

**67-475 Information Systems Applications  
Carnegie Mellon University  
MIWatch – Milestone 4**

**Team:**

Karen Chen  
Paul Dille  
Hannah Leung  
Chase Midler

**Advisor:**

Raja Sooriamurthi

**Client Contact:**

Phyllis Vine

## Table of Contents

<b>1. Executive Summary</b>	<b>3</b>
<b>2. Implementation Status</b>	<b>3</b>
<b>3. Defect Report</b>	<b>4</b>
<b>4. Usability Report</b>	<b>4</b>
<b>5. Project History and Lessons Learned</b>	<b>5</b>
5.1 System Description	5
5.2 Implementation Overview	5
5.3 Lessons Learned	6
5.4 Assessment of Quality	6
5.5 Time Accounting	8
<b>6. Individual Statement of Accomplishments</b>	<b>9</b>
6.1 Karen Chen	9
6.2 Paul Dille	9
6.3 Hannah Leung	10
6.4 Chase Midler	11
<b>7. Updates to Previous Milestones</b>	<b>12</b>
7.1 Suggested Database Structuring	12
7.2 MIWatch Web Poster	13

## 1. Executive Summary

MIWatch is an online news source for persons affected by mental illness(es) including, but not limited to patients, family members, service recipients, clinicians, and members of the press. Site features enable users an easy to navigate interface allowing searches to be conducted by date, author, and a unique search engine of the archives. The end-goal of MIWatch is to be an information hub for mental health issues.

People affected by mental illness(es) need a place where they are able to search for available mental help facilities nearest their location. Locations information would ideally be contained in a central hub which would serve as a one-stop-shop for all mental illness informational needs.

Utilizing the Google API, the team implemented a map of mental help facilities, which allows users to search for facilities nearest their address filterable by facility type. Selecting a specific location will allow the user to browse through relevant contact, services, insurance, and other general information. The option for the user to enter in their starting address to return directions to the facility is also offered. These locations are maintained through a locally managed database that is updated by the team, and later by MIWatch staff.

## 2. Implementation Status

A working version of the MIWatch map is currently embedded on the test site at [www.miwatch.org/cmu](http://www.miwatch.org/cmu) viewable to users who are given the hyperlink.

Implemented map features:

Users can search a zip code for locations situated in a selected x radius, which returns locations that fit the search criteria. Users can then filter down locations by selecting Mental Health or Substance Abuse categories. A list of relevant locations are shown from which a user can select a location. A display box will be shown attached to its corresponding pin on the map with tabbing that separates information into General, Directions, Contact, Services and Payment. Users can then select the 'Directions' tab to enter their personal address and retrieve travelling directions to selected location or browse through the tabs for more information about the selected location.

Unable to complete:

Locations of the map are currently limited to those in the NY/Manhattan area and New Jersey. Filters are limited to Mental Help and Substance abuse. While the client did want a broader database of locations, the client contacts were unable to provide the team with the necessary data for additional locations.

Changes in initial implementation plan:

The team was originally going to make changes to the site layout of MIWatch.org, but due to limited access to the site code and directions by the client to focus on implementing a successful map.

Project beyond the semester:

The client will be provided a DVD of all work done on project over the course of this semester along with detailed documentation on how to make updates to the current database along with other informational documents late teams can refer to for what work was done.

### 3. Defect Report

The only known defects with this application are the result of the data being used. As outlined in our presentation and our phase reports, the type of data required for our system must undergo extensive cleaning on both the part of the sender and receiver. This was not an aspect our team was required to focus on and we thus worked with data in the state that it was provided to us. The following issues can be duplicated:

- Map pins correspond to multiple locations
  - Searching with any zip code in the Manhattan and New Jersey area may bring up a list of locations that appear to have pins on the map corresponding to multiple locations. This is because facilities can reside in the same building and thus are considered by the geo-coordinates to be the same location.
- Doing a search with all possible category choices (2 at this time) may return a list that contains locations not corresponding to the filtering options provided.
  - This is because our system assumes at this time that there are only two filtering types: “mental health” and “substance abuse”. In the last weeks of the project, we were provided with New Jersey data that has additional category types. Thus, if all category types are selected for a search, our system returns a list of every location (making the assumption that there are only 2 categories) within the desired search radius.

There are no other known issues that exist with the application.

Though our application has been tested with Internet Explorer (6 and 7) and Firefox (3.0.x and 3.5.x) it is possible that some browser compatibility issues exist. None are standing out at this time, however.

