

Spring 5-1-2004

Customizable, Secure Web-Based Message-Board System

Daniel P. Ohnemus
Dakota State University

Follow this and additional works at: <https://scholar.dsu.edu/theses>

Recommended Citation

Ohnemus, Daniel P., "Customizable, Secure Web-Based Message-Board System" (2004). *Masters Theses*. 48.
<https://scholar.dsu.edu/theses/48>

This Thesis is brought to you for free and open access by Beadle Scholar. It has been accepted for inclusion in Masters Theses by an authorized administrator of Beadle Scholar. For more information, please contact repository@dsu.edu.

Customizable, Secure Web-Based Message-Board System

by

Daniel P Ohnemus

A project submitted in partial fulfillment

of the requirements for the

Master of Science in Information Systems

Dakota State University

2004



MSIS
PROJECT APPROVAL FORM

Student Name: Dan Ohnemus

Expected Graduation Date: 5/04

Master's Project Title: Customizable, Secure Web-Based Message-Board System

Date Project Plan Approved: 12/03

Date Project Coordinator Notified and Grade Submitted: 5/04

Approvals/Signatures:

Student: [Signature]

Date: 4/27/04

Faculty supervisor: [Signature]

Date: 4/27/04

Committee member: [Signature]

Date: 4/27/04

Committee member: [Signature]

Date: 4/27/04

To Ensure Certification of Completion:

Student must bring or send the original to the Graduate Programs Office.

Copies on acid free paper go to the library with the reports for binding.

Original to Graduate Programs Office

Acid-free copies with written reports to library

Copies to: Project supervisor and committee and to MSIS Coordinator

Abstract

The following paper outlines in detail the steps taken to create an online forum using PHP and MySQL, in partial fulfillment of the requirements for the MSIS degree at Dakota State University. This includes a brief introduction to the project and its intent; a statement of the problem that was meant to be addressed by the project; the objectives to be accomplished by the project, including deliverables; the scope of the project, including details of how the system was built; and appendices containing the original project plan and the source code for the system.

The project scope is a detailed explanation of how the system was constructed, how users interact with it, and why specific decisions were made in the system construction. The system consists primarily of two distinct means of interaction, the forums and the administrative functions. The two areas are separated by use of user security levels assigned to each unique user.

The project was completed essentially according to plan and the system behaves in a manner consistent with the objectives. The design is robust and extensible and the system is as secure as was intended from the planning stages. The experience of building the system was quite rewarding and I believe I was able to use many of the skills that I have acquired in my time at DSU.

Future expansion of the system is possible given the current architecture and several possible development tasks are outlined at the end of the paper. In this way, the project can be considered somewhat ongoing. Future development could create a viable commercial product, if time and the market allow.

Acknowledgements

The following people have been instrumental in my completion of this project and the MSIS program at Dakota State University.

Project Committee:

Dr. Zehai Zhou - Chairman
Dr. Ronghua Shan
Dr. William Figg

Advisor (To me and everyone else in the program): Dr. Terry Dennis

Graduate Department Director: Laurie Dennis

Graduate Department Secretary: Dawn Hamilton

I would also like to thank my wife and children for allowing me the time away from them to complete my degree. It has been difficult at times, but the rewards have been great.

Table of Contents

Project Approval Form	ii
Abstract	iii
Acknowledgements	iv
List of Figures	vi
List of Tables	vii
Introduction.....	1
Statement of Problem.....	3
Objectives and deliverables	6
Scope.....	8
Database Design.....	8
System infrastructure and initial steps	14
System overview.....	16
Administrative pages and system security	21
Main forum and message pages.....	32
Discussion, conclusion and future work	43
Works Referenced.....	47
A. Appendix A – Project Plan.....	A-1
B. Appendix B – Source Code.....	B-1
1. Utility Files	B-1
2. Include Files.....	B-3
3. System files.....	B-6

List of Figures

Fig. 1 – Initial database design	8
Fig. 2 – Final database design.....	10
Fig. 3 – Initial system page layout.....	17
Fig. 4 – Final system design	18
Fig. 5 – User interaction with the system	20
Fig. 6 – User Administration page.....	21
Fig. 7 – User add/edit page, with example form validation.....	22
Fig. 8 – Login page	23
Fig. 9 – IP logging of users.....	24
Fig. 10 – Main Page.....	25
Fig. 11 – Forum Administration	26
Fig. 12 – Forum User administration.....	27
Fig. 13 – Adding existing user error	28
Fig. 14 – Add/Edit Forum page	28
Fig. 15 – Skin Administration.....	29
Fig. 16 – Skin add/edit.....	30
Fig. 17 – Filter Administration	31
Fig. 18 – Main Forum page.....	32
Fig. 19 – Forums for the Literature implementation.....	33
Fig. 20 – Example forum, Computer Company implementation.....	34
Fig. 21 – MSIS implementation, no messages posted	34
Fig. 22 – Message (thread) detail page – Literature implementation	36
Fig. 23 – Search page.....	37
Fig. 24 – Search results.....	38
Fig. 25 – Self-registration page.....	39
Fig. 26 – User Forum, access request thread	40
Fig. 27 – Mockup of more complex design	41
Fig. 28 – Mouse-over mockup.....	42

List of Tables

Table 1 – Database table definitions.....	11
Table 2 – tblfilters data definitions.....	11
Table 3 – tblforum data definitions.....	11
Table 4 – tblmessage data definitions.....	12
Table 5 – tblsecurity data definitions.....	12
Table 6 – tblskin data definitions.....	12
Table 7 – tbluser data definitions.....	13
Table 8 – tbluserforum data definitions.....	13

Introduction

For my final project for the MSIS degree at Dakota State, I set out to build an online message board system, henceforth to be referred to as the forum. This forum was to be written in PHP using MySQL as the back-end DBMS. Additionally, I chose to use Apache as the Web server, rather than the IIS built into Windows XP Professional, as I am highly familiar with IIS and wished to have this project be both a challenge and a learning experience. I do not normally use open-source software or languages in my daily work, given our reliance primarily on Wintel servers running various incarnations of the IIS Web server. I installed Apache under Windows XP professional, but did consider installing it under Mandrake Linux, as well.

This forum was to be easily customized so that it could be used in a variety of situations with minimal changes. Three potential applications of the forum were initially identified: a literary forum, an MSIS forum, and a forum for a computer company to provide technical support information. I have constructed a system that can easily accommodate all three of those areas and have also created unique designs, or “skins” for each of them. These skins are implemented in the current system through a number of logos specific to each forum and through the use of different style sheets for each skin. For example, one forum may have links that are purple and turn orange when hovered over while another may move from green to red.

The forum consists of two distinct areas separated by user security levels. This main forum area is the heart of the system and is the means by which the majority of users would interact with the system. It consists of a list of available forums and within those forums, messages and threads, or lists of related messages. These messages and

forums are likely familiar to anyone who has spent time on the Internet and has joined in one of the many community forums available there. Our own interaction as Internet students in the MSIS program relies heavily on message board systems.

The administrative area is meant to support the core mission of the forums. It allows the system administrator(s) to create users and edit user information, create and edit forums, create and edit skins, and create and edit filters. All these items have been addressed and will be addressed in further depth in the pages to come.

In addition to a Web interface, the system has at its core a MySQL database that has been normalized and designed for reasonably good performance. Further refinement of the database for a very large implementation of the system might be required, but for the purposes of the system as it stands today, the database is more than adequate.

The objectives of the original system design were to create an extensible forum that could be reasonably tightly administered. Again, details of the system objectives and how those were addressed will be presented in the appropriate context, as will the details regarding how the system was constructed and what future developments are possible with the current architecture or with a modified architecture extended from the current system.

Statement of Problem

The original impetus for this project had been a request from the business unit with which I work to create a message board system for the technicians to use to communicate with one another at Gateway. I kept this at least in mind as I developed the system, but in addition to this application, there were additional venues that I felt would be appropriate for such a system. Sadly, the business has been stripped of any discretionary budget and it is very unlikely that anything I have created will be implemented at Gateway. Recent changes in management at Gateway make the PHP/MySQL architecture a bit more feasible than it was previously, but the budgetary constraints still remain and make the project unlikely to proceed in the near future.

There were two situations beyond the technical support forum identified in this project in which a robust message board system could be of great use. The intention in this system design was to create a forum that could be easily adapted to any of these situations by allowing various skins to be applied to the forums. I have created skins for each of the three areas mentioned above, as well as a generic system skin that is in use on the administration pages. Beyond the system I have constructed, there are currently a number of commercially available message board systems with differing levels of functionality and extensibility. Many of these would be very adequate for the situations this system is designed to address, but would incur cost beyond my simple labor and would have certainly robbed me of the learning opportunity.

Among these applications, a variety of technologies are employed. Several solutions are available written in the ASP programming language. This has the

advantages of a large installed customer base, given the prevalence of Windows-based servers. There are, however, many Web sites that do not operate on Windows-based systems and would therefore not be able to use these ASP-based solutions. There are also several .Net solutions on the market, but these are more proprietary than the ASP solutions, so would suffer the same lack of portability to non-Windows systems. An additional disadvantage to .Net architecture is the need for both the server and client to have the .Net framework installed.

A few compiled solutions are available that are portable to a number of platforms, but these solutions are only extensible to a point. Because they are proprietary solutions, these would not leave much room for the customer to make modifications. Additionally, the cost associated with more complex, compiled solutions is much higher than the other available systems.

There are several popular PHP-based solutions. These are relatively robust and allow you to create your system from scratch. If I was in the market for a more commercial application of a forum system, it is a possibility that I might choose one of these systems. PHPBB2 and Phorum are both good systems with relatively large user bases and relatively intuitive setups. I downloaded both of these systems from their respective Web sites to use as somewhat of a guide in my development. Upon examination, however, I came to the conclusion that these systems were not in keeping with what I was attempting and that looking at them too closely would not prove to be beneficial in my development efforts.

The purpose of this project was not only to implement a forum system, but to further my education in using open-source tools and languages and to provide somewhat

of a challenge in that regard. To that end, my buy vs. build decision was likely not based on an entirely realistic assessment of ROI. If this were to be used for customers, I would more than likely select one of the commercially available systems. I would be inclined to build the system myself if it were intended to be used primarily by technicians, however. The cost reduction and ability to customize the system to exact specifications would be important factors in this decision. While there is sometimes reduced cost in purchasing a system initially, if large customizations are going to be required, the costs often rise exponentially. There is also the problem of support for any highly customized system. Many software vendors will be unwilling and occasionally unable to support a product that has been customized beyond its original design.

Objectives and deliverables

Given the decision to build the system, the deliverables I have provided, in addition to this paper outlining the project, are the system itself, which I hope to make available at least in demonstration during the presentation to accompany this report, and appendices to the report that contain the original project plan and the entirety of the PHP source code for the system. This should give a very detailed picture of just what was created and how I set out to create it.

The deliverables are in line with what I had outlined in my planning documentation and turned out largely as expected. In looking at my planning document in retrospect, I am unsure how I had planned to provide the database itself as a deliverable, but it will certainly be at the heart of deliverables that are provided. While it is certainly there, two diagrams showing the original design and the eventual design of the database should suffice to show its structure.

The objective of the project was to provide a portable, extensible message board system that can tightly administered. This objective has been met fairly well, I believe, although minor modifications were made to the scope of the system as development progressed. I found that I had forgotten some aspects of the system that would be required when I had done my original planning and also decided to add functionality that I had not previously considered. The main item in this added functionality is the ability to search the entirety of the message titles and bodies in the system for specific keywords and return either a message or thread. This was something that I had found in other forum systems that I liked and that was relatively easy to implement.

In addition to the tangible objectives of this project, there were also intangible objectives that were implicit in my planning. I hoped to learn as much as possible about open-source technologies, given my limited opportunities in that arena in my work life. As should become apparent through the details in this report, I did achieve that objective to an extent that I am satisfied with. I am particularly pleased to have had the opportunity to make further use of the Apache Web server, which I am growing to like more and more despite its being configured entirely without the aid of a GUI.

Scope Database Design

As I had originally planned, my first step in the creation of the forum system was to design the database. I decided on the entities that would be required of a system such as this and what attributes those might need to have. The original database design is certainly not what I eventually ended up with, although the structure is very similar. It contains all of the entities I had originally envisioned, but the attributes associated with those entities have changed somewhat, with some being combined (tblSkin) and several being added (tblMessage). An E-R diagram of the original database can be found below:

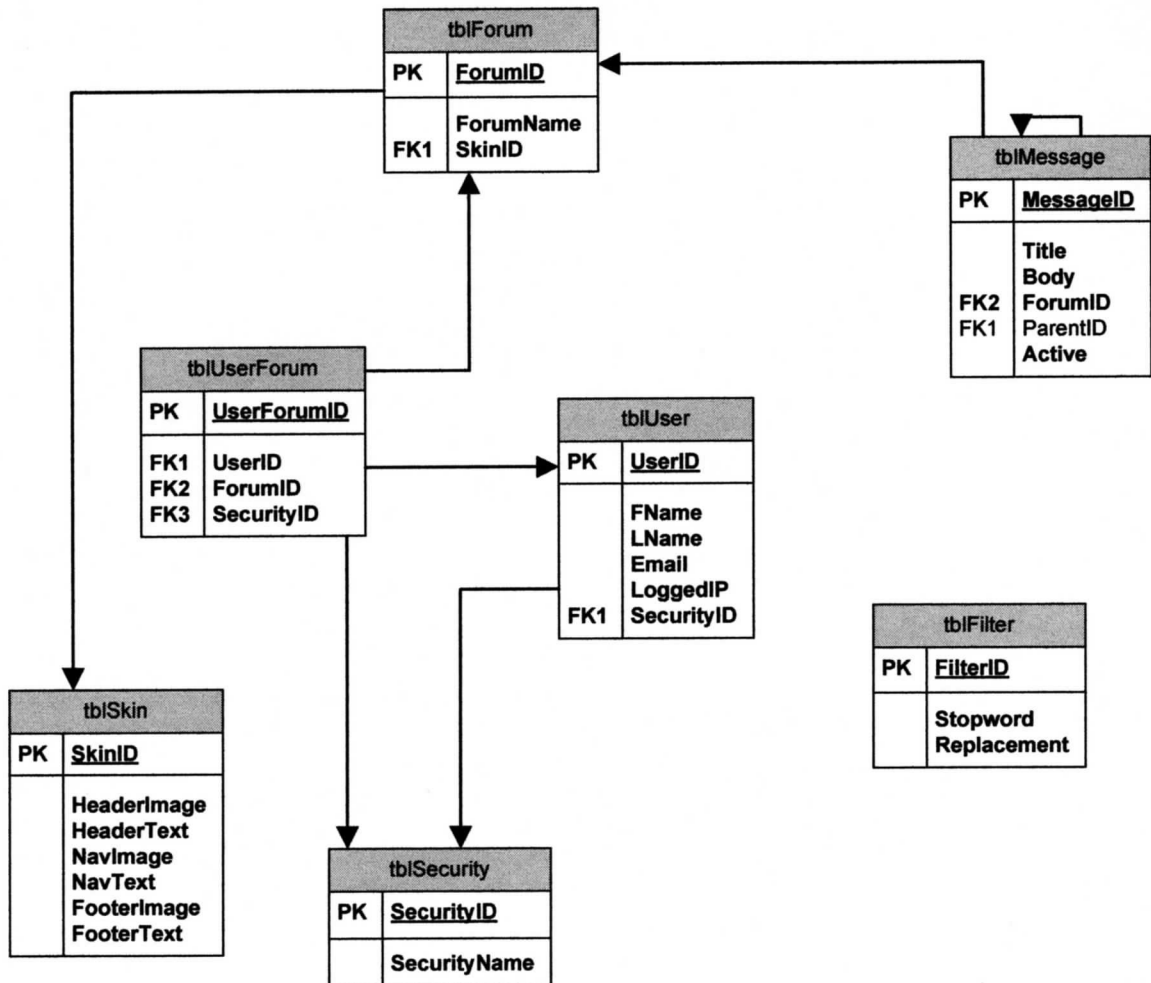


Fig. 1 – Initial database design

All ID fields in the database were to be self-incrementing data types in MySQL so that they were guaranteed to be unique for use as primary keys in each respective table. This is a method I have used in other applications and I find it to be extremely useful. Many commercial DBMS have this functionality and it is something that I have made liberal use of in the past and intend to use frequently in the future.

The SecurityID field in the UserForum table was originally intended to allow the system to assign different security levels to users for different forums. For example, a user could be a forum administrator in one forum, but merely a guest in another. The logic behind this proved to be somewhat problematic, however, and the possible situations where this could be useful functionality didn't seem to be very prevalent upon further examination. For these reasons, I eliminated the SecurityID foreign key prior to doing any of the PHP development for the system.

