Developing Instructional Technology

Using the

Tablet PC

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

Sunil M Kilaru
Summer 2007

Committee Members

Dr. Dulal C. Kar
Committee Chairperson

Dr. Long-Zuang Li
Committee Member

Dr. Jennifer Smith-Engle
Committee Member
Developing Instructional Technology
Using the
Tablet PC

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

Sunil M Kilaru
Summer 2007

Committee Members

Dr. Dulal C. Kar
Committee Chairperson

Dr. Long-Zuang Li
Committee Member

Dr. Jennifer Smith-Engle
Committee Member
ABSTRACT

Mobile PC applications including those for Tablet PCs are evolving as people seek new ways to be productive anywhere, anytime. A Tablet PC incorporates the convenient and intuitive aspects of natural input including using a pen and speech. The Tablet PC is an exciting new platform for education. Tablet PCs support active learning in the class. Rather than following a text, Tablet PCs allow students to explore and discover, to discuss with other students, and to more carefully reflect on their own learning. Digital notes increase the efficiency of students as they can carry their notes with them everywhere, and they can annotate documents. Digital ink plays an important role in this because of the flexibility that writing provides to students.

The purpose of the application is to improve student learning by developing instructional technology with the use of a Tablet PC. This is achieved by adapting more directly to the Tablet PC environment. The application is Ink enabled, where a student writes on a Web browser. The Web browser application grabs the Ink and saves it on the server. The project provides an environment where students share their knowledge with each other and interact with the instructor. The application allows the instructor to structure by components of a class and effectively present information and receive information from students.
TABLE OF CONTENTS

Abstract .................................................................................................................. ii
Table of Contents ................................................................................................ iii
List of Figures ........................................................................................................ v

1. Background and Rationale .................................................................................. 1
   1.1 Introduction ................................................................................................ 1
   1.2 Importance of Project .............................................................................. 2
   1.3 Intended Audience of the Proposed System ........................................... 5
   1.4 Similar Works .......................................................................................... 6

2. Developing Instructional Technology Using PC Tablet ...................................... 7
   2.1 Overview .................................................................................................. 7
   2.2 Features ................................................................................................... 7
   2.3 Description of User Interface ................................................................... 8
      2.3.1 Common Interface ........................................................................ 8
      2.3.2 Administrative Interface .............................................................. 11
      2.3.3 Instructor Interface ...................................................................... 21
      2.3.4 Student Interface ........................................................................ 31

3. Proposed System Design ................................................................................... 40
   3.1 System Requirement ................................................................................ 40
      3.1.1 Server Configuration ................................................................... 40
      3.1.2 Client Configuration ................................................................... 40
   3.2 Overview of the System ............................................................................ 41
   3.3 Main Components of the System ............................................................. 43
3.3.1 Question Bank ................................................................. 43
3.3.2 Creating and Uploading Lectures Notes .............................. 43
3.3.3 Peer Review ................................................................ 44
3.3.4 Messaging .................................................................. 44
3.3.5 View Report ................................................................. 44

3.4 The Design of Application .................................................. 45
3.4.1 Portal Architecture ...................................................... 45
3.4.2 Database Design .......................................................... 46
3.4.3 Security .................................................................... 49

4. Testing and Evaluation .......................................................... 50
4.1 Usability Testing ............................................................... 50
4.2 Security Testing .............................................................. 50

5. Expected Results and Conclusion ......................................... 51

Acknowledgements ................................................................. 52
Bibliography and References ...................................................... 53
Appendix 1 – InkControl.dll file ................................................. 54
Appendix 2 – Web Configuration file ........................................... 58
LIST OF FIGURES

