Online Instruction Collaboration Project

Michael David Cassens

The University of Montana

Let us know how access to this document benefits you.
Follow this and additional works at: https://scholarworks.umt.edu/etd

Recommended Citation

https://scholarworks.umt.edu/etd/8945
Permission is granted by the author to reproduce this material in its entirety, provided that this material is used for scholarly purposes and is properly cited in published works and reports.

**Please check "Yes" or "No" and provide signature**

Yes, I grant permission [ ]

No, I do not grant permission [ ]

Author's Signature: [Signature]

Date: [May 2, 2003]

Any copying for commercial purposes or financial gain may be undertaken only with the author's explicit consent.
Online Instruction Collaboration Project

by

Michael David Cassens

B.A. The University of Montana 1996

presented in partial fulfillment of the requirements

for the degree of

Master of Science

The University of Montana

2003

Approved by:

Chairperson

Dean, Graduate School

5-5-03

Date
The Computer Science Department at The University of Montana has utilized individual course websites effectively for many years. However, a great deal of time has been spent creating and maintaining these sites. The Online Instruction Collaboration (OIC) addressed this problem by creating a space where instructors and students alike can gather in a single online area and communicate with one another. The goal of the project was not to advocate the OIC over other products, but to ensure the viability of using custom software solutions in an arena where commercial products tend to dominate. This paper discusses the successes and challenges of using a customized tool. It also examines the usage of OIC in the classroom during the spring of 2003 with a cross-sample of computer science students and two instructors. The results of this study suggested that the OIC can provide a viable solution for the Computer Science Department.
# TABLE OF CONTENTS

Chapter 1 Overview and Background .................................................... 1  
Chapter 2 Requirements and Specifications ............................... 8  
Chapter 3 Design ...................................................................................... 26  
Chapter 4 Implementation ................................................................. 47  
Chapter 5 Testing ...................................................................................... 50  
Chapter 6 Usability Questionnaire .......................................................... 54  
Chapter 7 Conclusion ............................................................................... 59  
Chapter 8 Future Direction ................................................................. 61  
References .................................................................................................... 65  
Appendix A ................................................................................................. 67  
Appendix B ................................................................................................. 108  
Appendix C ................................................................................................. 124  
Appendix D ................................................................................................126
ACKNOWLEDGMENTS

I would like to thank my wonderful wife and daughter for all their support and love through this process. With them, all things are possible. Additionally, I would like to thank Dr. Joel Henry, Dr. Yolanda Reimer, and Dr. Carol Brewer for serving on my committee and guiding me through this process. Lastly, I would like to thank Gabriel Williams and William Best for their instrumental assistance in testing this product.
Overview and Background

The Online Instruction Collaboration (OIC) Project is an intuitive interactive enhanced web-based system for students and instructors. This system was created out of a need for a better online presence in the Computer Science Department. Currently, the Computer Science Department utilizes static web pages for instructors to post course information to students, and although this system has been a standard in the past, it requires maintenance for which the instructors do not have time. At present, some instructors find student teaching assistants to update and maintain their course web pages because of the time commitment required. Additionally, the interface that enrolled students have to navigate through to find their courses is cumbersome at best. Also, there is an inconsistency between the user interfaces of the individually driven pages. Lastly, there is an absence of individualized grade reporting. Many instructors keep a local copy of all student assessment information in a spreadsheet format. Although this is effective for the instructor, the student will never know their current course grade unless they confer with the instructor. If grades are posted online for a course, the instructor must assign randomly generated numbers to each student to protect their privacy. This is an accepted method of display at
The University of Montana, but it still requires the student to review their grades in a non-private manner.

Recently, another solution was introduced into the Computer Science Department, which previously had been used primarily for online courseware at The University of Montana. It is the corporate product Blackboard [1]. Since this was introduced as a tool for teaching online courses, it had not received much attention in the area of supplementing traditional courses until now. It is now being utilized by some traditional courses in the Computer Science Department, as well as other departments on campus. The acceptance and usage of this product solidifies the assertion that the Computer Science Department is searching for more functionality with their online resources than ever before.

Another online courseware product used is eCollege.com [2]. This is a web based system not currently accepted or supported as a whole by The University of Montana because of the adoption of Blackboard. However, it is heavily used in the School of Education with their online courses. This tool is similar to Blackboard with its functionality and its appearance. This system extends its functionality over Blackboard by providing the ability to administer streaming video with its basic product offering. Another advantage of eCollege.com over Blackboard is that its servers are located in a variety of mirrored sites throughout the United States instead of at one location in Missoula, Montana. This provides
better reliability and ensures access for universities that may feel somewhat hesitant about using this type of courseware.

With the adoption of Blackboard and eCollege.com, clearly The University of Montana would like to utilize these products to not only teach courses online, but also enhance traditional courses as well. With so many commercial products available, a valid question to consider is whether customized software would be viable in this product space. The OIC was developed to address this viability and determine whether custom software would be useful in the Computer Science Department.

The goal of OIC was to provide a simple yet effective interface that incorporates all the important functionality that the mainstream products have to offer, but with the simplicity of the static web page. Similar to the commercial products, OIC's focus was to provide specific departments a single point of entry that can be accessed by all students enrolled in courses in the department. However, it did not incorporate all aspects of the commercial products. By focusing on the core functions that instructors and students required, it was hypothesized that the OIC could be an option to supplement traditional courses.

To create this product, a rigorous product development cycle was followed to build a stable and reliable system. The development cycle incorporated the staged
delivery model, also known as an incremental implementation. This model was followed throughout the development of the product because the full list of requirements was fairly well defined, but not all requirements needed to be implemented to have a functional prototype. Since this model implements functionality in successive stages, it allows the developer to implement additional functionality in later stages [3]. This is a successful model because each stage is fully functioning and, therefore, useful to the client even without all the requirements implemented.

Knowing that this initial release would be utilized by approximately a hundred students, as well as two instructors, enhancements and alterations to the product were fully expected. Therefore, it was determined that requirement changes would be the primary risk factor in this project. Leveraging a staged or incremental development model decreases the change in requirements risk by defining and implementing core requirements and then adding additional requirements into latter stages as the product matures.

The users consisted of students and instructors from the two pilot courses, CS232 - Architecture and Assembly Language and CS332 - Introductory to Algorithms. These courses used the product throughout Spring Semester 2003. There was an assumed basic knowledge of computing and ability to navigate through web pages. Since these users have been in the Department for two or
more years, it was assumed that they had been exposed to the current course web pages.

Another main goal of the OIC was to decrease the amount of time spent using the product. Instructors and students alike do not have much time beyond their regular school and research duties to be weighed down with a cumbersome system. Therefore, the OIC system needed to provide a definite time savings if it was to be used consistently. Page response times had to be as close to immediate as possible. Although, these calculations were not tabulated formally, it is important to discuss their role in evaluating this product.

As an adjunct to this main goal, the OIC had to recover from errors in a timely fashion and provide meaningful feedback to the user. No matter the ability or expertise of a user, there will always be times when an inadvertent action will be performed. Thus the system needed to allow the user to continue on by choosing another option.

The first line of recovery from error is displaying feedback in crucial areas and creating constraints with certain functionality. In the OIC, this was implemented whenever information was updated or inserted, as well as when form information was submitted. These simple validation methods helped prevent users from performing actions that were unwanted.
Since the Student Area was mostly "read-only", feedback was displayed in more subtle ways. For example, since all lectures and homework were displayed as hyperlinks, the student made a selection, and the appropriate information was displayed. At the top of the displayed information, the selected lecture or homework title was shown verifying what they are viewing. Conversely on the Contact Instructor and the Change Password page, the functionality required more user feedback to ensure the changes had been applied. Therefore, specific feedback was displayed to the user indicating the task had been successfully completed.

The Instructor Area required much greater validation and error recovery. Each function in the Instructor Area entailed insertion, modification, or deletion of information. Therefore, more validation was required to ensure the correct function had occurred. OIC provided this feedback by displaying a message at the top of the page when information had successfully or unsuccessfully been inserted, deleted, or modified.

In other products, the validation method was signified by adding another confirmation page when information was changed or updated. Although this provided good user feedback, it created the issue of added time to effectively interact with the product. The OIC addressed this issue by creating good user
feedback for the instructor on the same page, which eliminated the confirmation page, and therefore decreased the number of pages that would be required to use the system thus saving time ultimately.

