A Web-based System for Assignment Creation, Management and Peer Evaluation

GRADUATE PROJECT TECHNICAL 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

Vishal Monpara
Spring 2006

Committee Members

Dr. Dulal C. Kar
Committee Chairperson

Dr. David R. Thomas
Committee Member

Dr. Long-Zhuang Li
Committee Member
ABSTRACT

The prime objective of the proposed assignment management system is to assist instructors and students in maintaining assignments at Texas A&M University-Corpus Christi. This system has a central question bank, online assignment submission and peer evaluation with a rubric—a scoring tool that lists the criteria for a piece of work. None of the freely available tools provides functionalities for assignment creation, management and peer evaluation facility in one package. This system saves a considerable amount of time when instructors create and manage assignments. In addition, this system helps students learn from peer evaluation of their work and improve their skills.
# TABLE OF CONTENTS

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

Table of Contents.................................................................................................................... iii

List of Figures......................................................................................................................... v

List of Tables........................................................................................................................ vii

1. Background and Rationale................................................................................................. 1
   1.1 Introduction.................................................................................................................. 1
   1.2 Importance of Project................................................................................................. 1
   1.3 Intended Audience of the Application..................................................................... 2
   1.4 Similar Works........................................................................................................... 2

2. A Web based System for Assignment Creation Management and Peer Evaluation 4
   2.1 Overview..................................................................................................................... 4
   2.2 Features....................................................................................................................... 4
   2.3 Description of User Interface.................................................................................... 5
       2.3.1 Common Interface................................................................................................. 5
       2.3.2 Administrative Interface.................................................................................... 8
       2.3.3 Instructor Interface............................................................................................. 14
       2.3.4 Student Interface................................................................................................ 23

3. System Design.................................................................................................................... 29
   3.1 System Requirement................................................................................................... 29
       3.1.1 Server Configuration......................................................................................... 29
       3.1.2 Client Configuration.......................................................................................... 29
   3.2 Overview of the System............................................................................................. 29
   3.3 Main Components of the System............................................................................ 31
       3.3.1 Question Bank.................................................................................................... 34
3.3.2 Rubric.................................................................................. 32
3.3.3 Peer Evaluation..................................................................... 32

3.4 The Design of Application...................................................... 33
  3.4.1 Portal Architecture........................................................... 33
  3.4.2 Database Design............................................................... 34
  3.4.3 Security............................................................................. 35

4. Testing and Evaluation............................................................. 37
  4.1 Usability Testing................................................................. 37
  4.2 Security Testing................................................................. 37

5. Conclusion............................................................................. 39

Acknowledgements.................................................................. 40

Bibliography and References..................................................... 41
LIST OF FIGURES

Figure 2.1: Login Page................................................................. 6
Figure 2.2: Interface for Users......................................................... 7
Figure 2.3: Forgot Password Page...................................................... 7
Figure 2.4: Change Profile Page......................................................... 8
Figure 2.5: Select Semester Page....................................................... 9
Figure 2.6: Add Semester Page....................................................... 10
Figure 2.7: Create Semester Page..................................................... 11
Figure 2.8: Add Course Page.......................................................... 12
Figure 2.9: Add User Page.............................................................. 13
Figure 2.10: Assign Instructor to Course Page................................. 14
Figure 2.11: Add Student to Course Page............................................ 15
Figure 2.12: View Question Bank Page............................................ 16
Figure 2.13: Add Question Page...................................................... 17
Figure 2.14: View Assignment Page................................................. 18
Figure 2.15: Create Assignment Page............................................... 19
Figure 2.16: Select Question from Question Bank............................ 20
Figure 2.17: Add Question to the Question Bank............................... 20
Figure 2.18: Create Rubric Page...................................................... 21
Figure 2.19: Student Assignment Page............................................. 22
Figure 2.20: View Student Comment Page....................................... 23
Figure 2.21: View Assignment List Page.......................................... 24
Figure 2.22: View Assignment Page................................................. 24
Figure 2.23: Upload Assignment Page............................................... 25
Figure 2.24: View Peer Assignment Page......................................... 26
LIST OF TABLES

Table 3.1 Example of Rubric................................................................. 32
1. BACKGROUND AND RATIONALE

1.1 Introduction

Computers have been used in education system for years. With the advent of the World Wide Web in education systems, the use of computers radically changed the way of teaching. As the technology grew, new software for course management system came into existence. Many of these software products are commercial, like WebCT, BlackBoard, etc. Some colleges have developed their own course management system to meet their specific requirements. Many course management systems are available free to meet general requirements of any college. Unfortunately, none of the freely available software has capability of assignment creation, management and peer evaluation integrated in a single system. This system, having features like assignment creation, management and peer evaluation, benefits instructors as well as students.