### 4. Usability Report

A final round of usability testing was conducted with four people: a PC user using Firefox, a PC user using Internet Explorer, a Mac user using Firefox, a Mac user using Safari. Users were given a set of verbal directions with which they had to do on the MIWatch location search page.

	PC User	Mac User
Search for locations x miles radius from your home zip code.	okay	okay
Filter search by Mental Health.	Locations returned limited to mental health locations.	Locations returned limited to mental health locations.
(In map view) change filters.	Bubbles appear/disappear depending on filters applied	Bubbles appear/disappear depending on filters applied
Find details for a certain location (click name of location)	Information bubble pops up over location	Information bubble pops up over location
Find directions location selected from your current address.	Selected 'Directions' tab and entered in address.	Selected 'Directions' tab and entered in address.
Change radius of search.	Used dropdown menu to change search radius.	Used dropdown menu to change search radius.
Search a new zip code for nearby locations.	Changed zip code in text box at top of map.	Changed zip code in text box at top of map.

It is found that users needed very little to no guidance in searching for nearby mental health locations as the site provided necessary guidance to the users.

## 5. Project History and Lessons Learned

### 5.1 System Description

The MIWatch.org website strives to be a one-stop shop for information related to mental illnesses and health information, as well as polls on relevant health issues. Before the participation of the team, the MIWatch.org system consisted mainly of news articles selected by the editor and the client contact for this project, Phyllis Vine. Intended users include those who are affected by mental illnesses, or family members and friends of those who are affected.

During the project, the team implemented and embedded a custom map with mental health locations along with search capabilities onto the MIWatch.org website. The map allows users to search locations based on zip code, distance from the zip code, and types of services provided (such as Mental Health and Substance Abuse). In addition, the map allows filtering of locations based on categories of services provided (more filtering options are intended in future development). Users can also read information about mental health locations including address, phone number, coverage (what types of insurance policies are accepted), as well as get directions to the location by entering their own address. This new added functionality increases the value of the system by giving users specific information catered to their needs.

Future development plans include the addition of a greater number of locations in other metropolitan areas, as well as implementing interactive features for users such as forums to discuss locations, services, at specific locations. In addition, the system will be hosted on a public server so that it can handle higher traffic rates necessary for a full deployment.

### 5.2 Implementation Overview

The client contact first provided the team with the website and a simple map that a previous team had worked on. Because the map was coded in PHP, the team decided to translate the code into Ruby on Rails, which is a language that is more familiar to the team. This allows more efficient implementation for future development.

Later, the team attempted to gain access so that updates can be made onto the existing site. This was somewhat problematic at first, since it is risky to make updates directly onto the existing site, and the creator of the original site was unsure what access to grant and how to do so. This problem was solved by creating a copy of the site that the team can modify without interfering with the original site.

Throughout the development phase, the team continued to add new functionality to the map, including filtering, search, and get directions. The search function allowed users to search for locations based on a zip code and a radius of 1 mile up to 20 miles. The filtering option lets users decide what kinds of mental health services are most relevant to them, and display only those results. This function was originally only available after an initial search was conducted, but was made available at the beginning so that fewer search results will be displayed and thus decreasing the level possible confusion for users. In addition, the team organized the information about each location in tabs, so that information was more accessible to users.

In order to increase the number of data points, the client provided the team with contacts to receive the information from. To the team's surprise, this was the most difficult portion of the project. Later discussion with John Pierce, who was in charge of the HumanServices.net project for the five years of its development, explain the information politics involved with this type of project. Not only does incentive need to be provided for people to give information, incentives must be given to have those people keep the information up to date and relevant, because out-of-date data will greatly decrease the value of the system. In addition, the integration of multiple

databases might become a major problem in the future of the project, as the same information might be named and formatted very differently.

Throughout the project, testing was completed and the client provided feedback on updates of the system. No major technical problems were experienced.

### *5.3 Lessons Learned*

#### **Client Communication**

One of the most important lessons learned from this project is how to communicate with client and contacts who are faraway and might not have a strong technical background. In order to keep both sides (the client and the team) informed, the team provided frequent updates to the client via email and conference calls and asked the client for plans of action and feedback on the work the team has done. In communicating technical ideas, the team used examples and step-by-step instructions and confirmation to ensure that the client and team are on the same page. During the phone conference with the client and her contacts, the team found it useful to contact each member individually in order to ask them to provide information to expand the database with.