There was an overlap between OIC and the commercial products, but the list also included additional benefits that were not addressed by the commercial products. The following is a summary of the major benefits of using the OIC system:

- Standardized system of display for instructors and students
- Time saving measures for instructors when keeping track of course related information
- Built in privacy for students when viewing a summary of grades
- Student rosters automatically populated by students
- Automatic emails to students made available without using an email client
- Automatic email to instructors without having to know the instructor’s email address
- Only “core” functionality exposed reducing time spent using the system
Chapter 2

Requirements and Specifications

To be successful, all software products must have requirements and specifications [4]. Typically, the client requesting the product defines the requirements of the product, and then the specifications are created to fulfill those requirements. In this section, the requirements and specifications are outlined delineating all the core functionality that the system should possess in order to be viable. The OIC has been separated into three distinct areas of functionality. The first area is the Administrator Area (AA). The second area is the Instructor Area (IA), and the third area is the Student Area (SA). Each of these areas has a set of distinct users, and these users are listed below.

Specific Breakdown of User Populations

- General Administrators
- Instructors
- Students
Throughout the rest of this section, a visual representation of how a user navigates through the OIC system as a student, instructor, and then as an administrator is shown.

Example steps to gain access to the SA

1. Log In from the Introductory Page (Figure 1)
   a. If they have not logged in before, create a new user (Figure 2)
   b. If they have forgotten their User Name and Password, access the Forgot Password screen to retrieve the information (Figure 3)

2. Select a Course that they were enrolled in by choosing a course from the drop down box and clicking the Insert button (Figure 4)

3. Under Student View, click on the Select hyperlink next to the course name to view the course information

4. View Announcements (Figure 5)

5. View Syllabus, Lectures, Homework, and Grades (Figures 6-9)

6. Contact the Instructor (Figure 10)

7. Change Password (Figure 11)
**LOCAL COMPANY – JOB LISTING**

**Intern Position For LogiSYS:**

I am looking for a self-motivated, attention-to-detail computer science student for a paid intern position at LogiSYS.

I would like to hire a student who has at least two years of Computer Science completed.

The job entails using source control, building and installing executable programs as well as report generation and various other tasks.

I need someone for approximately 20 hours a week and may hire two interns if the right candidates apply.

Send your resume to:

rburgad@logistic-systems.com

or call Ruth at 728-0921 ext.2116.
Welcome to the Computer Science Courses

Instructor Administration

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Description</th>
<th>Select</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS232</td>
<td>Assembly and Architecture</td>
<td>Select</td>
<td>Delete</td>
<td></td>
</tr>
<tr>
<td>CS332</td>
<td>Algorithms</td>
<td>Select</td>
<td>Delete</td>
<td></td>
</tr>
</tbody>
</table>

Add A Course:

CS232  INSERT

Student View

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Description</th>
<th>Select</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS232</td>
<td>Assembly and Architecture</td>
<td>Select</td>
<td>Delete</td>
<td></td>
</tr>
<tr>
<td>CS332</td>
<td>Algorithms</td>
<td>Select</td>
<td>Delete</td>
<td></td>
</tr>
</tbody>
</table>

Announcements

2/8/2003 I have posted your project groups. Please let me know if I have made any mistakes. Also, I have posted the algorithm requests that I have received so far. Also, I posted Homework 2.

Figure 4

Welcome to CS332 Algorithms

Announcements

2/8/2003 I have posted your project groups. Please let me know if I have made any mistakes. Also, I have posted the algorithm requests that I have received so far. Also, I posted Homework 2.

Figure 5
Welcome to CS332 Algorithms

Syllabus for CS 332

CS 332 Section 1
Instructor: Michael Cassens
Office: SS 404
Office Hours: Tues and Thurs 1:30 – 3:30
University Phone: 243-5605
Home: 721-3805
Office: 549-0616
e-mail: michael@silverleaf-consulting.com
URL: http://www.silverleaf-consulting.com/CSPortal

GOAL: To help you learn different techniques in algorithmic design and how to critically analyze the algorithms that you write

Attendance: Class meets on Tuesday and Thursday from 3:40 – 5:00 pm

Homework and Labs 70%
2-3 Exams 15% each test
You must take two of the three exams (no make ups will be given)

All Assignments will be turned at the beginning of class of the day they are due. No Exceptions.

Grading Scale
100-90 A  79-70 C  59- and beyond F
89-80 B  69-60 D

Requirements:
- Pre-requisites are CS 331 Data Structures and Math 225 – if you have not fulfilled these pre-requisites, then you will be taking this class at your own risk.

Suggestions:
- Ask questions whenever you have them and make sure you understand what I am teaching before we move on.
What are your perceptions of this class?

What do I have working against me?
A. Late in the day
B. Boring hard subject material (always a dangerous combination)
C. The class lasts 1.5 hours long!
D. Where are we going to use this?

So why do we take this class?
1. The department wants you to practice what you are already good at.
   Problem solving You wouldn’t be here if you weren’t good at that.

Where are we in the big scheme of things? Explain to me the different areas of software design and tell me where algorithms fit into this.

Two ideas lie gleaming on the jeweler’s velvet. The first is the calculus, the second, the algorithm. The calculus and the rich body of mathematical analysis to which it gives rise made modern science possible, but it has been the algorithm that has made possible the modern world.

David Berlinski, *The Advent of the Algorithm*

So what is an algorithm? It’s a way to solve a problem.

What do you need to know about algorithms to be successful in computer science?
1. A pool of algorithms from which you can draw
2. A knowledge of how to create new algorithms and then analyze them

These are important distinctions. It’s easy to look at other people’s work and use, critique, etc. with them. It’s a whole another thing to create your own and then critique them.
Welcome to CS332 Algorithms

Homework 1
Due: 2/8/2003

Part 8: Problem 4 Parts A,B
Problem 5 Parts A,B

Page 1*: Problem 2, 6, 9

Welcome to CS332 Algorithms

Grades

Extra Credit 1 5: 0
Homework 1 20/20
Homework 2 20/20
Homework 3 0/20
Project 1 0: 100

Figure 8

Figure 9
Welcome to CS332 Algorithms

Change Password:

Old Password:
New Password:
Confirm Password:

Send Message

Figure 10

Figure 11

16
Example steps to gain access to the IA

1. Log In from the Introductory Page (Figure 1)
   a. If they have not logged in before, create a new user (Figure 2)
      i. Must wait for the Administrator to designate them as an instructor
   b. If they have forgotten their User Name and Password, access the Forgot Password screen to retrieve the information (Figure 3)

2. Select a course being taught by choosing a course from the drop down box and clicking the Insert button (Figure 4)

3. Under Instructor Administration, click on the Select hyperlink next to the course name to change course information

4. Add content to one of the seven areas (Figure 12)
   a. Enter either HTML or rich formatted text
   b. Refer to HTML help link for assistance

5. Upload files for student download
   a. Follow directions to link to uploaded files

6. Remove student function
   a. Select student hyperlink
   b. Click ok on the confirmation dialog box

7. Email students
   a. Select a single student or all students from the drop down list
CS332-01 Algorithms

Welcome!

Please choose a link on the left to alter or add any course information.

Figure 12

Improvements were made to decrease time spent using the OIC. The major time saving benefit occurred when an instructor inserted or modified any of course information. When course information was submitted, the confirmation appeared on the same screen. Additionally, the OIC decreased time spent on the assessment process by allowing the instructor to change all student grades for an
assignment before submitting the changes with the verification of the update displayed on the same screen.

Example steps for gaining access to the AA

1. Log In from the Introductory Page (Figure 1)
2. View Options available only to the administrator (Figure 13)
3. Add/Modify Courses available
4. Delete all course related data
5. Post a General Announcement
6. Change Administrator Password

![Administration Area](image)

Welcome to the administration area.
Choose from the menu on the left to facilitate your administrative needs.

Figure 13
Functional Requirements

At the onset of the OIC project functional requirements were listed, classified by role, and then each requirement was given a priority. Functionality requirements with a high priority were classified as core functional units, and the product would not be released until these requirements were implemented and functional. The second-tier of requirements were classified with medium priority. They were requirements that were understood to be meaningful, but not vital to a functional system. These requirements were to be implemented if time permitted. Lastly, the requirements that fell into the low priority list were not given this classification because of insignificance, but rather because their functionality was truly an enhancement that could wait or did not fit within the scope of the project. Potentially another graduate student could take this project and focus upon the future direction to make it intelligently more feature rich.

