Electronic Learning in Nursing Education

GRADUATE PROJECT REPORT

Submitted to the Faculty of
the Department of Computing and Mathematical Sciences
Texas A&M University – Corpus Christi
Corpus Christi, Texas

in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

by

Shiraz Ahmed
Fall 2004

Committee Members

Dr. Mario Garcia
Committee Chairperson

Dr. Dulal Kar
Committee Member

Dr. David Thomas
Committee Member
Electronic Learning in Nursing Education

GRADUATE PROJECT REPORT

Submitted to the Faculty of
the Department of Computing and Mathematical Sciences
Texas A&M University – Corpus Christi
Corpus Christi, Texas

in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

by

Shiraz Ahmed
Fall 2004

Committee Members

Dr. Mario Garcia
Committee Chairperson

Dr. Dulal Kar
Committee Member

Dr. David Thomas
Committee Member
# TABLE OF CONTENTS

Abstract .................................................................................................................. ii

Table of contents ................................................................................................. iii-iv

List of figures ........................................................................................................ v-vi

1. Background and Rationale ................................................................................... 1
   1.1 How eLine works ........................................................................................... 1
   1.2 Overview ........................................................................................................ 1
   1.3 Process ........................................................................................................... 1-4
   1.4 Purpose .......................................................................................................... 5
   1.5 Overall Degree Plan ...................................................................................... 5-6
   1.6 Software Products ....................................................................................... 6

2. Narrative ............................................................................................................ 7
   2.1 Overview ........................................................................................................ 7
   2.2 The Process ................................................................................................... 7
   2.3 Description of user interface ........................................................................ 8
      2.3.1 Homepage .............................................................................................. 8-10
      2.3.2 Faculty Interface map ............................................................................ 11
         2.3.2.1 Change Password .......................................................................... 13-15
         2.3.2.2 Enter student information ............................................................... 16-17
         2.3.2.3 Register student for a course .......................................................... 18-20
         2.3.2.4 Drop student from a course ............................................................. 21-23
         2.3.2.5 Student course transcript ................................................................. 23-24
         2.3.2.6 Student module transcript ............................................................... 25-26
         2.3.2.7 List enrolled courses of a student ..................................................... 26-27
         2.3.2.8 List enrolled modules of a student .................................................... 28-29
         2.3.2.9 Check module’s grade for a course .................................................. 29-30
         2.3.2.10 List of students in a course ............................................................. 31-32
         2.3.2.11 Upload file to enter module grade .................................................. 32-33
         2.3.2.12 Create file for registering students ............................................... 34

3. System design .................................................................................................... 35
   3.1 Project Environment ..................................................................................... 35
   3.2 Database schema .......................................................................................... 35-36
   3.3 Entity relationship diagram ......................................................................... 37
   3.4 Data Flow diagram ...................................................................................... 38-41
   3.5 Listing of major scripts ............................................................................... 42-43
   3.6 Security features ......................................................................................... 43
      3.6.1 Session Id ............................................................................................. 43-44
      3.6.2 One time key parameters ..................................................................... 44
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.6.3</td>
<td>Password authentication</td>
<td>44</td>
</tr>
<tr>
<td>3.6.4</td>
<td>Using SSL</td>
<td>44</td>
</tr>
<tr>
<td>3.6.5</td>
<td>Lockout mechanism</td>
<td>44-45</td>
</tr>
<tr>
<td>3.6.6</td>
<td>Backing up data</td>
<td>45</td>
</tr>
<tr>
<td>3.6.7</td>
<td>Password expires</td>
<td>45</td>
</tr>
<tr>
<td>4.</td>
<td>Testing &amp; Evaluation</td>
<td>46</td>
</tr>
<tr>
<td>4.1</td>
<td>Usability testing</td>
<td>46</td>
</tr>
<tr>
<td>4.2</td>
<td>Security Evaluations</td>
<td>46-47</td>
</tr>
<tr>
<td>5.</td>
<td>Results and conclusion</td>
<td>48</td>
</tr>
<tr>
<td>6.</td>
<td>Future work</td>
<td>49</td>
</tr>
<tr>
<td>6.1</td>
<td>Access control</td>
<td>49</td>
</tr>
<tr>
<td>6.1.1</td>
<td>Identification and Authentication</td>
<td>49-50</td>
</tr>
<tr>
<td>6.1.2</td>
<td>Authorization</td>
<td>51</td>
</tr>
<tr>
<td>6.1.3</td>
<td>Accountability</td>
<td>51</td>
</tr>
<tr>
<td>6.2</td>
<td>Cryptography</td>
<td>52-53</td>
</tr>
<tr>
<td>6.3</td>
<td>Integrated approach for database security and fault tolerance</td>
<td>53-54</td>
</tr>
</tbody>
</table>

Acknowledgements ................................................................. 55

Bibliography and references .................................................. 56

Appendix A. Data Dictionary .................................................. 57-60

Appendix B. User manual ....................................................... 61-63
LIST OF FIGURES

Figure 1.1 Delmar college course matrix.........................................................3
Figure 1.2 TAMUCC course matrix.................................................................4
Figure 2.1 Home Page.......................................................................................9
Figure 2.2 Functionalities..............................................................................9
Figure 2.3 Invalid user..................................................................................10
Figure 2.4 Faculty Interface map.................................................................12
Figure 2.5 Change Password......................................................................14
Figure 2.6 Change Password successful...................................................14
Figure 2.7 Change Password unsuccessful................................................15
Figure 2.8 Enter Student information.........................................................16
Figure 2.9 Invalid Student Id...................................................................17
Figure 2.10 Overwriting new information................................................17
Figure 2.11 Student registration.................................................................18
Figure 2.12 Student registered successfully.............................................19
Figure 2.13 Student already registered for a course..............................19
Figure 2.14 Student’s record does not exist..........................................20
Figure 2.15 Course does not exist.............................................................20
Figure 2.16 Drop student from a course..................................................21
Figure 2.17 Student cannot be dropped after course completion.........22
Figure 2.18 Student dropped with grade “W”...........................................22
Figure 2.19 Student dropped from a course............................................23
ABSTRACT