1.2 Importance of Project

At Texas A&M University – Corpus Christi (TAMUCC), faculty members are using files to store assignments and tests that they create, so that they can refer to those questions later and create new assignment or test questions. TAMUCC faculty members are using paper to create, manage and grade assignments. Moreover, if any instructor is assigned to take another course, he/she has to create new assignments and test questions. Sharing of the question bank is limited due to paper based storing approach.

The main objective of the project is to provide a system which facilitates instructors in creating, managing and grading of assignments. This system has a central
question bank for each course. The instructor of the course can add, update or delete questions from the question bank. The instructor can choose questions from the question bank very quickly and have their assignment or test ready in no time.

The students can upload assignments to the system. They can upload assignments before the due date as many times as they want. After the due date, students can see their peer’s assignment. Peer evaluation is done by rubric—a scoring tool that lists the criteria for a piece of work [Andrade 1998].

1.3 Intended Audience of the Application

The system can be used at university to save time of instructors and students. This system primarily targets two categories of audience namely instructors and students.

The system aids instructors saving time for creating, managing and grading assignments. From the peer evaluation by students, instructors can have a fair idea about students’ assignments and hence their progress. From the statistics gathered by the system, instructors can improve course material.

The system enhances students’ skill in writing peer reviews and in learning from peer review. They will be free from managing paper based assignments. Due to the paper based assignment approach, the knowledge of students is not shared among their peers. This system facilitates student review of assignments completed by other students. They can get new ideas from other students as well as they can find their own mistakes and improve themselves.

1.4 Similar Works
There are many commercial systems available like WebCT and BlackBoard. All of these commercial software have many sophisticated features like upload course material, create quizzes, track student progress, chat tool, calendar event tool [WebCT 1997] [Blackboard 1997].

A system for assignment management and peer evaluation was developed using free available software at TAMUCC [Trivedi 2003]. It is helping instructors manage the assignments of students. This system adopts the architecture used by previously developed system and add module for assignment creation and refine module for rubric creation. It accommodates a new module for assignment and test creation.

WebCMS is a course management system developed at the University of North South Wales [Siew 2002]. This system provides modules such as calendar, group management and message board. This system gives common look and feel for all courses.
2. A WEB BASED SYSTEM FOR ASSIGNMENT CREATION MANAGEMENT AND PEER EVALUATION

2.1 Overview

The aim of A Web based System for Assignment Creation, Management and Peer Evaluation was to develop a Web based application which instructors can use in their classroom save their time as well as that of students. The instructor can create an assignment using the questions stored in a question bank. The instructor can create a rubric for a particular assignment. The assignment should be available to students for a certain period of time set by instructor. Students should upload their solutions. After the assignment deadline, student can view other’s assignment and if any rubric is created for that assignment, student has to fill the rubric. The rubric is used for partial grades for the assignment.

2.2 Features

This program is a Web-based system so it is accessible to students as well as instructors through computer having internet access. It has the following features:

- The administrator has privileges to create a new instructor as well as student login and could add courses offered by the instructor in a particular semester.
- The instructor is able to add students in his/her course.
- This system has a question bank for all courses. Each question may have an image. The questions may have variables. At the time of assigning a question to the assignment, the instructor must assign values to the variables. It gives
the instructor a power to create many questions from a single question by changing the value of variables.

- The instructor is able to add, update or delete question from the question bank for a particular course.
- The instructor can create a new assignment by selecting a question from the question bank.
- The instructor can create a new rubric for a particular assignment.
- The student can upload the assignment as many times as he/she wants before the due date.
- After the due date, the student can view other students’ assignment and use a rubric to evaluate a peer’s work.

2.3 Description of the User Interface

The user interface is a Web browser based. There are three types of users associated with this application. The first type of user is the administrator who handles courses and instructors. The second class of user is the instructor who handles assignments and its related tasks. The third type of user is the student who uploads solution of assignment. Thus all of the above users have there own interfaces. Moreover, there are some common interfaces which are used by all users.