Figure 1.1: Tablet PC. .................................................................1
Figure 1.2: Existing System Scenario I ...........................................2
Figure 1.3: Existing System Scenario II ..........................................3
Figure 1.4: New System .............................................................4
Figure 2.1: Login Page ...............................................................9
Figure 2.2: Incorrect Login .........................................................10
Figure 2.3: Content Display Page ...............................................11
Figure 2.4: Administrative Default Interface .................................12
Figure 2.5: Create Semester ......................................................13
Figure 2.6: Create User ............................................................14
Figure 2.7: Create Courses .......................................................15
Figure 2.8: Edit Courses .........................................................16
Figure 2.9: Assign Courses to Instructor .....................................17
Figure 2.10: Edit Assigned Courses to Instructor .........................18
Figure 2.11: Assign Students to Course .....................................19
Figure 2.12: Edit Assigned Students to Course .........................20
Figure 2.13: Instructor Default Page .......................................21
Figure 2.14: Add Question Page ..............................................22
Figure 2.15: Select Course Page ..............................................23
Figure 2.16: Create Quiz Page .................................................24
Figure 2.17: Upload Lecture Notes ...........................................25
Figure 2.18: Uploaded Work by students for Peer Review ...........26
Figure 2.19: View Messages
Figure 2.20: Select Student Name to View Quiz Report
Figure 2.21: Quiz Report
Figure 2.22: Quiz Report by Grade
Figure 2.23: Student Default Page
Figure 2.24: Select Quiz Course
Figure 2.25: Student Quiz Page
Figure 2.26: Download Lecture Notes
Figure 2.27: Upload Work
Figure 2.28: Peer Review
Figure 2.29: Student Messaging Page
Figure 2.30: Select Course to View Quiz Report
Figure 2.31: Quiz Report
Figure 3.1: Abstract System Model
Figure 3.2: Entity Relationship Diagram
1. BACKGROUND AND RATIONALE

1.1 Introduction

Educational advantages of Tablet PC over full-sized, varied functioning computers range from cost to ease of use. Students and instructors can organize their notes in ways that make most sense to them, and they can find all the references to any given word or phrase in seconds. As a result, students find studying for exams easier, and they respond more quickly in class. This project focuses on student learning, by providing means of interaction between instructor and student and sharing of knowledge between students and their peers.

The system provides the features of creating online quizzes, uploading of lecture notes, downloading of notes by students and peer evaluation (where students can see each other’s work) and posting of messages. In addition, instructor can view students’ performance in quizzes and their uploaded work for peer review.

Figure 1.1 Tablet PC
The system creates a pathway where student and instructor interact with each other. Students also interact with peers by reviewing their work. Instructor gets the fair idea of students’ grasp on the subject by reviewing students’ performance in quizzes and the work submitted.

The usual cost of a Tablet PC ranges from $1,000 to $3,000 depending upon the hardware configuration and software bundled with the system. Mobility is one of the biggest advantages one can have in using a Tablet PC. Tablet PCs can be taken practically anywhere without the user being confined to a computer lab or classroom. The ability to transfer or share data wirelessly overcomes the need for a hardwired infrastructure and adds to the mobility.

1.2 Importance of Project

Figure 1.2 Existing System Scenario I
As shown in Figure 1.2, the instructor creates lecture notes, uploads them on the Web or distributes the printouts of the notes in the class. The students carry the printout with them, annotate the notes, and use them as a primary source of reference material. As the notes are in printed format, there is always a possibility of losing them.

![Figure 1.2 Lecture Notes Distribution](image)

Figure 1.2 Lecture Notes Distribution

As shown in Figure 1.3, it is important to assess the understanding of subject matter by students. The instructor creates quizzes, and hands them out.
during the class time. The quiz takes a lot of instruction time away from the class, and assessment of quizzes takes extra effort by the instructor.

As shown in Figure 1.4, the objective of the project is to facilitate interaction between instructor and students. The instructor uploads lecture notes for the class. Students peruse through the notes. For better understanding of the subject matter, they go through a variety of books, visit different websites, and compile a unique text. This text acts as the knowledge bank. If the text is shared between peers, every student benefits from each other’s work.
An instructor creates an online quiz. The idea behind the quiz is to enable the instructor to view the acquired knowledge by students.

Students can post messages to the instructor and discuss their point of view on some particular topic covered in the class. In such a case a student can just write with a pen on a Tablet PC and post messages to the instructor. The instructor can see the point of discussion on the particular topic and could include them in lectures if the information is interesting and useful.

The aim of the project is to facilitate tools for students and instructor the tools to explore the potential of Tablet PC, to help make computing content and instruction more engaging and effective, resulting in higher success rates for students.

1.3 Intended Audience of Proposed Application

The audience of the system includes students and instructors. The system can be implemented at Texas A&M University – Corpus Christi computer science department.

The system provides the tool for managing questions, creating quizzes, viewing quiz results, uploading and downloading of lecture notes, and messaging.

The proposed system enhances the students’ capability in organizing their notes and learning a new point of view from peer review. New ideas are shared, which helps students in understanding the subject to a great detail.