The project under consideration deals with making a Web database application using PHP & MYSQL for eLine (Electronic Learning in Nursing Education). A Web interface is made outlining the basic activities like registration, addition and deletion of courses and modules, listing of courses and modules, automated grade calculation and creating the file format for student registration. This application is intended for those students pursuing nursing education at Delmar college and Texas A&M University – Corpus Christi. The database is created using MYSQL and Web interface is made using PHP and Dreamweaver. Various scripts in PHP are made to perform different processes that will communicate with the database.
1. BACKGROUND AND RATIONALE

1.1 How eLine Works

The partnership of Texas A&M University – Corpus Christi and Delmar College has created a way for nursing education to come via the internet. eLine offers the Nursing Degree programs of either Texas A&M University- Corpus Christi or Delmar College in an online format that is available anywhere and anytime [FIPSE 2001]. There is no need to travel long distances to class – the classroom comes to you.

1.2 Overview

In order to participate in eLine, a student must be admitted to either Delmar or TAMU-CC and be accepted into the nursing program for that school. A student can choose either the Associate Degree Nursing program at Delmar or the Baccalaureate Degree at TAMU-CC [FIPSE 2001]. A student after registering for a particular course will have access to the modules that make up for that course. Each module will be graded, all module grades will be weight averaged, and a final grade will be issued for that course.

1.3 Process

There are various courses being offered at Delmar College and TAMU-CC for the nursing program. Each course consists of many different modules. Some modules are shared between different courses. The course matrices for Delmar College and TAMUCC are illustrated in Figures 1.1 and 1.2. The modules that are red in color are shared between different courses. When a student is registered for a particular course, he or she gets automatically enrolled in the modules associated with that course. A course is completed only when all the modules associated with a particular course are completed.
A grade is assigned for each module and using the corresponding module weights for each module, the final course grade is calculated.
<table>
<thead>
<tr>
<th>Del Mar College First Semester</th>
<th>Del Mar College Second Semester</th>
<th>Del Mar College Third Semester</th>
<th>Del Mar College Fourth Semester</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Courses</strong></td>
<td><strong>Courses</strong></td>
<td><strong>Courses</strong></td>
<td><strong>Courses</strong></td>
</tr>
<tr>
<td>RNSG 1201: Pharmacology</td>
<td>RNSG 1161: Clinical (Medical-Surgical I-A)</td>
<td>RNSG 2160: Clinical (Mental Health)</td>
<td>RNSG 2161: Clinical (Medical-Surgical IV-A)</td>
</tr>
<tr>
<td>RNSG 1171: Introduction to Patient Care Communication</td>
<td>RNSG 1343: Concepts of Adult Health</td>
<td>RNSG 1512: Nursing Care of Childbearing and Childrearing Family</td>
<td>RNSG 2341: Advanced Clinical Decisions</td>
</tr>
<tr>
<td>RNSG 1160: Clinical (Fundamental)</td>
<td>RNSG 1162: Clinical (Medical-Surgical II-B)</td>
<td>RNSG 2260: Clinical-Maternal Child</td>
<td>RNSG 2162: Clinical (Medical-Surgical IV-B)</td>
</tr>
<tr>
<td>PRA101 BAA204 HPA208 PHA201 PHA206 CLA 201</td>
<td>PRA 204, DPA 203, DPA 206, DPA 211, DPA 221 CLA 202</td>
<td>DPA 201, DPA 202, DPA 207, PRA 203, DPA 208, DPA 218, PHA 203, PHA 205, PHA 206, PHA 208 CLA 207, CLA 208</td>
<td>DPA 205, DPA 216, DPA 204, DPA 208, DPA 218, PRA 106, ALL BAA, HPA, PHA MODULES CLA 204</td>
</tr>
<tr>
<td>PRA102 BAA205 HPA209 PHA202 PHA207 BAA 201, BAA 202, DPA 207, PRA 201, DPA 208, DPA 210, PHA 204, PHA 209</td>
<td>RNSG 1341 &amp; 1343 must be finished prior to CLA 202.</td>
<td>RNSG 1341 &amp; 1343 must be finished prior to CLA 202.</td>
<td>RNSG 2331 and 2341 must be finished prior to CLA 204.</td>
</tr>
<tr>
<td>PRA103 HPA201 HPA207 PHA203 PHA208 BAA 102, PRA 101</td>
<td>ALL BAA, HPA, PHA MODULES</td>
<td>ALL BAA, HPA, PHA MODULES</td>
<td>ALL BAA, HPA, PHA MODULES</td>
</tr>
<tr>
<td>PRA104 HPA202 HPA211 PHA204 PHA209</td>
<td>DPA 204, DPA 218, DPA 220, PHA 204 PHA 206, PHA 210</td>
<td>DPA 207, PRA 203, DPA 208, DPA 210, PHA 204, PHA 209</td>
<td>DPA 207, PRA 203, DPA 208, DPA 210, PHA 204, PHA 209</td>
</tr>
<tr>
<td>PRA201 HPA204 PHA202</td>
<td>CLA 202</td>
<td>CLA 202</td>
<td>CLA 202</td>
</tr>
<tr>
<td>BAA101 HPA205</td>
<td>RNSG 1341 &amp; 1343 must be finished prior to CLA 202.</td>
<td>RNSG 1341 &amp; 1343 must be finished prior to CLA 202.</td>
<td>RNSG 2331 and 2341 must be finished prior to CLA 204.</td>
</tr>
<tr>
<td>BAA102 HPA206 BAA206</td>
<td>ALL BAA, HPA, PHA MODULES</td>
<td>ALL BAA, HPA, PHA MODULES</td>
<td>ALL BAA, HPA, PHA MODULES</td>
</tr>
<tr>
<td>BAA103 HPA207</td>
<td>ALL BAA, HPA, PHA MODULES</td>
<td>ALL BAA, HPA, PHA MODULES</td>
<td>ALL BAA, HPA, PHA MODULES</td>
</tr>
</tbody>
</table>