Additional changes to the database included the addition of fields to the messages entity and the addition of foreign key relationships for several of these fields. These fields were added to the message table to accommodate better logging of user interaction with the system and tracking of messages. This tracking and logging should have been inherent to the system from the outset and the oversight on my part was a good learning experience. A current view of the database structure can be seen below:

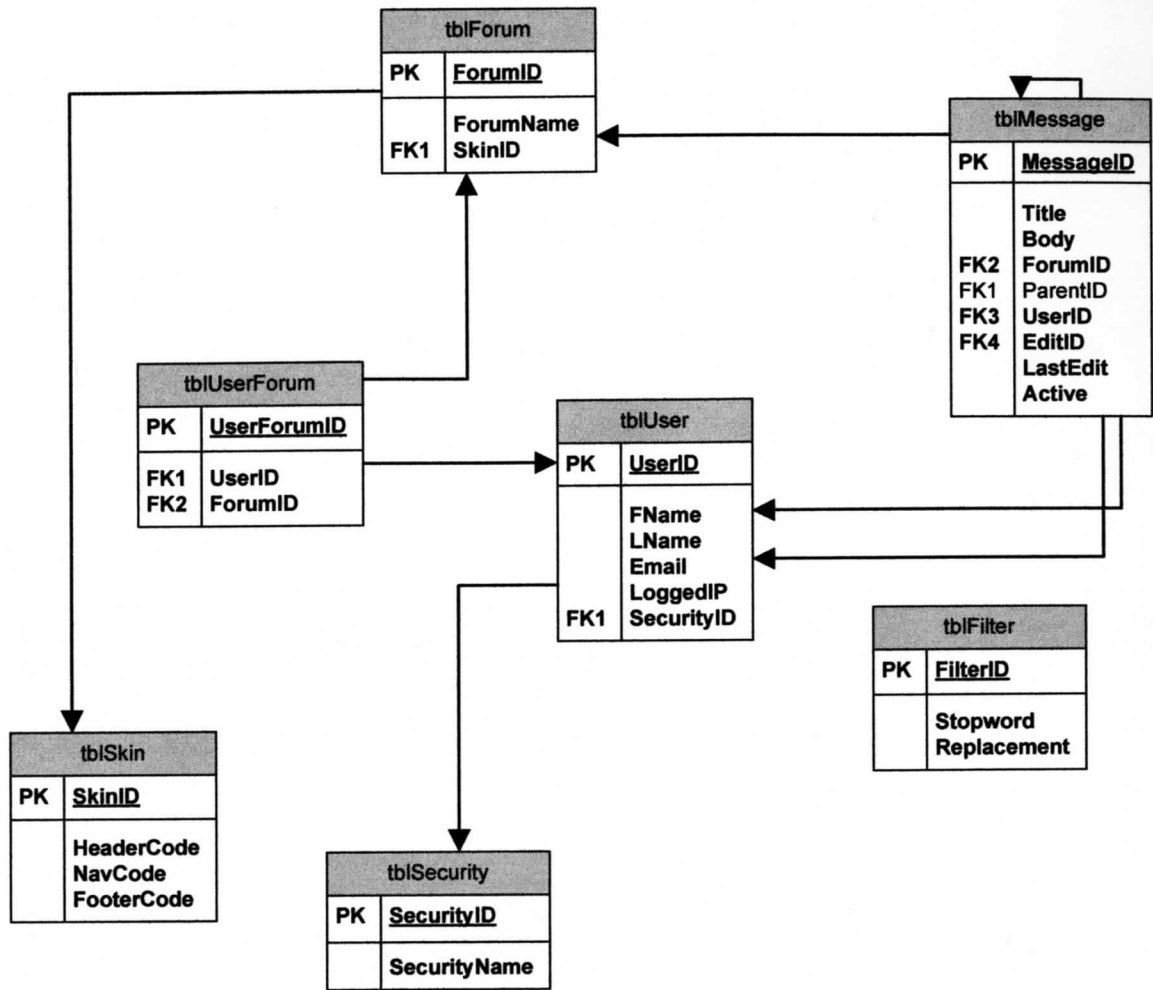


Fig. 2 – Final database design

It was my feeling that this structure would be adequate to meet the needs of the system. The database is fairly well normalized and allows for extensibility if additional functionality is desired from the system in the future. There is still the potential to assign security by forum by adding the SecurityID foreign key back into the bridge table, but I have no immediate plans to do so. Additionally, the skins could be expanded to allow the system administrator tighter control of the look, and the user table could have an icon added to allow personalization within the forums. This icon would be stored as a varchar field that was merely a link to an icon that would be stored within the same file structure

as the forum itself. The icon could be uploaded to the file system through the PHP code using a page that would allow users to alter their own information within the system.

The tables within the database are structured using mostly integer and varchar data types. The two exceptions to this are the Body and LastEdit fields in the tbluser table. These are text and date data types, respectively. Metadata for the database can be viewed below. All table identifiers are also auto_increments within the MySQL database. All data types are the MySQL physical types, rather than portable types.

Table Name	Table Definition	Identifier
tblfilters	Table to store stopwords and replacements for filtering of offensive language	FilterID
tblforum	Table to store information pertaining to forums	ForumID
tblmessage	Table to store all information related to messages	MessageID
tblsecurity	Table to store security levels and names	SecurityID
tblskin	Table to store location pointers for skin include files	SkinID
tbluser	Table to store all user information	UserID
tbluserforum	Table to bridge the user and forum tables – allows association of multiple users with multiple forums	UserForumID

Table 1 – Database table definitions

Field Name	Definition	Data Type and Size
FilterID	Identifier for the tblfilters table	int (10) unsigned
Stopword	Word to be replaced within title of body of message	varchar (30)
Replacement	Word to act as replacement for stopword	varchar (30)

Table 2 – tblfilters data definitions

Field Name	Definition	Data Type and Size
ForumID	Identifier for the tblforum table	int (10) unsigned
ForumName	Name for forum as it will appear to users	varchar (100)
SkinID	Foreign key with relationship to tblskins.SkinID	int (10) unsigned

Table 3 – tblforum data definitions

Field Name	Definition	Data Type and Size
MessageID	Identifier for the tblmessage table	int (10) unsigned
Title	Title of the message as it appears to users	varchar (50)
Body	Body of the message as it appears to users	text
ForumID	Foreign key with relationship to tblforum.ForumID	int (10) unsigned
ParentID	Recursive foreign key with relationship to tblmessage.MessageID	int (10) unsigned
UserID	Foreign key with relationship to tbluser.UserID	int (10) unsigned
EditID	Foreign key with relationship to tbluser.UserID	int (10) unsigned
LastEdit	Timestamp storing the last date and time the message was edited	datetime
Active	Boolean that marks the message as active or inactive	tinyint(1)

Table 4 – tblmessage data definitions

Field Name	Definition	Data Type and Size
SecurityID	Identifier for the tblsecurity table	int (10) unsigned
SecurityName	Name for the security level as it appears to users	varchar (20)

Table 5 – tblsecurity data definitions

Field Name	Definition	Data Type and Size
SkinID	Identifier for the tblskin table	int (10) unsigned
SkinName	Name of the skin as it appears to the users	varchar (100)
HeaderCode	Relative location of header include, expressed as a virtual path	varchar (100)
NavCode	Relative location of navigation include, expressed as a virtual path	varchar (100)
FooterCode	Relative location of footer include, expressed as a virtual path	varchar (100)

Table 6 – tblskin data definitions

Field Name	Definition	Data Type and Size
UserID	Identifier for the tbluser table	int (10) unsigned
UserName	User name for use in user authentication at login – Unique constraint applied	varchar (30)
Password	Password for use in user authentication at login	varchar (30)
FName	User's first name	varchar (50)
LName	User's last name	varchar (50)
Email	User's email address	varchar (50)
SecurityID	Foreign key with relationship to tblsecurity.SecurityID	int (10) unsigned
LoggedIP	IP address of the user at the time the last logged into the system	varchar (20)

Table 7 – tbluser data definitions

Field Name	Definition	Data Type and Size
UserForumID	Identifier for the tbluserforum table	int (10) unsigned
UserID	Foreign key with relationship to tbluser.UserID	int (10) unsigned
ForumID	Foreign key with relationship to tblforum.ForumID	int (10) unsigned

Table 8 – tbluserforum data definitions

Again, the database is normalized and optimized for a small implementation. Larger implementations could potentially split the message body from the rest of the message table to improve performance, but there would likely need to be a fairly large amount of data in the system before any performance degradations would be suffered. As it stands today, the system performance is quite good.

System infrastructure and initial steps

After the initial database design, my next step was to acquire the software needed to make the system function. I had decided on Apache, PHP, and MySQL and acquired the latest versions of each of them from their respective Web sites. The system is using Apache 2.0.48, PHP 4.3.4, and MySQL 4.0.17. Additionally, I downloaded and installed the MySQL Control Center 0.9.4 beta. As much as I enjoy doing all my database creation and manipulation from a command prompt, I thought it would make things go much more smoothly to actually have a GUI for MySQL and I have previously used and was quite impressed by the MySQL Control Center.

My decision to use Apache, PHP, and MySQL was also influenced heavily by their inherent compatibility. A great deal of effort has been put into assuring that all of these applications run together smoothly and I did not desire spending inordinate amounts of my development time in configuring the infrastructure. PHP within IIS was an unknown commodity for me and I did not want to introduce an additional element of risk. I also find Apache to be much more stable than IIS and the tight integration of PHP and MySQL with the built-in objects and methods for MySQL within PHP were very important to helping in rapid development. As I stated earlier, I also considered Mandrake Linux as an OS, but decided against it because I really have only one computer for both development and entertainment and, frankly, I wanted to make sure my games would still run.

One other fairly obvious factor in determining my choice of software that can be freely downloaded is the very fact that it can be freely downloaded. This eliminates the need for any proprietary software purchases and increases the portability of the system.

The system is platform independent when using open-source software and, again, the cost is definitely difficult to improve upon.

The next step after installation of all the required systems was to configure them. I installed PHP as a SAPI module in Apache, rather than as a CGI application. This was done to help overall system performance and security. I also configured the administration user in MySQL and created a user specifically for the forum database. I then created the initial version of the forum database using the MySQL CC and began planning the order in which I would create my system pages. This planning was influenced heavily by previous experiences I have had with systems that are at least similar in function and design. In most instances, I have begun with the administration pages to allow me to use the system to create basic functionality, as it is designed, rather than adding this functionality manually to the database. This allows more efficient unit testing and has always helped my development by giving some sense of continuous accomplishment.

One disadvantage to using MySQL as a DBMS is the fact that the current version does not support foreign key constraints. This exclusion is entirely deliberate on the part of the developers of the MySQL software. The MySQL Web site at one point had a somewhat involved treatise on why this decision was made, but at its core it basically states that anyone who relies on the DBMS to tell them when they are not managing their data inserts, etc, properly is not a very good programmer. While I have no intention of weighing in on whether foreign key constraints make for lazy application developers, I took pains to ensure I was dealing with relationships programmatically.

System overview

The system consists of approximately 29 PHP files, including two utility files, one for security and the other for database configuration. These utility files are included on nearly every page. Additionally, I created 12 include files for the four skins used in the system and one .js file for form validation. I will readily admit that the .js file was not created from whole cloth for this project. I have written a number of form validation functions for work and I felt no desire to re-invent the wheel, so to speak. I removed the majority of the validation functions from this utility file, however, as I had no need to validate if fields were equal or if one out of three fields was blank, etc, in this project. Beyond this code, I created several images for use as logos. I also created a number of button images and roll-over javascript code for those buttons. While these buttons were scrapped eventually, it could be added back in again at a later time and will be addressed in greater detail later in the report.

The dbconfig file, which is used on every page and consists mainly of the connection string for the database, was the first page created, followed immediately by the user administration pages. I started the more extensive development with these pages because they would be integral to further testing in the system. I could not test the security aspects until I had the ability to create a user that could attempt to log in, nor could I test other administrative functions without being an administrator who was currently logged into the system. Creation of one or more users could have been done directly in the database, but user administration seemed to be a logical starting point for the coding.

I had originally envisioned a fairly simple system with a majority of the form pages submitting to themselves. This proved to be somewhat less desirable in practice, and modifications to this were eventually made. Here, however, is the original layout for the Web pages used within the system:

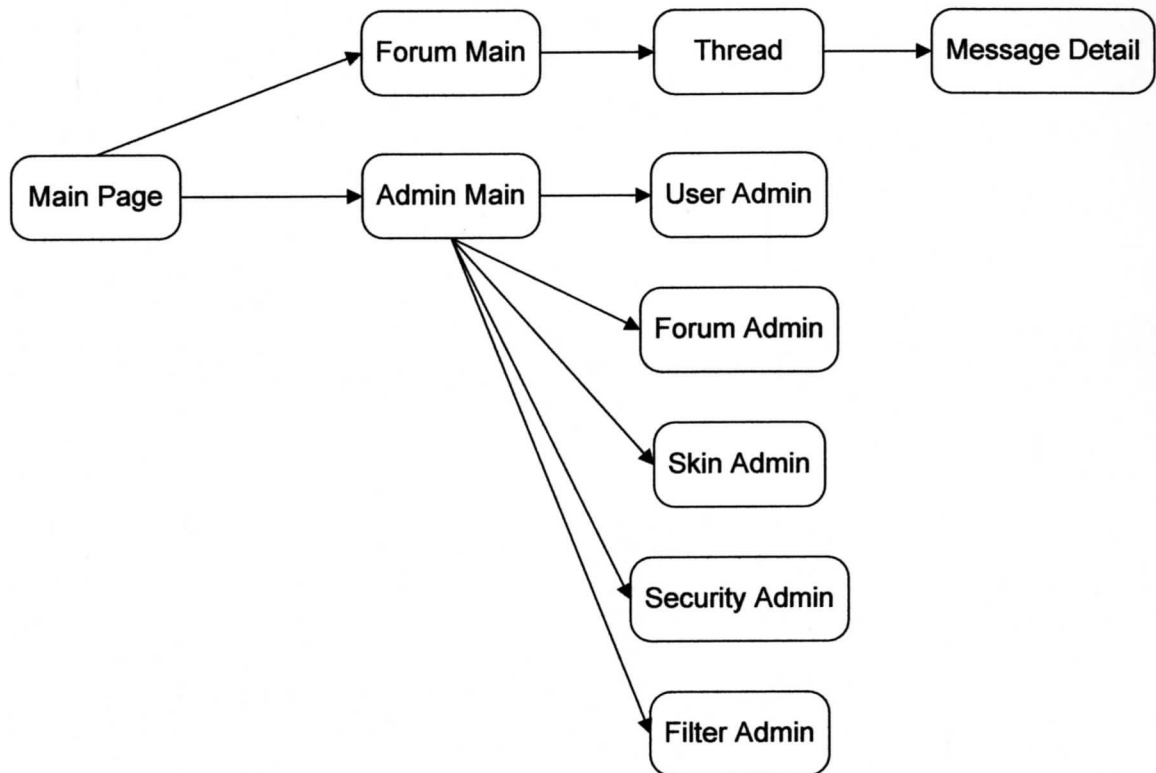


Fig. 3 – Initial system page layout

This initial layout was modified considerably as the development progressed. Individual pages were replaced with groups of pages, primarily in threes. These groups consist of a list page, an add/edit page, and a commit page that stores the data in the database. Examples of the various list and edit pages are provided within the respective administration areas. The commit pages simply show the user what they have entered and direct them back to the administration main page, so are likely not very worthwhile to include as screen-shots.

My decision to use a set of three pages for each area, rather than have one page submit to itself and commit the database changes, was based on a desire to keep simplicity in each page. The more complexity added to a single page, the greater the chance for problems and the more difficult it is to troubleshoot those problems, should they arise. While it is not possible to avoid complex systems in everyday development circumstances, an application such as this likely does not need to have complexity added artificially.

The system eventually became what is seen in the diagram below:

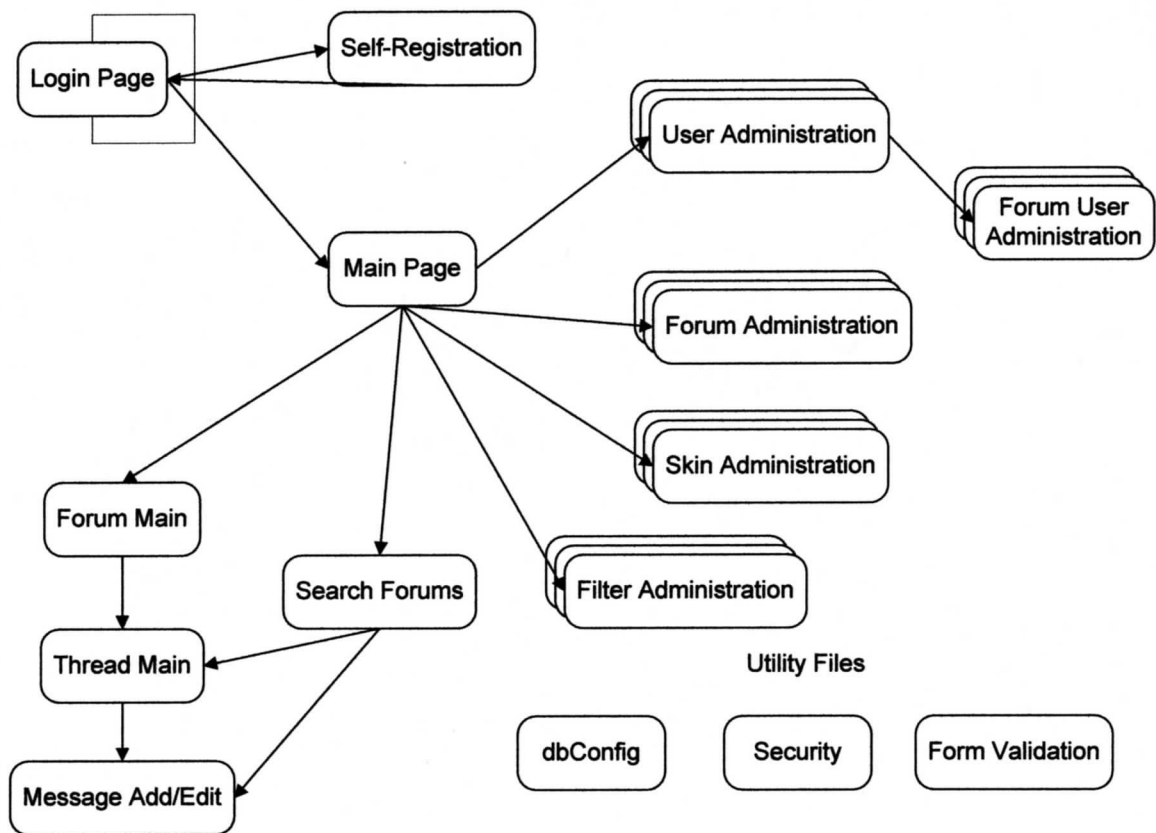


Fig. 4 – Final system design

Users are required to enter the system from the login page. This is governed by security functions that will be addressed in detail later, but the flow is always the same.

A user must first log in, then if the system validates that they are a currently active user, they are directed to the main page. If they are not a valid user, an error is returned that informs them of their status and presents them with the ability to either attempt logging in again or to proceed to the self-registration page. If they want to register as a forum guest, they may do so, but then must return to the login page to log into the system using their new username and password. The session variable for logged in users is currently only set by the login page itself.

After logging in, normal users are allowed to go to the main page, the forum main page, or the search forum page. Administrators are also given these three options, but are additionally given the ability to go to any of the system administration pages. If a user chooses to go to the forum pages, they are presented with a list of forums available to them, as defined by the system administrator. The User Forum is available to all users and allows them to request access to other forums and to increase their rights to the system.

