

Center for Addressing Health Disparities through Research and Education – Mentor / Mentee and Project Database

CS470 Project Write-up
Mathew Bergt
April 28, 2011

Table of Contents

Abstract.....	3
1. Introduction.....	3
2. Project Overview.....	3
2.1 Database Structure.....	3
2.2 Website Structure.....	5
3. Project Requirements.....	6
3.1 Functional Specifications.....	6
3.2 System Specifications.....	7
4. System Design.....	7
4.1 Data Structures.....	7
4.2 System Architecture.....	7
4.3 Algorithms.....	7
5. Development Process.....	8
5.1 Testing and Debugging.....	8
5.2 Activity Breakdown.....	8
6. Results.....	10
6.1 Administrator and Project Creator Interfaces.....	10
6.2 Search Results Modifications.....	11
6.3 Tag Cloud and Icon Search.....	12
6.4 Monthly Usage Report.....	12
6.5 User Feedback Form.....	13
6.6 Future Modifications.....	13
7. Summary and Conclusions.....	14
References.....	14
Appendix A: User Manual.....	15
Appendix B: Code Listing.....	19

Abstract

The Center for Addressing Health Disparities through Research and Education (CAHDRE) works to disseminate information regarding health disparities. CAHDRE has created a mentor program for those interested in health related careers. A prototype website was developed to assist in matching mentors and mentees, and allowing users to collaborate on research projects. This project implemented changes to the user interface which enhance the usability of the system, as well as provide functionality to gather data about usage of the system.

1. Introduction

This project serves the Center for Addressing Health Disparities through Research and Education (CAHDRE). CAHDRE works to disseminate information regarding health disparities, particularly to UAA faculty and students interested in pursuing careers related to the field. To encourage interested individuals, a program exists to match health mentors to mentees.

Currently, a mentor or mentee can express interest in the program by filling out the appropriate web form, located on the CAHDRE website. The forms must be manually reviewed, and mentors are matched to mentees based on the provided information. To increase the efficiency of this process, Dr. Kenrick Mock designed a prototype system which utilizes a database to store information about mentors, mentees, and their collaborative projects. This project will focus on implementing changes to the existing prototype which will increase the usability of the user interface (UI) and provide functionality for data collection.

2. Project Overview

2.1 Database Structure

The prototype website currently contains three main types of functionality:

- Creating and accessing user accounts.
- Searching the database for mentors and mentees.
- Searching the database for collaborative projects.

This functionality relies on access to a database to dynamically display information to the user.

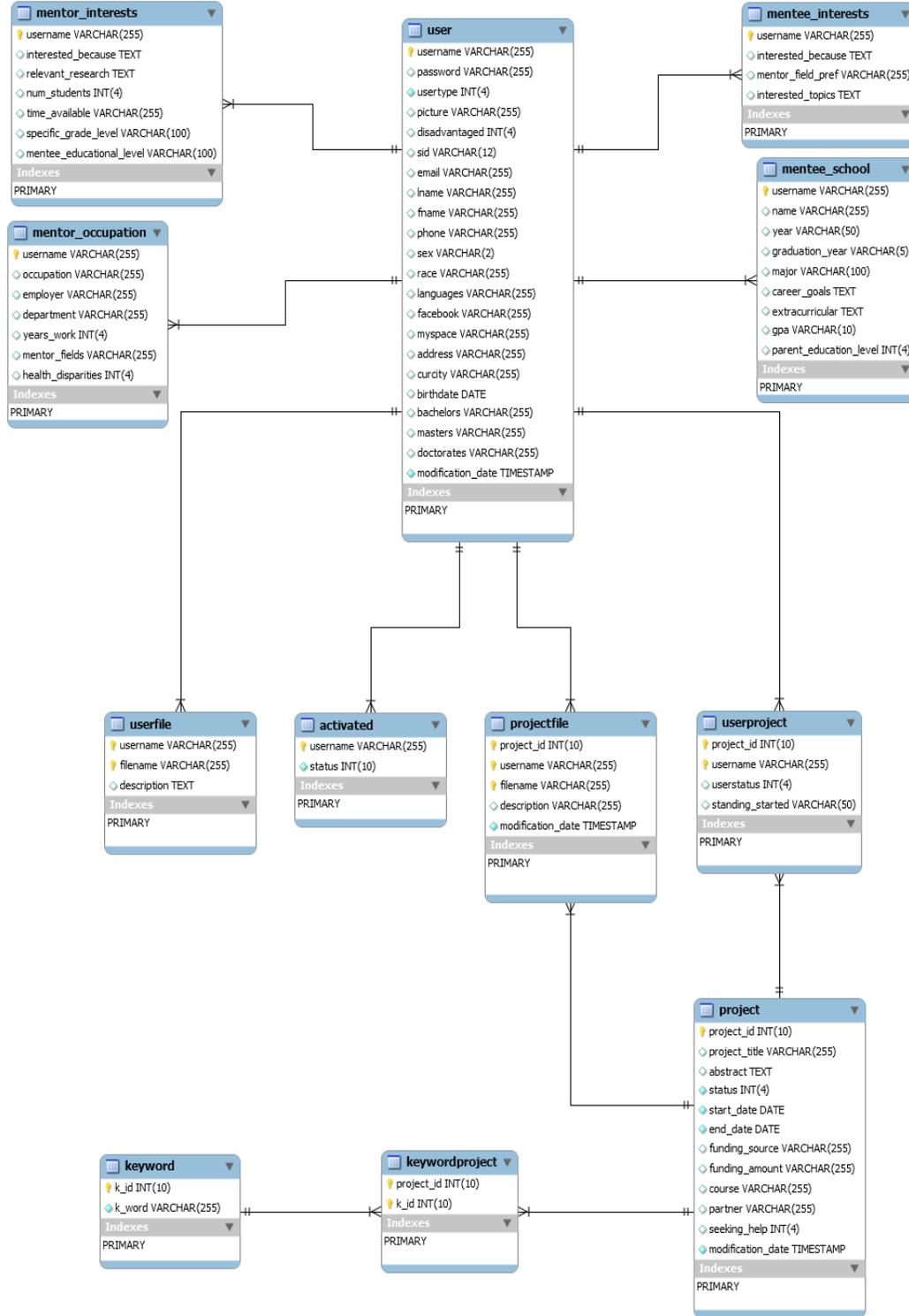
The database is designed such that a majority of the tables are in a many-to-one relationship with two main tables: the Project table and the User table. The Project table stores information about projects created by mentors, and the User table stores information gathered from mentors and mentees. This information is gathered upon creation of a new user account or editing the user profile.