Students:

High Priority

1. Student must be able to Create a User name and Password
2. Student must be able to Retrieve a forgotten password
3. Student must be able to login to the system with a user name and password

4. Student must be able select one of the courses they are enrolled in

5. Student must be able to view the syllabus of the selected course

6. Student must be able to view the assignments given in the course

7. Student must be able to view the lectures given in the course

8. Student must be able to view their grades

9. Student must be able to contact the instructor

10. Student must be able to view the announcements provided by the instructor

11. Student must be able to change their password

Medium Priority

1. Student should be able to delete themselves from a course

Low Priority

1. Student should be able to view the grade currently receiving in course

2. Student should be able to calculate what they would need to get on the final in order to estimate their grade

3. Student should be able to view the exam material given in the course
Instructors:

**High Priority**

1. Instructor must be able to Add syllabus
2. Instructor must be able to Edit the syllabus
3. Instructor must be able to Add a lecture
4. Instructor must be able to Edit a lecture
5. Instructor must be able to Delete a lecture
6. Instructor must be able to Add an assignment
7. Instructor must be able to Edit an assignment
8. Instructor must be able to Delete an assignment
9. Instructor must be able to Update grades for assignments
10. Instructor must be able to Update student’s login information
11. Instructor must be able to Email all students
12. Instructor must be able to Email students individually

**Medium Priority**

1. Instructor should be able to Set the date the assignment will show
2. Instructor should be able to set the date when the assignment will not show.

3. Instructor should be able to set the date the lecture will show.

4. Instructor should be able to set the date when the lecture will not show.

5. Instructor should be able to select which options they want available.

6. Instructor should be able to set the percentage weight associated to assignments and exams.

7. Instructor should be able to delete a student from the course.

8. Instructor should be able to upload a file to associate to another content area.

Low Priority

1. Instructor should be able to add examination information (online exam).

2. Instructor should be able to edit examination information.

3. Instructor should be able to delete examination information.

4. Instructor should be able to import a list of students from the Banner application.

General Administrators:

High Priority

23
1. Administrator must be able to Add a course
2. Administrator must be able to Edit a course
3. Administrator must be able to Delete a course
4. Administrator must be able to Add an instructor
5. Administrator must be able to Delete an instructor

Low Priority

1. Administrator should be able to Edit an instructor
   a. Change Name of Instructor
   b. Change Email of Instructor
   c. Change Username of Instructor

All high priority functional requirements were implemented and were functional.
Some of the medium priorities were implemented because they were requested by a user or its priority was upgraded to a high priority. The feature that was implemented as an enhancement was the upload file feature. This was requested by the other instructor involved in this study, and the rationale was justified enough to include it in this project. Instructors may want to add outside resources to supplement their lectures or assignments with uploaded files. Some instructors create slide presentations or PDF files that they would like to make
available to students. With the upload a file function, the instructor could upload and link multiple files to numerous lectures, assignments, and/or announcements.

Lastly, in an effort to anticipate future enhancement, the database structure was developed allowing for the medium priorities and low priorities to be implemented with a lower maintenance cost. Since most of this project is database dependent, it was crucial to make sure the database was as extensible as possible.
Chapter 3

Design

Presentation Layer

This project was separated logically into a three-tiered solution; the presentation layer (PL), the business layer (BL), and the data layer (DL). Each layer plays a significant role in most development projects and this project adhered to these rules as well.

The PL is typically defined at the “look and feel” layer. This is where the user interface is defined. This is where colors and navigation flow are most important. The PL is generally separated to allow for different platforms and mediums to be accommodated more readily. This layer also provides the interface between the user and the business logic. The main goal of the PL is to display the information that is returned by the BL. The PL is where usability testing plays a vital role and provides valuable information. This layer is often viewed as an enhancement; however, ignoring the importance of this layer can be detrimental to the whole project. The PL is the gateway to the user. If the user’s perception is poor at this level, any amount of functionality will be immediately forgotten because the user will not return to continue with this product.
In the OIC, this layer was addressed most heavily with the Student Area, which contained the largest user base for the alpha version. The goal was to provide a simple interface where navigation was unambiguous and effective. In order to provide an appealing association, the color scheme matched the current colors used at The University of Montana's website. The dynamic portions of the site were written so that black text would appear on a white background to ensure that most users were able to see the contrast. Additionally, alt tags and title tags were attached to all buttons and graphics to help identify their function.

The logical flow of the pages was built such that there would be one common start page with a login. After a user was authenticated, depending on the specific role of the user, their direction would be dictated based on processing that takes place in the BL (Figure 14).
The BL (Figure 15) works under the PL and interacts with the DL. Typically all business rules are encapsulated here. These rules can be abstracted into components so that only the results of their functions are returned to the PL. This layer is one of the most important layers because it performs all the actual processing in the product. The BL incorporates the rules and parameters that are
established by the client and reflect the expected results of various tasks. The BL can be specific to a machine platform meaning that one might need to change or alter the code in the BL to allow the same functionality on another system.

Figure 15

29

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
In the OIC, all the supporting classes called by the CSPortal class encompassed the BL. These classes exposed the information requested by a user. The OIC consisted of 12 supporting classes to the main interface class. Each class encapsulated the functionality that related to a portion of the system. In the design, it was most beneficial to focus on the classes that were being exposed such as the syllabus, lectures, and assignments.

**Data Layer**

The last tier is the DL (Figure 16). This is the data repository. Its main function is to retain data and expose data when called, and no business rules should be applied here. The BL should be calling the data that it needs and applying business logic to the data returned. Data flows between the layers, but the DL should be sheltered from any actual calculations on the data. The DL should be optimized to return data through indexing and stored procedures. Additionally, through the normalization of the tables, duplication is eliminated reducing the size of the data storage and the efficiency with which the data is returned. These techniques all lead to an increase in performance and ultimately lower maintenance of the database.
In the OIC, this layer consisted of a relational database with 14 tables containing information about a user, courses, assessment, lecture material, assignment material, as well as announcements that are specific to a course or to the general population. This information was exposed through the execution of stored procedures.
Justification of the 3-tiered System

Keeping these layers separated allowed for a more scalable system able to handle an increase in the number of users, thereby decreasing the load on each individual server. Additionally, one can increase server capacity in areas that are found to be overloaded instead of all areas resulting in decreased costs and wasted resources. This is most applicable if the tiers are separated physically. By separating each layer onto a separate physical server, the administrator can easily extend the system by adding an additional server to the overloaded layer.

In a layered approach such as this, one also decreases the amount of maintenance required to keep a product running or to correct errors. This is especially important in areas where multiple developers will be working on the same project. Typically, designers can work on the PL concurrently with component developers working on the BL, and finally database developers working on the DL. This allows for more rapid and reliable development over the life of the project [5].
Physical Design

Presentation Layer

The physical path from the login was outlined in Figure 17. This physical layout of pages determined the initial choices of the user. The user had the ability to sign up as a new user, retrieve their forgotten password, or login. Once the user logged in, they were directed to the proper path which is dependent on their role in the system. They were redirected into either the administrator role, the instructor role, or the student role. Additionally, they might have utilized a combination of instructor and student roles depending on their position in the department.
The most prominent path was the Student Path (Figures 18, 19). This path was primarily a read-only area where the student could view the information that was uploaded by the instructor. Initially, the student needed to add the course to which they belong. After this had been completed, they were able to see the syllabus, assignments, lectures, exam information, their grades, and had the ability to send an email to the instructor. Each section in a course had its own physical page that dynamically displayed course information. Additionally, students had the ability to change their email address as well as their password. Each of these functions also had their own physical page. All information displayed on each
page was specific based on the login of the user. All of these functions were available through horizontal navigation showing the user what had been chosen after it had been selected. The navigation was contained in one single file which was included in all other pages. This increased reusability and decreased maintenance since any changes were applied in only one file, but affected many files.

