

Do It Yourself Student Group Web Site

By

Sarah Pullen

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

March 2006

Do It Yourself Student Group Web Site

by

Sarah Pullen

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

© Copyright 2006 Sarah Pullen

The author grants to the Information Engineering Technology Program permission to reproduce and distribute copies of this document in whole or in part.

Sarah Pullen

Date

Dr. Hazem Said, Faculty Advisor

Date

Patrick C. Kumpf, Ed.D. Interim Department Head

Date

Acknowledgements

I want to thank everyone who helped me throughout the duration of my project. A special thanks goes to Dr. Said for being an awesome advisor. Dr. Said, thanks for giving me great guidance and challenging me along the way. I want to also thank Todd Davis from Seabury Design for providing training videos on how to create a module in DotNetNuke. They were very helpful. Thanks to friends and family who were very supportive of me, were very understanding of my busy schedule, and were there to help out whenever they could. A special thanks to Renee Merz for helping me debug some of my code when I was having difficulties. Sometimes it just takes another person to look at the code to open your eyes on how to solve the problem. Finally, I want to give a special thanks to my mom, my dad, and my boyfriend, James Bartlett, for being my morale support throughout the project. Thanks again everyone! I could not have done it without you!

Table of Contents

Section	Page
Acknowledgements	i
Table of Contents	ii
List of Figures	iv
Abstract	vi
1. Statement of the Problem	1
2. Description of the Solution	2
2.1 User Profiles	3
2.1.1 Host Administrator	4
2.1.2 Student Group Administrators	5
2.1.3 Student Group Members	5
2.1.4 Students	5
2.2 Design Protocol	5
2.2.1 DotNetNuke	5
2.2.2 Administrative Interface	7
2.2.2.1 Administrative Options	9
2.2.3 Help Pages	10
2.2.4 Modules List	11
2.2.5 UC Branding Standards	12
2.2.6 Organizational Scheme	14
3. Deliverables	14
4. Proof of Design	15
4.1 Skins	15
4.2 Templates	20
4.3 Modules	28
4.3.1 Events Module	28
4.3.2 Feedback Module	32
4.3.3 User Account Module	33
4.3.4 PhotoGallery Module	35
5. Design and Development	41
5.1 Timeline	41
5.2 Budget	41
5.3 Hardware	42
5.4 Software	42
5.5 Testing	43
6. Conclusions and Recommendations	45

Appendix A.	46
A 1. Events Module Code Snippet for Showing Details of an Event	46
A 2. User Account Module Code Snippet for Export to Excel	46
A 3. Photo Gallery Upload Files Code Snippet	47
References	49

List of Figures

Figure 1. Use Case Diagram	4
Figure 2. Application Hierarchy	6
Figure 3. DotNetNuke	7
Figure 4. DotNetNuke Administrative Interface	8
Figure 5. "Edit Text" Functionality of the Text/HTML Module	9
Figure 6. Module Help	10
Figure 7. UC Color Palette	13
Figure 8. UC Logo and Symbol with the "safe area" marked in gray	13
Figure 9. "All-Red" skin	16
Figure 10. "Red-Black-Gradient" skin	16
Figure 11. "Black-Gradient" skin	17
Figure 12. "Red-Gradient" skin	17
Figure 13. "Red and Black" skin	18
Figure 14. Choosing Skins	19
Figure 15. Simple template pages and sub pages	21
Figure 16. Home page of the Basic template with sample content	22
Figure 17. Events pages of the Basic template with sample events	23
Figure 18. Photo Gallery page of the Basic template with sample albums and photos	23
Figure 19. Contact Us page of the Basic template with sample content	24
Figure 20. General template pages and sub pages	24
Figure 21. Contact Us page for the General template with sample content	25
Figure 22. Deluxe template pages	25
Figure 23. Documents page of Deluxe template with sample documents when logged in as an Officer	26
Figure 24. Home page of the Deluxe template with the Users Online module and sample content	27
Figure 25. Add/Edit an event	29
Figure 26. Original Events module	30
Figure 27. Modified Events module	31
Figure 28. Sample details of an event	32
Figure 29. Modified Feedback module	32
Figure 30. Modified User Account module, registration page	33
Figure 31. User Account module showing "Download Results as Excel" link	34
Figure 32. Sample downloaded excel file of members information	35
Figure 33. Edit Photo Gallery screen	36
Figure 34. Add a New Album to Photo Gallery	36
Figure 35. Edit an Existing Album	37
Figure 36. Add New Photos to an Existing Album	38
Figure 37. Viewing the Photo Gallery	39
Figure 38. Editing a Photo	40

Table 1. List of Modules included in DotNetNuke and how it is modified	11
Table 2. Specific Module Modifications	12
Table 3. Project Timeline	41
Table 4. Project Budget	42
Table 5. Testing Plan	44

Abstract

The *Do It Yourself Student Group Web Site* is a Web portal customized especially for the College of Applied Science Student Groups. It is based on an open source application called DotNetNuke. This application permits those with no experience in programming a Web site, the ability to create their own dynamic Web site. This portal environment allows multiple student group sites to be created. Four new skins that adhere to UC's branding standards and four new site templates have been added. Several modules have been modified from the original DotNetNuke source and a new module has been created. The customized Events module displays event details in a more user friendly format. The User Account module allows student groups to register members with more relevant information and export the list of members to a spreadsheet file. The newly created Photo Gallery module allows the groups to create albums and upload pictures of their events. A student group can choose among these new skins, site templates, and modules to easily build their site.

Do It Yourself Student Group Web Site

1. Statement of the Problem

There are many student groups at the University of Cincinnati (UC) College of Applied Science (CAS) who do not have a Web site for their group and would like to. There are approximately 12 student groups at CAS, including CAS Student Tribunal and CAS Ambassadors. This does not include the new and upcoming student groups at CAS. Only two of these student groups have Web sites: the Associated Builders and Contractors (ABC) and the Women in Information Technology (WIT) (3, 15). Two other student groups have small descriptions posted on a Web page which is linked to a departmental Web site, but no Web site specifically for their group (2). The main reason most of these groups do not have Web sites is because they do not have the ability to create or update them.

Even the largest group at CAS, the CAS Student Tribunal, does not have a Web site. The Vice President of Tribunal has made it known that many student groups at CAS would like to have their own Web sites (6). This need has also been addressed at several Tribunal meetings.

Student groups' Web sites are great promotional tools to attract prospective members. Since the internet is the most common place people go to search for information, it is one of the best places to advertise. Web sites open up the doors to new students seeking to join a group. They can provide many things; a description of the group, the criteria needed to be a member, information about events, and communications with the members. A typical student group Web site has a welcome page describing the group, an events calendar, photos of the members and events, a member's registration page, and contact information for the officers. Some have more advanced features such as discussion boards, announcements section, feedback forms, and surveys.

2. Description of the Solution

The problem mentioned proves the need for the “Do It Yourself Student Group Web Site” application that all CAS student groups could use easily and conveniently to create their own dynamic Web sites. This application is structured on DotNetNuke. DotNetNuke is an open source application that provides the portal and site infrastructure upon which sites can be built. The “Do It Yourself Student Group Web Site” application makes it easy for student groups to create and maintain their dynamic Web sites without the knowledge of web technologies, such as HTML or ASP.NET.

The DotNetNuke portal and site infrastructure includes an administrative side with a user-friendly interface that allows the student groups to change the main content, headings, and links on their pages. It also provides many modules that can be added to the Web sites.

DotNetNuke does have limitations. One, it is a very general application that contains many features that can be confusing for novice web developers. It does not have modules specific for student groups in particular, nor is it secure due to its open source structure. Developers who are familiar with DotNetNuke could easily hack the application because they know the default database name and logins. Lastly, it does not relate to UC in particular.

The “Do It Yourself Student Group Web Site” application solves these issues. It is an application that is built on DotNetNuke, that modifies the original application to make it specific for CAS Student Groups and to follow the UC branding guidelines. Some measures have been taken to secure the database to make it more difficult for those familiar with DotNetNuke’s default logins to access the application. The default database name and default logins have been changed. Some unnecessary modules, skins and templates have been removed to make the application less complicated. Several modules have been modified from the original DotNetNuke

source, and a new Photo Gallery module has been created. More details on modules can be found in the modules section, 4.3. Also, four new skins and templates have been created according to UC Branding standards which are described in more detail in the Skins and Templates sections.

2.1 User Profiles

There are four different sets of users for this application: Host Administrator, Student Group Administrators, Student Group Members, and Students. The users and roles are denoted in the Use Case Diagram on the next page (See Figure 1.).

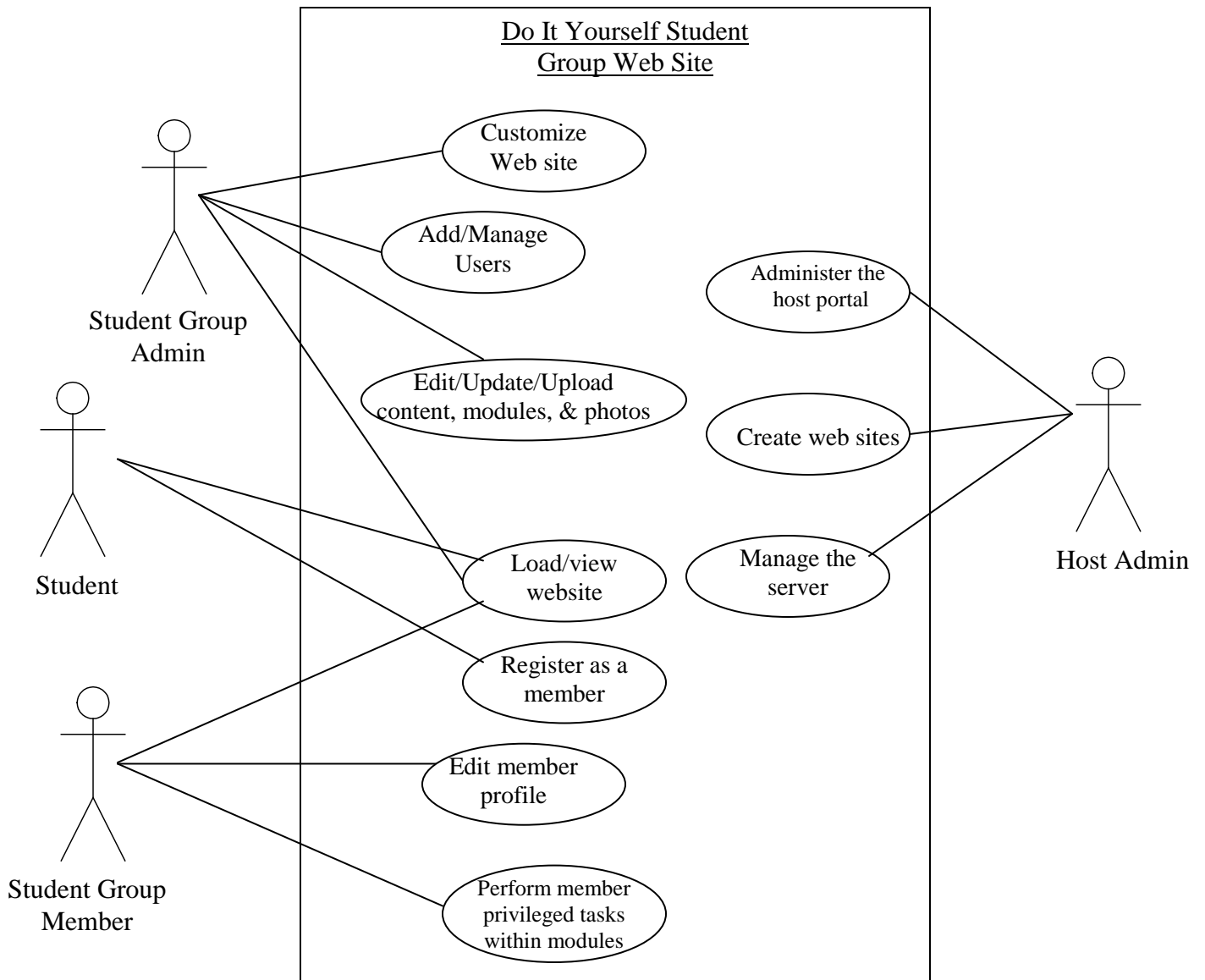


Figure 1. Use Case Diagram

2.1.1 Host Administrator

The Host Administrator will be the person designated by the Student Tribunal who will have access to the portal and will be able to change any of the site settings and host settings. The main role of the Host Administrator is to create Web sites for the individual student groups and maintain the server.

