permissions on this directory. If the user leaves blank all of the groups in either read or write and submits the form, a message asks the user to enter at least one group from the read option and at least one group from the write option to ensure that the .htaccess file is overwritten correctly. A successful submission of the form changes the selected directory’s read and write permissions as determined by the administrative assistant user and displays a confirmation message.

![Image of permission settings form]

**Figure 2-6**

"Edit Directory Permissions" form enables user to set Web Folder directory permissions.
2.4.5 The "Add or Edit New User" Form

This short "Add or Edit New User" administration form, shown below in Figure 2-7, is made available to the administrative assistant from the indexed "Administration" Page. The administrative assistant types in the user identification of the person to be added to the system. This should match the username that the person uses on other campus systems. For example, the name "Patrick Michaud" is entered as "pmichaud."

![Add User](image_url)

Figure 2-7.

"Add or Edit New User" form allows a new user to be added to EDMS.
2.4.6 The "User Administration" Form

The "User Administration" form is a shortened version of the form described in the following section and is displayed after the administrative assistant submits the "Add or Edit New User" form described in the previous section. The purpose of this form, shown in Figure 2-8, is to add new employees to the system and assign passwords for the new employee. If a person is already in the EDMS and the administrative assistant only wishes to edit information on the person, the two password fields at the bottom of the form are not displayed.

![LDAP User Administration Form](image)

**Figure 2-8.** "User Administration" form is used to enter new user into system.
2.4.7 "Personnel Update" Forms

After being added to the system and issued a password, an employee may add or update personal information. Unless specified in the group of "EDMS-Administrators", employees may only add or update information pertaining to themselves.

Activating the "Personnel Update Forms" link from the indexed "Faculty & Staff" Page, allows the faculty or staff member to select from a list of personnel in the college. Once an employee has been added to the system, the employee's name will appear in this list of personnel. The interactive data-entry form is available after the faculty or staff member selects one of the personnel links from the list.

This long self-explanatory form consists of the following four sections: contact information, employment information, faculty information and other work information. The faculty or staff member enters information such as name, titles, hire date, tenure status, tenure eligibility date, current publications, recent-seminars attended, recent-training-sessions attended, involvement with student activities, consulting jobs and number of consulting days taken in a given year by typing in the information in the available input boxes.
Figure 2-9.

"Personnel Update" form allows user to add and edit personal information.

Authorized users of this form are the personnel in the College of Science and Technology. The faculty or staff member may add new information or edit previously stored information. An example of how to fill out the form and the correct formatting of numerical data entry (particularly telephone numbers) is available for the faculty or staff member to reference on the right-hand side of the form (see Figures 2-9 and 2-10). This interactive form allows the faculty or staff member to enter and update their own personal information. If the person is already in the EDMS any previously entered and stored information attributed to this person appears in the input boxes in the form.
It is suggested on the form that staff personnel fill out the first two sections and then submit the form, as seen below in Figure 2-10. The last two sections of the form are intended for faculty-specific information. Upon completion of this form, the new information is submitted and the user is automatically returned to the indexed "Faculty & Staff" Page. The submitted information is immediately posted for viewing from any of the pages containing personnel records.

Figure 2-10.

"Faculty Information" section on the "Personnel Update" form.
2.4.8 Personnel Records

These records are displayed at various locations at the EDMS Web-site depending on the desired view of the information. For example, an indexed list of faculty is displayed by selecting the "Faculty Records" link from the indexed "Administration" Page (Figure 2-11). This partial view of faculty information displays a list of all faculty in the college as well as important pieces of information attributed to the faculty member such as tenure status, department and telephone number. Selecting a faculty member from this page displays the selected faculty member's complete record (see Figure 2-12).

![Faculty Records Page](image)

Figure 2-11. "Faculty Records" index has tenure status information.
Patrick R. Michaud

Contact Information:
Name: Patrick R. Michaud
Department: CAMS
Classification: Faculty
Office Location: CI 323
Mailing Address: Texas A & M University-Corpus Christi, 6300 Ocean Drive, Corpus Christi, Texas 78412
Telephone: +1 361-825-2751
Fax: +1 361-825-5704
Email: pmmichau@tamucc.edu

Employment Information:
Classification: Faculty
Job Title: Professor, Director
Hired Date: 11-1993
Termination Date:

Other Employment:
Number of Days Taken: 7
1996-1999—Corpus Christi Caller Times

Faculty Information:
Professional Title: Ph.D.
Academic Rank: Professor
Tenure Status: Tenured
Tenure Eligibility Date: 8-1993
Date Tenured: 8-1993
Instructor Qualifications:
Discipline Credit Hours: N/A
Recent Publications:

Seminars/Conferences Attended: 4-2000—Presenter, Consortium for Computing at Small Colleges
10-2000—Attendee, South Texas Environmental Conference
11-2000—Presenter, Emergency Management Conference
Training Sessions: 4-1997—FAMIS Training
Student Activities Involvement: 5-1990—Advisor, Computer Science Club

Figure 2-12. Complete view of faculty member's record.
The faculty records consist of the following fields: name, academic rank, highest-degree earned, tenure track, tenure date, instructor qualifications, office location, telephone number, e-mail address, current publications, recent seminars attended, recent training sessions attended, recent consulting jobs, and current number of consulting days taken in given year. This record contains the same information entered in the personnel update form.

Personnel information is displayed by selecting "Personnel Update Forms" from the indexed "Faculty & Staff" Page followed by the selection of the desired department. The "Telephone/Address List" links to a bird's-eye-view of the personnel record from the selected personnel in the directory listing.

2.4.9 The Telephone/Address List Index Page

There are four basic options on the "Telephone Directory" Page. The user may select to view the telephone listings by department or by conducting a general search. The default general telephone directory provides the user with a complete listing of all the personnel in the college, alphabetized by last name. Four other options are provided at the top of each directory page: "CAMS", "PALS", "SONHS" and "Departmental." Selecting the "Departmental Telephone Directory" displays the same complete alphabetized information as the general telephone directory but categorizes the listing by department. The telephone directory listings are formatted to provide the user with telephone numbers, e-mail addresses, office locations and alternate telephone numbers.
for the personnel in the college. Selecting a linked name from the list displays all contact information attributed to the selected person's record.

