

Summer 2010

# Collaboration Methods for Social Service Agencies: a Case Study of Johnson County, Indiana

Larry Noonan  
*Regis University*

Follow this and additional works at: <https://epublications.regis.edu/theses>



Part of the [Computer Sciences Commons](#)

---

## Recommended Citation

Noonan, Larry, "Collaboration Methods for Social Service Agencies: a Case Study of Johnson County, Indiana" (2010). *All Regis University Theses*. 444.

<https://epublications.regis.edu/theses/444>

This Thesis - Open Access is brought to you for free and open access by ePublications at Regis University. It has been accepted for inclusion in All Regis University Theses by an authorized administrator of ePublications at Regis University. For more information, please contact [epublications@regis.edu](mailto:epublications@regis.edu).

**Regis University**  
College for Professional Studies Graduate Programs  
**Final Project/Thesis**

**Disclaimer**

Use of the materials available in the Regis University Thesis Collection ("Collection") is limited and restricted to those users who agree to comply with the following terms of use. Regis University reserves the right to deny access to the Collection to any person who violates these terms of use or who seeks to or does alter, avoid or supersede the functional conditions, restrictions and limitations of the Collection.

The site may be used only for lawful purposes. The user is solely responsible for knowing and adhering to any and all applicable laws, rules, and regulations relating or pertaining to use of the Collection.

All content in this Collection is owned by and subject to the exclusive control of Regis University and the authors of the materials. It is available only for research purposes and may not be used in violation of copyright laws or for unlawful purposes. The materials may not be downloaded in whole or in part without permission of the copyright holder or as otherwise authorized in the "fair use" standards of the U.S. copyright laws and regulations.

**COLLABORATION METHODS FOR SOCIAL SERVICE AGENCIES: A CASE STUDY  
OF JOHNSON COUNTY, INDIANA**

A THESIS

SUBMITTED ON 8 OF JUNE, 2010

TO THE SCHOOL OF COMPUTER AND INFORMATION SCIENCES

OF REGIS UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF MASTER OF SCIENCE IN

DATABASE TECHNOLOGIES

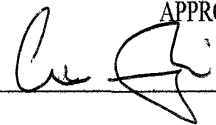
BY



---

Larry Noonan

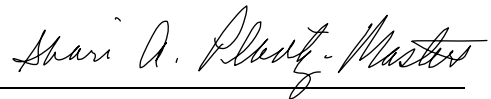
APPROVALS



Charles Thies, Thesis Advisor



Denise Duncan



Shari Plantz-Masters

## **Abstract**

Enabled by a shared Information Technology platform, social service agencies can collaborate and share information to better serve clients holistically, improve the agency's ability to raise funds, provide better statistics to stake holders, and predict social services needed in the future. Social service agencies have many barriers to overcome before successful collaboration such as security, client confidentiality, choosing a software stack, cost, and agreement among the agencies on what information they can share. This paper presents practical methodologies that can be used towards the development of an IT platform to support the collaboration of social service agencies in a local community such as Johnson County, Indiana.

## Table of Contents

<b>Abstract</b> .....	ii
<b>Table of Contents</b> .....	iii
<b>Chapter 1 – Introduction</b> .....	1
Statement of the problem .....	1
Agencies.....	1
Collect Information.....	3
Reporting.....	3
Referral .....	4
Collaboration.....	4
System Changes and Problem Resolution .....	4
<b>Chapter 2 – Review of Literature and Research</b> .....	4
Preliminary Literature Review.....	5
Messaging .....	8
Notification and Alert .....	10
Broadcast.....	11
Scheduling.....	12
Sharing Artifacts .....	13
Information Gathering .....	14
Software Solution Stack - LAMP .....	16
LINUX .....	17
APACHE.....	18
MySQL .....	19
PHP .....	20
LAMP Overview.....	20
Business Intelligence .....	21
Search Tools.....	22
Security .....	22
Change Management .....	28
<b>Chapter 3 – Methodology</b> .....	31
Work Plan, Methods and Procedures.....	31
<b>Chapter 4 – Data Analysis and Results</b> .....	35
<b>Chapter 5 – Recommendations and Conclusions</b> .....	46
Web Hosting Site .....	46
Database.....	46
Programming Language.....	48
Security .....	49
Collaboration Tools .....	50
Change Management .....	50
Business Intelligence Tools .....	51
<b>Chapter 6 – Areas for Further Research</b> .....	52