The following tables are in a many-to-one relationship with the User table:

- Activated. Indicates whether or not a user account has been activated for use.
- Mentee_Interests. Stores information about mentee interests.
- Mentee_School. Stores information about mentee academic information.
- Mentor_Interests. Stores information about mentor interests.
- Mentor_Occupation. Stores information about mentor employment information.
- Projectfile. Stores information about files uploaded for projects.
- Userfile. Stores information about files uploaded by users, such as their picture.
- Userproject. Stores information about the projects a user is working on.

The Projectfile and Userproject tables are also in a many-to-one relationship with the Project table. Thus, User and Project are in a many-to-many relationship with each other, using Projectfile and

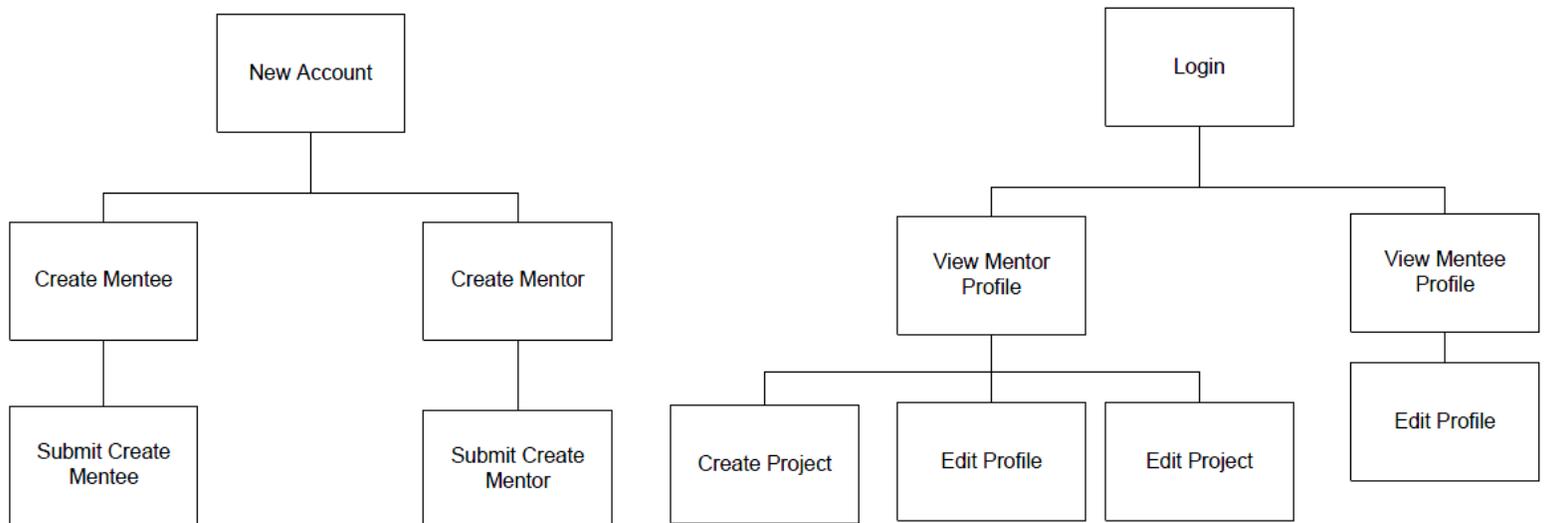
Userproject as junction tables. Finally, the Project table is in a many-to-many relationship with the Keyword table via the Keywordproject junction table. These tables store keywords for projects. Keywords are entered by mentors to improve the search visibility for their projects. The following entity-relationship diagram illustrates the design:



2.2 Website Structure

The prototype website contains functionality to manage user accounts and retrieve results from the database. Each of these components are composed of several different features. For instance, a user may search for mentees by specifying a keyword, specifying a preferred medical field for the mentor, or by browsing alphabetically by last name.

The account management functionality allows users to create new accounts, enter information and add a picture into their profile, and if the user is a mentor, create and edit projects. The following diagrams illustrate the data flows for new account creation and user account management:



When a new account request for a mentor is submitted, the website administrator must log in and manually approve or deny the application.

The mentor and mentee search functionality allows users to search for other users in a variety of ways. A user may search for a mentor by keyword, health expertise, employer, department, or browse mentors by last name. A user may search for a mentee by keyword, mentor field preference, or browse mentees by last name. The following screen capture illustrates the mentor search with the health expertise tab selected:

The screenshot shows the CAHDRE website interface. The top navigation bar includes tabs for 'Keyword', 'Health Expertise' (selected), 'Employer', and 'Department'. Below the tabs is a search form titled 'Search mentors by health area of expertise' with checkboxes for various health areas: Allied Health, Biomedical Research, Behavioral Health, Medicine, Nursing (checked), Oral Health / Dentistry, Physician Assistant, Public Health, and Social Work. There is also an 'Other:' text input field and a 'Submit' button.

Below the search form is a table titled 'All Mentors' with the following data:

Name	Expertise	Mentees Currently Assigned	Mentees Desired	Preferred Mentee Education Level	Specific Mentee Grade Level
BERGT, MATHEW		1	3	Post Graduate	
Mentor3, Joe	Biomedical Research, Nursing, Social Work, Sleep, Video Games	3	2	College, Post Graduate	13
Mentor4, Kenrick	Social Work, Information Technology	4	1	College	
Mentor5, Tom	Allied Health, Medicine	1	2	Post Graduate	
Mentor6, Bill	Nursing, Public Health	0	1	Elementary (K-6), Post Graduate	

The project search functionality allows users to browse by project title, project date, participant academic department, participant employer, search by keyword, or browse by clicking on a tag in a tag cloud. The tag cloud is implemented so that the size of the keyword displayed is proportional to the number of times the keyword is found in the Keyword table. Clicking on a tag displays a list of projects containing the specified keyword. The following screen capture illustrates that the keyword “Chugach” occurs most frequently since it is the largest:

CAHDRE

Main
New Account
About
Keyword Tags
People
Mentors
Mentees
Search Mentors
Search Mentees
Projects
Search
Title
Date
Department
Employer

All Keyword Tags

Keyword Tags are keywords entered by you, the users, to better define the scope of a research project. Entering keywords for a project will also help when searching projects. Below is a keyword histogram. It uses larger fonts for keywords that are used more frequently.

Want your project keywords bigger?
Please encourage entering keywords when creating or editing projects.

Keyword Tags sorted Alphabetically
Alaska, Cancer, **Chugach**, Forest, HIV, Health, Psychiatry

Keyword Tags sorted by Frequency
Chugach, Forest, Cancer, Alaska, HIV, Health, Psychiatry

3. Project Requirements

CAHDRE requested several modifications to the existing prototype website. The client indicated that new features beyond the initial requirements will not likely be requested, but that there may be changes to the requested features. The project schedule allows time for these possible revisions.

3.1 Functional Specifications

1. Functionality to manage administrator accounts must be modified from the current hard-coded method to a more dynamic implementation using the database.
2. There must be functionality for tag clouds which search for mentors and mentees by health area, such as biomedical research, nursing, or social work. A list of mentors or mentees with the specified tag clicked in their profile should be displayed.
3. There must be functionality to search for mentors and mentees by clicking graphical icons related to health areas. This requirement is almost identical to search by tag cloud, but uses icons instead of tags.
4. The system should generate a monthly usage report and email it to an administrator. The report should contain data about the total number of mentors and mentees, the number of new mentors and mentees, the total number of projects, the number of new projects, and total page hits.
5. The search result functionality should be modified to improve ease of use. Search results should be paginated and sortable.
6. The functionality which allows mentors to create projects should be modified to streamline the process of adding mentees to projects. The current functionality requires the mentor to know the username of a mentee before adding them. Mentors should be able to browse and select mentees from a pop-up list to add them to a project.
7. A feedback form should be displayed to users after an arbitrary number of logins. This form should gather feedback about the functionality of the website.