2.1.2 Student Group Administrators

The Student Group Administrator will be the person designated by each student group who will have full control of their site. The student group can choose to have more than one administrator. The Student Group Administrator will be able to customize their Student Group Web site by choosing a template, skin, and modules. The administrator will be able to add/remove/move modules to different places on their site's pages. Administrators will be able to perform all functions within a module. They will be able to add and manage users and set their permissions/roles.

2.1.3 Student Group Members

Members will be able to view their student group's Web site and also view certain modules that are not available to the public. They will also be able to edit their member profile. Depending on the role that a member is assigned to or the permissions set by the administrator he or she may also be able to perform tasks that non-members cannot. For example, post to a forum, add an event to the calendar or add pictures to the photo gallery.

2.1.4 Students

Students will be able to view all of the student groups' Web sites. If they are interested they can register and become a member of any particular site based on the group's criteria.

2.2 Design Protocol

2.2.1 DotNetNuke

As mentioned before this application uses DotNetNuke as its framework. "DotNetNuke is an open source framework ideal for creating and maintaining professional Web applications. It is a content management system ideal for creating and deploying projects such as commercial Web sites, corporate intranets and extranets, and online publishing portals" (5). The Home page of the

DotNetNuke installation is shown on the next page (See Figure 3.). The “Do It Yourself Student Group Web Site” application uses DotNetNuke as a portal to create multiple student group Web sites. It is a portal for Student Tribunal. From the Tribunal portal, many individual student group sites can be created, each having their own design and identity (See Figure 2.).

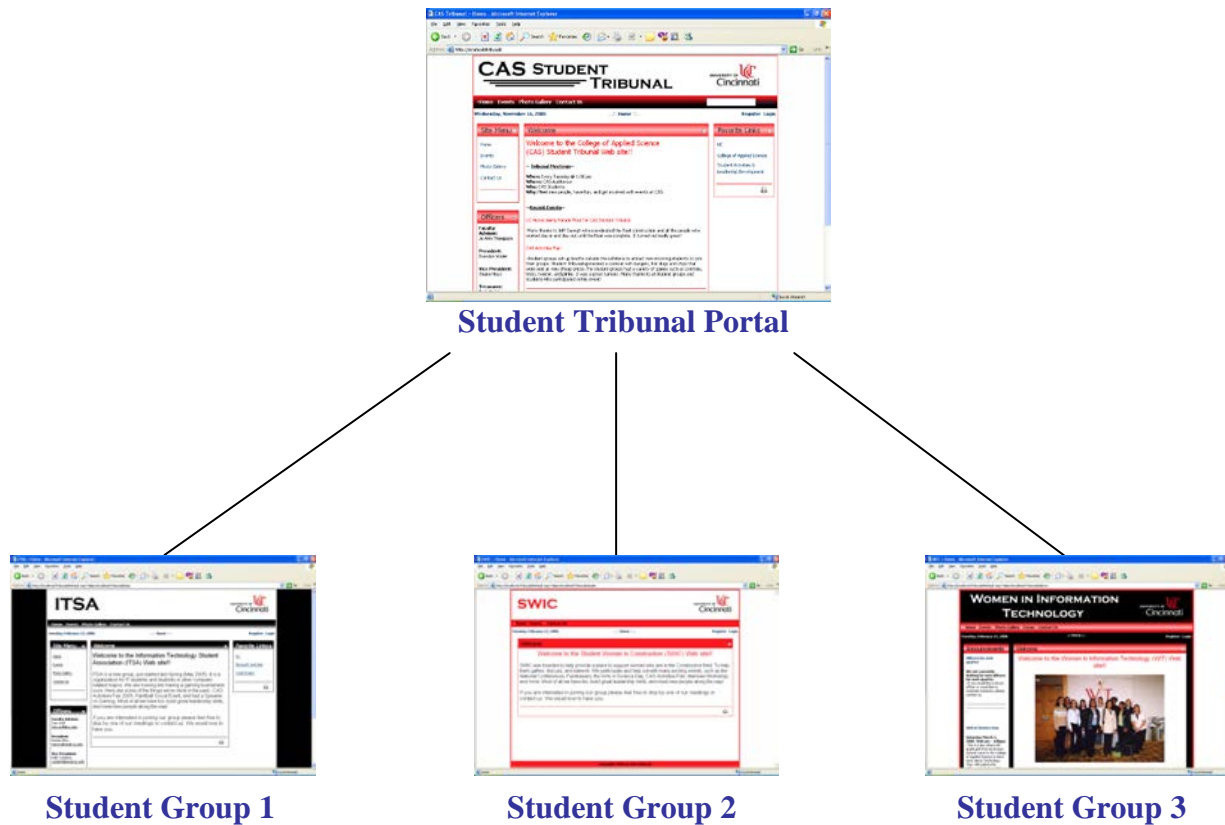


Figure 2. Application Hierarchy

Portal Technology by
DOTNETNUKE
<http://www.dotnetnuke.com>

Home Search

Wednesday, November 02, 2005 ...: Home ... Register Login

Links

- DotNetNuke@ Site
- Discussion Forum
- Defect/Feature Tracking
- Language Packs
- Modules

Welcome To My Website...

Administrator Login:

Username: admin
Password: admin

* Select the Admin tab to manage the portal settings as the Administrator. *Make sure you change the default password and specify a valid Email address for this account.*

Host Login:

Username: host
Password: host

* Select the Host tab to manage multiple portals as the Host. *Make sure you change the default password and specify a valid Email address for this account.*

Open Source License:

DotNetNuke@ - <http://www.dotnetnuke.com>
Copyright (c) 2002-2005
by Perpetual Motion Interactive Systems Inc. (<http://www.perpetualmotion.ca>)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Sponsors

Perpetual Motion

Microsoft ASP.net

MaximumASP
Serious Windows 2003 .NET Hosting

SOURCEWAY VAULT

telerik
deliver more than expected

Get this book from Amazon.com, written by members of the DotNetNuke@ Core Team. Shaun Walker, Patrick Santry, Joe Brinkman, Scott Willhite, Dan Caron, Scott McCulloch, Bruce Hopkins

Order your copy now at Amazon.com

Figure 3. DotNetNuke

2.2.2 Administrative Interface

DotNetNuke has a user-friendly administrative interface that is designed to make it easy for the student groups to manage all aspects of their Web sites. The interface shows the actual site with a control panel, outlines, and functionality for the administrator (See Figure 4.). It looks very similar to Figure 3, but Figure 4 points out the differences. The site wizard, help icons, and organization of the menus make the interface user-friendly (See Figure 4.).

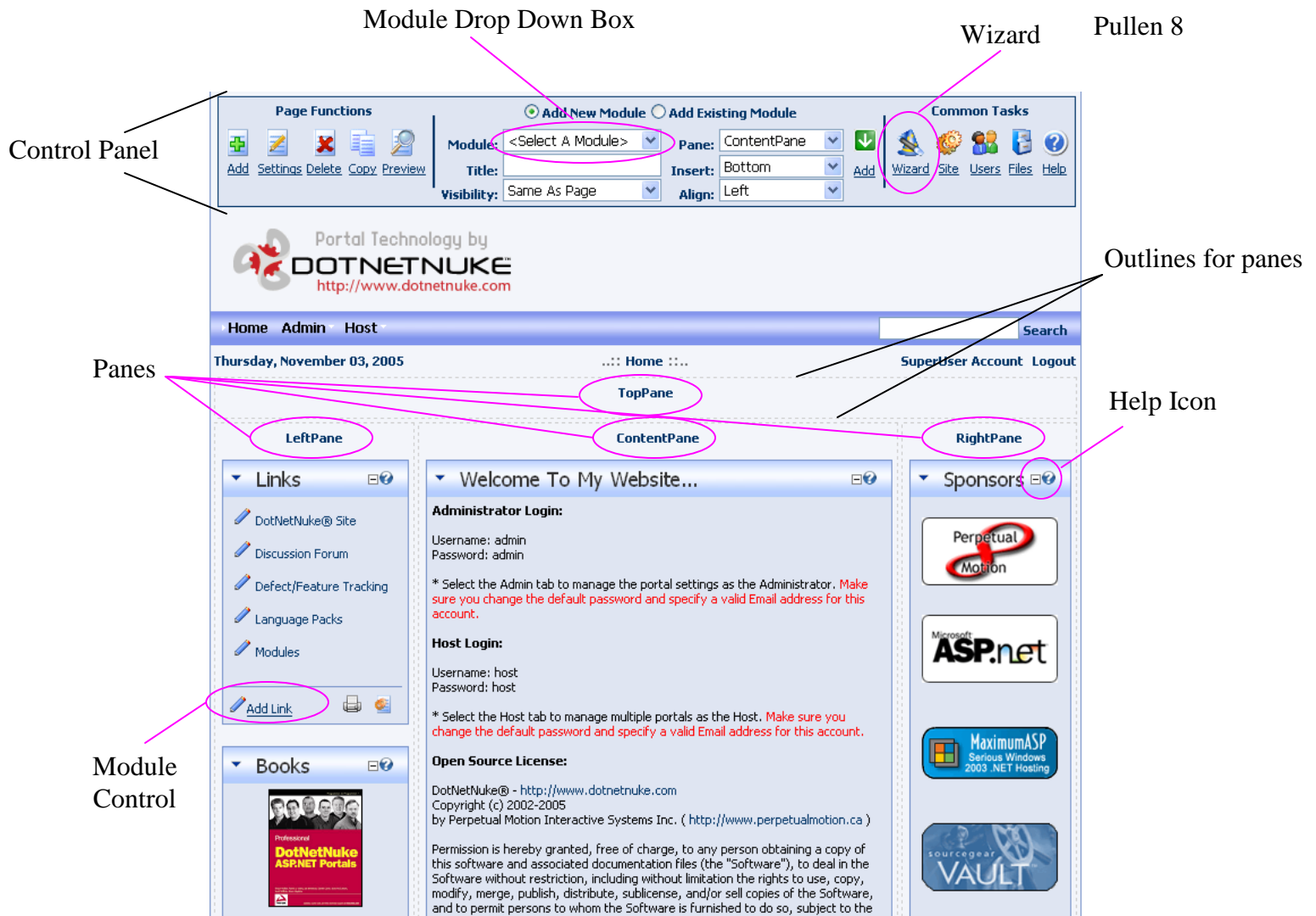


Figure 4. DotNetNuke Administrative Interface

The interface enables users to create and customize their own dynamic Web site. The student group administrators can add and remove pages by using the “Page Functions” section in the control panel, and can add a module by choosing one from the drop down list and placing it in the pane they want. They can move the module by clicking on the module and dragging it into the pane and location of their choice.

Content of any module can be changed by clicking the control in the lower left hand corner. Different modules have different controls. Some examples of these controls are “Edit Text” (Text/HTML module), “Add Link” (Links module), and “Add New Event” (Events module). The Text/HTML module allows editing of the content in a rich-text editor. It looks and

functions almost exactly like Microsoft Word. Figure 5 shows an example of editing the content of the “Welcome to My Website...” module (a Text/HTML module) that is in the content pane of Figure 4. You can also add pictures, hyperlinks (URLs), tables, dates, and times in a Text/HTML module (See Figure 5.).