**Figure 1.1 Delmar college course matrix**
<table>
<thead>
<tr>
<th>Semester</th>
<th>TAMUCC Courses</th>
<th>NURS 3202: Nurse as a Professional</th>
<th>NURS 3614: Fundamentals of Nursing Care</th>
<th>NURS 3318: Nurse as a Therapeuti</th>
<th>NURS 3435: Health Assessment-Clinical Lab</th>
<th>NURS 3342: Use of Pharmacology Principles in Humanistic Caring</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Modules</td>
<td>PRB 214, PRA 104, PRA 105, PRA 106</td>
<td>PRB 215, PHA 203, PHA 204, PHA 205, PHA 206, PHA 207, PHA 208, PHA 209, PHA 210, PHA 211, PHA 212, PRA 201, PRA 202, CLA 201</td>
<td>BAA 201, BAA 202, BAA 203, BAA 204, BAA 205, BAA 206, BAA 207, BAA 208, BAA 209, BAA 210, BAA 211, BAA 212, PRA 101, PRA 102, PRA 103, PRA 104</td>
<td>PHA 201, PHA 202, CLB 201, PRA 204, PRA 205, PRA 206, PRA 207, PRA 208, PRA 209, PRA 210, PRA 211, PRA 212, PHA 201, PHA 202, CLA 203</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Semester</th>
<th>TAMUCC Courses</th>
<th>NURS 3628: Nursing Care of Adults I</th>
<th>NURS 3548: Nursing Care of Children &amp; Their Families</th>
<th>NURS 3550: Nursing Care of Parents/Newborns</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Modules</td>
<td>DPA 201, DPA 202, DPA 203, DPA 205, DPA 211, DPA 220, DPA 221, CLA 202, PRA 203, PRA 101, PRA 102, PRA 201, DPA 218</td>
<td>FCA 201, FCA 202, FCA 203, FCA 204, FCA 205, FCA 206, FCA 207, CLA 207, PRA 101, PRA 102, PRA 201, PRA 204</td>
<td>FCA 208, FCA 209, FCA 210, FCA 211, FCA 212, CLA 208, PRA 101, PRA 102, PRA 201, PRA 204, DPA 220</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Semester</th>
<th>TAMUCC Courses</th>
<th>NURS 4564: Nursing Care Psychiatric Clients</th>
<th>NURS 4318: Nurse as Research Consumer</th>
<th>NURS 4628: Nursing Care of Adults II</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Modules</td>
<td>DPA 212, DPA 213, DPA 214, DPA 215, DPA 217, DPA 218, DPA 219, DPA 222, DPA 216, BAA 201, BAA 202, PRA 101, PRA 102, PRA 201, PRA 204, CLA 206</td>
<td>PRB 207, PRB 208, PRB 209, PRA 102, BAA 102</td>
<td>DPA 204, DPA 207, DPA 208, DPA 209, DPA 210, CLA 204, CLA 205, PRA 101, PRA 102, PRA 201, PRA 204, PRA 203, DPA 216</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Semester</th>
<th>TAMUCC Courses</th>
<th>NURS 4250: Professional Nursing Issues/Trends</th>
<th>NURS 4660: Nursing Care of Community Health Clients</th>
<th>NURS 4670: Nurse Coordinating Care as Leader/Manager</th>
<th>NURS 4390: Dimensions in Nursing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Modules</td>
<td>PRB 202, PRB 210</td>
<td>PRB 206, PRB 217, PRB 218, PRB 219, PRB 220, PRB 221, CLA 203, PRA 101, PRA 102, PRA 201, PRA 204</td>
<td>PRB 205, PRB 210, PRB 216, PRB 211, CLA 202, PRA 204, PRB 214</td>
<td>PRB 201</td>
</tr>
</tbody>
</table>

Figure 1.2 TAMUCC course matrix
1.4 Purpose

The main purpose of making this database application is to keep track of all the students who are taking part in the eLine program. Administrative staff can add or drop a student from a particular course or module any time. They have total control over the records of all the students. Only administration staff is allowed to access this application. When a student has completed a particular module, his grade is entered into the database using the interface. When all the modules are completed, the course grade is automatically calculated and entered into the database. The staff has the option of changing the student’s information and records according to their convenience.

1.5 Overall Degree Plan

Both Delmar College and TAMUCC offer courses related to nursing. There are basically two degrees offered – ADN (Associate Degree of Nursing) and BSN (Bachelor’s Degree of Nursing). Each course consists of many different modules. A module is represented by three characters followed by three numbers. For example, a module can be represented as: PRA 204 or CLA 105. If the last letter in module designation is A, then it is required for ADN and BSN students. If the last letter in module designation is represented by B, then it is required for BSN students only. If the first number in module designation is 1, then no prerequisites are required. If it is represented by 2, then prerequisites are required. Each module has a weight assigned in a course. This module weights are used to calculate the overall grade of the course. Based on the scores obtained, an appropriate grade is assigned. Grading system assigned for TAMUCC and Delmar colleges are based on the following criteria:

• If the score range is from 90 to 100, then Grade A is assigned.
• If the score range is from 83 to 89, then Grade B is assigned.

• If the score range is from 75 to 82, then Grade C is assigned.

• Grade D is assigned for TAMUCC, if the score range is from 67 to 74 but for Delmar college the score is in the range from 70 to 74.

• Grade F is assigned for TAMUCC for scores below 66 and for Delmar for scores below 69.