**References**..... 53  
**Appendix A – DB structure**..... 60  
**Appendix B – List of Needs for Client Assessment**..... 62  
**Appendix C – IRB Approval**..... 63

## **Chapter 1 – Introduction**

### **Statement of the problem**

Modern social service agencies need to collaborate and share information with each other enabling them to better serve clients holistically, compile statistics and data to assist them in raising funds, provide data to stakeholders, and predict social service needs in the future. Sharing information improves servicing of client needs, helps prevent lack of coordination between agencies, reduces miscommunication, reduces client effort to obtain services, and helps preserve their already limited funds.

### **Agencies**

Three social service agencies in Johnson County, Indiana led an effort to build a coalition that facilitated knowledge sharing and collaboration. These agencies included the Interchurch Food Pantry of Johnson County (IFPJC), the Ministerial Association (MA), and Christian Help Incorporated (CHI).

At the time this paper was written, the IFPJC mission was to alleviate hunger in Johnson County. As the largest food pantry in the County, it served 30,000 clients during 2009. IFPJC relied solely on volunteers to staff the pantry and funding came from over 40 churches, church related groups, individuals, non-profit organizations, and businesses. The IFPJC planned on continuing this model of operation. In addition, the IFPJC has a web hosting site that could house the electronic collaboration software for all agencies in the coalition. The hosting site costs less than ten dollars a month and includes 15 dedicated IP addresses, unlimited disk storage, unlimited MySQL or PostgreSQL databases with 10 GB of storage. It also includes support via phone, email, or chat, the PHP5 programming language, the PERL programming

language, unlimited email accounts, FTP accounts, website administration tools, and three free domains.

The MA consisted of over 40 churches from different denominations in Johnson County. They worked together to create an atmosphere of religious support for the community and the ministers. Most of these churches had benevolent funds that were used to assist people with utility bills, rent assistance, clothes, food, housing, etc. The MA planned to continue this method of operation keeping each church's benevolent funds separate.

CHI was created as a response to a lack of housing for persons who are losing their homes, living in their car, moving from place to place, or are just simply homeless by HUD definition. They relied heavily on volunteers and received much of their funding from government sources. They did not plan on changing their mission or method of operation.

These agencies were all independent organizations and used a variety of methods to keep records of their services. Most of them had little or no computer support, poor records, could not provide statistics, and did not know which clients were accessing multiple agencies. These agencies expressed a need to keep better records, collaborate with other agencies to share data, be more efficient by reducing redundant services, and prevent fraud by a few. The few who worked the system were taking from those that really needed the help. The coalition agencies all worked in different locations and needed to simultaneously use any computer application that was created to assist in their collaboration.

Most of the information the agencies collected included personally identifiable information. This information included a person's name and some other identifying piece of information such as date of birth, home address, driver's license number, or social security



number. This information had to be protected so that unauthorized personnel did not have access to it.

### **Collect Information**

Each agency had unique data collection and reporting needs. Custom applications were required to meet those needs. All agencies had some data in common and this is what could be shared across all agencies. All these agencies (and others that will join later) provided some kind of social service to families consisting of one to many individuals. The key here is an agency delivered a service to a family on a specific date and provided referrals for needs they could not meet. An example is when the IFPJC provided food to a family and then referred them to CHI for housing assistance. Capturing this data allowed agencies to report the number of families that they served, what service they provided, and other family needs that the agency couldn't meet but referred the family to another agency. Many statistics could be compiled from this information such as how many children were served, how many babies, how many are on welfare, food stamps, social security, single parents, zip code, city, township, county, etc.