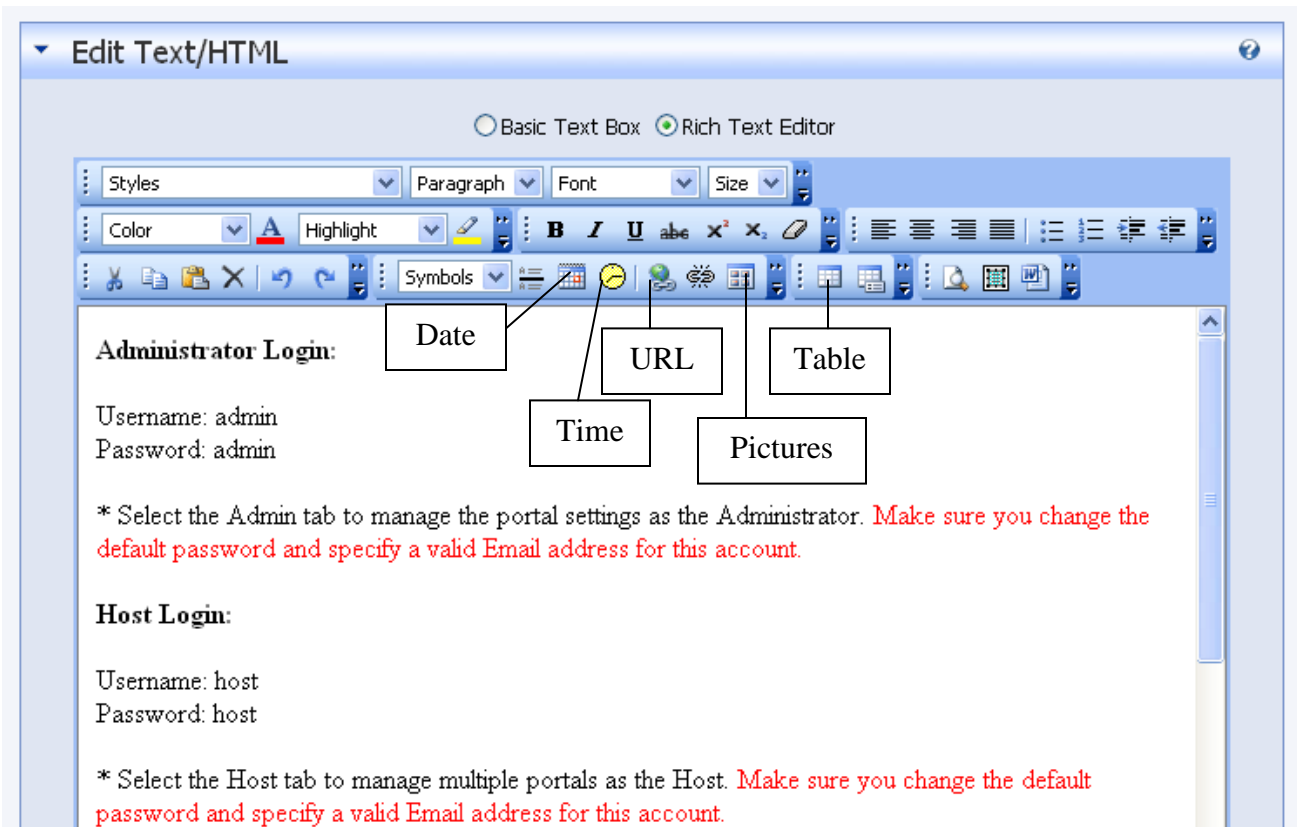


Figure 5. "Edit Text" Functionality of the Text/HTML Module

2.2.2.1 Administrative Options

DotNetNuke has divided the administrative options into two levels, host and administrator. The administrator has the ability to customize the Web site, change site settings, manage user accounts, create security roles and more. When logged on, the administrator will see an “Admin” tab which provides the Admin functions. The host administrator can perform all the same tasks that an administrator can and more. In addition to what the administrator can do,

the host administrator can create new Web sites, change host settings, manage host accounts, and more. When logged on, this person will see two tabs, “Admin” and “Host” (See Figure 4.).

2.2.3 Help Pages

An important feature that DotNetNuke provides is the Help pages. Help pages are located only on the Administrative side. In the control panel, on all of the modules, and on all form fields there is a help icon in the upper right hand corner or next to the object. Whenever an administrator needs help with something he or she can click on the icon. The help icon will then open up a help page displaying information about the object or giving instructions on how to complete a task. Figure 6 displays the module help for the Links module.

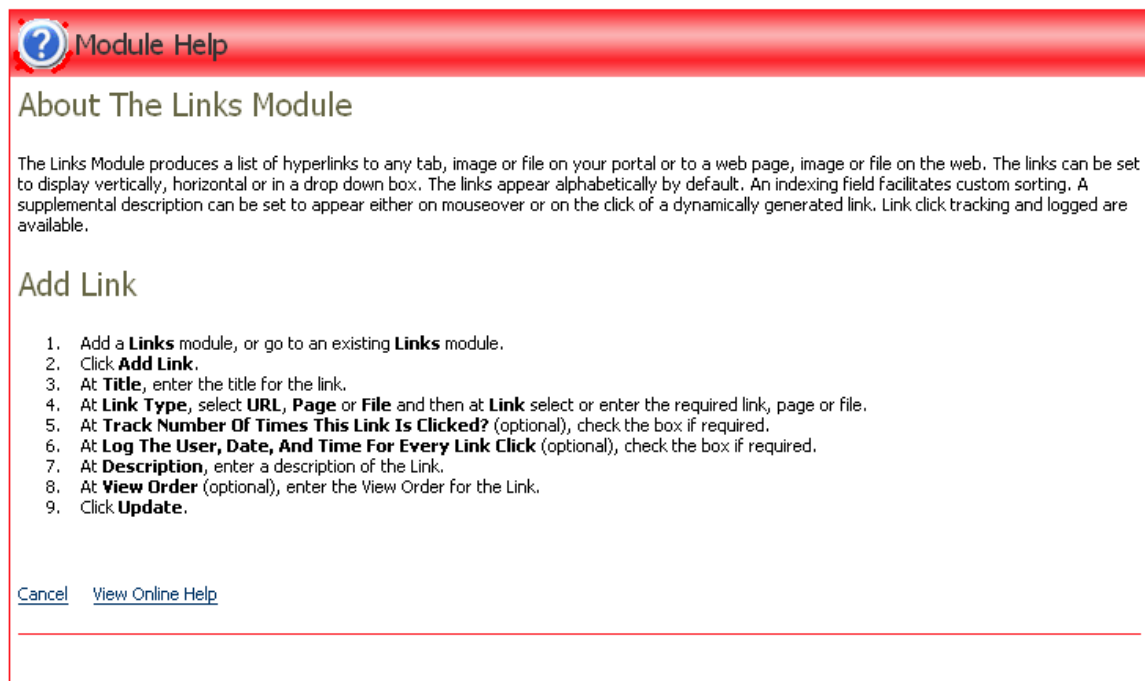


Figure 6. Module Help

2.2.4 Modules List

Table 1 is a list of all the modules included in DotNetNuke. The list defines which ones were kept, modified, added, and removed (See Table 1.).

All Modules	Modules Kept	Modules Modified	Modules Added	Modules Removed
Account Login	Account Login	Events	Photo Gallery	Banner
Announcements	Announcements	Feedback		IFrame
Banner	Documents	User Account		User Defined Table
Contacts	Discussions			Search Input
Documents	FAQs			Search Results
Discussions	Links			Image
Events	Survey			
FAQs	Text/HTML			
Feedback	Users Online			
IFrame	XML/XSL			
Image	Contacts			
Links	News Feeds (RSS)			
News Feeds (RSS)				
Search Input				
Search Results				
Survey				
Text/HTML				
User Account				
User Defined Table				
Users Online				
XML/XSL				

Table 1. List of Modules included in DotNetNuke and how it is modified

The Photo Gallery module was created and added to the application. The Events, Feedback, and User Account modules have been modified. The module modifications and details of the new module are shown below (See Table 2.).

Module Added	Modules Modified		
Photo Gallery	Events	Feedback	User Account
1) Create Albums	1) Calendar cells reformatted for a better visual appearance	1) Email and name fields were made static	1) Added new fields to registration form (Major and Graduation Year)
2) Edit and Remove Albums	2) Ability to click on the date of the event to see further details		2) Removed unnecessary fields (Fax and IM)
3) Upload photos to Albums	3) Only show the Title of the event on the events calendar		3) Ability to export user information to a spreadsheet file
4) Add/Edit descriptions of photos			
5) Remove photos			
6) Upload 3 photos at one time			
7) Navigation to view the next and previous photo within an album.			
8) Navigation to view the first and last photo within an album			

Table 2. Specific Module Modifications

2.2.5 UC Branding Standards

The “Do It Yourself Student Group Web Site” application adheres to the UC Branding Standards. The UC branding Web site (<http://www.uc.edu/branding/>) lays out the guidelines on what is allowed on UC Web sites and what is not. It details everything from color palettes to the

different forms of the UC logo to typefaces. UC has a very specific color palette that all UC sites must adhere to (See Figure 7.). The primary palette is red, black and white. These are the only colors that are allowed to be dominant on the sites. The primary palette and the remaining colors are what makes up the secondary palette. All of the colors on UC's color palette have been tested and are proved to be Web safe (13).



Figure 7. UC Color Palette

“The UC logo consists of two elements: the words "University of Cincinnati" and the UC symbol. The logo elements cannot be separated, altered, retyped or recreated in any way. The UC logo must appear in a prominent location on all materials published by the university, whether in print or in digital form” (13). It also has to have a certain amount of space surrounding it, approximately the width of the letter “C” in Cincinnati. The UC symbol can be used in addition to the UC logo but it too has to have a certain amount of space surrounding it, approximately half the height of the symbol. See Figure 8 below. The logo and symbol can only be changed to black, red, or white and appear on backgrounds that do not shadow the logo or symbol.



Figure 8. UC Logo and Symbol with the "safe area" marked in gray

Other elements such as the ingot and design architecture can be used when branding for UC. “The ingot is the small rectangle derived from the UC symbol. Its proportions are 1 unit

high by 3.5 units wide” (13). “The design architecture is a graphic derived from the shape and curves of the UC symbol” (13).

Those are some of the main branding concerns when creating a UC Web site. There are more minor details for UC branding that are not covered here but can be found in reference (13).

2.2.6 Organizational Scheme

When a student group wants to create a Web site they must go through the Host Administrator of the Tribunal portal. The Host Administrator will then create a site for that student group. The student group will need to assign a Student Group Administrator. The Student Group Administrator will customize the site to fit their group’s needs. Students will now be able to open the group’s site on the internet. From there, a student can register to become a member of that group. Once a person is a member, they will have extra access to certain parts of the group’s site and be able to modify their profile.

3. Deliverables

The deliverables of this project were determined during Senior Design II in the design phase. They are described in the following list:

1. A Web-based content management portal for CAS Student Tribunal in which multiple student group web sites can be created.
2. Change the default database name and default logins.
3. Create four new templates.
4. Create four new skins.
5. Modify the Events, Feedback and User Account modules and create a new PhotoGallery module (See Table 2.).
6. Student groups can customize their own individual sites by choosing from the templates, skins, and modules.

4. Proof of Design

4.1 Skins

The skins that DotNetNuke provides do not adhere to the UC Branding standards. They do not adhere because the dominant colors are not red, black, or white. The colors of their skins are blue, green, yellow, burgundy red, and gray. Therefore, new skins needed to be created.

The skins that were created for this application are the “All-Red”, “Red-Black-Gradient”, “Black-Gradient”, “Red-Gradient”, and “Red and Black” skins (See Figures 9, 10, 11, 12, and 13). These skins were created on the basis of one of the pre-existing skins. The code was analyzed and the cascading style sheets (CSS files) were carefully edited according to UC Branding standards. All of the skins have the University of Cincinnati logo in the top right hand corner. When the logo is clicked on, it sends the user to the UC home page, <http://www.uc.edu>, as necessary per UC branding. When the skins are viewed in a resolution of 800 x 600, the UC logo is still visible which is another requirement of UC branding. The logo that is at the top left of the page will not always be the same. When a student group site is created they will upload their logo and it will be placed in this position.