2.4.10 Grant Information Documents Viewed from the Web

These documents are intended primarily for the Dean’s use and include such documents as the letter of notification, grant narrative or abstract, grant amount, project establishment notice, list of advisory committees, advisement teams and grant management-committees.

The Summary of Proposals Budget document was created in Microsoft Excel and is not intended to provide complete grant information. It provides the Dean with a concise snapshot or bird’s-eye view of grant information needed for pending decisions and includes the following fields: grant title, agency, principle investigator, notification date, effective date, duration, amount per year. The actual grant documents, project establishment notices, letters of notification, and other requested documents related to the grant are displayed in the “Grant Information” sub-directories.
3. ENVIRONMENT

3.1 The Client-Server Architecture

The EDMS was developed in a Unix environment using a client-server architecture. All protocols used are platform independent.

3.1.1 The Server Side

Specifically, the server software for the pilot of this project is running on Penguin, a server belonging to the College of Science and Technology at Texas A&M University – Corpus Christi. This machine is currently running the Linux operating system (RedHat version 7.0). The server side of the EDMS has been developed and is running on Penguin using Apache Web server, open source Personal Home Page (PHP) embedded scripting language (now referred to as PHP Hypertext Preprocessor), the open source Lightweight Directory Access Protocol (OpenLDAP) directory server module, mod_ldap, and the Apache open source World Wide Web-based Distributed Authoring and Versioning (WebDAV) module mod_dav.

Mod_ldap and mod_dav, are components installed on the Web server. Mod_ldap interfaces with Apache and LDAP to provide user authentication for access to Web Folder directories.
3.1.2 The Client Side

The client side consists of desktop computers, platform irrelevant, since the protocols are platform independent. The protocols are compatible with Unix, MacOS and Windows 98-Second Edition or higher operating systems. These desktop computers are running Microsoft Web Folders and Web browsers of user choice such as Netscape and Internet Explorer. The only additional requirement for any client machines that will be publishing documents in PDF format is that they have Adobe Acrobat software installed in order to produce the Portable Document Format (PDF) documents.

3.2 The Protocols

The HyperText Transfer Protocol (HTTP) is implemented by the Apache Web server running on Penguin. WebDAV is a set of extensions to this protocol and was developed by the WebDAV working group of the Internet Engineering Task Force (IETF) in an attempt to define new methods which offer additional services for collaboratively editing and managing files on remote Web servers. “An important goal of DAV is to support virtual enterprises, being the primary protocol supporting a wide range of applications… A final goal of DAV is to leverage the success of HTTP in being a standard access layer for a wide range of storage repositories – HTTP gave them read access, while DAV gives them write access.” (Fuqua/Sovich)

Furthermore, “some view DAV as a network filesystem suitable for the Internet, one that works on entire files at a time, with good performance in high-latency
environments. Others view DAV as a protocol for manipulating the contents of a document-management system via the Web.” (Fuqua/Sovich)

Although the DAV specification itself contains no security specifications, by merely being based in HTTP, DAV takes advantage of strong authentication that the HTTP standard offers. Future versions of DAV will support access control lists directly through the DAV protocol itself.

In the development of this particular project, WebDAV was an ideal selection for a primary protocol since it can support a wide range of collaborative applications as well as a wide range of storage repositories. WebDAV also allows a distributed workgroup to share files. “A distributed workgroup composed of members of different organizations seeks a file-sharing solution that does not involve sharing passwords and cases the tasks of central administration… Strong authentication is built in HTTP and the users are added to the server. As the users begin working on the server, they may set permissions and groups for their files as necessary, and through the web, are authenticated to use the files they have created and to which others have given them access.” (Fuqua/Sovich)
4. SYSTEM DESIGN

4.1 Procedural Overview

Implementation of this project required gathering system requirements, determining the requirements list, designing system operations, and integrating the various sub-systems.

4.2 Gathering System Requirements

An incremental and iterative approach was used to gather the system requirements. Using concepts supported by the Unified Modeling Language (UML), "use cases" were constructed for the design of the EDMS. Replacing the traditional requirements list, “use cases” helped the deliverables meet the requirements by focusing the technical designer on the requirements list (Kulack). Similar to modules, each scenario (or use case variation) became a unit of work that had to be designed, constructed, and tested.
4.3 Determining the Requirements List

The functional requirements of the EDMS were used in the design phase of this project to develop the system. Functional requirements are functions and features of the system and are basically what users need for the system to work.

4.3.1 Application Overview and Requirements Analysis

The pilot for this application essentially provides a convenient storage area for files and documents shared by the administrators, faculty and students in the College of Science and Technology at Texas A&M University - Corpus Christi. It automates the document storage and retrieval process. The automated storage process has been organized in such a fashion as to speed up the document-retrieval process. The system allows administrators, faculty, and students to share documents according to their individual access rights.

Other important factors considered for determining the system specification were:

1. User Demography - The main users of the pilot of this system are the personnel in the College of Science and Technology at Texas A&M University-Corpus Christi (TAMU-CC). Hopefully, the users of this application will eventually extend to include the general population of the university.
2. Constraints - The intranet in the College of Science and Technology at TAMU-CC should be leveraged to link together the administrators, faculty and students in order to provide file sharing in the college.

3. Assumptions - Sufficient security and data integrity should be provided using the Internet.

4. Structural Organization Rules - Due to the read and write nature of WebDAV, certain organization rules should be provided to ensure the integrity and security of the documents stored in the EDMS.

4.3.2 Functional Requirements List

The functional requirements for the design of this pilot were as follows:

- Shared documents should be made easily accessible from any location.
- Data and document entry and retrieval operations should be easy for the user to perform.
- Document storage should be well-organized to allow for fast entry and retrieval.
- Document and data entry should be distributed among the users of the system in order to minimize the workload on the administrative staff.
- The intranet in the College of Science and Technology should be leveraged to link all users.
- The integrity of documents should be preserved.
- Highly secure documents should be protected.
4.4 Designing System Operations

Based on the above list and the analysis of it, a system specification was constructed. This was the actual design phase and determined how the system would operate, in terms of hardware, software, and network infrastructure. The user interface was developed and included the formatting of the on-line forms, and reports that would be used. The protocols were selected and the specific programs, databases, and files that would be needed were integrated. "Although most of the strategic decisions about the system were made in the development of the system concept during the analysis phase, the steps in the design phase determined exactly how the system would operate.” (Dennis/Wixom) Furthermore, according to Dennis and Wixom, the goal of the design phase is to create a blueprint for a system that made sense to implement by component integration, architectural design for the system, interface design, database and file specifications, and program design.