#### **Information Politics**

Information is a major hurdle in the long-term success of project, as it is difficult to find people who hold the information and give them enough incentive to provide and update the information. Although information politics made it more difficult to get the information we needed to further complete the project before the end of this phase and handing it over to a future team, this is one the most valuable lessons we have learned and realized how valuable information is, and how useful it would be for the users if the project can be completed for a variety of locations.

#### **Improving on an Existing Site**

Because the original MIWatch.org website and a separate basic map was already in place the team began the project, the team had to take efforts to gain access and work with the existing site. Because the website was developed using Movable Type and the map used PHP, the team went through a phase of code translation, and explaining to the creator of the original MIWatch.org what the team needs in order to improve the site and add our own work on top of the site. This is important lesson because most of the time it is necessary to develop new functionality based on existing ones. In addition, communicating with the creator of the site allowed the team to focus on what needs to be done and what issues might be encountered.

#### **Deploy Early and Often**

In order to allow the client to view the team's progress and comment on it, the team used a deploy early often strategy, which allowed the team to develop incrementally, and lets the client see any updates almost immediately, since each time an update was made, the team can test the newer versions and which are not incredibly different from the old versions that mistakes will be difficult to fix. This also allows the team to address and solve problems early. Overall, this has proved a very useful strategy, especially for this project because of the short amount of time the team was given to complete as much as possible, and the client can give feedback and follow along even without much instructions.

### *5.4 Assessment of Quality*

Our team employed various metrics to ensure that our project was on track and at the highest level of quality our team was able to deliver. Our metrics fell within two areas.

The first area was on the software side. Our team was required to build a mapping application for our client and this meant that our work would be used by a large population, as well built upon by other future teams. To ensure that our code would be as bug free and modifiable as possible, we employed the following software metrics. The first was that of code size. Bloated software is very common with applications today and it is never enjoyable for a developer to sift through thousands of lines of code. While larger applications obviously require more code than smaller applications, it is still important to keep code size as minimal as possible. Because Ruby was used to code our application, this meant that code could be written in a very succinct manner. Two other software metrics used were complexity and understandability. Again, Ruby is an excellent choice for minimizing code complexity as Ruby code can be written in a very logical manner and it was designed to be an easily understood programming language. Proper commenting was also employed with this metric and the code is well documented for future teams to understand how our team went about designing the application. The last software metric employed was that of program load time. Users are very aware of the speed in which a program operates and it is important to minimize the time a user is waiting as much as possible. Ruby and Rails are not known for their speed, however, our application is extremely responsive and very few calculations other than database lookups are required.

The other area was that of general project metrics. It was important that our team was on track with not only our own goals but the goals of our client. To ensure that our team was accomplishing what it needed to, we held weekly meetings to discuss what each of us had accomplished, in addition to what we would do in the next week. This worked out quite well and we never had to worry about who was doing what or which tasks were still left undone. To ensure that we were meeting our client's expectations, we pursued the policy of deploy early and often. It was important that our client saw progress and was able to provide input in a timely manner. We chose to host the application on our end and each time we made any significant changes to the application, our client was notified and often within 24 hours we received feedback. Usability was an important success metric for us and because our client was consistently seeing our progress and commenting on her interaction with it, we feel that we met this metric.

The various software and project metrics we employed resulted in a very successful project for us. Our team was consistently on task and our software metrics allowed for us to develop a very modular and bug free application. We feel that our success metrics were met and that our client was quite pleased with the progress we made.

Overall, the team used effective methods of communication and management. Although communication with the client and her contacts was one of the biggest challenges of the project due to the physical distance between the team and the client, the team was very successful in communicating with the client using email, phone conferences, and other means. The team strived to keep the client up-to-date with all developments and ideas in order to ensure that she is informed and can provide feedback. This also allowed the client to "participate" in the testing process, where she has to follow the newly added functionalities of the site, and confirm that she feels comfortable using the site.

The team was also successful in the management aspect, as all tasks were completed in a timely manner and to the satisfaction of the client. This was in part due to the effective communication between the client, contacts, and the team. In addition, the team took a task-oriented approach to the project, where necessary tasks are identified and completed within given expected time of completion. This was effective because the method made for easy version control, where the team can always rely on older versions of the code to work if new ones fail, and allowed the team to tackle one small part of the project at one time, which results in shorter cycles and allows the client to look at the project more often. Again, this is important because gaining input from the client frequently significantly reduces problems and avoids wasting time on tasks that do not suit the client's needs.

