Project Submission - Status Report

What has been completed:

The final web site structure has finally been integrated into the web site manager. Users are now provided with a user account main page where they can access all account related features. The adminstrator main page has also been implemented to assist with with account management.

Account management features such as account deletion, permission list manipulation, etc. were implemented. The tech lead successfully ran all the test cases outlined in the "Requirements Test Cases" documents; therefore we have proven that the functional requirements have all been met.

Lastly, links and buttons were added as necessary to web pages to increase connectivity between pages and allow for more user-friendly navigation of the Web Site Manager. In sum, the system is ready for deployment and meets all functional requirements. Combined with the user documentation, the system is user-friendly enough for open use.

What is to be done next:

There still needs to be the addition of the ability for a professor-approved intermediary to manage the web page on his/her behalf. We have proposed an "alias" feature, which allows a professor to assign one alias for his account. An alias is an alternate combination of email address and password that can be used to access a professor's web site. A professor could provide this alias to an intermediary to work on the account.

Data fields for the professor's "title" and "college/department" need to be integrated into the database and the web page creation process.

We also need to conduct a more thorough test of error-checking in order to assure quality of user experience. And as pointed out by the implementors earlier, more security measures are needed in order to prevent PHP/mySQL hacks. Lastly, it may be worthwhile to integrate the Web Site Manager with the Kerberos login verification system

Issues:

Due to a lack of persistent functional requirements testing, the development team forgot to include a "title" and "college/department" field in the database for a professor's web page.

Since we ran out of development time we were unable to rigorously test and improve error checking and security measures. Error checking is currently not descriptive enough to inform the user of the specific type of error and how to remedy it and the system is still vulnerable to PHP/mySQL hacks. Thus, we cannot yet make any guarantees concerning the uptime of the system.

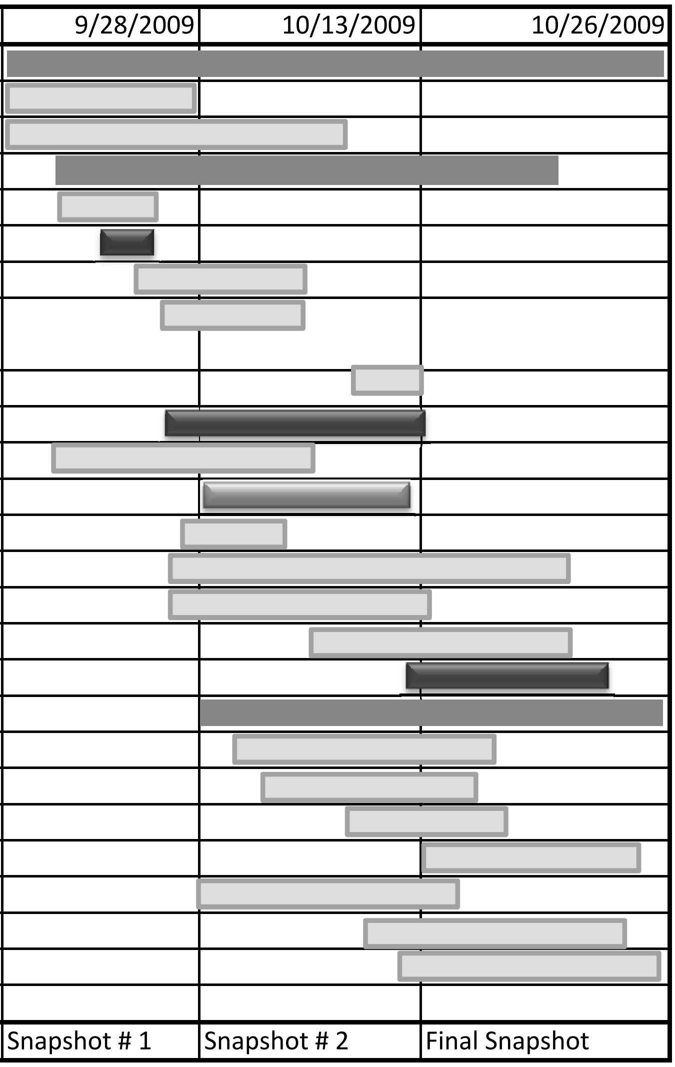
Finally, illness (swine flu and food poisoning in 2/3 of our implementator team) pushed back development of the "alias" system.

Prioritized List of Open Issues

1. Lack of a "title" and "college/department" field in the database for a professor's web page (this is a non-functional requirement we did not meet)
2. "Alias" system still needs to be implemented in order for professor's to allow another a person to work on their web site
3. Error checking is currently not descriptive enough to inform the user of the specific type of error and how to remedy it
4. System is still vulnerable to PHP/mySQL hacks
5. The Web Site Manager is currently incapable of sending emails containing confirmations URLs: users must rely on manually inputting them immediately after registering their account.

Team Opal
Project 1
Schedule

#	Task Name	Duration	9/28/2009	10/13/2009	10/26/2009
1	**Professor's Webpage**				
2	Requirments				
3	Design				
4	**Programming**				
5	Database				
6	Test				
7	LogIn				
8	Account Creation/Verification				
9	Password Retrieval				
10	Test				
11	Admin Page				
12	Test				
13	Webpages Look and Feel				
14	Edit Pages				
15	Edit Information				
16	Error Checking				
17	Test				
18	**Testing (Test Cases?)**				
19	Account Creation				
20	Password Retrieval				
21	Editing				
22	Error Checking				
23	Admin Functionality				
24	Fix Defects				
25	Regression Test				
26					
			Snapshot # 1	Snapshot # 2	Final Snapshot



CS411 - Project 1

Lead Tasks				
Requirements Document + Tests	9/28/2009	done	Chris	See Document "Requirements and Tests"
User Documentation	9/28/2009	in progress	Nastia	See Document "User Documentation"
System Design	9/28/2009	done	Chris/Nastia	See Document "System Design"
Implementation Tasks				
1. Set up web-server and SQL	9/21/2009	done	Josiah	
2. Create login page (email)	9/26/2009	task revised	Josiah	Tech lead accepted revision of task to data retrieval only
3. Create Database	9/21/2009	done	Lincoln and James	
4. Library of Functions	9/26/2009	done	Lincoln and James	Data (entire record) retrieval function has been implemented as of Snapshot #1
5. Test	9/27/2009	done	Chris	Testing done with implementers and errors noted
--> Snap Shot # 1 - September 28th, 2009				
Lead Tasks				
Requirements Document + Tests	09/28/09	done	Chris	See Document "Requirements and Tests"
User Documentation		in progress	Nastia	See Document "User Documentation"
System Design	09/28/09	done	Chris/Nastia	See Document "System Design"
Implementation Tasks				
1. Verification of Account	09/25/09	done		Tech lead accepted revision of task to data retrieval only
2. Ability to Edit/Preview and Display Main Page	09/25/09	done,revised	Implementers	Tech lead accepted revision of task to display of selected fields and profile picture of records manually inserted using phpmyadmin
3. HTML Template Creation		in progress	Josiah	
Testing	09/25/09	done		Tech lead and implementer (Josiah) 9/25
--> Snap Shot # 2 - October 13th, 2009				
Lead Tasks				
Requirements Document + Tests	10/12/09	done	Chris	See Document "Requirements and Tests"
User Documentation	10/12/09	done	Nastia	See Document "User Documentation"
System Design	10/12/09	done	Chris/Nastia	See Document "System Design"
Implementation Tasks				
1. Editing and Optional Pages	10/08/09	done		
2. Ability to "Personalize" Pages		cancelled		This task was vetoed by the tech lead due to this feature not beeing required and the final deadline looming
3. Error Handling	10/08/09	done		

4. Testing	10/09/09			
5. Authentication of the Account	10/08/09	done	Lincoln	
6. Error Handling for Editing	10/08/09	done	James	
7. Error Handling for Authentication	10/08/09	done	Lincoln	
8. Ability to edit existing information	10/08/09	done	James	
9. HTML Template		done	Josiah	
--> Project Due - October 26th, 2009				
1. Create User Main Page	10/25/09	done		
2. Create Administrator Main Page	10/25/09	done		
3. Implement Admin. Functions	10/25/09	done		
4. Improve web-site flow	10/25/09	done		
5. Add “Title” "College/Dept." fields	10/25/09	incomplete		Absence went unnoticed until 10/26/09
6. Add aliases for access	10/25/09	incomplete		Swine flu pushed back development of feature

James' Work Log

Date Start	Time Start	Date Finished	Time Finished	Task	Detail on what was done	Detailed Issues
9/21/2009	2:30:00	9/21/2009	4:30:00	Research	Read MyAQL tutorial and located PHP guide for future reference	
9/24/2009	4:30:00	9/24/2009	5:45:00	Meeting	Discussed implementation details, explained how to use the webserver, took minutes	
9/25/2009	9:30:00	9/25/2009	10:00:00	Email / Minutes	Edited and emailed minutes along with summary for upcoming meeting between Josiah and Tech lead	
10/6/2009	8:30:00	10/6/2009	11:00:00	Form for data entry	Downloaded the database from opal webserver and incorporated it into WAMP, called Josiah re: connection issues, researched HTML, PHP, and MySQL for potential solutions and general learning, began coding in earnest	HTML form for data entry completed with relevant variable names for input boxes, template for PHP functions instantiated for rapid prototyping in the future, consideration of error checking algorithm
10/8/2009	2:15:00	10/8/2009	7:15:00	Form for data entry	Coded entire edit page and error checking, allows you to update the database if logged in	Includes error checking and automatic data retrieval on entry page

10/25/2009	12:30:00	10/25/2009	3:05:00	Fixing issues	Removed 'login successful' echo, removed the ability to edit password hint and password hint answer, added ability to change name	Did not add functionality to error check ending date of course

Josiah's Work Log

Date Start	Time Start	Date Finished	Time Finished	Task	Detail on what was done	Detailed Issues
2009.09.20	16:00:00	2009.09.20	17:00:00	Set up webserver	FTP, MySQL	login info in email
2009.09.23	20:00:00	2009.09.23	21:00:00	Create basic design template on paper	make consistent with Studentlink Scheme?	on paper...need to create in dreamweaver/html?
2009.09.25	15:00:00	2009.09.25	16:00:00	Prepare code for demo to Christian	created a different version of test.php that Lincoln submitted - templated	
2009.09.25	16:00:00	2009.09.25		Demo to Christian		

Lincoln's Work Log

Date Start	Time Start	Date Finished	Time Finished	Task	Detail on what was done	Detailed Issues
10/6/2009	9:00:00	10/6/2009	12:30:00	authentication	created authentication files and simple test cases	
10/7/2009	20:00:00	10/8/2009	0:00:00	authentication, error handling	added error handling, change password, optimized code	

Product Overview

Boston University (BU) is the fourth-largest private university in the United States, with a student population of more than 30,000 students and nearly 4,000 faculty members. All faculty members are expected to maintain up-to-date web pages with relevant information such as their contact information, courses taught, and curriculum vitae. These faculty websites serve as valuable resources for students, fellow faculty, and those outside of the BU community who may wish to know more about the official and academic capacities of each faculty member.

However, given the large number of faculty and the varying amount of web design experience and time available amongst faculty members, it is difficult for all faculty to maintain their web pages to a common standard. Thus, it behooves the university to create a user-friendly system where faculty may create their academic web pages with minimal input. With an easy-to-use application to create and maintain their academic web pages, faculty are much more likely to keep their pages up to date and accessible for all interested parties.

The aim of the "The Professor's Web Site Manager" web application is to provide such a user-friendly web page creation system to Boston University. The web application will be a complete solution that will allow faculty to easily create web pages with a uniform layout, and a BU administrator to manage these websites quickly and efficiently. Using this tool, faculty will be able to create web pages containing the following non-optional information:

- University logo
- Recent studio photo
- Full name
- Title
- College Department
- Contact information (email address, office location, office phone number)
- Curriculum Vitae
- Courses currently taught
- Office hours
- A short bio

Additionally, a professor may choose to include:

- Research
- Publications
- Upcoming events (lectures, symposiums, etc.)
- Ph D Students
- Links (to lab website, etc)
- Awards
- Organizations, groups

More detailed information on the specific functionality of this application can be found in the Functional Requirements section. A narrative detailing the envisioned usage of this application can be found in the Usage Scenario section.

Usage Scenario

The following usage scenario describes how one might use "The Professor's Web Site Manager" (hereafter referred to as the "WSM" or "manager") to create and manage a faculty website. The user is assumed to be a professor of Boston University.

The user wants to create a faculty web page and begins with no account created in the web site manager database. The user is referred by the Boston University IT department to the login page of the WSM. The ***Login*** *page contains two text fields in which a login (BU email address) and password can be entered, as well as two links: the first is labelled "Password Retrieval" and the second is labelled "Create a New Account". The user clicks the latter link.*

The "Create a New Account" link leads to a ***Registration*** *page. The user inputs his email and clicks a submit button. However, the email address was mistyped and an error message returns stating that "No such email address listed in the faculty directory." Realizing his error, the user re-types his email address correctly and clicks the submit button once again. This time, a notification message appears that informs the user to check the submitted email address for a temporary password to log into the newly created account with.*

After a brief wait, the user receives the email and retrieves the password. The user is now able to successfully access the WSM from the ***Login*** *page and starts at the* ***User Control Panel****. Following instructions detailed in the prior email, the user navigates to the* ***Account Update Page*** *and immediately changes his password to his own liking. The user also sets a Password Retrieval Question and a hint for this question The user then navigates the* ***Content Edit Page*** *but does not intend to actually create the web site yet. After looking over the page and getting a feel for the interface the user accidentally clicks the submit button on the edit page despite having only filled in his full name. An error message is returned informing the user that although the data has been saved and the page updated, that certain non-optional data fields such as courses taught, office hours etc. have yet to be filled.*

Not wanting to leave the page unfinished, the user clicks a link on the edit page that allows him to view the initial state of the website. He then takes note of any changes or additions he'd like to make his website and implements them using the edit page. After editing, he views the changes made to website. By shuttling back and forth between the viewing page and edit page the user is able to modify the web page to his or her satisfaction, with complete disclosure of all non-optional data.

During the course of the year the user does not update often and forgets his password. At the start of the next semester he attempts to login in order to update his courses taught and realizes that he does not remember his password. He clicks on the "Password Retrieval" button on the ***Login*** *page and enters in his email address. The WSM returns a message stating the Password Retrieval Question along with its hint. The user is able to remember the answer to the question and submits the correct answer. A notification message is shown which informs the user that the password has been emailed to their faculty email address. The user retrieves his password from the email and is able to login to the WSM once more and update their web site.*

Functional Requirements

This section provides a list of functional capabilities referenced in the “Usage Scenario” document. The prefix ‘(+)’ was added to each item as it was completed during implementation.

The "Professor's Web Site Manager" is a web application that shall have:

(+) The ability to set up an account

(+) The ability to create an initial web page with minimal input

(+) The ability to update the initial web page, including the ability to add or remove non-optional items.

The application will include an administrator's interface that shall have:

(+) The ability to list all existing accounts

(+) The ability to add/edit/delete members to/on/from a "permitted" list

(+) The ability to delete an account

**Current as of 10/26/09*

Requirements Test Cases

The following test cases described will be a step by step series of instructions for a user to follow in order to test functionality of the Web Site Manager (WSM). The pluses denote functional requirements that have been implemented and should be capable of being tested. Test cases written for requirements that have yet to be implemented are tentative and may have their procedure changed in the future.

(+) The ability to set up an account

1. Go to the Registration Page at <http://www.opal411.legendaryfox.net/reg.php>
2. Enter your pre-approved BU email address and click "Submit Query"
3. Assuming that no error occurred so far, access the confirmation URL that is displayed on the web page and emailed* to the user. **Because the server currently lacks email capability the user must copy-paste the confirmation URL from the page following submission of a pre-approved BU email address.*
4. Select and re-enter a password, then click "Submit Query". The user should now be redirected to the user account Edit Page.
5. Check creation of the account by clicking the "Logout" link on the top right, and then logging back in with the email address address and password combination that the user just registered.

(+) The ability to create an initial web page with minimal input

1. Access the Login Page at <http://www.opal411.legendaryfox.net/login.php>
2. Login to an existing user account.
3. From the user's Main Account page click on the "Edit your page" link.
4. At the Edit Page enter all required information. Required text fields have their titles bolded (such as **Name**, **Office Address**, etc.); it is not possible to submit data for a web page without filling each of these text fields first.
5. Click the "Submit" link at the bottom of the page when finished; if there are any errors with the data provided the user will be returned to the Edit Page with problematic text fields marked with a red asterisk. Once they have fixed these errors they may try to submit the Edit Page form again. If no red asterisks are present the form has been submitted successfully and the web page data has been updated.
6. Click on the link at the bottom of the page to return to the Main Account page. At the Main Account Page click on the "View Page" link in order to see the updated web page. A new window will popup in order to display this page. If the user intends on frequently alternating between updating and viewing their page they may simply refresh this popup window after successfully submitting changes in the Edit Page.

(+) The ability to update the initial web page, including the ability to add or remove optional items.

1. Access the Login Page at <http://www.opal411.legendaryfox.net/login.php>
2. Login to an existing user account that has already submitted web page data.
3. From the user's Main Account page click on the "Edit your page" link.
4. The Edit Page should already have some of its text fields populated with existing web page data. These pre-existing contents were obtained from the SQL database and represent the data that is currently displayed in the user's web page. These contents can be edited and replaced at will in order to update the page. Furthermore, optional items can be removed from view by deleting their contents and submitting a blank field for that item. Likewise, optional items can be added by simply providing data in that field and re-submitting the form.
5. Once the desired changes have been made, click on the "Submit" link on the bottom of the page to store those changes.
6. Click on the link at the bottom of the page to return to the Main Account page. At the Main Account Page click on the "View Page" link in order to see the updated web page.

(+) The ability to list all existing accounts

1. Access the Login Page at <http://www.opal411.legendaryfox.net/login.php>
2. Login using "admin" for the login and "cs411rocks" for the password (no quotation marks). This will access an Administrator Page to manage accounts.
3. All existing accounts are listed from top to bottom on the Administrator Page, along with the email address and full name (if provided) associated with each account.

(+) The ability to add/edit/delete members to/on/from a "permitted" list

1. Access the Login Page at <http://www.opal411.legendaryfox.net/login.php>
2. Login using "admin" for the login and "cs411rocks" for the password (no quotation marks).
3. At the Administrator Page, click the word "here" in the message "Create a new user here" listed at the bottom of the page.
4. Enter an email address to add to the pre-approved list and enter the admin password "cs411rocks" (without quotations). Then click "Submit Query"
5. A message indicating success should be returned, click on the link to return to the Administrator's main page. If the newly added email address is visible in the list of existing accounts then a member has been admitted to the permitted list successfully.

6. From the list of existing accounts, select an account whose email address will be changed. Click on the "Change Email" link to the right of the user account whose email address you wish to change.
7. Enter a new email address and click "Change Email". A success message should be returned.
8. Click on the link to return to the Administrator main page. The account should now have the updated email address instead of the old address.
9. Lastly, to delete a member from the permitted list click the "Mark Inactive" link to the right of their account. If an account is inactive it can no longer be logged into using the WSM; however, their web page will still be stored in the database. The web page will also be accessible on the web; however, the page will be displayed with a message indicating that the account is inactive.
10. Test the delete functionality by attempting to login and register using the deleted member's email address. Both login and registration should fail as that email address is no longer permitted access.