4.4.1 Component Integration

Integration includes a description of environmental factors and how these factors relate to other existing systems. A majority of work in the design of the EDMS involved integrating the various components. Based on the environment described in section 3, component integration presented no problems since all of the protocols are platform independent. Fortunately, interfacing with the existing Apache Web server module was the only requirement for all of the major modules.
4.4.2 Architectural Design for the System

This system was developed to use the existing infrastructure in the College of Science and Technology at TAMU-CC. Technical architecture decisions were made with regard to the existing hardware, software, and network infrastructure described in section 3. Little modification to the existing infrastructure was needed to support the requirements identified during analysis.

4.4.2.1 Overview of the Functional Design of the EDMS

In order to understand how the EDMS was accommodated in the improved infrastructure a network model is provided in Figure 4-1. Each piece in this model can be described by what services it performs and what it requires from the other components:

- EDMS Web Folder directories hold the shared files and the .htaccess files with access control.

- Apache provides the interface between client computer's browser and the desktop Web Folder directories. Apache also performs the authentication and authorization checks for the documents stored in the EDMS directories according to the rules specified in .htaccess files.

- Mod_dav is an Apache module that enables documents to be read and written to the Web Folders.

- Mod_ldap is an Apache module that allows Apache to use an LDAP server to perform authentication and authorization.
- The PHP scripts generate HTML output to be sent to the user and provide for manipulation of the `.htaccess` files and LDAP directories. The specific scripts are described in the scripts table provided in Figure 4-2 on page 37.

- Some PHP scripts must perform authentication and authorization prior to accessing EDMS resources. These scripts use `AuthLDAP.php` to perform authentication checks and query the LDAP for authorization information.

*Figure 4-1. Overview of network model for the EDMS pilot*
The network model for the EDMS pilot shows clusters of client PCs from each of the three departments and the Dean’s office connected to Penguin via the existing campus network. Penguin is the computer running all of the server programs and protocols used by this project. The OpenLDAP, WebDAV, and PHP modules were the only new components added to this system in order to implement it. All three of these modules were installed on Penguin, which is currently running Apache. Mod_ldap and mod_dav are Apache server program extensions.

Security and data integrity represented two critical areas of concern in the design of the EDMS. This is due to the client-server architecture and the way that the processing of the system is organized on a central server, distributed across a network of many client PCs, and accessible via the Internet.

4.4.2.2 Data Integrity and Security Issues

Since “DAV allows users to place and manipulate files in a directory on your Web server, you should take particular care in configuring your DAV server. When you enable DAV for a directory or location, you should also enable authentication and authorization for that space. If authorization (for authenticated users) is not enabled, then an anonymous user would have full control of the DAV-enabled portion of your Web server… the files that are managed within the DAV directory should be read/write for the Web server process. Files and directories that are created by the DAV server will have read/write/exec privileges for the user and group (but not the world) of the server process and will be owned by the process’ user/group.”(Stein)
<table>
<thead>
<tr>
<th>Script Names</th>
<th>Script Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin.php</td>
<td>Administration index page</td>
</tr>
<tr>
<td>AuthLDAP.php</td>
<td>Authenticates user of the EDMS</td>
</tr>
<tr>
<td>contactinfo.php</td>
<td>Displays all contact information from employee's record upon further selection</td>
</tr>
<tr>
<td></td>
<td>from telephone directory listing</td>
</tr>
<tr>
<td>Facgroupadmin.php</td>
<td>Adds user of EDMS to group of users</td>
</tr>
<tr>
<td>Staffgroupadmin.php</td>
<td></td>
</tr>
<tr>
<td>facstaff.php</td>
<td>Faculty &amp; Staff index page</td>
</tr>
<tr>
<td>passwd.php</td>
<td>Allows employee to change user password</td>
</tr>
<tr>
<td>persform.php</td>
<td>Allows employee to add or update personal information</td>
</tr>
<tr>
<td>phonedir.php</td>
<td>Displays an indexed alphabetical listing of employees and contains department,</td>
</tr>
<tr>
<td>deptphonedir.php</td>
<td>telephone, e-mail address and office location for each employee</td>
</tr>
<tr>
<td>camsphonedir.php</td>
<td></td>
</tr>
<tr>
<td>palsphonedir.php</td>
<td></td>
</tr>
<tr>
<td>sonephonedir.php</td>
<td></td>
</tr>
<tr>
<td>persinfo.php</td>
<td>Displays an indexed alphabetical listing of employees' records and contains a</td>
</tr>
<tr>
<td>camsinfo.php</td>
<td>quick view of contact information. Further selection of an employee reveals the</td>
</tr>
<tr>
<td>palsinfo.php</td>
<td>employee's complete record</td>
</tr>
<tr>
<td>sonhsinfo.php</td>
<td></td>
</tr>
<tr>
<td>sciinfo.php</td>
<td></td>
</tr>
<tr>
<td>staffinfo.php</td>
<td></td>
</tr>
<tr>
<td>deptstaffinfo.php</td>
<td></td>
</tr>
<tr>
<td>camstaffinfo.php</td>
<td></td>
</tr>
<tr>
<td>palsstaffinfo.php</td>
<td></td>
</tr>
<tr>
<td>sonhstaff.php</td>
<td></td>
</tr>
<tr>
<td>scistaffinfo.php</td>
<td></td>
</tr>
<tr>
<td>facinfo.php</td>
<td>Displays an indexed alphabetical listing of faculty records and contains a quick</td>
</tr>
<tr>
<td>deptfacinfo.php</td>
<td>view of faculty member's tenure status. Further selection of faculty member</td>
</tr>
<tr>
<td>camsfacinfo.php</td>
<td>reveals the complete record</td>
</tr>
<tr>
<td>palsfacinfo.php</td>
<td></td>
</tr>
<tr>
<td>sonhsfacinfo.php</td>
<td></td>
</tr>
<tr>
<td>permissions.php</td>
<td>Allows administrator from Dean's office to set/reset group access rights on</td>
</tr>
<tr>
<td></td>
<td>EDMS Web Folder directories</td>
</tr>
<tr>
<td>useradmin.php</td>
<td>Adds new employee to database and assigns new employee a user password</td>
</tr>
</tbody>
</table>