Instructors can use the system to improve their course materials. Students can also view their performances on a particular quiz.
1.4 Similar Works

A system for assignment management and peer evaluation was developed using free available software at TAMUCC [Trivedi 2003]. This system helps instructors manage the assignment of students.

There are other commercial systems available such as WebCT and BlackBoard [WebCT] [Blackboard 1997]. WebCMS is a course management system developed at the University of New South Wales [Siew 2002].

There are many educational organizations that have implemented or are in the process of implementing a similar kind of utilization of PC Tablets [2006 Microsoft Corporation]. A few of them are Massachusetts Institute of Technology (MIT), Northwestern University, Rochester Institute of Technology (RIT), and Kansas State University.

In contrast to all other systems, the proposed system concentrates on the Tablet PC but is not confined to it. Ink enabled Web application makes a useful tool when it comes to the Tablet PC, as the user can just use pen instead of keypad. The system supports active learning in the class. Rather than just following the text, it allows students to explore and discover, to discuss with other students, and to more carefully reflect on their own learning.
2. Developing Instructional Technology
Using Tablet PC

2.1 Overview

The ambition of “Developing Instructional Technology Using Tablet PC” was to facilitate the tools for the instructor and students to explore the potential of Tablet PC. The application is a Web based application for Tablet PC which instructors can use in their classroom and save their time as well as that of students.

The application provides the interactive tools for course management, quizzes, peer review, and messaging.

2.2 Features

This application is a Web based system; it is accessible to students as well as instructors through PC’s having an Internet access. It has the following features:

- The administrator has the privileges to create new logins, add courses, assign student and instructor to courses for a specific semester.
- The instructor can add questions to a question repository for courses.
- The instructor is able to create new quizzes.
The students are able to download lectures, update their work, and upload them for peer review. A partial credit can be given for the peer reviews.

2.3 Description of the User Interface

The user interface is a Web browser. Three types of users are associated with this application. Each user is role based and has different interfaces. The roles of user are Administrator, Instructor, and Student.

The administrator manages courses, instructors, students and semesters. An instructor is able to create quizzes, upload the lectures, view peer reviews, and messages, and see the report per course. Students then take quizzes, download lectures, review them, and upload their work them for peer reviews. Moreover, there are some common interfaces which are used by all users.

2.3.1 Common Interface

All users have some common screens; these screens have fairly common information. These common interfaces guide a user to different screens based on the login privileges.
Login Page

Every user has to authenticate in order to enter into the system. There is a common Login Page for all users. On the form, the Username and Password have to be filled in. If the provided information is valid, the user is directed to the appropriate page based on the role. If the user has privilege of more than one role (administrator, instructor or student), all of the roles are displayed on the top, where the user can select the role he/she wants to browse. If the information is not valid, the user is transferred to the login page with an appropriate error message.

Figure 2.1 Login Page
Figure 2.2 Incorrect Login

*Login Specific Main page*

The main page varies according to the roles assigned or chosen by the user. On the top menu the user’s login and role information are displayed. Whenever the user clicks on the role, the user is directed to the particular role’s main page. The sub menu component is displayed according to the role selected. The contents are displayed in the center, as shown in Figure 2.3. On the top of the page a welcome message followed by the user’s first name is displayed.
2.3.2 Administrative Interface

The administrator is responsible for managing and creating logins, creating semesters and courses. The administrator plays a very key role in this application.

Administrative main page

The administrator main page has options to create an instructor account, add courses, assign courses to an instructor, create student logins, and assign student logins to the courses.
Create semester page

This page lists all available semesters. The administrator can create a new semester by specifying the semester name and year, as shown in Figure 2.5.
Create users account page

On this page, the administrator is able to create a user’s login. The administrator creates a user’s login and assigns privileges based on the request submitted. The administrator can assign the user privileges to Admin, Instructor or Student, as shown in Figure 2.6.
Create course page

In this page, the administrator views a list of courses. The administrator can edit any course listed or add a new one. This page contains a form that requires call numbers, course and course title, as shown in Figure 2.7. All this information is saved in the database.