(+) The ability to delete an account

1. Access the Login Page at <http://www.opal411.legendaryfox.net/login.php>
2. Login using "admin" for the login and "cs411rocks" for the password (no quotation marks).
3. Next to each account listed there should be a "Delete" link. Choose an account to delete and click on the "Delete" link to its right.
4. Once a success message has been returned, click on the link to return to the Administrator main page. The deleted account should no longer be present on the administrator's list of existing accounts.
5. Test the delete functionality by attempting to login and register using the deleted email address. Both operations will fail as the email address has been deleted from the account list.

Non-Functional Requirements

This section provides a list of non-functional capabilities for the web application. The prefix '(+)' was added to each item as it was completed during implementation. A '(/)' denotes partial implementation.

(+) The user interface design will follow contemporary web conventions.

(+) Appropriate aesthetics should be used.

(+) New browser windows should be created when appropriate.

(+) The system must gracefully handle errors, such as when a user submits a form without entering data in all required fields.

(+) The data must be persistent, i.e. stored in a database.

(+) The system must run on popular browsers on popular operating systems.
(/) Each web page contains/displays the following required data (italics indicates an item that was not implemented):

- Email address
- University logo
- Recent studio photo
- Full Name
- *Title*
- *College Department*
- Contact information (email address, office location, office phone number)
- Curriculum Vitae
- Course actively taught
- Office hours
- Short bio

(+) Each web page must be capable of displaying the following optional data:

- Research
- Publications
- Upcoming events
- Ph D Students
- Links (to lab website, etc)
- Awards
- Organizations, groups

**Current as of 10/26/09*

Anticipated Risks of Requirements

The risk analysis will be conducted in a line by line fashion for each item required. Requirements will be seperated into functional and non-functional groupings. Each requirement will have an Expected Time of Implementation (ETI) in days based on a best or worst case scenario.

Functional Requirements

1. The ability to set up an account
   - Straightforward; just requires entry of login and password fields in the master table of the database. **ETI = 1 day**
   - Possible Complications: A verification system requires the use of email or interfacing with the Kerberos system. **ETI = 1 – 2 days**
   - **Total ETI = 2-3 days**

2. The ability to create an initial web page with minimal input
   - Straightforward from an SQL standpoint; just requires input and retrieval of data from the master table. Only two PHP functions need to be implemented; data input and retrieval. **ETI = 1 day**
   - Possible Complications: Error handling in order to limit user input to accepted formats may require testing. Also, getting PHP to generate HTML with a clean format regardless of what data fields are filled, not filled, and to what extent they are filled may complicate issues. **ETI = 2 – 3 days**
   - **Total ETI = 3-4 days**
3. The ability to update the initial web page, including the ability to add or remove non-optional items.
   - Straightforward; assuming web page creation has been completed this implementation is very similar. **ETI = 1 day**
   - Possible Complications: No foreseeable complications assuming that the web page can be generated properly in the first place. **ETI = 0 days**
   - **Total ETI = 1 day**
4. The ability to list all existing accounts
   - Straightforward; just requires grabbing all entries in the username column. **ETI = 1 day**
   - Possible Complications: None foreseeable. **ETI = 0 days**

     Total ETI = 1 day

5. The ability to add/edit/delete members to/on/from a "permitted" list
    - Straightforward; data entry functions should have been written before. Assuming that a "permitted" field exists for all accounts this is trivial. **ETI = 1 day**
    - Possible Complications: None foreseeable. **ETI = 0 days**
    - **Total ETI = 1 days**
6. The ability to delete an account
    - Straightforward; deletion of a row is already built into SQL. **ETI = 1 day**
    - Possible Complications: There may be an issue of integrating file deletion if the web server hosts any images/files for the user. **ETI = 1 days**
    - **Total ETI = 2 days**

Total ETI for Functional Requirements: 10 (best case) – 12 (worst case) days

Non-Functional Requirements

1. The user interface design will follow contemporary web conventions.
    - Extensive; this requirement is vague and the group must come to its own consensus on what web conventions are. **ETI = 3 days, with meetings**
    - Possible Complications: There may not be agreement on what conventions are and additional research may be needed. **ETI = 2-3 days**
    - **Total ETI = 5-6 days**
2. Appropriate aesthetics should be used.
    - Simple; aesthetic choice has been discussed in class and expectations for clean, academia appropriate aesthetic has been conveyed. **ETI = 1 day**
    - Possible Complications: Making the website aesthetically pleasing as well as user-friendly could take some testing and tweaking. **ETI = 3-5 days**

    Total ETI = 4-6 days
3. New browser windows should be created when appropriate.
    - Straightforward; opening a new browser window is a simple task. **ETI = 1 day**
    - Possible Complications: None. **ETI = 0 days**
    - **Total ETI = 1 day**

4. The system must gracefully handle errors, such as when a user submits a form without entering data in all required fields.
    - ○ Straightforward but thorough; since we have already defined the master table we know exactly what data types are acceptable for each field. **ETI = 2 days**
    - ○ Possible Complications: Although error-handling on a case by case basis is not difficult to implement, it may be difficult to predict all possible sources of error. User testing will be required **ETI = 2-3 days**

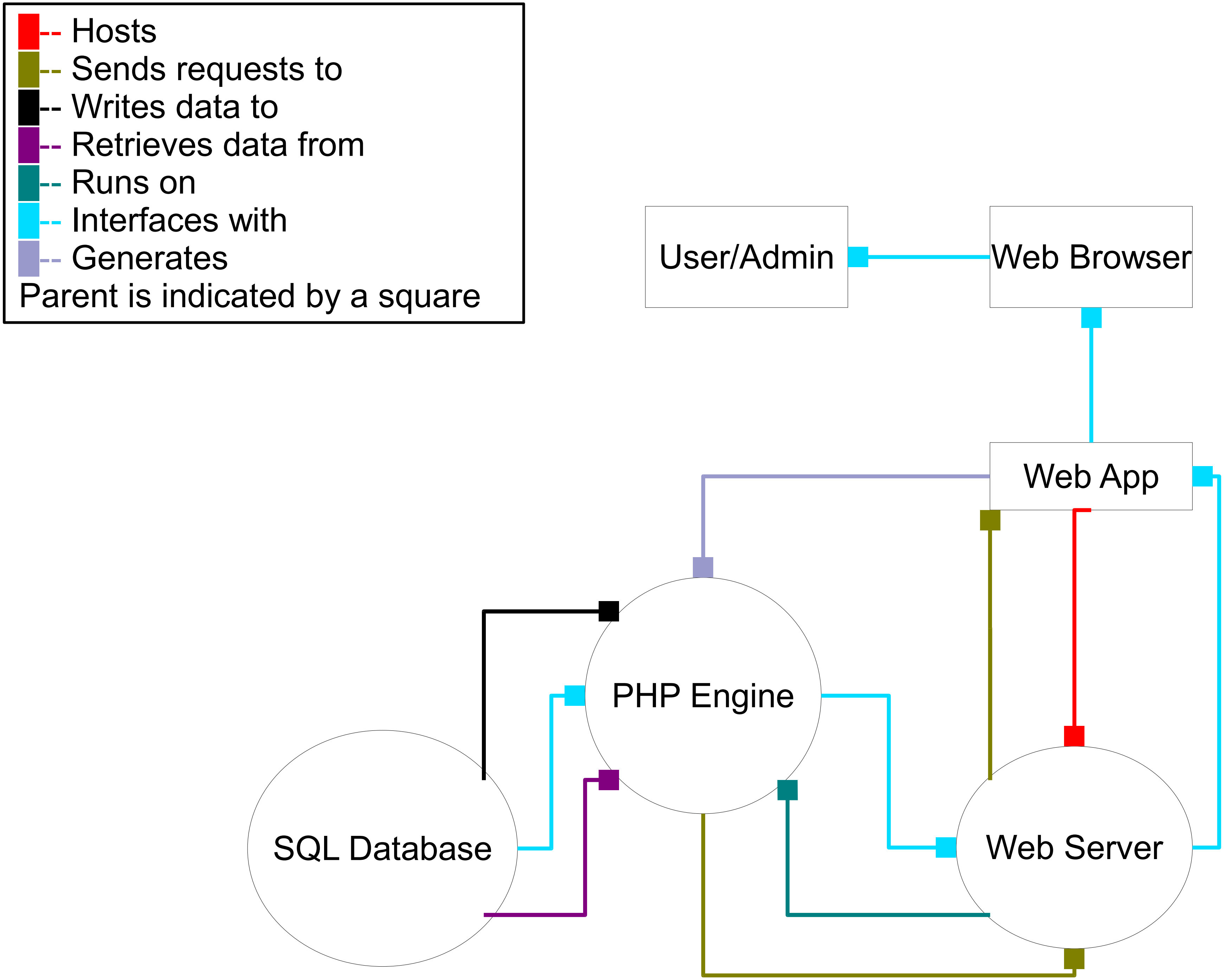
    Total ETI = 4-5 days
5. The data must be persistent, i.e. stored in a database.
    - ○ Straightforward; just requires the setup of SQL and use of PHP to transfer information. **ETI = 1 day**
    - ○ Possible Complications: Should be none as our team already has a member who has a web/PHP/SQL setup. **ETI = 0 days**
    - ○ **Total ETI = 1 day**
6. The system must run on popular browsers on popular operating systems.
    - ○ Extensive if we test all combinations of operating systems and browsers. But assuming we follow conventions for HTML/PHP the functionality should remain mostly intact regardless of user setup. Testing itself is straightforward as we will run through a test case using each setup **ETI = 2 days, with testing**
    - ○ Possible Complications: If a bug is found, debugging to correct behavior in a particular setup without affecting behavior in working setups may be time consuming. **ETI = 4-6 days**

    Total ETI = 4-6 days

Total ETI for Functional Requirements: 19 (best case) – 25 (worst case) days

<u>Total ETI for All Customer Requirements</u>: 29 (best case) – 37 days
<u>Start Date</u>: 9/21/09

<u>Projected End Date</u>: 10/20/09 - 10/28/09


■-- Hosts
■-- Sends requests to
■-- Writes data to
■-- Retrieves data from
■-- Runs on
■-- Interfaces with
■-- Generates
Parent is indicated by a square
User/Admin
Web Browser
Web App
PHP Engine
SQL Database
Web Server

System Design Document

System Overview

The philosophy behind the system design is to break down the deployed Web Site Manager into its core technologies (SQL Database, PHP Engine, and Web Server) and interfaces/users (User/Admin, Web Browser, Web App). By seperating development of the Web Site Manager into separate "technology" and "interface" nodes, we will maintain modularity of the system. This modularity will facilitate maintenance of the system by future developers.

A summary of the design philosophy is as follows:

1. Requests and operations originate from a user;
2. These requests/operations are sent through a Web Browser to the Web App;
3. The Web Server hosting the Web App begins processing the request;
4. The PHP Engine is called as necessary to process the Web Server's requests;
5. The SQL Database is accessed/manipulated as necessary to assist the PHP Engine;
6. The PHP Engine completes its task and generates the resulting HTML and/or return values;
7. The Web Server passes the results from the PHP Engine to the user's Web Browser
8. The Web Browser displays the results of the user's completed request/operation
9. The process repeats from Step 1 until the user has successfully created and/or updated their webpage

The System Design Diagram is a parent-child diagram that outlines the above design philosophy. More precisely, it shows the specific relationships between the components of the Web Site Manager system. The user (including the administrator as he is a type of user) is also displayed as a part of the entire working system. The diagram components are as follows:

- The User/Admin: Either an administrator or account-holder who is designing a web page. The User interfaces with the web application via a web browser in order to manage a create/manage/delete their web site. The Admin interfaces with the web application via a web browser in order to create/manage/delete user accounts.
- The Web App: The Web Site Manager is an application that is hosted on a web server. It takes commands from the user/admin via the web browser and incorporates function calls to a PHP engine in order to carry out those commands.
- The PHP Engine: A software engine capable of dynamically generating HTML according to functions defined by the implementers. It is passed requests/calls from the web server and uses data from the SQL database in order to fulfill those requests. It must be installed onto the web server in order to provide functionality.

- The SQL Database: A database that stores fields defined by the implementers, which were previously agreed upon at the conceptual level during team meetings. This is where all the information that is input by users is stored, as well as all account information.

Objects

There are two general types of objects that must be considered for this project. There are input objects which are passed from the input boxes of the web site manager to the database, and there is a storage object (referred to as the "master table") which will store all data conveyed in the input objects. More details on these objects are as follows

- Input Objects are pieces of data passed from the web site manager. They will all come from box inputs of some kind, and may contain the following values: *strings, numbers, blobs*.

- The Master Table (storage object) is constructed as a table with fields. *Field name: (data type [example/further details])*
  - Email (string)Unique ID (number)
  - Status (enumeration: [approved, verified, active, inactive])
  - Department (enumeration: [Computer Science, Mathematics, etc.])
  - Photo: (string: [filename])
  - Name: (string)
  - Contact Info
  - BU Office Address: (string)
  - BU Phone #: (number)
  - BU Email: already provided above under "Email"
  - CV: (string [filename); (blob [summary])
  - Course Info **We agreed on having FOUR copies of each field for a total of 4
    courses stored per semester for each instructor)
  - Course Name: (string)
  - Course Link: (string [filename])
  - Semester Taught: (string [auto-generated string such as "Fall 2009"])
  - Old Course Info: (string [text containing all previously taught courses; entered
    in manually for now])
  - Password Hint: (string)
  - Password Hint Answer: (string)

Anticipated Risks of Requirements

The risk analysis will be conducted in a line by line fashion for each item required. Requirements will be seperated into functional and non-functional groupings. Each requirement will have an Expected Time of Implementation (ETI) in days based on a best or worst case scenario.

Functional Requirements

1. The ability to set up an account
   - Straightforward; just requires entry of login and password fields in the master table of the database. **ETI = 1 day**
   - Possible Complications: A verification system requires the use of email or interfacing with the Kerberos system. **ETI = 1 – 2 days**
   - **Total ETI = 2-3 days**

2. The ability to create an initial web page with minimal input
   - Straightforward from an SQL standpoint; just requires input and retrieval of data from the master table. Only two PHP functions need to be implemented; data input and retrieval. **ETI = 1 day**
   - Possible Complications: Error handling in order to limit user input to accepted formats may require testing. Also, getting PHP to generate HTML with a clean format regardless of what data fields are filled, not filled, and to what extent they are filled may complicate issues. **ETI = 2 – 3 days**
   - **Total ETI = 3-4 days**
3. The ability to update the initial web page, including the ability to add or remove non-optional items.
   - Straightforward; assuming web page creation has been completed this implementation is very similar. **ETI = 1 day**
   - Possible Complications: No foreseeable complications assuming that the web page can be generated properly in the first place. **ETI = 0 days**
   - **Total ETI = 1 day**
4. The ability to list all existing accounts
   - Straightforward; just requires grabbing all entries in the username column. **ETI = 1 day**
   - Possible Complications: None foreseeable. **ETI = 0 days**

   Total ETI = 1 day

5. The ability to add/edit/delete members to/on/from a "permitted" list
    - Straightforward; data entry functions should have been written before. Assuming that a "permitted" field exists for all accounts this is trivial. **ETI = 1 day**
    - Possible Complications: None foreseeable. **ETI = 0 days**
    - **Total ETI = 1 days**
6. The ability to delete an account
    - Straightforward; deletion of a row is already built into SQL. **ETI = 1 day**
    - Possible Complications: There may be an issue of integrating file deletion if the web server hosts any images/files for the user. **ETI = 1 days**
    - **Total ETI = 2 days**

Total ETI for Functional Requirements: 10 (best case) – 12 (worst case) days

Non-Functional Requirements

1. The user interface design will follow contemporary web conventions.
    - Extensive; this requirement is vague and the group must come to its own consensus on what web conventions are. **ETI = 3 days, with meetings**
    - Possible Complications: There may not be agreement on what conventions are and additional research may be needed. **ETI = 2-3 days**
    - **Total ETI = 5-6 days**
2. Appropriate aesthetics should be used.
    - Simple; aesthetic choice has been discussed in class and expectations for clean, academia appropriate aesthetic has been conveyed. **ETI = 1 day**
    - Possible Complications: Making the website aesthetically pleasing as well as user-friendly could take some testing and tweaking. **ETI = 3-5 days**

    Total ETI = 4-6 days
3. New browser windows should be created when appropriate.
    - Straightforward; opening a new browser window is a simple task. **ETI = 1 day**
    - Possible Complications: None. **ETI = 0 days**
    - **Total ETI = 1 day**

4. The system must gracefully handle errors, such as when a user submits a form without entering data in all required fields.
   - Straightforward but thorough; since we have already defined the master table we know exactly what data types are acceptable for each field. **ETI = 2 days**
   - Possible Complications: Although error-handling on a case by case basis is not difficult to implement, it may be difficult to predict all possible sources of error. User testing will be required **ETI = 2-3 days**

   Total ETI = 4-5 days
5. The data must be persistent, i.e. stored in a database.
   - Straightforward; just requires the setup of SQL and use of PHP to transfer information. **ETI = 1 day**
   - Possible Complications: Should be none as our team already has a member who has a web/PHP/SQL setup. **ETI = 0 days**
   - **Total ETI = 1 day**
6. The system must run on popular browsers on popular operating systems.
   - Extensive if we test all combinations of operating systems and browsers. But assuming we follow conventions for HTML/PHP the functionality should remain mostly intact regardless of user setup. Testing itself is straightforward as we will run through a test case using each setup **ETI = 2 days, with testing**
   - Possible Complications: If a bug is found, debugging to correct behavior in a particular setup without affecting behavior in working setups may be time consuming. **ETI = 4-6 days**

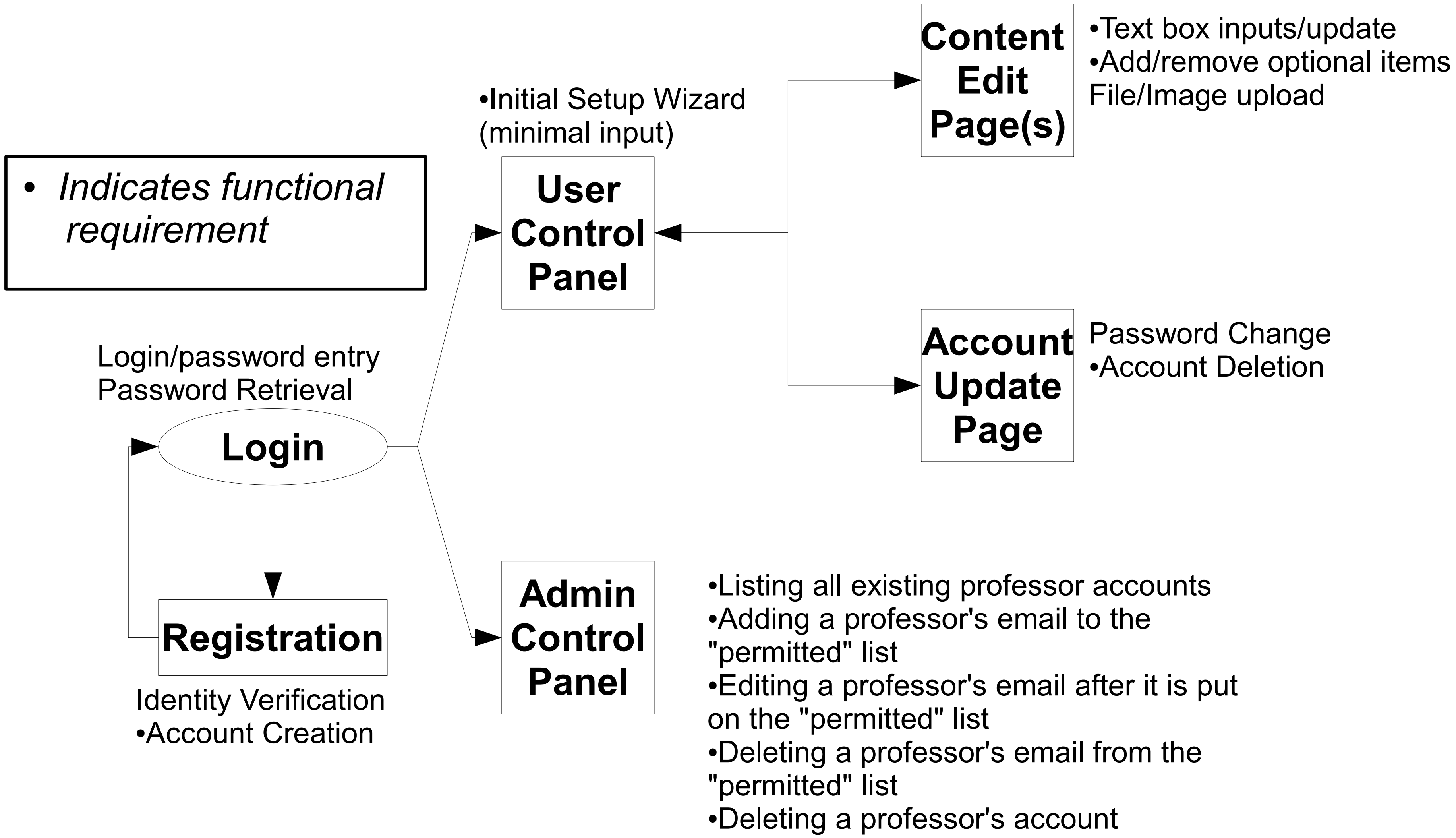
   Total ETI = 4-6 days

Total ETI for Functional Requirements: 19 (best case) – 25 (worst case) days

Total ETI for All Customer Requirements: 29 (best case) – 37 days
Start Date: 9/21/09