*Figure 4-2. Table of PHP scripts.*
4.4.2.3 Data Integrity

The issue of data integrity was highly interrelated to the security issue in the design of the EDMS. Data integrity, or correctness, means that the data entered into the system is preserved so that it is identical to the data that is retrieved from the system. Thus, the system was designed to prevent unauthorized users from tampering with or corrupting the data stored in the system. The integrity of the data was preserved by design of security features in the system.

4.4.2.4 Security

System security had been analyzed in depth to safeguard against misuse of data, corruption, and destruction of data. Misuse of data implies the intentional modification of data by an unauthorized user. It may also imply the intentional deletion of data or use of data for personal gains. An authorization scheme was designed into the system to protect the integrity of the data, thus ensuring that the correct user is accessing the data.

4.4.2.5 Authorization

Authorization identifies who is allowed to access the data. Several authorization schemes were considered for the design of the EDMS. One of them involved dynamically reading access control lists from an LDAP data repository. However, due to ease of implementation and time factors for project completion, a flat-file scheme for authorizing users attempting to access the Web Folders directories was selected.
Authorization is handled via flat .htaccess files that are dynamically created and stored in each of the Microsoft Web Folders by the permissions.php script. The .htaccess file identifies the groups of users allowed reading or writing privileges for the specified Web Folders directory and all sub-directories it contains. The access rights for each of these Web Folders are determined by the administrative assistant to the Dean and may be reset at any given time. Thus, the read and write permissions may vary from folder to folder and are set according to four groups of users of this system: 1) administrative staff, 2) faculty, 3) students and others, and 4) top-level administrators. The students and others group was purposely omitted from the write options on the "Edit Directory Permissions" form, since this group of users will never be entering documents or data into the EDMS.

The following exemplifies how the administrative assistant to the Dean may assign varying levels of read and write access rights according to group membership:

1. **Others Access Rights** - For the pilot of this application, the "others" group may be allowed access only to the course syllabi and the telephone/address list. These documents and Web pages are read-accessible only for this group of users.

2. **Faculty Access Rights** - The "faculty" users group may have more privileges set than the "others" group of users. Similar to the previous group of users, the faculty may have access to the course syllabi and the telephone/address list. However, these documents and records may be both read and write accessible for this group of users. The faculty group may also have read or write access, or both, to the Faculty
Handbook, grant information, and the academic calendar, and would definitely have authorization to access the personnel data entry forms page and the curricula vitae.

3. **Administrative Staff Access Rights** - The administrative staff may have similar access rights as the faculty to the documents stored in the EDMS. However, a separate group was defined for administrative staff to accommodate unforeseen procedural rules that may require the differentiation of the two groups. For instance, the administrative assistant to the Dean may determine that this group of users be allowed both read and write access to the Committee Minutes Web Folder directory and that the faculty group be allowed only read access to this set of documents.

4. **EDMS Administrators Access Rights**

EDMS administrators such as the Dean, the Assistant Dean, and the administrative assistant to the Dean may have both read and write access to all the documents and records stored in the EDMS and are thus defined in a separate group of users. For further security purposes, users belonging to this last group should only be defined and entered as a member attribute into the EDMS' "cn=EDMS-Administrators" entry by the system administrator.

Authorization for the PHP generated forms and pages is handled directly from the invoked PHP script such as the `useradmin.php` script. The `AuthLDAP.php` script includes the connection-related functions `ldap_connect` and `ldap_bind` to first establish a connection to the LDAP server and then to authenticate the user. The authenticated user is only allowed to view the user administration page if verified to be a member of the administrators group. For example, the following section of code only allows users
that are attributed members to the administrators group to access the PHP generated
"User Administration" form:

```php
$rid = ldap_read($ldap,
   "cn=EDMS-Administrators,dc=sci,dc=tamucc,dc=edu","(member=$binddn)";
if (!$rid) {
   $err[] = "$binddn is not authorized to view/use this page."
}
```

4.4.2.6 Authentication

Authorization determines who is able to access the data, while authentication
determines if the identity of the user is who the user claims to be. User authentication
is validated via the encrypted passwords stored in the LDAP Directory Information
Tree (DIT). These user passwords have been encrypted with the "crypt" function in
the `useradmin.php` script. When a new user is assigned a password upon submission
of the "User Administration" form to the server, the PHP script uses the "crypt"
function to generate a hash from the submitted password. The encrypted password is
stored in the LDAP DIT.

When an attempt is made to access a file in a protected Web Folder directory an
authentication request is made to the browser by Apache. A small dialog box allows
the user to enter username and password. The submitted authentication information is
returned to `mod_ldap` in Apache which scans the OpenLDAP DIT for a match. Access
to the requested document is allowed by the Apache Web server if the user is
authorized to do so, and in accordance with the `.htaccess` files `<LIMIT>` definition. As
mentioned in the previous section, authorization for a requested document is
determined by the presence or absence of a .htaccess file in a Web Folder directory. It may also be determined within the PHP script.

For the convenience of the user, the assigned user identification and protected password should be the same as those being used to authenticate the user when logging into the existing network. This would alleviate the problem of the user having to remember multiple passwords. However, this aspect is still pending on the entire campus system switching to a centralized LDAP system to store user passwords.