The most important lesson the team learned this semester is that there are many non-technical issues regarding a project that can actually be a significant issue later on in the project. In the planning phases, the team did not expect the amount of time that would be necessary to communicate with the client and her contacts. These additional tasks, which became apparent as the project proceeded, includes setting up and holding conference calls with the client and her contacts, and sending follow-up emails to request local mental health location

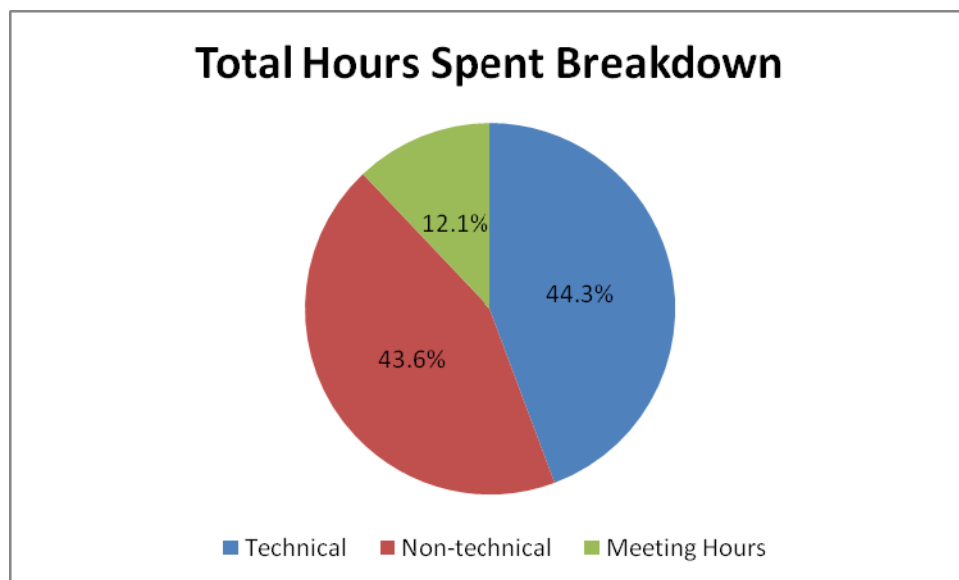
information in a specified format. Although not all contacts responded before the end of the term, the team was able to receive additional information to populate the database with.

However, these tasks proved to be incredibly useful. Not only was the team able to learn valuable communication skills in a group with unfamiliar members, some of whom are not familiar with technology, we were able to verify the difficulties and success of the project with a veteran of a similar project, John Pierce, who worked on HumanServices.net. Through John Pierce, the team came to understand information politics, its effects, and most importantly, that information is incredibly valuable, and people need incentive to share that information.

Other lessons the team learned include code translation and working with an existing site. As the team had to communicate with the site developer, Chad Everett, who was not familiar with the team's plans and needs. For example, the team had to figure out exactly what kind of access and what information on the website was necessary. In addition, the team had to spend a few weeks translating another team's work from PHP to Ruby on Rails, which was more familiar. These processes not only increased the team's technical capabilities, but made us realize that not only are the client's needs important, it was important to identify the needs of the team as well, so that various members of the project can coordinate smoothly.

### 5.5 Time Accounting

	Karen Chen	Paul Dille	Hannah Leung	Chase Midler
<b>Technical</b>	4 hours	34 hours	4 hours	24 hours
<b>Non-Technical</b>	29 hours	6 hours	25 hours	5 hours
<b>Meeting Hours</b>	18 hours			





## 6. Individual Statement of Accomplishments

### 6.1 Karen Chen

I was the project manager for the duration of this project. My responsibilities included: coordinating necessary meetings locations/times amongst our team members, with client contacts, and other IS teams; assigning and the integration of all parts needed for the four milestone documents; and putting together the midterm and final presentations.

The most important lesson/experience I took away from this project is the difficulty and limitations of working with a client without face to face time. Not only was it harder to coordinate meetings, but being able to see the client use the application and get live feedback. Discussions would also be more efficient because we would be able to immediately react with questions and answers to the client's responses.