Upon entering a forum, a user can view any of the available threads and messages and, if they have the proper permissions, they can reply to those messages. This functionality is relatively simple and is inherent to all forum systems. For a detailed view of a typical user flow through the system, refer to the flow-chart provided below:

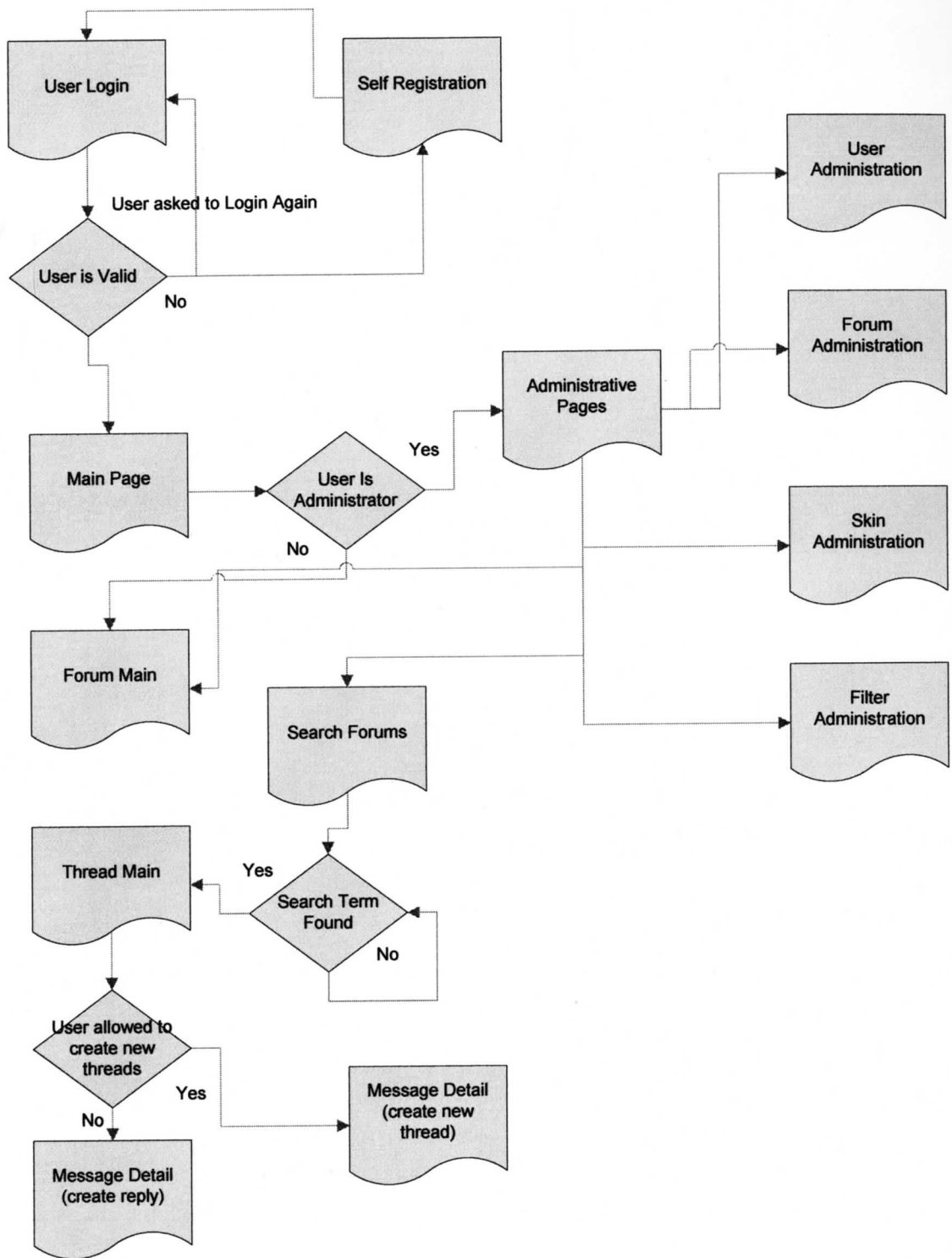


Fig. 5 – User interaction with the system

Administrative pages and system security

The user administration page is a simple listing of the active users within the system with links to edit their information and an add link to create a new user in the system. Clicking on the add or edit links will direct the user to the same page, but query strings control the presentation of that page. A query string instructing the edit page to create a new user will bypass a select statement against the database for the current user information. If the query string calls for an edit, the user information will be pulled from the database based on a second query string (form) field, the UserID, and auto-populated into the appropriate fields in the Add/Edit form.

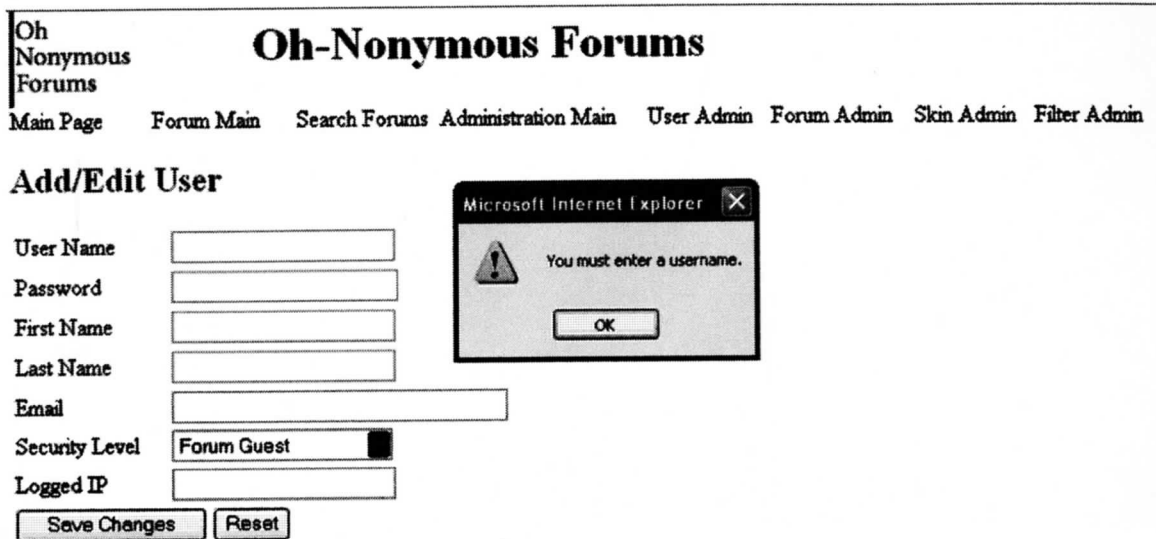
Oh Nonymous Forums	Oh-Nonymous Forums						
Main Page	Forum Main	Search Forums	Administration Main	User Admin	Forum Admin	Skin Admin	Filter Admin
User Administration							
							Add User
User ID	User Name	First Name	Last Name	Email	Security Level	Logged IP	
1	dano	Dan	Ohnemus	ohnemusd@pluto.dsu.edu	System Administrator	127.0.0.1	Edit User
2	johnny	Johnny	Test	jtest@company.com	Forum Guest	127.0.0.1	Edit User
3	billy	Billy	Test	btest@company.com	Power User	127.0.0.1	Edit User
8	aidan	Aidan	Ohnemus	aidan@localhost.com	Forum Guest	127.0.0.1	Edit User
7	jeanne	Jeanne	Ohnemus	johnemus3@localhost.com	Forum Guest	127.0.0.1	Edit User

Copyright 2004, Oh-Nonymous Forums

Fig. 6 – User Administration page

All form fields on the edit/add page are required, including the logged IP address. This is not likely to be manipulated by the administrator, but I did provide that functionality, should it be desired. For the most part, the logged IP would likely need to remain whatever was dynamically created by the system when the user last logged in. Once all the form fields are filled out and validated using the javascript code, the information is submitted via POST to a commit page. If a UserID is provided to the

commit page, an update action is performed against the database. If it is not provided, an insert is performed instead.



Copyright 2004, Oh-Nonymous Forums

Fig. 7 – User add/edit page, with example form validation

After the user administration was created and I had created a system administrator and a few normal users using the user administration functionality, I was able to create a security utility file and login functionality. System security is based on a simple login and on user security levels within the system. The levels from order of least to greatest access are forum guest, forum user, power user, forum administrator, and system administrator. It was my original intention to have these security levels be customizable, but after constructing several areas of the system it became apparent that the security levels would need to remain relatively static for the system to function properly.

Login Page

Username:
Password:

If you are not a current user of this forum system and would like to register a new guest account, [click here](#).

Copyright 2004, Oh-Nonymous Forums

Fig. 8 – Login page

The login page does a comparison of the entered username and password to the username and password stored in the database by performing a select with a compound where clause. If a match is found (greater than zero rows returned), several session variables are written, including the user's UserID. This UserID is then used to return the user's information, which is stored as an associative array for use on every page. This allows the system to compare the current user's security level against pre-determined levels that should have access to certain areas or features. Additionally, this provides a level of personalization for the user and allows the system to track who posted what messages. The navigation bars of each of the skins created for this exercise have functionality within them that hides administrative links from users that should not be allowed to see that functionality.

Each of the administrative pages contains a variable that identifies them as such and redirects the user back to the main page if they attempt to access an area that they should not. This redirect is an expansion of the basic functionality within the system that redirects anyone not logged into the system back to the login page and is included in the security utility file. Both of these redirects are based on a check for the UserID session variable. The redirect is accomplished by use of a `header()` call within the PHP that

effectively changes the URL of the current page on the fly. This is a method I have used extensively in the past in ASP with response.redirects and I was glad that similar functionality could be found in PHP. Because this redirect occurs at the very top of each page and effectively changes the HTTP headers for the page, it is reasonably difficult to bypass, particularly for the bulk of users a system such as this might have.

The last aspect of the security is the logging of user IP addresses. This occurs every time a user logs into the system and the update of this field occurs immediately following successful validation of username and password. If a user moves or uses a different computer, their most recent IP is stored. A possible future expansion of the system might be to keep a history of IP addresses associated with a particular user. This might allow for greater control if the user should become abusive and need to be banned from the system.

Oh
Nonymous
Forums

Oh-Nonymous Forums

[Main Page](#) [Forum Main](#) [Search Forums](#) [Administration Main](#) [User Admin](#) [Forum Admin](#) [Skin Admin](#) [Filter Admin](#)

Add/Edit User

User Name	<input type="text" value="johnny"/>
Password	<input type="password" value="*****"/>
First Name	<input type="text" value="Johnny"/>
Last Name	<input type="text" value="Test"/>
Email	<input type="text" value="jtest@company.com"/>
Security Level	<input type="text" value="Forum Guest"/>
Logged IP	<input type="text" value="127.0.0.1"/>

Logged IP address for user Johnny Test

Copyright 2004, Oh-Nonymous Forums

Fig. 9 – IP logging of users

Once I had users and those users had the ability to actually log in, it was time to proceed to the main page and forum administration pages. The main page was simply a list of links with only forum-related links presented to most users and administrative links presented to the system administrator(s). This functionality is somewhat redundant, given the navigation bar added to all pages in the system. It is my experience, however, that giving users a number of ways to navigate to the same content is always useful. Additionally, there is a small element of personalization to the main page that lets people know the system is aware of them. This is evident in the sentence that can be seen in the screenshot below, where my name is displayed as read from the database. In my experience, even this small level of personalization can be fairly impressive to novice users and increases their confidence in the system and their enjoyment of it.

Oh Nonymous Forums	<h2>Oh-Nonymous Forums</h2>
Main Page Forum Main Search Forums Administration Main User Admin Forum Admin Skin Admin Filter Admin	
<h3>Oh-Nonymous Forums Main Page</h3>	
<p>Please Select from the available options below, Dan Ohnemus:</p>	
Forum Options	
View Forums	
Search Forums	
Administration	
Administration Main	
User Administration	
Forum Administration	
Skin Administration	
Filter Administration	

Copyright 2004, Oh-Nonymous Forums

Fig. 10 – Main Page

The forum admin pages allow for the creation of new forums, association of users with selected forums, editing of existing forums, and the association of forums with

specific skins. It was a little problematic that the forum administration was created prior to the skin administration when I had to associate my newly created forums with skins that theoretically didn't exist, but a little manual creation in the database allowed development to progress on the forum administration prior to the creation of the skin administration. I wanted to be able to get to the real core of the system as quickly as possible so that I could do end to end testing with at least a cursory set of data.

I created a system forum and several forums for each of the three areas in the original plan. For example, the literature area contains Shakespeare, Modern Novel, and Mark Twain forums. The MSIS area has a Project forum and the Computer area has a Portable Support forum. All of these are intended to merely be examples. It is quite easy to create and edit forums using the administrative pages, so a system administrator would not need to have any knowledge of the system code to create a large base of forums.

Oh
 Nonymous
 Forums

Oh-Nonymous Forums

Main Page
Forum Main
Search Forums
Administration Main
User Admin
Forum Admin
Skin Admin
Filter Admin

Forum Administration

Forum ID	Forum Name	Add Forum	
3	Mark Twain	Edit Forum	Edit Forum Users
2	Modern Novel	Edit Forum	Edit Forum Users
5	MSIS Project	Edit Forum	Edit Forum Users
1	Shakespeare	Edit Forum	Edit Forum Users
4	User Forum	Edit Forum	Edit Forum Users

Copyright 2004, Oh-Nonymous Forums

Fig. 11 – Forum Administration

Again, the forum administration main page consisted primarily of a list of forums in the system with edit links to the right of each forum name and an add link. An edit

forum users link is also provided with each forum and takes the administrator to the Add/Edit Forum Users page. This area could likely be considered a major component area in itself and consists of much the same architecture as the remaining administration pages in the system. The Forum User Add/Edit presents a list of users associated with the current forum and a list of all users in the system. A user can be added to the forum via a provided link or removed in a similar manner.

Oh Nonymous Forums	Oh-Nonymous Forums	
Main Page Forum Main Search Forums Administration Main User Admin Forum Admin Skin Admin Filter Admin		
Add/Edit Forum Users - Mark Twain		
Current Forum Users		
First Name	Last Name	
No users are currently assigned to this forum and it can only be viewed by system administrators.		
Potential Forum Users		
First Name	Last Name	
Dan	Ohnemus	Add User
Johnny	Test	Add User
Billy	Test	Add User
Aidan	Ohnemus	Add User
Jeanne	Ohnemus	Add User

Copyright 2004, Oh-Nonymous Forums

Fig. 12 – Forum User administration

The system will check the user being added against the current users in the forum and will return an error message to the administrator if they attempt to add that user again. Given that the entire list of current users is displayed on the admin page, this is an unlikely event, but since it would almost certainly return a database error, I felt it was necessary to account for it. While descriptive errors presented by PHP for database or script problems are useful for debugging, they are much less appropriate for a production system with real users who could likely not understand the errors, anyway.

Add/Edit Forum User

The following user was NOT added to the Shakespeare forum:

First Name: Billy

Last Name: Test

The user was already associated with the selected forum.

Copyright 2004, Oh-Nonymous Forums

Fig. 13 – Adding existing user error

The forum add/edit page functions in much the same manner as the user add/edit page, with a query string differentiating between add and edit. Existing data is queried and the form fields auto-populated with that data if an edit is being called, whereas the add new functionality simply presents the empty form fields. As with the user administration, all form fields on the forum add/edit page are required and the form is submitted to another file that simply acts as the means to commit changes to the database.

Add/Edit Forum

Forum Name	<input type="text" value="Mark Twain"/>
Forum Skin	<input type="text" value="Literature"/>
<input type="button" value="Save Changes"/>	<input type="button" value="Reset"/>

Copyright 2004, Oh-Nonymous Forums

Fig. 14 – Add/Edit Forum page

Skin and filter administration were the next pages to tackle and both were relatively simple. Skins allow the system admin to point at three include files for each

named skin, a header, navigation bar, and footer. The name of each skin is also used to categorize the forums in the system, so Computer Company, Literature, and MSIS were the names used initially, with the System skin and category added later as it became apparent that it might be necessary during the course of system development. In a production system, there would likely only be one skin for all forums, but I wanted to show the abilities of the technology for this project.

Oh
 Nonymous
 Forums

Oh-Nonymous Forums

[Main Page](#)
 [Forum Main](#)
 [Search Forums](#)
 [Administration Main](#)
 [User Admin](#)
 [Forum Admin](#)
 [Skin Admin](#)
 [Filter Admin](#)

Skin Administration

Skin ID	Skin Name	Header Code	Nav Code	Footer Code	Add Skin
1	Literature	includes/lit_header.inc	includes/lit_nav.inc	includes/lit_footer.inc	Edit Skin
2	Computer Company	includes/comp_header.inc	includes/comp_nav.inc	includes/comp_footer.inc	Edit Skin
3	MSIS	includes/msis_header.inc	includes/msis_nav.inc	includes/msis_footer.inc	Edit Skin
4	System	includes/system_header.inc	includes/system_nav.inc	includes/system_footer.inc	Edit Skin

Copyright 2004, Oh-Nonymous Forums

Fig. 15 – Skin Administration

The add/edit page for the skin administration is, again, very similar in structure and code to the other add/edit pages. The include files are specified by path in the system and then a require statement on each of the forum pages pulls the appropriate includes for the forum as variables which are then placed in require statements. This allows each forum to theoretically have an entirely different look. The use of includes assumes that the system administrator will have at least rudimentary knowledge of Web development and can create and point at includes that are customized for their forum. A possible commercial route would be to provide this customization as a service to the customer who might buy the forum product. This effort would likely be minimal, but the rates for

onsite developers are still fairly good, even given the latest outsourcing trends, and could be a source of further income.

Oh Nonymous Forums	Oh-Nonymous Forums						
Main Page	Forum Main	Search Forums	Administration Main	User Admin	Forum Admin	Skin Admin	Filter Admin
Add/Edit Skin							
Skin Name	<input type="text" value="Computer Company"/>						
Header Code	<input type="text" value="includes/comp_header"/>						
Nav Code	<input type="text" value="includes/comp_nav.inc"/>						
Footer Code	<input type="text" value="includes/comp_footer.ir"/>						
<input type="button" value="Save Changes"/>	<input type="button" value="Reset"/>						

Copyright 2004, Oh-Nonymous Forums

Fig. 16 – Skin add/edit

Filter administration consisted only of giving the system admin the ability to designate a stopword and a replacement for that stopword. If the admin does not want profanity used within the system, they can specify the words they consider profane and what those are to be replaced with. There is logic in the page that commits the messages to the database that first checks the title and body for any stopwords and replaces them with the designated replacement. This logic uses simple string replacement, which processes fairly quickly, but must be very precise. If a word differs even slightly from your stopword, it will not be seen. This could be overcome at least partially by the user of a regular expression search, but this would slow things measurably. As personal preference, I would use the Perl style regular expression methods within PHP, which I am happy to see are installed now by default. I really do not like the POSIX RegEx that were the previous default in PHP.

Filter Administration

Stopword	Replacement	Add Filter
moron	nice person	Edit Filter
doofus	nice person	Edit Filter
jerk	nice person	Edit Filter
Dell	Gateway	Edit Filter

Copyright 2004, Oh-Nonymous Forums

Fig. 17 – Filter Administration

As you can see, my business loyalties cause me to add filters to the system that other users might not otherwise add.

The administration pages all combine to allow one or more system administrators to control all aspects of the system through a relatively simple user interface. As a developer, it is very important to me that an administrator with relatively little technical background be able to perform most of the functions in the system that are required to maintain it. This prevents the users from feeling they need to remain in constant contact with the developers simply to perform system maintenance.