1.6 Software Products

This Web database application is made using Dreamweaver MX, PHP & MYSQL. The Web interface through which user will access will be made using Dreamweaver. The database is created using MYSQL. Server side scripting language PHP will be used to communicate with the database. PHP and MYSQL are used because they are open source and are available free of cost [Welling 2001]. These products are readily available in our university. Compared to other scripting languages like ASP, Cold Fusion, Perl and Java, PHP is preferred mainly due to the following reasons:

• PHP is much faster when compared to ASP

• Unlike Cold Fusion, PHP runs on virtually every platform

• Format and syntax of PHP is easy when compared to Perl

• PHP is simpler to use than Java and makes it easy to architect Web applications
2. NARRATIVE

2.1 Overview

The eLine (Electronic Learning in Nursing Education) is a Web based system that provides faculty with information about each and every student in less time via the internet. The information of students can be updated and accessed as and when they require. The courses in which the student is enrolled, their respective grades in each and every module and their grades can also be calculated.

2.2 The Process

First the student information is entered into the database using the Web interface. The student’s ID, first name, last name, middle name and the school in which he is going to be enrolled will be entered in the database. Then using the student’s ID, he will be registered for the courses that he is willing to take. Once the student is registered for a particular course, the modules associated with the course will be automatically registered. The staff has also the option of dropping a course. The course transcript and module transcript of each and every student can be viewed either by using student’s ID or student last name. The list of enrolled courses or enrolled modules of a student can be viewed either by using student’s ID or student last name. Each module has a file associated with it. The file contains the grades obtained by different students for that particular module. When a particular file is uploaded, all the grades associated with a particular student are entered into the database. If after entering the module grade, the student completes all the modules for a particular course, then the course grade is calculated and entered into the database. All the above processes explained above are implemented by means of PHP scripts which interact with the MYSQL database.
2.3 Description of user Interface

There is only one type of user who is going to access the Web interface – faculty. The faculty will first have to login through the homepage. Various options will be provided to the faculty to perform different functions. The interface can be basically divided into two parts:

- Homepage
- Faculty interface map

2.3.1 Homepage

This is the first page the faculty member sees on entering the site. It consists of a header, footer and asks for the username and password. The links to TAMUCC and Delmar College and also the contact information for eLine is also provided. After implementation, the homepage looks as shown in Figure 2.1. After the users log in, they are presented with different functionalities as illustrated in Figure 2.2. If the user fails to log in, then the interface prompts the user to try again that is depicted in Figure 2.3.
Figure 2.1 Homepage

Figure 2.2 Functionalities
Figure 2.3 Invalid user
2.3.2 Faculty Interface Map

The Faculty Interface map provides a list of all processes involved. It lists all the different options to be performed. The faculty can select the appropriate functionality depending on the operation needed to be performed. Various options provided by the Faculty Interface map are summarized as shown in Figure 2.4.
Fig 2.4 Faculty Interface Map
The various functionalities associated with Faculty Interface Map can be explained as follows:

2.3.2.1 Change Password:

This process allows the faculty to change the password of a particular username. This password corresponding to a particular username is entered at the Homepage. This option interface accepts two parameters: Old Password and New Password. It replaces the Old Password by New Password. This interface is shown in Figure 2.5. If the Old Password is entered successfully, then the message is prompted that the password has been changed successfully as shown in Figure 2.6, otherwise a message is prompted to check the password and try again as depicted in Figure 2.7.
Figure 2.5 Change Password

Figure 2.6 Change Password successful
Figure 2.7 Change Password unsuccessful

Your password is not changed. Please check your password and try again.
2.3.2.2 Enter student information:

This process allows the faculty to enter the student information into the database. The interface for this option accepts parameters: Student ID, First Name, Middle Name, Last Name and School. This is shown in Figure 2.8. The Student ID entered should be in appropriate format. Error messages will be displayed upon wrong input as shown in Figure 2.9. If a Student ID already exists and the user tries to enter the same Student ID again then the user is prompted by a message whether to overwrite the new information with the old information as shown in Figure 2.10.

![Figure 2.8 Enter student information](image)
Figure 2.9 Invalid Student Id

Figure 2.10 Overwriting new information
2.3.2.3 Register student for a course:

The faculty can register a student in a particular course using this option. The interface accepts parameters: Student ID and CourseNo as shown in Figure 2.11. The student is registered in the appropriate CourseNo as shown in Figure 2.12. If the student is already registered for a course, then appropriate messages are displayed as shown in Figure 2.13. The script will check for the validity of Student ID and CourseNo in the database. If either Student ID or CourseNo is not present, then error messages will be displayed as shown in Figures 2.14 and 2.15.
Figure 2.12 Student registered successfully

Figure 2.13 Student already registered for a course
Figure 2.14 Student’s record does not exist

Figure 2.15 Course does not exist
2.3.2.4 Drop student from a course:

This process allows the faculty to drop a student from a particular course. The interface accepts parameters: Student ID and CourseNo. This is shown in Figure 2.16. The student is dropped in the appropriate CourseNo. The interface will ask the user whether the student should be dropped with a grade “W” or not. Grade “W” represents withdrawal of a student from a particular course. If a student has already completed the course, then he cannot be withdrawn from a course as shown in Figure 2.17. A student can be withdrawn from a course with or without grade “W” as shown in Figures 2.18 and 2.19.
Figure 2.17 Student cannot be dropped after Course completion

Figure 2.18 Student dropped with Grade “W”
2.3.2.5 Student course transcript:

This process displays the grades obtained by a student in all courses. If the student has not completed a particular course, then a grade of “I” is assigned. Grade “I” represents that the course is incomplete. The interface accepts parameters: Student ID or Lastname as shown in Figure 2.20. Course transcript will also display the date when a particular course was started and completed as shown in Figure 2.21.
Figure 2.20 Interface for Student course transcript

Figure 2.21 Student course transcript
2.3.2.6 Student module transcript:

This process displays the grades obtained by a student in all modules. If the student has not completed a particular module, then a default grade of 0 will be assigned. Grade 0 represents that the module is incomplete. The interface accepts parameters: Student ID or Lastname as shown in Figure 2.22. Module transcript will also display the date when a particular module was started and completed as shown in Figure 2.23.
Figure 2.23 Student module transcript

2.3.2.7 List enrolled courses of a student:

The interface accepts parameters: Student ID or Lastname as shown in Figure 2.24. List of courses enrolled by a student can be accessed either using Student ID or Lastname. It will list all courses currently enrolled by a student but not yet completed as shown in Figure 2.25.
Figure 2.24 Interface to list all courses of a student

Figure 2.25 Currently enrolled courses of a student
2.3.2.8 List enrolled modules of a student:

The interface accepts the following parameters: Student ID or Lastname. List of modules enrolled by a student can be accessed either using Student ID or Lastname as shown in Figure 2.26. It will list all modules currently enrolled by a student but not yet completed as shown in Figure 2.27.