It was also very helpful to have the conference calls and meetings we did with the client's contacts about their experiences working with similar projects. It put into perspective what we were doing with our project and the difficulties we could expect to encounter. In regards to the meeting with John Pierce from HumanServices.net, I really wished we had that meeting earlier so we could have maybe anticipated the problems associated with gathering data for the map database earlier, and possibly mitigated some of those issues.

This project also taught me the importance of communication tailored to different audiences. Because our client was less technologically experienced, the team to make sure the language used was more everyday and understandable so there was less miscommunication. i.e. in the beginning phases of the project, the client wanted a 'wiki' implemented onto the MIWatch site, and the team couldn't figure out why the wiki was needed. Later on it was clarified that what the client wanted was a discussion board. Following this, the team was careful to explain all terms we used to ensure that there were no miscommunications for what the team was doing for the project.

The most enjoyable part of this project was the team I got to work with. It was a great experience working with a team that really meshed well together because not only did everyone get along great, but we trusted each other to get our assigned work done. I have worked with groups in the past where people weren't accountable for their parts of the group work or weren't willing to contribute, and it was a great detriment to the success of the group project. In addition to the great team, I was also very appreciative of Paul who stepped up as the technical lead (something I didn't have in my 67-371 team.) He really led and pulled the technical aspects of the project together and was a great contributor to this project's success. All in all, not only was the MIWatch team very responsible and responsive to communication emails, but they were wonderful to work with.

### 6.2 Paul Dille

I was the main technical lead throughout the semester. As a result, I did the majority of the coding and was the individual responsible for communicating with our client's technical advisor, Chad Everett.

Because I was already familiar with the Google Maps API, a component critical to our application, I lead the work on implementing the map and implementing the search and filtering capabilities of our application. My familiarity with how mapping worked with Ruby on Rails enabled our team to quickly translate the previous MISM's team's work and get right to implementing new and exciting features.

In addition to doing the majority of the coding, I was also responsible for hosting our application. Our team believed in a policy of deploying early and often, and as such this required a means of having our map available to our client contacts across the United States. Because of complications that arose from attempting to deploy to our client's servers (restricted access, hierarchy of people to go through, etc), it was decided that it would be easier to just host it from our own computers. Because I had the available hardware and software, I was the one who ended

up hosting the application. Because of the use of SVN, changes to our work would be backed up, and in real time everyone on our team (as well as our clients) could see our latest efforts. This allowed us to quickly get valuable feedback and have our work organized in a sane manner.

In addition to the technical aspects of the project, I also was responsible for work done on documentation. I was in charge of the technical and administrative documentation for our application. I was also responsible for various other miscellaneous tasks required to complete the phase reports. Another piece of documentation I wrote occurred early on in the semester and it was done simply to ensure that our team was on the same page. SVN, or subversion control, was a critical component of our project and remembering complications that arose with my 67-371 team in regards to version control, I wanted to ensure that this wouldn't happen again. I therefore wrote a simple outline of how to download and install an application to communicate with the main SVN server and then steps detailing how to properly check data into the SVN server and retrieve the latest work. It was important that each of us were working with the most recent version of our application and that we would not accidentally overwrite someone's hard work. In the end, things went extremely smoothly and the team was quite applicative of having the guidelines.

Overall, this project was a great experience and taught me many things about projects in the real world. I gained a greater knowledge of the Google Maps API, as well the technical aspects of JavaScript and AJAX. I also was reaffirmed about the importance of keeping the client up-to-date on the progress of the project, in order to get quick and valuable feedback. One of the most important things that our team learned was that of information politics. Proper data was crucial for our project, since without locations or correct information for the locations, our mapping application served little purpose. Of the 4 people who promised us data sets, only 1 sent us data, and even that took some effort. As a result, we learned that information sharing is not as easy as we initially thought it to be. Our client was also surprised with this, as she too expected information to quickly be transmitted among the contacts. Going along with this, we also learned valuable information from a meeting we had with John Pierce, deputy director of HumanServices.net. He too told us about the people aspect of his project and that it took his team 5 years to get all the data they needed in a clean and concise format. He told us that it is important to have incentives in place for individuals to not only give you data but continue to update the data as necessary. This meeting was valuable to us as well as our client who was listening in. I feel projects that give teams real life hands on experience are extremely beneficial and this project was exactly that. I was quite pleased with how everything turned out in the end and had a great experience working with each member of my team.

