Flexible Writing Tutorial Web Site for HMCS

By

David Wooley

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements
for
the Degree of Bachelor of Science
in Information Engineering Technology

University of Cincinnati
College of Applied Science

Flexible Writing Tutorial Web Site for HMCS

By

David Wooley

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements
for
the Degree of Bachelor of Science
in Information Engineering Technology

*Signature Goes Here*____________________________ ____________
David Wooley Date

*Signature Goes Here*____________________________ ____________
Advisor/Department Head Date

Acknowledgements

I would like to give special thanks to Dr. Hazem Said, Department Head and Academic Advisor. I would like to give special thanks to Professor Humpert, Senior Design. They have assisted with my difficult work schedule and given me every opportunity to succeed.

Table of Contents

Figures Contents

Abstract

Currently the Humanities, Media, Cultural Studies Department (HMCS) has a website that contains only department contacts and emails for students inquiring about help for writing assignments. This process of emailing all assignments, consultants manually filling out a CAS Writing Consultant Report Form's is labor intensive, with a large number of students utilizing the services at once; the department can get overwhelmed quickly. Feedback to students and instructors can become delayed with no trace ability between the document send/receive and comments pertaining to its review. The *Flexible Writing Tutorial Web Site* has been developed utilizing C#, ASP.NET, and SQL Server database, allowing for a central repository for students, professors, and consultants to collaborate. Students can submit assignments for review; consultants are notified for the review and submit the assignment back to students for revision work to be done, if any is needed. Trace ability for assignments and their corresponding reviews are now directly connected in the database for reports or queries to assist in evaluating work flow, overall usage of the site, or track improvement by a student using the services.

Statement of Problem

1.1. Introduction

Writing tutoring at the College of Applied Science (CAS) is a staple in giving students the ability to help themselves. Teresa Cook (2) part of Humanities, Media, Cultural Studies Department (HMCS) has expressed concern of not having a vehicle offering writing tutoring services a way of making tools available for the students that provides easy searches on topics and answers to common questions without direct intervention by a staff member. Access to a website providing required writing assignment information before submission, would will help reduce time needed for staff and allow for more time to determine the students individual needs.

This issue has been addressed by Purdue University (1) providing resources for students via a website known as the OWL. The only problem is that this web site is hard to navigate and does not readily provide information. I found it to be a little confusing trying to lookup information, just the opposite result needed for CAS. Sending emails for assistance seemed very simple going through this website but that leads to an excess of requests, which are difficult to track and sort.

There are online utilities that provide document management services such as UGS (6), Citric (5), MatrixOne (7), and SAP (8). These companies specialize in what are referred to Product Lifecycle Management systems (PLM), which handle a large variety of tasks. Some of these systems contain revision control and very complex flows that do not bode well for an everyday user who is trying to submit a paper for review or project. Nor, do these systems provide an easy management environment for maintainability of professors to use.

The solution must be a combination of available software and utilities coupled with ease of use for both students and professors who may be tracking these documents.

1.2. Definition of the Need

Humanities, Media, Cultural Studies Department (HMCS) has a website that contains only department contacts and emails for students inquiring about help for writing assignments. All initial writing assignment requests get placed through JoAnn Thompson, who then forwards the request to the necessary consultant for review. The consultant reviews the document and utilizing track changes functionality, they edit the writing assignment and email it back to the student. The final task of the consultant is too fill out a CAS Writing Consultant Report Form keeping a copy for records and forward a copy to the class instructor.

This process is labor intensive, with a large number of students utilizing the services at once; the department can get overwhelmed quickly. Feedback to students and instructors can become delayed with no trace ability between the document send/receive and comments pertaining to its review. These disconnections from reviews, documents, and persons involved make it difficult to analyze a student's progress or statistics on various attributes logged by the consultants.

2. Solution Description

Web based portal application with database back end will provide the necessary capabilities to resolve HMCS issues. Utilizing Microsoft Visual Studio 2005 .NET in combination with SQL Server database to generate a web portal with ASP.NET. Such a portal gives easy access to application through the Web Browser without installing software. The database interface allows for data management and retrieval capabilities for better trace ability during the process flow. Email notifications will be sent to individuals facilitating continuous process monitoring.

2.1. User profiles

2.1.1. Students

Prevention of mistakes submitting documents to consultants reducing the amount of rework or time delay gathering needed information consultants require before evaluating documents. Simplify communication with consulting staff allowing for ease of use.

2.1.2. Faculty

2.1.2.1. Writing Center Faculty

Notifications to be sent when documents are submitted and guaranty of student information included in the email for better trace ability. Digitized means of filing the CAS Writing Consultant Report Form which done manually for every evaluated paper.

2.1.2.2. Class Instructor

Simple searches for papers submitted for the class assignments they teach. Trace ability of documents evaluated by consultants and attributes about the review.

2.1.2.3. Administrator

Maintenance tasks for removing old data from the database also Add/delete for users, classes available, and quarters. These tasks are held by administrators only to prevent data corruption and add stability.

3. Objectives of Project

Provide an application to University of Cincinnati CAS that provides access to writing tutoring materials, writing assignment collaboration/management tools through the application, known as a portal, data repository for tracking papers or other documents for courses taken.

3.1. Design Protocol

The Flexible Writing Tutorial Web Site has various components needed for operation. There are four user roles with different functionality within the application. The different levels of access for these groups are demonstrated in Figure 1.