All user information is stored in the LDAP DIT. The data stored in this hierarchical directory tree structure consists of faculty and staff information entered via the PHP-driven data entry forms (refer to sections 2.4.4, 2.4.6 and 2.4.7) residing on the Web server. The persform.php script is called when a request is made from the client. The browser sends back the HTML form in this script, the user enters the data, and upon submission of the form, the data is stored in the entry of the LDAP DIT where the server has binded. This interactive form requests information from the faculty or staff user and then stores the information in the tree-structured directory in the LDAP repository. This hierarchical tree structure, consisting of entries with attributed name-value pairs, contains all of the information pertinent to personnel for the College of Science and Technology at TAMU-CC.

The selection of the OpenLDAP module in the use of this system lends itself extremely well to an authentication scheme. This module performs basically two
major functions for the system: 1) storage and maintenance of organizational records, and 2) user authentication. A variety of new user information is entered into the EDMS system only once via a single user interface. It may be occasionally updated, but is fairly static information. This information, stored on the LDAP server, can be used by any "directory-enabled" application. "Once the data is being stored and updated on an LDAP server, other applications can take advantage of this resource." (Markey) However, the Apache Web server takes care of authenticating the users via HTTP and the .htaccess files.

4.4.3 Interface Design

The interface design involves the way that the user interacts with the system. The design of the graphical user interface from the user's perspective has already been described in the Narrative Section (section 2) of this project report. However, system inputs and system outputs are identified and discussed here from the designer's perspective. It is important to understand how all of the components interface with each other.

Four major scenarios occur between the client and the server: document entry, document retrieval, data entry, and data retrieval. The first two scenarios handle entry into and retrieval from the EDMS of fairly static documents that become active. These documents may be modified according to permissions set on the Web Folder where the document is stored. However, any modifications to the document occur directly from the desktop, in a word-processing program or editor program. No changes to these
documents occur directly from the browser unless the document was created in a "WebDAV-aware" program such as Microsoft Excel. The last two scenarios handle dynamic entry into and retrieval from the EDMS of data. This dynamically stored data consists of information on the user of the EDMS and can be entered into and retrieved from the EDMS via a Web browser.

4.4.3.1 Document Entry

The user accesses the appropriate Microsoft Web Folder from the desktop. The Microsoft Web Folders program provides the interface to allow the user to interact with the server via the desktop.

Upon selection of the appropriate Web Folder, the mod_dav Apache module scans the selected Web Folder’s .htaccess file to determine if the user has read or write access rights to the documents stored in that Web Folder. The group that the user is identified with must have read or write privileges to the folder to even be able to access any documents stored in that folder. If the user's group does have the appropriate write permission, then the user is able to open the folder and enter the document intended for that folder or is able to open that document in another application for editing purposes. Once the document has been dragged-and-dropped into the appropriate folder, it is then uploaded to the server immediately. As soon as the document reaches the server, it is available for viewing.

It is important to keep in mind here that when a user requests a read or write protected document, an authentication dialog box appears on the screen. After submitting the
username and password data to the server, the authentication information is retained until the browser is closed.

4.4.3.2 Document Retrieval

*Mod_dav* again interacts with *mod_ldap* when requests are made to Apache to retrieve documents. *Mod_ldap* authenticates the user via the organizational records stored in the OpenLDAP directory server, and passes the information back to Apache where authorization is then validated. Apache scans the *htaccess* file associated with the requested document for access rights. The *htaccess* file associated with the document is conveniently stored in the same Web Folder where the document requested by the user is stored. If the user has been authorized to access the document, then the document is immediately made available to the user.

4.4.3.3 Data Entry

As stated previously, data entry into the EDMS is handled dynamically from the Internet via a Web browser. Taking advantage of the PHP programming module, forms are created and stored on the Web server allowing faculty and administrative staff to enter information concerning personnel records. The submitted information is stored in the LDAP DIT. According to the *Professional PHP Programming* book, "PHP code is a script code embedded in an HTML page, which is executed on the server before being sent to the browser." (Castagnetto) This feature of PHP offers several advantages: 1) network traffic is minimized because of limited communication between the client and server, 2) loading is faster since only a page of HTML is
actually downloaded, 3) security measures are further improved since the PHP code can never be viewed from the browser, and 4) PHP pages are platform independent.

The information entered is stored in the OpenLDAP directory structure via the PHP scripts. The attributes of faculty vary from the attributes set for administrative staff. The attributes of the objects stored in the OpenLDAP repository vary among organizational groups, and thus may be manipulated according to organizational preferences.

For example, the attributes of either the "Faculty" entry, the "Staff" entry, or the "Administrators" entry in the DIT consist of the two required object classes, "top" and "groupOfNames." All the other attributes for one of these entries consists of members. Each member attribute has a value associated with it containing the member's uid and location in the DIT as seen below in Figure 4-3.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>member</td>
<td>uid=pmichaud,ou=People,dc=sci,dc=tamucc,dc=edu</td>
</tr>
</tbody>
</table>

**Figure 4-3. Example of attribute name-value pair for "cn=Faculty" entry in DIT.**

Figure 4-3 demonstrates an attribute name-value pair as defined for the entry "cn=Faculty" in the DIT. A value for an attribute named member consists of the group member's uid, followed by a comma and the path back out to the root of the tree.
A view of the populated DIT reveals further that an entry may have many attributes of the same name, each defined with a different value. This tree structure follows a generally standard organizational format and "naming scheme" that lends itself well to organizations such as the university. It is formatted to contain personnel records.

A real asset for OpenLDAP is that it runs on top of the Transmission Control Protocol/Internet Protocol (TCP/IP), facilitating interconnectivity with other machines; this also applies to WebDAV. TCP/IP enables these computers to access the Internet via a Local Area Network (LAN). TCP uses IP to transfer a reliable stream of information across a TCP/IP Internet. The server protocols provide access and control of the resources to be shared over the network.