Figure 2.6 Create User
The administrator can also edit a previously created course. The editing of the course is based on the call numbers, as shown in the Figure 2.8.
Assign courses to instructor

In this form the administrator assigns courses to instructor. The administrator selects semester, Instructor and assign courses to the instructor, as shown in Figure 2.9.
Editing assigned courses to instructors

The administrator can also edit the courses assigned to an instructor. The administrator selects semester, instructor, and edits the assigned courses, as shown in Figure 2.10.
Figure 2.10 Edit Assigned Courses to Instructor

Add students to courses and semester

The administrator assigns students to the selected course and semester, as shown in Figure 2.11. The usernames of the students are obtained based on role assigned.
Edit assigned students to courses

To edit the course assigned to students, administrator selects semester, course, and highlights the student’s username and clicks on update, as shown in the Figure 2.12.
Figure 2.12 Edit Assigned Students to Course
2.3.3 Instructor Interface

The instructor plays a key entity in this application. The instructor maintains a question repository, lecture notes, create quizzes, views peer review by students, and checks the report on quizzes.

**Instructor default page**

The instructor’s main page has options to add/view questions, create quizzes, upload lecture notes, view peer review work, and view the report on quizzes.

![Figure 2.13 Instructor Default Page](image)

**Figure 2.13 Instructor Default Page**
Add question page

Instructor adds a question using this page. The page contains a form which enables the instructor to add a new question. The instructor provides a question, the answer, and the explanation. The question will be added and saved in the question repository, as shown in Figure 2.14.

Figure 2.14 Add Question page
Create quiz page

To create a quiz, the instructor selects the subject on which he/she wishes to create the quiz as shown in Figure 2.15. After selecting the course, the instructor is forwarded to create a quiz page.

Figure 2.15 Select Course Page
The instructor creates a quiz using this page. Instructor provides a quiz topic, and the start and end date. The question from the question repository is displayed, and the instructor chooses the question for the quiz, as shown in Figure 2.16.

![Create Quiz Page](image)

**Figure 2.16 Create Quiz Page**
Upload lecture notes

This page provides the form to upload the lecture notes. The instructor can upload the lecture notes by sections or by whole chapters, as shown in Figure 2.17.

![Figure 2.17 Upload Lecture Notes](image-url)
View student peer-reviews

The Instructor views the uploaded work by students, as shown in Figure 2.18. The instructor has to select the course for peer review.

Figure 2.18 Uploaded Work by Students for Peer Review
**View Messages**

The instructor views the messages sent to him by students, as shown in Figure 2.19.

![View Messages](image)

**Figure 2.19 View Messages**
View quiz report

To view a quiz report, the instructor selects the course. The names of the students enrolled in the selected course are displayed, as shown in Figure 2.20. The quiz report is generated after the instructor selects the name of the student, as displayed in Figure 2.21. To view course grades, the instructor selects “Grade For The Course” link, then a report is generated, as shown in Figure 2.22.

Figure 2.20 Select Student Name to View Quiz Report
Figure 2.21 Quiz Report
Figure 2.22 Quiz Report by Grade

Number of Students In Grades

- Grade A: 2
- Grade B: 1
- Grade C: 2
- Grade D: 0
- Grade F: 1

Number of Students in following Grades are listed below

<table>
<thead>
<tr>
<th>Grade</th>
<th>Grade A</th>
<th>Grade B</th>
<th>Grade C</th>
<th>Grade D</th>
<th>Grade F</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
2.3.4 Student Interface

A student is the heart and soul of this application. This application is designed to allow students to explore and discover, to discuss with other students, and more carefully reflect on their own learning. The student has the privilege to download lectures, upload their work, review peers work, send messages, and view report on quizzes. To accomplish these tasks, students can view the following pages.

*Student's main page*

In a student’s main page student are able to view the menu which shows options of taking quiz, sending messages, downloading notes, review peers work, upload their work, and view quiz report, as shown in Figure 2.23.

![Student Default Page](image)

*Figure 2.23 Student Default Page*
Quiz page

The student selects the particular course for which they are supposed to take the quiz, as shown in Figure 2.24.

![Select Quiz Course](image)

Figure 2.24 Select Quiz Course

After selecting the course, the student is forwarded to an actual quiz where he/she attempts the quiz, as shown in Figure 2.25.
Figure 2.25 Student Quiz Page

View lecture notes

This page shows the lecture notes posted by the instructor. Students download the notes posted from this screen. Before reaching this screen, they must select the particular course which they want to view, as shown in Figure 2.26.
In this page a student can upload work. Students have to select a topic before they can upload their work, as shown in Figure 2.26. They can also upload the ink image.