### *6.3 Hannah Leung*

My responsibilities in this project included research, design, testing, and documentation. While the team looked at various competitors of MIWatch.org, I identified strengths and weaknesses of various sites. For example, the NIMH (National Institute of Mental Health) is more academic and not as suitable for a general audience. Some sites did not include a mapping feature. The HumanServices.net system included a large number of locations, which increased load time and can be confusing for users. Most have navigation issues, where it is difficult to figure out exactly how to find what is needed. Identifying these problems allowed the team to make the map on MIWatch.org more user friendly and efficient.

While planning on a site redesign, I created some wireframes for the client to view and comment on, in order to explain more clearly. They were useful in initiating more feedback from the client and better understanding the client needs. Although the final site redesign was minimal, as only the map was imbedded onto the site, leaving most of the site design the same, it was beneficial as a means to communicate with the client and explore the competitors of the MIWatch.org website.

During the phases of development, I assisted with testing and identifying bugs. I tested the system using multiple browsers with various options. This was done throughout the development phases, with testing completed at most updates to ensure that all functionality was implemented properly. Some minor bugs were found and fixed.

My greatest contribution to this project was documentation. I completed a significant portion to all phase reports, poster, and presentations. I mostly identified and discussed risks, risk mitigation, and project management aspects. Risks include technical failure, unsuccessful client communication, time management, and usability issues. Most of these risks were minimized through addressing directly these potential problems, with tools such as conference calling, version control, setting up small tasks, and testing early and often. In addition, I developed success metrics based on both user and technical aspects, including ease of use and lack of technical faults.

#### *6.4 Chase Midler*

Throughout the semester, the group had a great dynamic to accomplish the project. We had two programmers and two organizers. Neither of the groups was exclusive, but our strengths were separated as such. I was part of the programmer group.

As a participating member of team MIWatch, I attended meetings on a regular basis. To start the semester, our group had three meetings a week. I would attend these meetings, plan out what we needed to do before we met next, and made sure to keep in constant contact with our client. One of the most time consuming parts of our project was to bounce ideas off of each other to figure out what exactly the client wanted. Our mapping interface needed to be just right for the client, and we needed to make efficient decisions so as to please our client. Our meetings could range as short as thirty minutes, or they could last hours.

After attending a meeting the team would break up and work on their specific implementations of the project. Although Paul was the main programmer for the group, I offered him my input on changes and implementations for the site and helped with CSS changes. As the semester continued on and the project grew our database needed adjustments to be made for future teams. I changed the database to be normalized by adding new tables and fields in order to organize the data.

After a large part of the semester was over the team was organized on a conference call with our client and some of her associates. We were discussing the capabilities of the mapping interface and how we could expand the database to include other information. Being the most experienced out of the group on how to handle multiple sources of imports, I led the way on contacting individuals to collect the data. I suggested a uniform way of the contacts to export their data into a CSV document. In most databases exporting to a CSV document is simple and importing the data from a CSV file can make an import program much easier to implement. When the clients' data came in I wrote a script in order to efficiently import this data into the applications database. The script is easily configurable to be able to run on MIWatch.org's server. I also advised the client on how to easily have an update script to run on the server using a cron job.

Aside from the normal organizational and technical parts of the project I also helped a somewhat with the milestones. There were parts to each milestone that needed a technical take on, and I volunteered myself to complete such parts of the milestones as an ERD, a review of system requirements, design, etc.