Figure 2.26 Interface to list all modules of a student
2.3.2.9 Check module’s grade for a course:

This interface accepts either Student Id or last name and Course No as shown in Figure 2.28. It will display the module grades obtained by a student in a particular course as shown in Figure 2.29.
Figure 2.28 Interface to check module’s grade for a course

Figure 2.29 Module’s grade for a course
2.3.2.10 List of students in a course:

This interface accepts the Course No as shown in Figure 2.30 and displays the list of students enrolled in that course as depicted in Figure 2.31.

Figure 2.30 Interface to list students in a course
2.3.2.11 Upload file to enter module grade:

This process accepts the file associated with any module as shown in Figure 2.32. It then parses the file and retrieves the module grade for a particular student ID. It then inserts this module grade into the database. Each course consists of many modules. Modules are also shared among different courses. After the module grade is inserted, the script does the checking to find out if any courses are completed. If any course is completed, the grade is calculated and entered into the database. The process is repeated for all the student ID’s present in the file. This script is automated because just by uploading the module file, course grade is calculated and entered into the database. This is illustrated in Figure 2.33.
Figure 2.32 Interface to upload module file

Figure 2.33 Entering module grades from file
2.3.2.12 Create file for registering students in WebCT:

A file is created using the information from the database. It gives information about all students and also displays the courses in which the students are currently enrolled. The information is presented in a specific format as shown in Figure 2.34 and is sent as an e-mail to one of the faculty members.

Figure 2.34 File format for registering students in WebCT
3. SYSTEM DESIGN

3.1 Project Environment

The system is designed to run on any Web browser such as Netscape or Internet Explorer. The MYSQL database running under Red Hat Linux is used as the back end. The user interfaces that make the front end of the system is developed using Dreamweaver MX. The communication between the front end interfaces and the MYSQL database is provided by Hypertext Preprocessor (PHP) scripts [Williams 2002].

3.2 Database Schema

The database schema [Welling 2001] for the software of the system is named as “eline”. It comprises the following tables:

Catalog (CourseNo, ModuleNo):

It contains the list of all possible courses and the modules associated with each and every course. CourseNo represents a particular course and ModuleNo represents a particular module. Both these fields are primary keys.

Course (CourseNo, CourseName):

It contains the list of all courses along with their names. CourseNo represents a particular course and CourseName represents the name associated with that course. CourseNo represents the foreign key.

Module (ModuleNo, ModuleName):

It contains the list of all modules along with their names. ModuleNo represents a particular module and ModuleName represents the name associated with that module. ModuleNo represents the foreign key.
**CourseTranscript** *(StudentID, CourseNo, Grade, NumberGrade, StartDate, ImportDate):*

It gives the grade obtained in a course by a particular student. It also tells us when the course was started by a particular student and the date when the course grade was calculated and entered into the database. StudentID represents the primary key. NumberGrade indicates the numerical score and Grade represents the letter grade obtained by the student.

**ModuleTranscript** *(StudentID, ModuleNo, Grade, StartDate, ImportDate):*

It gives the grade obtained in a module by a particular student. It also tells us when the module was started by a particular student and the date when the module grade was uploaded into the database. StudentID represents the primary key. Grade indicates the numerical score.

**StudentInfo** *(StudentID, FirstName, MiddleName, LastName, School):*

It gives the detailed information about a student. It tells us about his first name, middle name, last name and the school in which the student is enrolled in. StudentID represents the primary key.

**Login** *(UserId, Password):*

It gives the list of all usernames and their respective passwords. Both UserId and Password are primary keys.

**Sessions** *(SessionId, UserId):*

It records the session id associated with each and every user.
3.3 Entity – Relationship diagram

An entity relationship diagram shows the list of all possible entities and the relationships that exist among these entities [Pressman 2002]. Catalog contains the list of all possible courses and modules. A student can get enrolled in many courses and modules. A course will consist of many modules. Each student has only one course transcript that displays the grades obtained in all courses. Similarly, each student has only one module transcript that displays the grades obtained in all modules. These entities and relationships are shown in Figure 3.1.

![Entity Relationship Diagram](image)

Figure 3.1 Entity – Relationship diagram
3.4 Data Flow diagram

A data flow diagram serves two purposes [Pressman 2002]:

- Provides an indication of how data are transformed as they move through the System and
- Depicts the functions that transform the data flow [Pressman 2002].

The Level 0 DFD for the Electronic learning in Nursing education is illustrated in Figure 3.2. The faculty on login is presented with three options. It can register the course, calculate the grade and create file for registering students in WebCT.

Figure 3.2 eLine process
The Level 1 DFD’s for the Electronic learning in Nursing education is illustrated in Figure 3.3 and Figure 3.4. The course registration process is illustrated in Figure 3.3. When the student information is entered, it gets validated from the database. If the information is valid, and then the course number is entered. The modules associated with the corresponding course will be assigned to a particular student.

Figure 3.3 Course Registration Process
The grade calculation process is illustrated in Figure 3.4. The file associated with a module is uploaded. The file is parsed to get the module grade of a student and inserted in the database. If any course is completed, then course grade is calculated by taking the module weights.
The file calculation process is illustrated in Figure 3.5. Information is extracted from the database and stored in an output string. A file is created at an appropriate location on the server and is opened in write mode for displaying contents and in read mode to be sent as an email to WebCT administrator.