3.2 System Specifications

The existing prototype system is written in PHP and Javascript, so development will continue in these environments. Queries must be made to the database using MYSQL. The end-user must have a web browser which supports Javascript to interact with the system. The end-user must also possess an internet connection. Relatively slow internet connections will likely deteriorate the experience of the end-user.

4. System Design

Since much of the foundation of the system has been implemented, many of the desired modifications will be refactorings of existing code. The system is essentially event-driven, responding to different page requests given some parameters, so new implementations will closely follow this design.

4.1 Data Structures

The desired features will rely almost primarily on the database. I will implement new tables for administrator accounts, usage statistics, and feedback data. The Administrator table will be in a many-to-one relationship with the User table, and contain the embedded foreign-key username as the primary-key. The Usage_Statistics table will contain monthly totals for users, projects, and page hits. This table will likely be populated each month with a query to the User and Project tables, so it will not be in a relationship with the other tables. The Feedback table will contain feedback about the website from users, and will be identified by the embedded foreign-key username. The Feedback table will be in a many-to-one relationship with the User table. Further desired features can be accomplished using the existing database infrastructure.

4.2 System Architecture

To implement administrator accounts, the query for the existing administrator login must be refactored. For extra security, I will research password hashing and salt before implementing this feature. Additionally, the administrator interface will likely be refactored to display personalized information and options to the user.

Implementation of search by tag clouds and icons will likely borrow heavily from the existing implementation of the tag cloud. Displaying the search results will simply utilize the existing search result functionality, with the specified health area checked.

Maintenance of monthly usage reports will occur in two steps: the User and Project tables will be queried for totals and the Usage_Statistics table will be appropriately updated. Keeping track of hits to the main page will require a simple refactoring in the existing code for the main page. I have allowed enough time in the schedule for this feature in case any problems are encountered with automatic monthly email generation.

Modification of the search results functionality and the project creator will require refactoring of existing components. I will need to research search result pagination, as well as the search-as-you-type functionality before implementation of these features. It is not likely that major re-design will be necessary however. Similarly, implementation of the feedback form should be very simple.

4.3 Algorithms

Since the system relies on queries to the database, no notable algorithms are expected to be employed in the development process. Great care will be taken to construct efficient queries, as well as prevent

any sort of injection attacks. If the project schedule permits, the search functionality queries will likely be refactored for improved search results.

5. Development Process

After the initial set of requirements were presented, the client indicated that no further features would be requested, but revisions to current features may occur. Given this uncertainty, combined with a moderate learning curve for this project, I chose to pursue a form of the prototyping development methodology. Demonstration of prototypes to the client were largely abandoned however, due to schedule constraints.

5.1 Testing and Debugging

Testing and debugging were unique processes throughout the course of this project. Since many of the modifications were to the UI, these activities were difficult to discern from actual implementation. Once the features were implemented to satisfy the specified requirements, testing was essentially complete. The implemented features were tested in Google Chrome, Microsoft Internet Explorer, and Mozilla Firefox.

Debugging was rather difficult however. The use of TextPad for a development environment provided no debugging tools. Thus, the use of print statements were often helpful in the debugging process. This method was not ideal, but proved to be sufficiently effective. Additionally, Internet Explorer provided a useful built-in debugging tool for Javascript, which indicated the type of error and line number.

I encountered several bugs during implementation which took a significant amount of time to solve, though the solutions were generally simple. I attribute much of these difficulties to my lack of experience using PHP, HTML, and jQuery. One problem which arose was lack of compatibility of the jQuery Tablesorter plugin between browsers. At one point, the search result functionality which implemented this plugin was working in both Chrome and Internet Explorer, yet did not display results correctly in Firefox. Many hours were spent attempting to solve this problem. The solution was simple: Chrome and Internet Explorer support a special table footer tag which I was using, but Firefox does not. The pagination script was removed from this footer and placed outside the table, and the problem disappeared. Another problem arose, this time in Internet Explorer, when attempting to delete a user record from a table using an image as a form input. After some internet research, I found out that Internet Explorer does not return the 'id' of the image input clicked, but instead returns the screen coordinates. This solution was also simple: an additional hidden input was created in the form with the required 'id'.