Figure 2.26 Download Lecture Notes

Upload student’s work

In this page a student can upload work. Students have to select a topic before they can upload their work, as shown in Figure 2.26. They can also upload the ink image.
This page allows the student to upload specific work. Students can review each other’s work by downloading each other’s work, as shown in Figure 2.28.
Figure 2.28 Peer Review
Messaging

The students can post a message to the instructors, as shown in Figure 2.29.

Figure 2.29 Student Messaging Page
View Quiz Report

To view a quiz report for a specific class, the student selects the course, as shown in Figure 2.30. The quiz report is generated, as shown in Figure 2.31.

Figure 2.30 Select Course To View Quiz Report
Figure 2.31 Quiz Report
3. PROPOSED SYSTEM DESIGN

3.1 System Requirement

The application is a Web based application. It follows client server architecture. ASP.NET pages, components and database (SQL Server 2005) reside on the server. Whenever a client makes a request, the server fetches the requested data and sends it back to the client.

The following configurations are implemented for the application.

3.1.1 Server Configuration

1) Operating System: Windows server 2003 installed with Internet Information Server 6.x or Windows XP Professional installed with Internet Information Server 5.x.

2) Microsoft.NET Framework: Microsoft.NET Framework 2.x is installed.

3) Microsoft SQL Server: Microsoft SQL Server 2005 is installed.

4) Microsoft Tablet PC SDK 1.7 is installed.

5) Microsoft Handwriting Recognition software is installed.

3.1.2 Client Configuration

This application is specifically designed for the Tablet PC but any PC with internet access can act as a client. The client is a Web browser running on a PC. The recommended Web browser is Internet Explorer 7.0 or higher. The URL of the
application must be added to the list of fully trusted website under the security tab in Internet options. If the client is a machine other than a Tablet PC, Microsoft Tablet PC SDK 1.7 is installed.

3.2 Overview of the System

The proposed system is a Web based application and follows Hyper Text Transfer Protocol (HTTP). The Web pages are dynamically generated by the Active Server Pages.NET (ASP.NET). The ASP.NET is an integral part of Microsoft .NET Framework. SQL Server 2005 is used as a database server. The ASP.NET server side scripts and components are coded in C#.NET. The script written in ASP.NET processes the input given by the user, accesses the database, and returns the result in a Hyper Text Markup Language (HTML) page to the user.

Figure 3.1 shows the abstract view of the workings of the application. The browser sends a request to the server. The server forwards this request to .Net component for processing. The .NET component processes the request, fetches the data, and sends the data back to the page.
Figure 3.1 Abstract System Model
3.3 Main Components of the System

None of the available Web based application systems are designed for Tablet PCs. This application is designed specifically for Tablet PCs, keeping in mind the exclusive features provided by Tablet PCs. A Tablet PC lets a user take, organize, and access notes, have more productive study sessions, and access the information one needs quickly and easily [2006 Microsoft Corporation]. A student can write on the screen of a Tablet PC and save the notes in any format. Later these notes can be uploaded.

3.3.1 Question Bank

The application has a question repository where questions for different courses are stored. Only the instructor who has created the questions has the permission to view or change the questions. While creating questions, the instructor specifies a question and four possible solutions to the question, and he/she also provides the correct answer by putting down the number (1, 2, 3 or 4). The instructor also explains why the answer is correct. The instructor can add as many questions as he/she wants to use and can create as many quizzes as required for the course.

3.3.2 Creating and Uploading of Lecture Notes

The instructor creates the notes and uploads them. The notes can be on one section of a chapter, or a few sections of a chapter, or the complete chapter. Students download them and go through the notes and rewrite what they read in their own format to understand better. They also add the text which they read from different textbooks or
from different websites, compiling into notes which are very unique to individual students. This work, if shared between students, can be very resourceful, as every student has something extra to share, which helps everyone in achieving more knowledge. The students can upload their work to share with others.

3.3.3 Peer Review

Peer review is a key component of this application because students can share their resources with others. It creates a knowledge bank, which in turn benefits everybody who is part of the class. Peer review can be given as an essential part of the grade to complete the course.

Tablet PCs make it very easy to do the peer review. On the tablet PC, a student can just write in his/her own handwriting, which is stored in image format, and is uploaded by the Web browser. The handwriting recognition software is native to Tablet PCs where students can write and upload the notes as part of their work.

3.3.4 Messaging

Messaging plays a key role for students. They can ask questions directly to the instructor at any time, even during the class session. As long as the instructor is logged on, he/she can view the messages.