Figure 3.5 File Creation Process
### 3.5 Listing of Major Scripts

Figure 3.6 shows the listing of major scripts and their brief description.

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Script Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>index.php</td>
<td>Welcomes the users to the website.</td>
</tr>
<tr>
<td>2</td>
<td>login.php</td>
<td>Verifies the username and password.</td>
</tr>
<tr>
<td>3</td>
<td>changepassword.php</td>
<td>Allows the user to change his password.</td>
</tr>
<tr>
<td>4</td>
<td>AddStudentInfo.php</td>
<td>Allows the user to enter student information.</td>
</tr>
<tr>
<td>5</td>
<td>registerstudent.php</td>
<td>Enables a student to be registered for a particular course.</td>
</tr>
<tr>
<td>6</td>
<td>DropCourse.php</td>
<td>Allows a course to be dropped for a particular student.</td>
</tr>
<tr>
<td>7</td>
<td>CourseTranscript1.php</td>
<td>Displays the course transcript of a particular student.</td>
</tr>
<tr>
<td>8</td>
<td>ModuleTranscript1.php</td>
<td>Displays the module transcript of a particular student.</td>
</tr>
<tr>
<td>9</td>
<td>CurrentCourses1.php</td>
<td>Displays the courses student is currently enrolled in.</td>
</tr>
<tr>
<td>10</td>
<td>CurrentModules1.php</td>
<td>Displays the modules student is currently enrolled in.</td>
</tr>
<tr>
<td>11</td>
<td>CourseModuleGrade.php</td>
<td>Displays the module grades for a particular course.</td>
</tr>
<tr>
<td>12</td>
<td>liststudent.php</td>
<td>Displays the list of students in a particular</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>13</td>
<td>Moduleupload.php</td>
<td>Uploads the module grade from a file into the database and calculates the course grade if needed.</td>
</tr>
<tr>
<td>14</td>
<td>eLineWebCTRregistration.php</td>
<td>Creates file for registering students in WebCT.</td>
</tr>
<tr>
<td>15</td>
<td>logout.php</td>
<td>Logout from your account</td>
</tr>
</tbody>
</table>

**Figure 3.6 Scripts listing**

### 3.6 Security features

Security is one of the most important features today in any web database application because some of the information on the database needs to be kept confidential. Security implemented in this project can be explained by means of the following topics:

- Session Id
- One time key parameters
- Password authentication
- Using SSL(secure sockets layer)
- Lock out mechanism
- Backing up data
- Password expires

#### 3.6.1 Session Id:

Each user is identified by means of a Session Id. All the scripts that access the database first check whether the Session Id is still valid or not. If by any chance the
session is expired the user is directed back to the main login page. The sessions are stored in the database as soon as they are created and if they exist for an interval of more than 8 hours then they are deleted.

3.6.2 One time key parameters:

The concept of one time key parameters is used in order to make sure that any form is not re-submitted using browser’s back button that would cause previous submit information to be re-sent from browser’s cache. A one time key is generated as soon as the user goes to the home page. When the user logs in, the one time key is checked whether it is already used or not. If one time key is already used, then user is directed back to the main page.

3.6.3 Password authentication:

Passwords are encrypted and stored in the database using MYSQL function PASSWORD ( ). This function uses a non-reversible hashing algorithm.

3.6.4 Using SSL (secure sockets layer):

The protocol used is HTTPS and not HTTP. HTTPS is the secure version of HTTP. Instead of using plain text socket communication, HTTPS encrypts the session data using SSL protocol, thus ensuring protection from eavesdroppers. SSL will enable the server and the user’s browsers to communicate at the most secure level. The SSL layer exists between the transport layer and the application layer. It modifies the data from our HTTP application before giving it to the transport layer to send it to its destination.

3.6.5 Lock out mechanism:

Using this feature the number of attempts to login is restricted. After a certain
number of attempts, the user is not allowed to log in that will prevent the user from random guessing of username and password.

3.6.6 **Backing up data:**

The MYSQL database is backed up frequently by the systems administrator. This is done in order to recover from any disaster that forms an integral part of disaster recovery plan.

3.6.7 **Password expires:**

This mechanism is implemented in order to make sure that the username and password expires after a certain number of days. In order to know the new username and password the Systems Administrator need to be contacted.
4. TESTING AND EVALUATION

The eLine (Electronic Learning in Nursing Education) is a Web based system that will replace the manual paper-based system. The testing and evaluation of this system was a very important factor in the success of the system. A major effort was made to test the usability of the system and ensure that it is completely secure. There were many individual program modules involved in the project and each module was tested alone thoroughly in an attempt to discover any errors in its code.

4.1 Usability Testing

The system was designed to make sure that it is user friendly. This project was developed for the Nursing department of our University and Delmar college. Once the project was developed it was shown to users in the Office of Academic Affairs department of our University who work in close collaboration with the Nursing department. Feedback was taken from the users regarding the various interfaces of this system. Based on the feedback changes were made to the system to satisfy maximum users of this system.

4.2 Security Evaluations

Security was a major feature to be evaluated in this system. Each user is identified by means of a Session Id. All the scripts that access the database first check whether the Session Id is still valid or not. If by any chance the session is expired the user is directed back to the main login page. Thorough testing was done to check the working of this technique. The concept of one time key parameters is used in order to make sure that any
form is not re-submitted using browser’s back button that would cause previous submit information to be re-sent from browser’s cache.
5. RESULTS AND CONCLUSION

This is a Web based system implemented for eLine (Electronic Learning in Nursing Education) project. This project is very important because it will allow the staff to access the information of each and every student. Manual work for the staff is greatly reduced because the grade is calculated automatically. The goal is to design and develop a Web database application for the administrative staff of Delmar college and TAMUCC. The system will keep track of students registering for the nursing courses at both these schools. The information of students and their grades can be altered by the staff as and when they like.