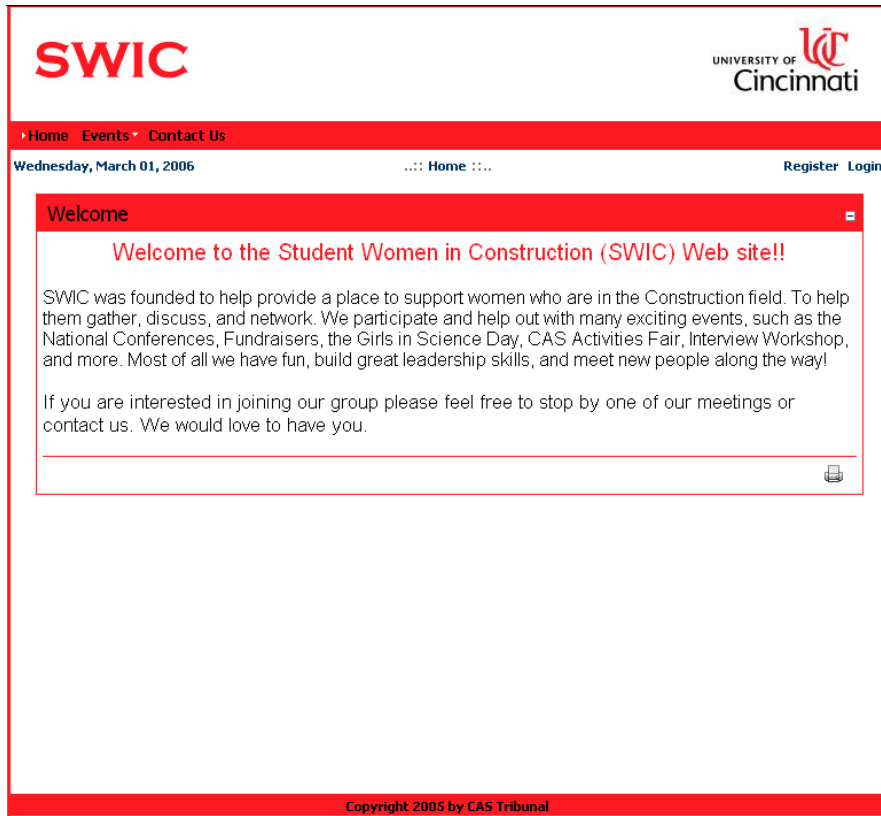


Figure 9. "All-Red" skin

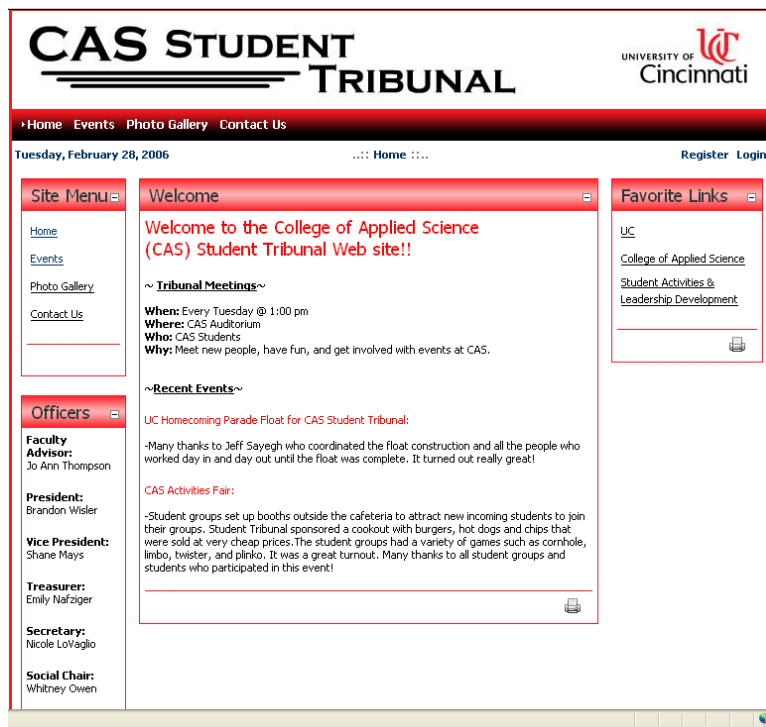


Figure 10. "Red-Black-Gradient" skin

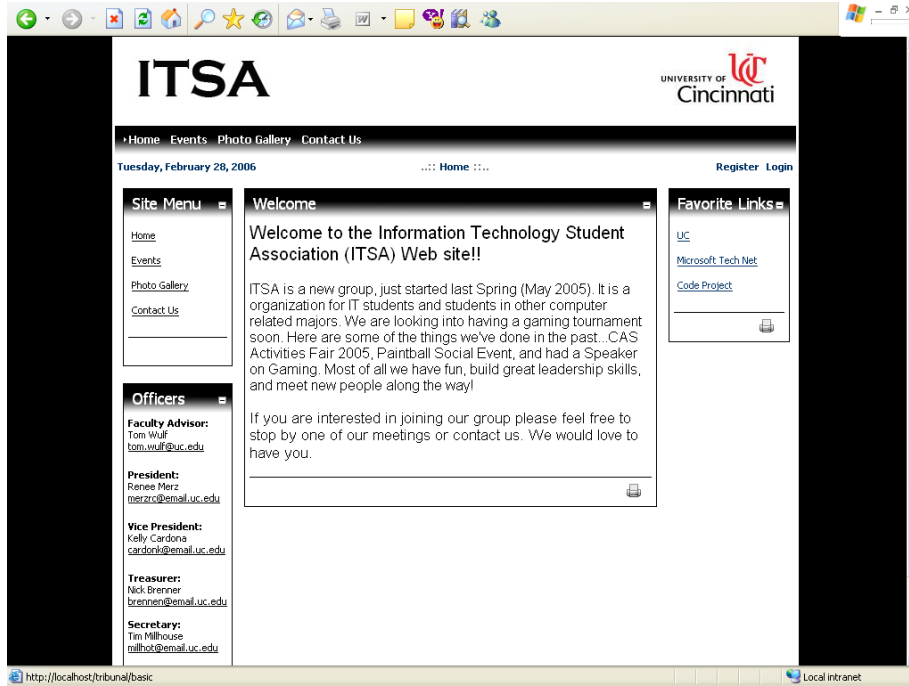


Figure 11. "Black-Gradient" skin



Figure 12. "Red-Gradient" skin



Figure 13. "Red and Black" skin

There are two style sheets that are associated with a particular skin, “skin.css” and “container.css”. “Skin.css” is the style sheet for the outer part of the page and “container.css” is the style sheet for the modules. Both of these files have been changed on each of the four skins. When a student group selects the skin for their site, they will need to choose the page skin (“skin.css”) and the module skin (“container.css”). The student group administrator can do this through the skins page under the admin tab (See Figure 14.).

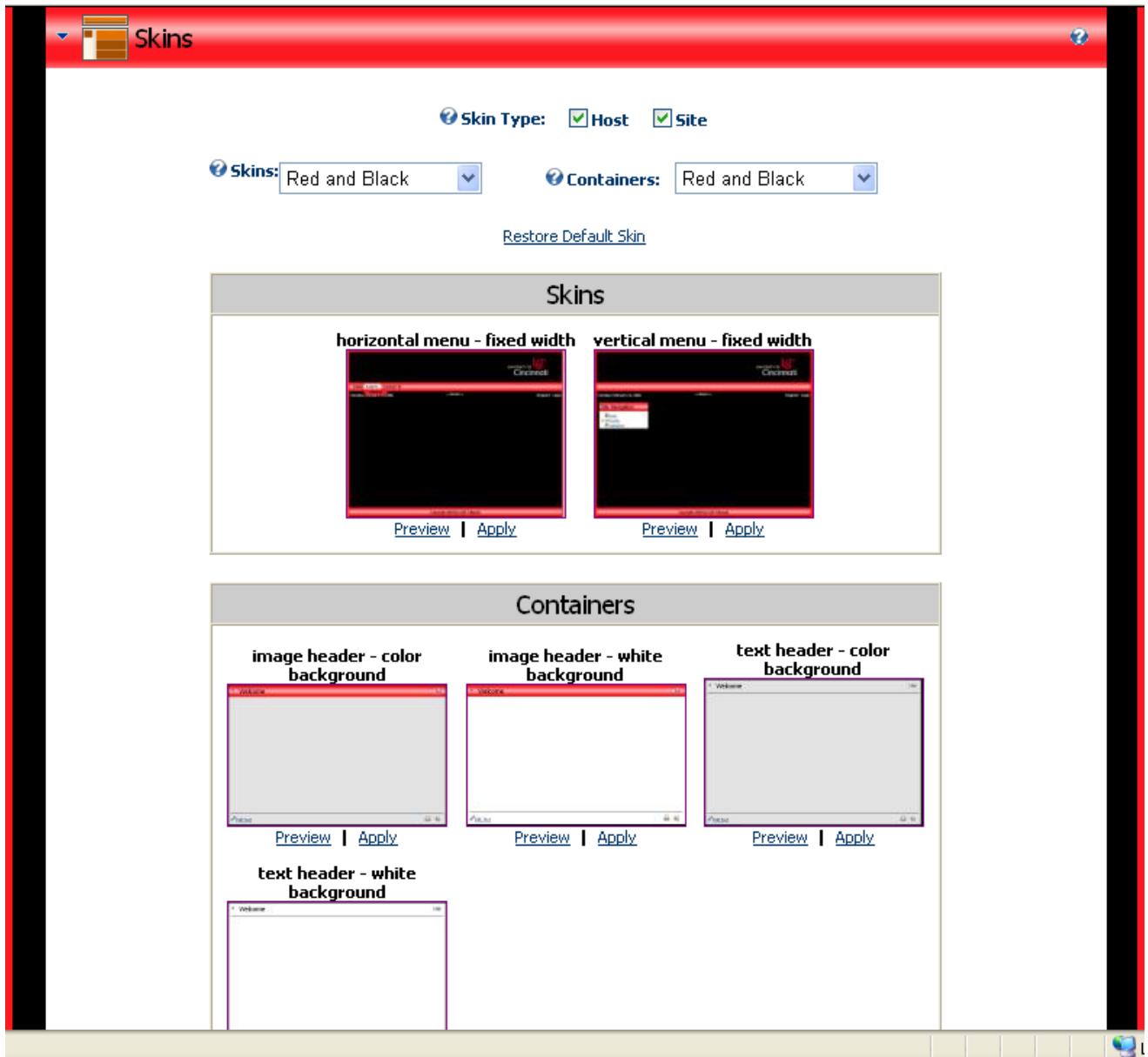


Figure 14. Choosing Skins

Each skin has the option for a horizontal menu or a vertical menu, both at fixed widths. The module skins each have at least two options, an image header with a color background which is gray or an image header with a white background. An image header means that it will use the particular gradient of that skin as the header for the module. The “All-Red” skin uses the color red instead of a gradient for the image header. The “All-Red”, “Red-Black-Gradient”, and the

“Red and Black” skin have two other options for the module skins also. These are, a text header with a color background which is gray, or a text header with a white background. Joining these different page skins and module skins gives us a total of 32 possible skin combinations.

In order for the skins to be used they must first be packaged and uploaded to the application. In order for the application to recognize a skin it must be packaged with a particular folder structure and placed in a zip file. Skins can be either uploaded to the Host which will make it available to the Host Administrator and the individual Student Group Administrators, or only uploaded to a particular student group’s site. For this project, the skins have been uploaded to the host, making them available to all group administrators.

Besides changing the style sheets of the skins, several gradients were created using Adobe Photoshop®. The “Red-Black-Gradient”, “Black-Gradient”, “Red-Gradient”, and “Red and Black” skins use these gradients (See Figures 10, 11, 12, and 13). The navigation bars, footer bars, and module title bars use the gradients. A banner for the Student Tribunal has been created and placed at the top of the page, in place of the CAS logo (See Figure 10.).

4.2 Templates

Templates are pre-defined layouts of the pages and modules. Basically, a template contains certain pages and modules based on the group’s need. Sites can be built using a template or from scratch (adding pages and modules one by one). DotNetNuke provides a few templates but they are not specific for a student groups therefore they have been removed. Four new templates have been created. They are called the “Simple”, “Basic”, “General”, and “Deluxe” templates. Every template has a Home page, Events page, Photo Gallery page, and Contact Us page. The Events page has the events module which includes a calendar. Events can be added to the calendar. The events module has been modified for this application. The Photo

Gallery page includes the newly created Photo Gallery module which allows for adding albums and photos (See section 4.3 on Modules for more details). Figures 17 and 18 show examples of these two pages. The modules on the Home and Contact Us pages vary depending on the template.


The “Simple” template provides all of the pages and modules necessary for the most simplistic student group Web site. It includes three main pages, Home, Events, and Contact Us and a sub page under the Events page called Photos (See Figure 15.). The Home page has a Text/HTML module in which student groups can write a welcome message to the visitors. The Photos sub page includes the Photo Gallery module. Lastly, the Contact Us page includes a Text/HTML module in which Contact information can be added.



Figure 15. Simple template pages and sub pages

The “Basic” template provides all of the pages and modules necessary for the most basic student group Web site. The pages it includes are Home, Events, Photo Gallery, and Contact Us. All pages, except the Events page, have a site map in the left pane. The Home page has two Text/HTML modules, Welcome and Officers, in which a welcome message and officer information can be inserted. There are two Link modules on this page as well, Site Menu and Favorite Links. These allow for adding hyperlinks. The last page in this template is the Contact Us page with a Text/HTML module to add the officers’ contact information. The following figures show each page of this template (See Figures 16-19).

CAS STUDENT TRIBUNAL



[Home](#) [Events](#) [Photo Gallery](#) [Contact Us](#)

Saturday, November 26, 2005
...: Home ...
[Register](#) [Login](#)

Site Menu

- [Home](#)
- [Events](#)
- [Photo Gallery](#)
- [Contact Us](#)

Welcome

Welcome to the College of Applied Science (CAS) Student Tribunal Web site!!

~ Tribunal Meetings ~

When: Every Tuesday @ 1:00 pm
Where: CAS Auditorium
Who: CAS Students
Why: Meet new people, have fun, and get involved with events at CAS.