Maintenance is often a very large percentage of any development effort and anything that can be done to minimize maintenance should be viewed as a positive step.

Main forum and message pages

Now that we had users, forums, skins, and filters for all the profanities I planned to write into my test messages, it was time to create the main forum page. Access to forums is controlled by the system administrator through the forum admin page and the only forums that are displayed for any particular user are those forums to which they have been granted access. The only exceptions to this is users with system administrator rights and the User Forum, which will be discussed at greater length later. Administrative users are shown every forum available in the system. Again, the user security level was integral to this functionality, as was a bridge table between the user and forum tables that associates multiple users with multiple forums.

Oh Nonymous Forums	<h1>Oh-Nonymous Forums</h1>
Main Page Forum Main Search Forums Administration Main User Admin Forum Admin Skin Admin Filter Admin	
<h2>Forum Main</h2>	
<p>Please choose from one of the available forums below. If a forum you wish to view is not shown below, please contact the forum administrator to request addition to the user group for that forum.</p>	
Available Forums	
<hr/>	
Mark Twain - Literature	
<hr/>	
Modern Novel - Literature	
<hr/>	
MSIS Project - MSIS	
<hr/>	
Shakespeare - Literature	
<hr/>	
User Forum - System	
<hr/>	

Copyright 2004, Oh-Nonymous Forums

Fig. 18 – Main Forum page

Within each forum, there are a number of available threads. These threads consist of a posted original message and any associated messages. Threads can only be created

by users with power user or higher access rights. Each of the threads can be clicked on and the user is taken to a detail page for that thread showing the original message and all replies. The thread page was created immediately after the forum page. In the example shown below, I had logged in as a system administrator. I therefore had the ability to create new threads within any forum in the system and the link appeared for me. A user logging in as a forum user would not see the Create New Thread link.

small town
literary
society **Literary Society Book Forums**

[Main Page](#) [Forum Main](#) [Search Forums](#) [Administration Main](#) [User Admin](#) [Forum Admin](#) [Skin Admin](#) [Filter Admin](#)

Shakespeare Forum

Please choose from one of the available discussion threads below.

Available Threads for Shakespeare ([Create New Thread](#))

[Hamlet \(The play, not any of the movies\)](#)

[MacBeth](#)

[The Tempest](#)

Copyright 2004, SmallTown Literary Society

Fig. 19 – Forums for the Literature implementation

I created the Literature forums first in my final incarnation, although my first design attempts were for the Computer Company forums. The eventual design of the Computer Company forums and an example of the possible subjects to be addressed in such a forum can be seen in the example below.



Computer Company, Inc. Forums

[Main Page](#) [Forum Main](#) [Search Forums](#) [Administration Main](#) [User Admin](#) [Forum Admin](#) [Skin Admin](#) [Filter Admin](#)

Portables Forum

Please choose from one of the available discussion threads below.

[Available Threads for Portables](#) ([Create New Thread](#))

[Video Problems](#)

Copyright 2004, Computer Company, Inc.

Fig. 20 – Example forum, Computer Company implementation

The example below shows a forum that currently contains no available threads. This display is controlled by a simple check against the number of rows returned by a select of all messages in the system with the current ForumID and a ParentID of 0. All main messages, the thread starting points, are assigned a ParentID of 0, while all replies have as their ParentID the MessageID of the message to which they are replies.



MSIS Project Forums

[Main Page](#) [Forum Main](#) [Search Forums](#) [Administration Main](#) [User Admin](#) [Forum Admin](#) [Skin Admin](#) [Filter Admin](#)

MSIS Project Forum

Please choose from one of the available discussion threads below.

[Available Threads for MSIS Project](#) ([Create New Thread](#))

There are currently no threads in this forum.

Copyright 2004, Dan Ohnemus

Fig. 21 – MSIS implementation, no messages posted

The message detail page was the next item created and it also takes advantage of the system security levels. Each message has a posted title and body, as well as the name

of the person who posted it and the date and time the message was last edited. Forum users and higher are shown a link to reply to any message, while forum guests can only view existing posts. Forum administrators and above are given the added ability to edit existing posts. This allows them to alter posts that are deemed inappropriate or abusive, but a record of the date and time of the edit and the UserID of the editor are recorded for tracking purposes. The forum that I used for most of my examples was the Literature implementation, as this allowed me to remove myself a bit from work and from my current education and reminisce about the good old days of reading Shakespeare. The example shown below reflects that and is a thread created for the discussion of the play, Hamlet. The postings I created were not entirely serious and served as a means to release some of the tension felt with an approaching deadline and lots of code yet to write. This light tone can be found in the example presented here.

All messages related to Hamlet (The play, not any of the movies)

Hamlet (The play, not any of the movies) (Edit) (Reply)

Posted by: Dan Ohnemus

Last edited: 2004-03-31 00:00:00

This is the thread for the discussion of the play, Hamlet. Hamlet is a really good play that everyone should read. Lots of people die and there's a ghost and people go crazy and talk to themselves a bunch and there's a play within a play and all sorts of fun stuff

RE: Hamlet (The play, not any of the movies) (Edit) (Reply)

Posted by: Dan Ohnemus

Last edited: 2004-04-04 21:54:11

Yeah, I was groovin' on this play myself. I used to like the Mel Gibson version, but now that Mel is a bit odd, I don't like it as much. The Kenneth Branagh version is good and Rosencrantz and Guildenstern are Dead just plain rules.

RE: RE: Hamlet (The play, not any of the movies) (Edit) (Reply)

Posted by: Dan Ohnemus

Last edited: 2004-03-31 20:23:12

I am testing the functionality of my handy-dandy date function.

RE: Hamlet (The play, not any of the movies) (Edit) (Reply)

Posted by: Dan Ohnemus

Last edited: 2004-03-31 20:52:31

Hamlet was a total nice person.

Copyright 2004, Oh-Nonymous Forums

Fig. 22 – Message (thread) detail page – Literature implementation

Again, all of the areas covered in the forum are not single pages for each function. As stated earlier, most of the admin areas have an initial list page, an add/edit page, and a commit page that writes any changes to the database. This is also true of the message details area, where I created a message add/edit page and the associated commit page. Several small tweaks were added to each of these pages during the course of development, as I found functionality that I wanted but that was not present. This was a typical iterative development process and one that I have used very successfully in the past. The disadvantage to this type of approach is that it does not allow for the level of

documentation that is sometimes required of software development, particularly in the world of CMM levels. For the purposes of a relatively simple application such as this, however, it seemed the best approach.

I next created the search page. This is simply a keyword search that does a like comparison against the title and body fields of the message table and returns a list of messages that contain that keyword in one of those fields. The user can choose whether they wish to search through all forums available to them or simply to search one specific forum. Regardless of the choice, the search is limited only to those forums that the user has access to, even if they choose to search all forums.

Oh Nonymous Forums	Oh-Nonymous Forums						
Main Page	Forum Main	Search Forums	Administration Main	User Admin	Forum Admin	Skin Admin	Filter Admin
Search Forums							
Search Term:	<input type="text"/>						
Choose Forum:	All Forums <input type="checkbox"/>						
<input type="button" value="Search"/>	<input type="button" value="Reset"/>						

Copyright 2004, Oh-Nonymous Forums

Fig. 23 – Search page

The results are presented as a series of titles and the user is given the option of viewing an individual message or an entire thread related to that message.

Search Forums

One or more matches to your search term were found in the following messages:

Hamlet (The play, not any of the movies) ([View Message](#)) ([View Thread](#))

RE: Hamlet (The play, not any of the movies) ([View Message](#)) ([View Thread](#))

RE: RE: Hamlet (The play, not any of the movies) ([View Message](#)) ([View Thread](#))

RE: Hamlet (The play, not any of the movies) ([View Message](#)) ([View Thread](#))

Copyright 2004, Oh-Nonymous Forums

Fig. 24 – Search results

At this point, I realized that it could be a bit of an inconvenience for system administrators to always have to create users in the system, so I created a self-registration page. This page allows users to enter their username, password, first name, last name, and email address. This differs from the user add/edit page in that it does not even show the LoggedIP or User Level fields, as these are pre-determined by the system. Their IP address is recorded at the time of the account creation and all self-created users are initially forum guests. From the self-registration thank you page, the new users are directed back to the login page. This functionality could very likely be changed in a future iteration of the system, however, so that upon entering their user information, the new user is automatically logged into the system and directed to the User Forum to request further access rights.

Register New Guest Account

User Name

Password

First Name

Last Name

Email

Copyright 2004, Oh-Nonymous Forums

Fig. 25 – Self-registration page

The one exception to this guest rule is the special User Forum I created. Anyone can view and post to this forum and it is intended to field user requests for forum access and increasing of rights. This guest forum was necessary so that any user could have at least some level of interaction with the system administrator. This could also be conducted via email, but I wanted to make the system as self-contained as possible. There should be no reason for a user or administrator to venture outside the confines of the system in order to make changes or have interaction that relates to the system.

All messages related to Forum access requests

Forum access requests (Edit) (Reply)

Posted by: Dan Ohnemus

Last edited: 2004-04-01 20:35:40

Post your forum access requests here.

RE: Forum access requests (Edit) (Reply)

Posted by: Jeanne Ohnemus

Last edited: 2004-04-01 20:41:42

Hi, I'd really like access to the Shakespeare forum. Thanks.

Copyright 2004, Oh-Nonymous Forums

Fig. 26 – User Forum, access request thread

I had already created the majority of the system at this point, but still needed to go back and add form validation to all pages that had form fields. This javascript validation will prompt a user if they have left a field blank and place the cursor focus in the blank field. Additionally, I added email validation so that the entered email address at least has to look something like an email address. This is done with a simple regular expression comparison. An example of the error returned by the form validation was contained in the figure for the user administration earlier in this report.

When I initially began building the system, I questioned somewhat my decision to use PHP as a programming language. I had not used it in some time and it took me a little while to remember the syntax. Once I had created several pages, however, I gained a great deal of confidence and the coding progressed much more rapidly. I also, as mentioned earlier, have come to the conclusion that I am no designer, so the system is somewhat minimalist in look. I did attempt at one point to create a bit more complex

design. This worked in principal, but given that my main goal was the creation of the back end functionality, I did not want to continue spending large amounts of time in getting it to work properly. A mockup of this original design can be seen here:

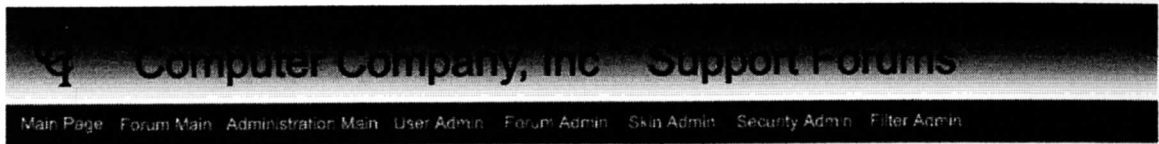


Fig. 27 – Mockup of more complex design

In addition to this mockup, I also created a mockup that would show the navigation links in a mouseover state, in this instance black text instead of white. I also created all the javascript necessary to have these mouseovers function, but eventually scrapped all this work in favor of a simpler interface. This was a somewhat difficult decision, as I had spent several hours in getting it working properly. I found the design rather blocky and unattractive, however, and felt it wasn't worth further pursuit. It does illustrate, however, that with time and a dedicated designer to assist in the creation, some relatively complex

skins could be created for the system. The version of the mockup with the mouseover images can be seen below.

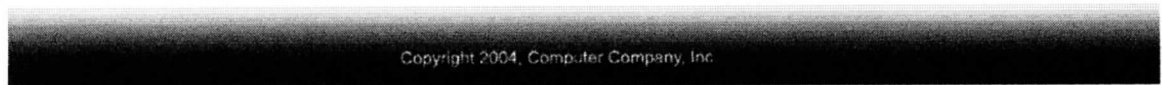
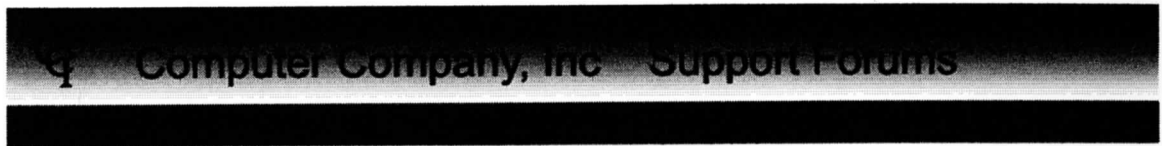


Fig. 28 – Mouse-over mockup

Discussion, conclusion and future work

The system, as created, meets essentially the goals I set out in my planning document. It is very extensible and contains all of the security and administrative functions I wanted it to have. It is perhaps a bit less complex than I had originally envisioned, but when I began coding I came to the conclusion that creating classes or other advanced programming techniques for an implementation such as this would be a bit of overkill. While it may have helped a little in the intangible learning objective mentioned earlier, it would have been of little use in the actual production system. Additionally, the look of the system is not as exciting as I would have liked, but my attempts at design were frightening and somewhat blocky, as can be seen in the mockups that conclude the scope section.

The extensibility of the system is largely what I desired and system performance, at least with the limited number of users and messages currently stored within the database, is quite good. This may be due in large part to the fact that I am accessing Apache from the localhost and therefore latency is reduced to zero. The database calls are, however, still impressively fast and are very much in line with what I have come to expect from much more expensive commercial DBMS products such as Oracle and Microsoft SQL Server. I believe that for many small to medium sized applications that MySQL would perform admirably and I have seen a number of commercial sites that use it to good end.

I learned a great deal of PHP and MySQL development in the course of this project and I believe that is something that will serve me well. My previous experience has been primarily with ASP and MS SQL Server, although I do get to do some Perl and

Python programming on occasion. Since one of my goals with the MSIS program was to increase my marketability, learning new languages and technologies was something I felt was quite important. PHP is a particularly good language to learn at this point, as its use continues to grow. Whereas two years ago I saw very few sites that made use of PHP, I am seeing more and more every day. The language is very robust for Web development, particularly when there is a need to add a database interface, and it is very fast in response, particularly when coupled with Apache and installed as a SAPI module. In comparison to similar technologies such as ASP, I believe PHP is a good competitor. When compared to something like Cold Fusion or WebSphere, there is no contest. While these two solutions have more system-level abilities, their overhead makes them slow in response to the point that I will leave a site almost immediately after seeing a .cfm or .jsp extension in the browser

All in all, this project has proven to be an enjoyable and productive exercise. I did procrastinate a bit, but ultimately was able to create essentially the system I wanted within the allotted time. The system code is available in the appendices, as is the original project plan. The code is fairly extensive and presented in a reduced font in the interest of space within the paper and in saving our old growth forests. The code should still be legible, but should not take up the bulk of this report. A pattern can be seen very quickly in the pages, as I used a technique that has likely served Web developers well for years. Given that the pages were very similar in function, I created the user administration first, then used it as a template for the remainder of the areas, changing names, variables, and SQL statements as appropriate for whatever area I was currently writing.

There is a great deal that could be done with this system in future development, if desired. I could certainly add functionality in the message area, including the ability for users to designate and upload an icon for themselves, as I have seen in some other forums. A field could be added to the database in the user table that follows the same pattern as the include fields in the skin table and file system functionality could be added to the self registration page that allow the user to upload an image to a directory determined by the system. User administration could also be extended to allow system users to alter their own information, but not anyone else's.

Additionally, given enough time I could probably make the design a little more attractive, but that is not my forte and therefore takes up much more time that it is worth. As I stated earlier, the creation of even the relatively simple design with mouseovers that I had mockups for earlier in this report took several hours that could have been more productively spent in coding my PHP. I enjoy creating designs for my Web properties, but they always end up quite blocky looking for some reason. I set out explicitly to avoid that with the design shown in the report, but my blocky demons would not let me create the elegant design I desired.

One of the other elements I would like to address, if I had additional time, is the user flow within the administrative pages. I would like to alter the commit pages so that they automatically redirect the user back to the main page for whatever administrative area they were in at the time they made the final submission to enter the data. For example, after committing the changes for the user add/edit, the user would be directed automatically back to the user administration main page. I also find that I overlooked the

ability to delete elements in the administration pages. This oversight could likely be addressed fairly easily, given additional time.

Some other small areas that could likely be expanded are the inclusion of encryption in the login for passwords and expanded functionality to allow the banning of users from the system. The encryption was not added at this time simply because it seemed a bit over the top for an online forum, although the methods built into PHP and MySQL for encryption would make it relatively easy to add.