Use Case Diagram for CAS Writing Lab Automation

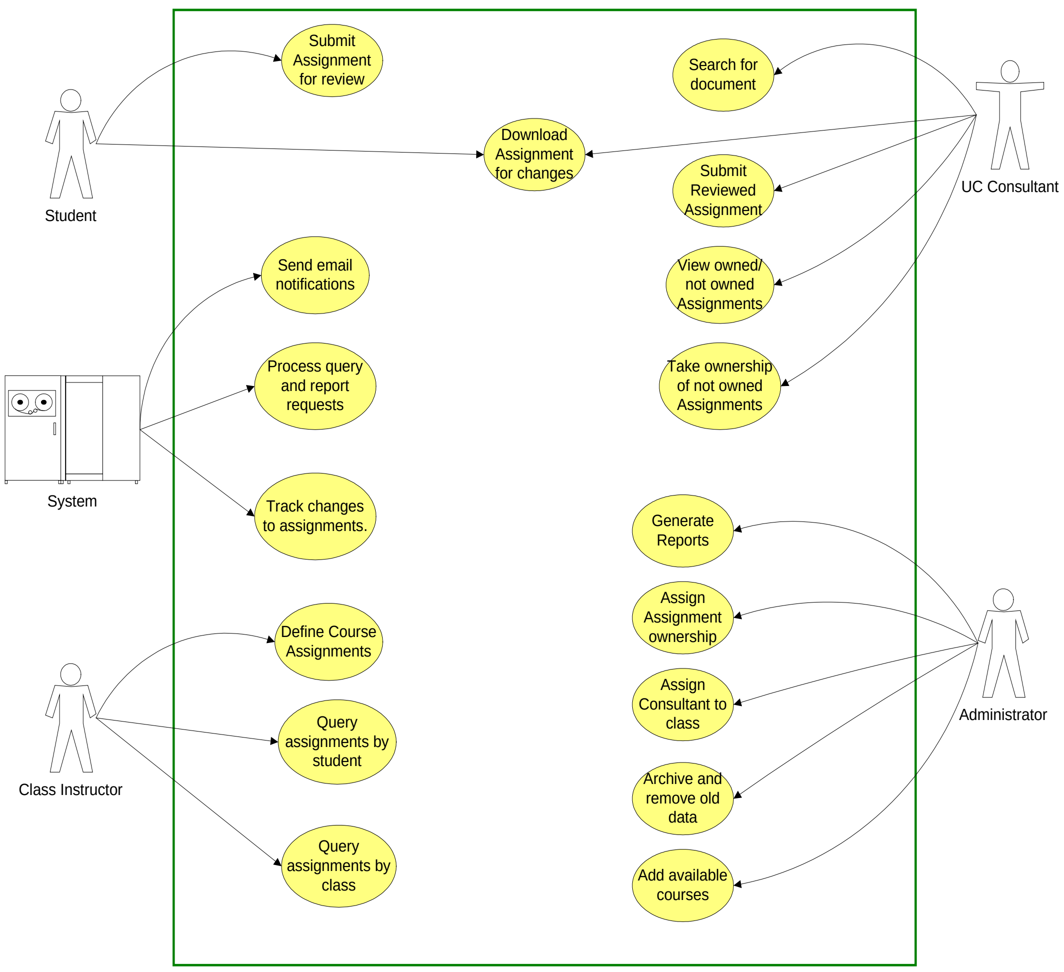

Submit Assignment for review
Search for document
Student
Download Assignment for changes
UC Consultant
Submit Reviewed Assignment
Send email notifications
View owned/ not owned Assignments
Process query and report requests
Take ownership of not owned Assignments
System
Track changes to assignments.
Generate Reports
Assign Assignment ownership
Define Course Assignments
Administrator
Assign Consultant to class
Query assignments by student
Class Instructor
Archive and remove old data
Query assignments by class
Add available courses

Figure 1: Use Case Diagram

3.2. Areas of IT Applied

This project utilizes two areas of IT to achieve application completion.

- Programming: Microsoft Visual Studio .NET, ADO.NET, and C# code sets to build a web portal for the application.
- Database: Microsoft SQL Server 2003 was configured for one relational database utilized by the application.

3.3. Database

The Flexible Writing Tutorial Web Site utilizes one relational database developed on Microsoft SQL Server for data storage needs. Records are stored, tracked, or updated using different SQL queries executed in the C# portion of the application. Figure 2 illustrates the relational database design and table structure used for this implementation.

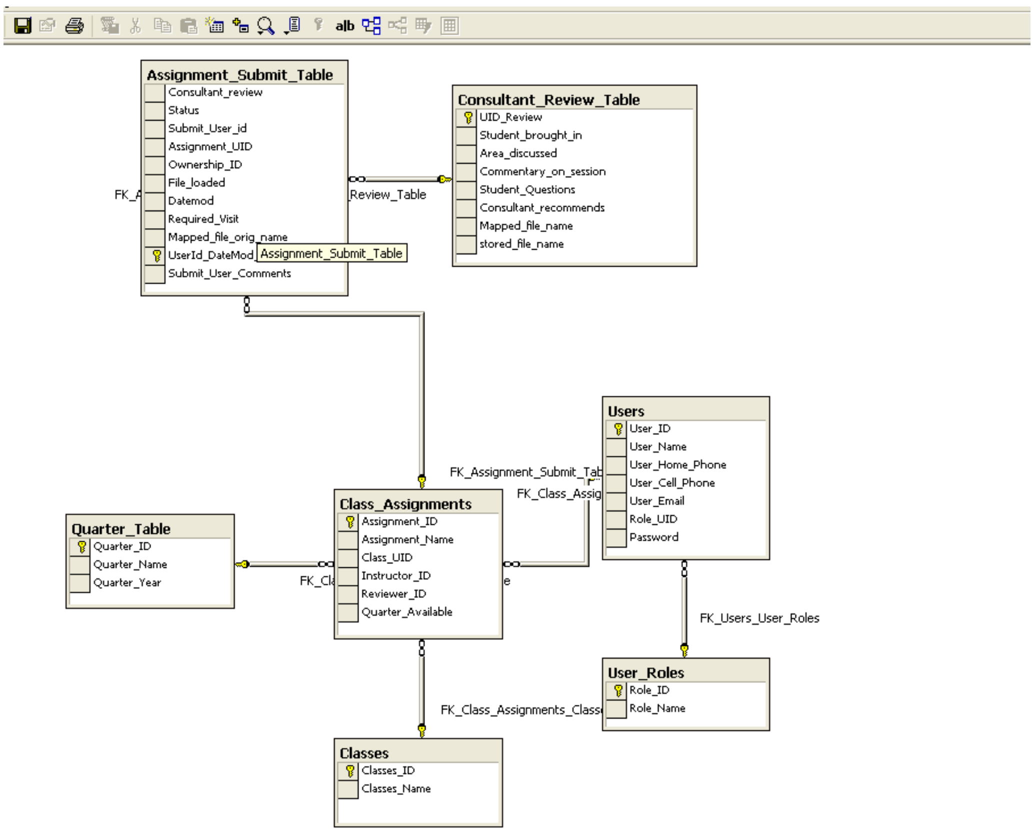


Figure 2: Relational Database Architecture

3.4. Deliverables

- Develop an application easily accessible to Faculty and Students
- Develop and create a database that is capable to provide tracking of documents
- Develop Web portal interface to the database
- Email notifications that are light weight to not fill up a mailbox

3.5. Technical Details

This application will be created utilizing many different software tools and hardware. List provided below with brief descriptions:

3.5.1. Hardware

- Server in the IT computer lab

  The Senior Server will be used to test this project. It has most of the programs installed and hardware needed to complete it.

- Blackboard host server

  Need capability to test or access modules as well as determine storage limits for project.

3.5.2. Software

This project will be using software that is available in the IT computer lab.

- **Windows Server 2003**

  Needed for Microsoft SQL Server.

- **Internet Information Services (IIS)**

  It is required for creating ASP.NET Web pages.

- **Microsoft Visual Studio .NET 2003**

  Utilized ASP.NET, C#, and ADO.NET functionality for creating the Web interface

- **Microsoft SQL Server**

  SQL Server will be used to create the database, table, relations, and queries that this project needs.