~ Recent Events ~

UC Homecoming Parade Float for CAS Student Tribunal:

-Many thanks to Jeff Sayegh who coordinated the float construction and all the people who worked day in and day out until the float was complete. It turned out really great!

CAS Activities Fair:

-Student groups set up booths outside the cafeteria to attract new incoming students to join their groups. Student Tribunal sponsored a cookout with burgers, hot dogs and chips that were sold at very cheap prices. The student groups had a variety of games such as cornhole, limbo, twister, and plinko. It was a great turnout. Many thanks to all student groups and students who participated in this event!

Favorite Links

- [UC](#)
- [College of Applied Science](#)
- [Student Activities & Leadership Development](#)

Officers

Faculty Advisor:
Jo Ann Thompson

President:
Brandon Wisler

Vice President:
Shane Mays

Treasurer:
Emily Nafziger

Secretary:
Nicole LoVaglio

Social Chair:
Whitney Owen

Figure 16. Home page of the Basic template with sample content

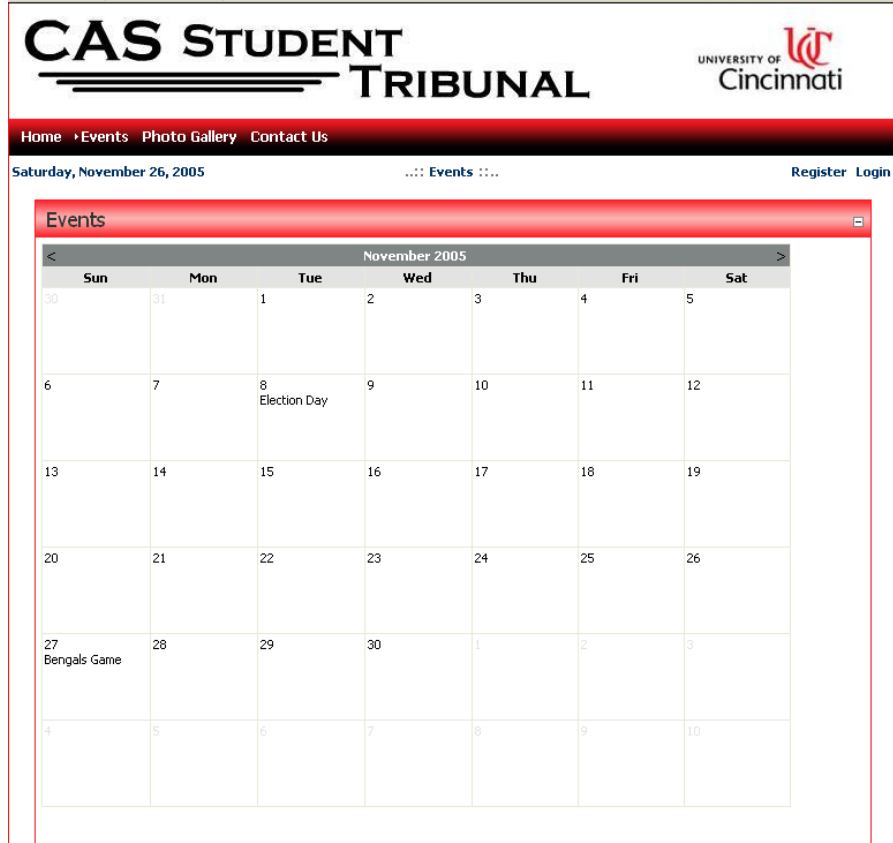


Figure 17. Events pages of the Basic template with sample events

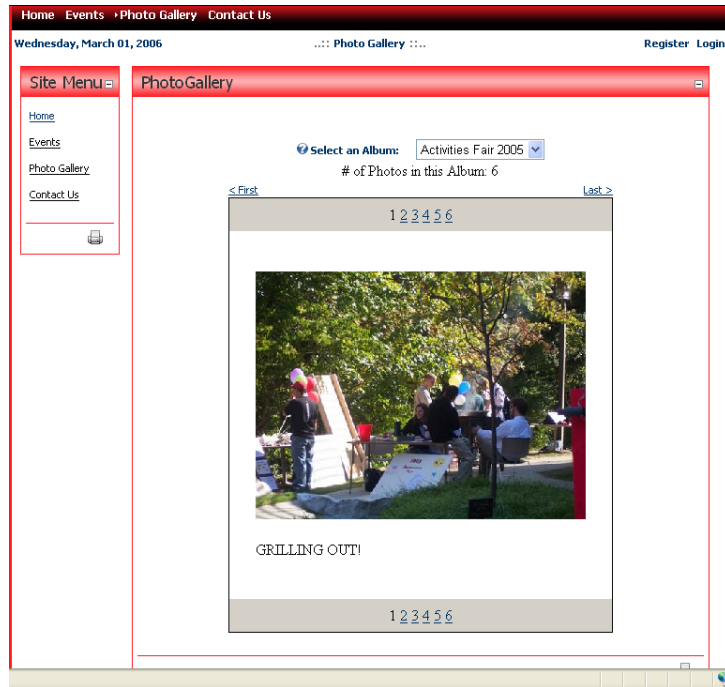


Figure 18. Photo Gallery page of the Basic template with sample albums and photos

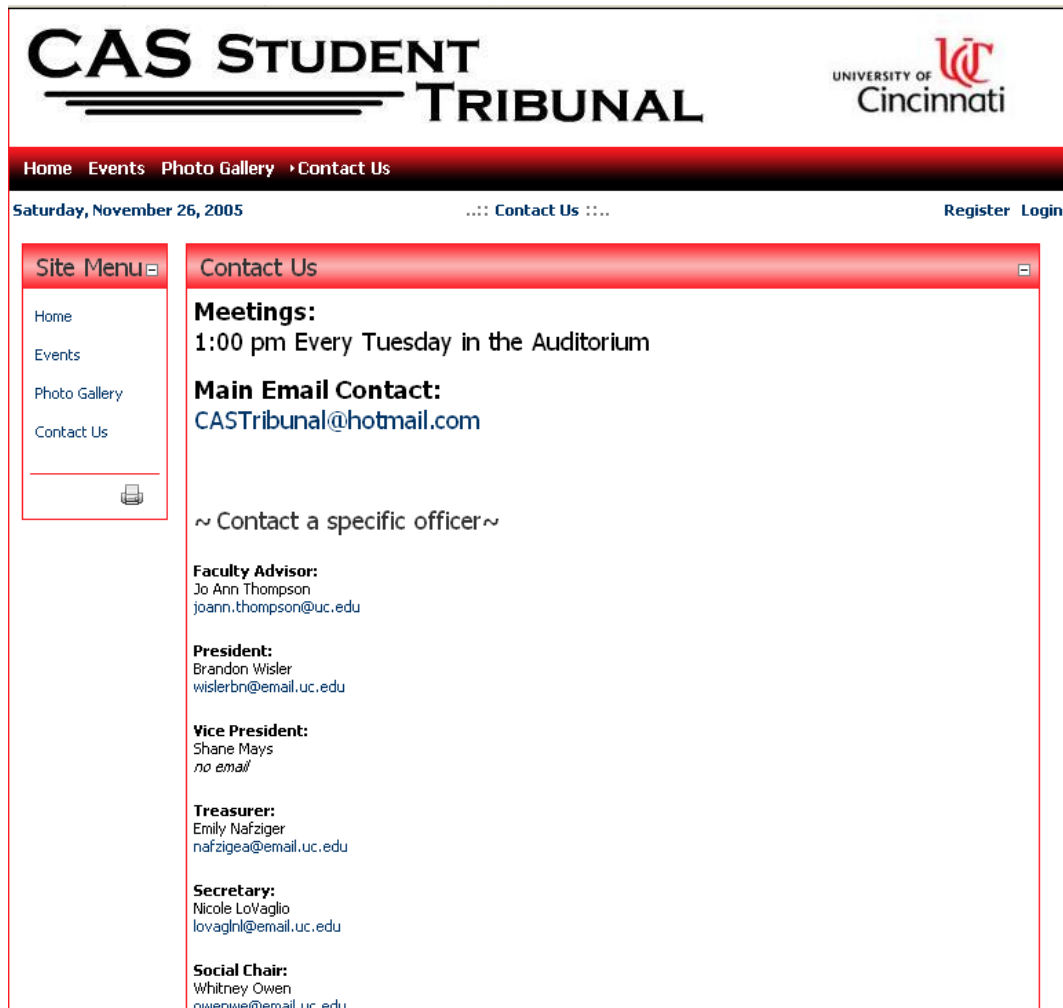


Figure 19. Contact Us page of the Basic template with sample content

The “General” template has four main pages, Home, Events, Messages, and Contact Us. It includes two sub pages as well, Photos and Forum. Photos is located under the Events page, and Forum is located under the Messages page (See Figure 20.).



Figure 20. General template pages and sub pages

The Home page in this template has a Text/HTML module for a welcome message. The Messages page has an Announcements module in which the student group administrator can post

announcements. The Forum page includes a Forum module which allows registered users to post discussions and replies. The last page in this template is the Contact Us page with two Text/HTML modules (Officers and Feedback Message) and a Feedback module. The Feedback module will only appear for registered users (See Figure 21.).

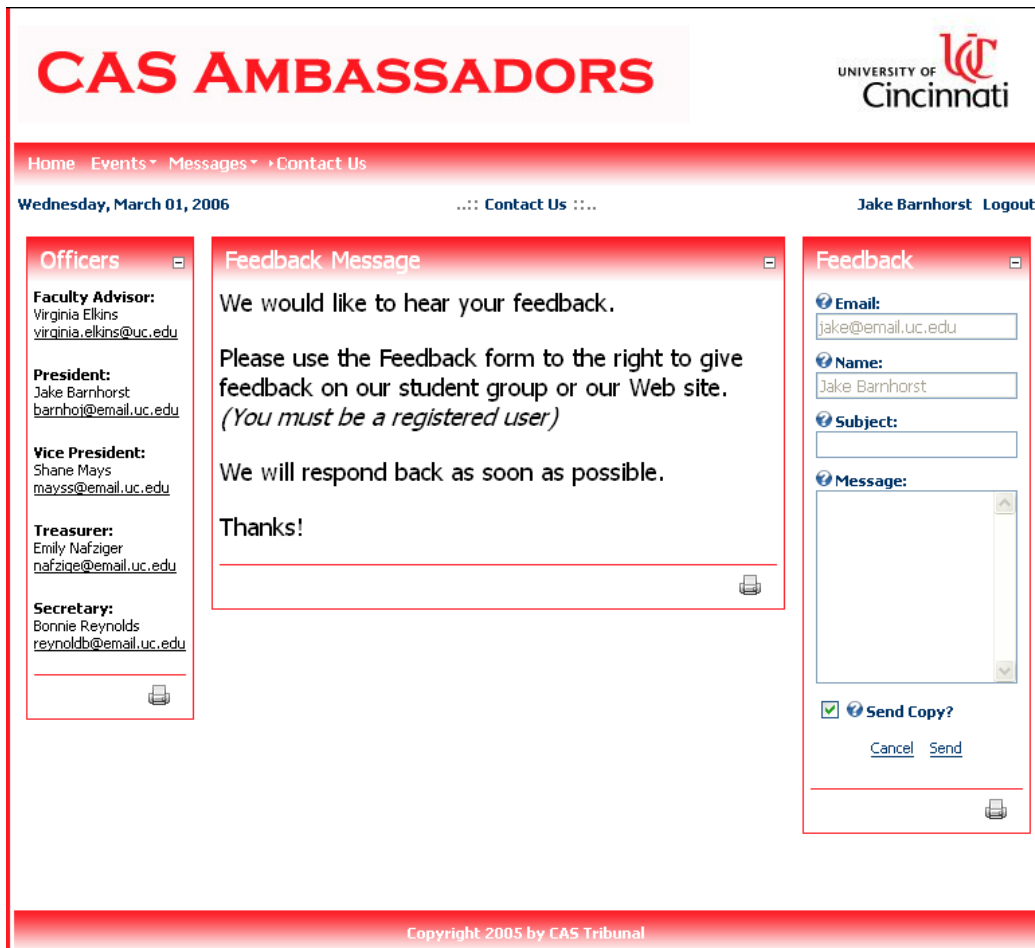


Figure 21. Contact Us page for the General template with sample content

The last template is the “Deluxe” template. It is the most complex of the templates. It includes six pages, Home, Events, Photo Gallery, Forum, Documents, and Contact Us (See Figure 22.).



Figure 22. Deluxe template pages

A security role was created for this particular template. It is the Officers role which is specifically for the Officers of the student group. Officers can perform other functions that regular users cannot. In this template users who belong to the Officers role can add announcements and events, edit the photo gallery, post to the forum, and exchange documents. The Documents page is only visible to the Officers. This page includes a documents module which allows Officers to upload and download documents (See Figure 23.).