3.3.5 View Report

The instructor views the performance of all students in a quiz. He can use the report as a performance evaluation tool for the class.
3.4 The Design of the Application

A good application requires an efficient database. The database is a key essence of this application. The efficiency of the database is achieved by properly structured tables. An entity relational diagram is used to create the database, as shown in Figure 3.2.

3.4.1 Portal Architecture

This application is developed with portal standards. According to the user’s requirements, the system accomplishes several tasks. Various 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 application. As each tab represents a logical group of tasks, each tab is further divided into modules. Each module represents an individual task, and each module is represented as a sub menu in the layout of the application. Many modules are hidden, and they are accessed using several buttons.

This application assigns a role to every user. There are three user defined roles. These roles are Admin, Instructor, and Students, and each role is given authorization to access certain tabs and modules of the application. When the user logs into the system using the username and password, the application checks the authenticity of the user information provided. If the provided information is valid, the system checks the privileges based on the user role and allows the user into the system.

The system follows “Three Tier Architecture”: Web page is the first tier, the middle tier is called DAL (Data Access Layer), and the third tier is the database. All the
communication to database is achieved through DAL, thus adding on effective security feature that protects database information from being revealed at any point of time. As the system is protected by the user roles, it provides added security.

3.4.2 Database Design

The database is designed to achieve efficient minimal storage and quick access. Each table is given a primary key for efficient retrieval of data. Relationships between tables are defined in order to maintain referential integrity. The database consists of 24 tables, 93 stored procedures. The E-R diagram of this application is shown in Figure 3.2.

Figure 3.2 Entity Relationship Diagram
The entities are derived from information and requirement analysis. The detailed description is given below:

- **Administrator**: The administrator creates Courses, Semesters, UserName (Studentid, Instructorid), Password, FirstName, LastName, Email. UserName is the key entity.

- **Instructor**: This entity is related to the other main entities Course and Question. This entity holds attributes UserName, Password, Name and Email. This entity is responsible for Question Quizzes and Lecture Notes.

- **Course**: This entity is related to the other entities Instructor, Student, Questions and Quiz. CourseID is the key entity. This entity is responsible for Instructor, Course, Quiz and Question.

- **Student**: This entity is related to Instructor and Course. The attributes of this entity are UserName, Password, Name and Email. UserName is the key entity.

- **Quiz**: This entity is related to Question Bank and Instructor. Quizzes are created by the Instructor entity. The main attributes are QuizID, Description, StartingDate, EndingDate. QuizID is the key entity.
- **Question Bank**: The *Question Bank* has questions and answers. Only the *Instructor* entity can change questions and its parameters. Using this entity, the *Instructor* entity can create the *Quiz* entity. The main attributes are *QuestionID*, *Question*, *Description*, *CourseID*, and *InstructorID*. *QuestionID* is the key attribute.

- **Lecture Notes**: This entity is related with *Instructor* entity. This entity is created by Instructor entity. It has attributes like *NotesID*, *CourseID*, and *UserName*, in which *NotesID* is the key entity.
3.4.3 Security

The basic security is implemented by restricting access to application. The application is built on a portal framework, which gives access to only those modules which are authorized by a role. The application prevents two kinds of attacks namely SQL injection and URL manipulation. These kinds of attacks are very common. To prevent SQL injection, application uses stored procedures to fetch the data from database. URL manipulation attacks are prevented by checking if the URL request is genuine or postback. Before making any requests for the data, the username is fetched from Session-ID. If no username is found or session is expired, the user is transferred to the login page.
4. TESTING AND EVALUATION

The application is a Web based. It is accessible to anyone who knows the URL of the application. The administrator, instructor, and student profiles are stored in the database. The student grades are also stored in the database. The testing is a major component in the success of the application. This application is tested for its usability and security.

4.1 Usability Testing

The success of the application lies in the user experience of 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 is performed by getting feedback from users (i.e., instructors and students) of this system. The feedback relates to user interface and usability. Based on this feedback, changes were made to the system.

4.2 Security Testing

Security was a major feature to evaluate for this system. The long term and success of this system depend on highly effective security. One of the evaluations is to monitor the database transactions. Maximum emphasis is given on not leaving the database connection open for a long time. Various test cases were designed to check that only eligible users are allowed to access the application. Sessions are very important in a Web based application to prevent unauthorized access. ASP.NET encrypts all session variables and makes it very difficult to view any data. The .NET framework 2.x gives additional security features to the application.
5. EXPECTED RESULTS AND CONCLUSION

The application provides an exciting new platform for interaction between instructor and students. The application supports active learning in the classroom. Rather than following a text, peer review allows students to explore and discover, to discuss with other students, and to more carefully reflect on their own learning. Quizzes, on other hand, help students to evaluate their performance.