![Student Path Diagram](image)

Figure 18
Another major path was the Instructor Path (Figures 20, 21). This allowed the instructors to administer the courses under their direct supervision. Once a course had been selected, an instructor had the ability to add, edit, or delete course information. The instructor also had the ability to send emails to individual students as well as all the students that had enrolled in the course.
Again, all major content areas had their own files dynamically generating information. These files affected only specific course information. This structure provided some insulation from changes that occurred on other pages and provided a simple schema to follow and update if necessary.

Sunday, October 27, 2002

Instructor Path

![Diagram]

Figure 20

37

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 21
The Administrator Path (Figure 22) was the third role that existed in the OIC. This role controlled the courses that were offered in the OIC. As well, it determined which users were designated as instructors. The administrator also could delete all the previous data from the database. Lastly, the administrator could add and update general announcements that all users of the OIC could view. Each function was separated into a different physical file for simplicity and reduction of maintenance.
Figure 22
The physical design of the database (Figure 23) was engineered using Visio.NET creating all tables and their relationships. Additionally, all the column names and data types were defined and saved in the diagram. Using this diagram, a script was generated that was executed in the relational database management system. The benefit of using this design tool was that changes could occur in the database design, and then a new script could be produced with these new changes implemented. This reduced documentation maintenance because of the integration inherent between the products.
Figure 23
The code design showed that there was one major class with 13 support classes. These classes interacted such that all classes were dependent on the Connection Class, with the CSPortal Class dependent on all other classes except the Connection Class. The other class that was used most often by every class was the CommonFunction class. The CSPortal Class (Figure 24) was the main interface class. This class instantiated all the supporting classes, and then used the return information to display, insert, update, or delete the appropriate information on a page. All code design documents were created in Rational Rose (see Figures 25-35 in Appendix A). Since most of the functionality was modular, the supporting classes were broken into subgroups showing the specific dependencies. Additionally, all class interactions were outlined in a series of sequence diagrams (see Figures 36-66 in Appendix A) describing common functions in the OIC [4].
Figure 24
This design indicated that if another area were added to the site, such as discussion boards, a new class could be formed and the dependencies would most likely be with the Connection Object and the CommonFunctions class. Although an extensible addition such as this would not increase the maintenance costs significantly, one must factor in the ramifications of making changes to either the Connection class or the CommonFunctions class, since they are so highly used.

The sole purpose of the Connection class was to generate a connection string that could be used to open a connection to a database. This class does not have to change if the application is running on a system that would allow reading from the registry. By creating a private function that accesses the registry, the Connection class could look to the registry to find the Database and the Server name to insert into the connection string. The benefits of a model such as this were reflected in the fact that one does not have to recompile this class and its dependents whenever the server or the database names change or move. This reduces the maintenance costs of the product. However, if the system or the administrator do not allow registry manipulation, the connection must be hard coded into the Connection class. Therefore, whenever the connection string changes, the Connection class would need to be recompiled as well as any class that is dependent on the Connection class. This increases the maintenance of the
overall project, but having the connection string in one central class minimizes the effects.

If one examines the reusability of this design, it was readily apparent that a generic version of the Connection class was the most reusable because it did not depend on any other classes to function correctly. Conversely, the CSPortal class was the least reusable because it is tied to the specific web pages and their server controls that are calling the functions of the supporting classes. The overall reusability of the other supporting classes is fairly high because they were dependent only on the Connection class. Therefore, implementing another supporting class was dependent only on importing the Connection object and then providing the correct functionality.
Implementation

In this project, the PL is associated to W3C accepted standards. The main part of the presentation layer is written in HTML and JavaScript. This gives flexibility in accommodating a variety of browsers, which will allow greater access. The types of browsers that the OIC has been designed for include Internet Explorer (IE) 5 and above and Netscape 6 and above. These browsers were chosen because of their prevalence in the marketplace. They account for the majority of users that would be utilizing this system.

For this project, the BL is written in VB.NET, ASP.NET, while ADO.NET is leveraged to access the DL. These tools allow greater functionality than their predecessors and support physical separation of PL and BL more definitively [6][7][8]. Using ASP.NET allows construction to happen in a more separate manner. Previously, in ASP (Active Server Pages) 3.0, the BL would be intertwined with the PL, creating more maintainability issues. ASP.NET alleviates this problem by physically removing the code to be executed from the display of the information returned. Therefore, one can make updates to the BL and have minimal impact on the PL [7].

47
OIC leverages Microsoft SQL Server 2000 as its Database Management System (DBMS). This DBMS was chosen because of the optimization created between ASP.NET and SQL Server 2000. The drivers for database connectivity allow for increased performance with SQL Server 2000. Additionally, this DBMS integrates with Visio.NET allowing for faster database design and creation. SQL Server 2000 is equipped to handle databases that hold terabytes of data. This allows the users of this DBMS to work without constraints of programs like Microsoft Access or any other small scale DBMS. It is anticipated that the OIC will never reach this limitation.

This DBMS also supports the usage of store procedures, which provide a two-fold benefit. By using stored procedures, the queries to the database become compiled code resulting in faster execution. Stored procedures also decrease the amount of maintenance because if the query changes or a different parameter needs to be returned, the stored procedure can be changed and the components are affected immediately. Unless the interface to the stored procedure is altered or the return value is a new type, the compiled business logic will not have to be recompiled to reflect these changes. The stored procedures become the interface between the DL and BL.
Lastly, this DBMS is also easier to maintain. As with most windowing tools, the task of backing up the database can be done through a wizard-type mechanism where one can back up to a file on the local server or a backup device. Additionally, this maintenance can be set up on a scheduled timeframe therefore decreasing the amount of time an administrator would need to maintain the system.
Chapter 5

Testing

The test plan was given to two independent testers to be executed. These testers ran through the test plan identifying defects and errors in the system. These tests were run before the system was released to the students and instructors. The results of these tests were outlined in the subsequent descriptions as well as in Appendix B.

Test 1

In the Initial testing script, there were 55 test cases performed. Of those cases, there were 11 defects and 5 errors detected. All defects and errors were examined and corrected. These tests were run by two independent volunteers that agreed to run through the test cases that were created and add test steps, if appropriate. These two volunteers were well versed in testing procedure and provided valuable feedback on a variety of issues. It was concluded that this test plan was effective given the errors and defects that were uncovered.
Test 2

After the initial round of errors and defects were corrected, the product was released as an alpha version. Suggestions and issues were found by an instructor and students were recorded in the usability testing and summarized below.

Usability Suggestions

1. File Upload Area for Instructors
2. Display more information about the student on the Remove Student page
3. Instructors should be able to change not only their password, but other information like email as well
4. Students should be able to change not only their password, but other information like email as well

Usability Defects

1. Page did not refresh when a lecture was added or updated
2. Page did not refresh when a homework was added or updated
3. Page did not refresh when announcement was added or updated
4. Page did not refresh when grades were updated
5. Email function for the instructor did not send mail
6. Forgot password function did not send mail
7. Students could add into the same course multiple times
8. Student could delete themselves from a course without a confirmation prompt

To summarize in quantitative terms, it was found after the usability round of testing that there were two errors and six more defects discovered. There were four suggested enhancements as well.

Testing Conclusion

Although more formal testing needs to be performed on this system, the comparison between the initial testing scripts and the usability testing issues that arose was quite promising. The number of errors decreased by over half the original number and the number of defects was down by almost half as well. These results must be viewed with caution, however, because in the second round of testing there was not a number of test cases and test steps with which one could normalize against. Therefore, the results may be skewed in a positive direction.