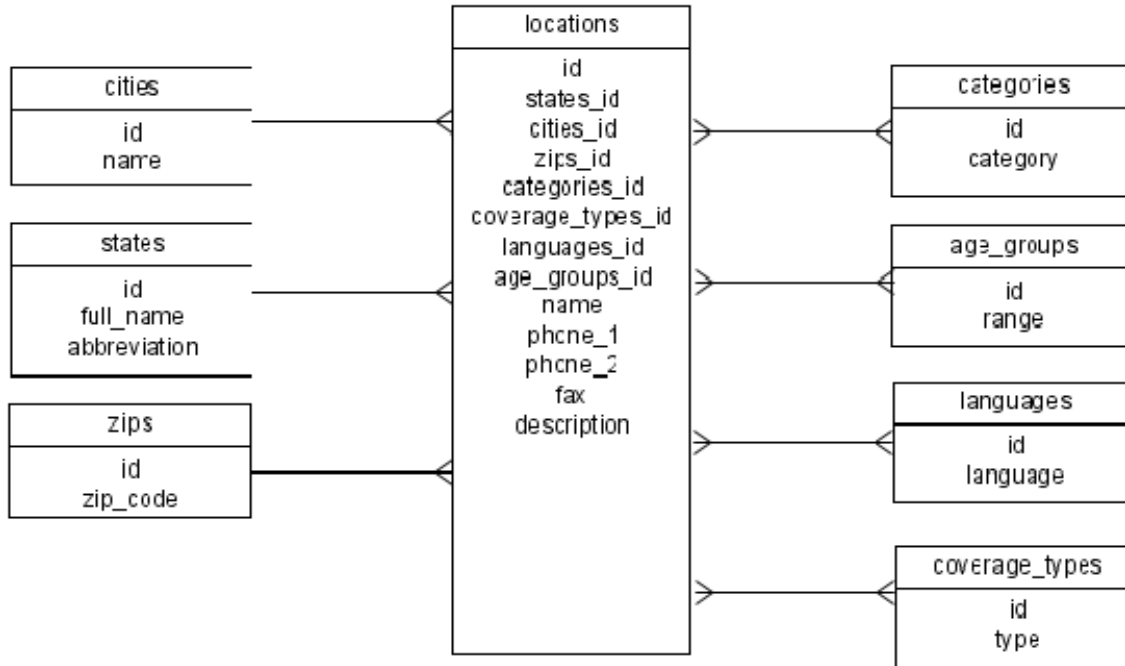
I personally learned the importance of incentives from this project. Being in charge of gathering data allowed me the experience to learn that individuals and organizations are very concerned about sharing data. Incentives are needed in order for organizations to be more agreeable with giving data.

Another lesson I learned from this project is the necessity of a good group dynamic. This is the best project group I have worked with, and the project went much smoother because of it. The ability to ask a team member to do something and have them happily willing to do it is a great environment.

Overall the project was a great success this semester. I felt like I contributed my fair share to the project and can take these valuable lessons learned onto future projects. The concepts and experiences I have gained from this project course will help me with any future projects I have.

## 7. Updates to Previous Milestones

### 7.1 Suggested Database Structuring



# MIWatch.org

## NEWS ABOUT MENTAL ILLNESS

# MIWatch.org

news about mental illness

December 2, 2009

[About Us](#) | [Donate](#) | [Find Mental Health Locations](#) | [Subscribe to RSS feed](#) | [Join our List](#) | [Submit News](#) |  [Search MIWatch](#)

**POLL: Would you forego the public option for insurance in order to pass health reform?**

Would you forego the public option for insurance in order to pass health reform?

☐ Yes

☐ No

☐ It depends

☐ Other:

[Vote](#)

[View Results](#) [Share This](#)  
Polladdy.com

**Follow Us on Twitter**



**Subscribe via Email**

**Browse by Topic**

- '08 election (4)
- addiction (38)
- ADHD (18)
- adolescents (26)
- advance directives (6)
- advocacy (48)
- alcohol (8)

### New on MIWatch

#### Mental health team in pediatric clinic



Making behavioral health a seamless part of the visit to a pediatrician is leading to early screening for emotional problems and access to easy follow-up. [Full Story](#)

Posted on: September 29, 2009 | Comments (0)

Topics: children, mental health, prevention

#### Track Legislation

[Check for Updates →](#)

Follow the progress of proposed legislation.

#### From our Readers

[Full Archives →](#)

[Care I receive in Canada →](#)

#### Did you know?

[Full Archives →](#)

Mental Illness Awareness week was first proclaimed by Pres. George Bush 20 years ago. [→](#)

#### Unions sue for mental health services in D. C.



In Washington, D.C., unions are suing the Department of Mental Health where budget, not quality concerns, have led to outsourcing mental health services despite insufficient staff to absorb the need. . . [Full Story](#)

Posted on: September 23, 2009 | Comments (0)

Topics: budget, legal, mental illness, services

#### Find Mental Health Locations



**Need help finding mental health locations in your area? → [Click here.](#)**

#### Recent Columns

A little money helps a lot of kids  
by [Carla Liso](#)

Making a difference: "The Soloist" and Steve Lopez  
by [Arlene Noto Morgan](#)

A courtroom miracle: mental health court  
by [Matthew J. D'Emic](#)

"Take These Broken Wings"  
by [Darby Penney](#)

Coming off medications: A harm reduction approach  
by [Will Hall](#)

[Full Columnist Listing](#)

[Archives by Author...](#)