Although WebDAV itself contains no security specifications, by merely being based in HTTP, it can take advantage of the authentication features that the HyperText Transfer Protocol (HTTP) standard offers. HTTP is the standard protocol that allows Web browsers to talk to Web servers. However, HTTP alone performs clear text transmission of a password. At present, the EDMS uses HTTP Basic Authentication to obtain usernames and passwords, and uses mod_ldap to verify passwords against the encrypted passwords stored in the LDAP database. For stronger security, Secure Socket Layer protocols can be added to Apache and OpenLDAP to encrypt the passwords while in transit.
It is important to recognize here the efficiency and security that OpenLDAP provides. It is being used in the EDMS for two key functions of the system: 1) storage of personnel information, 2) and user authentication. “LDAP is a secure protocol in that it makes use of authentication to ensure that transactions are secure… Apart from just authenticating transactions, LDAP provides a very rich set of access control features, which can be used to control who accesses what, in which manner.” (Castagnetto)

4.4.3.4 Data Retrieval

Data retrieval from the EDMS is also handled dynamically from the Internet via a Web browser and using the PHP scripts. These PHP scripts provide the interface between the LDAP directory, the Web server and the client so that the Web server and client may interact. The query contained in the script is answered and the data retrieved from the LDAP repository.

Based on the long-time standard X.500 Directory Server, “LDAP is now an Internet standard for directory services that run over TCP/IP. One or more LDAP servers contain the data that make up the LDAP directory tree. An LDAP client connects to an LDAP server and submits a query to request information or submits information to be updated. If access rights for the client are granted, the server responds with an answer or possibly with a referral to another LDAP server where the client can have the query serviced.” (Markey)
4.4.4 Database and File Specifications

The object-oriented hierarchical naming scheme, consisting of Distinguished Names (DN) entries, was created in lieu of the traditional relational database rows of tables. The attributes of these distinguished names entries behave like fields or indexes. A standard tree template was followed to construct the directory containing the relevant personnel information.

4.4.4.1 The Directory Information Tree

According to Bruce Markey in his article, “A System Administrator’s View of LDAP”, he explains that “an LDAP server is not simply a form of a database, but a specialized server for directories. A directory can be distinguished from a general-purpose database by the usage pattern. A directory contains information that is often searched but rarely modified. Host names or user names, for example, are assigned once (sic) and then looked up thousands of times. LDAP servers are tuned for this type of usage, whereas relational databases are much more geared toward maintaining data that’s constantly changing.” (Markey)

The DIT is comprised of one entry at the top of the hierarchy with a distinguished name of "dc=sci,dc=tamucc,dc=edu" and six distinguished name entries directly below. When binding to a location in the DIT, the path is followed from the desired location in the tree back up to the root of the tree (refer to Figure 4-4).
Figure 4-4. Data organization in the EDMS LDAP DIT.
4.4.2 File Specifications

As mentioned previously, the `.htaccess` file for a given Web Folder directory is created dynamically via the `permissions.php` script and controls all privileges for all directories stored below it, unless otherwise specified.

4.4.5 Design of the PHP Scripts

When invoked, a PHP script identifies and establishes the server connection. The `useradmin.php` and the `persform.php` scripts are basically responsible for maintaining the LDAP directory tree structure. The interactive HTML data entry forms containing the requested fields of information identified in section 2.4.7 are posted at the Web site upon linked requests from the browser to the Web server. Once the information submitted in the forms is uploaded to the Web server and stored in the DIT, the PHP scripts handle queries to the Web server via the `persinfo.php` and `phonedir.php` scripts. `Mod_ldap` is invoked after Apache authenticates the user, thus allowing the authorized user access to the internally-stored directory information that has been requested by the user.

Three basic operations occur any time a request is made to the LDAP server:

1. A connection is established to the LDAP server and access privileges of the connection are established.

2. A search operation is performed on the directory or the directory is modified.

3. The connection is closed.
Since the first operation is frequently performed, the PHP `ldap_connect()` and `ldap_bind()` functions are contained in the AuthLDAP.php script which is included in other scripts. The `ldap_connect()` function establishes the connection while the `ldap_bind()` function establishes the access privileges of the connection. Once the connection is made and privileges are established, the `ldap_read()` function performs the search on the directory according to the specified search filter. The `ldap_first_entry()` function is used to read and return the first entry while the `ldap_first_attribute()` function is used to read and return the first attribute in the entry.

Similarly, the `ldap_add()` function is used to modify objects or entries by adding new entries to the directory while the `ldap_mod_add()` function modifies existing attributes by adding values to the named attributes of the specified entry. It was determined that the `ldap_delete()` function would not be necessary in the development of this application since the data stored in the EDMS' LDAP DIT is to be retained even after an employee leaves the organization.

The HTML forms are embedded in the PHP scripts. When a request is made to the server for a form, only the HTML form in the PHP script is returned to the client via the browser. The user submits the form after the data has been entered and the browser returns the data to the Web server. Any necessary processing is completed and the data is stored in the LDAP repository.
5. RESULTS

The documents are entered, retrieved, modified, deleted and archived via the recently developed WebDAV protocol. This WebDAV protocol enables simple Web page authoring. This means that faculty and administrators do not have to be at personal desktops in offices to access and read the documents. Both groups are able to benefit from the quick and easy access of organized files in a centralized location, as well as the clean up of what may be referred to as hard-copy desk clutter.

Documents never need to leave the system for modification. Centralization and standardization of process is provided in the EDMS. For instance, the old system managed course syllabi by individual departments. There was very little standardization or organization in this paper-based management process for course syllabi. A central on-line repository manages course syllabi by college and introduces the beginnings of a standardized process for document submission. This new system thus alleviates disorganization. The EDMS also provides convenience and consistency in the solution for the Faculty Handbook. Placement of the Faculty Handbook in a centralized on-line location allows for quick and easy access despite location.

The server-side protocol, LDAP, was also an excellent choice since it provides an authentication scheme to help manage the document system. LDAP is very efficient for storing user information which is repeatedly looked up in daily operations, again adding to the convenience and centralization of accessing information. Personnel information no longer needs to leave the system to be shared by all personnel.
5.1 Nonfunctional Requirements

According to Daryl Kulak and Eamonn Guiney, “nonfunctional requirements are principles that run through parts or most of the system and dictate how the system is structured. The way an architecture is structured is part of its architecture, so you can see how nonfunctional requirements contribute to the architecture of an application”. (Kulack/Guiney) Fourteen of these recommended principles were considered and applied to the development of this system: availability, maintainability, data integrity, development cost, delivery date, extensibility, flexibility, installability, leveragability/reuse, operability, performance, portability, robustness, and scalability. These particular principles or metrics were used as guidelines to measure the intended success of the developed EDMS system.(Dennis/Wixom)

5.1.1 Availability

The system must be available to all users 24 hours a day, 7 days a week. The use of the Internet and Internet-based systems fulfills this requirement.