### **Reporting**

The coalition needed a robust reporting capability. They needed to provide static information such as operating hours, directions, maps, mission, vision, and other guidelines. They needed to produce periodic reports to stakeholders such as how many families were served last month. They wanted the capability to dynamically populate web pages based on inputted data. For example, the IFPJC inputted data daily showing 42 families were served on a particular date. They wanted this data to populate their web page to show service statistics without manual intervention. They wanted ad hoc reporting so they could pick and choose the data they wanted. They wanted the capability to deliver reports periodically via email, RSS, or

text message without manual intervention. Agencies needed to be able to choose what data they shared to protect the privacy of their clients.

### **Referral**

Agencies needed referral information so they could provide the information to their clients. Most clients needed more than one service but did not know where to obtain those services. The coalition needed part of the solution to include a listing of agencies they could use to refer clients to. They also needed to contact the agency they referred a client to. These referrals needed to be part of the reporting capability so an agency can see how many referrals they handed out and who referred clients to them.

### **Collaboration**

The coalition also needed some collaboration tools that would help them work together such as Wikis, Blogs, Forums, calendars, and email lists. A Wiki could provide the ability for an organization or person to immediately add or modify information about agencies that provide services. This would increase the knowledge available to all agencies. A Blog could allow each agency to provide a running account of their agency activities. A Forum could allow anyone to pose or answer questions. Online calendars could keep everyone informed about upcoming events. Email lists could allow anyone to subscribe to periodic reports or newsletters.

### **System Changes and Problem Resolution**

The coalition needed a change management procedure to maintain the system including a method to submit trouble tickets when the system was not performing as required and a way to authorize changes to the system. They also needed a backup and recovery capability in addition to an audit process.

## **Chapter 2 – Review of Literature and Research**

### **Preliminary Literature Review**

Collaboration is a “process people employ when working together in a group, organization, or community to plan, create, solve problems, and make decisions” (Straus, 2002). Russell Linden says “the essence of collaboration is suggested by the word itself. Collaboration is about co-labor, about joint effort and ownership. The end result isn't mine or yours, it's ours” (Linden, 2002). His working definition of collaboration is “Collaboration occurs when people from different organizations (or units within one organization) produce something together through joint effort, resources, and decision making, and share ownership of the final product or service” (Linden, 2002). Linden lists five reasons to explain what is causing the high level of interest in collaboration:

1. Better use of scarce resources or cost savings: “Organizations often collaborate for the most pragmatic of reasons, to achieve cost savings through economies of scale” (Linden, 2002).
2. “Ability to create something that you can't create on your own: This benefit is a fundamental principle for collaborators. It makes sense to collaborate if it allows you to create something you can't make on your own. As veteran consultant Alien Hard puts it, “If you can do it on your own, don't collaborate! Collaborating is tough; if there's any other way to accomplish the task, do it yourself” (Linden, 2002).  
...collaboration isn't easy, but it does enable organizations to create products and services they can't produce independently” (Linden, 2002).
3. Higher quality, more integrated product or service for the end users: Linden uses the example of a family with multiple problems. A social worker visits on one day to discuss allegations of abuse, a nurse comes on a different day to discuss sexually

- transmitted diseases, another day a community action agency staffer shows up to discuss a job training center, and finally a school counselor calls to talk about one of the children's truancy problems. How are those services integrated? If they don't all collaborate, how can anyone be sure the family isn't getting contradictory advice (Linden, 2002). Are all these services really dealing with all the family's problems?
4. Potential for organization and individual learning: Linden uses a quote from Charles Paulk, chief information officer of Accenture Consulting who says, "When one of our consultant's shows up, the client should get the best of the firm, not just the best of that consultant". "Paulk is talking about customers' expectation of getting what they need quickly, easily, in one stop". They don't want to be passed around to find the person with the answer (Linden, 2002). How does the organization provide this kind of service (Linden, 2002)?
  5. Better ability to achieve important outcomes: Collaboration is worthwhile only if it passes this critical test: Does it help the organization achieve the outcomes that it is in business to achieve (Linden, 2002)?