Works Referenced

PHP: Hypertext Preprocessor. (n.d.) Retrieved March 15, 2004, from <http://www.php.net>

Apache Software Foundation. (n.d.) Retrieved March 15, 2004, from <http://www.apache.org>

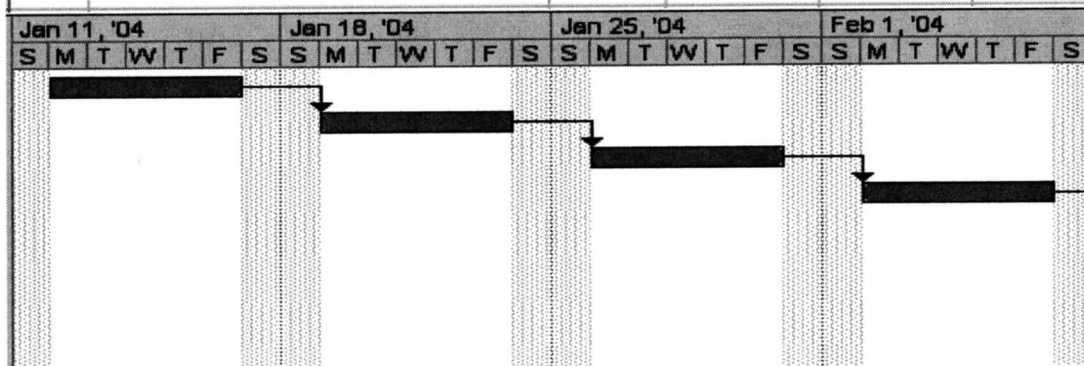
MySQL: The World's Most Popular Open Source Database. (n.d.) Retrieved March 15, 2004, from <http://www.mysql.com>

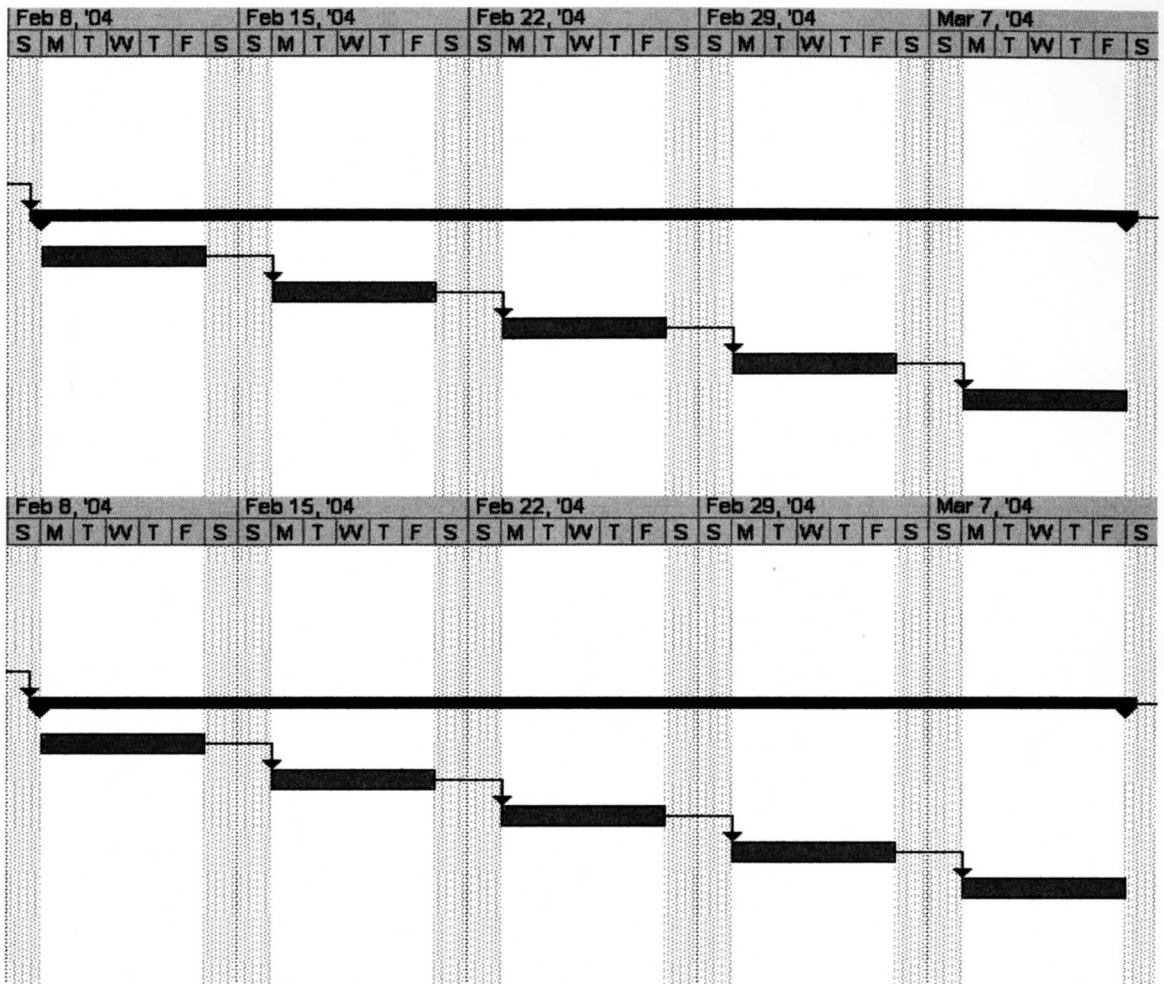
phpBB.com :: Creating Communities. (n.d.) Retrieved March 17, 2004, from <http://www.phpbb.com>

Phorum: PHP Message Board. (n.d.) Retrieved March 17, 2004, from <http://www.phorum.org>

A. Appendix A – Project Plan

❶	Task Name	Duration	Start	Finish	Predecessors
☐	Create user flow design	5 days	Mon 1/12/04	Fri 1/16/04	
☐	Create system architecture diagram	5 days	Mon 1/19/04	Fri 1/23/04	1
☐	Create E-R diagram for DB	5 days	Mon 1/26/04	Fri 1/30/04	2
	Create database using E-R diagram	5 days	Mon 2/2/04	Fri 2/6/04	3
☐	Create system pages	25 days	Mon 2/9/04	Fri 3/12/04	4
	Create System Administration	5 days	Mon 2/9/04	Fri 2/13/04	
	Create User Administration	5 days	Mon 2/16/04	Fri 2/20/04	6
	Create Forum Administration	5 days	Mon 2/23/04	Fri 2/27/04	7
	Create Content Administration	5 days	Mon 3/1/04	Fri 3/5/04	8
	Create Default System Forum	5 days	Mon 3/8/04	Fri 3/12/04	9
☐	Create example forums	3 days	Mon 3/15/04	Wed 3/17/04	5
	Create Gateway forum	1 day	Mon 3/15/04	Mon 3/15/04	
	Create MSIS forum	1 day	Tue 3/16/04	Tue 3/16/04	12
	Create literature forum	1 day	Wed 3/17/04	Wed 3/17/04	13
☐	Complete Project Documentation	5 days	Thu 3/18/04	Wed 3/24/04	11
	Write introduction	1 day	Thu 3/18/04	Thu 3/18/04	
	Write statement of problem	1 day	Fri 3/19/04	Fri 3/19/04	16
	Write objectives	1 day	Mon 3/22/04	Mon 3/22/04	17
	Write scope of project	1 day	Tue 3/23/04	Tue 3/23/04	18
	Write results and conclusions	1 day	Wed 3/24/04	Wed 3/24/04	19





B. Appendix B – Source Code

1. Utility Files

db_config.php

```
<?php
@ $db = mysql_pconnect('localhost','forumUser','forumuser');

if (!$db)
{
    echo 'Error: Could not connect to database. Please try again later.';
    exit;
}

mysql_select_db('myforum',$db);

?>
```

security.php

```
<?php
session_start();
if (!isset($_SESSION['logged_in']))
{
    header("Location: http://localhost/myForum/login.php");
    exit;
}
else
{
    $user_query = "select * from tblUser where UserID = " . $_SESSION['userid'];
    $user_result = mysql_query($user_query);
    $user_info = mysql_fetch_array($user_result);
}

if (isset($admin_check))
{
    if ($user_info['SecurityID'] < 5)
    {
        header("Location: http://localhost/myForum/login.php");
        exit;
    }
}

?>
```

forms.js

```
function CheckTextFieldNotEmpty(objField, strError)
//RECEIVES: objField - the object to check
//          strError - error message to display if the field is empty
//RETURNS: true if the field is not empty. otherwise false
//PURPOSE: checks a text form field to see if the value is empty. if so it puts focus
//          on the field and displays an alert error message.
{
    if (objField.value == "")
    {
```

```

        objField.focus();
        alert (strError);
        return false;
    }
    return true;
}

function CheckSelectFieldNotEmpty(objField, strError)
//PRE: the first item in the select must have a value of 0
//RECEIVES: objField - the object to check
//      strError - error message to display if the field is empty
//RETURNS: true if the field is not empty. otherwise false
//PURPOSE: checks a select form field to see if the value is set to 0. if so it puts focus
//      on the field and displays an alert error message.
{
    if (objField.value == 0)
    {
        objField.focus();
        alert (strError);
        return false;
    }
    return true;
}

function CheckInputLength(ObjField, strLength, strError)
//RECEIVES: Input string from form, desired length of string, error string
//RETURNS: False if string is shorter than desired length, otherwise true
{
    if (ObjField.value.length < strLength)
    {
        alert(strError);
        ObjField.focus();
        return false;
    }
    return true;
}

function CheckEmailPattern(ObjField, strError)
//RECEIVES: Form field string and error string
//RETURNS: True if field looks like an email address and is not "none@none.com," else false
{
    pattern = ^w+@\S+\.\w+$/i;
    if ((ObjField.value.search(pattern) == -1) || (ObjField.value == "none@none.com"))
    {
        alert(strError);
        ObjField.focus();
        return false;
    }
    return true;
}

function CheckTextPattern(ObjField,regPattern,strError)
//RECEIVES: Text field to validate, regular expression pattern to validate against, error string to return
//RETURNS: True if the field does not contain the pattern or excluded characters, false if it does

```

```

{
    pattern = regPattern;
    if (ObjField.value.search(pattern) != -1)
    {
        alert(strError);
        ObjField.focus();
        return false;
    }
    return true;
}

```

2. Include Files

system_header.inc

```

<style>
a {text-decoration:none;color:purple}
a:hover {text-decoration:none;color:orange}
</style>
<center>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
  <TR>
    <TD COLSPAN=2>
      <IMG SRC="images/sys_logo.gif" WIDTH=84 HEIGHT=67 ALT=""></TD>
    <TD COLSPAN=8><h1>Oh-Nonymous Forums</h1></TD>
  </TR>
</TABLE>

```

system_nav.inc

```

<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
  <TR>
    <TD width="100" align="left"><a href="default.php">Main Page</a></TD>
    <TD width="100" align="left"><a href="forums.php">Forum Main</a></TD>
    <TD width="100" align="left"><a href="search.php">Search Forums</a></TD>
    <?php
      if (isset($user_info))
      {
        if ($user_info['SecurityID'] == 5)
        {
          ?>
          <TD><a href="adm_default.php">Administration Main</a></TD>
          <TD><a href="adm_user.php">User Admin</a></TD>
          <TD><a href="adm_forum.php">Forum Admin</a></TD>
          <TD><a href="adm_skin.php">Skin Admin</a></TD>
          <TD><a href="adm_filters.php">Filter Admin</a></TD>
          <?php
            }
          }
        ?>
      }
    </TR>
</TABLE>

```

system_footer.inc

```

<br><br>
<table width="800">
  <TR>

```

```

                <TD align="center">Copyright 2004, Oh-Nonymous Forums</TD>
            </TR>
        </TABLE>
    </center>

```

msis_header.inc

```

<style>
a {text-decoration:none;color:green}
a:hover {text-decoration:none;color:red}
</style>
<center>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
    <TR>
        <TD COLSPAN=2>
            <IMG SRC="images/ms_logo.gif" WIDTH=84 HEIGHT=67 ALT=""></TD>
        <TD COLSPAN=8><h1>MSIS Project Forums</h1></TD>
    </TR>
</TABLE>

```

msis_nav.inc

```

<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
    <TR>
        <TD width="100" align="left"><a href="default.php">Main Page</a></TD>
        <TD width="100" align="left"><a href="forums.php">Forum Main</a></TD>
        <TD width="100" align="left"><a href="search.php">Search Forums</a></TD>
        <?php
            if (isset($user_info))
            {
                if ($user_info['SecurityID'] == 5)
                {
                    ?>
                    <TD><a href="adm_default.php">Administration Main</a></TD>
                    <TD><a href="adm_user.php">User Admin</a></TD>
                    <TD><a href="adm_forum.php">Forum Admin</a></TD>
                    <TD><a href="adm_skin.php">Skin Admin</a></TD>
                    <TD><a href="adm_filters.php">Filter Admin</a></TD>
                    <?php
                        }
                    }
                }
            }
        </TR>
</TABLE>

```

msis_footer.inc

```

<br><br>
<table width="800">
    <TR>
        <TD align="center">Copyright 2004, Dan Ohnemus</TD>
    </TR>
</TABLE>
</center>

```

lit_header.inc

```

<style>
a {text-decoration:none;color:green}
a:hover {text-decoration:none;color:red}

```

```

</style>
<center>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
  <TR>
    <TD COLSPAN=2>
      <IMG SRC="images/lit_logo.gif" WIDTH=84 HEIGHT=67 ALT=""></TD>
    <TD COLSPAN=8><h1>Literary Society Book Forums</h1></TD>
  </TR>
</TABLE>

```

lit_nav.inc

```

<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
  <TR>
    <TD width="100" align="left"><a href="default.php">Main Page</a></TD>
    <TD width="100" align="left"><a href="forums.php">Forum Main</a></TD>
    <TD width="100" align="left"><a href="search.php">Search Forums</a></TD>
    <?php
      if (isset($user_info))
      {
        if ($user_info['SecurityID'] == 5)
        {
          ?>
          <TD><a href="adm_default.php">Administration Main</a></TD>
          <TD><a href="adm_user.php">User Admin</a></TD>
          <TD><a href="adm_forum.php">Forum Admin</a></TD>
          <TD><a href="adm_skin.php">Skin Admin</a></TD>
          <TD><a href="adm_filters.php">Filter Admin</a></TD>
          <?php
            }
          }
        ?>
      }
    </TR>
</TABLE>

```

lit_footer.inc

```

<br><br>
<table width="800">
  <TR>
    <TD align="center">Copyright 2004, SmallTown Literary Society</TD>
  </TR>
</TABLE>
</center>

```

comp_header.inc

```

<style>
a {text-decoration:none;color:blue}
a:hover {text-decoration:none;color:yellow}
</style>
<center>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
  <TR>
    <TD COLSPAN=2>
      <IMG SRC="images/cc_logo.gif" WIDTH=84 HEIGHT=67 ALT=""></TD>
    <TD COLSPAN=8><h1>Computer Company, Inc. Forums</h1></TD>
  </TR>
</TABLE>

```

comp_nav.inc

```

<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
  <TR>
    <TD width="100" align="left"><a href="default.php">Main Page</a></TD>
    <TD width="100" align="left"><a href="forums.php">Forum Main</a></TD>
    <TD width="100" align="left"><a href="search.php">Search Forums</a></TD>
    <?php
      if (isset($user_info))
      {
        if ($user_info['SecurityID'] == 5)
        {
          ?>
          <TD><a href="adm_default.php">Administration Main</a></TD>
          <TD><a href="adm_user.php">User Admin</a></TD>
          <TD><a href="adm_forum.php">Forum Admin</a></TD>
          <TD><a href="adm_skin.php">Skin Admin</a></TD>
          <TD><a href="adm_filters.php">Filter Admin</a></TD>
          <?php
            }
          }
        }
      }
    ?>
  </TR>
</TABLE>

```

comp_footer.inc

```

<br><br>
<table width="800">
  <TR>
    <TD align="center">Copyright 2004, Computer Company, Inc.</TD>
  </TR>
</TABLE>
</center>

```

3. System files

add_filters.php

```

<HTML>
<HEAD>
<TITLE>Oh-Nonymous Forums: Filter Add/Edit</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<script language="javascript" src="includes/forms.js"></script>
<script language="javascript">

function validate()
{
  if (!CheckTextFieldNotEmpty(document.filter_edit.stopword, "You must enter a stopword."))
return false;
  if (!CheckTextFieldNotEmpty(document.filter_edit.replacement, "You must enter a
replacement.")) return false;
  document.forms['filter_edit'].submit();
}

```

```

</script>
<?php
$admin_check = true;
require("includes/db_config.php");
require("includes/security.php");
require("includes/system_header.inc");
require("includes/system_nav.inc");
if ($_GET['verb'] != "new")
{
    $query = "select * from tblFilters where FilterID = " . $_GET["filterid"];
    $result = mysql_query($query);
    $row = mysql_fetch_array($result);
}

?>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
<TD>
<h2>Add/Edit Filter</h2>
<br>
<table width="400">
<form name="filter_edit" method="post" action="commit_filters.php">
<?php
if ($_GET['verb'] == "new")
{
    echo '<tr><td>Stopword</td><td><input type="text" name="stopword" size="22"></td></tr>';
    echo '<tr><td>Replacement</td><td><input type="text" name="replacement"
size="22"></td></tr>';
    echo '<tr><td colspan="2"><input type="button" onclick="validate();" value="Save
Changes">&nbsp;  <input type="reset"></td></tr>';
}
else
{
    echo '<input type="hidden" name="filterid" value="' . stripslashes($row['FilterID']) . "'>';
    echo '<tr><td>Stopword</td><td><input type="text" name="stopword" size="20" value="' .
stripslashes($row['Stopword']) . "'></td></tr>';
    echo '<tr><td>Replacement</td><td><input type="text" name="replacement" size="20" value="' .
stripslashes($row['Replacement']) . "'></td></tr>';
    echo '<tr><td colspan="2"><input type="button" onclick="validate();" value="Save
Changes">&nbsp;  <input type="reset"></td></tr>';
}
?>
</form>
</table>
</TD>
</TR>
</table>
<?php
require("includes/system_footer.inc");
?>
</BODY>
</HTML>

add_forum.php
<HTML>
<HEAD>

```



```

<TITLE>Oh-Nonymous Forums: Forum Add/Edit</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<script language="javascript" src="includes/forms.js"></script>
<script language="javascript">

function validate()
{
    if (!CheckTextFieldNotEmpty(document.forum_edit.forumname, "You must enter a Forum
Name.)) return false;
    document.forms['forum_edit'].submit();
}

</script>
<?php
$admin_check = true;
require("includes/db_config.php");
require("includes/security.php");
require("includes/system_header.inc");
require("includes/system_nav.inc");
if ($_GET['verb'] != "new")
{
    $query = "select * from tblForum where ForumID = " . $_GET["forumid"];
    $result = mysql_query($query);
    $row = mysql_fetch_array($result);
}
$query2 = "select * from tblSkin order by skinid";
$result2 = mysql_query($query2);
$num_skin = mysql_num_rows($result2);

?>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
<TD>
<br>
<h2>Add/Edit Forum</h2>
<table width="400">
<form name="forum_edit" method="post" action="commit_forum.php">
<?php
if ($_GET['verb'] == "new")
{
    echo '<tr><td>Forum Name</td><td><input type="text" name="forumname"
size="22"></td></tr>';
    echo '<tr><td>Forum Skin</td>';
    echo '<td><select name="skinID">';
    for ($i=0; $i < $num_skin; $i++)
    {
        $row2 = mysql_fetch_array($result2);
        echo '<option value="' . stripslashes($row2['SkinID']) . "'>' .
stripslashes($row2['SkinName']) . '</option>';
    }
    echo '</select></td></tr>';
    echo '<tr><td colspan="2"><input type="button" onclick="validate();" value="Save
Changes">&nbsp;  <input type="reset"></td></tr>';
}
}

```

```

else
{
    echo '<input type="hidden" name="forumid" value="'. stripslashes($row['ForumID']) . "'>';
    echo '<tr><td>Forum Name</td><td><input type="text" name="forumname" size="20" value="'.
stripslashes($row['ForumName']) . "'></td></tr>';
    echo '<tr><td>Forum Skin</td>';
    echo '<td><select name="skinID">';
    for ($i=0; $i < $num_skin; $i++)
    {
        $row2 = mysql_fetch_array($result2);
        echo '<option value="'. stripslashes($row2['SkinID']) . "'";
        if (stripslashes($row2['SkinID']) == stripslashes($row['SkinID']))
        {
            echo ' selected';
        }
        echo '>' . stripslashes($row2['SkinName']) . '</option>';
    }
    echo '</select></td></tr>';
    echo '<tr><td colspan="2"><input type="button" onclick="validate();" value="Save
Changes">&nbsp;  <input type="reset"></td></tr>';
}
?>
</form>
</table>
</TD>
</TR>
</table>
<?php
require("includes/system_footer.inc");
?>
</BODY>
</HTML>