Projected End Date: 10/20/09 - 10/28/09

Web Site Manager Page Design
(Team Opal)



- *Indicates functional requirement*

Login/password entry
Password Retrieval

Login

Registration

Identity Verification
•Account Creation

•Initial Setup Wizard
(minimal input)

User Control Panel

Content Edit Page(s)

•Text box inputs/update
•Add/remove optional items
File/Image upload

Account Update Page

Password Change
•Account Deletion

Admin Control Panel

•Listing all existing professor accounts
•Adding a professor's email to the "permitted" list
•Editing a professor's email after it is put on the "permitted" list
•Deleting a professor's email from the "permitted" list
•Deleting a professor's account

User Documentation

**The following instructions assume a root (/) directory of http://www.opal411.legendaryfox.net/ Furthermore, the instructions will be written for use with the current implementation of the web application; instructions may change as development continues. Instructions are limited to whatever functionality has been currently implemented so far.*

Creating an Account

1. If the user does not have an account, go to /reg.php
2. User enters his email address and clicks "Submit Query" in order to begin the registration process.
3. If the email address is already registered or not on the Permitted List an error message is returned. Otherwise, the user will be notified that a confirmation email has been sent to their submitted email address.
4. Once the confirmation email has been received, the user clicks confirmation link and is directed to /confirm.php.
5. At this page, the user must submit a password for their web site manager account.
6. After the password is submitted, the user is logged in and directed to /edit_page.php
7. At this point the user's account has been successfully created and they are free to continue creating their faculty web page or to logout.

Logging in to an Existing Account

1. The user can login via /login.html
2. Once at the login page, the user supplies a valid login and password combination and clicks the “Submit Query” button
3. If the login and password combination is correct, the user is taken to /edit_page.php
   Using this edit page, the user can edit all of his website information

Creating/Editing a Web Page

1. First, the user must either create an account or log on to an existing account.
2. At the edit page, the user will be presented with a page containing a long form with different text fields for data entry. If data has already been stored for this account, the appropriate text fields will be automatically populated with data stored for that field.
3. In any case, the user should fill out the entire form to the best of their ability and click the submit button on the bottom of the page.
4. If there are any required (bolded) fields left blank, the form will return the user to the edit page and inform them that there are still required fields to be filled. These fields are marked with a red asterisk.
5. If the user desires to edit an automatically filled form they may do so by simply going to the text field and editing at will.

Viewing the Page

1. The user page is accessible from the professor index at the root index <http://opal411.legendaryfox.net>
2. Alternately, if logged onto a user account you can view the page in a popup window by clicking the “View page” link in the user account main page.

Changing the User Password

1. Login to a user's account
2. From the user's main account page click on the “Change Password” link
3. Follow the instructions and click “Submit Query” once your password has been changed

Logging Out

1. The user can log out by clicking "Log Out" buttons wherever they are present.
2. Alternately, the user may “manually” log out by directing their browser to /logout.php If successful, the user will be redirected to the login page and be prompted a login and password again.

Adding an Address to the Permitted List [Administrator]

1. First, the administrator must login administrator control page. This is done by accessing /login.php and logging in with username “admin” and password “cs411rocks” (without quotation marks).
2. At the bottom of the administrator control page there is message that says “Create a new user here”. Click the word “here”.
3. At this page you may add a new email address to the permitted list by entering the new email address and the admin password and clicking the submit button.

Changing the Status of an Account between Active and Inactive [Administrator]

1. Login to the administrator control page.
2. To the right of the account whose status you wish to change, click the link named “Mark Active” or “Mark Inactive”, depending on the account you are toggling.

Deleting an Account

1. Login to the administrator control page.
2. To the right of the account you wish to delete, click the link named “Delete”

Changing an Account's Email Address

1. Login to the administrator control page
2. To the right of the account whose email you wish to change, click the link named “Change Email”
3. At the next page enter the new email address and click the “Change Email” button.

Resetting an Account's Password

1. Login to the administrator control page
2. To the right of the account whose password you wish to reset, click the link named “Reset Password”
3. At the next page the new, randomly generated password will be posted.
4. Provide this randomly generated password to the user so that they may change their password to something else.

Tech Lead and Assistant Tech Lead Pre-Meeting

Date of Meeting: 9/18/09
People Present: Christian, Anastasia
Location: Undergraduate CS Lab
Start Time: 9 AM
End Time: 10:30 AM

1) **"Homework" before meeting:**
Implementers: Read customer requirements; identify specific tasks/technologies that they feel comfortable with (we will be assigning tasks to individuals on Saturday)

Assistant Tech Lead: Tentative schedule (can be made-up, just figure out process to create/maintain calender), bring 5 printouts, create Google Docs group (pages for each member's individual work log (5 total), one page for task list, one page for bulletin board, minutes page (chat logs, meeting logs, not meant to be read, emails)

Tech Lead: Create webpage diagram (nodes = webpages, branches = functionality, edges = page links), email reminding implementers

2) **Goals for tomorrow:**

- Hand out web page flowchart, tentative schedule
- Topics:
  - Introduce everybody (past experience with PHP/HTML for implementers) [Christian]
  - Design philosophy / breakdown tasks into pages or functionality or maybe some combination / use whiteboard [Christian]
  - Assign tasks / balance work load/ do schedule up to 1st snapshot (9/28) (prioritize tasks that are required first; than tasks which can be comfortably completed before 1st snapshot)
  - Issue of "source control" (leaning towards making sure no two people ever work on same code/file) [Christian]
  - Who will run, maintain server, where do people submit their updated pages/code? Can all tasks be worked on remotely? [Christian]
  - Work logs (make sure to date every entry), unified task list*, "bulletin board" for technical questions/announcements/concerns* ((Google Docs)), discuss channels of communication [Christian + Nastia]
  - Any questions/suggestions, did we meet all requirements? [Nastia]
  - Schedule next weekly meeting (chat, or in person)

Team Meeting

Date of Meeting: 9/19/09
People Present: Christian, Anastasia, Lincoln, James, Josiah
Location: Undergraduate CS Lab
Start Time: 10 AM
End Time: 11:30 AM

1) Agreed on overall concept (see “Initial System Design”) developed by Technical and Assistant Technical Lead at the Pre-Meeting on 9/18

2) Assigned Josiah to complete the following tasks before 9/28:

- Web server/FTP setup
- SQL setup on server
- Creating a test page to add an email account to the database

3) Assigned Lincoln and James to complete the following tasks before 9/28:

- Create database to store all required data
- Create a library of functions to add/manipulate/retrieve data from the database

4) Agreed on a rough timeline before the snapshot

- All tasks assigned to the implementers in the previous sections should be completed and integrated by Friday, the 25th/Saturday the 26th
- On Saturday one of the implementers will walk the tech lead through a demo of the product so far in order to verify that the assigned tasks have been successfully completed
- On Saturday the 26th/Sunday the 27th the tech lead and assistant tech lead will complete the documentation and deliverables for the 1st Project Snapshot on Monday, 9/28
- THE GOAL is to have a simple web page completed where a user can input a new email address into a text box and have the SQL server generate a new account (verification system need not be implemented yet).

5) Nastia introduced our Google Docs setup which will be used by the team to document the process of engineering the web tools for our client

- ALL TEAM MEMBERS must diligently record all work done on the project; whether or not any coding is done, any research, consultation, discussion of the project must be entered into the Google Docs system as work done
- There is a schedule for a broad overview of outstanding tasks and due dates
- There is a more specific task sheet for more detailed information on who is assigned which tasks and when they are due

6) Miscellaneous Questions/Concerns/Comments

- Agreed that commenting of code (including identification of who wrote what) is necessary (suggested by James)
- Using pre-existing libraries may be an option to assist us in keeping our project simple in code and modular (suggested by Josiah) *we need to ask the instructor about this
- We might want to look into communicating with the Kerberos login system in order to better integrate this system with existing BU services (suggested by Lincoln)

Implementer Meeting

Date of Meeting: 9/23/09
People Present: Lincoln, James, Josiah
Location: George Sherman Union (Study Lounge)
Start Time: 4:30 PM

Purpose is to decide who writes what code, and what we are writing

1) **Implementation Details and Proposals (all proposals will be/were reviewed by Tech Lead)**

- We voted not to use Smarty software
- We will take free templates from free websites for graphical purposes
- We voted to make all the information consistently displayed, but bottom part of page is anything using HTML box, perhaps a rich text box if we can find a free package at the end.
- Allow professors to upload their own image.
- Anything that needs multiple options can be a dynamic part of the input page, each getting a blob in the database that parses each fresh line as a new link, student, etc.
- Write own code for the parser
- For consistency, all table entry fields in the database are lowercase

2) **Assigned Tasks**

- Lincoln and James will write the functions and write the comments, and a Tech person will compile all the information into a document
- Lincoln will fix the sql table to reflect the updated requirments
- Both Lincoln and Josiah will write the query function in order to know how to do it and then we will select it, by Friday
- Lincoln and James will send the library functions to Josiah before noon on Friday
- Josiah will meet with Chris on Friday and they will arrange time to meet
- Josiah will email the sample sql code to the mailing list

3) **Issues/Concerns/Questions**

- Ask Keklak if we can use free embedded HTML editor in page.
- We need to meet to discuss the design of the page with the tech people.

Snapshot 1 Demo Meeting

Date of Meeting: 9/23/09
People Present: Christian, Josiah
Location: CS Undergraduate Lab
Start Time: 4:00 PM

1) **Retrieval process completed (pre-registered)**

- Retrieves entire record, but HTML only displays select entries
- Works through PHP by searching for the username; matches against the email field by looking at the characters before the @ sign
- No error handling

2) **Database is (nearly) completed**

- Because data insertion method has not been written, dummy data must be manually entered using phpMyAdmin (password protected, but implementers all currently have access)

3) **Perspective of the demo is of a student retrieving records**

- Demonstrates data retrieval functionality
- Located at http://www.opal411.legendaryfox.net/test_templated_ysjwang.php?uid=username
  username corresponds to the username identifying a particular record
  - Only existing records are lwaller and ysjwang
- The flow of data for the demo is as follows:

  User ID (BU email address) is passed to PHP engine à PHP engine dynamically requests record from SQL database à SQL passes record back to PHP à HTML selects what to display and how

Comments

- Template needs to be more refined (empty fields shouldn't) display. A more selective retrieval process may be considered (only selecting the specific data which is requested, for security/efficiency purposes)

Informal Schedule Estimate

- Based on discussion between Josiah and Christian
- Informal, will be proposed more formally at the next team meeting
- **Projection (1 week)**:
  - Registration, data entry will be finished within (functional, but not optimized or secure)
  - Division of labor: Same as before, Josiah (HTML), Lincoln and James (PHP)
  - Schedule mid-week check to see if Josiah needs to assist with PHP coding

Tech Lead and Implementers Meeting

Date of Meeting: 10/02/09
People Present: Christian, Anastasia, Lincoln, James
Location: Undergraduate CS Lab
Start Time: 9 AM
End Time: 10:30 AM

Assignments to be finished by Friday

1) **Josiah**

- Display layout of the webpage, with logos and links to the main BU website and the main department website (all of the professors pages now have those links).
- Look and feel of the login page
  - Font: model after the BU pages like Directory. Since this site is intended to be used by the BU professors, it should look Boston University official

2) **Lincoln**

- Authentication (of the user trying to login to edit the page)
- Validation (when creating the account by the user and the admin) and account creating
  - To Consider: Weblogin as Authentication

3) **James**

- Retrieving functions for data in database to the web page
- Entry into the website and error handling (form checking)

System Design Issues

- Administrator v. Professor Login: How to distinguish the login between the administrator of the whole system and just a professor who wants to edit his page.
  - Solution: Different login pages. Button somewhere on the website that says "Admin LogIn" --> queries separate field in the database?

Error Handling

- When to make sure that the professor entered the right type of data and filled out all the fields?
  - Decision: *After* the professor hits submit, the form is checked for mistakes

Letting "Authorized" people edit the page

- How can someone access the professor's page to edit a page for them?
  - 1st Solution: If Weblogin can be used as the authentication method-->Tab where the professor can add a person to edit their pages. Conditions: professor can only add one person. That person can only edit one professor's page. (room for improvement..)
  - 2nd Solution: If there is a separate log in name and password for the website --> professor can give the password to the "authorized" person (**issue:** people should not be giving out their passwords to anyone for any reason...

Goals for Next week

- Retrieve the filled form to edit the contents of the pages and fields that are already filled out.

Snapshot 2 Demo Meeting

Date of Meeting: 10/09/09
People Present: Christian, Lincoln, James, Josiah
Location: Undergraduate CS Lab
Start Time: 9 AM
End Time: 10:30 AM

Account Approval Page (admin.php [pass:cs411rocks])

- Used to add accounts; also sends notification after proper registration with a confirmation link
- Error handling for valid/invalid email addresses (lacking .net/.com) and login credentials

Registration Page (reg.php)

- Assumes a preapproved email address; if not returns error. Else, submits a confirmation message
- Typo in the notification address

Confirmation Link (confirm.php?confirm=<confirmation_key>)

- When clicked, takes you to a page to input your account password (has error checking).
- When a valid password has been entered takes you to the initial data entry page (edit_page.php).
- Red asterisks denoting required fields doesn't work (template change broke it?)
- Should remove fields that shouldn't have to be changed with each page edit (eg: password hint/answer)

Edit Page (edit_page.php)

- Automatically populates page with existing data if applicable by reading from the database using login key/session id.
- Submit doesn't work. *Fixed by Josiah

View Page (profpage.php?uid=<username only>)

- This php file will be used to display the professor's information

Login Page (login.html)

- Any incorrect combination of login and password will yield login error; correct entry goes to edit page

Logout (logout.php)

- Accessing this php file logs you out

Index of Professors (list_index.php)

- Lists all approved and confirmed accounts (still buggy; seems like it allocates space for accounts without a name entered)

Change Password (changepw.php)

- User can change password. Actually processed by pwscript.php

Password Retrieval

- Not implemented yet; is required

Limitations of Form Checking

- Only checks for whether data is entered or not, not format or length. Required fields only

Misc Notes:

- Logout buttons for each page
- Edit page does not have a field for name.
- List Index grabs all fields regardless of department.
- We need an index page; currently opal411.legendaryfox.net shows the entire directory.

Implementers Meeting

Date of Meeting: 10/17/09
People Present: Lincoln, James, Josiah
Location: Undergraduate CS Lab
Start Time: 9 AM
End Time: 10:30 AM

Discussion on redirecting to admin page

- Design choice – change login check page to see if username = "admin" or something; if so, redirect to admin control panel
    1. All "admin_" pages should have a different login-check – different session variable?? Basically checks login, and takes one step further to check if user = admin
- At admin control panel, do a listing of usernames – link directs to something like admin_edit_page.php?uid=username
    1. Features on admin page
        - Mark as inactive
        - Edit email address (ask why this was there???? Doesn't really make sense…)
        - Delete
        - Conscious decision to NOT edit all their fields
        - Password reset

"Main Account Page" – edit_account.php – has links to

- Change password + hints page
- Change account info (edit_page.php)
- "View page"

Misc fixes

- Change up template for displaying info – display more info!
- Login check should redirect to login.php, not login.html
- Remove 'login successful'
- Give ability to change 'name' field on edit_page.php
- Logout buttons (add to template?)

Edit_page.php

- Should we not display 'password hint' and 'password hint answer'
    1. We can move the password hint stuff to when the account is first created?

Non-functional requirements

- Do we really need to open new browser windows?
- Error handling seems to be in place

Outstanding Tasks

- Josiah
    1. Create admin pages
    2. Profpage.php – don’t display extraneous empty fields
- Lincoln
    1. Forgot password page
    2. Add logout button, redirect “login_checks” to login.php (login checks should be done either very first or very last in this session)
- James
    1. Ability to change your name on edit_page
    2. Remove the password hint + answer from edit_page
    3. Remove “login successful” from edit_page
    4. Error check for edit_page – make sure course expiration dates are current (ie in the future)
- Note to Nastya - Error checking doesn’t necessarily tell you what the error is, only gives a red asterisk

Final Meeting

Date of Meeting: 10/25/09
People Present: Christan, Nastia, Josiah
Location: Undergraduate CS Lab
Start Time: 5 PM
End Time: 5:50 PM

edit_page.php

- bold the required fields
- provide image upload

admin_index.php

- use user email as default display name (so it's not blank)
- be able to edit user's email

inactivity

- remove page from public's eye (or warn public about inactivity)
- disable ability to log in

alias table: user(email),pass,prof,full name

- activate alias
- each prof only has one alias (w/ unique password)

Misc Notes

- 'main account page' = edit_page.php
- view page
- edit password: add back the forget password hint/answer
- *assign alias*
- mailto link to admin - describe as well (do as [?])
- edit account:
- main_account.php
- FLOW

EMAIL LOGS - SNAPSHOT # 1

Team Opal (CS411)
19 messages
----
* Lincoln Waller <lwaller@bu.edu> * * Wed, Sep 16, 2009 at 9:18 AM *
To: "Seibel,James,Parker" <seibelj@bu.edu>, "Wang, Young-Sheng, Josiah" <ysjwang@bu.edu>, "Vashkevich, Anastasia, M" <nvashkev@bu.edu>, "Ling, Christian, Sing Hee" <linger@bu.edu>

Good morning fellow software engineers! I just was wondering if anyone had thought about which roles they might want to have on our team for the first project.

Lincoln
----
* Anastasia Vashkevich <nvashkev@bu.edu> * * Wed, Sep 16, 2009 at 9:46 AM *
To: Lincoln Waller <lwaller@bu.edu>

Morning!
I think i'd do better at assistant lead, or team lead than programming in php, because i don't know how to (though it seems like most of the class doesn't either, so it might not be such a big program). My first preference is Assistant team lead.