Some of the factors driving collaboration today are the complexity of today's challenges, blurring of organization boundaries, the networked nature of entities today, increasing diffusion of authority over issues we face today, rapid advances in technology, and a public unwilling to accept poor performance (Linden, 2002). "The most important expectations citizens have of public and nonprofit agencies are to solve complex, cross-cutting issues, but such issues can only be dealt with through collaboration across agency boundaries" (Linden, 2002).

"The walls and lines separating organizations from one another, separating public from private sector, separating agencies from their customers and clients, are certainly blurring—if not

coming down altogether. This is one of the most powerful and fascinating stories of our new organizational society. It's happening in a wide variety of settings. And when such blurring occurs, it creates a need for collaboration” (Linden, 2002).

Advances in information technology are making it possible for collaboration among those who are geographically or temporally disbursed. Software used to support such collaboration is often termed collaborative software or groupware (Nemiro & Beyerlein & Bradley & Beyerlein, 2008). The various tools or modules needed to support a collaborative environment can be different for every project. The tools can be installed and used separately or a suite can be used that includes many or all that a project may need.

Collaboration tools can include:

- Messaging tools such as Instant Messaging, email, chat, threaded discussions, and audio/video conferencing.
- Notification and alert tools such as text messaging, email, and RSS.
- Broadcast methods such as blogs, podcasts, web pages, and newsletters.
- Scheduling tools such as calendars and project management.
- Sharing artifacts such as file sharing and Wiki's.
- Information gathering such as surveys or custom applications.
- A querying and reporting capability that allows different levels of users to request and receive data according to their access. They should be able to choose from lists of data and choose the output format.
- Search tools that allow users to search all collaboration tools for specific information.

## **Messaging**

**Email** is an asynchronous method and is a useful tool in collaboration. Email allows a person to push information to a group of people they select. All parties to an email can work on their own schedule and reply when they are ready making it easier to work across shifts or time zones (Zelenka & Sohn, 2008). Once they determine that an individual is important to their collaborative endeavor they just need the person's email address to push the information. Email tools (whether web or client based) are easy to obtain and almost always work well together (Zelenka & Sohn, 2008). Email lists can be built to reflect an official or ad hoc group of people interested in a specific topic.

Email has some negative aspects also. A user cannot control their incoming mail flow whether is it from legitimate business, friends, or considered spam (Zelenka & Sohn, 2008). Organizing large amounts of email can be time consuming and difficult (Zelenka & Sohn, 2008). An email meant for one person can be forwarded to those you might not want to read it. Discussion threads can be difficult to follow via email since many email tools are not able to sort or categorize by a particular thread of discussion (Zelenka & Sohn, 2008). Not everyone wants to use email; some people do not have a computer or email account, and some prefer the newer forms of communication such as texting, chat, and forums.

**Threaded discussions** include internet forums, message boards, discussions, and bulletin boards. These are asynchronous methods of communication that let team members discuss specific topics (Brown & Huettner & James-Tanny, 2007). A forum creates a space where individuals can post and respond to messages. These threaded discussions have attributes in common such as ways to categorize messages, threading which ties responses to the original message, all members can view all messages and their responses, and messages are available for

search (Brown&Huettner&James-Tanny, 2007). Threaded discussions make it much easier to follow a single issue or problem than email. Members can join part way through a discussion and are able to easily read previous posts and catch up on the discussion.

Using a threaded discussion tool to collaborate does require members to check for posts although some tools offer notifications via email, instant message, or RSS feeds. Many threaded discussions need a moderator to ensure that topics are categorized to suit the group, that messages stay on topic, and that inappropriate messages are deleted (Brown & Huettner & James-Tanny, 2007). Threaded discussions also require software and the hardware for it to run on, adding a cost to the collaboration.