```

add_forum_users.php

```

<HTML>
<HEAD>
<TITLE>Oh-Nonymous Forums: Forum User Add/Remove</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<?php
$admin_check = true;
require("includes/db_config.php");
require("includes/security.php");
require("includes/lit_header.inc");
require("includes/lit_nav.inc");

?>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
<TD>
<br>
<h2>Add/Edit Forum Users</h2>
<B>Current Forum Users</B>
<table width="600">

```

```

<tr>
<td>First Name</td>
<td>Last Name</td>
<td></td>
</tr>
<?php
$current_query = "Select u.UserID, u.FName, u.LName from tblUser u join tblUserForum f on u.userid =
f.userid where f.ForumID = " . $_GET['forumid'];
$current_results = mysql_query($current_query);
$current_num = mysql_num_rows($current_results);
if ($current_num == 0)
{
    echo '<tr><td colspan="3">No users are currently assigned to this forum and it can only be
viewed by system administrators.</td></tr>';
}
else
{
    for ($i=0; $i < $current_num; $i++)
    {
        $current_row = mysql_fetch_array($current_results);
        echo '<tr>';
        echo '<td>' . $current_row['FName'] . '</td>';
        echo '<td>' . $current_row['LName'] . '</td>';
        echo '<td><a href="commit_forum_users.php?verb=remove&userid=' .
$current_row['UserID'] . '&forumid=' . $_GET['forumid'] . "'>Remove User</a></td>';
        echo '</tr>';
    }
}
?>
</form>
</table>
<br><br><B>Potential Forum Users</B>
<table width="400">
<tr>
<td>First Name</td>
<td>Last Name</td>
<td></td>
</tr>
<?php
$user_query = "Select * from tblUser";
$user_result = mysql_query($user_query);
$user_num = mysql_num_rows($user_result);
if ($user_num == 0)
{
    echo '<tr><td colspan="3">No users are available to assign to this forum.</td></tr>';
}
else
{
    for ($j=0; $j < $user_num; $j++)
    {
        $user_row = mysql_fetch_array($user_result);
        echo '<tr>';
        echo '<td>' . $user_row['FName'] . '</td>';
        echo '<td>' . $user_row['LName'] . '</td>';
    }
}
}

```

```

                echo '<td><a href="commit_forum_users.php?verb=add&userid=' . $user_row['UserID']
        . '&forumid=' . $_GET['forumid'] . '">Add User</a></td>';
                echo '</tr>';
        }
}

?>
</form>
</table>
</TD>
</TR>
</table>
<?php
require("includes/lit_footer.inc");
?>
</BODY>
</HTML>

```

add_skin.php

```

<HTML>
<HEAD>
<TITLE>Oh-Nonymous Forums: Skin Add/Edit</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<script language="javascript" src="includes/forms.js"></script>
<script language="javascript">

function validate()
{
    if (!CheckTextFieldNotEmpty(document.skin_edit.skinname, "You must enter a Skin Name. "))
return false;
    if (!CheckTextFieldNotEmpty(document.skin_edit.headercode, "You must enter a location for
Header Code. ")) return false;
    if (!CheckTextFieldNotEmpty(document.skin_edit.navcode, "You must enter a location for Nav
Code. ")) return false;
    if (!CheckTextFieldNotEmpty(document.skin_edit.footercode, "You must enter a location for
Footer Code. ")) return false;
    document.forms['skin_edit'].submit();
}

</script>
<?php
$admin_check = true;
require("includes/db_config.php");
require("includes/security.php");
require("includes/system_header.inc");
require("includes/system_nav.inc");
if ($_GET['verb'] != "new")
{
    $query = "select * from tblSkin where SkinID = " . $_GET["skinid"];
    $result = mysql_query($query);
    $row = mysql_fetch_array($result);
}

?>

```

```

<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
<TD>
<br>
<h2>Add/Edit Skin</h2>
<table width="400">
<form name="skin_edit" method="post" action="commit_skin.php">
<?php
if ($_GET['verb'] == "new")
{
    echo '<tr><td>Skin Name</td><td><input type="text" name="skinname" size="22"></td></tr>';
    echo '<tr><td>Header Code</td><td><input type="text" name="headercode"
size="22"></td></tr>';
    echo '<tr><td>Nav Code</td><td><input type="text" name="navcode" size="22"></td></tr>';
    echo '<tr><td>Footer Code</td><td><input type="text" name="footercode"
size="22"></td></tr>';
    echo '<tr><td colspan="2"><input type="button" onclick="validate();" value="Save
Changes">&nbsp;<input type="reset"></td></tr>';
}
else
{
    echo '<input type="hidden" name="skinid" value="'. stripslashes($row['SkinID']) . "'>';
    echo '<tr><td>Skin Name</td><td><input type="text" name="skinname" size="20" value="'.
stripslashes($row['SkinName']) . "'></td></tr>';
    echo '<tr><td>Header Code</td><td><input type="text" name="headercode" size="20" value="'.
stripslashes($row['HeaderCode']) . "'></td></tr>';
    echo '<tr><td>Nav Code</td><td><input type="text" name="navcode" size="20" value="'.
stripslashes($row['NavCode']) . "'></td></tr>';
    echo '<tr><td>Footer Code</td><td><input type="text" name="footercode" size="20" value="'.
stripslashes($row['FooterCode']) . "'></td></tr>';
    echo '<tr><td colspan="2"><input type="button" onclick="validate();" value="Save
Changes">&nbsp;<input type="reset"></td></tr>';
}
?>
</form>
</table>
</TD>
</TR>
</table>
<?php
require("includes/system_footer.inc");
?>
</BODY>
</HTML>

```

add_user.php

```

<HTML>
<HEAD>
<TITLE>Oh-Nonymous Forums: User Add/Edit</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<script language="javascript" src="includes/forms.js"></script>
<script language="javascript">

```

```
function validate()
```

```

{
    if (!CheckTextFieldNotEmpty(document.user_edit.username, "You must enter a username. "))
return false;
    if (!CheckTextFieldNotEmpty(document.user_edit.password, "You must enter a password. "))
return false;
    if (!CheckTextFieldNotEmpty(document.user_edit.fname, "You must enter a first name. ")) return
false;
    if (!CheckTextFieldNotEmpty(document.user_edit.lname, "You must enter a last name. ")) return
false;
    if (!CheckTextFieldNotEmpty(document.user_edit.email, "You must enter an email address. "))
return false;
    if (!CheckEmailPattern(document.user_edit.email, "Your email address appears to be invalid. "))
return false;
    if (!CheckTextFieldNotEmpty(document.user_edit.loggedip, "You must enter an initial IP
address. ")) return false;
        document.forms['user_edit'].submit();
}

```

</script>

<?php

\$admin_check = true;

require("includes/db_config.php");

require("includes/security.php");

require("includes/system_header.inc");

require("includes/system_nav.inc");

if (\$_GET['verb'] != "new")

{

 \$query = "select

u.userid,u.username,u.password,u.fname,u.lname,u.email,u.loggedip,u.securityid,s.securityname from

tblUser u join tblSecurity s on u.securityid = s.securityid where userid = " . \$_GET["userid"];

 \$result = mysql_query(\$query);

 \$row = mysql_fetch_array(\$result);

}

\$query2 = "select * from tblSecurity order by securityid";

\$result2 = mysql_query(\$query2);

\$num_security = mysql_num_rows(\$result2);

?>

<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>

<TR>

<TD>

<h2>Add/Edit User</h2>

<table width="400">

<form name="user_edit" method="post" action="commit_user.php">

<?php

if (\$_GET['verb'] == "new")

{

 echo '<tr><td>User Name</td><td><input type="text" name="username" size="22"></td></tr>';

 echo '<tr><td>Password</td><td><input type="password" name="password"

size="22"></td></tr>';

 echo '<tr><td>First Name</td><td><input type="text" name="fname" size="22"></td></tr>';

 echo '<tr><td>Last Name</td><td><input type="text" name="lname" size="22"></td></tr>';

 echo '<tr><td>Email</td><td><input type="text" name="email" size="35"></td></tr>';

 echo '<tr><td>Security Level</td>';

```

echo '<td><select name="securityID">';
for ($i=0; $i < $num_security; $i++)
{
    $row2 = mysql_fetch_array($result2);
    echo '<option value="" . stripslashes($row2['SecurityID']) . "">' .
stripslashes($row2['SecurityName']) . '</option>';
}
echo '</select></td></tr>';
echo '<tr><td>Logged IP</td><td><input type="text" name="loggedip" size="22"></td></tr>';
echo '<tr><td colspan="2"><input type="button" onclick="validate();" value="Save
Changes">&nbsp;<input type="reset"></td></tr>';
}
else
{
    echo '<input type="hidden" name="userid" value="" . stripslashes($row['userid']) . "">';
    echo '<tr><td>User Name</td><td><input type="text" name="username" size="20" value="" .
stripslashes($row['username']) . ""></td></tr>';
    echo '<tr><td>Password</td><td><input type="password" name="password" size="20" value="" .
stripslashes($row['password']) . ""></td></tr>';
    echo '<tr><td>First Name</td><td><input type="text" name="fname" size="20" value="" .
stripslashes($row['fname']) . ""></td></tr>';
    echo '<tr><td>Last Name</td><td><input type="text" name="lname" size="20" value="" .
stripslashes($row['lname']) . ""></td></tr>';
    echo '<tr><td>Email</td><td><input type="text" name="email" size="35" value="" .
stripslashes($row['email']) . ""></td></tr>';
    echo '<tr><td>Security Level</td>';
    echo '<td><select name="securityID">';
    for ($i=0; $i < $num_security; $i++)
    {
        $row2 = mysql_fetch_array($result2);
        echo '<option value="" . stripslashes($row2['SecurityID']) . ""';
        if (stripslashes($row2['SecurityID']) == stripslashes($row['securityid']))
        {
            echo ' selected';
        }
        echo '>' . stripslashes($row2['SecurityName']) . '</option>';
    }
    echo '</select></td></tr>';
    echo '<tr><td>Logged IP</td><td><input type="text" name="loggedip" size="20" value="" .
stripslashes($row['loggedip']) . ""></td></tr>';
    echo '<tr><td colspan="2"><input type="button" onclick="validate();" value="Save
Changes">&nbsp;<input type="reset"></td></tr>';
}
?>
</form>
</table>
</TD>
</TR>
</table>
<?php
require("includes/system_footer.inc");
?>
</BODY>
</HTML>

```

adm_default.php

```
<HTML>
<HEAD>
<TITLE>Oh-Nonymous Forums</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<?php
$admin_check = true;
require("includes/db_config.php");
require("includes/security.php");
require("includes/system_header.inc");
require("includes/system_nav.inc");
?>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
<TD>
<br>
<h2>Oh-Nonymous Forums Administration Main Page</h2>
<p><B>Please Select from the available options below, <?php echo $user_info['FName'] . " " .
$user_info['LName']; ?>:</B></p>
<?php
echo '<p><B>Administration</B><br>';
echo '<p><a href="adm_user.php">User Administration</a><br>';
echo '<a href="adm_forum.php">Forum Administration</a><br>';
echo '<a href="adm_skin.php">Skin Administration</a><br>';
echo '<a href="adm_filter.php">Filter Administration</a><br>';

?>
</TD>
</TR>
</table>
<?php
require("includes/system_footer.inc");
?>
</BODY>
</HTML>
```

adm_filters.php

```
<HTML>
<HEAD>
<TITLE>Oh-Nonymous Forums: Filter Administration</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<?php
$admin_check = true;
require("includes/db_config.php");
require("includes/security.php");
require("includes/system_header.inc");
require("includes/system_nav.inc");
?>
<br>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
<TD>
```



```

<h2>Filter Administration</h2>
<?php

$query = "select * from tblFilters";
$result = mysql_query($query);
$num_results = mysql_num_rows($result);

?>
<table width=400 border=0 cellpadding=0 cellspacing=0>
<tr>
<td></td>
<td></td>
<td><a href="add_filters.php?verb=new">Add Filter</a>&nbsp;&nbsp;&nbsp;<td>
</tr>
<tr>
<td><B>Stopword</B></td>
<td><B>Replacement</B></td>
<td></td>
</tr>
<?php
if ($num_results == 0)
{
    echo '<tr><td colspan="3" align="center">There are currently no filters in the
database.</td></tr>';
}
else
{
    for ($i=0; $i < $num_results; $i++)
    {
        $row = mysql_fetch_array($result);
        echo "<tr>";
        echo "<td>" . stripslashes($row['Stopword']) . "</td>";
        echo "<td>" . stripslashes($row['Replacement']) . "</td>";
        echo '<td><a href="add_filters.php?verb=edit&filterid=' . stripslashes($row['FilterID'])
.'">Edit Filter</a></td>';
        echo "</tr>";
    }
}

?>

</table>
</TD>
</TR>
</table>
<br><br>
<?php
require("includes/system_footer.inc");
?>
</BODY>
</HTML>

```

```

adm_forum.php
<HTML>
<HEAD>
<TITLE>Oh-Nonymous Forums: Forum Administration</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<?php
$admin_check = true;
require("includes/db_config.php");
require("includes/security.php");
require("includes/system_header.inc");
require("includes/system_nav.inc");
?>
<br>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
<TD>
<h2>Forum Administration</h2>
<?php

$query = "select * from tblForum";
$result = mysql_query($query);
$num_results = mysql_num_rows($result);

?>
<table width=600 border=0 cellpadding=0 cellspacing=0>
<tr>
<td></td>
<td></td>
<td></td>
<td><a href="add_forum.php?verb=new">Add Forum</a>&nbsp;&nbsp;&nbsp;<td>
</tr>
<tr>
<td><B>Forum ID</B></td>
<td><B>Forum Name</B></td>
<td></td>
<td></td>
</tr>
<?php
if ($num_results == 0)
{
    echo '<tr><td colspan="4" align="center">There are currently no forums in the
database.</td></tr>';
}
else
{
    for ($i=0; $i < $num_results; $i++)
    {
        $row = mysql_fetch_array($result);
        echo "<tr>";
        echo "<td>" . stripslashes($row['ForumID']) . "</td>";
        echo "<td>" . stripslashes($row['ForumName']) . "</td>";
        echo '<td><a href="add_forum.php?verb=edit&forumid=' . stripslashes($row['ForumID'])
.'">Edit Forum</a></td>';
    }
}
}

```

```

                echo '<td><a href="add_forum_users.php?forumid='. stripslashes($row['ForumID'])
.'">Edit Forum Users</a></td>';
                echo "</tr>";
            }
        }
    }
?>

</table>
</TD>
</TR>
</table>
<br><br>
<?php
require("includes/system_footer.inc");
?>
</BODY>
</HTML>

```

adm_skin.php

```

<HTML>
<HEAD>
<TITLE>Oh-Nonymous Forums: Skin Administration</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<?php
$admin_check = true;
require("includes/db_config.php");
require("includes/security.php");
require("includes/system_header.inc");
require("includes/system_nav.inc");
?>
<br>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
<TD>
<h2>Skin Administration</h2>
<?php

$query = "select * from tblSkin";
$result = mysql_query($query);
$num_results = mysql_num_rows($result);

?>
<table width=800 border=0 cellpadding=0 cellspacing=0>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><a href="add_skin.php?verb=new">Add Skin</a>&nbsp;&nbsp;&nbsp;<td>
</tr>
<tr>
<td><B>Skin ID</B></td>

```

```

<td><B>Skin Name</B></td>
<td><B>Header Code</B></td>
<td><B>Nav Code</B></td>
<td><B>Footer Code</B></td>
<td></td>
</tr>
<?php
if ($num_results == 0)
{
    echo '<tr><td colspan="6" align="center">There are currently no forums in the
database.</td></tr>';
}
else
{
    for ($i=0; $i < $num_results; $i++)
    {
        $row = mysql_fetch_array($result);
        echo "<tr>";
        echo "<td>" . stripslashes($row['SkinID']) . "</td>";
        echo "<td>" . stripslashes($row['SkinName']) . "</td>";
        echo "<td>" . stripslashes($row['HeaderCode']) . "</td>";
        echo "<td>" . stripslashes($row['NavCode']) . "</td>";
        echo "<td>" . stripslashes($row['FooterCode']) . "</td>";
        echo '<td><a href="add_skin.php?verb=edit&skinid=' . stripslashes($row['SkinID'])
.">Edit Skin</a></td>';
        echo "</tr>";
    }
}
?>

</table>
</TD>
</TR>
</table>
<br><br>
<?php
require("includes/system_footer.inc");
?>
</BODY>
</HTML>

```

adm_user.php

```

<HTML>
<HEAD>
<TITLE>Oh-Nonymous Forums: User Administration</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<?php
$admin_check = true;
require("includes/db_config.php");
require("includes/security.php");
require("includes/system_header.inc");
require("includes/system_nav.inc");
?>

```

```

<br>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
<TD>
<h2>User Administration</h2>
<?php

$query = "select userid,username,fname,lname,email,loggedip,securityname from tblUser join tblSecurity
on tblUser.securityid = tblSecurity.securityid";
$result = mysql_query($query);
$num_results = mysql_num_rows($result);

?>
<table width=800 border=0 cellpadding=0 cellspacing=0>
<tr><td align="right" colspan="8"><a href="add_user.php?verb=new">Add
User</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<td></tr>
<tr>
<td><B>User ID</B></td>
<td><B>User Name</B></td>
<td><B>First Name</B></td>
<td><B>Last Name</B></td>
<td><B>Email</B></td>
<td><B>Security Level</B></td>
<td><B>Logged IP</B></td>
<td></td>
</tr>
<?php
if ($num_results == 0)
{
    echo '<tr><td colspan="8" align="center">There are currently no users in the database.</td></tr>';
}
else
{
    for ($i=0; $i < $num_results; $i++)
    {
        $row = mysql_fetch_array($result);
        echo "<tr>";
        echo "<td>" . stripslashes($row['userid']) . "</td>";
        echo "<td>" . stripslashes($row['username']) . "</td>";
        echo "<td>" . stripslashes($row['fname']) . "</td>";
        echo "<td>" . stripslashes($row['lname']) . "</td>";
        echo "<td>" . stripslashes($row['email']) . "</td>";
        echo "<td>" . stripslashes($row['securityname']) . "</td>";
        echo "<td>" . stripslashes($row['loggedip']) . "</td>";
        echo '<td><a href="add_user.php?verb=edit&userid=' . stripslashes($row['userid'])
.'">Edit User</a>';
        echo "</tr>";
    }
}

?>

</table>
</TD>
</TR>
</table>