However, even though the users did not have a formal testing script to follow, considering the functionality that was used, they were able to test the majority of
the SA through normal usage. This provided a good cross section of issues that might arise. Additionally, since the instructors using the system implemented its use in the classroom, most of the IA major functions were utilized in a reasonable manner. This would indicate that the most common uses of the system were tested well throughout the semester.
Chapter 6

Usability Questionnaire

An additional measure that was added to this project was the use of a usability questionnaire (Appendix C). Since this project was released to two different courses, the quantitative feedback would be valuable. Between the two courses, there would be approximately one hundred student users and two instructor users. At the time of this release a designated administrator was not identified, so the developer of the system acted as the administrator as well. The questionnaire’s goal was to assess the simplicity of the OIC system, which had implemented the design heuristics to simplify the structure of tasks, make things visible, and exploit the power of constraints [9]. The quantified results were reported in Table 1.
Usability Results

<table>
<thead>
<tr>
<th>Question</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Mean</th>
<th>Variance</th>
<th>Std.Dev.</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td>4.64</td>
<td>2.91</td>
<td>1.71</td>
<td>25</td>
</tr>
<tr>
<td>#2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>21</td>
<td>4.88</td>
<td>0.67</td>
<td>.82</td>
<td>25</td>
</tr>
<tr>
<td>#3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>19</td>
<td>4.71</td>
<td>1.92</td>
<td>1.39</td>
<td>24</td>
</tr>
<tr>
<td>#4</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>11</td>
<td>3.95</td>
<td>8.35</td>
<td>2.89</td>
<td>22</td>
</tr>
<tr>
<td>#5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>12</td>
<td>4.86</td>
<td>0.42</td>
<td>.065</td>
<td>14</td>
</tr>
<tr>
<td>#6</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>18</td>
<td>4.64</td>
<td>2.52</td>
<td>1.59</td>
<td>25</td>
</tr>
<tr>
<td>#7</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>9</td>
<td>10</td>
<td>4.08</td>
<td>4.85</td>
<td>2.20</td>
<td>25</td>
</tr>
</tbody>
</table>

Table 1

Questions

1. Logging into the CSPortal was:
2. Adding into a course was:
3. Overall, finding course related material (navigating) was:
4. Using “Contact” to email the Instructor was:
5. Changing my password was:
6. Viewing your grades using the CSPortal was:
7. The viewable announcements before logging in(job postings, scholarships, etc.) were:
Questionnaire Results

There were eight quantifiable questions and three free form questions. The results of the first seven questions are tabulated in Table 1. The sample sizes ranged predominantly from 22-25 with one question with a size 14. The calculated standard deviation for all the questions was under three. The first five questions addressed the functionality of the OIC site and examined its ease of use. The results overwhelmingly showed that the site is simple to navigate and utilize. The only questions that contained lower scores were questions 4 and question 7. Question 4 dealt with the ease with which one could email the instructor. The lower mean value and higher standard deviation was most likely
due to the technical issues that were experienced by some when using the product. As this product matures, it could be extrapolated that these students would move their responses towards a more favorable response since the defect was corrected. In regards to question 7, it is possible that some students did not find these announcements useful or applicable. This might also be the case that students have not had enough applicable announcements filter through the system.

The eighth and ninth questions were written to find out how many students had used other online systems and what system they had utilized. Approximately half of the students that responded indicated that they had used another online system, and the majority had used Blackboard.

The tenth question was written to decipher whether or not departments should adopt online systems. The overwhelming response was yes and the reasoning was because the students appreciated being able to access course information online whenever they needed to. It was also stated a few times, however, that students are only interested in having online courseware as a supplement to a traditional course and not for online instruction.

The eleventh question was directed at gathering more information about what functionality might be missing from the OIC site. Most respondents indicated
that they would appreciate a new discussion board function. This functionality could also be augmented by sending out email alerts to students when a new post has been uploaded to the discussion board. It was suggested that there be a student index that shows the names of students enrolled in the course (This could cause privacy concerns however). A request for a digital drop box where students can upload their assignments was suggested as well. Another suggestion was to allow lectures to be downloaded into a text file and not just exposed as a web page. Lastly, one student indicated an interest in seeing how grades were compared to the rest of the class. The last question was a free form question that allowed students to write additional comments. Most comments were quite positive in support of using the OIC. The reasoning focused on the simplicity and the ease of navigation. There were a couple user interface design suggestions which would enhance the navigation of the site.
Chapter 7

Conclusion

Initially this project was directed at replacing the current web presence that exists in the Computer Science Department. Currently, the Computer Science Department allows the instructors to create and maintain their own web sites statically. There are many disadvantages to this process. The most apparent disadvantage is that instructors do not have time to update these pages, and they either become out of date or they have to be maintained by an intern student. Another disadvantage is that students must search through a directory of classes to find the course that they are taking. This can be somewhat confusing and time consuming. This project addresses these issues by creating a portal where students and instructors can go to access information that pertains to the department. For the instructors, tools exist that allow easy addition, editing, or deletion of course information. With these tools they do not have to go into any of the HTML pages and manipulate them. The major benefit is that it frees the instructor to do what they do best, teach courses. For the student, the benefits are that they can go to one place and see any departmental announcements that might be available, and then view all the courses that they are taking in the Computer Science Department. This relieves the searching component of the
previous process. Additionally, they can contact their instructors without having to access an email client. Furthermore, all of their grades are individualized for each class providing another level of privacy.

Examining the Usability Questionnaire results show that the OIC is a viable candidate in this decision process. Additionally, there was tremendous support for adopting a product that would allow students to view all their course information in one central location. This study suggests that customizable products should be considered in the Computer Science Department however, a more formalized study would need to be administered to confirm these findings.
Chapter 8

Future Direction

This project has an incredible number of directions it could go. Depending on the focus of future work, a multitude of enhancements could be incorporated. One of the most beneficial enhancements would be to incorporate a process that would allow the instructor to weight the grades that are assigned to a class. Initially, this seems as if it would be quite trivial to incorporate, however, it depends on how it will be used. Some possible scenarios are as follows.

One can weight each assignment, but then the instructor is forced to determine the weight of each individual assignment and exam that is given. This scenario works wonderfully, if one knows in advance how many assignments and exams will be given. However, many instructors are more fluid with their curriculum and tailor their courses towards the needs of the students, creating changes in how they assign homework and administer exams. Therefore, simply assigning weights to each individual assignment can be very challenging for the instructor.

Another approach is to assign a weight to all homework and tests as a group. This is more typical for most instructors, but the final grade cannot be tabulated
until all homework and exams have been corrected and entered into the system. This solution works, but could not possibly give an accurate standing for a student until all the scores had been added. The most challenging part of this scenario is if one were to consider trying to show the student what their grade was so far through the system, but they had not taken any exams yet. This would lead to difficulty because the instructor would have to assume that the homework was worth one hundred percent of the grade, while actually, they wanted it to only be weighted fifty percent. If these percentages were established in the beginning, the final percentage would be incorrect and it would create a situation wherein the student could not depend on what the system indicated for their final grade.

Based on these two scenarios, it was determined that it was not appropriate to incorporate this piece of functionality at this time. However, it would present a fascinating endeavor for another graduate student to tackle in the future.

Another piece of value-added functionality that could be incorporated into this system would be an alert system. With the prevalence of cell phones, personal digital assistants, and other mobile devices, alerts to these devices would be quite beneficial to the student. As students become more scheduled, it becomes more difficult to keep track of all of the commitments one has made. If one could receive an alert on their cell phone indicating a due date or an exam date, it would
increase the chance of meeting and preparing for key deadlines by the student, and provide yet another tool that would allow a student to excel.

Another piece of functionality that could be incorporated in this project fairly easily would be the idea of a digital drop box. This feature was also requested by students in the questionnaire. This functionality currently exists with Blackboard and other online courseware. This feature was intentionally disregarded because it was determined that if one wanted this type of functionality, they would have to first evaluate what the needs were of the department and examine how many instructors would find this feature useful.

Another functional unit that could be added is a discussion board. This would allow students to post questions and read responses from other classmates regarding material in the course. This is a feature that is also provided in other online courseware, and would most likely find success in this product as well. The additional piece that was requested and is not offered on these commercial products would be emailing the student when a new comment has been posted on the discussion board.

Another powerful feature that could be added to this system would be integrating the student rosters with course enrollment forms from the Registrar’s Office. This would alleviate students having to log in and create another username and
password. However, this might be quite challenging because the department would have to set up a partnership indicating that the request is legitimate.

Another improvement to the site would be to give the instructors a way to preview all the work they have done. With the current design, the instructor must be a student (even a fictitious student) in order to view the information they have just recently uploaded. Since the system is geared to be more efficient than other commercial products, a preview page would benefit the instructor and again reduce the amount of time required to use the OIC. If the instructor does not have to navigate away from their administration page, they might be more apt to embrace the technology and fully utilize the functionality.