3.6. Proposed Budget

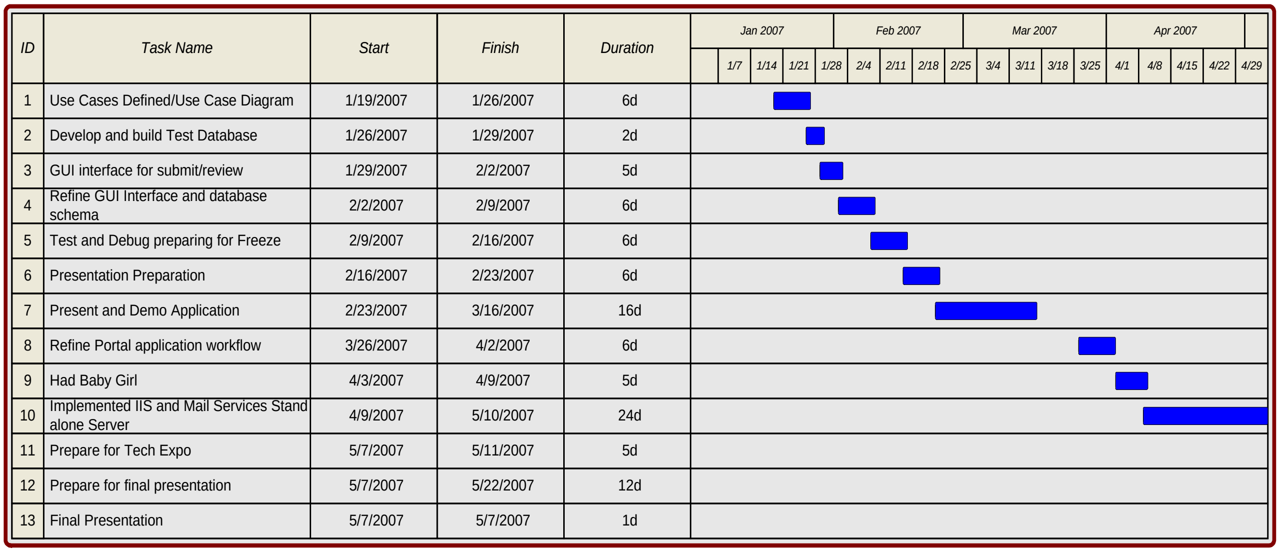
Budget has changed over the development period of the application and was reduced due to not requiring the application. Out of pocket expense was $0.00, all materials needed were either provided by the college or already owned.

Item	Description	Retail Cost	Cost Incurred
Server	Provided by the IT computer lab	$538.00	$0.00
Windows Server 2003	Provided by the IT computer lab	$769.00	$0.00
Web Server	Provided by the IT computer lab	$538.00	$0.00
Internet Information Services (IIS)	Included in Windows	$0.00	$0.00
Microsoft Visual Studio .NET 2005	Provided by the IT computer lab	$686.94	$0.00
Microsoft SQL Server	Provided by IT computer lab	$600.00	$0.00
	Retail Total:	**$3131.94**	
	My Total:		**$0.00**

Table 1: Project Budget

3.7. Proposed Project Timeline

Project timeline has been updated to reflect some pauses in development due to my daughter being born 04/3/2007. This caused some issues making some of the deliverables but the overall project was completed and presented.



ID	Task Name	Start	Finish	Duration
1	Use Cases Defined/Use Case Diagram	1/19/2007	1/26/2007	6d
2	Develop and build Test Database	1/26/2007	1/29/2007	2d
3	GUI interface for submit/review	1/29/2007	2/2/2007	5d
4	Refine GUI Interface and database schema	2/2/2007	2/9/2007	6d
5	Test and Debug preparing for Freeze	2/9/2007	2/16/2007	6d
6	Presentation Preparation	2/16/2007	2/23/2007	6d
7	Present and Demo Application	2/23/2007	3/16/2007	16d
8	Refine Portal application workflow	3/26/2007	4/2/2007	6d
9	Had Baby Girl	4/3/2007	4/9/2007	5d
10	Implemented IIS and Mail Services Stand alone Server	4/9/2007	5/10/2007	24d
11	Prepare for Tech Expo	5/7/2007	5/11/2007	5d
12	Prepare for final presentation	5/7/2007	5/22/2007	12d
13	Final Presentation	5/7/2007	5/7/2007	1d

Table 2: Project Timeline

3.8. Risk Analysis

The largest risk to my project is the time constraints due to working full-time and school. I have been attempting to minimize the impact by taking some vacation to focus on school and provide the best demonstration of the project I can. Power – outage caused a lot of problems for productivity loosing 3 days with no backup generator for my house where the majority of development is being done.

As you can see below in (Table 3), I have identified many of my key risks for this project. Some of these will become less risk as I progress but many will be issues until Senior Design is completed.

Risk	**Cause**	**Abatement**	**Owner**	**Level of Impact**	**Date Identified**
Limited Time for completion	Work Full time and school	Took some vacation to allow for more time to dedicate for project	David Wooley	High	01/01/2007
Loosing Data	Computer Crash	Have multiple maintained copies of application and database	David Wooley	Low	01/20/2007
Power outage	Ice storm took down power grid	Dedicate additional time to key areas of functionality before design freeze	David Wooley	High	02/13/2007
Application not working for demo	Issues with machine in room.	Have application set up on multiple machines to minimize impact	David Wooley	Medium	01/20/2007
Login Functionality	Not sure what the functionality is for	May need to customize unique login for this application.	David Wooley	Low	02/01/2007