```

```

<br><br>
<?php
require("includes/system_footer.inc");
?>
</BODY>
</HTML>

```

commit_filters.php

```

<HTML>
<HEAD>
<TITLE>Oh-Nonymous Forums: Filter Add/Edit</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<?php
$admin_check = true;
require("includes/db_config.php");
require("includes/security.php");
require("includes/system_header.inc");
require("includes/system_nav.inc");

if (isset($_POST['filterid']))
{
    $query = "update tblFilters set stopword=" . addslashes($_POST['stopword']) . ",replacement=" .
addslashes($_POST['replacement']) . " where filterid=" . addslashes($_POST['filterid']);
    $result = mysql_query($query);
}
else
{
    $query = "insert into tblFilters (stopword,replacement) values (" .
addslashes($_POST['stopword']) . "," . addslashes($_POST['replacement']) . ")";
    $result = mysql_query($query);
}

?>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
<TD>
<br>
<h2>Add/Edit Filter</h2>

```

The following information was added or changed in the filter database:

```

<?php
echo '<p>Stopword: ' . $_POST['stopword'];
echo '<br>Replacement: ' . $_POST['replacement'];

?>
</TD>
</TR>
</table>
<?php
require("includes/system_footer.inc");
?>
</BODY>

```

```
</HTML>
```

commit_forum.php

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Oh-Nonymous Forums: Forum Add/Edit</TITLE>
```

```
</HEAD>
```

```
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0  
MARGINHEIGHT=0>
```

```
<?php
```

```
$admin_check = true;  
require("includes/db_config.php");  
require("includes/security.php");  
require("includes/system_header.inc");  
require("includes/system_nav.inc");
```

```
$$Skin_Query = "Select skinname from tblSkin where SkinID = " . $_POST['skinID'];  
$$Skin_Result = mysql_query($$Skin_Query);  
$$Skin_Rows = mysql_fetch_array($$Skin_Result);  
$$Skin_Name = $$Skin_Rows['skinname'];
```

```
if (isset($_POST['forumid']))
```

```
{
```

```
    $query = "update tblForum set forumname=" . addslashes($_POST['forumname']) . ",skinid=" .  
addslashes($_POST['skinID']) . " where forumid=" . addslashes($_POST['forumid']);  
    $result = mysql_query($query);
```

```
}
```

```
else
```

```
{
```

```
    $query = "insert into tblForum (forumname,skinid) values (" . addslashes($_POST['forumname'])  
    . ", " . addslashes($_POST['skinID']) . ")";  
    $result = mysql_query($query);
```

```
}
```

```
?>
```

```
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
```

```
<TR>
```

```
<TD>
```

```
<br>
```

```
<h2>Add/Edit Forum</h2>
```

The following information was added or changed in the forum database:

```
<?php
```

```
echo '<p>Forum Name: ' . $_POST['forumname'];  
echo '<br>Skin Name: ' . $$Skin_Name;
```

```
?>
```

```
</TD>
```

```
</TR>
```

```
</table>
```

```
<?php
```

```
require("includes/system_footer.inc");
```

```
?>
```

```
</BODY>
```

</HTML>

commit_forum_users.php

<HTML>

<HEAD>

<TITLE>Oh-Nonymous Forums: Forum User Add/Edit</TITLE>

</HEAD>

<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>

<?php

```
$admin_check = true;
require("includes/db_config.php");
require("includes/security.php");
require("includes/system_header.inc");
require("includes/system_nav.inc");
```

```
$forum_query = "select * from tblForum where forumid = " . $_GET['forumid'];
$forum_results = mysql_query($forum_query);
$forum_info = mysql_fetch_array($forum_results);
```

```
$FUser_query = "select * from tblUser where userid = " . $_GET['userid'];
$FUser_results = mysql_query($FUser_query);
$FUser_Info = mysql_fetch_array($FUser_results);
```

```
$user_added = false;
```

```
if($_GET['verb'] == "add")
```

```
{
    $check_query = "select * from tblUserForum where forumid = " . $_GET['forumid'] . " and userid
= " . $_GET['userid'];
    $check_results = mysql_query($check_query);
    $check_user = mysql_num_rows($check_results);
```

```
    if($check_user > 0)
```

```
    {
        $user_added = true;
    }
```

```
    else
```

```
    {
        $query = "insert into tblUserForum (userid,forumid) values (" . $_GET['userid'] . "," .
$_GET['forumid'] . ")";
        $result = mysql_query($query);
    }
```

```
    }
    if($_GET['verb'] == "remove")
```

```
{
    $query = "delete from tblUserForum where userid = " . $_GET['userid'] . " and forumid=" .
$_GET['forumid'];
    $result = mysql_query($query);
}
```

?>

<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>

<TR>

<TD>

<h2>Add/Edit Forum User</h2>

```
<?php
if ($_GET['verb'] == "add")
{
    if ($user_added == true)
    {
        echo '<p>The following user was NOT added to the ' . $forum_info['ForumName'] . '
forum: ';
        echo '<p>First Name: ' . $User_Info['FName'];
        echo '<br>Last Name: ' . $User_Info['LName'];
        echo '<br>The user was already associated with the selected forum.';
    }
    else
    {
        echo '<p>The following user was added to the ' . $forum_info['ForumName'] . ' forum: ';
        echo '<p>First Name: ' . $User_Info['FName'];
        echo '<br>Last Name: ' . $User_Info['LName'];
    }
}
if ($_GET['verb'] == "remove")
{
    echo '<p>The following user was removed from the ' . $forum_info['ForumName'] . ' forum: ';
    echo '<p>First Name: ' . $User_Info['FName'];
    echo '<br>Last Name: ' . $User_Info['LName'];
}
?>
</TD>
</TR>
</table>
<?php
require("includes/system_footer.inc");
?>
</BODY>
</HTML>
```

commit_message.php

```
<HTML>
<HEAD>
<TITLE>Oh-Nonymous Forums: Message Add/Edit</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<?php
require("includes/db_config.php");
require("includes/security.php");
require("includes/system_header.inc");
require("includes/system_nav.inc");

$filter_query = "select * from tblFilters";
$filter_result = mysql_query($filter_query);
$filter_num = mysql_num_rows($filter_result);

if (isset($_POST['parentid']))
{
```

```

$title = $_POST['title'];
$body = $_POST['body'];

if ($filter_num > 0)
{
    for ($i=0; $i < $filter_num; $i++)
    {
        $filter_words = mysql_fetch_array($filter_result);
        $title =
str_replace($filter_words['Stopword'],$filter_words['Replacement'],$title);
        $body =
str_replace($filter_words['Stopword'],$filter_words['Replacement'],$body);
    }
}

$query = "insert into tblMessage (title,body,forumid,parentid,userid,lastedit,active) values (" .
addslashes($title) . "," . addslashes($body) . "," . addslashes($_POST['forumid']) . "," .
addslashes($_POST['parentid']) . "," . addslashes($_SESSION['userid']) . "," . date("Y-m-d H:i:s") . ",1)";
$result = mysql_query($query);
}
else if (isset($_POST['messageid']))
{
    $title = $_POST['title'];
    $body = $_POST['body'];

    if ($filter_num > 0)
    {
        for ($i=0; $i < $filter_num; $i++)
        {
            $filter_words = mysql_fetch_array($filter_result);
            $title =
str_replace($filter_words['Stopword'],$filter_words['Replacement'],$title);
            $body =
str_replace($filter_words['Stopword'],$filter_words['Replacement'],$body);
        }
    }

    $query = "update tblMessage set title=" . addslashes($title) . ",body=" . addslashes($body)
    . ",editid=" . $_SESSION['userid'] . ",lastedit=" . date("Y-m-d H:i:s") . " where messageid=" .
    addslashes($_POST['messageid']);
    $result = mysql_query($query);
}
else
{
    $title = $_POST['title'];
    $body = $_POST['body'];

    if ($filter_num > 0)
    {
        for ($i=0; $i < $filter_num; $i++)
        {
            $filter_words = mysql_fetch_array($filter_result);
            $title =
str_replace($filter_words['Stopword'],$filter_words['Replacement'],$title);
            $body =
str_replace($filter_words['Stopword'],$filter_words['Replacement'],$body);
        }
    }
}

```

```

    }
}

$query = "insert into tblMessage (title,body,forumid,parentid,userid,lastedit,active) values (" .
addslashes($title) . "," . addslashes($body) . "," . addslashes($_POST['forumid']) . ",0," .
addslashes($_SESSION['userid']) . "," . date('Y-m-d H:i:s') . ",1)";
$result = mysql_query($query);
}

```

```

?>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
<TD>
<br>
<h2>Message Saved</h2>

```

The following information was added or changed in the message database:

```

<?php
echo "<p>Title: " . $_POST['title'];
echo "<br>Body: " . $_POST['body'];

?>
</TD>
</TR>
</table>
<?php
require("includes/system_footer.inc");
?>
</BODY>
</HTML>

```

commit_register.php

```

<HTML>
<HEAD>
<TITLE>Oh-Nonymous Forums: User Add/Edit</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<?php
require("includes/db_config.php");
//require("includes/security.php");
require("includes/system_header.inc");
require("includes/system_nav.inc");

```

```

$Security_Query = "Select securityname from tblSecurity where securityID = " . $_POST['securityID'];
$Security_Result = mysql_query($Security_Query);
$Security_Rows = mysql_fetch_array($Security_Result);
$Security_Name = $Security_Rows['securityname'];

```

```

$query = "insert into tblUser (username,password,fname,lname,email,securityid,loggedip) values (" .
addslashes($_POST['username']) . "," . addslashes($_POST['password']) . "," .
addslashes($_POST['fname']) . "," . addslashes($_POST['lname']) . "," . addslashes($_POST['email']) . "," .
addslashes($_POST['securityID']) . "," . addslashes($_POST['loggedip']) . ")";

```

```

$result = mysql_query($query);

$query_user = "select * from tblUser where username = " . $_POST['username'] . """;
$result_user = mysql_query($query_user);
$row_user = mysql_fetch_array($result_user);

$forum_query = "select * from tblForum where ForumName = 'User Forum'";
$forum_result = mysql_query($forum_query);
$forum_row = mysql_fetch_array($forum_result);

$insert_query = "insert into tblUserForum (forumid,userid) values (" . $forum_row['ForumID'] . "," .
$row_user['UserID'] . ")";
$insert_result = mysql_query($insert_query);

```

```

?>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
<TD>
<br>
<h2>Register</h2>

```

The following information was added or changed in the user database:

```

<?php
echo '<p>User Name: ' . $_POST['username'];
echo '<br>First Name: ' . $_POST['fname'];
echo '<br>Last Name: ' . $_POST['lname'];
echo '<br>Email: ' . $_POST['email'];
echo '<br>Security Level: ' . $Security_Name;
echo '<br>Logged IP: ' . $_POST['loggedip'];

echo '<p>You may now <A HREF="login.php">log in</A> using your new username and password.';

```

```

?>
</TD>
</TR>
</table>
<?php
require("includes/system_footer.inc");
?>
</BODY>
</HTML>

```

commit_search.php

```

<HTML>
<HEAD>
<TITLE>Oh-Nonymous Forums: Search Forums</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<?php
require("includes/db_config.php");
require("includes/security.php");
require("includes/system_header.inc");
require("includes/system_nav.inc");

```

```

?>
<br>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
<TD>
<h2>Search Forums</h2>
<?php

if ($_GET['forumid'] == "all")
{
    $query = "select * from tblMessage where body like '%" . $_GET['query'] . "%' or title like '%" .
$_GET['query'] . "%'";
}
else
{
    $query = "select * from tblMessage where (body like '%" . $_GET['query'] . "%' or title like '%" .
$_GET['query'] . "%') and forumid = " . $_GET['forumid'];
}

$result = mysql_query($query);
$num_results = mysql_num_rows($result);

if ($num_results > 0)
{
    echo "<b>One or more matches to your search term were found in the following
messages:</b><hr>";
    for ($i=0; $i < $num_results; $i++)
    {
        $row = mysql_fetch_array($result);
        $parent = $row['ParentID'];
        if ($row['ParentID'] == 0)
        {
            $parent = $row['MessageID'];
        }
        echo $row['Title'] . '&nbsp;(<a href="messages.php?parentid=' . $row['MessageID'] .
">View Message</a>);
        echo '&nbsp;(<a href="messages.php?parentid=' . $parent . ">View Thread</a><hr>';
    }
}
else
{
    echo "Your search did not return any results.";
}

?>

</TD>
</TR>
</table>
<br><br>
<?php
require("includes/system_footer.inc");
?>
</BODY>
</HTML>

```

commit_skin.php

```
<HTML>
<HEAD>
<TITLE>Oh-Nonymous Forums: User Add/Edit</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<?php
$admin_check = true;
require("includes/db_config.php");
require("includes/security.php");
require("includes/system_header.inc");
require("includes/system_nav.inc");

if (isset($_POST['skinid']))
{
    $query = "update tblSkin set skinname=" . addslashes($_POST['skinname']) . ",headercode=" .
addslashes($_POST['headercode']) . ",navcode=" . addslashes($_POST['navcode']) . ",footercode=" .
addslashes($_POST['footercode']) . " where skinid=" . addslashes($_POST['skinid']);
    $result = mysql_query($query);
}
else
{
    $query = "insert into tblSkin (skinname,headercode,navcode,footercode) values (" .
addslashes($_POST['skinname']) . "," . addslashes($_POST['headercode']) . "," .
addslashes($_POST['navcode']) . "," . addslashes($_POST['footercode']) . ")";
    $result = mysql_query($query);
}

?>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
<TD>
<br>
<h2>Add/Edit Skin</h2>

The following information was added or changed in the skin database:

<?php

echo '<p>Skin Name: ' . $_POST['skinname'];
echo '<br>Header Code: ' . $_POST['headercode'];
echo '<br>Nav Code: ' . $_POST['navcode'];
echo '<br>Footer Code: ' . $_POST['footercode'];

?>
</TD>
</TR>
</table>
<?php
require("includes/system_footer.inc");
?>
</BODY>
```

```
</HTML>
```

```
commit_user.php
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Oh-Nonymous Forums: User Add/Edit</TITLE>
```

```
</HEAD>
```

```
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0  
MARGINHEIGHT=0>
```

```
<?php
```

```
$admin_check = true;  
require("includes/db_config.php");  
require("includes/security.php");  
require("includes/system_header.inc");  
require("includes/system_nav.inc");
```

```
$Security_Query = "Select securityname from tblSecurity where securityID = " . $_POST['securityID'];  
$Security_Result = mysql_query($Security_Query);  
$Security_Rows = mysql_fetch_array($Security_Result);  
$Security_Name = $Security_Rows['securityname'];
```

```
if (isset($_POST['userid']))
```

```
{
```

```
    $query = "update tblUser set username=" . addslashes($_POST['username']) . ",password=" .  
addslashes($_POST['password']) . ",fname=" . addslashes($_POST['fname']) . ",lname=" .  
addslashes($_POST['lname']) . ",email=" . addslashes($_POST['email']) . ",securityid=" .  
addslashes($_POST['securityID']) . ",loggedip=" . addslashes($_POST['loggedip']) . " where userid=" .  
addslashes($_POST['userid']);  
    $result = mysql_query($query);
```

```
}
```

```
else
```

```
{
```

```
    $query = "insert into tblUser (username,password,fname,lname,email,securityid,loggedip) values  
(" . addslashes($_POST['username']) . "," . addslashes($_POST['password']) . "," .  
addslashes($_POST['fname']) . "," . addslashes($_POST['lname']) . "," . addslashes($_POST['email']) . "," .  
addslashes($_POST['securityID']) . "," . addslashes($_POST['loggedip']) . ")";  
    $result = mysql_query($query);
```

```
}
```

```
?>
```

```
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
```

```
<TR>
```

```
<TD>
```

```
<br>
```

```
<h2>Add/Edit User</h2>
```

The following information was added or changed in the user database:

```
<?php
```

```
echo '<p>User Name: ' . $_POST['username'];  
echo '<br>First Name: ' . $_POST['fname'];  
echo '<br>Last Name: ' . $_POST['lname'];  
echo '<br>Email: ' . $_POST['email'];  
echo '<br>Security Level: ' . $Security_Name;  
echo '<br>Logged IP: ' . $_POST['loggedip'];
```

```
?>
</TD>
</TR>
</table>
<?php
require("includes/system_footer.inc");
?>
</BODY>
</HTML>
```

default.php

```
<HTML>
<HEAD>
<TITLE>Oh-Nonymous Forums</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<?php
require("includes/db_config.php");
require("includes/security.php");
require("includes/system_header.inc");
require("includes/system_nav.inc");
?>
<br>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
<TD>
<h2>Oh-Nonymous Forums Main Page</h2>
<p><B>Please Select from the available options below, <?php echo $user_info['FName'] . " " .
$user_info['LName']; ?>:</B></p>
<p><B>Forum Options</B><br>
<?php
echo '<a href="forums.php">View Forums</a><br>';
echo '<a href="search.php">Search Forums</a><br>';
if ($user_info['SecurityID'] == 5)
{
echo '<P><B>Administration</B><br>';
echo '<a href="adm_default.php">Administration Main</a><br>';
echo '<a href="adm_user.php">User Administration</a><br>';
echo '<a href="adm_forum.php">Forum Administration</a><br>';
echo '<a href="adm_skin.php">Skin Administration</a><br>';
echo '<a href="adm_filter.php">Filter Administration</a><br>';
}
?>
</TD>
</TR>
</table>
<?php
require("includes/system_footer.inc");
?>
</BODY>
</HTML>
```


detail.php

```
<HTML>
<HEAD>
<TITLE>Oh-Nonymous Forums: Message Add/Edit</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<script language="javascript" src="includes/forms.js"></script>
<script language="javascript">

function validate()
{
    if (!CheckTextFieldNotEmpty(document.message_edit.title, "You must provide a title.)) return
false;
    if (!CheckTextFieldNotEmpty(document.message_edit.body, "You must provide a message
body.)) return false;

        document.forms['message_edit'].submit();
}

</script>
<?php
require("includes/db_config.php");
require("includes/security.php");

if (isset($_GET['forumid']))
{
    $skin_forum = $_GET['forumid'];
}
else
{
    $skin_result = mysql_query("select forumid from tblmessage where messageid = " .
$_GET['messageid']);
    $skin_row = mysql_fetch_array($skin_result);
    $skin_forum = $skin_row['forumid'];
}