WOMEN IN INFORMATION TECHNOLOGY

UNIVERSITY OF Cincinnati

Home Events Photo Gallery Forum Documents Contact Us

Wednesday, March 01, 2006 ... Documents ... Sarah Pullen Logout

Documents

Title	Owner	Category	Last Updated	Size (Kb)
By Laws	WIT Admin		2/18/2006	30.21 Download
Meeting Minutes 12-2-04	WIT Admin		2/18/2006	30.21 Download

[Add New Document](#)

Copyright 2005 by CAS Tribunal

Figure 23. Documents page of Deluxe template with sample documents when logged in as an Officer

On the Home page all users will see an Announcements module and a Text/HTML module for a welcome message. Registered users will also see a Feedback module with which they can send feedback to the student group administrator. The Officers will see an additional

Users Online module which gives them a brief summary of membership and people online (See Figure 24.).

WOMEN IN INFORMATION TECHNOLOGY

UNIVERSITY OF Cincinnati

Home Events Photo Gallery Forum Documents Contact Us

Wednesday, March 01, 2006 ...: Home ::... Sarah Pullen Logout

Users Online

Membership:
 Latest: **rachel**
 New Today: **0**
 New Yesterday: **0**
 Overall: **10**

People Online:
 Visitors: **0**
 Members: **0**
 Total: **0**

Online Now:

Announcements

Officers for next quarter

We are currently looking for new officers for next quarter.
 -If you would like to be an officer or would like to nominate someone, please contact us.

Girls in Science Day

Welcome

Welcome to the Women in Information Technology (WIT) Web site!!

WIT girls.jpg

WIT was founded in the Fall of 2002 to help provide a place to support women who are interested in the Information Technology / Computer Science (IT/CS) field. To help them gather, discuss, and network. We participate and help out with many exciting events, such as the WIT Conference, Ohio Celebration of Women in Computing Conference, Pizza Sales, Bake Sales, Girls in Science Day, CAS Activities Fair, Interview Workshop, and more.

Figure 24. Home page of the Deluxe template with the Users Online module and sample content

The Events, Photo Gallery, and Forum pages include their respective modules. Registered users have the ability to post discussions and replies to the forum.

Student groups can choose from these four templates. After choosing a template they have the option to add more pages and modules to the layout. If they do not want to use any of the templates they can add their own pages and modules to create a custom layout. With the combination of skins and templates there are 128 different possible Web sites.

4.3 Modules

“These are modular pieces of code which give some form of specific functionality to the user, and are grouped together in units to form complete web pages. Modules can be easily added, edited, deleted, restored, moved around on a page, or transferred to other pages” (5). DotNetNuke provides 21 modules in version 3.0.13 (5) (See Table 1.). Table 1 also displays which modules were kept, modified, removed and added. All of the modules that were modified and created were done in Microsoft Visual Studio 2003 using Active Server Pages.Net (ASP.Net) and Visual Basic.Net (VB.Net).

4.3.1 Events Module

The Events module was modified according to the following criteria:

- 1) Reformat the calendar cells to make it more visually appealing.
- 2) Provide the ability to click on a date of an event to see further details.
- 3) Only show the title of the event on the events calendar.

An event can be added when logged in as an administrator by clicking the “Add New Event” link at the bottom of the module. This opens an Edit Events page which allows the Event information to be entered (See Figure 25.). Figure 26 shows what the Events module looked like previously (See Figure 26.). The calendar cells change size depending on the length of the details. This is not visually appealing. The text within the cells is crowded which makes it hard

to read and follow. It also makes the cells of the calendar vary in size. This is intensified as new events are added.

Edit Events

Title:

Description: Basic Text Box Rich Text Editor

Styles Paragraph Font Size
 Color Highlight **B** *I* U abc x² x₂
 Symbols

Image: **File Location:** Root
File Name: <None Specified>
[Upload New File](#)

Alternate Text:

Occurs Every:

Start Date:

Time:

Expiry Date:

[Update](#) [Cancel](#)

Figure 25. Add/Edit an event

November 2005						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31	1	2	3 Volleyball game 8:00 PM UC Rec Center, be there 15 min early if you are playing.	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24 Thanksgiving UC Offices closed, no classes	25	26
27 Bengals Game 1:00 PM Come watch the Bengals vs. the Ravens on my big screen tv! The party will continue after the game. BYOB. Feel free to bring a couple friends, just let me know in advance. Game starts at 1:00 and party will continue into the evening hours.	28	29	30	1	2	3
4	5	6	7	8	9	10

Figure 26. Original Events module

The calendar cells have been reformatted to make the calendar and events more visually appealing. The cells in the modified module are more equally proportioned and only have the title of the event displayed on the calendar (See Figure 27.). Additional functionality was added to provide the ability to click on an event date and see the details on a new screen (See Figure 28.). To see a code snippet of this please see Appendix A. The Back to Calendar button has been added to allow a user to go back to the Calendar view.

November 2005						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31	1	2	3	4	5
6	7	8 Election Day	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27 Bengals Game	28	29	30	1	2	3
4	5	6	7	8	9	10

Figure 27. Modified Events module

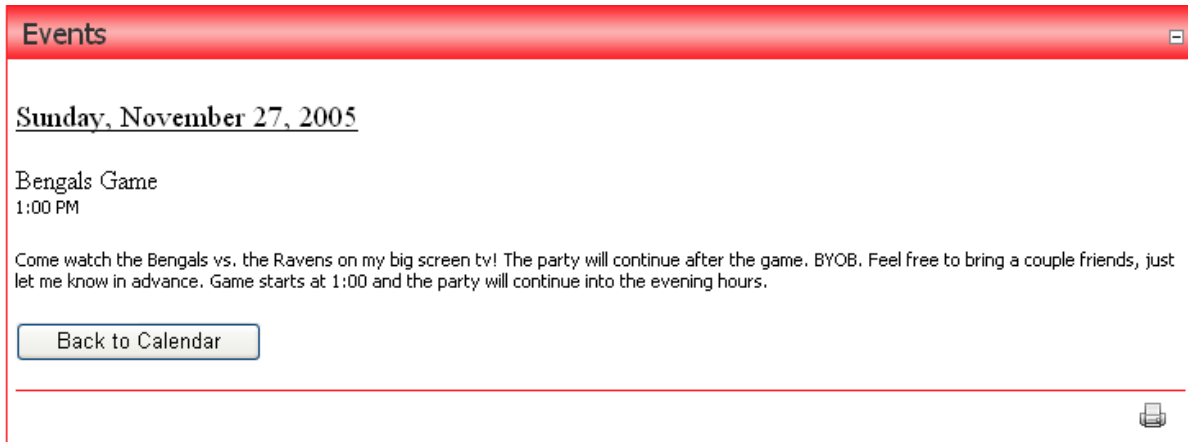


Figure 28. Sample details of an event

4.3.2 Feedback Module

Another module that was modified was the Feedback module (See Figure 29.). The email and name fields were made static. The text now appears gray. This ensures that the person sending the feedback cannot change their email address or name.

 A screenshot of a web browser window titled "Feedback". The window has a red header bar with the title "Feedback" and a close button. Below the header, there are four input fields: "Email:" with the value "pullensm@email.uc.edu", "Name:" with the value "Sarah Pullen", "Subject:" which is empty, and "Message:" which is a large text area. Below these fields is a checkbox labeled "Send Copy?" which is checked. At the bottom of the form are two buttons: "Cancel" and "Send". At the bottom right corner of the window, there is a small printer icon.

Figure 29. Modified Feedback module

4.3.3 User Account Module

The User Account module was modified according to the following criteria:

- 1) Added new fields to registration form (Major and Graduation Year)
- 2) Removed unnecessary fields (Fax and IM)
- 3) Ability to export user information to a spreadsheet file

The fields, Fax and IM (Instant Messenger) were removed from the module and the fields, Major and Graduation Year were added to make the form more relevant for student groups (See Figure 30.).

The screenshot shows a registration form titled "Register" with a red header. A note states: "*Note: Membership to this portal is Public. Once your account information has been submitted, you will be immediately granted access to the portal environment. All fields marked with an asterisk (*) are required." The form is divided into two columns of input fields. The left column includes: First Name, Last Name, User Name, Password, Confirm, Email Address, Major (circled in pink), Graduation Year (a dropdown menu with "2006" selected, also circled in pink), and Website. The right column includes: Street, Unit #, City, Country (dropdown menu with "United States" selected), Region (dropdown menu with "<Not Specified>" selected), Postal Code, Telephone, and Cell. Below the main form is a "Preferences" section with a collapse icon, containing: Preferred Language (dropdown menu with "English (United States)" selected) and Time Zone (dropdown menu with "(UTC -05:00) Eastern Time (US & Canada)" selected). At the bottom are "Register" and "Cancel" buttons.

Figure 30. Modified User Account module, registration page

The function to download registered users information to excel was added to this module. The button “Download Results as Excel” will download the results showing on the screen to an excel file (See Figure 31.). Please see Appendix A for a code snippet of this functionality. This allows

the administrator to save the members' contact information in an excel sheet on their computer for later use (See Figure 32.). They can also take the members' email addresses from the excel file and add them to an email distribution list or a List Serv.

User Accounts

Search: Username

Display Date: Last Login Created Date

Records Per Page: 10

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z All Unauthorized

Username	Name	Major	Year	Email	Telephone	Cell	Address	Last Login	Authorized
adam	Adam Sandler	Movie Production	2008	adam@email.com	9015647842	9015647777	1234 Hollywood Ave, Hollywood, California, United States, 91010	1/12/2006 4:01:55 AM	<input checked="" type="checkbox"/>
Adrienne	Adrienne Adams	Education	2009	Adrienne@email.com	5135551234		1234 Oak St, Cincinnati, Ohio, United States, 45255	1/9/2006 8:06:24 PM	<input checked="" type="checkbox"/>
cara	Cara Collins	IT	2006	cara@email.com	5135227575	9374425657	5252 Victory Pkwy, Apt 5, Ohio, United States, 45212	2/18/2006 3:19:19 AM	<input checked="" type="checkbox"/>
carol	Carol Pullen	Business	2007	caroloo77@hotmail.com	5134814444		5555 Montana Ave, Cincinnati, Ohio, United States, 45222	2/18/2006 3:07:18 AM	<input checked="" type="checkbox"/>
groupadmin	Administrator Account			sweeticera@hotmail.com				2/18/2006 8:20:00 PM	<input checked="" type="checkbox"/>
jessica	Jessica Young	Nursing	2009	jessicayoung@hotmail.com		5133825656	1111 River Rd, Fairfield, Ohio, United States, 45621	1/10/2006 11:59:35 PM	<input checked="" type="checkbox"/>
magic	Magic Johnson	Athletics	2006	magic@email.com	5135555555		street, LA, United States	1/12/2006 2:21:26 AM	<input checked="" type="checkbox"/>
margo	margo young	MET	2008	margo@email.com	5139390496	5139390496	Cincinnati, Ohio, United States	1/16/2006 1:59:54 PM	<input checked="" type="checkbox"/>
matt	Matt Bolte	Business	2010	matt@email.com	8595555555	8595120311	1234 Street, apt 5, Florence, Kentucky, United States, 41015	1/9/2006 7:11:26 PM	<input checked="" type="checkbox"/>
patty	Patty Morrison	AET	2011	patty@email.com			Cincinnati, Ohio, United States	1/16/2006 2:23:37 PM	<input checked="" type="checkbox"/>

Page 1 of 2 First Previous [1] 2 Next Last

[Delete Unauthorized Users](#) [Download Results as Excel](#)

Figure 31. User Account module showing "Download Results as Excel" link

	A	B	C	D	E	F	G	H	I	J	K	
1	Username	Full Name	Major	Grad. Year	Email	Telephone	Cell	Street	Add	City	State	Zip Code
2												
3	adam	Adam Sandler	Movie Production	2008	adam@email.com	9015647842	9015647777	1234 Holly		Hollywood	California	91010
4	Adrienne	Adrienne Adams	Education	2009	Adrienne@email.com	5135551234		1234 Oak		Cincinnati	Ohio	45255
5	cara	Cara Collins	IT	2006	cara@email.com	5135227575	9374425657	5252 Victory Pkwy		Alexandria	Ohio	45212
6	carol	Carol Pullen	Business	2007	caroloo77@hotmail.com	5134814444		5555 Mont		Cincinnati	Ohio	45222
7	groupadmin	Administrator Account			sweeticera@hotmail.com							
8	jessica	Jessica Young	Nursing	2009	jessicayoung@hotmail.com		5133825656	1111 River		Fairfield	Ohio	45621
9	magic	Magic Johnson	Athletics	2006	magic@email.com	5135555555		street		LA		
10	margo	margo young	MET	2008	margo@email.com	5139390496	5139390496			Cincinnati	Ohio	
11	matt	Matt Bolte	Business	2010	matt@email.com	8595555555	8595120311	1234 Street		Florence	Kentucky	41015
12	patty	Patty Morrison	AET	2011	patty@email.com					Cincinnati	Ohio	
13												

Figure 32. Sample downloaded excel file of members information

4.3.4 PhotoGallery Module

The PhotoGallery module is a new module that was created particularly for this application. The specific functionalities are listed below:

- 1) Create Albums
- 2) Edit and Remove Albums
- 3) Upload photos to Albums
- 4) Add/Edit descriptions of photos
- 5) Remove photos
- 6) Upload three photos at a time
- 7) Navigation to view the next and previous photo within an album
- 8) Navigation to view the first and last photo within an album

The Edit screen for the Photo Gallery has four links (See Figure 33.). “Add a New Album” gives the functionality to create a new album by typing in the name of the album (See Figure 34.). “Edit an Existing Album” allows for changing the name of an album or deleting an album (See Figure 35.). “Add New Photos to an Existing Album” is where new photos can be

uploaded to an existing album. A total of three photos can be uploaded at a time. A description for each photo can be uploaded at the same time (See Figure 36.). To see a code snippet of the upload functionality please see Appendix A.