**Chat and Instant Messaging**– Online chat is normally a synchronous method of communication performed over the Internet. It is generally an informal method of communication using text but some tools allow voice and video. Users can chat privately but many chat rooms are open to anyone who logs in. Instant messaging is a form of chat but is normally between two people but some tools allow more than two people to communicate. Instant messaging tools usually require an individual to specifically select other participants but chat rooms are normally open to whoever wished to log in.

**Audio and Video Conferencing** uses telecommunications of audio and video to bring people at different sites together for a meeting. This is often referred to as a virtual conference. This can be as simple as a conversation between two people in private offices (point-to-point) or involve several sites (multi-point) with more than one person in large rooms at different sites. Besides the audio and visual transmission of meeting activities, videoconferencing can be used to share documents, computer-displayed information, and whiteboards. Audio and video conferencing can save travel costs for an organization. (ezinearticles, 2010)

## Notification and Alert

Many collaboration tools allow users to subscribe to articles, discussions, newsletters, etc from the site. The users then receive a notification or alert that the document has been updated. Today's users want control and choices over the data they want (Linden, 2002). According to Linden these users want to choose the notification alert format they receive including email, text message, twitter, or RSS feed. This kind of pervasive or ubiquitous computing allows users to interact via many different types of computer devices such as cell phones, tablet computers, desktop computers, or laptop computers (Zheng & Ni, 2006).

Push technology is more efficient than pull technology and can be more immediate. The publisher of an artifact (newsletter, web page, document update) only has to create or modify the artifact. The user chooses to receive a notification or alert. The user may also get to choose the method of notification such as twitter, instant message, or text message. Under this scenario, push technology would send out one notification every time there is a change to the subscribed document. Pull technology is where the client polls the server every so often which means one change to a document might require the client to poll for information 100 times before it detects a change.

**Text Messaging** is often referred to as “texting” and refers to short messages sent between mobile phones over cellular networks. These short messages are called “text messages” or “texts”. Texting is generally used as a person-to-person messaging method but texts can be sent to more than one person. Users could choose texting as their primary or alternate method of receiving alerts or notifications that the collaboration site has been changed or that another user is attempting to contact them. (phonescoop, 2010)



**Email** has already been discussed but it can be used to alert or notify a user that some portion of the collaboration site has been updated or that another user is trying to contact them. Email is another option for users to choose as their primary or alternate notification method.

**RSS** is commonly expanded as “Really Simple Syndication” and is a web feed. Publishers can syndicate content and provide it in a standard XML format that allows many programs (feed aggregators) to subscribe to the information. Users can choose their feed aggregator and point it to all the RSS feeds they are interested in. The aggregator can be based on a desk top computer, laptop computer, smart phone, or some other kind of mobile device. Their feed aggregator is then set up to download subscribed articles at the user’s convenience. (press-feed, 2010) This makes RSS a pull technology rather than a push since the user determines how often to download content. (press-feed, 2010)

### **Broadcast**

**Blogs** are short for weblogs and used to broadcast information to an audience. Blogs are used like a diary or journal that displays information in reverse chronological order (newest information at the top) (Brown & Huettner & James-Tanny, 2007). Current blog software allows authors to hyperlink to other material, post photos, and attach files (Godwin-Jones, 2003). Most blogs allow readers to post comments and even rate the article.

**Podcasts** are “episodic programs delivered via the Internet using an XML protocol called RSS” (Apple FAQs: For Podcast Fans, 2010). These programs can be any combination of audio, video, or documents. The publisher podcasts the program by posting the episodes and the XML to a web server where consumers can subscribe to the podcast using podcatcher client software which regularly reads the XML and downloads new episodes (Apple FAQs: For Podcast Fans, 2010). Podcasts can deliver many types of media such as television and radio show, lectures,

performances, slide presentations, or any kind of recorded event. Consumers can then watch at their time of choosing.

**Web Pages** are a common method for any entity to publish information or data on the Internet. Once published to a web server they are available for anyone to read unless some form of security protection is put into place. Both static and dynamic web pages are useful for collaborative efforts.