Another potential usage of this project would be to have the Computer Science Department host many customized sites such as the CSPortal, and then sell the service to other departments. This could result in a potential revenue source for the department. Since this is a free product for the department, it would alleviate the additional costs of using a commercial product. Finally, by selling this service to other departments, it would offset any administration costs that the department might incur.
REFERENCES


65


16. Microsoft Developers Network, Visual Basic.NET
   http://msdn.microsoft.com/vbasic/, 2003, Microsoft Corporation
APPENDIX A

Design and Sequence Diagrams

Figure 25

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 26
Figure 27
Figure 28
Figure 29
Figure 30
<table>
<thead>
<tr>
<th>CSPortal</th>
<th>CommonFunctions</th>
</tr>
</thead>
<tbody>
<tr>
<td>★UpdatePassword();</td>
<td>★CheckSessionID();</td>
</tr>
<tr>
<td>★SendMessage();</td>
<td>★GetAILEmailAddresses();</td>
</tr>
<tr>
<td>★SetNavigationLinks();</td>
<td>★GetSendersEmailAddress();</td>
</tr>
<tr>
<td>★InsertAnnouncement();</td>
<td>★GetInstructorEmailAddress();</td>
</tr>
<tr>
<td>★PopulateAnnouncement();</td>
<td>★ChangePassword();</td>
</tr>
<tr>
<td>★InstructorNavigationLinks();</td>
<td>★CheckValidOldPassword();</td>
</tr>
<tr>
<td>★InitializeStudents();</td>
<td>★CheckNewPasswords();</td>
</tr>
<tr>
<td>★InitializeDates();</td>
<td>★GetAllNonInstructorUsers();</td>
</tr>
<tr>
<td>★InsertExam();</td>
<td>★GetUserID();</td>
</tr>
<tr>
<td>★PopulateExam();</td>
<td>★DeleteAllInstructorStudents();</td>
</tr>
<tr>
<td>★UpdateGrades();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★PopulateHomework();</td>
<td>★DeleteAllInstructor();</td>
</tr>
<tr>
<td>★InsertLecture();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★PopulateLecture();</td>
<td>★DeleteAllInstructor();</td>
</tr>
<tr>
<td>★InsertSyllabus();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★PopulateSyllabus();</td>
<td>★DeleteAllInstructor();</td>
</tr>
<tr>
<td>★UploadNewFile();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★DisplayFile();</td>
<td>★DeleteAllInstructor();</td>
</tr>
<tr>
<td>★SetAdministrationNavigationLinks();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★GetCourse();</td>
<td>★DeleteAllInstructor();</td>
</tr>
<tr>
<td>★FillCourseInfo();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★InsertCourse();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★ConfirmDelete();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★DeleteData();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★GetAnnouncement();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★FillAnnouncementInfo();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★InsertGeneralAnnouncement();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★GetInstructors();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★GetUsers();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★InsertInstructor();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★DeleteInstructorCourse();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★DeleteStudentCourse();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★SetPassword();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★Create();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★DisplayStudentCourse();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★DisplayInstructorCourses();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★DisplayInstructor();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★DisplayStudent();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★InsertNewStudentCourse();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★InsertNewInstructorCourse();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★Login();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
<tr>
<td>★GetAnnouncements();</td>
<td>★DeleteAllNonInstructor();</td>
</tr>
</tbody>
</table>

Figure 31

73

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 32
Figure 33
Figure 34
Figure 35
Sequence Diagram: Use Case View / GetCourseInformation

Figure 36
Sequence Diagram: Use Case View / InsertCourseInformation

CSPortal  CommonFunctions  Course  Connection

CheckSessionID()

InsertNewCourse()

GetConnection()

Figure 37

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 38
Figure 40
Figure 41
Figure 42
Figure 43
Sequence Diagram: Use Case View / GetStudentCourse

Figure 45

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Sequence Diagram: Use Case View / InsertStudentCourse

Figure 46

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 47

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 48
Figure 49
Figure 50
Figure 51
Figure 52
Sequence Diagram: Use Case View / Delete Syllabus

CSPortal

CommonFunctions

Syllabus

Connection

Figure 53
Figure 54
Figure 55
Figure 56

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 57
Sequence Diagram: Use Case View / InsertHomework

CSPortal → CommonFunctions

CheckSessionID() → InsertNewHomework() → GetConnection()
Sequence Diagram: Use Case View / DeleteHomework

- CSPortal
- CommonFunctions
- Homeworks
- Connection

CheckSessionID() → DeleteHomework() → GetConnection()

Figure 59
Figure 61
Figure 63
Sequence Diagram: Use Case View / Update Grades

CSPortal — Grades — Connection

- CSPortal
- CommonFunctions
- Grades
- Connection

CheckSessionID!

UpdateAssignmentGrades!

GetConnection!

Figure 64

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Sequence Diagram: Use Case View / Delete Grades

Figure 65
**APPENDIX B**

Test Case Results

**Test Case 1: Administrator Area**

<table>
<thead>
<tr>
<th>Case Number</th>
<th>Related Steps</th>
<th>Step Description</th>
<th>Expected Results</th>
<th>Pass/Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Add a Course with no other courses inserted</td>
<td>Course should appear in the display</td>
<td>Pass</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Add a course after another course has been inserted</td>
<td>Course should appear in the display along with other course(s) inserted already</td>
<td>Pass</td>
</tr>
<tr>
<td>3</td>
<td>1, 2</td>
<td>Modify a course that has already been inserted, by changing the name, the title and the toggle the display checkbox</td>
<td>Modifications to the course should appear take place</td>
<td>Pass</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>Delete a course that has been inserted with other courses inserted</td>
<td>Course should no longer appear in the display, however the other courses should remain</td>
<td>Fail/Pass, when clicking on the delete link a first time the course is not deleted. However, clicking the link a second time does delete the course</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>Delete a course when there is only one course in the list</td>
<td>Course should no longer appear in the display</td>
<td>Fail/Pass, when clicking on the delete link a first time</td>
</tr>
<tr>
<td>Step</td>
<td>Task Description</td>
<td>Expected Outcome</td>
<td>Actual Outcome</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
<td>------------------</td>
<td>---------------</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Add a blank course name</td>
<td>An error message should notify the user that a course name is needed</td>
<td>Fail, a blank course was added to the display</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Add a blank course description</td>
<td>An error message should notify the user that a course title is needed</td>
<td>Fail, a course with a blank description was added to the display</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Add instructor from the drop list with names in the list</td>
<td>Instructor should appear in the display</td>
<td>Pass</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Delete an instructor from the list</td>
<td>Instructor should not appear in the display, however the other instructors should remain</td>
<td>Pass</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Go to Login for that instructor and see if they can add courses they are teaching</td>
<td>Instructor Administration section should appear in their account. The instructor should be able to add a course to this section.</td>
<td>Pass</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Delete All data for the semester</td>
<td>All data associated to all courses will be deleted</td>
<td>Pass/Fail, deleted all the data, but deleted the admin account</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Add a new announcement with no other announcements entered</td>
<td>Announcement should appear in the display</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Edit announcement that has been previously entered, change the announcement, display checkbox</td>
<td>Modifications to the announcement should take place</td>
<td>Pass</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Delete a previously entered announcement</td>
<td>Announcement should not appear in the display, however other announcements should still appear</td>
<td>Pass</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Delete the last announcement</td>
<td>Announcement should not appear in the display</td>
<td>Pass</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Change password with non-matching new passwords</td>
<td>Error message should notify the user that the new passwords do not match, password should not be changed</td>
<td>Pass</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Change password with incorrect previous password</td>
<td>Error message should notify user to use the correct old password, new</td>
<td>Pass</td>
<td></td>
</tr>
<tr>
<td>Case Number</td>
<td>Related Steps</td>
<td>Step Description</td>
<td>Expected Results</td>
<td>Pass/Fail</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------</td>
<td>------------------</td>
<td>------------------</td>
<td>----------</td>
</tr>
<tr>
<td>18</td>
<td></td>
<td>Change password with password blank</td>
<td>Error message should notify user to enter a new password, blank password should not take effect</td>
<td>Fail, a blank password is accepted as the new password</td>
</tr>
<tr>
<td>19</td>
<td></td>
<td>Add a course as an instructor from dropdown list</td>
<td>Course should appear in the Instructor Administration section of the instructors account</td>
<td>Pass</td>
</tr>
<tr>
<td>20</td>
<td>19</td>
<td>Delete a course for an instructor</td>
<td>Course should no longer appear</td>
<td>Pass</td>
</tr>
<tr>
<td>21</td>
<td>19, 20</td>
<td>Add course previously deleted and see if information was previously entered for the course persists</td>
<td>None of the previous data for the course should appear</td>
<td>Fail, the previous data for the course still exists</td>
</tr>
</tbody>
</table>