This project accomplishes all the processes involved in the student registration system. This is done by creating a database to store information of all students in MYSQL and writing server side scripts using PHP to communicate with the database. The information in the database is modified as and when required. This site is designed in such a way so as to make it easy for any level of computer user to navigate. The project is evaluated by checking whether any changes made in the front end is reflected in the back end database.
6. FUTURE WORK

Regarding the future work that needs to be done, there were not many features that need to be implemented as the web application is complete. But sometimes modules assigned to a particular course change, weights assigned to modules also change and the grading criteria also change depending on what criteria is applied in the appropriate semester. So changes need to be made to the database and the scripts as and when required.

Apart from the security features implemented in this project, there can be many other security related issues that can be dealt in the near future. Security features can be enhanced with the help of the following:

- Access control
- Cryptography
- Integrated approach for database security and fault tolerance

6.1 Access control

Access control is the ability to permit or deny the use of an object such as a file by a subject such as an individual. Access control systems provide three essential services [Miller 2003]:

- Identification and Authentication
- Authorization
- Accountability

6.1.1 Identification and Authentication

It is a two-step process that determines who can log onto a system.
• Identification is how a user tells a system who he or she is (for example by username). The identification component of an access control system is normally a relatively simple mechanism based on either username or user id. Identification is usually based on Computer name, Media Access control (MAC) address, Internet Protocol (IP) address and Process ID (PID).

• Authentication is the process of verifying a user’s identity. It is based on one of these three factors:

1. Something we know, such as a password or a personal identification number (PIN). This assumes that only the owner of the account knows the password or PIN needed to access the account. Unfortunately, passwords are often shared, stolen, or guessed.

2. Something we have, such as a smart card or token. This assumes that only the owner of the account has the necessary smart card or token needed to unlock the account.

3. Something we are, such as fingerprint, voice, retina or iris characteristics. This assumes that the finger or eyeball attached to our body is actually ours and uniquely identifies us.
6.1.2 Authorization

Authorization defines a user’s rights and permissions on a system. After a user is authenticated, authorization determines what that user can do on the system. Most modern operating systems define sets of permissions that are variations or extensions of three basic types of access: Read (R), Write (W) and Execute (X). These rights and permissions are implemented differently in systems based on access control techniques that are either discretionary access control (DAC) or mandatory access control (MAC).

- Discretionary access control is an access policy determined by the owner of a file. The owner decides who is allowed access to a file and what privileges they have.
- Mandatory access control is an access policy determined by the system, not the owner. MAC is used in multilevel systems that process highly sensitive data, such as classified government and military information. A multilevel system is a single computer system that handles multiple classification levels between subjects and objects.

6.1.3 Accountability

Accountability uses such system components as audit trails (records) and logs to associate a user with his actions. Audit trails and logs are important for detecting security violations and re-creating security incidents. If no one is regularly viewing our logs and they are not maintained in a secure and consistent manner, they may not be admissible as evidence.
6.2 Cryptography

Database security can also be done from a cryptographic point of view. Modern cryptography technology can be integrated into relational database management systems to solve some major security problems. Cryptographic support is an indispensable ingredient for a modern RDBMS to provide a secure environment for storing and processing huge amount of business data. It is complementary to access control and both should be used to guide the storage and access of confidential data in a database system. There are three general approaches for integrating cryptographic support into an RDBMS [Wang 2001].

The first approach is loose coupling. A third party crypto service can be consulted by a database server and there are only minor changes on the server side. For example, a set of stored procedures can be pre-installed in the server. Each stored procedure provides a special crypto service to the database users by calling the crypto primitives supplied by the third party package. One example is an encryption PL/SQL package that encrypts a table column with a user supplied encryption key.

The second approach is tight coupling. A complete set of basic crypto primitives are built into the database server as a set of new SQL statements, together with the necessary control and execution context to ensure that those new SQL statements can be executed securely. This approach is a much harder task than the previous one in terms of implementation, but is preferable in the long run. The reason is simple: loose coupling is likely to open many security holes.

The third approach is somewhere in between of the above two. To accommodate the urgent need of security enhancement, only a small subset of crypto primitives are
integrated into the database server, based on which other services can be built using other
database utilities such as user defined functions and stored procedures.

6.3 Integrated approach for database security and fault tolerance

A systematic approach to integrate database security and fault tolerance can be
done so that the total number of overheads can be significantly reduced [Zhang 2004].
Fault tolerance is the ability of a system to respond to unexpected hardware or software
failure. An improved database encryption scheme is used for fault tolerance. Database
authentication by means of cryptographic checksums provides the integrity of data and
ensures the authenticated origin of data. A checksum is used to detect whether data has
been altered during transmission. Receiving programs recomputed the checksum to
compare with the checksum sent or stored with the data. Checksums are used for data
integrity in database records. All the checksum approaches aim to provide integrity of
data against illegal data manipulations, but can neither tell whether data was originated
from an intended data provider, nor can it correct any errors when occurred. Another
aspect of database security is database encryption. One disadvantage of this scheme is
that selections and projections cannot be performed without decrypting unneeded fields.
In order to avoid these pitfalls, database security and fault tolerance are integrated by
sharing cryptographic checksums. This significantly reduces the total overheads.
Redundant residue number system (RRNS) is used to generate additional fields that serve
as the checksums for error correction and database authentication. Unlike all other
checksum approaches, this approach is able to ensure whether a record is provided by an
intended data provider or not. Additionally, the integrated approach is able to detect and
correct a single error within each record. An improved database encryption scheme with a
consideration for fault tolerance can also be done that provides faster key generation and allows the primary key to be fault tolerant. The proposed approach is ideal for many database applications in which both authentication and fault tolerance are required such as the encrypted database backup and secure database communications in a client-server environment.
ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to Dr. Mario Al. Garcia, Assistant Professor of Computer Science, Texas A&M University – Corpus Christi, for his expertise in the area of database security, excellent guidance and encouragement during the period of this project work.

My sincere thanks to Dr. Dulal Chandra Kar, Assistant Professor of Computer Science, Texas A&M University – Corpus Christi, for his patience and timely help in answering my questions has been very helpful.