5.2 Activity Breakdown

For several reasons, this project gradually fell behind schedule. For most activities, required time estimates were either accurate or too liberal. Estimates for implementation were extremely inaccurate however, as this activity put this project far behind schedule. Implementation took longer than expected due to my general inexperience with web development languages. Another issue which hindered implementation was my lack of understanding about how the previously implemented system was designed. A lack of clear design patterns resulted in a good deal of searching through scripts to make the required modifications. Additional up-front research of the system would have been a wise investment. The following table depicts the estimated and actual times for each activity:

Activity	Estimated Time	Approximate Actual Time
Implementation	26.4 hours	80 hours
Design	16 hours	6 hours
Write-Up	7.2 hours	8 hours
Testing	19.2 hours	8 hours
Requirements	4 hours	4 hours
Presentation	7.2 hours	4 hours
TOTAL	80 hours	110 hours

The decision to use the prototyping methodology was essentially abandoned mid-way through the semester. I met with the client in early March to demonstrate the administrator interface and search result modifications, but no further meetings took place. My approach rapidly devolved into the “code-like-hell” methodology subsequent to this meeting. The client was presented with the final version of the prototype during the short demonstration day, at which time they requested various minor modifications to the UI, most of which were successfully implemented. The following table depicts my actual schedule for the project:

Project: CAHDRE Website

Current Week																
Tasks	Weeks	01/17 to 01/24	01/24 to 01/31	01/31 to 02/07	02/07 to 02/14	02/14 to 02/21	02/21 to 02/28	02/28 to 03/07	03/07 to 03/14	03/14 to 03/21	03/21 to 03/28	03/28 to 04/04	04/04 to 04/11	04/11 to 04/18	04/18 to 04/25	04/25 to 05/02
Initial Planning / Pre-Proposal		█														
Initial Requirements			█													
Initial Design			█													
Write Proposal				█												
Research Search-As-You-Type Functionality				█	█											
Implement Administrator Accounts					█	█										
Implement Tag Cloud Search						█	█									
Implement Icon Search						█	█									
Proposal Presentation							█									
Testing								█	█							
Show Prototype									█							
Revise UI										█						
Research Features to Manipulate Search Results (jQuery)											█					
Implement Changes to Search Results												█	█			
Testing													█			
Code Inspections														█		
Revise UI															█	
Revise Requirements / Design																█
Implement Monthly Report Function																█
Implement User Feedback Survey																█
Implement Changes to Project Creator Interface																█
Testing																█
Final Presentation																█
Short Demo / Show Prototype																█
Revise UI																█
Final Writeup																█
Tasks	Weeks	01/17 to 01/24	01/24 to 01/31	01/31 to 02/07	02/07 to 02/14	02/14 to 02/21	02/21 to 02/28	02/28 to 03/07	03/07 to 03/14	03/14 to 03/21	03/21 to 03/28	03/28 to 04/04	04/04 to 04/11	04/11 to 04/18	04/18 to 04/25	04/25 to 05/02
Current Week																

6. Results

Each of the specified requirements for this project were met, mostly to the satisfaction of the client. The desired modifications to the UI and functionality for report generation were successfully implemented. The client has indicated they will be using the new interface sometime in the near future. Several further modifications to the project were requested upon the final demonstration, some of which are complex enough for future mid-sized projects.

6.1 Administrator and Project Creator Interfaces

The modifications to the administrator and project creator interfaces are essentially identical. The administrator page was modified so that administrators which are designated with full administrative privileges are now able to add and remove other administrators. A user can add an administrator by typing the name or username of the user they wish to add into a search field, and selecting from the

resulting auto-suggest list which appears. The list of administrators is then displayed in a sortable and paginated table, created using the jQuery Tablesorter plugin. The new project and edit project pages were modified using the same scheme, allowing the user to add and remove mentors and mentees from collaborative projects. The following screen-capture depicts the new administrator interface:

Add/Remove Administrator
Begin typing the name or username of an existing user and select from the suggestion list:

ment Privilege level: Basic

- Mentor3, Joe: mentor3
- Mentor4, Kenrick: mentor4
- **Mentor5, Tom: mentor5**
- Mentor6, Bill: mentor6

Name	Username	Privilege Level	Delete Administrator
Bergt, Mathew	matt	Basic	<input type="button" value="x"/>
Mentor5, Tom	mentor5	Basic	<input type="button" value="x"/>
Bergt, Matt	test	Basic	<input type="button" value="x"/>
Clapton, Eric	test10	Full	
May, Brian	test17	Basic	<input type="button" value="x"/>

1/1

6.2 Search Results Modifications

The jQuery Tabs and Tablesorter plugins were the primary tools utilized for search results modifications. The Tabs plugin allows users to select various types of searches on different tabs without reloading the page. The Tablesorter plugin allows the user to sort on each column, and also view paginated results. The following screen capture shows a search for mentors by health expertise, in this case with the health areas of biomedical research and medicine selected:

CAHDRE

Keyword **Health Expertise** Employer Department

Search mentors by health area of expertise

Allied Health Biomedical Research Behavioral Health
 Medicine Nursing Oral Health / Dentistry
 Physician Assistant Public Health Social Work

Other:

Mentors with expertise in: Biomedical Research, Medicine

Name	Expertise	Mentees Currently Assigned	Mentees Desired	Preferred Mentee Education Level
Mentor3, Joe	Biomedical Research, Nursing, Social Work, Sleep, Video Games	3	2	College, Post Graduate
Mentor5, Tom	Allied Health, Medicine	1	2	Post Graduate

1/1

6.3 Tag Cloud and Icon Search

The implementation for the tag cloud search and icon search is mostly similar. The tag cloud displays health area tags contained in the database with their size proportional to the number of occurrences. The icon table displays icons graphically related to each health area. Clicking on a tag or icon displays a list of mentors and mentees on separate tabs, and once again in a sortable and paginated table using jQuery plugins. The following screen capture demonstrates the tag cloud and icon searches:

Mentor Health Areas Sorted Alphabetically

Allied Health, Biomedical Research, Computer Programming, Health Education, Information Technology, Medicine, Nursing, **Public Health**, Sleep, Social Work, Video Games

6.4 Monthly Usage Report

Functionality was implemented to create and mass-email a monthly report with statistics for database usage. This script will be automatically run on the first day of each month by a cron job scheduler, and generates statistics for the total and monthly number of users and projects, as well as the total and unique number of website visits for the month. This report is emailed to administrators. Two tables, 'usagestatistic' and 'websitevisit', were created in the database to carry out this task. The 'websitevisit' table tracks the number of hits on the main page by IP address, while the 'usagestatistic' table contains the necessary fields to generate the report. The following entity-relationship diagram depicts these simple tables:

usagestatistic	
generation_date	DATE
total_mentors	INT(10)
total_mentees	INT(10)
total_projects	INT(10)
total_page_hits	INT(10)
unique_page_hits	INT(10)
Indexes	

websitevisit	
hit_id	INT(10)
ip_address	VARCHAR(255)
timestamp	DATETIME
Indexes	

The following code-excerpt exemplifies the types of queries executed on the database to generate the monthly usage report:

```
// Retrieve total number of mentees
$sql = "select * from user where usertype = 1;";
$result = mysql_query($sql);
$totalMentees = mysql_num_rows($result);

// Retrieve total number of mentors
$sql = "select * from user where usertype = 2;";
$result = mysql_query($sql);
$totalMentors = mysql_num_rows($result);

// Get date from previous month
$year = date('Y');
$previousMonth = (intval(date('m')) - 1) % 12;
if ($previousMonth < 10)
    $previousMonth = '0' . $previousMonth;
$day = '01';
$previousDate = $year . '-' . $previousMonth . '-' . $day;

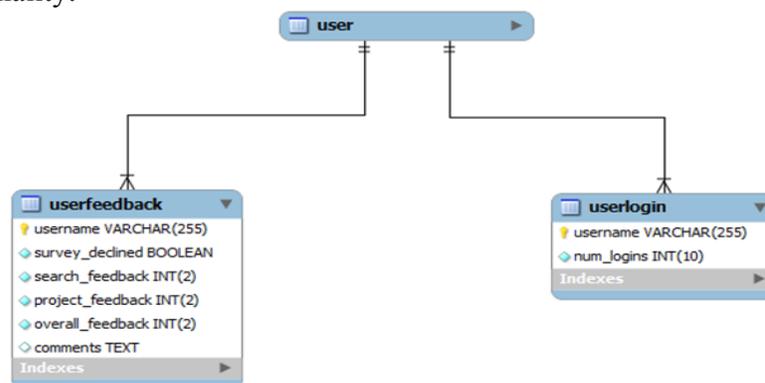
// Get the totals from last month and find number of new users
$sql = "select * from usagestatistic where generation_date = '" . $previousDate . "'";
$result = mysql_query($sql);
$row = mysql_fetch_assoc($result);

$newMentees = $totalMentees - $row['total_mentees'];
$newMentors = $totalMentors - $row['total_mentors'];
```

Here, the PHP date function is utilized to find the date of the first day of the previous month from the current month. This is then used to look up the previous month record in the 'usagestatistic' table, since it is identified by the date of generation. Finally, the number of new mentors and mentees for the month is found by subtracting the previous number of new mentors and mentees from the current number.