**Test Case 1: Instructor Area**

<table>
<thead>
<tr>
<th>Case Number</th>
<th>Related Steps</th>
<th>Step Description</th>
<th>Expected Results</th>
<th>Pass/Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td></td>
<td>Select a course as an instructor</td>
<td>Should lead to the administrative area for the course</td>
<td>Pass</td>
</tr>
<tr>
<td>23</td>
<td></td>
<td>In instructor area, add a new announcement</td>
<td>Announcement should appear in the announcement section of the administrative area</td>
<td>Pass, it should be noted that the preview function does not seem to work. The announcement can be accessed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td><strong>24</strong></td>
<td><strong>23</strong></td>
<td>Instructor, edit a previously entered announcement with new title, description and display, verify change through student’s view</td>
<td>All changed to announcement should appear in the announcement section of the administrative area and in the student view</td>
<td>Pass/Fail, changes to title and description take effect, however unchecking the display box results in deletion of the announcement</td>
</tr>
<tr>
<td><strong>25</strong></td>
<td><strong>23</strong></td>
<td>Instructor delete an announcement previously entered</td>
<td>Announcement should not appear in the announcement section of the administrative area or in the student view. All other announcements should still appear</td>
<td>Pass</td>
</tr>
<tr>
<td><strong>26</strong></td>
<td><strong>23</strong></td>
<td>Instructor delete the last announcement previously entered</td>
<td>No announcements should appear in the administrative area or in the student view</td>
<td>Pass</td>
</tr>
<tr>
<td><strong>27</strong></td>
<td></td>
<td>Add a new syllabus with no other previously entered syllabus</td>
<td>Syllabus should appear in the administrative section and the student view</td>
<td>Pass</td>
</tr>
<tr>
<td><strong>28</strong></td>
<td><strong>27</strong></td>
<td>Edit a previously entered syllabus by changing the description and the display checkbox</td>
<td>Changes to syllabus should appear in the administrative area and the student view</td>
<td>Pass/Fail, changes to the description take effect, however changes to the display box do</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>27</td>
<td>Delete a syllabus previously entered</td>
<td>Syllabus should no longer appear in the administrative area or the student view</td>
<td>Pass</td>
</tr>
<tr>
<td>30</td>
<td>27</td>
<td>Delete the last syllabus previously entered</td>
<td>No syllabuses :) should appear in the administrative area or the student view</td>
<td>Pass</td>
</tr>
<tr>
<td>31</td>
<td></td>
<td>Add a new lecture</td>
<td>Lecture should appear in the administrative area and the student view</td>
<td>Pass</td>
</tr>
<tr>
<td>32</td>
<td>31</td>
<td>Edit a previously entered lecture by changing the title, description, and display box</td>
<td>Changes to lecture should appear in the administrative area and the student view</td>
<td>Pass/Fail, changes to the title and description take effect, however unchecking the display box results in deletion of the lecture</td>
</tr>
<tr>
<td>33</td>
<td>31</td>
<td>Delete a lecture previously entered</td>
<td>Lecture should not appear in the administrative area or the student view</td>
<td>Pass</td>
</tr>
<tr>
<td>34</td>
<td>31</td>
<td>Delete the last lecture</td>
<td>No lectures should appear in</td>
<td>Pass</td>
</tr>
</tbody>
</table>

not (i.e. if display box is not checked, it still appears in the student view)
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>previously entered</td>
<td>the administrative area or the student view</td>
</tr>
<tr>
<td></td>
<td>Add a new homework</td>
<td>Homework should appear to the instructor and the student</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pass, note however that if an instructor attempts to add a homework without filling out the description or total points, a server error results</td>
</tr>
<tr>
<td>36</td>
<td>35</td>
<td>Edit a previously entered homework by changing the title, description, total points, and display box</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Changes to the homework should appear to the instructor and the student</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pass/Fail, changes to the title, description, and total points take effect, however unchecking the display box results in the deletion of the homework</td>
</tr>
<tr>
<td>37</td>
<td>35</td>
<td>Delete a homework previously entered</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Homework should not appear to the instructor or the student</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pass</td>
</tr>
<tr>
<td>38</td>
<td>35</td>
<td>Delete the last homework previously entered</td>
</tr>
<tr>
<td></td>
<td></td>
<td>No homeworks should appear to the instructor or the student</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pass</td>
</tr>
<tr>
<td>39</td>
<td></td>
<td>Add a new exam</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Exam should appear to the instructor and the student</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pass</td>
</tr>
<tr>
<td>40</td>
<td>39</td>
<td>Edit a previously entered exam by</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Changes to exam should</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pass/Fail, changes to the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Changing the title, description, and display box</td>
<td>appear to the instructor and the student</td>
<td>title and description take effect, however unchecking the display box results in deletion of the exam</td>
</tr>
<tr>
<td>41</td>
<td>39</td>
<td>Delete an exam previously entered</td>
</tr>
<tr>
<td>42</td>
<td>39</td>
<td>Delete the last homework previously entered</td>
</tr>
<tr>
<td>43</td>
<td>35</td>
<td>Add a grade to an existing student for an existing homework</td>
</tr>
<tr>
<td>44</td>
<td>35, 43</td>
<td>Change the grade that an existing student received on an existing homework</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
<table>
<thead>
<tr>
<th></th>
<th>Task Description</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>45</td>
<td>Export grades to an Excel spreadsheet</td>
<td>All student and grade data should appear in an Excel spreadsheet</td>
</tr>
<tr>
<td>46</td>
<td>Send an email as an instructor to an existing student</td>
<td>Instructor should be notified that email was sent. Student should receive email at the appropriate email address</td>
</tr>
<tr>
<td>47</td>
<td>Remove an existing student from a course</td>
<td>Student should no longer appear to the instructor for that course</td>
</tr>
<tr>
<td>48</td>
<td>Remove the last student from a course</td>
<td>No students should appear to the instructor for that course</td>
</tr>
<tr>
<td>49</td>
<td>Update a password with non-matching new passwords as an instructor</td>
<td>Error message should notify the user that the new passwords do not match, password should not be changed</td>
</tr>
<tr>
<td>50</td>
<td>Update a password with incorrect previous password as an instructor</td>
<td>Error message should notify user to use the correct old password, new password should not take effect</td>
</tr>
</tbody>
</table>
| 51 | Update a password with blank password                                         | Fail, a blank password is
<table>
<thead>
<tr>
<th>Case Number</th>
<th>Related Steps</th>
<th>Step Description</th>
<th>Expected Results</th>
<th>Pass/Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>52</td>
<td></td>
<td>Update a password with non-matching new passwords as a student</td>
<td>Error message should notify the user that the new passwords do not match, password should not be changed</td>
<td>Pass</td>
</tr>
<tr>
<td>53</td>
<td></td>
<td>Update a password with incorrect previous password as a student</td>
<td>Error message should notify user to use the correct old password, new password should not take effect</td>
<td>Pass</td>
</tr>
<tr>
<td>54</td>
<td></td>
<td>Update a password with password blank as a student</td>
<td>Error message should notify user to enter a new password, blank password should not take effect</td>
<td>Fail, a blank password is accepted as the new password</td>
</tr>
</tbody>
</table>