The application achieves its objective by focusing on student learning by providing tools with which students interact with other peers and the instructor.
ACKNOWLEDGEMENTS

I am very grateful for the co-operation and interest of Dr. Dulal Chandra Kar, Associate Professor of Computer Science, Texas A&M University – Corpus Christi. Without his guidance and expertise this project would never have come into existence.

My sincere thanks to Dr. Long-Zuang Li, Assistant Professor of Computer Science, Texas A&M University – Corpus Christi, for encouragement and timely help.

My sincere thanks to Dr. Jennifer Smith-Engle, Associate Dean of Science and Technology, for having faith in me and timely help.
BIBLIOGRAPHY AND REFERENCES


[Esposito] Esposito, Dino. ASP.NET 2.0 Core Reference. First Ed. Microsoft Press 2005


APPENDIX 1
INKCONTROL.DLL FILE

The InkControl.dll file is used to grab the ink from the webpage. The code for the DLL file is listed below:

```csharp
// File: InkArea.cs
// Simple control that accepts either ink or text input, depending
// on the capabilities of the user's machine.
//
// The features used are: InkCollector, handwriting recognition, and
// ink serialization.

using System;
using System.Diagnostics;
using System.IO;
using System.Windows.Forms;

namespace InkControl
{
    /// The InkArea Control class
    public class InkArea : UserControl
    {
        // Declaring the control used for collecting input
        private Control inputArea = null;
        // InkCollector used for collecting
        private object inkCollector = null;

        #region Standard Template Code
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;
        #endregion

        public InkArea()
        {
            // calling windows form designer.
            InitializeComponent();

            // attempting to create an InkCollector and attach it to
            // a panel control. If successful, the client can
            // support ink within a web page else use a text area
            try
            {
                inputArea = CreateInkCollectionSurface();
            }
            catch (FileNotFoundException)
            {
                inputArea = new TextBox();
                ((TextBox)inputArea).Multiline = true;
            }

            inputArea.Size = this.Size;

            // Adding the control for collecting input
            this.Controls.Add(inputArea);
        }
    }
}
```
/// Clean up any resources being used.
protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        if (components != null)
        {
            components.Dispose();
        }
    }
    base.Dispose(disposing);
}

/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
private void InitializeComponent()
{
    this.SuspendLayout();
    this.Name = "InkArea";
    this.ResumeLayout(false);
}

/// Event handler from this control's Resize event
/// Updates the child control to have the same size.
protected override void OnResize(EventArgs e)
{
    if (null != inputArea)
    {
        inputArea.Size = this.Size;
    }
}

/// finds out whether inking is supported on the client machine
/// Returns true if inking is supported and false otherwise
public bool InkEnabled
{
    get
    {
        return inkCollector != null;
    }
}

/// If inking is supported, returns the serialized ink data.
/// Otherwise, returns an empty string.
public string InkData
{
    get
    {
        if (InkEnabled)
        {
            return SerializeInkData();
        }
        else
        {
            return String.Empty;
        }
    }
}
/// If inking is supported, returns the recognized ink.  
/// Otherwise, returns the contents of the text box.
public string TextData
{
    get
    {
        if (InkEnabled)
        {
            return RecognizeInkData();
        }
        else
        {
            return ((TextBox)inputArea).Text;
        }
    }
}

/// Disposing the internal child control used for collecting ink.
public void DisposeResources()
{
    if (InkEnabled)
    {
        ((InkCollector)inkCollector).Dispose();
    }
}

#region Helper Methods (private)
/// trys to create an InkCollector and attach it to a 
/// Panel control.  If unable to create the InkCollector, 
/// an exception is thrown.
protected Control CreateInkCollectionSurface()
{
    try
    {
        Panel inkPanel = new Panel();
        inkPanel.BorderStyle = BorderStyle.Fixed3D;
        inkCollector = new InkCollector(inkPanel);
        ((InkCollector)inkCollector).Enabled = true;
        return inkPanel;
    }
    catch
    {
        throw;
    }
}