$forum_query = "select f.ForumName,s.HeaderCode,s.NavCode,s.FooterCode from tblForum f join
tblSkin s on f.skinid = s.skinid where ForumID = " . $skin_forum;
$forum_result = mysql_query($forum_query);
$forum_name = mysql_fetch_array($forum_result);

require($forum_name['HeaderCode']);
require($forum_name['NavCode']);

if ($_GET['verb'] != "new")
{
    $query = "select * from tblMessage where MessageID = " . $_GET["messageid"];
    $result = mysql_query($query);
    $row = mysql_fetch_array($result);
}

?>
<br>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
```

```

<TD>
<?php
if ($_GET['verb'] == "new")
{
    echo '<h2>Start New Thread</h2>';
}
else if ($_GET['verb'] == "edit")
{
    echo '<h2>Edit Message</h2>';
    echo '<b>' . $row['Title'] . '</b><br>';
}
else
{
    echo '<h2>Reply to Message</h2>';
    echo '<b>' . $row['Title'] . '</b><br>';
}
?>
<table width="800">
<form name="message_edit" method="post" action="commit_message.php">
<?php
if ($_GET['verb'] == "new")
{
    echo '<input type="hidden" name="forumid" value="' . $_GET['forumid'] . '">';
    echo '<tr><td>Title</td><td><input type="text" name="title" size="80"></td></tr>';
    echo '<tr><td valign="top">Body</td><td><textarea name="body" cols="60"
rows="20"></textarea></td></tr>';
    echo '<tr><td></td><td><input type="button" onclick="validate();" value="Save
Changes">&nbsp;<input type="reset"></td></tr>';
}
else if ($_GET['verb'] == "edit")
{
    echo '<input type="hidden" name="messageid" value="' . stripslashes($row['MessageID']) . '">';
    echo '<input type="hidden" name="forumid" value="' . stripslashes($row['ForumID']) . '">';
    echo '<tr><td>Title</td><td><input type="text" name="title" size="80" value="' .
stripslashes($row['Title']) . '"></td></tr>';
    echo '<tr><td valign="top">Body</td><td><textarea name="body" cols="60" rows="20">' .
stripslashes($row['Body']) . '</textarea></td></tr>';
    echo '<tr><td></td><td><input type="button" onclick="validate();" value="Save
Changes">&nbsp;<input type="reset"></td></tr>';
}
else
{
    $ParentID = stripslashes($row['ParentID']);
    if (stripslashes($row['ParentID']) == "0")
    {
        $ParentID = stripslashes($row['MessageID']);
    }
    echo '<input type="hidden" name="parentid" value="' . $ParentID . '">';
    echo '<input type="hidden" name="forumid" value="' . stripslashes($row['ForumID']) . '">';
    echo '<tr><td>Title</td><td><input type="text" name="title" size="80" value="RE: ' .
stripslashes($row['Title']) . '"></td></tr>';
    echo '<tr><td valign="top">Body</td><td><textarea name="body" cols="60"
rows="20"></textarea></td></tr>';
    echo '<tr><td></td><td><input type="button" onclick="validate();" value="Save
Changes">&nbsp;<input type="reset"></td></tr>';
}
}

```

```
?>
</TD>
</TR>
</table>
<?php
require("includes/system_footer.inc");
?>
</BODY>
</HTML>
```

default.php

```
<HTML>
<HEAD>
<TITLE>Oh-Nonymous Forums</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<?php
require("includes/db_config.php");
require("includes/security.php");
require("includes/system_header.inc");
require("includes/system_nav.inc");
?>
<br>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
<TD>
<h2>Oh-Nonymous Forums Main Page</h2>
<p><B>Please Select from the available options below, <?php echo $user_info['FName'] . " " .
$user_info['LName']; ?>:</B></p>
<p><B>Forum Options</B><br>
<?php
echo '<a href="forums.php">View Forums</a><br>';
echo '<a href="search.php">Search Forums</a><br>';
if ($user_info['SecurityID'] == 5)
{
echo '<P><B>Administration</B><br>';
echo '<a href="adm_default.php">Administration Main</a><br>';
echo '<a href="adm_user.php">User Administration</a><br>';
echo '<a href="adm_forum.php">Forum Administration</a><br>';
echo '<a href="adm_skin.php">Skin Administration</a><br>';
echo '<a href="adm_filter.php">Filter Administration</a><br>';
}
?>
</TD>
</TR>
</table>
<?php
require("includes/system_footer.inc");
?>
</BODY>
</HTML>
```

detail.php

```
<HTML>
<HEAD>
<TITLE>Oh-Nonymous Forums: Message Add/Edit</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<script language="javascript" src="includes/forms.js"></script>
<script language="javascript">

function validate()
{
    if (!CheckTextFieldNotEmpty(document.message_edit.title, "You must provide a title.)) return
false;
    if (!CheckTextFieldNotEmpty(document.message_edit.body, "You must provide a message
body.)) return false;

        document.forms['message_edit'].submit();
}

</script>
<?php
require("includes/db_config.php");
require("includes/security.php");

if (isset($_GET['forumid']))
{
    $skin_forum = $_GET['forumid'];
}
else
{
    $skin_result = mysql_query("select forumid from tblmessage where messageid = " .
$_GET['messageid']);
    $skin_row = mysql_fetch_array($skin_result);
    $skin_forum = $skin_row['forumid'];
}

$forum_query = "select f.ForumName,s.HeaderCode,s.NavCode,s.FooterCode from tblForum f join
tblSkin s on f.skinid = s.skinid where ForumID = " . $skin_forum;
$forum_result = mysql_query($forum_query);
$forum_name = mysql_fetch_array($forum_result);

require($forum_name['HeaderCode']);
require($forum_name['NavCode']);

if ($_GET['verb'] != "new")
{
    $query = "select * from tblMessage where MessageID = " . $_GET["messageid"];
    $result = mysql_query($query);
    $row = mysql_fetch_array($result);
}

?>
<br>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
```

```

<TD>
<?php
if ($_GET['verb'] == "new")
{
    echo '<h2>Start New Thread</h2>';
}
else if ($_GET['verb'] == "edit")
{
    echo '<h2>Edit Message</h2>';
    echo '<b>' . $row['Title'] . '</b><br>';
}
else
{
    echo '<h2>Reply to Message</h2>';
    echo '<b>' . $row['Title'] . '</b><br>';
}
?>
<table width="800">
<form name="message_edit" method="post" action="commit_message.php">
<?php
if ($_GET['verb'] == "new")
{
    echo '<input type="hidden" name="forumid" value="'. $_GET['forumid'] . '">';
    echo '<tr><td>Title</td><td><input type="text" name="title" size="80"></td></tr>';
    echo '<tr><td valign="top">Body</td><td><textarea name="body" cols="60"
rows="20"></textarea></td></tr>';
    echo '<tr><td></td><td><input type="button" onclick="validate();" value="Save
Changes">&nbsp;<input type="reset"></td></tr>';
}
else if ($_GET['verb'] == "edit")
{
    echo '<input type="hidden" name="messageid" value="'. stripslashes($row['MessageID']) . '">';
    echo '<input type="hidden" name="forumid" value="'. stripslashes($row['ForumID']) . '">';
    echo '<tr><td>Title</td><td><input type="text" name="title" size="80" value="'.
stripslashes($row['Title']) . '"></td></tr>';
    echo '<tr><td valign="top">Body</td><td><textarea name="body" cols="60" rows="20">'.
stripslashes($row['Body']) . '</textarea></td></tr>';
    echo '<tr><td></td><td><input type="button" onclick="validate();" value="Save
Changes">&nbsp;<input type="reset"></td></tr>';
}
else
{
    $ParentID = stripslashes($row['ParentID']);
    if (stripslashes($row['ParentID']) == "0")
    {
        $ParentID = stripslashes($row['MessageID']);
    }
    echo '<input type="hidden" name="parentid" value="'. $ParentID . '">';
    echo '<input type="hidden" name="forumid" value="'. stripslashes($row['ForumID']) . '">';
    echo '<tr><td>Title</td><td><input type="text" name="title" size="80" value="RE: '.
stripslashes($row['Title']) . '"></td></tr>';
    echo '<tr><td valign="top">Body</td><td><textarea name="body" cols="60"
rows="20"></textarea></td></tr>';
    echo '<tr><td></td><td><input type="button" onclick="validate();" value="Save
Changes">&nbsp;<input type="reset"></td></tr>';
}
}

```

```
?>
</form>
</table>
</TD>
</TR>
</table>
<?php
require($forum_name['FooterCode']);
?>
</BODY>
</HTML>
```

forums.php

```
<HTML>
<HEAD>
<TITLE>Oh-Nonymous Forums: Forum Main</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<?php
require("includes/db_config.php");
require("includes/security.php");
require("includes/system_header.inc");
require("includes/system_nav.inc");
?>
<br>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
<TD>
<h2>Forum Main</h2>
<p>Please choose from one of the available forums below. If a forum you wish to view is not shown
below, please contact the forum administrator to request addition to the user group for that forum.
<?php

if ($user_info['SecurityID'] == 5)
{
    $query = "select * from tblForum";
}
else
{
    $query = "select * from tblForum f join tblUserForum u on f.forumid=u.forumid where u.userid="
. $_SESSION['userid'];
}

$result = mysql_query($query);
$num_results = mysql_num_rows($result);

?>

<p><B>Available Forums</B>
<hr>
<?php
if ($num_results == 0)
{
    echo 'There are currently no forums available to you.</td></tr>';
}
}
```



```

<td><B>Skin Name</B></td>
<td><B>Header Code</B></td>
<td><B>Nav Code</B></td>
<td><B>Footer Code</B></td>
<td></td>
</tr>
<?php
if ($num_results == 0)
{
    echo '<tr><td colspan="6" align="center">There are currently no forums in the
database.</td></tr>';
}
else
{
    for ($i=0; $i < $num_results; $i++)
    {
        $row = mysql_fetch_array($result);
        echo "<tr>";
        echo "<td>" . stripslashes($row['SkinID']) . "</td>";
        echo "<td>" . stripslashes($row['SkinName']) . "</td>";
        echo "<td>" . stripslashes($row['HeaderCode']) . "</td>";
        echo "<td>" . stripslashes($row['NavCode']) . "</td>";
        echo "<td>" . stripslashes($row['FooterCode']) . "</td>";
        echo '<td><a href="add_skin.php?verb=edit&skinid=' . stripslashes($row['SkinID'])
.'">Edit Skin</a></td>';
        echo "</tr>";
    }
}
?>

</table>
</TD>
</TR>
</table>
<br><br>
<?php
require("includes/system_footer.inc");
?>
</BODY>
</HTML>

```

adm_user.php

```

<HTML>
<HEAD>
<TITLE>Oh-Nonymous Forums: User Administration</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<?php
$admin_check = true;
require("includes/db_config.php");
require("includes/security.php");
require("includes/system_header.inc");
require("includes/system_nav.inc");
?>

```



```

{
    $username = $_POST['username'];
    $pass = $_POST['password'];

    $query = "select * from tblUser where UserName='".$username.'" and Password='".$pass.'";
    $result = mysql_query($query);
    $num_rows = mysql_num_rows($result);

    if ($num_rows > 0)
    {
        $row = mysql_fetch_array($result);
        $_SESSION['logged_in'] = $username;
        $_SESSION['userid'] = $row['UserID'];
        $IP_Query = "Update tblUser set LoggedIP = " . getenv("REMOTE_ADDR") . " where
UserID = " . $_SESSION['userid'];
        $IP_Result = mysql_query($IP_Query);
    }
}

?>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
<TD>
<h2>Login Page</h2>
<?php
if (isset($_SESSION['logged_in']))
{
    echo 'You may proceed to the <a href="default.php">Main Page</a> now or use any of the
navigation options at the top of the screen.';
    echo '<p>If you were already logged in an were brought to this page, you may have attempted to
view a page you are not authorized to view.';
}
else
{
?>
<form name="login" method="post" action="login.php">
Username: <input type="text" name="username"><br>
Password: <input type="password" name="password"><br>
<input type="button" onclick="validate();" value="Log In">
</form>
<p>If you are not a current user of this forum system and would like to register a new guest account, <A
HREF="register.php">click here</A>.
<?php
}
?>
</TD>
</TR>
</table>
<?php
require("includes/system_footer.inc");
?>
</BODY>
</HTML>

```

messages.php

```

<HTML>
<HEAD>
<TITLE>Oh-Nonymous Forums: Messages Main</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<?php
require("includes/db_config.php");
require("includes/security.php");

$query = "select * from tblMessage where Active=1 and (MessageID=" . $_GET['parentid'] . " or
ParentID=" . $_GET['parentid'] . ") order by MessageID";

$parent_query = "select Title from tblMessage where MessageID = " . $_GET['parentid'];
$parent_result = mysql_query($parent_query);
$parent_name = mysql_fetch_array($parent_result);

$result = mysql_query($query);
$result2 = mysql_query($query);
$num_results = mysql_num_rows($result);
$row2 = mysql_fetch_array($result2);

$skin_query = "select f.ForumName,s.HeaderCode,s.NavCode,s.FooterCode from tblForum f join tblSkin s
on f.skinid = s.skinid where ForumID = " . $row2['ForumID'];
$skin_result = mysql_query($skin_query);
$skin_name = mysql_fetch_array($skin_result);

require($skin_name['HeaderCode']);
require($skin_name['NavCode']);

$forum_result = mysql_query("Select * from tblforum where ForumName='User Forum'");
$forum_row = mysql_fetch_array($forum_result);

?>
<br>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
<TD>
<p><B>All messages related to

<?php
echo $parent_name['Title'] . "</B><hr>";
if ($num_results == 0)
{
    echo 'There are currently no messages associated with this message.</td></tr>'; //This will never,
    ever happen.
}
else
{
    for ($i=0; $i < $num_results; $i++)
    {
        $row = mysql_fetch_array($result);
        echo "<b>" . $row['Title'] . "</b>";
        echo '&nbsp;&nbsp;&nbsp;';
    }
}
}

```

```

        if ($user_info['SecurityID'] > 3)
        {
            echo('<a href="detail.php?verb=edit&messageid=' . $row['MessageID'] .
">Edit</a>)&nbsp;&nbsp; ';
        }
        if ($user_info['SecurityID'] > 1 || $forum_row['ForumName'] == "User Forum")
        {
            echo '<a href="detail.php?verb=reply&messageid=' . $row['MessageID'] .
">Reply</a>';
        }
        $result2 = mysql_query("select * from tbluser where userid=" . $row['UserID']);
        $message_name = mysql_fetch_array($result2);
        echo "<br>Posted by: " . $message_name['FName'] . " " . $message_name['LName'];
        echo "<br>Last edited: " . $row['LastEdit'];
        echo "<br><br>" . $row['Body'] . "<br>";
        echo '<hr>';
    }
}
?>

```

```

</TD>
</TR>
</table>
<br><br>
<?php
require("includes/system_footer.inc");
?>
</BODY>
</HTML>

```

register.php

```

<HTML>
<HEAD>
<TITLE>Oh-Nonymous Forums: Register New Guest Account</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<script language="javascript" src="includes/forms.js"></script>
<script language="javascript">

```

```

function validate()
{
    if (!CheckTextFieldNotEmpty(document.user_edit.username, "You must enter a username.))
return false;
    if (!CheckTextFieldNotEmpty(document.user_edit.password, "You must enter a password.))
return false;
    if (!CheckTextFieldNotEmpty(document.user_edit.fname, "You must enter a first name.)) return
false;
    if (!CheckTextFieldNotEmpty(document.user_edit.lname, "You must enter a last name.)) return
false;
    if (!CheckTextFieldNotEmpty(document.user_edit.email, "You must enter an email address.))
return false;
    if (!CheckEmailPattern(document.user_edit.email, "Your email address appears to be invalid.))
return false;
}

```

```

        document.forms['user_edit'].submit();
    }

</script>
<?php
require("includes/db_config.php");
//require("includes/security.php");
require("includes/system_header.inc");
require("includes/system_nav.inc");

?>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
<TD>
<br>
<h2>Register New Guest Account</h2>
<table width="400">
<form name="user_edit" method="post" action="commit_register.php">
<?php
    echo '<tr><td>User Name</td><td><input type="text" name="username" size="22"></td></tr>';
    echo '<tr><td>Password</td><td><input type="password" name="password"
size="22"></td></tr>';
    echo '<tr><td>First Name</td><td><input type="text" name="fname" size="22"></td></tr>';
    echo '<tr><td>Last Name</td><td><input type="text" name="lname" size="22"></td></tr>';
    echo '<tr><td>Email</td><td><input type="text" name="email" size="35"></td></tr>';
    echo '<input type="hidden" name="securityID" value="1">';
    echo '<input type="hidden" name="loggedip" value="" . getenv("REMOTE_ADDR") . "">';
    echo '<tr><td colspan="2"><input type="button" onclick="validate();" value="Save
Changes">&nbsp;<input type="reset"></td></tr>';
?>
</form>
</table>
</TD>
</TR>
</table>
<?php
require("includes/system_footer.inc");
?>
</BODY>
</HTML>

search.php
<HTML>
<HEAD>
<TITLE>Oh-Nonymous Forums: Search Forums</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0 MARGINWIDTH=0
MARGINHEIGHT=0>
<script language="javascript" src="includes/forms.js"></script>
<script language="javascript">

function validate()
{
    if (!CheckTextFieldNotEmpty(document.search.query, "You must enter a search term. ")) return
false;
    document.forms['search'].submit();
}

```

```

}

</script>
<?php
require("includes/db_config.php");
require("includes/security.php");
require("includes/system_header.inc");
require("includes/system_nav.inc");
?>
<br>
<TABLE WIDTH=800 BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR>
<TD>
<h2>Search Forums</h2>
<form name="search" method="get" action="commit_search.php">
<?php

if ($user_info['SecurityID'] == 5)
{
    $query = "select * from tblForum";
}
else
{
    $query = "select * from tblForum f join tblUserForum u on f.forumid=u.forumid where u.userid="
. $_SESSION['userid'];
}

$result = mysql_query($query);
$num_results = mysql_num_rows($result);

echo 'Search Term: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="text" name="query" size="30">';
echo '<br>Choose Forum: <select name="forumid"><option value="all">All Forums</option>';
for ($i=0; $i < $num_results; $i++)
{
    $row = mysql_fetch_array($result);
    echo '<option value="' . $row['ForumID'] . '>' . $row['ForumName'] . '</option>';
}
echo '</select>';
echo '<br><br><input type="button" onclick="validate();" value="Search">&nbsp;&nbsp;<input type="reset">';

?>
</form>

</TD>
</TR>
</table>
<br><br>
<?php
require("includes/system_footer.inc");
?>
</BODY>
</HTML>

```



```
}  
}
```

```
?>
```

```
</TD>
```

```
</TR>
```

```
</table>
```

```
<br><br>
```

```
<?php
```

```
require($forum_name['FooterCode']);
```

```
?>
```

```
</BODY>
```

```
</HTML>
```