Test Case 2: Administrator Area

<table>
<thead>
<tr>
<th>Case Number</th>
<th>Related Steps</th>
<th>Step Description</th>
<th>Expected Results</th>
<th>Notes</th>
<th>Pass/Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Add a Course with no other courses inserted</td>
<td>Add a course and the Course # and name appear.</td>
<td>Will allow you to enter a blank course</td>
<td>Pass, Didn’t show the name of the course</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Add a course after another course has been inserted</td>
<td>Add one course then it shows up, then add another course and it shows as well.</td>
<td>When you add a blank record, and then delete it, I think it messes something up.</td>
<td>Pass, no trouble whatever with this part, just don't add a blank record</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Modify a course that has already been inserted, by changing the name, the title and the toggle the display checkbox</td>
<td>Go into the record to modify it the name and the title pop up and change them. Then click submit and it shows the results.</td>
<td>??what does the display toggle do? I'm confused if it's supposed to not display the course if it's toggled off, just confusing.</td>
<td>Pass, Modified a record, changed it to what I want and see notes for display toggle.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Delete a course that has been inserted with other courses inserted</td>
<td>Click on the delete link and then the page should refresh with the deleted record gone.</td>
<td>I clicked on one record and it didn't show up that it was deleted but I then clicked on another to delete it, and it showed the previous one deleted so I'm thinking it's a refresh thing...</td>
<td>Fail, I clicked on the link and The course doesn't show up as being deleted.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Delete a course when there is only one course in the list</td>
<td>I click on the delete link and the list shows nothing in the list.</td>
<td>I clicked on the delete link but the record still shows up.</td>
<td>Pass and Fail</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Add a blank course name</td>
<td>I click on the submit button and no course</td>
<td>Just don't know when you'd want a</td>
<td>Fail</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Description</td>
<td>Expected Outcome</td>
<td>Actual Outcome</td>
<td>Result</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>-----------------</td>
<td>---------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Add a blank course name and something in the course description.</td>
<td>An error message asking me for a course name</td>
<td>The site is still freezing... don't know if your working on it but I'll stop my testing for now</td>
<td>Fail</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Add a blank course description</td>
<td>The course is added to the list.</td>
<td>Typed in a course number and no description and it was added to the list</td>
<td>Pass</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Add instructor from the drop list with names in the list</td>
<td>Choose an instructor out of the list and it is put in the list below.</td>
<td>I have chosen a name and added it to the list, and it shows up below.</td>
<td>Pass</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Delete an instructor from the list</td>
<td>The instructor is deleted from the list.</td>
<td>I clicked on the delete link next to the instructor and the instructor was removed from the list</td>
<td>Pass</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>??? not sure what you mean by that one???</td>
<td>Go to Login for that instructor and see if they can add courses they are</td>
<td>Login and add a course to my list of courses that I'm teaching</td>
<td>Added myself to the system, added myself as an instructor, and</td>
<td>Pass</td>
</tr>
<tr>
<td>12</td>
<td><strong>Delete All data for the semester</strong></td>
<td><strong>All data associated to all courses will be deleted</strong></td>
<td>Clicked on the delete all info, and got a confirmation that all the associated data was deleted. I go into the add edit a course all the courses are still there and it seems to have done nothing.</td>
<td>Fail</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td><strong>Add a new announcement with no other announcements entered</strong></td>
<td><strong>I type CS111 in the textbox and expect to see the announcement show up, with the name and maybe the course number some where</strong></td>
<td>A blank record showed up in the list, no information displayed at all.</td>
<td>Pass/Fail</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td><strong>Edit announcement that has been previously entered, change the announcement, display checkbox</strong></td>
<td><strong>Click on the edit link and I change the text and uncheck the display box.</strong></td>
<td>Changed the text just fine, but it still displayed the announcement even though I unchecked the display check box</td>
<td>Fail</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td><strong>Delete a previously entered announcement</strong></td>
<td><strong>Click on the delete link and the list should update.</strong></td>
<td>Clicked on the delete link and the list was refreshed without the</td>
<td>Pass</td>
<td></td>
</tr>
<tr>
<td>Case Number</td>
<td>Related Steps</td>
<td>Step Description</td>
<td>Expected Results</td>
<td>Notes</td>
<td>Pass/Fail</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------</td>
<td>------------------</td>
<td>------------------</td>
<td>-------</td>
<td>----------</td>
</tr>
<tr>
<td>16</td>
<td>When I add any announcement the list doesn’t display any information about the content of the announcement</td>
<td>Delete the last announcement</td>
<td>Click on the delete link of the last announcement and it is deleted.</td>
<td>I clicked on the last delete link and the last announcement was deleted</td>
<td>Pass</td>
</tr>
<tr>
<td>17</td>
<td>Change password with non-matching new passwords</td>
<td></td>
<td>An error message saying that the two new passwords don’t match</td>
<td>An error message, stating that the two passwords have to match</td>
<td>Pass</td>
</tr>
<tr>
<td>18</td>
<td>Change password with incorrect previous password</td>
<td></td>
<td>An error message saying that the old password wasn’t correct.</td>
<td>An error message stating that the old password is not correct.</td>
<td>Pass</td>
</tr>
<tr>
<td>19</td>
<td>Change password with password blank</td>
<td></td>
<td>An error message saying you must enter a password</td>
<td>It changed the password to a blank password</td>
<td>Fail</td>
</tr>
</tbody>
</table>

Test Case 2: Instructor Area

<table>
<thead>
<tr>
<th>Case Number</th>
<th>Related Steps</th>
<th>Step Description</th>
<th>Expected Results</th>
<th>Notes</th>
<th>Pass/Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>Add a course as an instructor from dropdown list</td>
<td></td>
<td>Select the course and then click the add button which then shows the new course added</td>
<td>The course was added to the list.</td>
<td>Pass</td>
</tr>
<tr>
<td>21</td>
<td>Delete a course for an instructor</td>
<td>Click on the delete link and the course is removed from the list</td>
<td>Clicked on the link and the course was removed</td>
<td>Pass</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Add course previously deleted and see if information was previously entered for the course persists</td>
<td>Add the course and the course is blank.</td>
<td>The course added was blank and the information was deleted</td>
<td>Pass</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Select a course as an instructor</td>
<td>Click on the course and add it to the list</td>
<td>Clicked on the course and it successfully added it to the list</td>
<td>Pass</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>In instructor area, add a new announcement</td>
<td>Add an announcement and see that it was added to the list.</td>
<td>The announcement was added successfully</td>
<td>Pass</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Instructor, edit a previously entered announcement with new title, description and display, verify change through student's view</td>
<td>Change the announcement and see it in students view.</td>
<td>The announcement was changed and works fine.</td>
<td>Pass</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Instructor delete an announcement previously entered</td>
<td>Click on the delete announcement and verify that it was deleted.</td>
<td>The announcement was deleted and verified through the student view</td>
<td>Pass</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Instructor delete</td>
<td>Click on the</td>
<td>Clicked on the</td>
<td>Pass</td>
<td></td>
</tr>
<tr>
<td></td>
<td>the last announcement previously entered</td>
<td>delete announcement and make sure that it was deleted.</td>
<td>delete link and deleted the last entered one then clicked on the first entered one and deleted it and works fine.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>----------------------------------------</td>
<td>-----------------------------------------------------</td>
<td>------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Add a new syllabus with no other previously entered syllabus</td>
<td>Enter information and click on the submit button and submit the syllabus and verify it through student section</td>
<td>Entered the information and verified that the syllabus was there.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>Edit a previously entered syllabus by changing the description and the display checkbox</td>
<td>Edit the information and check to see if it was displayed after changing the display checkbox to be not checked.</td>
<td>The syllabus still showed after I submitted the changed information and tried to check it via the student view.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX C

Post-Usage Questionnaire

The following is text used to gather input from the student. This questionnaire was distributed to the student user population.

This is a questionnaire targeted at finding out information about your usage of the CSPortal that was created by Michael Cassens as his Masters Project as the University of Montana. The answers provided on the questionnaire will in no way be reflected upon you as a student and all answers will be compiled using complete anonymity.

1. Logging into the CSPortal was:

   difficult 1 2 3 4 5 easy

2. Adding into a course was:

   difficult 1 2 3 4 5 easy

3. Overall, finding course related material (navigating) was:

   difficult 1 2 3 4 5 easy

4. Using “Contact” to email the Instructor was:

   difficult 1 2 3 4 5 easy

5. Changing my password was:

   difficult 1 2 3 4 5 easy

6. Viewing your grades using the CSPortal was:

   not beneficial 1 2 3 4 5 very beneficial

7. The viewable announcements before logging in (job postings, scholarships, etc.) were:
8. Have you ever used a product such as the CSPortal before?

YES  NO

9. If you answered YES to question 8, please specify which product you have used.

10. Do you believe departments should adopt an online system to supplement classes?

YES  NO

Please explain:

11. Is there any additional functionality that you wish the CSPortal had?

12. Please feel free to elaborate with any additional comments that you may have.
APPENDIX D

Code and Code Design

Included on CD