### **Scheduling**

**Calendars** – there are many web based calendar software packages that allow users to maintain events for themselves, their organization, or a specific group. WebCalendar (WebCalendar, 2010) is an example of an open source web based calendar that provides many of the capabilities that users and groups need to collaborate. It allows users to share their calendars, receive email reminders for upcoming events, export events to other calendars, and to keep multiple calendars so they can keep the many parts of their life separate or combine them in one view (WebCalendar, 2010).

**Project Management** – “A project is a temporary endeavor undertaken to create a unique product, service, or result. The temporary nature of projects indicates a definite beginning and end. The end is reached when the project’s objectives have been achieved or when the project is terminated because its objectives will not or cannot be met, or when the need for the project no longer exists. Temporary does not mean short in duration” (PMI, 2008). Projects can take years or even decades but the resulting product may last for centuries.

“Project management is the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements” (PMI, 2008). Project management is comprised of five process groups: Initiating, Planning, Executing, Monitoring and Controlling,

and Closing (PMI, 2008). Managing the projects typically includes identifying requirements, addressing expectations of stakeholders, balancing constraints including scope, quality, schedule, budget, resources, and risk (PMI, 2008).

There are many project management software packages that can assist a collaboration effort by a team to manage and complete a project. These packages vary some in their features, cost, and scalability. These packages need to be compared against the particular requirements of a group so the software that is the best fit can be selected. An example is WorkZone software that provides many capabilities: tracks multiple projects, allows status alerts, time tracking, group calendar, permissions and roles, sharing work, approval process, and charts (WorkZone, 2010).

### Sharing Artifacts

**File Sharing** – Collaboration efforts almost always need to share documents and having them in a central repository is very convenient. Members are often geographically disbursed but need to work on documents jointly. Many organizations use some type of shared drive that is accessible from the Internet or at least inside the organization firewall. There are many ways to implement this capability but most still require some type of cost since some type of hardware is necessary. The collaboration group has already acquired a web hosting site that has storage space available and sufficient tools to provide this capability (IX Web Hosting, 2009).

**Wikis** are collaborative by nature featuring a loosely structured set of pages, linked in multiple ways to each other and to Internet resources (Godwin-Jones, 2003). They are by design an open editing system which allows anyone to edit pages (Godwin-Jones, 2003). Wikis have the following characteristics: pages stored in a central, shared repository, anyone can edit pages,

editing should be easy and accessible and not require special tools, and formatting information pages should be much simpler than using HTML (Woods & Thoeny & Cunningham, 2007).

Wiki sites can grow over time to be a shared repository of knowledge (Godwin-Jones, 2003). Wikipedia is an example of a very large collaborative Wiki that has grown to enormous size and has a very large number of contributors. Many Wikis are created for specific projects and can have permission settings to limit who can modify pages. There are often used for communities of practice to expand knowledge and improve practice in a specific area (Godwin-Jones, 2003).

There are many Wiki software packages that are available for use. IX Web Hosting provides a few choices as part of the hosting plan including PhpWiki. It has many features including storing to a database, authentication, blacklisting, formatting commands, notifications, searching, linking, RSS feed support, file attachments, embedded video, calendar plug ins, forums, blogs, and more (Wikimatrix, 2010).

### **Information Gathering**

**Surveys** – Businesses are interested in maintaining customer satisfaction (Gunasekaran, 2007) and this can easily extend to non-profit organizations including their clients and volunteers. Non-profit organizations often rely heavily on their volunteers so their input is important to the organization. Organizations have used surveys via paper or telephone for decades to obtain feedback from customers (Gunasekaran, 2007). The Internet has driven organizations to accomplish surveys via the web. This method has advantages and disadvantages. Advantages are lower costs, time savings, reaching distant groups, better item completion rate, and enriched survey experience. Disadvantages are lower response rates, bias due to sampling errors, privacy concerns, multiple submissions, technical issues, and

sugarcoating and clipping (Gunasekaran, 2007). There are many software packages available that allow organizations to create, distribute, collect, and analyze the results from surveys.