5.1.2 Maintainability

The system must be maintainable by the network administrator with a minimal amount of effort, since the University's Computer Services department does not currently support any of the required software. However, it is intended that the University will purchase and support Adobe Acrobat Writer by the time this project is completed.
5.1.3 Data Integrity

This principle was a driving factor in the success of this project. Information contained in all documents stored in the EDMS are unalterable by students. The .htaccess authorization scheme ensures that secure documents are protected against students. If permissions are not correctly set to authorized users of documents, the write aspect feature WebDAV offers could possibly be abused. Hence, security was the guiding principle in the development of this project.

5.1.4 Development Cost

The cost of the software used in this project was free. As mentioned previously, the Apache Web server, WebDAV, Open LDAP, and PHP modules are free. These modules were installed on the Web server, Penguin, the machine to support the EDMS environment. The only other cost incurred was for the cost of the Adobe Acrobat Writer software and for any document scanner hardware acquired during or after the implementation of this project. Of course, much time was spent to develop the EDMS.

It is still recommended that the University purchase Adobe Acrobat Writer, and acquire a site license so that the Adobe Acrobat Writer may be installed on all personal desktops in the College of Science and Technology at TAMU-CC. It is also still recommended that four Hewlett Packard 9100 scanners be purchased, one for each department in the College of Science and Technology at TAMU-CC, and the fourth for the Dean’s office suite.
5.1.5  Delivery Date

This system had the architectural design and core functionality in place prior to the projected May 2001 delivery date.

5.1.6  Extensibility

This system was built in a way to incorporate the LDAP module in the campus-wide network infrastructure. One of the future goals in the development of this project is to eventually extend the use of OpenLDAP to the campus community at large.

5.1.7  Flexibility

This system is able to interface with other colleges and departments on the TAMU-CC campus based on the nature of the Internet-protocols.

5.1.8  Installability

This system required the installation of the WebDAV module (mod_dav), the OpenLDAP module (mod_ldap), and the PHP module on an Apache Web server. The system also required the installation of a Web browser (i.e., Netscape or Internet Explorer) on all personal desktops and all lab computers. This did not present a problem. The modules were installed on Penguin. All machines on campus already have at least one Web browser installed. Adobe Acrobat Writer still needs to be installed on all personal desktops in the College of Science and Technology at TAMU-CC so that the .pdf file extension may be appropriately used for all documents being written to the EDMS.
5.1.9 Leveragability/Reuse

This system handles user authentication and authorization for any protected documents in the system. There was no problem using Apache's HTTP to authenticate users of the EDMS. The dynamically created .htaccess files are placed in any Microsoft Web Folder directory requiring added protection. It was intended that the user would be able to use the same system log-in password to access protected documents in the EDMS. This would alleviate the problem of multiple passwords. Although the OpenLDAP module of this system conveniently handles this aspect, further implementation must be determined by the Computer Services Department on campus.

The EDMS project provides a successful example application that integrated OpenLDAP with other system components. The OpenLDAP feature of the EDMS may be reused and loaded into other servers across campus to handle user authentication and information.

5.1.10 Operability

The system operations are easily handled. The desktop interface is similar to any drag-and-drop to folder technique that many users have already experienced. The access to the Web-based interface is easily navigable, making access to documents easy to locate. The on-line help is minimal and stored in a Microsoft Web Folder labeled “Help”, which is easily accessible from either the desktop or the browser.

5.1.11 Performance

The on-line response time for all users logged on to the campus network from campus is well within the proposed 10-20 seconds. Of course access time for the EDMS
varies, pending on network traffic. The response time to access the Web Folders from the desktop is less than three seconds, and to drag a document into a Web Folder is less than thirty seconds, pending on the size of the file. Unfortunately, the issue of off-campus network traffic is in the jurisdiction of the main campus in this university system, located at College Station, Texas.

5.1.12 Portability

This system is usable with any Web browser of the user’s choice. This feature is inherent in the natures of the WebDAV protocol and the LDAP protocol, and lends itself to the beauty of this system. Browser-support for PHP pages is not an issue since the PHP scripts are executed prior to the page being sent to the browser. Thus, just by the mere nature of the EDMS being a Web based project and being supported by the aforementioned protocols, it is platform independent.

The Web was originally intended as a shared environment for all users. This intention has been fully realized in the EDMS project via WebDAV. Documents can be written to the Web across any platform, and without lengthy scripting, thus conquering yet another frontier in the development of Internet technology. The scripting in the EDMS project was used to protect documents, to authenticate users of the EDMS, and to enter, update and retrieve information on the users of the EDMS.
5.1.13 Robustness

This system provides authorized desktop users with a Web Folder labeled "Help" containing help documentation. The help documents may be created at any time by the users of the EDMS and are accessible from all Web pages or from the desktop.

The help documentation currently focuses on helping the user with interactive parts of the site such as the Data Entry Forms Pages. However, the Web graphical user interface is fairly self-explanatory in its design so as to provide the user with the capacity for easy navigation.

This system handles most user errors while running. For example, it provides users with error-handling messages when wrong passwords are entered for authentication reasons, or when an unauthorized user attempts to access a protected document. An improper response from a user does not "hang" the system or cause it to malfunction.

5.1.14 Scalability

This system provides a fast response time with a high load of concurrent users. Aside from the issue of the network traffic, this system scales very well due to the document storage nature and tree directory structure of the system.
5.2 Social Impact

The social impact of WebDAV is best summarized in the following statements by Jim Whitehead in his article “Web-based Development of Complex Information Products” from the journal *Communications of the ACM*:

“Taken together, the WebDAV extensions to HTTP provide the standard needed to make the Web a writable, collaborative medium… As WebDAV technology is deployed, it will initially have its largest impact on small to medium sized workgroups, which homogeneously support DAV, allowing their practices to coalesce around a local Intranet… WebDAV additionally shows significant promise as an infrastructure for development of distributed software engineering environments and other complex information products… WebDAV in the home will make Web page creation significantly easier, since Web pages will be editable in-place… WebDAV will allow school children to collaborate easily on projects and reports… **By giving more voices access to the global distribution of the Web, and by making it easier to collaborate, WebDAV technologies will have broad social impact** (sic).” (Whitehead)

It is clear that the Web is becoming a rich infrastructure for collaborative applications, and the protocol WebDAV is at the forefront in respect to new methods of editing and managing files on remote servers.