Anastasia
----
* linger.man@gmail.com <linger.man@gmail.com> * * Wed, Sep 16, 2009 at 9:45 AM *
To: Lincoln Waller <lwaller@bu.edu>
Cc: "Seibel,James,Parker" <seibelj@bu.edu>, "Wang, Young-Sheng, Josiah" <ysjwang@bu.edu>, "Vashkevich, Anastasia, M" <nvashkev@bu.edu>

I'd prefer to be the tech lead or assistant tech lead. I've been the pointman for write-ups and documentation in all my group projects and I don't want to buck the trend.

(Sorry for sending this to you twice Lincoln, I forgot to hit "Reply All" the first time)
----
* Young-Sheng Josiah Wang <ysjwang@bu.edu> * * Wed, Sep 16, 2009 at 10:31 AM *
To: Anastasia Vashkevich <nvashkev@bu.edu>
Cc: Lincoln Waller <lwaller@bu.edu>, "Seibel,James,Parker" <seibelj@bu.edu>, "Ling, Christian, Sing Hee" <linger@bu.edu>

I've done various programming projects in PHP/MySQL for more than 5 years (on and off), so I'm pretty fluent in it. I guess that makes me an implementer...

-(Young-Sheng) Josiah Wang

----
* seibelj@bu.edu <seibelj@bu.edu> * * Wed, Sep 16, 2009 at 6:15 PM *
To: Lincoln Waller <lwaller@bu.edu>
Cc: "Wang, Young-Sheng, Josiah" <ysjwang@bu.edu>, "Vashkevich, Anastasia, M" <nvashkev@bu.edu>, "Ling, Christian, Sing Hee" <linger@bu.edu>

I have no preference, I'll take whatever position remains unfilled.

I also think the tech and assistant tech leads will be forced to be implementers in the second project, so almost everyone will get to try a different position next time.

-James Seibel
----
* Josiah Wang <legendaryfox@gmail.com> * * Wed, Sep 16, 2009 at 7:07 PM *
To: "seibelj@bu.edu" <seibelj@bu.edu>
Cc: Lincoln Waller <lwaller@bu.edu>, "Wang, Young-Sheng, Josiah" <ysjwang@bu.edu>, "Vashkevich, Anastasia, M" <nvashkev@bu.edu>, "Ling, Christian, Sing Hee" <linger@bu.edu>

what's everyone's experience in php/mysql/web design? just curious
----
* Anastasia Vashkevich <nvashkev@bu.edu> * * Wed, Sep 16, 2009 at 7:18 PM *
To: Josiah Wang <legendaryfox@gmail.com>
Cc: "seibelj@bu.edu" <seibelj@bu.edu>, Lincoln Waller <lwaller@bu.edu>, "Wang, Young-Sheng, Josiah" <ysjwang@bu.edu>, "Ling, Christian, Sing Hee" <linger@bu.edu>

I have no experience with php/mysql/web design,

Hopefully the next project will be based on JAVA. I can do JAVA
----
* Lincoln Waller <lwaller@bu.edu> * * Wed, Sep 16, 2009 at 7:32 PM *
To: Josiah Wang <legendaryfox@gmail.com>
Cc: "seibelj@bu.edu" <seibelj@bu.edu>, "Wang, Young-Sheng, Josiah" <ysjwang@bu.edu>, "Vashkevich, Anastasia, M" <nvashkev@bu.edu>, "Ling, Christian, Sing Hee" <linger@bu.edu>

moderate amount of php/mysql, self taught. reasonable amount of basic web design otherwise, but nothing too fancy. (basic HTML with a little scripting. Occasional CSS).

Everyone good with Christian & Anastasia being our leads?
----
* linger.man@gmail.com <linger.man@gmail.com> * * Wed, Sep 16, 2009 at 8:11 PM *
To: Lincoln Waller <lwaller@bu.edu>
Cc: Josiah Wang <legendaryfox@gmail.com>, "seibelj@bu.edu" <seibelj@bu.edu>, "Wang, Young-Sheng, Josiah" <ysjwang@bu.edu>, "Vashkevich, Anastasia, M" <nvashkev@bu.edu>

HTML illiterate.

Alright, so it sounds like I'll be the tech lead and Anastasia will be the assistant tech lead. I know that the professor suggested doing chats but I'd really like to meet in person for our first meeting so that we can all put each others names to actual faces. I can also bring a laptop to take minutes of that first meeting. How does Saturday or Sunday afternoon for our first meeting sound in the Undergraduate CS Lab (in the building next to Espresso Royale, 3rd floor)?

----

* Lincoln Waller <lwaller@bu.edu> * * Wed, Sep 16, 2009 at 8:25 PM *
To: linger.man@gmail.com
Cc: Josiah Wang <legendaryfox@gmail.com>, "seibelj@bu.edu" <seibelj@bu.edu>, "Wang, Young-Sheng, Josiah" <ysjwang@bu.edu>, "Vashkevich, Anastasia, M" <nvashkev@bu.edu>

Sunday's out for me. I'll be at the lab already 10a-4:30p Saturday, so anytime in there works!

----

* seibelj@bu.edu <seibelj@bu.edu> * * Wed, Sep 16, 2009 at 8:50 PM *
To: Lincoln Waller <lwaller@bu.edu>
Cc: linger.man@gmail.com, Josiah Wang <legendaryfox@gmail.com>, "Wang, Young-Sheng, Josiah" <ysjwang@bu.edu>, "Vashkevich, Anastasia, M" <nvashkev@bu.edu>

I have little experience with php/mysql/web design but I'm 100% confident I can figure it out, and I want the opportunity to learn it also.

Saturday is no good for me unless we meet before noon, my apologies, but Sunday is fine after 2:00pm. Weeknights after 8 are usually wide open for me.

----

* Anastasia Vashkevich <nvashkev@bu.edu> * * Wed, Sep 16, 2009 at 10:04 PM *
To: seibelj@bu.edu
Cc: Lincoln Waller <lwaller@bu.edu>, linger.man@gmail.com, Josiah Wang <legendaryfox@gmail.com>, "Wang, Young-Sheng, Josiah" <ysjwang@bu.edu>

Sundays, im free after 10 (i am busy in the mornings). Saturdays im also free.
Weds and Thurs nights are not good for me. Monday nights.. well we have class then...so yea. other nights im cool.

----

* linger.man@gmail.com <linger.man@gmail.com> * * Wed, Sep 16, 2009 at 10:31 PM *
To: Anastasia Vashkevich <nvashkev@bu.edu>
Cc: seibelj@bu.edu, Lincoln Waller <lwaller@bu.edu>, Josiah Wang <legendaryfox@gmail.com>, "Wang, Young-Sheng, Josiah" <ysjwang@bu.edu>

By general consensus it sounds like Saturday, 10AM in the CS Undergrad Computer Lab.

----

* Young-Sheng Josiah Wang <ysjwang@bu.edu> * * Wed, Sep 16, 2009 at
11:58 PM *
To: linger.man@gmail.com
Cc: Anastasia Vashkevich <nvashkev@bu.edu>, seibelj@bu.edu, Lincoln
Waller <lwaller@bu.edu>

saturday 10AM sounds good. should we all distribute phone numbers? just in case

i'll start, 908-327-5898

----

* linger.man@gmail.com <linger.man@gmail.com> * * Thu, Sep 17, 2009 at
12:46 AM *
To: Young-Sheng Josiah Wang <ysjwang@bu.edu>
Cc: Anastasia Vashkevich <nvashkev@bu.edu>, seibelj@bu.edu, Lincoln
Waller <lwaller@bu.edu>

808-281-0623

----

* Anastasia Vashkevich <nvashkev@bu.edu> * * Thu, Sep 17, 2009 at 7:55
AM *
To: linger.man@gmail.com
Cc: Young-Sheng Josiah Wang <ysjwang@bu.edu>, seibelj@bu.edu, Lincoln
Waller <lwaller@bu.edu>

617 - 283 - 6190

----

* seibelj@bu.edu <seibelj@bu.edu> * * Thu, Sep 17, 2009 at 8:00 AM *
To: Anastasia Vashkevich <nvashkev@bu.edu>

10:00am on Saturday it is. My number is 508-574-2748

-james

----

* seibelj@bu.edu <seibelj@bu.edu> * * Thu, Sep 17, 2009 at 9:03 AM *
To: Anastasia Vashkevich <nvashkev@bu.edu>
Cc: linger.man@gmail.com, Young-Sheng Josiah Wang <ysjwang@bu.edu>,
Lincoln Waller <lwaller@bu.edu>
I don't think I remembered to hit 'reply to all' for this message.

Saturday 10:00am is fine. My number is 508-574-2748

-james

"Homework" before Group Meeting Tomorrow
3 messages

----

* linger.man@gmail.com <linger.man@gmail.com> * * Fri, Sep 18, 2009 at

2:35 PM *
To: seibelj@bu.edu, nvashkev@bu.edu, ysjwang@bu.edu, lwaller@bu.edu

Hello Team,

Just wanted to let you guys know that everyone is expected to have read the Customer's Requirements thoroughly before coming to the meeting tomorrow at 10AM in the CS Undergraduate Lab.

To the implementers, please go over all the requirements and think about which technical parts of the project you would feel comfortable/prefer to work on. Some of the goals tomorrow are to agree on an overall conceptual design; divide up the design into discrete tasks; allocate said tasks to individual implementers; and hammer out a schedule with proposed deadlines up until the 1st project snapshot due date (9/28). If everyone comes to the meeting with at some idea of what kinds of coding/scripting tasks they feel confident doing then allocation of the tasks will be much more efficient.

Thanks and I look forward to seeing you all tomorrow.

-Christian

----

* seibelj@bu.edu <seibelj@bu.edu> *   * Fri, Sep 18, 2009 at 2:42 PM *
To: linger.man@gmail.com
Cc: nvashkev@bu.edu, ysjwang@bu.edu, lwaller@bu.edu

That sounds good. One of the tech leads should be saving (copy -> paste) all correspondence we have by email/chat/etc. in a chronological document. The emails we have already had can only help boost our grade, which seems to be more about how we communicate than how our actual project turns out.

-James Seibel

----

* Anastasia Vashkevich <nvashkev@bu.edu> *   * Fri, Sep 18, 2009 at 3:03 PM *
To: seibelj@bu.edu
Cc: linger.man@gmail.com, ysjwang@bu.edu, lwaller@bu.edu

Got the correspondence covered. Btw, is it safe to assume that everyone has a gmail account?

Google Groups: You've created the new group Opal
1 message

----

* noreply@googlegroups.com <noreply@googlegroups.com> *   * Sat, Sep 19, 2009 at 11:31 AM *
To: nvashkev@gmail.com

Hello nvashkev@gmail.com <mailto:nvashkev@gmail.com>,

Congratulations: you've successfully created your Google Group, Opal.

Here are the essentials:

* Group name: Opal
* Group home page: http://groups.google.com/group/411opal
<http://groups.google.com/group/411opal>
* Group email address 411opal@googlegroups.com
<mailto:411opal@googlegroups.com>

And here are links to a few more Google Group-related goodies:

* Change group settings: http://groups.google.com/group/411opal/manage
<http://groups.google.com/group/411opal/manage>
* Invite more users:
http://groups.google.com/group/411opal/members_invite
<http://groups.google.com/group/411opal/members_invite>

If you have questions about this or any other group, please visit the Google Groups Help Center at http://groups.google.com/support?hl=en
<http://groups.google.com/support?hl=en>.

Enjoy your group and make us proud!

The Google Groups Team

Notes on Database Fields
2 messages

----

* linger.man@gmail.com <linger.man@gmail.com> * * Sat, Sep 19, 2009 at 12:19 PM *
To: Lincoln Waller <lwaller@bu.edu>, "Wang, Young-Sheng, Josiah" <ysjwang@bu.edu>, "Seibel,James,Parker" <seibelj@bu.edu>, "Vashkevich, Anastasia, M" <nvashkev@bu.edu>

Format is as follows
Field name: (data type [example/further details])

Email (string)

Unique ID (number)

Status (enumeration: [approved, verified, active, inactive])

Department (enumeration: [Computer Science, Mathematics, etc.])

Photo: (string: [filename])

Name: (string)

Contact Info

- BU Office Address: (string)
- BU Phone #: (number)
- BU Email: already provided above under "Email"

CV: (string [filename); (blob [summary])

Course Info **We agreed on having FOUR copies of each field for a total of 4 courses stored per semester for each instructor)

- Course Name: (string)
- Course Link: (string [filename])
- Semester Taught: (string [auto-generated string such as "Fall 2009"])

Old Course Info: (string [text containing all previously taught courses; entered in manually for now])

Password Hint: (string)
Password Hint Answer: (string)
----
* Lincoln Waller <lincoln.waller@gmail.com> *   * Sat, Sep 19, 2009 at 12:27 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com

Thanks Christian!

The googlegroup is active so we can all just email and reply to 411opal@googlegroups.com <mailto:411opal@googlegroups.com>

And as such all communication will be documented in the group so we can print that out later if need be.

Team Opal - Email Issue
1 message
----
* linger@bu.edu <linger@bu.edu> *   * Sun, Sep 20, 2009 at 5:03 AM *
To: lwaller@bu.edu, seibelj@bu.edu, ysjwang@bu.edu, nvashkev@bu.edu, linger@bu.edu

Hey guys, my Gmail account has apparently been compromised by some asshole in Spain (or some other Spanish speaking country) and I don't have access to it anymore. Please use my email address linger@bu.edu

<mailto:linger@bu.edu> for now until I can get the issue resolved. And
Nastia, I'd appreciate it if you could send my linger@bu.edu
<mailto:linger@bu.edu> email an invite for the Google Docs (you should
also probably disable my Gmail from access just in case). Thanks!

-Chris

[Opal] Team Opal - Email Issue (update; please read)
4 messages

----

* linger.man@gmail.com <linger.man@gmail.com> *   * Sun, Sep 20, 2009 at
5:38 AM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com

Hey team,

Alright I finally got control of my email account back. Google's account
recovery is a wonderful, wonderful thing. Just so you guys know its me
we have CS411 together in the Computer Science building and the room we
had lecture in has a noisy fan. Our "instructor" Mr. Keklak isn't really
a professor at BU and he doesn't like the idea of professors putting
pictures of their family up on their university websites. Alright see
you guys later!

----

* Josiah Wang <legendaryfox@gmail.com> *   * Sun, Sep 20, 2009 at 2:43 PM *
Reply-To: 411opal@googlegroups.com
To: "411opal@googlegroups.com" <411opal@googlegroups.com>

hey guys,

just wanted to keep you updated, my hosting service has been running
maintenance today but I should have the accounts setup by tonight, just FYI

-Josiah

----

* Josiah Wang <legendaryfox@gmail.com> *   * Mon, Sep 21, 2009 at 12:35 AM *
Reply-To: 411opal@googlegroups.com
To: "411opal@googlegroups.com" <411opal@googlegroups.com>

ok guys, so here's the preliminary stuff i set up so far.

http://opal411.legendaryfox.net <http://opal411.legendaryfox.net> is the
main site.

when you ftp in, username is opal411, password is cs411rocks. i haven't
set up multiple users yet. do we need multiple users, or will this be a
problem?

mysql stuff: when connecting, the mysql host is *NOT* localhost; it's
'opal411mysql.legendaryfox.net <http://opal411mysql.legendaryfox.net>' i
put in the request for it, it should be set up by tomorrow morning. i'll
provide the link for the phpmyadmin tool later on.

for now, it hink most of us are doing our development on our own
computers using LAMP/WAMP, am i correct? in the meantime i'll continue
to set up the server stuff

just to keep you updated

-josiah

----

* Josiah Wang <legendaryfox@gmail.com> *   * Mon, Sep 21, 2009 at 12:39 AM *
Reply-To: 411opal@googlegroups.com
To: "411opal@googlegroups.com" <411opal@googlegroups.com>

alright, phpmyadmin is set up. http://opal411mysql.legendaryfox.net/
<http://opal411mysql.legendaryfox.net/> . user/pass is opal411/cs411rocks

----

[Opal] database
1 message

----

* Lincoln Waller <lincoln.waller@gmail.com> *   * Mon, Sep 21, 2009 at
10:54 AM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com

thank you for getting the server setup.
I created a table called "profs" in the database this morning and gave
it all of the fields from the notes, plus a password field (don't know
how that didn't make it on our list...) But maybe we can use weblogin
and not worry about it anyways. So currently the table is 26 fields.
I've only put in CAS departments, not university wide departments.
Therefore they are just entries like "CS" and "MA", not "CASCS". If
ya'll think we should change to the "CASCS" style (even if we don't
actually add other college's departments) just let me know.

Just a thought when it comes to coding, the code should prevent a user
from logging in while they are just "approved" (e.g. they should be
directed to a verified page). I say this because until they set a
password during verification, their password field is empty...

Also something that I thought would make things just a bit easier is
that the "semester" option for the courses is actually a date field, and
the date there represents the end date of the course. This makes it
easier to check when a course should or should not be listed. (Because
we don't need to have "fall 09" stored; we know that during fall 09 all
courses which do get displayed are fall 09 courses. If that makes sense)

----

*[Opal] Sample Code that demonstrates mysql functions and extraction to
variables (from 2009/09/23 implementer's meeting)*
1 message

----

* Josiah <legendaryfox@gmail.com> * * Wed, Sep 23, 2009 at 5:35 PM *
Reply-To: 411opal@googlegroups.com
To: Opal <411opal@googlegroups.com>

```
<?php

mysql_connect('localhost', 'root', ''); //basic connection; arguments
are hostname, username, password


mysql_select_db("opal411"); //select the database


$sql = "SELECT * FROM profs WHERE id = 0"; //query
$res = mysql_query($sql); //store the mysql resource handler


$res_array = mysql_fetch_array($res); //format the resource handler
into an array that php can use
extract($res_array); //'explodes' the array into easily useable names
that correspond to field names. see php manual on extract function for
more powerful applications

print "Hello my name is $name, email is $email, and my phone number is
$phone";
?>
```

[Opal] Team Opal - Email Issue (update; please read)
20 messages

----

* Josiah Wang <legendaryfox@gmail.com> * * Sun, Sep 20, 2009 at 2:43 PM *
Reply-To: 411opal@googlegroups.com
To: "411opal@googlegroups.com" <411opal@googlegroups.com>

hey guys,

just wanted to keep you updated, my hosting service has been running
maintenance today but I should have the accounts setup by tonight, just FYI

-Josiah

[Quoted text hidden]

----
* Josiah Wang <legendaryfox@gmail.com> * * Mon, Sep 21, 2009 at 12:35 AM *
Reply-To: 411opal@googlegroups.com
To: "411opal@googlegroups.com" <411opal@googlegroups.com>
ok guys, so here's the preliminary stuff i set up so far.

http://opal411.legendaryfox.net <http://opal411.legendaryfox.net> is the main site.

when you ftp in, username is opal411, password is cs411rocks. i haven't set up multiple users yet. do we need multiple users, or will this be a problem?

mysql stuff: when connecting, the mysql host is *NOT* localhost; it's 'opal411mysql.legendaryfox.net <http://opal411mysql.legendaryfox.net>' i put in the request for it, it should be set up by tomorrow morning. i'll provide the link for the phpmyadmin tool later on.

for now, it hink most of us are doing our development on our own computers using LAMP/WAMP, am i correct? in the meantime i'll continue to set up the server stuff

just to keep you updated

-josiah
[Quoted text hidden]

----
* Josiah Wang <legendaryfox@gmail.com> * * Mon, Sep 21, 2009 at 12:39 AM *
Reply-To: 411opal@googlegroups.com
To: "411opal@googlegroups.com" <411opal@googlegroups.com>
alright, phpmyadmin is set up. http://opal411mysql.legendaryfox.net/ <http://opal411mysql.legendaryfox.net/> . user/pass is opal411/cs411rocks
[Quoted text hidden]

----
* jamesishere <jamesishere@gmail.com> * * Mon, Sep 21, 2009 at 9:07 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
I am able to access it fine from my desktop, I don't know what was up with my laptop.