2.3.1 Common Interface

Login Page

Every user requires a username and a password to enter into the system. There is a common Login Page for all users as shown in Figure 2.1. The form asks for the
username and password from the user. When the user presses the **Login button**, it checks for the validity of username and password. If the user identifying information is valid, according to the role (i.e. administrator, instructor or student) specified in the application to the user, he/she is redirected to appropriate page. If the identifying information is not valid, the user is transferred back to the **Login Page** with appropriate error message.

![Figure 2.1 Login Page](image)

**Figure 2.1 Login Page**

**Main Page**

Every user has the same **Main Page** components for the sake of a consistent look and feel of the application. There is a title of the component on the top. A user can perform his/her permitted tasks using two level menus. The top menu gives a general idea about the task. When the user clicks on the top level menu, he/she is offered with a left hand side menu which is a menu for logical grouping of the permitted tasks. A User can find his/her permitted tasks on the top menu as well as the left hand side menu. The
component currently used by the user is displayed in the middle. The Common layout is shown in Figure 2.2. If a user clicks on the left hand side menu, he/she gets the page to accomplish that task.

<table>
<thead>
<tr>
<th>Top Level Menu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title of Component</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User Menu</th>
<th>Component</th>
</tr>
</thead>
</table>

**Figure 2.2: Interface for Users**

**Forgot Password Page**

The user is asked to input his/her username and press the **Submit button**. The password is sent to his/her email.

**Figure 2.3 Forgot Password Page**
**Change Profile Page**

Once the user logs into the system, his/her full name is displayed on the top right hand. When the user clicks on it, he/she is directed to his/her profile page as depicted in Figure 2.4. This page shows a form which contains password, email, phone number, address, SSN etc. After changing the profile, a user has to press the **Save button** to save the information to database.

![Figure 2.4 Change Profile Page](Image)

**2.3.2 Administrator Interface**

The administrator is responsible for managing the application specific settings. The administrator uses an internet enabled computer / laptop and goes through these various Web pages.

**Administrator Main Page**
The Administrator main page shows the options to manage departments, semesters, courses and users. The administrator can change profile of any user from this page and, finally, logs out of the system.

Select Semester Page

This system is divided into the basic building blocks of an education system such as semesters and courses. This system automatically detects the current semester. Until and unless the administrator selects other semester for his/her session, all the commands are applied to the current semester. To change the selected semester for the session, the administrator visits this page which is shown in Figure 2.5.

![Select Semester Page](image)

Figure 2.5 Select Semester Page

Create Semester Page

This page lists all available semesters as shown in Figure 2.6. To modify the existing semester, the administrator selects the semester from the dropdown list. The rest
of the information is populated from the saved information. The administrator can create a new semester by specifying the semester name (e.g. Fall 2005, Spring 2006 etc.), start date and end date.

Create Department Page

The administrator can add/modify/delete any department from this page as shown in Figure 2.7. This page requires department name and the prefix name to be used for the course for a particular department. For example, Computer Science courses might have a prefix like “COSC”. One department can have more than one prefix.
In this page, the administrator can see a list of courses by selecting the appropriate department. The administrator can edit any course listed or create a new one. This page contains a form which asks for the course name, course number and description as shown in Figure 2.8. The administrator fills information and presses the **Save Course button**.
On this page, the administrator is able to modify as well as create user login as shown in the Figure 2.9. This page contains a form where the administrator can enter the role of the user i.e. Administrator, Instructor or Student, social security number (SSN), name, initial password, email address and phone number of the user and press the **Save** button which saves the details of that user account into the database. The same page can be used to modify the existing user profile.
Assign Instructor to Course Page

At this page, the administrator first selects the available semesters. It shows department list as shown in Figure 2.10. Once the administrator selects a particular department, he/she is given a list of all courses and instructors. The administrator can change the instructor assigned to a course or he/she can add new course for the selected semester and assign instructor to it. He could then select the name of an instructor and course, then hit the **Save button**. This action assigns an instructor to a selected course.
2.3.3 Instructor Interface

The instructor is responsible for adding the students to the course, creating the assignments and maintaining the question bank. The instructor can go through these pages to do his/her tasks.