The Apache Software Foundation (ASF) represents a major force behind the open-source movement. The ASF has encouraged software companies to incorporate the free mod_dav add-on module. The co-founder of the ASF, Brian Behlendorf
comments that, “the evolution of WebDAV has been exciting to watch. Now Apache’s 9 million users worldwide can gain the benefit of managing files on a Web site, querying the properties of the files, editing resources on the Web server without downloading them first (sic), and setting access permissions on files during editing.” (Behlendorf) Roy Fielding’s comment sums up further the impact of the WebDAV technology:

“We have reached an important milestone in the progress of the Web from being a read-only tool to reaching its true goal: making the Internet collaboration as easy as using a hypertext browser. The addition of mod_dav to Apache will make it possible for Internet Service Providers, both large and small, to give their customers complete write access to their own personal and project web sites without requiring the use of vendor-specific tools or complex file-transfer mechanisms (sic).” (Fielding)

5.3 Conclusion

The EDMS offers many benefits to the personnel in the College of Science and Technology. The WebDAV component allows for high portability. Documents can be electronically shared from a centralized and easily accessible on-line location. Personnel do not have to be on campus to access documents such as the Faculty Handbook. Policies can be posted in this centralized location. The LDAP DIT component can store faculty and staff information.
Modifications to documents and data are made entirely within this electronic system, thus reducing hard-copy clutter. The distributed document and data entry is no longer a redundant procedure. The single greatest advantage of the EDMS is the amount of human time it saves. Furthermore, the simplicity in the design of the EDMS provides easy maintenance and is readily expandable. Future developers need only to create new scripts.

5.4 Future Work

The Internet provides a convenient platform for distributed groupware systems. Recently we have seen a variety of Web-based groupware systems that are implemented using client/server-side scripting. These scripts are used to implement the basic tasks of collaboration.

The protocol WebDAV completes the original vision of the Web as a shared environment by providing write access to the Web without the use of heavy scripting. However, scripting can be used to simplify maintenance of the application. The EDMS uses the permissions.php script to support this concept. Future work may be done to further simplify maintenance of the EDMS. For example, an option should be provided to allow the administrative user to set or reset permissions for Web Folder sub-directories when additional protection is needed on the subdirectories, and security may be tightened around individual ownership of documents if the need arises.
Locking features and access control lists will be offered in future releases of WebDAV. Upgrades to WebDAV on Penguin will accommodate these added features. Also, the Secure Socket Layer may be added to ensure further security on the passwords in transit. The Computer Services department is currently researching this protocol.

Further views of the data stored in the DIT may be requested by the users of the EDMS and can be easily represented in future work. There is a wide variety of possibilities for additional information to be stored in the DIT.

One final area of future work may entail the creation of additional features on the WebFolder directory interface. For instance, descriptions of modifications to documents may be added to this interface. Also, the EDMS user may benefit from the Web Folder directory interface identifying the person responsible for changes made to a document.
REFERENCES

Books


Journals


Articles


Web Site Articles


Whitehead, Jim. "WebDAV: Evolving the Web into a Read and Write Medium An Interview with Jim Whitehead, Chair IETF WebDAV". http://www.webdav.org/. Last Update Last Access
Web Sites

LINUX Center Hong Kong Ltd: Resources

Open LDAP official WWW site.

PHP official WWW site.

WebDAV official WWW site.

Apache official WWW site.
http://www.apache.org. Copyright date N/A. Last update N/A.
ACKNOWLEDGEMENTS

I wish to thank the people who contributed to the success of this project. Their input, advice, patience and support helped make this endeavor a rewarding experience for me. First and foremost, I would like to acknowledge the open-source community since this project is heavily dependent on them. The time and effort spent in developing open-source code makes projects and applications of this nature possible. Secondly, I wish to thank my committee chairperson, Dr. Patrick Michaud, who spent many hours with me during the development and implementation phases of this project answering a multitude of questions. He has been my guiding light throughout this project and I sincerely appreciate the high standards he set for me. I hope to some day become as eloquent a programmer as he is. I also wish to thank my committee member, Dr. Stephen Dannelly, for directing me to this project, and for the many pertinent and useful skills I learned from him in the Software Engineering course. My other committee member and advisor, Dr. Michelle Moore, who shared the learning experience with me in this project is also due many thanks. Her insights and questions greatly contributed to my understanding, particularly in regard to the two major protocols used. Her support and encouragement as an advisor also got me through a personally difficult semester.

I wish to thank Dr. Diana Marinez, the Dean of the College of Science and Technology at Texas A&M University - Corpus Christi, for the time she took out of her busy schedule for client interviews. Her executive secretary, Kenneth Brown, was also a valuable source of information, and I greatly appreciate his time for client interviews to ensure the correct product was delivered. Further, I acknowledge the administrative secretary for the department of Computer Science, Magdalena Garza, whose feedback from the testing of an early prototype of the software aided in the success of the deliverable application.

My friends and colleagues, aside from being an inspiration throughout this program, have been a wealth of information. Specifically, Jonathon Scott Duff who is very knowledgeable about hierarchical databases such as the LDAP directory tree structure. He was very helpful in my initial exposures to the directory information tree and how it worked. Another friend and colleague, David Shontz, has been a great sounding board, particularly in the development of the graphical user interface phase. He was always willing to answer the occasional odd question. I appreciate his support, as well as his constant readiness to debate an issue, thus contributing to my learning experience in general. Many thanks go to Nancy Cameron whose constant encouragement throughout this program indirectly influenced the successful completion of this project. One final acknowledgement is due to my sister, Judi, a computer science professional, who fully supported my decision to make a career change.