I'll let the implementers know by tomorrow night if I am able to attend Wednesday's meeting.

Have a good night everybody,

-James
[Quoted text hidden]

----
* Lincoln Waller <lincoln.waller@gmail.com> *   * Mon, Sep 21, 2009 at
9:15 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Thanks for the update!

Don't think my phone number made it out last time - 7042011146

James- you mentioned being swamped this week. I've got it pretty light right now but am expecting a heavier workload in the two weeks to come. Would you like me to go ahead and write most of the code for this first step, then you can look at it and write some of the other code in the weeks to come? (Or wait until tomorrow night / at the wed meeting to decide?)

Lincoln
[Quoted text hidden]

----
* jamesishere <jamesishere@gmail.com> *   * Mon, Sep 21, 2009 at 9:43 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Let's wait until I know if I'll be able to make the meeting to decide about the coding. Even though this week I'm swamped, I could easily be just as swamped next week, this whole semester is insane =(

-James
[Quoted text hidden]

----
* Lincoln Waller <lincolnwaller@gmail.com> *   * Mon, Sep 21, 2009 at
9:45 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Sounds good!
[Quoted text hidden]

----
* Josiah Wang <legendaryfox@gmail.com> *   * Tue, Sep 22, 2009 at 11:26 AM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Hey implementers,

Sorry, I totally forgot that I have a previous appointment from 6-7PM tomorrow (Wednesday), so that timeslot I won't be free. Lincoln, are you free beforehand? I can meet up with you from 5 to 6, then I can be back again from 7 to 8 if either Lincoln or James is free. My meeting is at the GSU, so it works out somewhat okay.

Sorry!

-Josiah
[Quoted text hidden]

----
* Lincoln Waller <lincoln.waller@gmail.com> * * Tue, Sep 22, 2009 at
11:30 AM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
I am free at 5. Let's see tonight when/if James is free and go from there!
Thanks
Lincoln
[Quoted text hidden]

----
* jamesishere <jamesishere@gmail.com> * * Tue, Sep 22, 2009 at 4:33 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Hi all,

I just received tomorrow afternoon off from my boss, so anytime after
1:00pm tomorrow is good for me.

-James
[Quoted text hidden]

----
* Josiah Wang <legendaryfox@gmail.com> * * Tue, Sep 22, 2009 at 6:48 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
I have a meeting until 3. After that, I'm free (until the 6 to 7 gap).
I'll probalby be at teh GSU around 4:30ish working on the project stuff;
we can just impromptu meet up from there/then?

-Josiah (908-327-5898)
[Quoted text hidden]

----
* jamesishere <jamesishere@gmail.com> * * Tue, Sep 22, 2009 at 8:36 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
How about 4:30pm in the second floor study room of GSU?

-james
[Quoted text hidden]

----
* Josiah Wang <legendaryfox@gmail.com> * * Tue, Sep 22, 2009 at 9:32 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Sounds good. See you then!
[Quoted text hidden]

----

* linger.man@gmail.com <linger.man@gmail.com> *   * Wed, Sep 23, 2009 at 12:00 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Hey I just wanted to remind you implementers that someone should document what happens in the meeting today. It doesn't have to be detailed: just a brief outline of major topics will do.

And I need to meet with one of you implementers on Friday so I can see a demo of our project so far and gather some info for snapshot 1. I'm leaning towards Josiah just because he has the most experience with PHP/SQL and has set up all the server stuff. However, you implementers may elect another representative if someone else has a better understanding of all the technical components/better availability this weekend.
[Quoted text hidden]

----

* Josiah Wang <legendaryfox@gmail.com> *   * Wed, Sep 23, 2009 at 5:43 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Hey Chris, I can meet up with you on Friday. I have class 9:30 to 12:30 and then from 2 to 3; besides that, any other time is fine. Lemme know when is a good time

-Josiah (9083275898)
[Quoted text hidden]

----

* linger.man@gmail.com <linger.man@gmail.com> *   * Wed, Sep 23, 2009 at 5:50 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Josiah: After 3 okay? We shouldn't take much longer than an hour so there'll still be plenty of time for you to enjoy your Friday.

Oh, and whoever took notes for the implementer meeting don't forget to email the notes to Nastia/me. Thanks!
[Quoted text hidden]

----

* Josiah Wang <legendaryfox@gmail.com> *   * Thu, Sep 24, 2009 at 6:48 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Yeah sure. How about 4 tomorrow? Just to be on the safe side. I'm not sure I'll have time tonight to compile everything together tonight [swamped with a problem set for an engineering class]. By then I'll definitely have everything together. I'll let you know if that changes. Where is convenient for you to meet? I end class in the SED building, but I live in South Campus. I have a bike, so anywhere is convenient.

-Josiah (908-327-5898)
[Quoted text hidden]

----
* linger.man@gmail.com <linger.man@gmail.com> * * Thu, Sep 24, 2009 at 8:27 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Sounds good. Let's meet at the Ugrad Lab at 4.

-Christian
[Quoted text hidden]

----
* jamesishere <jamesishere@gmail.com> * * Fri, Sep 25, 2009 at 10:04 AM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Hey guys,

Just an update on the coding (or lack there-of).

Basically, the code Lincoln and I were going to have ready by noon today was the same extract code Josiah put together at the meeting, except with an extra function to print each separate variable from the selected Professor key on a new line.

I have not completed this yet, but the output is simple to understand, so I think Josiah can "demo" the project to Chris by showing him the database, how the extract code works, and explaining the fundamental plan of how the information will be outputted to the screen on the final version.

If Lincoln has completed the code he can email it to the group by noon today, otherwise I don't think not having it for today's meeting is a big problem.

-James
[Quoted text hidden]

EMAIL LOGS - SNAPSHOT 2

[Opal] Next Meeting
11 messages

----

* linger.man@gmail.com <linger.man@gmail.com> *   * Wed, Sep 30, 2009 at 2:53 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Hey guys,

Sorry for not sending this message sooner-- I've been slacking a bit in my duties since I have a lot of things on my plate this week-- but we need to arrange a group meeting to review our progress now that we have feedback from the instructor. We'll also need to schedule tasks for the next two weeks so that we finish our project on time. I will be available on Friday, Saturday or Sunday. Thanks for all the good work so far (it sounds like our snapshot was received fairly well) and please respond with your availability for a team meeting during the remainder of this week.

----

* Anastasia Vashkevich <nvashkev@gmail.com> *   * Wed, Sep 30, 2009 at 3:04 PM *
To: 411opal@googlegroups.com
hey,

I can do Friday any time (preferably morning though). On saturday im free after 10AM and on sunday im free between 10-5.
[Quoted text hidden]

----

* Lincoln Waller <lincoln.waller@gmail.com> *   * Wed, Sep 30, 2009 at 3:28 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
I'm going out of town for the weekend... I can do Friday morning if we meet at the lab in the 9a-11a range. What about Thursday afternoon/evening as an option? I'm free after 5.
[Quoted text hidden]

----

* Anastasia Vashkevich <nvashkev@gmail.com> *   * Wed, Sep 30, 2009 at 3:59 PM *
To: 411opal@googlegroups.com
Thursday nights are not good for me.
[Quoted text hidden]

----

* Josiah Wang <legendaryfox@gmail.com> *   * Wed, Sep 30, 2009 at 4:59 PM *
Reply-To: 411opal@googlegroups.com

To: "411opal@googlegroups.com" <411opal@googlegroups.com>
Cc: "411opal@googlegroups.com" <411opal@googlegroups.com>

I won't really have a free time slot till this weekend; on Sunday I
won't be free until 2 or 3ish.
[Quoted text hidden]

----
* linger.man@gmail.com <linger.man@gmail.com> *   * Wed, Sep 30, 2009 at
5:56 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Alright, just waiting for James to reply. But it sounds like we're going
to meet Friday morning and forward the minutes to Josiah.
[Quoted text hidden]

----
* jamesishere <jamesishere@gmail.com> *   * Wed, Sep 30, 2009 at 7:12 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Friday morning is fine for me, unless some emergency happens at my work,
which is unlikely.

-James
[Quoted text hidden]

----
* linger.man@gmail.com <linger.man@gmail.com> *   * Thu, Oct 1, 2009 at
11:50 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Okay, meeting tomorrow (Friday) morning from 9-11AM. It sounds like
everyone except Josiah will be present.
[Quoted text hidden]

----
* linger.man@gmail.com <linger.man@gmail.com> *   * Thu, Oct 1, 2009 at
11:50 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
It will be at the Undergrad CS Lab as usual.
[Quoted text hidden]

----
* Josiah Wang <legendaryfox@gmail.com> *   * Fri, Oct 2, 2009 at 3:42 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Hey guys, sorry I couldn't make it. Are there meetings notes / tasks
posted from it?

-Josiah
[Quoted text hidden]

----
* linger.man@gmail.com <linger.man@gmail.com> *   * Fri, Oct 2, 2009 at
7:08 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Josiah- no worries. Nastia should be sending you a copy of the minutes
soon. But we've agreed that your primary task is to have the HTML layout
completed for the primary display page by next Friday. Use a preexisting
dummy record for now to read data from and be sure that ALL fields are
displayed even if they're empty (or even better, fill all the fields
with some text and write the HTML so it doesn't show empty field). We've
also agreed that the aesthetic of the page should be similar to the
Studentlink pages. Basically red and white like our school colors.
[Quoted text hidden]

[Opal] Pages to design for
2 messages
----
* Josiah Wang <legendaryfox@gmail.com> *   * Mon, Oct 5, 2009 at 1:15 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Hi implementers,

I'm gonna be working on the HTML templates this week. Are all the
functional PHP currently on the server right now? Just wanted to get in
mind what I have to work with.

Thanks!

-Josiah

----
* jamesishere <jamesishere@gmail.com> *   * Mon, Oct 5, 2009 at 3:41 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
I have blocked tomorrow and Thursday nights for coding the php
functions. If you design the HTML using the sample database professor it
should theoretically work for all other professors as well. I believe I
am working on the php functions alone, to be finished by Friday, and
Lincoln is working on the account creation/authentication code.

-James
[Quoted text hidden]

[Opal] confirmation code
4 messages
----
* Lincoln Waller <lincoln.waller@gmail.com> *   * Mon, Oct 5, 2009 at
7:37 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
is it cool for me to add a second table to the database to use to store
data regarding the confirmation email sent to a professor for account
verification?

This also opens up the possibility of entering the professor in that
table first and not in the other database until they are confirmed.

Thoughts?

----
* linger.man@gmail.com <linger.man@gmail.com> *   * Mon, Oct 5, 2009 at
8:34 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Sure, go ahead. I'll ask Keklak about using Kerberos soon.
[Quoted text hidden]

----
* Lincoln Waller <lincoln.waller@gmail.com> *   * Mon, Oct 5, 2009 at
8:39 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Yeah I looked around at kerberos for a while and (of course) it may be
more difficult than imagined. That said the other may be easier than
imagined... But go ahead and check just 'cuz that'd be cool to use if
we can
[Quoted text hidden]

----
* linger.man@gmail.com <linger.man@gmail.com> *   * Mon, Oct 5, 2009 at
9:01 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Keklak gave the okay but he said it definitely goes above and beyond
what he expects from us. Like you seem to be suggesting, he saw it as
being a more difficult implementation than managing the authentication
ourselves.
[Quoted text hidden]

[Opal] authentication

9 messages

----

* Lincoln Waller <lincoln.waller@gmail.com> *   * Tue, Oct 6, 2009 at
1:35 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
hey all, the basic authentication process is done; it hasn't been fully commented nor has error handling been implemented, but the basic functionality is there. try it out if you'd like; instructions are briefly provided in the user documentation part of the google docs.
Also, on the "Project Schedule, Tasks, Work Logs" workbook, I've created an additional spreadsheet called "files". This lists all of our php/html files and info about how they work. specifically, for the ones i did today, it specifies how each page should be accessed (via another page if necessary), what it does, and where it forwards (if anywhere).

I had a little trouble with the php mail function. i probably generated 3 or so confirmation emails and i only got one...45 mins later. any thoughts on this? for now the registration page will display the confirmation url for testing purposes, should you all wish to create accounts.

commenting to hopefully be done this afternoon or tomorrow. feedback is welcome! and questions, too.

Lincoln

----

* Lincoln Waller <lincoln.waller@gmail.com> *   * Tue, Oct 6, 2009 at
1:48 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
and the commenting is done. error handling later. its all on the site and you can access any of the pages at www.opal411.legendaryfox.net <http://www.opal411.legendaryfox.net>
[Quoted text hidden]

----

* Josiah Wang <legendaryfox@gmail.com> *   * Tue, Oct 6, 2009 at 3:38 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Thanks Lincoln. I'll work on templating it and stuff in the next few days; I'm assuming these are relatively final-ish copies, so I can work off them without fear of too much additional changes?

-Josiah
[Quoted text hidden]

----

* Lincoln Waller <lincoln.waller@gmail.com> *   * Tue, Oct 6, 2009 at
3:43 PM *

Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Yup, I'll add the error handling but there are comments where that will go. Had a little trouble FTPing in about an hour ago... I'll try again in a while. Wanted to add a "verification" so email addresses must be valid before they can be submitted. I think that's it though.
[Quoted text hidden]

----
* jamesishere <jamesishere@gmail.com> *   * Tue, Oct 6, 2009 at 6:03 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
I will be working on the data entry function all of tonight and Thursday night if necessary. If I have problems tonight it will give the other implementers a window to assist me.

-James
[Quoted text hidden]

----
* jamesishere <jamesishere@gmail.com> *   * Tue, Oct 6, 2009 at 8:54 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
I am having issues connecting to the database. How do I now access phpmyadmin? opal411.legendaryfox.net <http://opal411.legendaryfox.net> only accesses the public directory now. Were the login details changed?
The error I get on my database code is*:

Warning*: mysql_connect() [function.mysql-connect <http://localhost/function.mysql-connect>]: Access denied for user 'opal411'@'busw037-0b01-dhcp175.bu.edu <http://busw037-0b01-dhcp175.bu.edu>' (using password: YES)

-james
[Quoted text hidden]

----
* Josiah Wang <legendaryfox@gmail.com> *   * Tue, Oct 6, 2009 at 9:13 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Hey James,

The servername should just be' localhost' instead of the 'busw037' etc (this is assuming you're running it off the server; if you're running this on your own WAMP installing, you'll have to change your credentials). The phpmyadmin on the live server is http://opal411mysql.legendaryfox.net <http://opal411mysql.legendaryfox.net>, user/pass = opal411/cs411rocks

call me if there's still a problem, 908-327-5898

-Josiah

[Quoted text hidden]

----

* Josiah Wang <legendaryfox@gmail.com> *   * Tue, Oct 6, 2009 at 9:24 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Hi all,

just for record keeping sakes (no harm in keeping too many records, right?) the decision was to export a temporary copy of the database to james' computer to work off of; issue has been resolved. for future reference, i think if our code may do some major database changing, it might be best/safest to grab fresh copies of all files and export the database (go through phpmyadmin) to your own WAMP installations.

-josiah
[Quoted text hidden]

----

* Lincoln Waller <lincoln.waller@gmail.com> *   * Thu, Oct 8, 2009 at 12:45 AM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Okay, all of the PHP for authentication and account creation is done. I also added a script for users to change their password and an error display page. Let me know your thoughts on these. To use the error display page, use the following code

```
session_start(); //if not already invoked
$_SESSION['error'] = "message here - word 'error' will be pre-appended";
//store the error
header("location:error.php"); //direct to the error page
exit(); //go there now!
```

The mail() function appeared to be working much better tonight. Let me know if ya'll have questions.

Lincoln

P.S. the part that should go at the top of every "protected" page (where a user must be logged in) is:

```
session_start();
if(!($_SESSION['unm'])){
header("location:login.html");
exit();
```

[Quoted text hidden]

[Opal] Meeting Times Tomorrow
6 messages

----
* linger.man@gmail.com <linger.man@gmail.com> *   * Thu, Oct 8, 2009 at 12:37 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
*Reminders for Snapshot #2*

James: Tomorrow I'm meeting with you at 2:00PM in the CS Ugrad Lab. Although it's fine if your error handling is incomplete, make sure that whatever code you have by tomorrow is stable and can run on its own (albeit with errors).

Lincoln: Since you said you'll be in the CS Ugrad Lab I'll tentatively schedule meeting with you for 2:30PM. You're also welcome to sit in with James and my meeting in order for us to work out how the different functions are going to fit together.

Josiah: I'm expecting a link to the templates you've put together. I'd like to have an opportunity to discuss the design choices you've made for the templates; can we schedule a time to chat online about it after you're done with the work? Please have this ready by Friday evening.

Nastia: Like we've discussed before please have your portion of the documentation complete and submitted to me by Saturday evening.

Keep up the good work guys. And thank you to the implementers for logging your work schedules more diligently: it's making our documentation more thorough and our engineering process more legitimate.

----
* Lincoln Waller <lincoln.waller@gmail.com> *   * Thu, Oct 8, 2009 at 4:11 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Do you want to meet me a little before 2? I actually get off work then. If not I'll stick around, no worries.
[Quoted text hidden]

----
* jamesishere <jamesishere@gmail.com> *   * Thu, Oct 8, 2009 at 4:20 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
I'll be there at 2. Also, an update: I got the error checking to work, need to finish up the script to write to the database and I'll be all finished. Was making the forms autofill with the data already in the database required for tomorrow?

-james
[Quoted text hidden]

----

* jamesishere <jamesishere@gmail.com> *   * Thu, Oct 8, 2009 at 4:43 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Also how to I upload my files to the webserver?
[Quoted text hidden]

----

* Lincoln Waller <lincoln.waller@gmail.com> *   * Thu, Oct 8, 2009 at
4:47 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
use winSCP (winscp.net <http://winscp.net>) to connect to
opal411.legendaryfox.net <http://opal411.legendaryfox.net> with the FTP
protocol (not SFTP). uname/pass opal411/cs411rocks
[Quoted text hidden]

----

* linger.man@gmail.com <linger.man@gmail.com> *   * Thu, Oct 8, 2009 at
5:27 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Lincoln: Meeting at 2:30 would work the best for me. See you guys tomorrow!
[Quoted text hidden]

[Opal] PLEASE READ - josiah beginning implementation of template
10 messages

----

* Josiah Wang <legendaryfox@gmail.com> *   * Thu, Oct 8, 2009 at 5:44 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Hi implementers,

Just wanted to let you know that as of right now (Oct 8, 2009, 5:38PM), I'm downloading a snapshot of the code of the server and applying the template to it. I've backed up the files to a backup folder (should be visible in FTP). I'll be applying the templates locally (on my own WAMP server) between now and tomorrow noon; continue to make code changes as you normally would, but if you upload anything, PLEASE also notify me [with which part of the code has changed] so I can apply the changes to the templated versions I have as well; I'll let everyone know when I'm done templating so that work can resume normally.

thanks!

-Josiah

908-327-5898

----
* jamesishere <jamesishere@gmail.com> *   * Thu, Oct 8, 2009 at 5:58 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
I'm 97% finished with my code, just fixing one last thing, but we'll have to make it work with the login code to only access the information of the person whose logged. But we can do that later.