Instructor Main Page

The Instructor Main Page shows a menu for adding students to a course, adding questions to the question bank, creating an assignment, creating a rubric and viewing students’ assignments.

Add Students Page

When the instructor visits this page, he/she has to first select the course for which he/she wants to view all students enrolled for that course as shown in Figure 2.11. When he/she selects a course, he/she can see the list of all students who are taking the selected
course. At the same time the instructor can add a student from the list of students. By
default, the instructor can see the courses which are offered by him in the current
semester. If he/she wants to see all his/her courses offered in previous semesters, he/she
has to select other semester for his/her current session as explained previously. The
instructor can select a student from the list. On hitting the **Add Student button** student
will be added to the course. If the student is already registered in that course, the
application shows the appropriate error message.

![Image of Add Student to Course Page](image-url)

**Figure 2.11 Add Student to Course Page**

*View Question Bank Page*

This page shows the question bank to instructor for only those courses for which
he/she is currently assigned as shown in Figure 2.12. Though the instructor has changed
the selected semester to the previous semester, he/she cannot access the question bank for
the course which he/she was taking in previous semesters. The instructor can view all the
questions in the question bank for the selected course. If he/she wants to edit the question or view the full description of the question, he/she should press the **Edit button** given adjacent to the description of the question.

![Figure 2.12 View Question Bank Page](image_url)

**Add Question Page**

This page gives the instructor an access to add new question to the question bank for the course he/she is assigned in the current semester as shown in the Figure 2.13. First the instructor selects the course for which he/she wants to add new question. The instructor writes description of the question. The question can accommodate variables to make it very flexible for the instructor to create new questions on the fly from the template. If the instructor wants to have a variable in question, he/she should write the variable name within “$$” sign in question description. Once the instructor writes a
proper description which includes variables, he/she presses **Process Variable button**.

The application extracts all the variables from the question description and shows it to the instructor for further process. By default the data type of each variable is integer and the Hi and the Low value of the variable is 0. If the instructor wants to change the variable type, lower and upper bound of the value of the variable, he/she can click on the **Update button**. This action gives the instructor a space to set values for a particular variable. If the instructor wants to attach an image to the question, he/she can upload the image by clicking on the **Browse button**. Now the instructor presses the **Save Question button** and the question is saved. If the instructor does any mistake, appropriate error message is shown.

![Figure 2.13 Add Question Page](image)

*Figure 2.13 Add Question Page*

*View Assignment Page*
As shown in the Figure 2.14, this page lists all the assignments for a selected course. It shows the assignment name and few icons which redirects the instructor for different tasks related to that assignment. All these tasks include Edit assignment, Modify questions, Edit rubric, View the answers uploaded by the student and delete the assignment.

**Figure 2.14 View Assignment Page**

**Create Assignment Page**

The instructor provides an assignment name, start date, due date, rubric start date, number of students to evaluate a particular assignment, total points and then press the **Save button**. This action creates an assignment as shown in Figure 2.15. Now the instructor is redirected to the **Assignment List Page**. Here the instructor can see a newly added assignment. Now the instructor presses a button to view the assignment. The instructor can see the question bank to add a question to the assignment as shown in the
Figure 2.16. The instructor can see a list of question along with selection icon and checkbox. If the instructor wants the application to randomly generate the variable value, he/she should click on the checkbox and press the **Add Question button** else he/she should press the **Select Icon**. It leads him/her to the page where he/she can see the question description as shown in the Figure 2.17. If the question contains a variable, the instructor is asked provide value for that variable as well as grade. A question in the question bank might contain an image. If the selected question contains an image, the instructor can write whatever he/she wants on the image. Now the instructor presses the **Add Question button**.

![Figure 2.15 Create Assignment Page](image-url)
Figure 2.16 Select Question from Question Bank

Figure 2.17 Add Question to the Question Bank
Create Rubric Page

This page provides a form for creating a rubric for a particular assignment as shown in the Figure 2.18. The instructor can navigate to this page by pressing the Rubric Icon on the Assignment View Page. It asks for criteria using which students evaluate the work of their peers. The instructor inputs the criteria and then hits the Add button.