/// Serializeing the ink data as GIF using a Base 64 string.
protected String SerializeInkData()
{
    Debug.Assert(InkEnabled, null, "Client must be ink-enabled");
    // ObtainING the ink
    Ink ink = ((InkCollector)inkCollector).Ink;
    // Serializing the ink
    if (ink.Strokes.Count > 0)
    {
        byte[] inkDataBytes = ink.Save(PersistenceFormat.Gif);
        return Convert.ToBase64String (inkDataBytes);
    }
    // returning the empty string.
    return String.Empty;
}

/// recognizing ink by using the recognizer.
protected String RecognizeInkData()
{
Debug.Assert(InkEnabled, null, "Client must be ink-enabled");

// get the ink associated
Ink ink = ((InkCollector)inkCollector).Ink;

if (ink.Strokes.Count > 0) {
    try {
        // try to create a recognition context
        RecognizerContext recognizerContext = new RecognizerContext();
        RecognitionStatus recognitionStatus;
        recognizerContext.Strokes = ink.Strokes;
        RecognitionResult recognitionResult = recognizerContext.Recognize(out recognitionStatus);
        if ((recognitionStatus == RecognitionStatus.NoError) && (null != recognitionResult)) {
            return recognitionResult.TopString;
        }
    }
    catch (Exception) {
        // an exception takes place and returns empty string
    }
    return String.Empty;
}

#endregion
APPENDIX 2
WEB CONFIGURATION FILE

Configuration files are typically created offline or during the development of the application. They are deployed with the application and can be altered at any time by administrators. Changes to configuration file are detected by the ASP.NET runtime and causes all the pages to recompile. A configuration file is used to fine tune the application settings and enforces strict security policies.

```xml
<?xml version="1.0"?><configuration>
<configSections>
      </sectionGroup>
    </sectionGroup>
  </sectionGroup>
</configSections>

<system.web>
  <httpHandlers>
  </httpHandlers>

  <profile enabled="true">
    <properties>
      <add name="FirstName" type="string"/>
      <add name="LastName" type="string"/>
      <add name="uname" type="string"/>
    </properties>

    <membership>
      <providers>
        <remove name="AspNetSqlMembershipProvider"/>
        <add name="AspNetSqlMembershipProvider" type="System.Web.Security.SqlMembershipProvider, System.Web, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f1d50a3a" connectionStringName="LocalSqlServer" enablePasswordRetrieval="false" enablePasswordReset="true" requiresQuestionAndAnswer="false" applicationName="" requiresUniqueEmail="false" passwordFormat="Hashed" maxInvalidPasswordAttempts="5" minRequiredPasswordLength="1" minRequiredNonalphabeticCharacters="0" passwordAttemptWindow="10" passwordStrengthRegularExpression=""/>
      </providers>
    </membership>
  </profile>
</system.web>
```
<roleManager enabled="true" cacheRolesInCookie="true" defaultProvider="SqlRoleProvider" />
<roleManager enabled="true" cacheRolesInCookie="true"/>
<authentication mode="Forms"/>
<pages theme="Skin1">
<controls/>
</pages>

Set compilation debug="true" to insert debugging symbols into the compiled page. Because this affects performance, set this value to true only during development.

<compilation debug="false">
<assemblies>
<add assembly="System.Web.Extensions, Version=1.0.61025.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"/>
<add assembly="Microsoft.ReportViewer.WebForms, Version=8.0.0.0, Culture=neutral, PublicKeyToken=B03F5F71D50A3A"/>
<add assembly="Microsoft.ReportViewer.Common, Version=8.0.0.0, Culture=neutral, PublicKeyToken=B03F5F71D50A3A"/>
</assemblies>

<buildProviders/>
</compilation>

<httpHandlers>
<remove verb="*" path="*.asmx"/>
</httpHandlers>

<httpModules>
</httpModules>

<connectionStrings>
<remove name="LocalSqlServer"/>
<add name="LocalSqlServer" connectionString="Data Source=localhost;InitialCatalog=tablet_pc;Integrated Security=True" providerName="System.Data.SqlClient"/>
<add name="tablet_pc" connectionString="SERVER=TAMUCC; DATABASE=tablet_pc; UID=sa; PWD=****" providerName="System.Data.SqlClient"/>
</connectionStrings>
</system.web>
<add key="CrystalImageCleaner-AutoStart" value="true" />
<add key="CrystalImageCleaner-Sleep" value="60000" />
<add key="CrystalImageCleaner-Age" value="120000" />
</appSettings>
</location>
</configuration>