-james
[Quoted text hidden]

----
* Josiah Wang <legendaryfox@gmail.com> *   * Thu, Oct 8, 2009 at 6:00 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Sounds good. I actually just noticed that my domain doesn't seem to be up right now...I've dropped the tech department of my hosting company an email, hopefully it's just scheduled maintenance.

-Josiah
[Quoted text hidden]

----
* Lincoln Waller <lincoln.waller@gmail.com> *   * Thu, Oct 8, 2009 at
6:00 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
James - take a look at "edit_page.php"

On Thu, Oct 8, 2009 at 5:58 PM, jamesishere <jamesishere@gmail.com
<mailto:jamesishere@gmail.com>> wrote:
[Quoted text hidden]

----
* Josiah Wang <legendaryfox@gmail.com> *   * Thu, Oct 8, 2009 at 6:15 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
FYI, domain is back up. Sorry for the inconvenience.

-Josiah
[Quoted text hidden]

----
* jamesishere <jamesishere@gmail.com> *   * Thu, Oct 8, 2009 at 6:43 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
I uploaded my code and I want to check it but I can't seem to get login verification to work. Anyone want to check out edit_page.php and entercheck.php to see if my code is the problem? It should only be the

very top part.
[Quoted text hidden]

----
* jamesishere <jamesishere@gmail.com> *   * Thu, Oct 8, 2009 at 6:45 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Ahhh I see, the hashing of the md5 is ruining the check. I'm going to
upload a new version of logincheck.
[Quoted text hidden]

----
* Lincoln Waller <lincoln.waller@gmail.com> *   * Thu, Oct 8, 2009 at
6:56 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
DB has been updated so that all logins have md5 passwords stored. you
can use lwaller@bu.edu <mailto:lwaller@bu.edu> with password if you need
a login.
or we can create another entry using the admin & reg process if you'd like
[Quoted text hidden]

----
* jamesishere <jamesishere@gmail.com> *   * Thu, Oct 8, 2009 at 7:01 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Everything should work now, try logging in

-james
[Quoted text hidden]

----
* jamesishere <jamesishere@gmail.com> *   * Thu, Oct 8, 2009 at 7:03 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
If you click submit and it doesn't do anything that means you have an
error, scroll up and enter something where the red asterisks say you
must enter something per requirements.
[Quoted text hidden]

[Opal] finished edit pages
10 messages

----
* jamesishere <jamesishere@gmail.com> *   * Thu, Oct 8, 2009 at 6:59 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Everything works fine, except there is some bug with the hashing of
passwords. People who log in get their passwords changed to the hashed

function, and then it doesn't change back to the original. Otherwise, I
had it working before, and if we fix that then we should be all set.

-james

----
* Lincoln Waller <lincoln.waller@gmail.com> * * Thu, Oct 8, 2009 at
7:01 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
they aren't supposed to go back to the original... so i'm not sure i
understand what needs to be fixed
[Quoted text hidden]

----
* jamesishere <jamesishere@gmail.com> * * Thu, Oct 8, 2009 at 7:03 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Sorry, I think you changed it right as I figured out the error
[Quoted text hidden]

----
* jamesishere <jamesishere@gmail.com> * * Thu, Oct 8, 2009 at 7:03 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
There is no error haha, all set now
[Quoted text hidden]

----
* Lincoln Waller <lincoln.waller@gmail.com> * * Thu, Oct 8, 2009 at
7:06 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
awesome!
[Quoted text hidden]

----
* jamesishere <jamesishere@gmail.com> * * Thu, Oct 8, 2009 at 7:13 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
OK I'm done for tonight, send me any bugs you find or cosmetic issues
and I'll take care of them sometime soon. Josiah can take the edit page
and make it look nicer if wants.

-James
[Quoted text hidden]

----
* Josiah Wang <legendaryfox@gmail.com> * * Thu, Oct 8, 2009 at 9:23 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
Implementers,

anyone want to just give me a quick functional walkthrough of the page
progression? just want to make sure my design makes sense in terms of
what to apply to where

thanks!

-josiah
[Quoted text hidden]

----
* Lincoln Waller <lincoln.waller@gmail.com> * * Thu, Oct 8, 2009 at
9:46 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com

James correct me if I'm wrong on any of this.
Administrator goes to admin.html, enters a professor's email address to approve them and the admin password. This is processed by newuser.php, which currently doesn't direct anywhere, just echo's a simple "user added".
Once pre-approved, the professor goes to reg.html and enters his/her email address. This is processed by checkreg.php which finds the registration entry in the confirmation table and sends an email with the confirmation code to the professor.
The professor then clicks the confirmation code link which goes to confirm.php, there they choose a password. The chosen password and confirmation code are passed to create.php which verifies that the passwords match and the confirmation code is correct. If so it updates the entry in the profs table and removes the entry in the confirmation table. It also logs the professor in by registering the email in a session variable. Then they are directed to edit_page.php where they can enter their data.
Login.html is where an already registered professor can go to sign in. The credentials are passed to logincheck.php which find the corresponding entry in the profs table. If found it registers the session variable and redirects to edit_page.php.

I'm not sure on the ins and outs of edit_page.php, but I believe a javascript validates that all fields are properly filled and then passes the data to entercheck.php which updates the table entry.

Logout.php destroys the session and directs the user back to login.html.

Errors are directed to error.php

A user can change their password by going to changepw.html (OH CRAP I DIDN'T WRITE A PASSWORD RESET FUNCTION...do we want that for tomorrow?)
Anyways, changepw.html directs the old password and two instances of the new password to pwscript.php which checks to see that the user is logged in, that the passwords match, and that the old password is correct. If so it updates the entry and displays an approval message.

No one should ever come to rest on the following pages:

logincheck.php
logout.php
create.php
entercheck.php (though I believe currently the error is handled on the
page instead of redirected)

The following pages the user rests on with only a simple message saying
the script completed, so they could be directed somewhere else:
newuser.php
pwscript.php
regcheck.php

Hope this helps - let me know if it needs clarification.

Lincoln
[Quoted text hidden]

----
* jamesishere <jamesishere@gmail.com> * * Thu, Oct 8, 2009 at 10:03 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
That's a very good summary, I don't really have much to add. I'm sure we
can demo it to Chris easily tomorrow.

However, I will comment my code more to make it easier to understand. I
was learning as I was coding so it came about organically, which is
always confusing.

The other thing that we need is a "logout" button on every page.
Otherwise, next week should be about refinement and cosmetics. My
javascript function could be cleaned up into a loop, but that is
unnecessary as the user will see the same effect.

'Entercheck' is a misnomer from an earlier version of my code, but I
don't see a problem with leaving it the same name even though it doesn't
'check' anymore.

-james
[Quoted text hidden]

----
* Josiah Wang <legendaryfox@gmail.com> * * Thu, Oct 8, 2009 at 10:16 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com
thanks guys, great coding! thanks for the summaries too!

in terms of display functions, not much has changed right? we're just
spitting out all the info as we did before?

-Josiah

[Opal] [Opal][Implementers]SEMI-URGENT - question about coding choice
6 messages

----

* Josiah Wang <legendaryfox@gmail.com> *   * Thu, Oct 8, 2009 at 9:40 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com

Hi implementers,

I'm going through the code right now, and I have a suggestion to make; this comes from my experience in working with templating my code pages and making it easier to maintain, so hear me out.

One thing I noticed is that the database name, host, password, user, etc, is all hardcoded into each page; what i've done in the past is keep all the values (DBNAME, DBHOST, DBUSER, DBPASS) into a file i called config.php; because a lot of my work had to be ported across multiple servers, if i needed to change this, i'd simply change it in config.php; all the other pages needed to do is simply include it.

another thing that i found that making maintenance easier is keeping a simple 'templating' function. so if you look at the anatomy of an outputted php page, all text tends to be centered around a certain area in the code; be it in the <body> sections, or inside a <td> cell (as my current design has it). What i propose (and am testing out on my own WAMP) is a simple dispBegin and dispEnd() function that more or less draws the beginning and ending HTML for the 'template', leaving the center body (in my design case, the <td> tag) open to the developer.

are we okay with this implementation? i've made the changes on my own latest copies (i just re-updated my local files with james's/lincoln's stuff, thanks), so it's working perfectly; however, i just wanted to get the okay from everyone. from my point of view and experience, this speeds up development and makes maintenance a tad easier

-josiah

----

* Josiah Wang <legendaryfox@gmail.com> *   * Thu, Oct 8, 2009 at 9:41 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com

Also, there seem to be some .html pagse; apparently apache ignores PHP tags in HTML pages? i'm proposing to rename them to .php instead (there won't be adverse effects, .php pages can have plain html and no PHP code as well; it'll also look consistent)

-Josiah

----

* Lincoln Waller <lincolnwaller@gmail.com> *   * Thu, Oct 8, 2009 at 9:51 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com

Yeah apache ignores php in .html pages. When I created them they were just simply to collect data to pass to the php pages I was working on so feel free to modify them however. Sounds good with the config.php. Is this something you'd like each of us to do (and if so does it need to happen by tomorrow), or will you just drop it in as you go through each template page?

Lincoln

----
* Josiah Wang <legendaryfox@gmail.com> *   * Thu, Oct 8, 2009 at 9:57 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com

Nope, I've more or less taken care of it on my local copies. Had spare time....I'll test them out live in a few minutes, just a few more thigns to hammer out. It should be live by tonight.

James, would you be okay with it?

-Josiah

----
* jamesishere <jamesishere@gmail.com> *   * Thu, Oct 8, 2009 at 10:06 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com

Those are very good suggestions, and I appreciate you implementing them. You should add the same header/footer to every page, like the login and edit pages.

-James

----
* Josiah Wang <legendaryfox@gmail.com> *   * Thu, Oct 8, 2009 at 10:18 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com

Cool, I'll commit these changes tonight then. should've doen this earlier, sorry about the rush [if there was any]. i'll send out an email accompanying the changes and what the coding implications are.

-josiah

[Opal] Template Implementation finished -
4 messages

----

* Josiah Wang <legendaryfox@gmail.com> * * Thu, Oct 8, 2009 at 10:48 PM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com

Hi all,

I've committed my changes to the live version - everything should be up. I made backups in their approriately named folders.

The style is more or less what the http://people.bu.edu/username looks like; does it look too similar? i can have another design drawn up if we wish to.

in terms of actual coding changes, basically, if you want to make database calls, instead of explicitly calling the username/pass/whatnot, here's how i have it

```
mysql_connect($DBHOST, $DBUSER, $DBPASS)
mysql_select_db($DBNAME);
```

all these are stored inside the config.php; as such, i've made a file, inchook.php (for 'include hook') to link up that and any future files we may need to include. currently, inchook.php brings in config.php and dispFuncs.php

dispFuncs.php houses more or less just the dispBegin($title) and dispEnd() functions. these are basically echo-ers of the start and ending HTML of the template;

i think that about sums it up.

one thing i noticed, for edit_page.php, we don't have a field for the professor's actual name?

-josiah

----

* jamesishere <jamesishere@gmail.com> * * Fri, Oct 9, 2009 at 8:15 AM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com

Hi, I'm at work right now so I can't write a long message, but the template looks really good, excellent work! I will be at the meeting at *2:30* as previously discussed.

-James

----

* linger.man@gmail.com <linger.man@gmail.com> * * Fri, Oct 9, 2009 at 9:11 AM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com

Sounds like you guys have done a great job! No joke--I'm looking forward to today's demonstration(s).

Just a quick note on the times, I thought James was going to be coming in at 2:00PM. But coming in at 2:30PM is fine if Lincoln is fine with it.

Lincoln, did you want to move your time slot up to 2:00PM? Or do you want to just sort of squeeze in somewhere between 2:30PM and 3:15PM, since Josiah is coming in at 3:15.

----
* Lincoln Waller <lincoln.waller@gmail.com> * * Fri, Oct 9, 2009 at 9:20 AM *
Reply-To: 411opal@googlegroups.com
To: 411opal@googlegroups.com

2pm is great!

```
-- phpMyAdmin SQL Dump
-- version 2.11.9.4
-- http://www.phpmyadmin.net
--
-- Host: opal411mysql.legendaryfox.net
-- Generation Time: Oct 26, 2009 at 10:22 AM
-- Server version: 5.0.67
-- PHP Version: 5.2.9

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Database: `opal411`
--

-- --------------------------------------------------------

--
-- Table structure for table `confirmation`
--

CREATE TABLE IF NOT EXISTS `confirmation` (
  `user` varchar(64) NOT NULL,
  `conf` varchar(32) NOT NULL,
  `id` int(10) unsigned NOT NULL auto_increment,
  PRIMARY KEY  (`id`),
  UNIQUE KEY `user` (`user`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;

--
-- Dumping data for table `confirmation`
--


-- --------------------------------------------------------

--
-- Table structure for table `profs`
--

CREATE TABLE IF NOT EXISTS `profs` (
  `id` int(5) unsigned NOT NULL auto_increment COMMENT 'unique ID',
  `email` varchar(30) NOT NULL COMMENT 'Email Address',
  `status` enum('a','v','t','i') NOT NULL default 'a' COMMENT 'Status',
  `department`
enum('AN','AR','AH','AS','BI','CH','CL','CN','CS','ES','EC','EN','GE','HI','IR','MA','MU','PH','PY','P
O','PS','RN','RS','SO') default NULL COMMENT 'Department',
  `img` varchar(256) default NULL COMMENT 'Photo URL',
  `name` varchar(64) default NULL COMMENT 'Full Name',
  `addr` varchar(256) default NULL COMMENT 'Office Num/Addr',
  `phone` varchar(14) default NULL COMMENT 'Office Phone Num',
  `bio` blob COMMENT 'Biography / CV summary',
  `CV` varchar(128) default NULL COMMENT 'CV URL (pdf)',
  `crs1` varchar(32) default NULL COMMENT 'course taught 1',
  `lnk1` varchar(128) default NULL COMMENT 'URL course 1',
  `sem1` date default NULL COMMENT 'end date course 1',
  `crs2` varchar(32) default NULL COMMENT 'course taught 2',
  `lnk2` varchar(128) default NULL COMMENT 'URL course 2',
  `sem2` date default NULL COMMENT 'end date course 2',
  `crs3` varchar(32) default NULL COMMENT 'course taught 3',
  `lnk3` varchar(128) default NULL COMMENT 'URL course 3',
  `sem3` date default NULL COMMENT 'end date course 3',
  `crs4` varchar(32) default NULL COMMENT 'course taught 4',
  `lnk4` varchar(128) default NULL COMMENT 'URL course 4',
  `sem4` date default NULL COMMENT 'end date course 4',
  `oldcourse` blob COMMENT 'previous courses',
  `pwh` varchar(64) default NULL COMMENT 'hint for lost password',
  `pwa` varchar(32) default NULL COMMENT 'challenge answer for lost password',
  `pw` varchar(32) default NULL COMMENT 'password',
  `officeh` varchar(32) default NULL,
  `research` blob,
  `publications` blob COMMENT 'publications',
  `events` blob,
  `students` blob,
  `links` blob,
  `awards` blob,
  `orgs` blob COMMENT 'groups/organizations',
  PRIMARY KEY  (`id`),
  UNIQUE KEY `email` (`email`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COMMENT='entries for profs websites'
AUTO_INCREMENT=1 ;

--
-- Dumping data for table `profs`
--
```

HTML and PHP File Index

Page	Come From	Expected Args	Description	Go To
admin_changeactive.php	admin_index.php	POST uid	Toggles the active/inactive status of a given professor	admin_index.php
admin_changeemail.php	admin_index.php	POST uid	Interface to change a professor's email	admin_changeemail2.php
admin_changeemail2.php	admin_changeemail.php	POST oldemail, newemail	Executes changes to a professor's email	admin_index.php
admin_create.php	admin_index.php	none	Interface to add a professor to the valid list	admin_create2.php
admin_create2.php	admin_create.php	POST email, pw	Executes the addition of an email user to the system	admin_index.php
admin_deleteuser.php	admin_index.php	POST unm	Deletes all records of unm from the system	admin_index.php
admin_index.php	login.php (with correct creditials)	none	admin control panel	multiple admin pages (this is a hub)
admin_resetpass.php	admin_index.php	POST unm	resets the password for professor with unm to a random password	admin_index.php
changepw.php	mainaccount.php	none	interface for a professor to change their password	pwscript.php
config.php	none	none	server configuration file	none
confirm.php	anywhere (typically from a link in an email the professor receives)	POST confirm	interface for validating and verifying a professor creating an account, given the correct 'confirm' code	create.php
create.php	confirm.php	POST conf, pw, pwb	Verifies the confirmation code, and that passwords match. Immediately goes to edit_page.php for account creation	edit_page.php
dispFuncs.php	none	none	library of functions for displaying templates	none
edit_page.php	confirm.php; mainaccount.php	POST multiple	main page for editing. uses entercheck.php for backend verification	multiple

email_help.php	mainaccount.php	none	Basic information web page regarding emailing an administrator	none
entercheck.php	edit_page.php	POST multiple	executes the backend verification of things in edit_page.php. returns back to edit_page.php	edit_page.php
error.php	multiple	$_SESSION[error]	Returns error types to the user when something invalid is executed.	multiple
errorFuncs.js	none	none	Javascript file to provide error checking on edit_page.php	none
forgot.php	login.php	none	interface to enter your email to begin password reset process	forgot2.php
forgot2.php	forgot.php	POST email	Interface to enter answer to password hint, and choose a new password	forgpw.php
forgpwd.php	forgot2.php	POST pwha, pw, pwb	verifies answer to password hint, and updates the password	login.php
inchook.php	none	none	a configuration file 'hooks' with other configuration files	none
index.php	mulitple	none	basic index; directory listing of all professors in the system	multiple (profpage.php)
login.php	user invoked; index.php	none	interface for a professor to log in	logincheck.php
logincheck.php	login.php	POST email, pwd	verifies login information	mainaccount.php; admin_index.php (if authorized)
logout.php	multiple	$_SESSION	logs a user out	none
mainaccount.php	multiple	none	account interface for a logged in professor to select actions	multiple
profpage.php	index.php, mainaccount.php	POST unm	displays to a user the information for a given professor	multiple
pwscript.php	changepw.php	POST unm, old, newa, newb	compares passwords from changepw.php, and verifies and changes	mainaccount.php