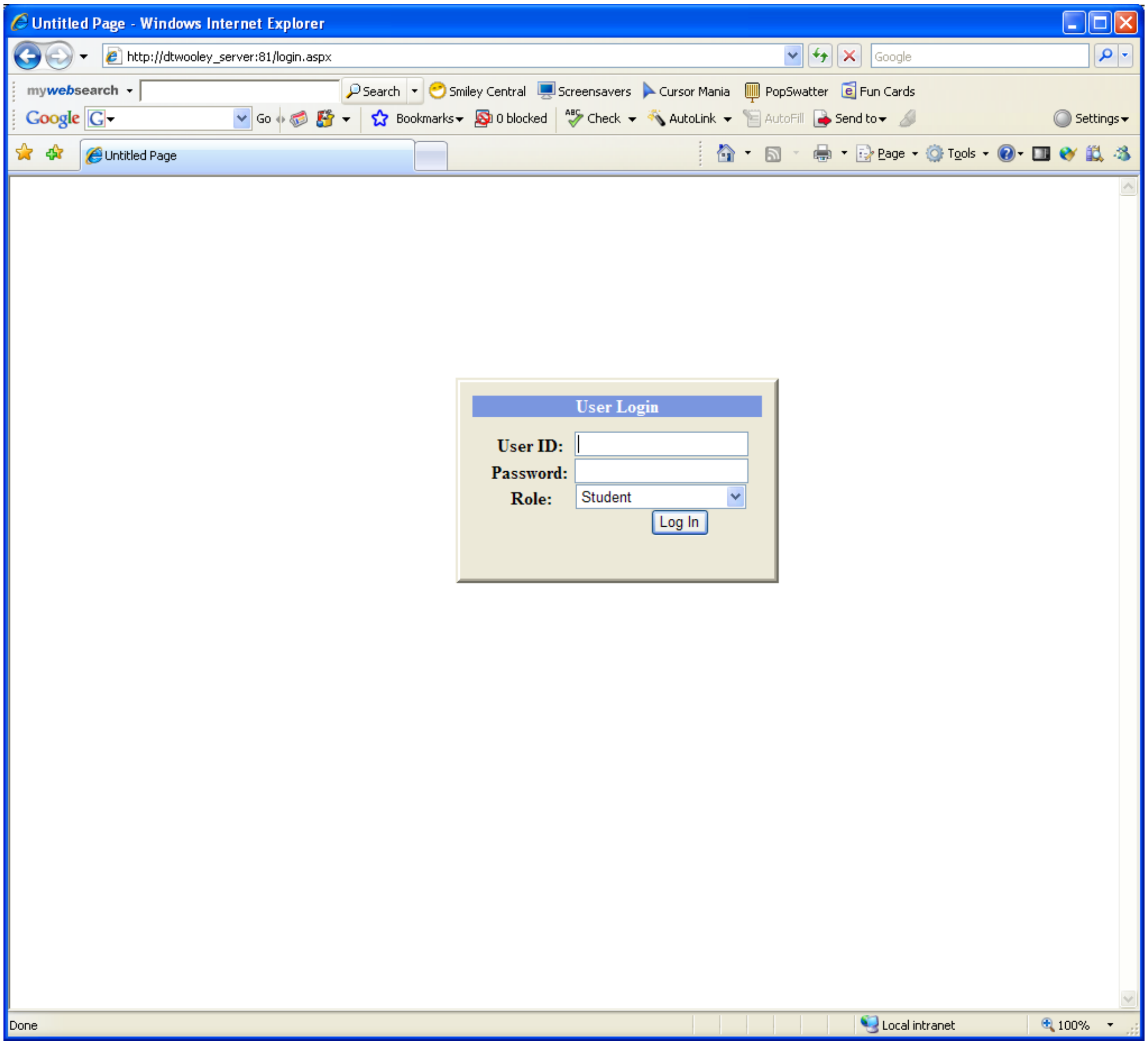
Table 3: Risk and Abatements

4. Proof of Design

There are many varying use cases for the application allowing for flexibility. Every user must login as a specific role but the roles give the user different functionality in the portal. Sections below describe the different functionality in more detail.

4.1. User Login

As you can see in Figure 3 the login interface gives the user capability to login as different roles. By creating a login with this functionality allowed control of the Web pages dynamically based on the role.



User Login

User ID:

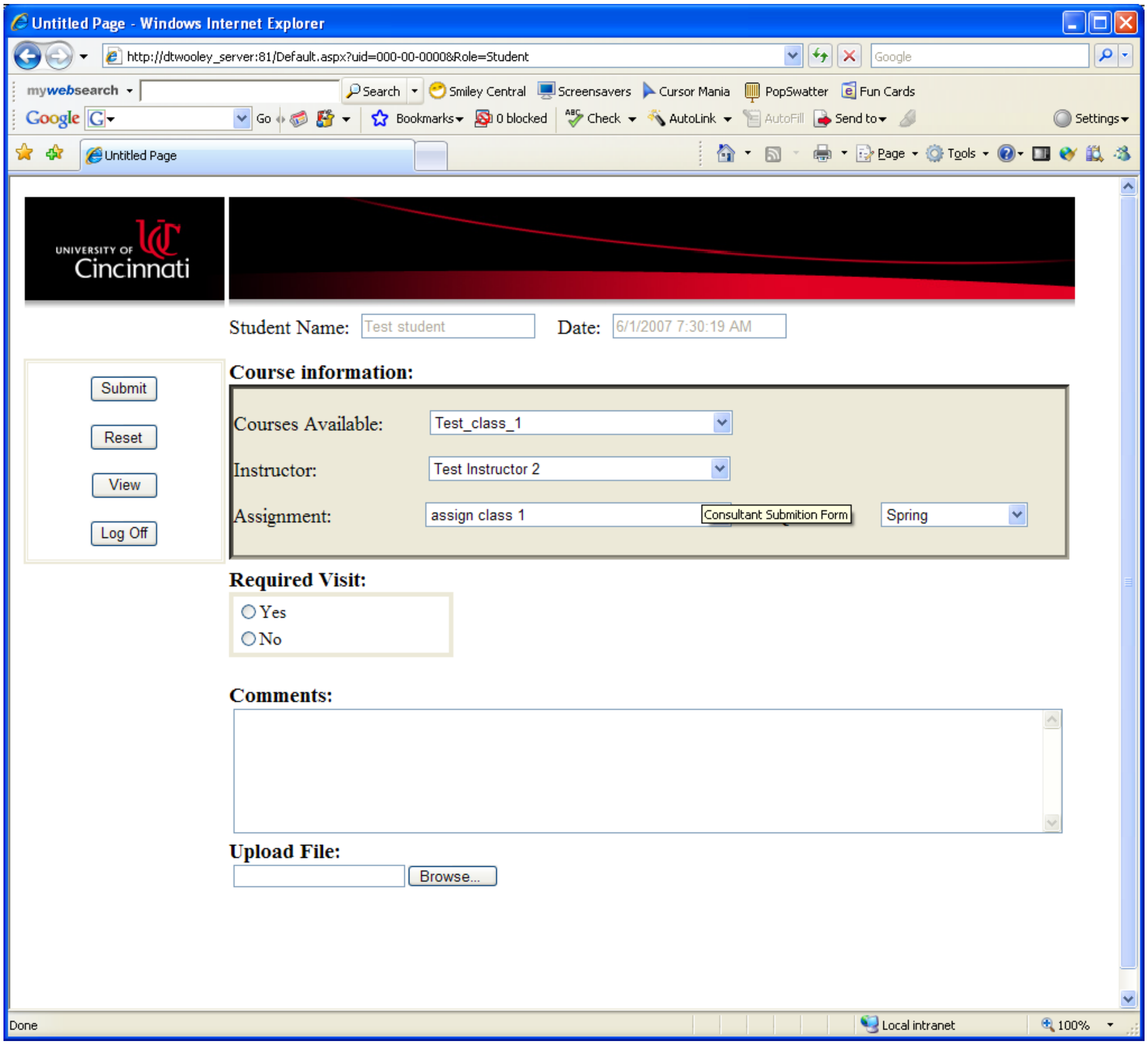
Password:

Role: Student

Figure 3: Login Interface

4.2. Student Submission Form

One of the issues resolved with this form is the missing required data the consultant needs to do an evaluation on the document. In figure 4, the course, assignment, quarter, and student submitting will be captured as part of the email notification sent once submitted.



Student Name: Test student Date: 6/1/2007 7:30:19 AM

Course information:

Courses Available:	Test_class_1
Instructor:	Test Instructor 2
Assignment:	assign class 1 Consultant Submition Form Spring

Required Visit:

○ Yes
○ No

Comments:

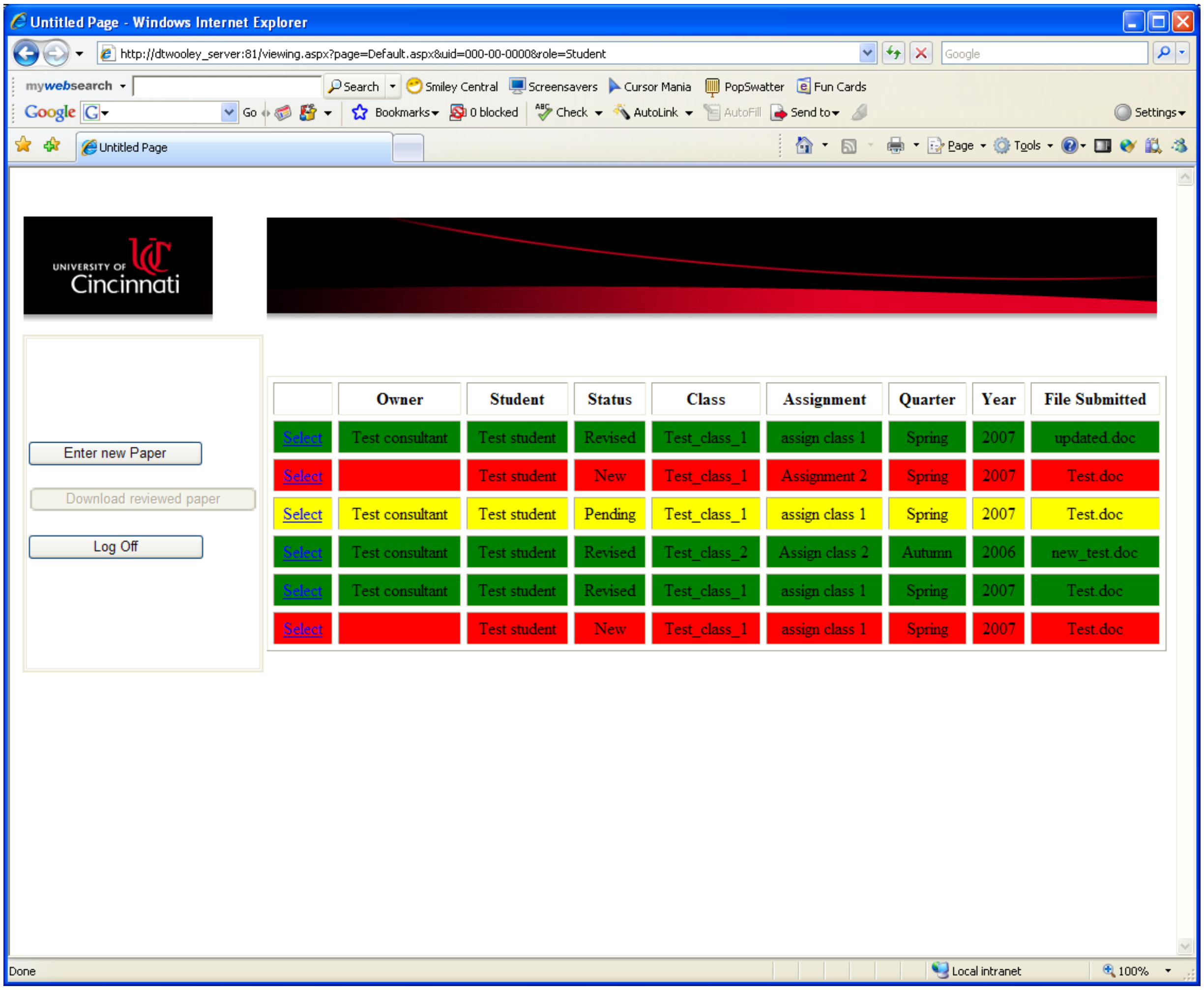
Upload File:

Browse...

Figure 4: Student Submission Form

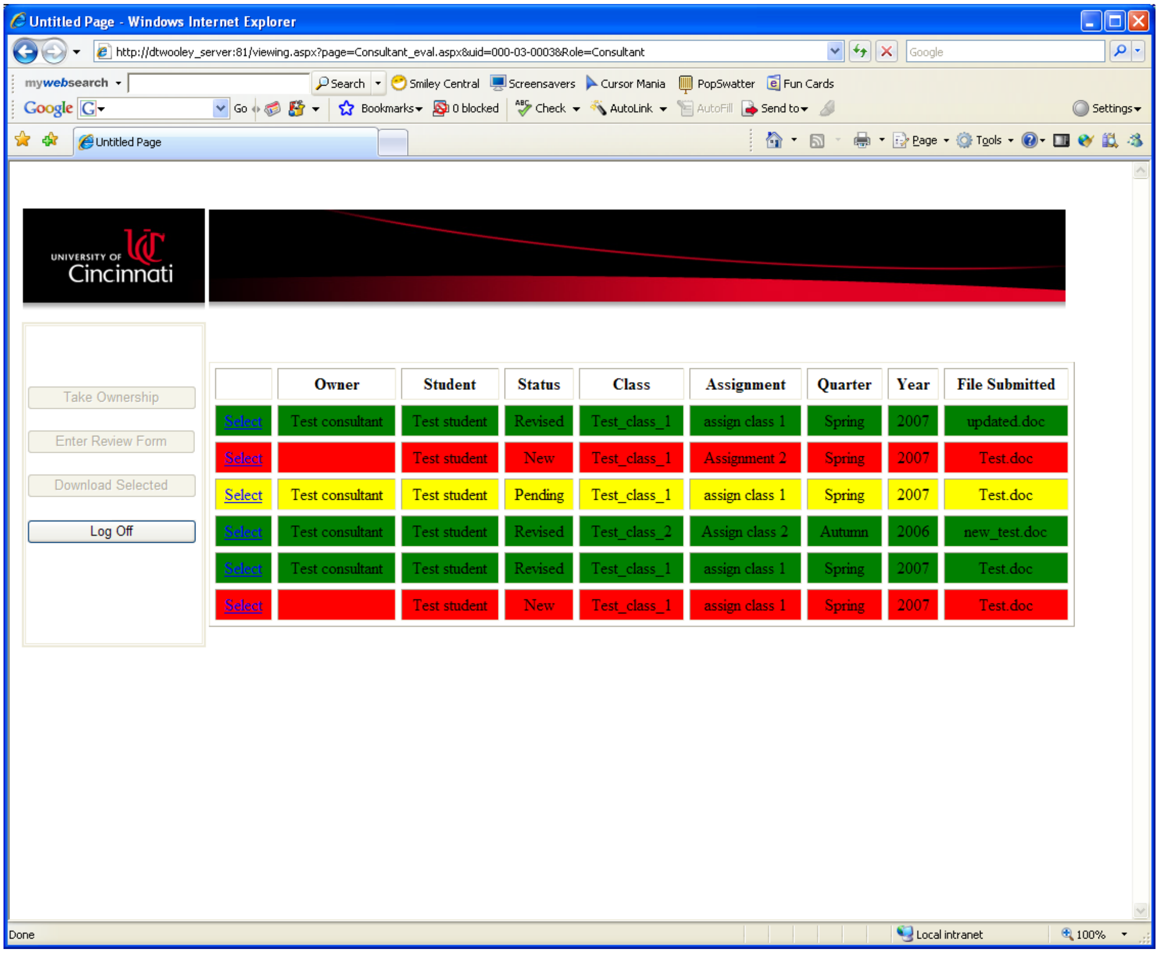
4.3. Viewing Submitted Records

Allowing the user to be able to view the status of their records was a critical step in the application development. Figure 5 shows the page as it would be if a student were to login to the application. Students only require the functionality to see the status of their records and receive revised documents to make necessary changes. In Figure 6, the same page looks different for a consultant allowing for all submitted files owned by the consultant and records without an owner will be shown allowing the option for the consultant to take ownership of the record. This is where the process flow changes hands as seen in Figure 1.