6.5 User Feedback Form

A simple user feedback form was created to gather data about the effectiveness of the functionality provided by this website. Upon logging in for the third time, the user is asked to fill out the form or decline. The form requests that the user rate the search, project, and overall functionality of the website, and leave any comments if desired. Once the form is submitted, the results are emailed to all administrators. The report shows the scores given by the user, as well as the average overall scores contained in the database. The following entity-relationship diagram depicts the tables necessary to implement this functionality:



The 'userlogin' table keeps track of the number of times a user has logged in, and the 'userfeedback' table keeps track of the feedback for each user.

6.6 Future Modifications

Before making further modifications to this website, I recommend heavy refactoring of the existing code. There are countless instances of duplicate or similar code throughout scripts, which could easily be broken into several generalized functions. There is also no clear design pattern employed. HTML

views are combined in scripts with database queries, significantly diminishing the readability of the code. A model-view-controller design pattern would be an effective solution to this problem.

Further modifications to the functionality of the website have been requested by the client. The most complex of these is a Facebook-style interface which allows mentors to link to other mentors and mentees and vice versa, thus creating a 'friend-network' of sorts. Other modifications requested are an option to send a mass-email to users, an interface for administrators to add and remove users from collaborative projects, and a more cohesive interface for administrators to match mentors and mentees, perhaps visually in a drag-and-drop interface. I personally desire to further enhance the report generation functionality to provide more statistics. This could be incorporated into the new administrator interface and even display some PHP generated charts and graphs.

7. Summary and Conclusions

Modifications were made to the existing prototype website and database for CAHDRE to enhance the UI and provide functionality for data collection. These modifications met the initial requirements specified by the client. This project initially started using the prototyping methodology, but schedule constraints later in the life-cycle forced the abandonment of this strategy.

This project was my first real venture into web development and the use of PHP and jQuery. I found learning how these languages work together with HTML and MySQL to be challenging and rewarding. I now possess a basic knowledge of various methods to dynamically display content on the web. I also gained some experience in the realm of project management, and learned some valuable lessons about project scheduling and time management.

It seems I would have benefited from more up-front research, particularly of jQuery, PHP, and the structure of the existing prototype website. Even with this research however, some problems likely could not have been avoided. Browser specific problems are tedious and tend not to be obvious, unless you have some experience in web development.

References

- Tablesorter plugin, Christian Bach.
<http://tablesorter.com/docs/>
- AutoCompleter plugin, Jamie McConnell.
<http://www.nodstrum.com/2007/09/19/autocomplete/>

Appendix A: User Manual

This manual provides a simple guide to use the new modifications to the CAHDRE website and database.

Administrator Interface

- You must be granted administrative privileges to access the administrator interface. To access the interface, log in and click the 'Administrator Functions' button at the top of the page.



- As an administrator, you will be granted either basic or full administrative privileges.
 - Basic:** You may approve or deny pending mentor accounts.
 - Full:** You may approve or deny pending mentor accounts, and you may add or remove other administrator privileges.
- The administrator interface allows administrators to add and remove other administrators.
 - Add administrator:** Type the name or username of an existing user into the search field and select from the auto-suggestion list which appears. Select the privilege level and click submit.
 - Delete administrator:** Click the red delete button in the row of the administrator you wish to remove privileges from.

Welcome Mathew Bergt.
Administrator privilege-level: full

Activate Mentor Account

1. Test Mentor.

Add Administrator

Begin typing the name or username of an existing user and select from the suggestion list:

Privilege level: Basic ▾

Name	Username	Privilege Level	Remove Administrator Privileges
Bergt, Mathew	matt	Full	
Bergt, Matt	test	Basic	<input type="button" value="X"/>
Clapton, Eric	test10	Full	<input type="button" value="X"/>

1/1

Project Interface

- The new project interface allows you to use an auto-complete search field to add mentors and mentees to projects.
- The design is roughly identical to the add/remove functionality of the administrator interface.
 - **Add Mentor:** Type the name or username of an existing mentor into the mentor search field and select from the auto-suggestion list which appears. Click 'Add Mentor'.
 - **Add Mentee:** Type the name or username of an existing mentee into the mentee search field and select from the auto-suggestion list which appears. Click 'Add Mentee'.
 - **Delete Mentor / Mentee:** Click the red delete button in the row of the mentor or mentee you wish to remove from the project.
- Click 'Submit' at the bottom of the page when you are finished editing or creating your project.

Edit Your Current Practice / Project

Begin typing the name or username of an existing mentor and select from the suggestion list:

Begin typing the name or username of an existing mentee and select from the suggestion list:

Name ↕	Username ↕	Type ↕	Remove ↕
BERGT, MATHEW	matt	Mentor	
Bergt, Matt	test	Mentee	
May, Brian	test17	Mentee	

⏪
⏩

⏪
⏩
Results per

page: ▾

Monthly Usage Report and User Feedback

- The monthly usage report is set to only generate on the first day of each month.
 - A cron job scheduler must be run for the first day of each month in order to automatically generate the report.
 - The report is sent to all administrators.