reg.php	admin_index.php	none	interface for admin to add a user to the system	regcheck.php
regcheck.php	reg.php	POST email	adds the email from reg.php to the system	admin_index.php

```
<?php

include ('./inchook.php');


session_start();
$unm=$_SESSION['unm'];
if($unm!="admin")
{
	header("location:error.php");
	//print "FAIL";
	exit();
}


dispBegin("ADMIN : Toggle Activity");


$conn = mysql_connect($DBHOST,$DBUSER,$DBPASS);
mysql_select_db($DBNAME) or die("Unable to select database!");


$unm = $_REQUEST['uid'];

$req_email = $unm . '@bu.edu';

$sql = "SELECT status FROM profs WHERE email = '$req_email'"; //query

$res = mysql_query($sql); //store the mysql resource handler

if (!$res) {
	die('Invalid query: ' . mysql_error());
}

$res_array = mysql_fetch_array($res); //format the resource handler into an array that php can use
extract($res_array);

if($status == 't' || $status == 'v')
{
	//currently active - let's mark inactive
	$new_status = "inactive";
	$sql = "UPDATE profs SET status = 'i' WHERE email = '$req_email'";
} else {
	//currentl inactive - mark as active
	$new_status = "active";
	$sql = "UPDATE profs SET status = 't' WHERE email = '$req_email'";
}

$res = mysql_query($sql);

print "Status Updated. $unm is now marked as $new_status";
print "<br><br>Return to <a href='admin_index.php'>admin main page</a>.<br><br><br>\n";
dispEnd();




?>
```

```
<?php

include ('./inchook.php');


session_start();
$unm=$_SESSION['unm'];
if($unm!="admin")
{
        header("location:error.php");
        //print "FAIL";
        exit();
}



$conn = mysql_connect($DBHOST,$DBUSER,$DBPASS);
mysql_select_db($DBNAME) or die("Unable to select database!");


$unm = $_REQUEST['uid'];

$req_email = $unm . '@bu.edu';



dispBegin("ADMIN : Change email for $req_email");

?>

<form action="admin_changeemail2.php" method="POST">
Enter a new email for <?php echo $req_email; ?>: <input name="newemail" type="text" value='<?
php echo $req_email; ?>'><br>
<input type="hidden" name="oldemail" value='<?php echo $req_email; ?>'>
<input type="submit" value="Change Email">
</form>

<?php

print "<br><br>Return to <a href='admin_index.php'>admin main page</a>.<br><br><br>\n";

dispEnd();

?>
```

```php
<?php

include ('./inchook.php');


session_start();
$unm=$_SESSION['unm'];
if($unm!="admin")
{
        header("location:error.php");
        //print "FAIL";
        exit();
}



$conn = mysql_connect($DBHOST,$DBUSER,$DBPASS);
mysql_select_db($DBNAME) or die("Unable to select database!");


$req_email = $_POST['oldemail'];
$newemail = $_POST['newemail'];



dispBegin("ADMIN : Change email for $req_email");





$conn = mysql_connect($DBHOST,$DBUSER,$DBPASS);
mysql_select_db($DBNAME) or die("Unable to select database!");


$sql = "UPDATE profs SET email = '$newemail' WHERE email = '$req_email'";
mysql_query($sql);

print "Changed email for $req_email to $newemail.";


print "<br><br>Return to <a href='admin_index.php'>admin main page</a>.<br><br><br>\n";

dispEnd();

?>
```

```
<?php
session_start();

$unm=$_SESSION['unm'];
if ($unm!="admin"){
header("location:login.php");
exit();
}

include('./inchook.php');

dispBegin("Admin Page");

?>

Enter the email address to approve and the admin password.<br>
<form name="addusr" method="POST" action="admin_create2.php">
Email: <input type="text" name="email"><br>
Admin Password: <input type="password" name="pw"><br>
<input type="submit" name="submit">
</form>

<?php

print "<br><br>Return to <a href='admin_index.php'>admin main page</a>.<br><br><br>\n";

dispEnd();

?>
```

```
<?php
include('./inchook.php');



session_start();

$unm=$_SESSION['unm'];
if($unm!="admin"){
        header("location:login.php");
exit();
}

//session_start();

dispBegin("New User");
//get posted variables
$usr=$_POST['email'];
$pw =$_POST['pw'];

//check email & password
if(!$usr || !filter_var($usr, FILTER_VALIDATE_EMAIL)){
  $_SESSION['error'] = "Email Address Not Vaild";
  header("location:error.php");
  exit();
}

if(strcmp($pw,$ADMINPASS)!=0){
  $_SESSION['error'] = "Invalid Admin Password";
  header("location:error.php");
  exit();
}

//add user to profs table as "approved"
//add user to confirmation with a random confirmation code
echo "Adding $usr ";
mysql_connect($DBHOST, $DBUSER, $DBPASS);
mysql_select_db($DBNAME); //select the database


$conf=md5(uniqid(rand()));

//do this first to check for dupicate entry--insert fails b/c 'email' is unique.
$sql="INSERT INTO profs (id, email, status) VALUES (NULL, '$usr', 'a')";
$res=mysql_query($sql);

//error handling
if(!$res){
  $_SESSION['error'] = "Inserting Main Table Record";
  header("location:error.php");
  exit();
}
```

```
$sql="INSERT INTO confirmation (user, conf, id) VALUES ('$usr', '$conf', NULL)";
$res=mysql_query($sql);


//now we need to mail the pass over to the guy
//mail('$email', 'Your Confirm link' , "Please go to <a href='$LOCALDOMAIN/create.php?
conf=$conf'>here</a> to finish registration.");
//print "(THIS WOULD NORMALLY BE EMAILED) Please go to <a
href='$LOCALDOMAIN/create.php?conf=$conf'>here</a> to finish registration.";

//error handling
if(!$res){
  $_SESSION['error'] = "Inserting Confirmation Record";
  header("location:error.php");
  exit();
}

echo " successfully";

print "<br><br>Return to <a href='admin_index.php'>admin main page</a>.<br><br><br>\n";

dispEnd();



?>
```

```
<?php

include ('./inchook.php');


session_start();
$unm=$_SESSION['unm'];
if($unm!="admin")
{
        header("location:error.php");
        //print "FAIL";
        exit();
}


dispBegin("ADMIN : Delete user");


$conn = mysql_connect($DBHOST,$DBUSER,$DBPASS);
mysql_select_db($DBNAME) or die("Unable to select database!");


$unm = $_REQUEST['uid'];

$req_email = $unm . '@bu.edu';


$sql = "DELETE FROM profs WHERE email = '$req_email' LIMIT 1";
mysql_query($sql);

print "All records to $req_email have now been deleted.";



print "<br><br>Return to <a href='admin_index.php'>admin main page</a>.<br><br><br>\n";
dispEnd();











?>
```

```
<?php

include ('./inchook.php');


session_start();
$unm=$_SESSION['unm'];
if($unm!="admin")
{
        header("location:error.php");
        //print "FAIL";
        exit();
}


dispBegin("ADMIN : List of Professors");


$conn = mysql_connect($DBHOST,$DBUSER,$DBPASS);
mysql_select_db($DBNAME) or die("Unable to select database!");

$sql = "SELECT name, email, status FROM profs";
$res = mysql_query($sql); //store the mysql resource handler

if (!$res) {
        die('Invalid query: ' . mysql_error());
}

/*
$res_array = mysql_fetch_array($res); //format the resource handler into an array that php can use
extract($res_array);
print_r($res_array);*/

$profCounter=0;
while($row = mysql_fetch_array($res))
{
        extract($row);
        $dispname[$profCounter] = $name;
        $dispemail[$profCounter] = str_replace('@bu.edu', '', $email);
        $actualemail[$profCounter] = $email;
        $activeState[$profCounter] = $status;
        $profCounter++;
}

print "<h2>Admin: List of Boston University's CS Department Professors</h2><br><br>\n";

print "<table>\n";




for($counter=0; $counter<$profCounter; $counter++)
```

```
{
        //print "<a href='profpage.php?
uid=$dispemail[$counter]'>$dispname[$counter]</a><br>\n";

        print "<tr>";
        print "<td>$actualemail[$counter] ($dispname[$counter]) </td>";
        if($activeState[$counter] == 't' || $activeState[$counter] == 'v') //if active
        {
                $txt_active = 'Mark Inactive';
        }
        else
        {
                $txt_active = 'Mark Active';
        }


        print "<td><a href='admin_changeactive.php?
uid=$dispemail[$counter]'>$txt_active</a></td>";
        print "<td><a href='admin_deleteuser.php?uid=$dispemail[$counter]'>Delete</a></td>";
        print "<td><a href='admin_changeemail.php?uid=$dispemail[$counter]'>Change
Email</a></td>";
        print "<td><a href='admin_resetpass.php?uid=$dispemail[$counter]'>Reset
Password</a></td>";
        print "</tr>\n";
}
print "</table>\n\n\n";

print "<br><br>\n";

print "Create a new user <a href='admin_create.php'>here</a>.";


dispEnd();

?>
```

```
<?php

include ('./inchook.php');


session_start();
$unm=$_SESSION['unm'];
if($unm!="admin")
{
        header("location:error.php");
        //print "FAIL";
        exit();
}


dispBegin("ADMIN : Reset Password");


$conn = mysql_connect($DBHOST,$DBUSER,$DBPASS);
mysql_select_db($DBNAME) or die("Unable to select database!");


$unm = $_REQUEST['uid'];

$req_email = $unm . '@bu.edu';

$newpass = rand_str();
$md5pass = md5($newpass);



$sql = "UPDATE profs SET pw = '$md5pass' WHERE email = '$req_email' LIMIT 1";
mysql_query($sql);

//mail($req_email, 'New Password for CS Login', "Hello.\n The administrator has reset your
password. Your new password is $newpass.");


print "Successfully reset password. $unm's password is now $newpass.";







// Generate a random character string
function rand_str($length = 32, $chars =
'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890')
{
   // Length of character list
   $chars_length = (strlen($chars) - 1);
```

```
    // Start our string
    $string = $chars{rand(0, $chars_length)};

    // Generate random string
    for ($i = 1; $i < $length; $i = strlen($string))
    {
        // Grab a random character from our list
        $r = $chars{rand(0, $chars_length)};

        // Make sure the same two characters don't appear next to each other
        if ($r != $string{$i - 1}) $string .=  $r;
    }

    // Return the string
    return $string;
}
print "<br><br>Return to <a href='admin_index.php'>admin main page</a>.<br><br><br>\n";

dispEnd();


?>
```

```
<?php

include('./inchook.php');

session_start();
if(!($_SESSION['unm'])){
  header("location:login.php");
  exit();
}

dispBegin("Change Password");

?>

<br><br>
Enter old password and a new one twice:<br>
<form name="updatepass" method="POST" action="pwscript.php">
Old Password: <input type="password" name="old"><br>
New Password: <input type="password" name="new1"> <br>
New Password: <input type="password" name="new2">
<br>
<input type="submit" name="submit">
</form>

<br><br>


<?php

print "<br><br>Return to <a href='mainaccount.php'>main account page</a>.<br><br><br>\n";

dispEnd();

?>
```

```php
<?php
//database configuration values.
$DBNAME = 'opal411';
$DBUSER = 'opal411';
$DBPASS = 'cs411rocks';
$DBHOST = 'opal411mysql.legendaryfox.net';

$LOCALDOMAIN = 'http://opal411.legendaryfox.net'; //no ending slash.


$ADMINPASS = 'cs411rocks';

?>
```

```
<?php

include('./inchook.php');

dispBegin("Confirm");

?>



Select a password and re-enter it for verification.<br>
<form name="selectPW" method=POST action="create.php">
Password: <input type="password" name="pw1"><br>
Password: <input type="password" name="pw2"><br>
<input type="hidden" name="conf" value="<?php echo $_GET['confirm']; ?>">
<input type="submit" name="submit">
</form>


<?php

dispEnd();

?>
```

```
<?php

include('./inchook.php');



session_start();
//connect to the database
mysql_connect($DBHOST, $DBUSER, $DBPASS);
mysql_select_db($DBNAME);

//get posted variables
$conf = $_POST['conf'];
$pw = $_POST['pw1'];
$pwb= $_POST['pw2'];

//check posted variables

//check if passwords match
if(strcmp($pw,$pwb)!=0){
  $_SESSION['error']="Passwords do not Match";
  header("location:error.php");
  exit();
}

//find confirmation code in confirmation table
$sql = "SELECT * FROM confirmation WHERE conf='$conf'";
$res = mysql_query($sql); //store the mysql resource handler

//error handling if confirmation code not valid
if(count($res)!=1){
  $_SESSION['error']="Invalid Confirmation Code.  Please contact system administrator.";
  header("location:error.php");
  exit();
}


//get username
$row=mysql_fetch_array($res);
$usr=$row['user'];

echo "$usr";

//hash password
$pw = md5($pw);

//update the profs table with the password, remove entry from confirmation table
$sql = "UPDATE profs SET pw='$pw',status='v' WHERE email='$usr'";
$res = mysql_query($sql);

//error handling
if(!$res){
  $_SESSION['error']="Upating Record.";
  header("location:error.php");
  exit();
}

$sql = "DELETE FROM confirmation WHERE user='$usr'";
$res = mysql_query($sql);

//log user in and direct them to page editing
$_SESSION['unm'] = $usr;
header("location:edit_page.php");
?>
```

```
<?php


function dispBegin($title)
{

echo '
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<link rel="stylesheet" type="text/css" href="./mainstyle.css" />
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>';
echo $title;

echo '</title>
</head>

<body>
<center>
<table width="760" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td><img src="images/bu-header.gif" width="630" height="34" /></td><td><a
href="logout.php">Logout</a></td>
  </tr>
  <tr>
    <td colspan=2 style="background-image:url(images/fillerbar.png); border-style:solid; border-
width:thin; border-color:#999">&nbsp;</td>
  </tr>
  <tr>
    <td colspan=2 style="border-style:solid; border-width:thin; border-color:#999; vertical-align: top;
padding: 5px">
            ';

            }


function dispEnd()
{


//$nowtime = time();

$dispdate = date("l, g:i Y-m-d. ");

echo '
</td>
  </tr>
  <tr>
    <td colspan=2><img src="images/footer.gif" width="760" height="61" /></td>
  </tr>
</table>
<p>';

echo $dispdate;
//HH:MM YYYY-MM-DD.
echo '(C) Team Opal (CS411)</p>
</center>';

//readfile ('./errorFuncs.js');


echo '









</body>
</html>
';

}
```

```
<?php

include('./inchook.php');

dispBegin("Edit Page");


session_start();
if(!($_SESSION['unm'])){
header("location:login.php");
exit();
}
else{
$unm=$_SESSION['unm'];
//echo "login successful $unm";
}



$conn = mysql_connect($DBHOST,$DBUSER,$DBPASS);
mysql_select_db($DBNAME) or die("Unable to select database!");

$result =  mysql_query("SELECT * FROM profs WHERE email='$unm'");
// get the row of values
$row = mysql_fetch_row($result);

mysql_close($conn);
$pid = str_replace('@bu.edu', ", $unm);

//print "<a href='$LOCALDOMAIN/profpage.php?uid=$pid'>View Page</a><br><br><a
href='$LOCALDOMAIN/changepw.php'>Change Password</a><br><br>\n\n";

?>



<table>
<tr><td>

<form name= "submit" action="entercheck.php" method="post">

<!-- Each item antered has it's name, id number for functions, and an invisible red asterisk next to it
that only appears if it is required and incorrectly entered -->

<!-- This is hidden user id -->

<input type="hidden" name="userid" value="<?php echo $unm; ?>" />

<b>Name:</b> <input type="text" name="myname" id="28" size="40" value="<?php echo
$row[5]; ?>" /> <b id="Emyname" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />
```

```
<b>Office Address:</b> <input type="text" name="offaddress" id="1" size="40" value="<?php
echo $row[6]; ?>" /> <b id="Eoffaddress" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />
<b>Office Phone Number:</b> <input type="text" id="2" name="offphone" value="<?php echo
$row[7]; ?>" /> <b id="Eoffphone" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />
<b>Biography / CVSummary:</b> <br /><textarea rows="3" cols="40" name="cvsummary"
id="3"><?php echo $row[8]; ?></textarea> <b id="Ecvsummary" style="display: none"><font
color="red" size="2" face="Verdana">*</font></b>
<br /> <br />
<b>CV Link (URL):</b> <input type="text" name="cvurl" id="4" size="40" value="<?php echo
$row[9]; ?>" /> <b id="Ecvurl" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />
<b>First Course Taught:</b> <input type="text" name="crs1" id="5" size="8" value="<?php echo
$row[10]; ?>" /> <b id="Ecrs1" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />
<b>Course URL:</b> <input type="text" name="crsurl1" id="6" size="30" value="<?php echo
$row[11]; ?>" /> <b id="Ecrsurl1" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />
<b>1st end date (YYYY-MM-DD):</b> <input type="text" name="end1" id="7" size="10"
value="<?php echo $row[12]; ?>" /> <b id="Eend1" style="display: none"><font color="red"
size="2" face="Verdana">*</font></b>
<br /> <br />
Second Course Taught: <input type="text" name="crs2" id="8" size="8" value="<?php echo
$row[13]; ?>" /> <b id="Ecrs2" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />
Course URL: <input type="text" name="crsurl2" id="9" size="30" value="<?php echo $row[14]; ?
>" /> <b id="Ecrsurl2" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />
2nd end date (YYYY-MM-DD): <input type="text" name="end2" id="10" size="10" value="<?php
echo $row[15]; ?>" /> <b id="Eend2" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />
Third Course Taught: <input type="text" name="crs3" id="11" size="8" value="<?php echo
$row[16]; ?>" /> <b id="Ecrs3" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />
Course URL: <input type="text" name="crsurl3" id="12" size="30" value="<?php echo
$row[17]; ?>" /> <b id="Ecrsurl3" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />
3rd end date (YYYY-MM-DD): <input type="text" name="end3" id="13" size="10" value="<?php
echo $row[18]; ?>" /> <b id="Eend3" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />
```

```
Fourth Course Taught: <input type="text" name="crs4" id="14" size="8" value="<?php echo
$row[19]; ?>" /> <b id="Ecrs4" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />
Course URL: <input type="text" name="crsurl4" id="5" size="30" value="<?php echo $row[20]; ?
>" /> <b id="Ecrsurl4" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />
4th end date (YYYY-MM-DD): <input type="text" name="end4" id="16" size="10" value="<?php
echo $row[21]; ?>" /> <b id="Eend4" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />
List Old Courses: <br /><textarea rows="3" cols="40" name="oldcrs" id="17"><?php echo
$row[22]; ?></textarea> <b id="Eoldcrs" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />
<b>Enter Office Hours:</b> <input type="text" name="offhours" id="20" size="40" value="<?php
echo $row[26]; ?>" /> <b id="Eoffhours" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />
Enter any research: <br /><textarea rows="3" cols="40" name="research" id="21"><?php echo
$row[27]; ?></textarea> <b id="Eresearch" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />
List Published Work: <br /><textarea rows="3" cols="40" name="pubs" id="22"><?php echo
$row[28]; ?></textarea> <b id="Epubs" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />
List Events: <br /><textarea rows="3" cols="40" name="events" id="23"><?php echo $row[29]; ?
></textarea> <b id="Eevents" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />
List PhD Students: <br /><textarea rows="3" cols="40" name="students" id="24"><?php echo
$row[30]; ?></textarea> <b id="Estudents" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />
List Any Links: <br /><textarea rows="3" cols="40" name="links" id="25"><?php echo
$row[31]; ?></textarea> <b id="Elinks" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />
List Awards: <br /><textarea rows="3" cols="40" name="awards" id="26"><?php echo $row[32]; ?
></textarea> <b id="Eawards" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />
List Groups/Organizations: <br /><textarea rows="3" cols="40" name="groups" id="27"><?php
echo $row[33]; ?></textarea> <b id="Egroups" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />

<b>Password Question:</b> <input type="text" name="pwh" id="28" size="40" value="<?php
echo $row[23]; ?>" />
<br /> <br />
```

```
<b>Answer</b> <input type="text" name="pwha" id="28" size="40" value="<?php echo
$row[24]; ?>" /> <b id="Emyimage" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />


<b>URL to Image:</b> <input type="text" name="myimage" id="28" size="40" value="<?php
echo $row[4]; ?>" /> <b id="Emyimage" style="display: none"><font color="red" size="2"
face="Verdana">*</font></b>
<br /> <br />

<A href="javascript: submitform()">Submit</A>

</form>



<!-- does this to determine whether to submit after error checking-->
 <SCRIPT language="JavaScript">
function submitform()
{
	if(check() == true){
		document.submit.submit();
		}
}
</SCRIPT>

<!-- javascript error checking routine-->
<script type='text/javascript'>

function check(){
	// correct determines if the form is submitted, which only stays true if there are no errors
	var correct = true;
	// test holds the text element inputted by the user
	var test = document.getElementById('1')
	// if nothing is inputted, set correct to false and make it have a red asterisk
	if(test.value.length == 0){
		error('Eoffaddress', "");
		correct = false;
	}
	// if no error then remove any asterisk that was there
	else {
		error('Eoffaddress', "none");
		}
	// repeat for each data element
	var test = document.getElementById('2')
	if(test.value.length == 0){
		error('Eoffphone', "");
		correct = false;
	}
	else {
		error('Eoffphone', "none");
		}
var test = document.getElementById('3')
if(test.value.length == 0){
		error('Ecvsummary', "");
		correct = false;
}
else {
		error('Ecvsummary', "none");
		}
var test = document.getElementById('4')
if(test.value.length == 0){
		error('Ecvurl', "");
		correct = false;
}
else {
		error('Ecvurl', "none");
		}
var test = document.getElementById('5')
if(test.value.length == 0){
		error('Ecrs1', "");
		correct = false;
}
else {
		error('Ecrs1', "none");
		}
var test = document.getElementById('5')
if(test.value.length == 0){
		error('Ecrs1', "");
		correct = false;
}
else {
		error('Ecrs1', "none");
		}
var test = document.getElementById('6')
if(test.value.length == 0){
		error('Ecrsurl1', "");
		correct = false;
}
else {
		error('Ecrsurl1', "none");
		}
var test = document.getElementById('7')
if(test.value.length == 0){
		error('Eend1', "");
		correct = false;
}
else {
		error('Eend1', "none");
		}
var test = document.getElementById('20')
if(test.value.length == 0){
```

```
                error('Eoffhours', "");
                correct = false;
        }
        else {
                error('Eoffhours', "none");
        }
        var test = document.getElementById('28')
        if(test.value.length == 0){
                error('Emyname', "");
                correct = false;
        }
        else {
                error('Emyname', "none");
        }
        return correct;
        }


</script>

<!-- js script to toggle red error asterisk-->
<script language="javascript">
function error(location, set) {
target = document.getElementById(location);
target.style.display=set;

}
//-->
</script>

</td></tr></table>

<?php


print "<br><br>Return to <a href='mainaccount.php'>main account page</a>.<br><br><br>\n";

dispEnd();

?>
```

```
<?php


include ('./inchook.php');

dispBegin("Why would I email the administrator?");


?>
```