![Create Rubric Page](image)

**Figure 2.18 Create Rubric Page**

View Student Assignment Page

At this page, the instructor can view the list of assignments uploaded by students and download it as shown in the Figure 2.19.
Figure 2.19 Student Assignment Page

View Student Comment Page

When the instructor presses **View Student Comment Icon**, he/she can visit this page as shown in the Figure 2.20. This page shows the instructor in detail all the comments about a particular student as well as all the ratings from other students according to the rubric. It helps instructor evaluate the assignment of the student and to decide for the partial grade for the assignment.
2.3.4 Student Interface

Students are given privileges to view the assignments in a course, upload an assignment and evaluate a peer’s work. To accomplish these tasks, students can view following pages.

Student Main Page

Here a student can view a menu which shows options to view assignments, upload assignments and evaluate peer work.

View Assignment Page

This page shows the assignment posted by the instructor by selecting a course as shown in the Figure 2.21. When the user presses View Assignment Icon, he/she is directed to the page where the student can view all the questions and its score as shown in the Figure 2.22. A student can download the assignment in Microsoft Word format by
pressing **Download button**. The student can do multiple tasks on the assignment by pressing different icons.

**Figure 2.21 View Assignment List Page**

**Figure 2.22 View Assignment Page**
Upload Assignment Page

This page gives a student access to upload his/her assignment before the due date. The student specifies the assignment filename to be uploaded and presses the Submit button as shown in the Figure 2.23. If the due date is passed, then the student cannot upload his/her assignment. The student can review, make changes and upload his/her assignment as many times as possible before the due date. When the student uploads the assignment, the application generates new Globally Unique Identifier (GUID) and saves the assignment file using it.

![Upload Assignment Page](image)

Figure 2.23 Upload Assignment Page

View Peer Assignment Page

This page shows the list of all the assignments uploaded from student’s classmates after the due date of the assignment is passed. As shown in the Figure 2.24, a student can click on the button to download the assignment. Here a student cannot
recognize which assignment belongs to whom. If a student wants to grade any assignment based on the rubric created for the assignment, he/she presses the **Select Icon** and it leads the student to the Peer Evaluation Page.

![Peer Evaluation Page](image)

**Figure 2.24 View Peer Assignment Page**

*Peer Evaluation Page*

This page shows the student the rubric created by the instructor for the assignment. The student must evaluate the rubric on the scale of 1-5. A student can give a grade, write down some comment and press the **Save button** as shown in the Figure 2.25. Once the student evaluates the rubric for any student he/she cannot edit it.
This page shows the student his/her own assignment evaluation and comments posted by his/her peers as shown in the Figure 2.26. This page gives feedback about the student’s work and hence can improve his/her work.
<table>
<thead>
<tr>
<th>Task</th>
<th>View Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Comment</td>
<td></td>
</tr>
</tbody>
</table>

**Assignment**

<table>
<thead>
<tr>
<th>Description</th>
<th>Student 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is the idea clearly indicated</td>
<td></td>
</tr>
<tr>
<td>Demonstrated knowledge in understanding the questions</td>
<td>5</td>
</tr>
<tr>
<td>Well organized</td>
<td>5</td>
</tr>
</tbody>
</table>

**Comments**

Excellent work. Keep it up. I got some new ideas by your method of solving problems.

Figure 2.26 View Comment Page
3. SYSTEM DESIGN

3.1 System Requirement

This application is a Web-based, so it follows the client-server architecture. The application is deployed on the server. The client (i.e. user) makes request to the server. The server executes the request and sends the response back to the user. To operate the system in the normal way, the following configurations are required.

3.1.1 Server Configuration

The server is a machine having Internet Information Server (IIS) installed. The operating system of the computer is Microsoft® Windows XP Professional or Windows Server. This server has Microsoft® .Net 1.x framework installed. The server also has Microsoft® SQL Server / Microsoft Database Engine (MSDE) as a backend.

3.1.2 Client Configuration

The client machine can be any personal computer or Macintosh computer provided with an internet connection and a Web browser. The recommended Web browsers are Internet Explorer 5.x or above or any equivalent browser.

3.2 Overview of the System

The application is the Web-based system and use Hyper Text Transfer Protocol (HTTP). The Web pages are dynamically generated by the Active Server Pages (ASP) .Net. The ASP .Net is a part of the Microsoft® .Net framework. The MSDE is used as a backend database server. Scripts written in the ASP .Net process the input given by the user access the database and return the result as the Hyper Text Markup Language
(HTML) page to the user. The Structured Query Language (SQL) is used to access data from MSDE.