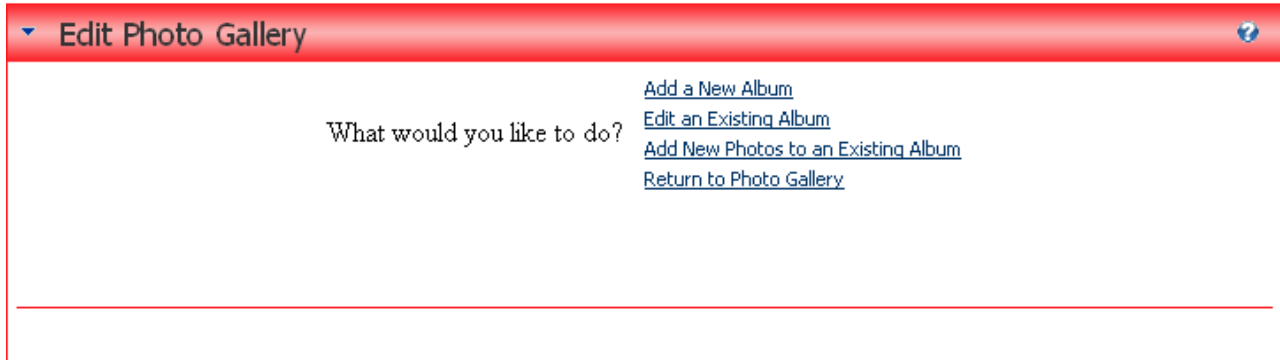


Figure 33. Edit Photo Gallery screen

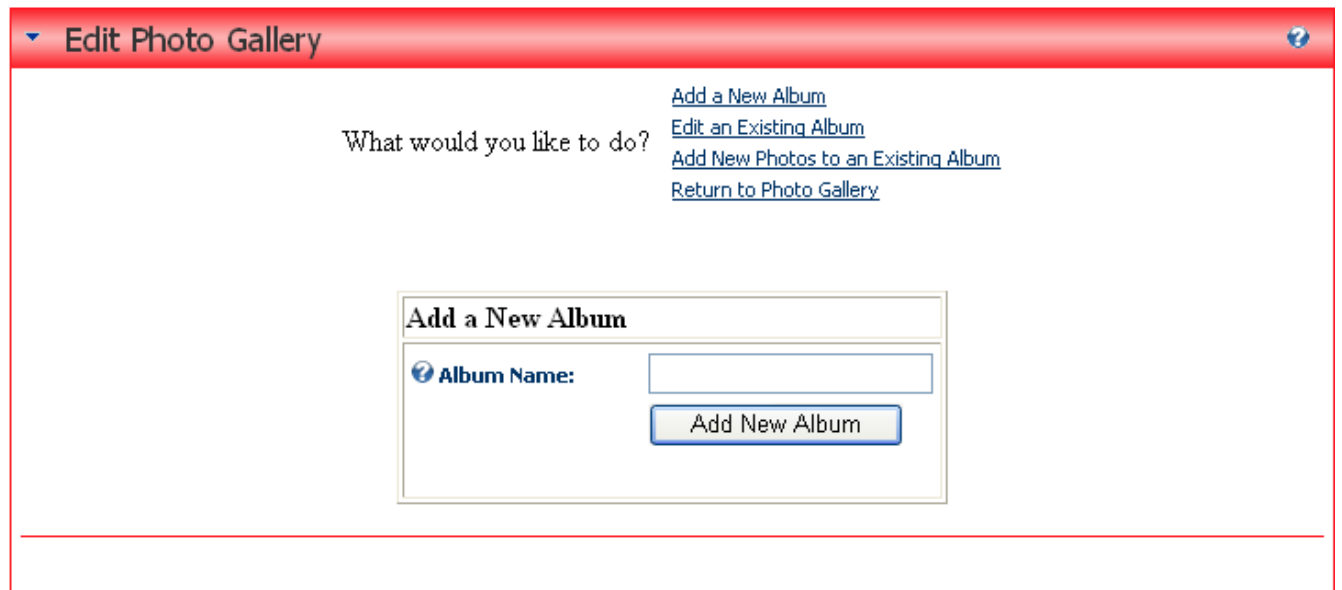


Figure 34. Add a New Album to Photo Gallery

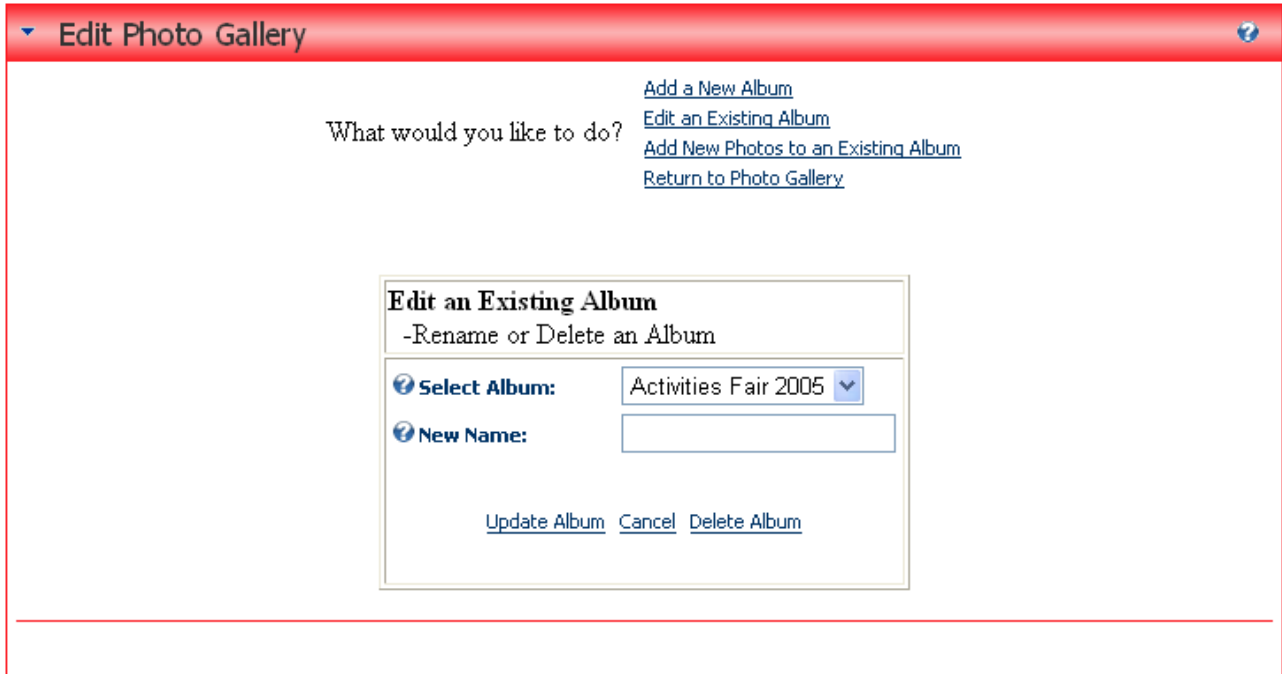


Figure 35. Edit an Existing Album

▼ Edit Photo Gallery
?

What would you like to do?

[Add a New Album](#)
[Edit an Existing Album](#)
[Add New Photos to an Existing Album](#)
[Return to Photo Gallery](#)

Add New Photos to an Existing Album

🔍 **Select Album:**

📷 **Upload Photo(s)** You may upload up to 3 photos at a time.

🔍 **Photo #1:**

📄 **Description #1:**

🔍 **Photo #2:**

📄 **Description #2:**

🔍 **Photo #3:**

📄 **Description #3:**

Figure 36. Add New Photos to an Existing Album

Once albums and photos have been added to the Photo Gallery, the photos within the albums can be viewed. A user can select the album they wish to see from the drop down box and the first photo within that album will appear (See Figure 37.). It will also tell how many photos are in the album. The photos can be viewed singularly by clicking on the next number. The user can go to the first photo or last photo in the album by clicking on the “< First” and “Last >” links.



Figure 37. Viewing the Photo Gallery

Photos can also be changed or deleted by clicking on the Edit pencil icon next to the photo. If a description needs to be added to a photo or changed it can be done in the edit screen (See Figure 38.).

▼ Edit Photo Gallery



Description: GRILLING OUT!

[Update Photo](#) [Cancel](#) [Delete Photo](#)

The image shows a screenshot of a photo gallery editing interface. At the top, there is a red header bar with the text '▼ Edit Photo Gallery'. Below this is a large white area containing a photograph of an outdoor event. The photo shows several people gathered around tables, with colorful balloons (red, white, blue, yellow) and a sign that says 'GRILLING OUT!'. Below the photo, there is a 'Description:' label followed by a text input field containing the text 'GRILLING OUT!'. Underneath the input field are three links: 'Update Photo', 'Cancel', and 'Delete Photo'.

Figure 38. Editing a Photo

5. Design and Development

5.1 Timeline

Table 3 displays the timeline of this project for Senior Design I, II and III.

Week of	Milestone/Task
Senior Design I	
4/4/2005-4/18/2005	Search for a project
4/18/2005	Select a project
4/25/2005 -5/9/2005	Research chosen project
5/9/2005	Prepare Budget
5/16/2005-5/30/2005	Prepare Proposal and Presentation
Senior Design II	
10/3/2005	Get development environment setup, Explore/Research DotNetNuke
10/10/2005	Explore DotNetNuke modules to find out which need to be modified
10/17/2005	Change database name and default logins, Remove modules and skins that are not necessary for student groups
10/24/2005	Create 1 skin design for the pages and modules according to UC branding standards
10/31/2005	Create 1 template for the site specifically for CAS Student Groups.
11/7/2005-11/21/2005	Modify the events module
11/21/2005	Finish Design Freeze Document
11/28/2005	Testing and fixing any errors/bugs. Preparing for Senior Design III
Senior Design III	
1/3/2006	Modify the User Account module with the items specified in Table 2
1/9/2006	Finish modifying the User Account module, Modify the Feedback module with the items specified in Table 2
1/16/2006-2/6/2006	Create the new Photo Gallery module with the items specified in Table 2
2/6/2006 - 2/13/2006	Create 3 new templates, Create 3 new skins
2/20/2006	Testing
2/27/2006	Development of Project Complete

Table 3. Project Timeline

5.2 Budget

The total retail cost of this project is \$3,407.90. Since the developer user her home computer as the Web server, and all the software and licensing agreements with UC are provided by the IT computer lab, there was only a \$70.00 cost for this project in total. This was for the 1 GB flash drive that was purchased. The developer decided to purchase a flash drive to back up

her project since flash drives are known to be much safer than zip disks. See Table 4 for complete budget details.