Email the administrator if you wish to ask about technical questions, or have a request (for example, to mark your account as inactive, or to change your email).

```
<?php

dispEnd();

?>
```

```
<?php

session_start();
if(!($_SESSION['unm'])){
header("location:login.php");
exit();
}
else{
$unm=$_SESSION['unm'];
}

include('./inchook.php');

dispBegin("Enter Check");


// retrieves the entry in forms
/*$user = "opal411";
$password = "cs411rocks";
$database = "opal411";
$server = "opal411mysql.legendaryfox.net";*/ //obsoleted by config.php - 2009.10.08 21:53 Josiah
$unm = $_POST['userid'];
// location of form input page for redirection
$redirect = "$LOCALDOMAIN/edit_page.php";



$myname = addslashes($_POST['myname']);
$myimage = addslashes($_POST['myimage']);
$offaddress = addslashes($_POST['offaddress']);
$offphone = addslashes($_POST['offphone']);
$cvsummary = addslashes($_POST['cvsummary']);
$cvurl = addslashes($_POST['cvurl']);
$offaddress = addslashes($_POST['offaddress']);
$crs1 = addslashes($_POST['crs1']);
$crsurl1 = addslashes($_POST['crsurl1']);
$end1 = addslashes($_POST['end1']);
$crs2 = addslashes($_POST['crs2']);
$crsurl2 = addslashes($_POST['crsurl2']);
$end2 = addslashes($_POST['end2']);
$crs3 = addslashes($_POST['crs3']);
$crsurl3 = addslashes($_POST['crsurl3']);
$end3 = addslashes($_POST['end3']);
$crs4 = addslashes($_POST['crs4']);
$crsurl4 = addslashes($_POST['crsurl4']);
$end4 = addslashes($_POST['end4']);
$oldcrs = addslashes($_POST['oldcrs']);
$pwh = addslashes($_POST['pwh']);
$pwha = addslashes($_POST['pwha']);
$offhours = addslashes($_POST['offhours']);
$research = addslashes($_POST['research']);
$pubs = addslashes($_POST['pubs']);
```

```
$events = addslashes($_POST['events']);
$students = addslashes($_POST['students']);
$links = addslashes($_POST['links']);
$awards = addslashes($_POST['awards']);
$groups = addslashes($_POST['groups']);


print "RESEARCH IS NOW $research compared to $_POST[research]";
$conn = mysql_connect($DBHOST,$DBUSER,$DBPASS);
mysql_select_db($DBNAME) or die("Unable to select database!");

// update the database
$query = "UPDATE profs SET name = '$myname', img = '$myimage', addr='$offaddress',
phone='$offphone', bio='$cvsummary', CV='$cvurl', crs1='$crs1', lnk1='$crsurl1', sem1='$end1',
crs2='$crs2', lnk2='$crsurl2', sem2='$end2',
        crs3='$crs3', lnk3='$crsurl3', sem3='$end3', crs4='$crs4', lnk4='$crsurl4', sem4='$end4',
oldcourse='$oldcrs', pwh='$pwh', pwa='$pwha', officeh='$offhours', research='$research',
        publications='$pubs', events='$events', students='$students', links='$links',
awards='$awards', orgs='$groups' WHERE email='$unm'";

mysql_query($query) or die('Query failed: ' . mysql_error());


print("Updated Info! Redirecting..." . "<br />");

mysql_close($conn);
redirect($redirect);

function redirect($url)
{
        header('Location: '.$url);
        exit();
}

dispEnd();

?>
```

```
<?php
include ('./inchook.php');
dispBegin("ERROR.");
session_start();
echo "Error ".$_SESSION['error'];
$_SESSION['error'] = "";

dispEnd();
?>
```

```
<?php
include('./inchook.php');

dispBegin("Forgot Password");
?>

<form method=POST action="forgot2.php">Enter your email address: <input type=text
name=uid><br>
<input type=submit value=submit></form>
<?php dispend(); ?>
```

```
<?php
include('./inchook.php');
dispBegin("Forgot Password");

$email=$_POST['uid'];

$user = "opal411";
$password = "cs411rocks";
$database = "opal411";
$server = "opal411mysql.legendaryfox.net";

$conn = mysql_connect($DBHOST,$DBUSER,$DBPASS);
mysql_select_db($DBNAME) or die("Unable to select database!");

$result =  mysql_query("SELECT * FROM profs WHERE email='$email'");
// get the row of values
$row = mysql_fetch_row($result);
$pwh = $row[23];
mysql_close($conn);

echo ("Here is your password hint: $pwh");
?><br>Please enter your answer: <form action=forgpwd.php method=POST>
<input type=hidden name=uid value=<?php echo $email; ?>>
<input type=text name=answer><br>
Enter your new password: <input type=password name=pw1><br>
Confirm new password: <input type=password name=pw2><br>
<input type=submit value=submit></form>
<?php dispend(); ?>
```

```
<?php
include('./inchook.php');
$unm = $_POST['uid'];
$ans = $_POST['answer'];
$pw1 = $_POST['pw1'];
$pw2 = $_POST['pw2'];

if(strcmp($pw1, $pw2)){
session_start();
$_SESSION['error'] = "passwords do not match";
header("location:error.php");
exit();}

$user = "opal411";
$password = "cs411rocks";
$database = "opal411";
$server = "opal411mysql.legendaryfox.net";

$conn = mysql_connect($DBHOST,$DBUSER,$DBPASS);
mysql_select_db($DBNAME) or die("Unable to select database!");

$result =  mysql_query("SELECT * FROM profs WHERE email='$unm' AND pwa='$ans'");
// get the row of values
$row = mysql_fetch_row($result);

if(!$row){
session_start();
$_SESSION['error'] = "incorrect answer";
header("location:error.php");
exit();}

$pw = md5($pw1);
mysql_query("UPDATE profs SET pw='$pw' WHERE email=$'unm'");

mysql_close($conn);
header("location:login.php");
exit();
?>
```

```
<?php

//'hook' to include major files

include('./config.php');
include('./dispFuncs.php');

?>
```

```
<?php

include ('./inchook.php');

dispBegin("List of Professors");


$conn = mysql_connect($DBHOST,$DBUSER,$DBPASS);
mysql_select_db($DBNAME) or die("Unable to select database!");

$sql = "SELECT name, email FROM profs";
$res = mysql_query($sql); //store the mysql resource handler

if (!$res) {
        die('Invalid query: ' . mysql_error());
}

/*
$res_array = mysql_fetch_array($res); //format the resource handler into an array that php can use
extract($res_array);
print_r($res_array);*/

$profCounter=0;
while($row = mysql_fetch_array($res))
{
        extract($row);
        $dispname[$profCounter] = $name;
        $dispemail[$profCounter] = str_replace('@bu.edu', '', $email);
        $profCounter++;
}

print "<h2>List of Boston University's CS Department Professors</h2><br><br>\n";



for($counter=0; $counter<$profCounter; $counter++)
{
        print "<a href='profpage.php?uid=$dispemail[$counter]'>$dispname[$counter]</a><br>\n";
}

print "<br><br>\n";


dispEnd();

print "<br><br><br><center>Are you a professor? Login <a href='login.php'>here</a>.</center>";

?>
```

```
<?php
include('./inchook.php');
dispBegin("Login");
?>

Login with your email address and password. <br>
<form name="login" method="post" action="logincheck.php">
Email: <input name="email" type="text"><br>
Password: <input name="pwd" type="password"><br>
<input name="Login" type="submit">
</form>

<br><br><br>
Forgot your password? Click <a href="forgot.php">here</a>.


<?php
dispEnd();
?>
```

```
<?php
include ('./inchook.php');
session_start();
//connect to the database
mysql_connect($DBHOST, $DBUSER, $DBPASS);
mysql_select_db($DBNAME);

//get posted variables
$unm = $_POST['email'];
$pw = $_POST['pwd'];

//hash password
$pw = md5($pw);

//lookup user, password and valid status in database
$sql = "SELECT * FROM profs WHERE email='$unm' AND pw='$pw' AND (status='v' or
status='t')";
$res = mysql_query($sql); //store the mysql resource handler

//error check

print "unm was $unm and pw was $pw (comapred to $ADMINPASS)";
if($unm == "admin" && $pw == md5($ADMINPASS))
{
        $_SESSION['unm'] = 'admin';
        header("location:admin_index.php");
        exit();
} elseif (!$res || (mysql_num_rows($res) != 1)) {
  $_SESSION['error'] = "Invalid Login";
  //print "unm was $unm and pw was $pw (comapred to $ADMINPASS)";
  header("location:error.php");
  exit();
}

$_SESSION['unm'] = $unm;
header("location:mainaccount.php");

?>
```

```
<?php
//remove all session variables, redirect to login page
session_start();
session_destroy();
header("location:login.php");
?>
```

```php
<?php

include('./inchook.php');

dispBegin("Main Account Page");


session_start();
if(!($_SESSION['unm'])){
header("location:login.php");
exit();
}
else{
$unm=$_SESSION['unm'];
//echo "login successful $unm";
}



$conn = mysql_connect($DBHOST,$DBUSER,$DBPASS);
mysql_select_db($DBNAME) or die("Unable to select database!");

$result =  mysql_query("SELECT * FROM profs WHERE email='$unm'");
// get the row of values
$row = mysql_fetch_row($result);

mysql_close($conn);
$pid = str_replace('@bu.edu', '', $unm);

print "<h2>Welcome to your account.</h2><br>\n";
print "<a href='edit_page.php'>Edit your page</a><br><br>\n";
print "<a href='profpage.php?uid=$pid' target='_blank'>View Page</a><br><br>\n";
print "<a href='changepw.php'>Change Password</a><br><br>\n\n";
print "<a href='mailto:admin@opal411.legendaryfox.net'>Email the website administrator</a> [<a href='email_help.php' target='_blank'>?</a>]";

//print "<a href='manage_alias.php'>";

dispEnd();



?>
```

```
<?php

include('./inchook.php');



mysql_connect($DBHOST, $DBUSER, $DBPASS);
mysql_select_db($DBNAME); //select the database

$unm = $_REQUEST['uid'];
$sql = 'SELECT * FROM profs WHERE email = \" . $unm . '@bu.edu\"; //query

$res = mysql_query($sql); //store the mysql resource handler

if (!$res) {
        die('Invalid query: ' . mysql_error());
}

$res_array = mysql_fetch_array($res); //format the resource handler into an array that php can use
extract($res_array);


dispBegin("$name's Profile Page");


if ($status == 'i')
{
        print "<center><h1><font color='red'>This user has been marked as
inactive.</font></h1></center><br>\n";
}


print "<img src='$img' height=100px width=100px><br><br>\n";
print "<h3>Basic Contact Info</h3>\n";
print "<table>";
print "<tr><td><b>Name:</b></td><td><a href='mailto:$email'>$name</a></td></tr>\n";
print "<tr><td><b>Department:</b></td><td>$department</td></tr>\n";
print "<tr><td><b>Address:</b></td><td>$addr</td></tr>\n";
print "<tr><td><b>Phone</b></td><td>$phone</td></tr>";
print "</table>\n\n";

print "<h3>Course Info</h3>\n\n";

print "<table>\n";
if ($crs1 != ")
{
        print "<tr><td><b>Course 1:</b></td><td><a
href='$lnk1'>$crs1</a></td><td><b>Semester:</b></td><td>$sem1</td></tr>\n";
}
if ($crs2 != ")
{
	print "<tr><td><b>Course 2:</b></td><td><a
href='$lnk2'>$crs2</a></td><td><b>Semester:</b></td><td>$sem2</td></tr>\n";
}
if ($crs3 != '')
{
	print "<tr><td><b>Course 3:</b></td><td><a
href='$lnk3'>$crs3</a></td><td><b>Semester:</b></td><td>$sem3</td></tr>\n";
}
if ($crs4 != '')
{
	print "<tr><td><b>Course 4:</b></td><td><a
href='$lnk4'>$crs4</a></td><td><b>Semester:</b></td><td>$sem4</td></tr>\n";
}

print "</table>\n\n";


print "<b>Office Hours:</b> $officeh<br>\n";
if ($oldcourse != '')
{
	print "<b>Previous Courses:</b> $oldcourse<br>\n";
}

print "<h3>Other Info</h3>";

if ($CV != '')
{
	print "<b>Link to Curriculum Vitae:</b> $CV<br>\n";
}

if ($bio != '')
{
	print "<b>Biography:</b><br>\n $bio<br>\n";
}

if ($research != '')
{
	print "<b>Research:</b><br>\n $research<br>\n";
}

if ($students != '')
{
	print "<b>Students:</b><br>\n $students<br>\n";
}


if ($awards != '')
{
	print "<b>Awards:</b><br>\n $awards<br>\n";
}

if ($orgs != '')
```

```
{
	print "<b>Organizations:</b><br>\n $orgs<br>\n";
}

if ($links != '')
{
	print "<b>Other Links:</b><br>\n $links<br>\n";
}
/*
	<img src="<?php echo $img; ?>" height=100px width=100px><br><br>

	<h3>Basic Contact Info</h3>
	<b>Name:</b>  <a href="mailto:<?php echo $email; ?>"><?php echo $name; ?></a><br>
	<b>Department:</b> <?php echo $department; ?> <br>
	<b>Address:</b> <?php echo $addr; ?> <br>
	<b>Phone:</b> <?php echo $phone; ?><br>

	<h3>Course Info</h3>


	<b>Course 1:</b> <a href="<?php echo $lnk1; ?>"><?php echo $crs1; ?></a></b>  &nbsp; &nbsp; &nbsp; <b>Semester:</b> <?php echo $sem1 ?> <br>
	<b>Course 2:</b> <a href="<?php echo $lnk2; ?>"><?php echo $crs2; ?></a></b>  &nbsp; &nbsp; &nbsp; <b>Semester:</b> <?php echo $sem2 ?> <br>
	<b>Course 3:</b> <a href="<?php echo $lnk3; ?>"><?php echo $crs3; ?></a></b>  &nbsp; &nbsp; &nbsp; <b>Semester:</b> <?php echo $sem3 ?> <br>
	<b>Course 4:</b> <a href="<?php echo $lnk4; ?>"><?php echo $crs4; ?></a></b>  &nbsp; &nbsp; &nbsp; <b>Semester:</b> <?php echo $sem4 ?> <br>
	<b>Office Hours:</b> <?php echo $officeh ?><br>

	*/

print "<br><br>Return to <a href='index.php'>main directory listing</a>.<br><br><br>\n";

dispEnd();

?>
```

```php
<?php
//verify login, if invalid direct to login page
include ('./inchook.php');
session_start();
if(!($_SESSION['unm'])){
  header("location:login.php");
  exit();
}

dispBegin("Changing Password");

$unm=$_SESSION['unm'];
$old=$_POST['old'];
$newa=$_POST['new1'];
$newb=$_POST['new2'];

if(strcmp($newa,$newb)!=0){
  $_SESSION['error'] = "New passwords do not match";
  header("location:error.php");
  exit();
}

$old=md5($old);
$newa=md5($newa);

//connect to the database
mysql_connect("opal411mysql.legendaryfox.net", "opal411", "cs411rocks");
mysql_select_db("opal411");


$sql = "UPDATE profs SET pw='$newa' WHERE email='$unm' AND pw='$old'";
$res = mysql_query($sql);

//error handling
if($res!=1){
  $_SESSION['error']="Could not change password.".$res;
  header("location:error.php");
  exit();
}

echo "Change Successful";

print "<br><br>Return to <a href='mainaccount.php'>main account page</a>.<br><br><br>\n";

dispEnd();

?>
```

```
<?php

include('./inchook.php');


dispBegin("Registration");

?>

Enter your pre-approved email address to begin the verification process.<br>
<form name="reg" method="POST" action="regcheck.php">
<input type="text" name="email"><br>
<input type="submit" name="submit">
</form>

<?php
dispEnd();

?>
```

```
<?php
session_start();
include('./inchook.php');

dispBegin("Registration Check");

//connect to the database
mysql_connect($DBHOST, $DBUSER, $DBPASS);
mysql_select_db($DBNAME); //select the database

//get posted variable
$unm = $_POST['email'];

//find user in confirmation table
$sql = "SELECT * FROM confirmation WHERE user='$unm'";
$res = mysql_query($sql);

//error handle if not pre-approved
if(!$res){
  $sql = "SELECT * FROM profs WHERE user='$unm'";
  $res = mysql_query($sql);
  if(!res){
    $_SESSION['error']="User not yet approved.";
  }
  else{
    $_SESSION['error']="User already confirmed.";
  }

  header("location:error.php");
  exit();
}

$num = mysql_num_rows($res);
if($num==1){
  //retrieve confirmation code and email it to user
  $row=mysql_fetch_array($res);
  $conf=$row['conf'];
  $subject="Website Account Confirmation";
  $header="from: noreply@bu.edu";
  $message="Please access the URL below to confirm:\r\n";
  $message.="$LOCALDOMAIN/confirm.php?confirm=$conf";
  //$sent = mail($unm,$subject,$message,$header);

  //let user know what just happened
  echo "A confirmation has been emailed to you.";

  //This is here for debug purposes, in case the server lacks email capability.
  print "This is here for debug purposes, in case the server lacks email capability.<br>\n";
  echo $message;

}
else{
  $_SESSION['error']="Retrieving Record";
  header("location:error.php");
  exit();
}


dispEnd();
?>
```