This is an auto-generated monthly report from the CAHDRE Mentor/Mentee and Project database containing statistics for the usage of the website. You have received this email because you are designated as an administrator.

Total mentors: 9
 Total mentees: 17
 New mentors: 9
 New mentees: 17
 Total projects: 13
 New projects: 13
 Total page hits: 226
 Unique page hits (by unique IP address): 10

[↩ Reply](#) [↩ Reply to all](#) [→ Forward](#)

- The feedback form displays after a user's third login.
 - The feedback is compiled into a report along with additional average feedback statistics from the entire database.
 - The report is sent to all administrators.

A user has submitted feedback about the CAHDRE website. You are receiving this email because you are designated as an administrator.

Name: Bergt, Matt
Search functionality rating: 1/5
Project functionality rating: 3/5
Overall functionality rating: 1/5
Comments: Some comments here.

Database Totals
Average search functionality rating: 0.7500
Average project functionality rating: 2.7500
Average overall functionality rating: 2.2500

[↩ Reply](#) [↩ Reply to all](#) [→ Forward](#)

Search Functionality

- All search functionality is designed the same. Each column of the search results is sortable.
 - To sort, click the header row of the column.
 - To sort on multiple columns, hold control and select the header rows of the columns you wish to sort.
- All search results are paginated. The current page and total number of pages is displayed in a field below the search results.
 - Clicking the arrow buttons navigates to the previous or next page.
 - Select the number of results per page from the drop down menu.
- Some search results are tabbed. Each tab allows you to conduct a specific type of search. Simply click the desired tab and enter your search specifications.

Keyword
Mentor Field Preference
Mentee Demographics

Search mentees by preferred field for a mentor

Allied Health
 Biomedical Research
 Behavioral Health
 Medicine
 Nursing
 Oral Health / Dentistry
 Physician Assistant
 Public Health
 Social Work

Other:

Mentees with mentor field preferences of: Biomedical Research

Name	School	Year in School	Graduation Year	Mentor Field Preference	Mentor Field Preference	Number Active Projects	Mentors
Garcia, Francisco	UAA	Senior	2011	Biomedical Research	Physician Assistant	1	
Bergt, Matt	UAA	Senior	2011	Biomedical Research	Radiology	1	MATHEW BERGT
Clapton, Eric	UAF	Senior	2015	Biomedical Research	Computer Programming	1	MATHEW BERGT, Jerry Garcia
Page, Jimmy				Biomedical Research	Public Health	0	
Garcia, Jerry	UAA	Freshman		Biomedical Research	Medicine	3	MATHEW BERGT, Kenrick Mentor4, Ted Student2, Jerry Garcia, Vanessa Hiratsuka
Allman, Duane	UAS	Sophomore	2013	Biomedical Research	Nursing	2	Joe Mentor3, Tom Mentor5, Bill Mentor6, Duane Allman
Gilmour, David	UAA	Freshman	1969	Biomedical Research	Public Health	0	

Appendix B: Code Listing

- *Created and modified files:*
 - submitFeedback.php
 - generateReport.php
 - menteeSearch.php
 - tagSearch.php
 - login.php
 - admin.php
 - logout.php
 - new_project.php
 - edit_project.php
 - addResearcher.php
 - newProjectParticipant.php
 - autoSuggest.php
 - healthTag.php
 - main.php
 - view_mentor.php
 - view_mentees.php
 - deleteResearcher.php
 - feedbackForm.php
 - includes.inc
 - addAdmin.php
 - deleteAdmin.php
 - view_employer.php
 - view_dept.php
 - view_date.php
 - view_title.php
 - search.php
 - mentorSearch.php
 - cahdre_dev.sql
 - create_database.php
 - doHtml.inc
 - look.inc
 - DIRECTORY: jquery.tablesorter/
 - DIRECTORY: icons/

- *Files not modified:*
 - approveMentor.php
 - alltags.php
 - showAdmins.php
 - about.php
 - cancel_delete.php
 - create.php

- createMentee.php
- createMenteeSubmit.php
- createMentor.php
- createMentorSubmit.php
- index.php
- load_project_file.php
- load_user_picture.php
- project.php
- user.php
- view_project.php
- view_user.php
- db_cahdre.sql
- test.html
- menteeinfo.txt
- mentorinfo.txt
- DIRECTORY: jquery-ui-1.8.6.custom/
- aStyleSheet.css
- doJavascript.inc
- foobar.css
- footer.inc
- log.inc
- mainHeader.inc
- orig-css.css
- uaa.css
- DIRECTORY: files/