	Owner	Student	Status	Class	Assignment	Quarter	Year	File Submitted
Select	Test consultant	Test student	Revised	Test_class_1	assign class 1	Spring	2007	updated.doc
Select		Test student	New	Test_class_1	Assignment 2	Spring	2007	Test.doc
Select	Test consultant	Test student	Pending	Test_class_1	assign class 1	Spring	2007	Test.doc
Select	Test consultant	Test student	Revised	Test_class_2	Assign class 2	Autumn	2006	new_test.doc
Select	Test consultant	Test student	Revised	Test_class_1	assign class 1	Spring	2007	Test.doc
Select		Test student	New	Test_class_1	assign class 1	Spring	2007	Test.doc

Figure 5: Record Viewer Student

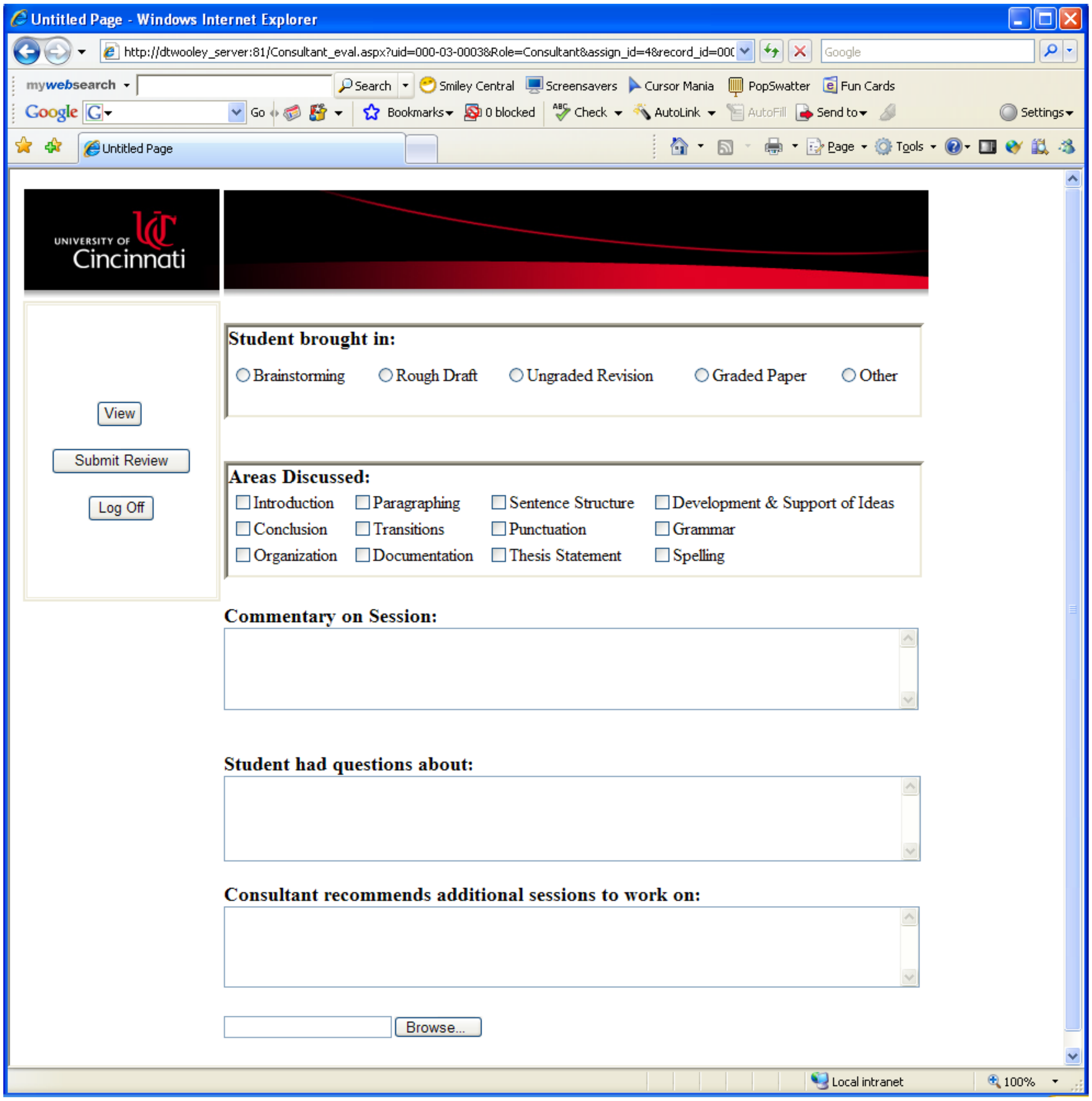


	Owner	Student	Status	Class	Assignment	Quarter	Year	File Submitted
Select	Test consultant	Test student	Revised	Test_class_1	assign class 1	Spring	2007	updated.doc
Select		Test student	New	Test_class_1	Assignment 2	Spring	2007	Test.doc
Select	Test consultant	Test student	Pending	Test_class_1	assign class 1	Spring	2007	Test.doc
Select	Test consultant	Test student	Revised	Test_class_2	Assign class 2	Autumn	2006	new_test.doc
Select	Test consultant	Test student	Revised	Test_class_1	assign class 1	Spring	2007	Test.doc
Select		Test student	New	Test_class_1	assign class 1	Spring	2007	Test.doc

Figure 6: Record Viewer Consultant

4.4. Consultant Form

The consultant evaluation form in Figure 7 was created based off of the "CAS Writing Consultant Report Form" allowing the consultant to evaluate the document directly as part of the process flow. This also creates a direct relationship between the record submitted and the reviewed document with comments. Trace ability is no longer a concern with this relationship established.



View

Submit Review

Log Off

Student brought in:

○ Brainstorming ○ Rough Draft ○ Ungraded Revision ○ Graded Paper ○ Other

Areas Discussed:

□ Introduction	□ Paragraphing	□ Sentence Structure	□ Development & Support of Ideas
□ Conclusion	□ Transitions	□ Punctuation	□ Grammar
□ Organization	□ Documentation	□ Thesis Statement	□ Spelling

Commentary on Session:

Student had questions about:

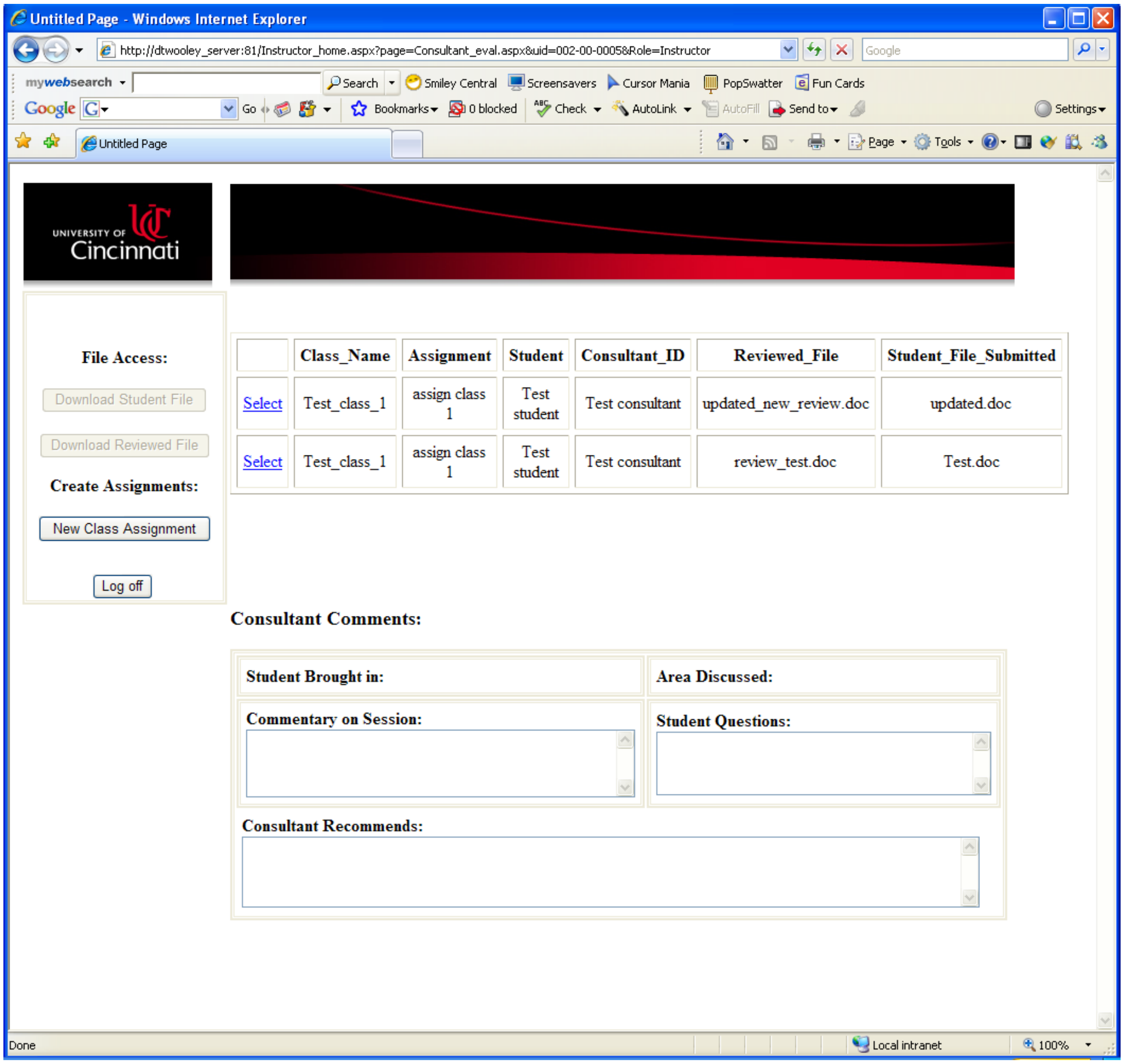
Consultant recommends additional sessions to work on:

Browse...

Figure 7: Consultant Evaluation Form

4.5. Instructor Viewing

The instructor who creates the assignments has the ability to query all records regarding the class. Direct relationships now are contained between the submitted record and the evaluation, which allows the instructor to see all comments about the document. Access to the documents themselves is available.



	Class_Name	Assignment	Student	Consultant_ID	Reviewed_File	Student_File_Submitted
Select	Test_class_1	assign class 1	Test student	Test consultant	updated_new_review.doc	updated.doc
Select	Test_class_1	assign class 1	Test student	Test consultant	review_test.doc	Test.doc

Consultant Comments:

Student Brought in:

Area Discussed:

Commentary on Session:

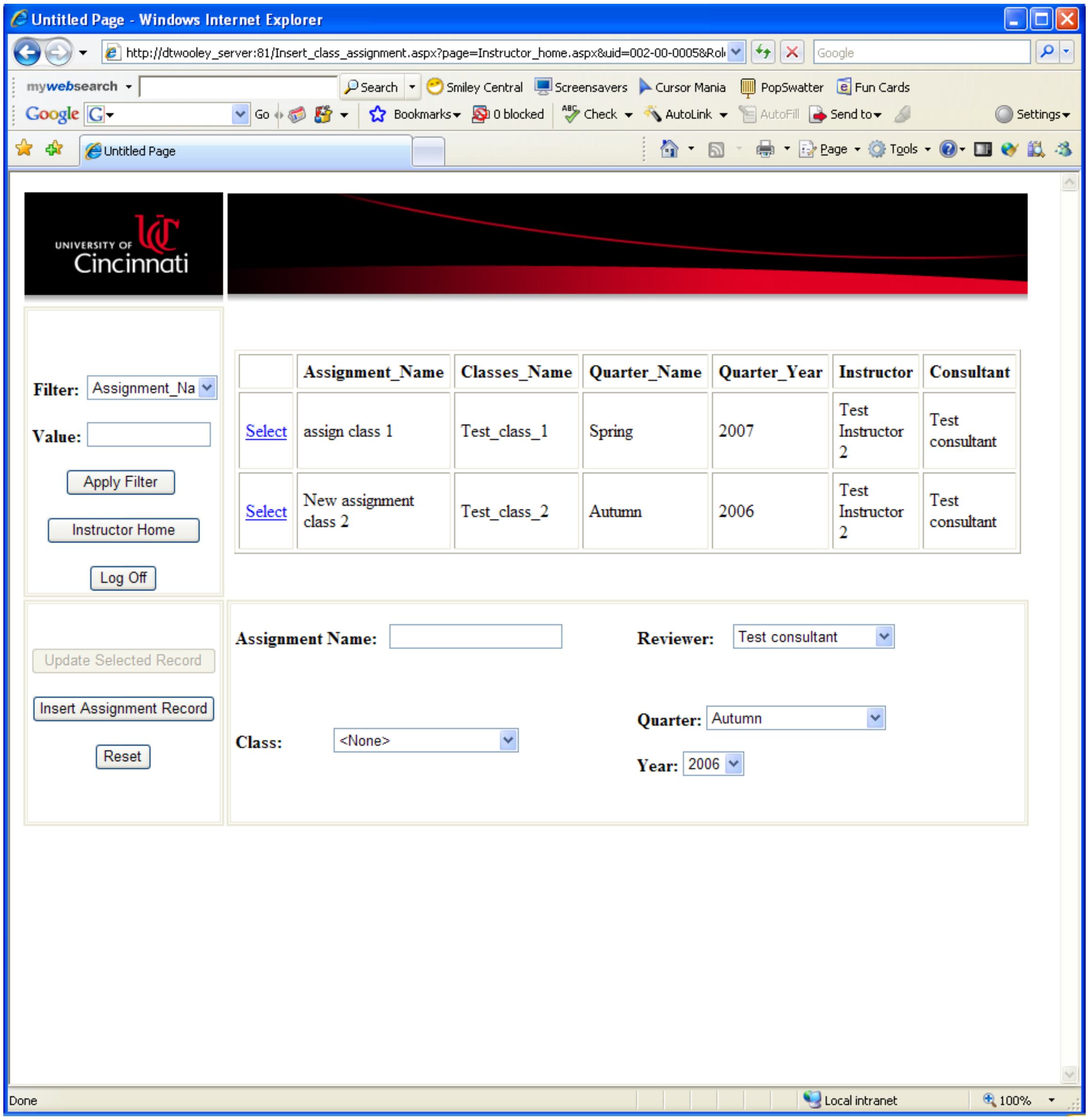
Student Questions:

Consultant Recommends:

Figure 8: Instructor Review

4.6. Creating Assignments

Due to instructors owning the assignments given for any course taught, this form in Figure 9 was created for managing assignments available for each instructor of the course. Assignments shown are only the instructor that has logged in, preventing confusion and possible changes to the incorrect assignment.



Filter: Assignment_Na

Value:

	Assignment_Name	Classes_Name	Quarter_Name	Quarter_Year	Instructor	Consultant
Select	assign class 1	Test_class_1	Spring	2007	Test Instructor 2	Test consultant
Select	New assignment class 2	Test_class_2	Autumn	2006	Test Instructor 2	Test consultant

Assignment Name:

Reviewer: Test consultant

Quarter: Autumn

Class: <None>

Year: 2006

Figure 9: Instructor Adding/Editing Assignments

4.7. Modular Design

Establishing classes for reusable code was one of the first development challenges met. I defined classes that would be used throughout the portal as you can see in Figure 10. The message box class was the most useful due to ASP.NET does not have a message box class so one was built using JavaScript to execute. Database calls were consolidated into one class providing ease of use and configurable to point at different servers seen in Figure 11.