Figure 3.1 shows the abstract view of the working of the application [Webb 2003]. The user types the Uniform Resource Locator (URL) of system’s Website into his/her browser. The browser sends a request to the server. The server forwards this request to the .Net Runtime for processing. The .Net Runtime executes the ASP .Net script requested by the user. The ASP .Net script processes the input provided by the user, uses the SQL to fetch data from the database and generates an HTML page.

![Figure 3.1 Abstract System Model](image-url)
The solid arrows describe the processes seen by the end user. The dotted arrows placed in the Figure 3.1 describe the flow of process internally, which is not visible to the end user.

### 3.3 Main Components of the System

None of the freely available course management system provides assignment creation, management and peer evaluation as a single system. This system has three main components to achieve the goal.

#### 3.3.1 Question Bank

This application has a question bank where questions for different courses are stored. Only the instructor of the course has a permission to view and change the questions. The important feature about these questions is that a question can have a variable and an image. The instructor has to specify the variable using the delimiter “$$”. The instructor is prompted to set properties of the variables when he/she adds a question to the question bank. The instructor can specify the Minimum Value and the Maximum Value of the variable. Each variable has its own data type (i.e. Integer, Float). The instructor can also insert formula of variables in the description of the question. The formula should be contained within “[[” and “]]” sign. The value of the variable can be inserted by the instructor at the time of adding the question to the assignment or the instructor can choose to automatically generate a random value ranging between the Minimum Value and the Maximum Value for that variable. Depending on the value of variables the formula is evaluated and the value of the formula is automatically inserted in the question description. An image can be attached to a question. When the instructor
chooses a question for the assignment from the question bank, he/she has the option to put the value on the image at any (x, y) position in the image. An image with the value can be generated on the fly. The instructor can put as many variable values as he/she wants on the image. When the instructor finally submits the question, the unique question is generated for the assignment using variable values and an image attached to the question. The feature of assigning different values to the variables and writing values on the image makes a question unique for a particular assignment. The instructor can generate as many questions as he/she wants from a single question by assigning different values to the variables and images. This feature reduces the time of the instructor for creating an assignment, its related images and for reuse of the question.

3.3.2 Rubric

Rubric is a scoring tool that lists the criteria for a piece of work [Andrade 1998]. It lists different criteria for any work. The evaluation of the rubric is done by giving grade to work considering each criterion from excellent to poor. Figure 3.2 shows the example of a rubric. The left hand side column of a rubric has criteria. The right hand side columns describe varying degree of quality, from excellent to poor.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demonstrated knowledge in understanding the questions</td>
<td></td>
</tr>
<tr>
<td>Well organized</td>
<td></td>
</tr>
<tr>
<td>Thoughtful use of well chosen evidence</td>
<td></td>
</tr>
<tr>
<td>Free of errors</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1 Example of Rubric

The rubric is created by the instructor at the time of creating the assignment. To help the instructor creating rubric, generic templates is shown. The instructor can use any of these templates and modify the template to create a unique rubric for the assignment.
The instructor can even save the rubric as a template for later use. There is no limit on the number of criteria of the rubric.

3.3.3 Peer Evaluation

As a part of the assignment, the student must evaluate peer’s work. At the time of creating an assignment, the instructor chooses how many assignments a student can evaluate. When students login to the system, they see a list of assignments of their classmates. A student randomly picks any assignment and evaluates it based on the rubric created by the instructor. If the student has not evaluated enough assignments, he/she is prompted for evaluating peer work. This process is done anonymously.

3.4 The Design of the Application

The database is the main component in this application. A highly structured and efficient database is required in the application. If the database is not properly designed, it will be difficult to build the application. In addition, expansion of the application is limited if the database design is poor. The database for this application is designed with the use of Entity Relation (ER) diagram. Figure 3.2 shows the ER diagram for this application.

3.4.1 Portal Architecture

This application is developed as a portal. According to the application requirement, there are several tasks to be accomplished using this system. Different tasks are logically grouped and each group is represented by a unique name. This unique name is considered as a tab. Each tab is given a privilege of being one of the main menu items in the layout of the application. As each tab represents logical group of tasks, each tab is
further divided into modules. Each module represents an individual task. Each module is represented as a sub menu in the layout of the application. Many modules are hidden so it is not shown as one of the menu items but they are implicitly accessed using several buttons.