Item	Description	Retail Cost	Cost Incurred
Web Server	Used own computer	\$760.00	\$0.00
1 GB Flash Drive	Purchased	\$70.00	\$70.00
Windows XP Professional	Provided by the IT computer lab	\$299.00	0.00
Internet Information Services (IIS)	Included in Windows Components	0.00	0.00
Microsoft Visual Studio .NET 2003	Provided by the IT computer lab	489.95	0.00
DotNetNuke 3.0.13	“Provided free, as open-source software, and licensed under a BSD-style agreement.”	0.00	0.00
Adobe Photoshop 6.0	Provided by the IT computer lab	599.00	0.00
Microsoft SQL Server	Provided by the IT computer lab	1,189.95	0.00
	Retail Total:	\$3,407.90	
	Actual Total:		\$70.00

Table 4. Project Budget - Sources (7, 10, 11)

5.3 Hardware

- **Web Server**

The developer’s home computer was used as the Web server for this project. It had all the programs installed and hardware needed except Microsoft SQL Server. SQL Server was installed to provide the database for the project.

- **1 GB Flash Drive**

A 1 GB Flash Drive was used to backup all data for the project.

5.4 Software

This project used software that was provided by the IT computer lab.

- **Windows XP Professional**

One of the operating systems required to run DotNetNuke 3.0.13 and to run IIS.

- **Internet Information Services (IIS)**
IIS 5.0 or higher is required in order to run DotNetNuke 3.0.13 and it is also required for creating ASP.NET Web pages.
- **Microsoft Visual Studio .NET 2003**
“Visual Studio .NET 2003 enables programmers to rapidly build applications for Windows, the Web, mobile Web devices, the Pocket PC, and other embedded devices powered by the .NET Compact Framework”(9). This environment was necessary to have in order to program the project in VB.NET
- **DotNetNuke 3.0.13**
MS Visual Studio .NET 2003 and DotNetNuke were used in conjunction to code the “Do It Yourself Student Group Web Site” application.
- **Adobe Photoshop®**
Photoshop was used to create the designs/images necessary for this project. It was also used to edit the images.
- **Microsoft SQL Server**
SQL Server was used to create the database, table, relations, and stored procedures that this project needs. All the data in this project is stored in a SQL database.

5.5 Testing

This application was tested throughout the entire project as changes were made. The skins were tested individually to ensure all the colors on the skin were according to UC’s color palette and that the UC logo was on the page and still visible at a resolution of 800 x 600. The templates were tested to ensure the modules and pages were in the desired location and function properly. The new modifications to the Events, User Account, and Feedback modules were tested to make sure they function properly. The Photo Gallery module was also thoroughly tested to make sure each of the functionalities worked properly. Creating multiple sites has also been tested. Each of these modules have been added to a new site and tested to make sure nothing overlaps from any other site. The table on the next page shows each task that was tested (See Table 5.). Some of these items passed the first time, and some did not. The errors and issues that were found in the application have been fixed and tested again. All have now passed testing.

	Task	Pass/Fail
1	Create a site with the Simple template and All-Red skin	Pass
2	Create a site with the Basic template and Black-Gradient skin	Pass
3	Create a site with the General template and Red-Gradient skin	Pass
4	Create a site with the Deluxe template and Red and Black skin	Pass
5	Test each of pages and modules of the templates	Pass
6	Make sure each of the skins are displaying as expected	Pass
7	Register a new user and make sure that Major and Graduation fields are on the form and Fax and IM are not	Pass
8	Register a new user on each of the sites and make sure they have only privileges to areas they are supposed to	Pass
9	Log in each of the administrators accounts that were created when the site was created and make sure that they only have access to administration on the site to wish they were assigned	Pass
10	Register an Officer to the site with the Deluxe template, log on as the administrator, locate the user just created and place the new user in the Officer Role. Log back into the site as the Officer and make sure they can post announcements, add events, photos, post to the forum, and be able to exchange documents	Pass
11	Add an event to the Events module with details	Pass
12	Click on the event that was added and make sure the details appear on another page	Pass
13	Test the Export to Excel feature on the User Account module	Pass
14	Make sure the name and email field of the Feedback module are static	Pass
15	Test adding albums to the Photo Gallery module on a couple of the sites	Pass
16	Test adding an album without typing a name for the album, an error message will appear	Pass
17	Test adding 0 photos to an album to make sure an error message appears	Pass
18	Test adding only 1 photo to an album	Pass
19	Test adding only 2 photos at one time to another album	Pass
20	Test adding 3 photos to an album	Pass
21	Test changing the album name and make sure the photos stay with that album	Pass
22	Test changing the album name to no name, an error message appears	Pass
23	Test deleting an album and make sure the album and photos are deleted from the database and all the files of the photos are deleted as well	Pass
24	Test adding a description to a photo	Pass
25	Test deleting one photo from an album, make sure it is deleted from the database and the file is deleted also	Pass
26	Switch between sites and make sure the photo gallery of one site didn't replace the photo gallery of another	Pass

Table 5. Testing Plan

6. Conclusions and Recommendations

The “Do It Yourself Student Group Web Site” application makes it easy for student groups to get their Web sites up and running. It gives student groups the choice between multiple templates and skins to make their sites unique, while at the same time adhering to the UC branding standards. They can choose the modules they want on their site as well. This application is great for student groups to be able to create and manage dynamic Web sites.

A few challenges were faced while working on this project. Getting to know and understand someone else’s code is always a challenge, especially an application as large as DotNetNuke. Therefore, if someone wants to add functionality to this application it is recommended to review the DotNetNuke code and understand it before starting to program. Some helpful training videos can be found on DotNetNuke’s Web site at www.dotnetnuke.com. It is highly suggested that the developer watch these videos first. Always backup the application before making any changes. Also, if the developer wants to take advantage of the latest features and get the most recent bug fixes, he/she should upgrade to the latest version of DotNetNuke. Once the application is upgraded, the skins that are provided by DotNetNuke will need to be deleted since they are not UC branded. New UC branded skins, templates, and modules could be added to the “Do It Yourself Student Group Web Site” application to give the groups more choices on design, layout, and functionality of their sites.

Appendix A. Code Snippets

A 1. Events Module Code Snippet for Showing Details of an Event

```
Public Sub calEvents_SelectionChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles calEvents.SelectionChanged
    If Not arrDetails(calEvents.SelectedDate.Day) = "" Then
        calEvents.Visible = False
        lblEventDate.Visible = True
        lblEventDate.Text = "<br>"
        lblEventDate.Text += FormatDateTime(calEvents.SelectedDate)
        lblEventDetails.Visible = True
        lblEventDetails.Text = arrDetails(calEvents.SelectedDate.Day)
        btnBacktoCalendar.Visible = True
    End If
End Sub

Private Sub btnBacktoCalendar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBacktoCalendar.Click
    'Return to Calendar
    calEvents.Visible = True
    lblEventDetails.Visible = False
    lblEventDate.Visible = False
    btnBacktoCalendar.Visible = False
End Sub
```

A 2. User Account Module Code Snippet for Export to Excel

```
Private Sub btnExportExcel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnExportExcel.Click
    BindData("", "username")

    Response.Clear()
    Response.AddHeader("content-disposition",
"attachment;filename=Userdata.csv")
    Response.Charset = ""
    Response.Cache.SetCacheability(HttpCacheability.NoCache)
    Response.ContentType = "application/vnd.xls"
    Response.Write(strUserExport.ToString())
    Response.End()
End Sub
```

A 3. Photo Gallery Upload Files Code Snippet

```

Private Sub UploadFile(ByVal strFileLocation As String, ByVal intFileNum As
Integer)
    Try
        Dim strFileName As String =
System.IO.Path.GetFileName(strFileLocation)
        Dim exe As String = System.IO.Path.GetExtension(strFileName)

        If strFileLocation <> "" And (exe.ToLower = ".gif" Or exe.ToLower
= ".jpg" Or exe.ToLower = ".jpeg") Then

            Dim strCmd As String
            Dim strUploadedFileName As String

            'Take care of duplicate file names.
            Dim file_append As Integer = 0

            While System.IO.File.Exists(Request.PhysicalApplicationPath
& "DesktopModules\PhotoGallery\Images\" & strFileName)

                file_append = file_append + 1
                strFileName =
System.IO.Path.GetFileNameWithoutExtension(strFileName) &
file_append.ToString() + exe
            End While

            strUploadedFileName = strFileName

            'If Photo #1
            If intFileNum = 1 Then
                'Test File Size. Can't be greater than 5 MB
                If ffUploadFile.PostedFile.ContentLength <= 5485760 Then

                    'Save to the Server
                    ffUploadFile.PostedFile.SaveAs(Request.PhysicalApplic
ationPath & "DesktopModules\PhotoGallery\Images\" &
strUploadedFileName)

                    'Update Database
                    UpdatedB(strUploadedFileName, txtDescription1.Text)

                Else
                    lblErrorOnUpload.Text = "Sorry, Image #" & intFileNum
& " is too large. It cannot be larger than 5 MB"
                    lblErrorOnUpload.Visible = True
                End If
            End If

            'If Photo #2
            If intFileNum = 2 Then
                'Test File Size. Can't be greater than 5 MB
                If ffUploadFile2.PostedFile.ContentLength <= 5485760 Then
                    'Save to the Server

```

```

ffUploadFile2.PostedFile.SaveAs(Request.PhysicalApplic
ationPath & "DesktopModules\PhotoGallery\Images\" &
strUploadedFileName)

'Update Database
UpdateDB(strUploadedFileName, txtDescription2.Text)

Else
    lblErrorOnUpload.Text = "Sorry, Image #" & intFileNum
& " is too large. It cannot be larger than 5 MB"
    lblErrorOnUpload.Visible = True
End If
End If

'If Photo #3
If intFileNum = 3 Then
    'Test File Size. Can't be greater than 5 MB
    If ffUploadFile3.PostedFile.ContentLength <= 5485760 Then
        'Save to the Server

        ffUploadFile3.PostedFile.SaveAs(Request.PhysicalApplic
ationPath & "DesktopModules\PhotoGallery\Images\" &
strUploadedFileName)

        'Update Database
        UpdateDB(strUploadedFileName, txtDescription3.Text)

    Else
        lblErrorOnUpload.Text = "Sorry, Image #" & intFileNum
& " is too large. It cannot be larger than 5 MB"
        lblErrorOnUpload.Visible = True
    End If
End If
Else
    lblErrorOnUpload.Text = "Sorry, there was a problem with Image
#" & intFileNum & ". It must be an image file of JPEG (.jpg, .jpeg) or GIF
(.gif) format"
    lblErrorOnUpload.Visible = True

End If
Catch exc As Exception
    ProcessModuleLoadException(Me, exc)
End Try
End Sub

```

References

1. Adobe Store. 2005. Adobe Systems Incorporated. 28 May 2005. <http://store.adobe.com>.
2. "Annual Report". Sigma Lambda Chi. University of Cincinnati. 16 May 2005.
<http://www.uc.edu/cas/cs/sigma/pages/annualreport.htm>.
3. Associated Builders and Contractors Student Chapter. University of Cincinnati. 16 May 2005. <http://www.uc.edu/cas/cs/ABC/home.htm>.
4. Davis, Todd. Seabury Design. Creating a Module in DotNetNuke. Online training videos.
<http://dotnetnuke.com/About/TrainingVideos/tabid/810/Default.aspx>.
5. DotNetNuke. 2002-2005. 2 Nov 2005. <http://dotnetnuke.com/>.
6. Mays, Shane. Vice President of CAS Student Tribunal. Personal interview. 10 May 2005.
7. Microsoft. 2005. Microsoft Corporation. 27 May 2005.
<http://www.microsoft.com/windowsserver2003/howtobuy/licensing/pricing.msp>.
8. Microsoft Product Information Center. 2006. Microsoft Corporation. 9 Feb 2006.
<http://www.microsoft.com/products/info/product.aspx?view=22&type=ovr&pcid=2abf99cd-a5e4-469c-802e-55ca8ec542d5>.
9. MSDN Home Page. 2006. 9 Feb 2006.
<http://msdn.microsoft.com/vstudio/previous/2003/overview/default.aspx>.
10. Mwave. 2004. 28 May 2005. <http://www.mwave.com>.
11. Newegg. 2005. 27 May 2005. <http://www.newegg.com>.
12. TrainCert.net. DotNetNuke 3.0 Product Overview. Online training video.
<http://dotnetnuke.com/About/TrainingVideos/tabid/810/Default.aspx>.
13. UC Branding Initiative. 2002. 2 Nov 2005. <http://www.uc.edu/branding/>.

14. Walker, Shaun, Patrick J. Santry, Joe Brinkman, Daniel Caron, Scott McCulloch, Scott Willhite, Bruce Hopkins. Professional DotNetNuke ASP.NET Portals. Indianapolis, IN: Wiley Publishing, 2005.
15. Women in Information Technology. 2002-2004. University of Cincinnati. 16 May 2005.
<http://uc.edu/wit>