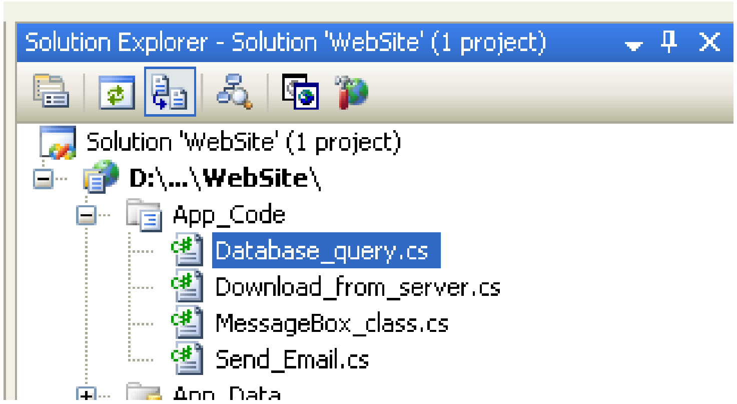
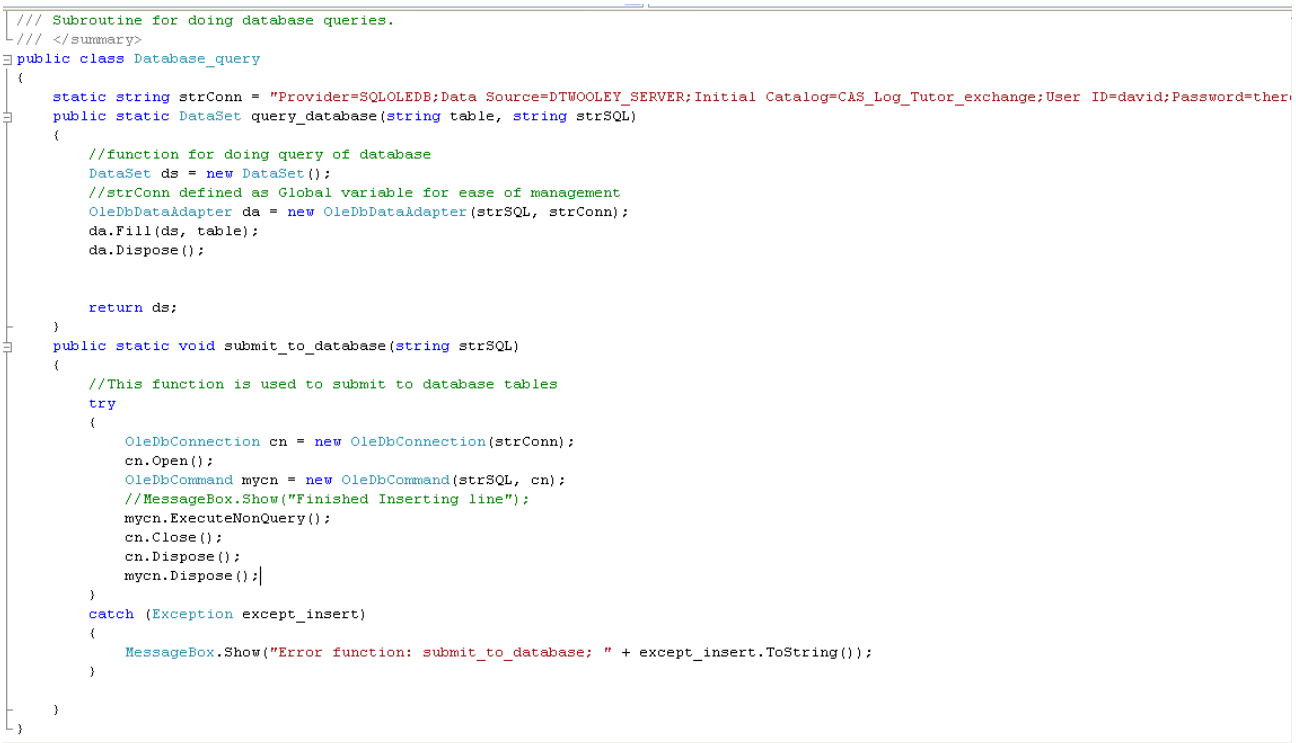


Figure 10: Modular Code



```
/// Subroutine for doing database queries.
/// </summary>
public class Database_query
{
    static string strConn = "Provider=SQLOLEDB;Data Source=DTWOOLEY_SERVER;Initial Catalog=CAS_Log_Tutor_exchange;User ID=david;Password=ther
    public static DataSet query_database(string table, string strSQL)
    {
        //function for doing query of database
        DataSet ds = new DataSet();
        //strConn defined as Global variable for ease of management
        OleDbDataAdapter da = new OleDbDataAdapter(strSQL, strConn);
        da.Fill(ds, table);
        da.Dispose();


        return ds;
    }
    public static void submit_to_database(string strSQL)
    {
        //This function is used to submit to database tables
        try
        {
            OleDbConnection cn = new OleDbConnection(strConn);
            cn.Open();
            OleDbCommand mycn = new OleDbCommand(strSQL, cn);
            //MessageBox.Show("Finished Inserting line");
            mycn.ExecuteNonQuery();
            cn.Close();
            cn.Dispose();
            mycn.Dispose();
        }
        catch (Exception except_insert)
        {
            MessageBox.Show("Error function: submit_to_database; " + except_insert.ToString());
        }

    }
}
```

Figure 11: Database Query Class

5. Conclusion

This project has great potential for both students and professors especially the different capabilities that have been developed. Faculty can view any number of records submitted to the writing lab, view consultant comments, and even retrieve documents submitted. Not only is this program proposal beneficial to assisting faculty track the writing assignments of U.C. students it also provides viable, visible information to the staff in regards to student development and focus areas.

Though this project was not deployed, I do see the need for such an application for tracking such processes. An individual that just contracted out an application that did the exact same thing for $120,000 approached me at Tech Expo. The continued interest in Product Lifecycle Management (PLM) systems is increasing daily. The knowledge I gained developing this solution was irreplaceable. My timeline was definitely impacted by the birth of my daughter 04/03/2007 and production issues at my full-time employer taking time away from time needed for this project but I feel that I have provided a proof of concept and a working application.

6. Bibliography

1. Owl, Purdue University English Tutoring Website. 06 December 2006.

   http://owl.english.purdue.edu/

2. Teresa Cook, Problem and requirements phone interview. 28 November 2006.

3. Microsoft SharePoint Software, 28 December 2006.

   http://www.microsoft.com/office/sharepoint/prodinfo/overview.mspx

4. Federation Software, 24 October 2006. http://www.federationweb.com/

5. Centric, 13 January 2006. http://www.centricsoftware.com/

6. UGS, 30 October 2006. http://www.ugs.com/

7. MatrixOne, 20 December 2006. http://www.matrixone.com/

8. SAP, 02 January 2007.
   http://www.sap.com/community/pub/events/2005_11_23_edm/index.epx

9. Enovia, 05 January 2006, http://www.3ds.com/brand/enovia/

10. Blackboard Content System, 09 January 2007.
    http://www.blackboard.com/products/academic_suite/content_system/index.Bb