My sincere thanks to Dr. David Thomas, Assistant Professor of Computer Science, Texas A&M University – Corpus Christi, for his keen interest and support that motivated in completing my project.

And finally heartfelt gratitude to Lois Barry, eLine Project Manager and Rita Rushing, eLine Advisor for their moral support and warm wishes that helped me to finish the project in an efficient manner.
BIBLIOGRAPHY AND REFERENCES


APPENDIX A – DATA DICTIONARY

The database of the “Electronic Learning in Nursing Education” is named “eline” and it contains the following tables:

**Catalog:**

This table contains the information of all courses and the modules associated with all the courses.

- **CourseNo:**
  It represents the number assigned to a course. It will be the primary key of the table.

- **ModuleNo:**
  It represents the number assigned to a module.

**Course:**

This table contains the information of all courses along with their respective names.

- **CourseNo:**
  It represents the number assigned to a course. It will be the primary key of the table.

- **CourseName:**
  It represents the name of a course.

**Module:**

This table contains the information of all modules along with their respective names.

- **ModuleNo:**
  It represents the number assigned to a module. It will be the primary key of the table.
• ModuleName:
  It represents the name of a module.

CourseTranscript:
This table displays the course transcript for each and every student.

• StudentId:
  It represents the unique number assigned to each and every student. It is the primary key of the table.

• CourseNo:
  It represents the number assigned to a course.

• Grade:
  It represents the grade obtained by the student in a particular course.

• NumberGrade:
  It represents the numerical score obtained by the student in a particular course.

• StartDate:
  It represents the starting date when the student registers for a particular course.

• ImportDate:
  It represents the date when the course grade is calculated and entered into the database.

ModuleTranscript:
This table displays the module transcript for each and every student.

• StudentId:
  It represents the unique number assigned to each and every student. It is the primary key of the table.
• ModuleNo:
  It represents the number assigned to a module.

• Grade:
  It represents the grade obtained by the student in a particular module.

• StartDate:
  It represents the starting date when the student registers for a particular module.

• ImportDate:
  It represents the date when the module grade is calculated and entered into the database.

**StudentInfo:**
This table displays the information about each and every student.

• StudentId:
  It represents the unique number assigned to each and every student. It is the primary key of the table.

• FirstName:
  It represents the first name of a student.

• MiddleName:
  It represents the middle name of a student.

• LastName:
  It represents the last name of a student.

• School:
  It represents the school in which the student is enrolled. The student can be enrolled either in Delmar college or TAMUCC.
**Login:**

This table displays the information required to login.

- **UserId:**
  It represents the username of the user.

- **Password:**
  It represents the password for the particular username.

**Sessions:**

This table displays the session information for each and every student.

- **UserId:**
  It represents the username of the user.

- **SessionId:**
  It represents the session assigned for a particular user.

- **Timestamp:**
  It is useful for transaction reporting.
APPENDIX B – USER MANUAL

To access the web application, go to link https://www.eline.tamucc.edu/eline.
Then enter the Username and Password to access the different functionalities associated
with the application. The different functionalities can be accessed in the following
manner:

1. **Change Password:**

   To change password enter the Old Password and New Password. It replaces the
   Old Password by New Password. If the Old Password is entered successfully, then the
   message is prompted that the password has been changed successfully, otherwise a
   message is prompted to check the password and try again.

2. **Enter student information:**

   This process allows the faculty to enter the student information into the database.
   Enter Student ID, First Name, Middle Name, Last Name and School. The Student ID
   entered should be 9 digits long. Error messages will be displayed upon wrong input. If a
   Student ID already exists and the user tries to enter the same Student ID again then the
   user is prompted by a message whether to overwrite the new information with the old
   information.

3. **Register student for a course:**

   The faculty can register a student in a particular course using this option. The
   interface accepts parameters: Student ID and CourseNo. The student is registered in the
   appropriate CourseNo. If either Student ID or CourseNo is not present, then error
   messages will be displayed.
4. **Drop student from a course:**

   This process allows the faculty to drop a student from a particular course. The interface accepts parameters: Student ID and CourseNo. The student is dropped in the appropriate CourseNo. The interface will ask the user whether the student should be dropped with a grade “W” or not. Grade “W” represents withdrawal of a student from a particular course. If a student has already completed the course, then he cannot be withdrawn from a course. A student can be withdrawn from a course with or without grade “W”.

5. **Student course transcript:**

   This process displays the grades obtained by a student in all courses. If the student has not completed a particular course, then a grade of “I” is assigned. Grade “I” represents that the course is incomplete. The interface accepts parameters: Student ID or Lastname. Course transcript will also display the date when a particular course was started and completed.

6. **Student module transcript:**

   This process displays the grades obtained by a student in all modules. If the student has not completed a particular module, then a default grade of 0 will be assigned. Grade 0 represents that the module is incomplete. The interface accepts parameters: Student ID or Lastname. Module transcript will also display the date when a particular module was started and completed.
7. **List enrolled courses of a student:**

   The interface accepts parameters: Student ID or Lastname. List of courses enrolled by a student can be accessed either using Student ID or Lastname. It will list all courses currently enrolled by a student but not yet completed.

8. **List enrolled modules of a student:**

   The interface accepts the following parameters: Student ID or Lastname. List of modules enrolled by a student can be accessed either using Student ID or Lastname. It will list all modules currently enrolled by a student but not yet completed.

9. **Check module’s grade for a course:**

   This interface accepts either Student Id or last name and Course No. It will display the module grades obtained by a student in a particular course.

10. **List of students in a course:**

    This interface accepts the Course No and displays the list of students enrolled in that course.

11. **Upload file to enter module grade:**

    The user should upload the file associated with any module. If any course is completed, the grade is calculated and entered into the database. The process is repeated for all the student Id’s present in the file.

12. **Create file for registering students in WebCT:**

    A file is created using the information from the database. Just by clicking on the link a file is created and is sent as an e-mail to the WebCT administrator.