This application assigns role to every user. There are total three user roles namely Administrator, Instructor and Student. Each role is given authorization to access certain tabs and modules of the application. If the user tries to access tabs or modules on which his/her role does not allow him to access, the application prompts for access denial. This portal also partially supports skinning of the application. Skinning is a programming technique in which one can change the look and feel of the application without changing the code and recompiling the application. Two files Default.aspx and Styles.css can be changed to change the look and feel of the application keeping in mind to place all the necessary user controls in the newly developed files. When the user logs into the system using his/her username and password, the application checks his/her associated role and fetches the settings of the application. Different tabs and modules are displayed according to the fetched settings. One important feature of this application is its extensibility. As the application is divided into modules, the application can be extended by creating modules and without worrying about other aspects of the application. Once the module is developed, it can be inserted into the system very easily by manipulating the database and assigning appropriate access right to the different roles.

3.4.2 Database Design

The database is designed to achieve efficient minimal storage and quick access. Each table is given primary key for efficient retrieval of the data. Relationships between
tables are properly defined to maintain referential integrity [Date 1995]. This database consists of 26 tables. To fetch the data from the database, 110 stored procedures are written. The E-R diagram of this application is shown in the Figure 3.2. This diagram shows only core table relationship.

Figure 3.2 Entity Relationship Diagram

3.4.3 Security
This application requires special care to implement security of the application. The basic security is implemented by restricting access the application to only those users who has the correct username and password. As the application is built as a portal, portal framework gives access to only those modules which are authorized by a role of the logged user. If any user tries to access the module which is not permitted by his/her role, he/she is shown an appropriate error message. This application prevents two kinds of attacks namely SQL injection and URL manipulation. These kinds of attacks are very common. To prevent the SQL injection attack the application uses stored procedures to fetch data from the database, so the chances of SQL injection are decreased. Moreover, each and every input taken from the user is checked. A user can enter only alphabets, digits and few special characters. URL manipulation attack is prevented by checking the URL parameters in each and every module. Before the actual data becomes visible to the user, the application checks if the user has right to view it or not. If user is not permitted to view the item, an appropriate error message is displayed.
4. TESTING AND EVALUATION

The system is a Web based system, so it is accessible to anyone who knows the URL of the application. The administrator, instructor and student profile is stored in database. The student grades are also stored in database. This application also stores student assignments in a file. The testing is a major component in the success of the application. Software testing and usability testing techniques are used to test this application [McGraw Hill 2000].

4.1 Usability Testing

The success of the application lies in the fact how the user of the system thinks about the system. No matter how robust the system is built, if the user is not comfortable with its use, the system fails. So the usability testing is one of the major testing of the application. This testing was performed by getting feedback from the users (i.e. instructors and students) of this system. The feedback has been taken for the user interface and usability. Based on this feedback, changes are made to the system.

4.2 Software Testing

This application consists of portal architecture and three main modules. The portal architecture was developed and tested separately from the modules. The main modules are supported by other modules. All the modules have been tested individually to check if the application met the requirements of the intended audience. All modules are merged after testing. After module testing, the overall functionality, and efficiency of the
application was tested. This application is written in such a way that it does not require any maintenance and could be maintained by any computer novice. This application was initially tested by the developer.
5. CONCLUSION

This project is important for the instructors and students of the university. This project replaces the manual system of assignment creation and management with the Web based system. This system facilitates the instructor share the question bank and reduces the time for creating assignments. It also provides a method by which students can manage their assignments. This system helps students learn from peer work and peer evaluation.
ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to Dr. Dulal Kar, Associate Professor of Computer Science, Texas A&M University- Corpus Christi, for his excellence guidance, encouragement and his valuable suggestions throughout the period of this project work.

My sincere thanks to Dr. Long-zhuang Li, Assistant Professor of Computer Science and Dr. David Thomas, Associate Professor of Computer Science and Mathematics for their keen interest, support and motivation in completing my project.

I take this opportunity to thank faculty and students of Department of Computer Science and Mathematics at Texas A&M University-Corpus Christi for taking part in the usability testing and providing their valuable feedbacks which helped in the betterment of this project.
BIBLIOGRAPHY AND REFERENCES