**Custom Applications** – A main goal of the social services coalition is to allow them to input their data into an information system so it could be analyzed and exploited to improve services. Each agency has different needs and will need a user interface and database to collect their critical data. A computer based platform and network will be essential to host these applications. The coalition has already acquired an account at IX Web Hosting and it provides all the necessary hardware and software to create custom applications for each agency. (such as directory service and specialized input screens)

**Cloud and LAMP** – (includes the LAMP parts and why they are necessary to support the collaboration tools listed previously.

The need of the agencies to share computer resources and information encompasses many different knowledge areas. They will need a complete IT solution to meet their needs including a software solution stack to build custom applications and common collaborative modules, a web hosting provider, network connectivity, software developers, agency points of contact to provide IT requirements, change management and trouble reporting procedures, data sharing agreements between agencies, security policies, backup and recovery procedures, and audit trail policies.

The agencies have already acquired a web hosting environment and expect applications to be available via the Internet. This allows them to work from any geographic location. The web host provides many capabilities including unlimited domains, unlimited web space, unlimited data transfer, eight dedicated IP addresses, programming languages such as PHP, Ruby on Rails, and Perl, internet forums, Blogs, 5,000 email addresses with a total email box of 5 GB, 100 databases with 5GB limits each, SSL capability, server load balancing, server UPS and

backup generators, FTP accounts, and web administration tools. The cost is approximately \$8.00 per month. This solution can scale by simply upgrading the hosting plan (IX Web Hosting, 2009).

Web applications provide many benefits to organizations. Security is enhanced because data and program code is only deployed on servers that are often in a data center (Db Net Solutions, 2009). Fewer people have access to the data servers and data is not stored on multiple PCs (Db Net Solutions, 2009). Geographic limits are practically eliminated as users can access the applications from anywhere that has an Internet connection (Db Net Solutions, 2009). Data centers typically offer 99.9% uptime of their servers. It can be easier to scale applications because new users only need to know the URL and have a username and password. User PCs can be out of date but still access robust application (Db Net Solutions, 2009). Clients do not normally need software licensing. Applications can be run on many platforms because they just need a browser to operate. Overall it is easier to manage a web application than a desktop or client-server application (Db Net Solutions, 2009).

### **Software Solution Stack - LAMP**

The acronym LAMP refers to a solution stack of open source software used to run dynamic Web sites or servers (JK Design, 2010). The acronym expansion is:

LINUX is the operating system.

APACHE is the web server.

MySQL or PostgreSQL is the database server.

PHP, Perl, or Python is the programming language.

These components can be acquired and utilized for no licensing fee. These components were not originally designed to work together but they are commonly available at web hosting companies.

Individuals or organizations can purchase web space at one of these web hosting companies very cheaply partially because the web hosting company does not have to pay license fees for these components.

## **LINUX**

Linux is an operating system created by Linus Torvalds. He started developing in 1991 and released the Linux Kernel in 1994 (Linux.org, n.d.). The kernel is the operating system code that runs the computer. It is under constant development by volunteers around the world. The kernel design is modular and the amount of code is small. The kernel is built to only load the modules that it needs making it very efficient. This also makes it fast and helps it to limit memory use. Linux boxes are known for stability running for months and years without needing a reboot (LinuxJournal, n.d.).

The Kernel is released under the GNU General Public License which makes its source code freely available to everyone (linux.org, 2007). It can be obtained by downloading a distribution from one of many web sites. A distribution includes the Linux kernel, a collection of programs and applications that run on Linux, and an installation program (Rosen & Host & Klee & Farber & Rosinski, 2007). There are hundreds of different distributions of Linux. The many distributions work on different hardware and have different purposes. Some of the processors that it runs on are PowerPC, Macintosh, Dec Alpha, Sun Sparc, and mainframe. Some are for servers, others for home use, and many have very specific uses. Any vendor can extend the Linux kernel for their particular piece of the software market (Rosen & Host & Klee & Farber & Rosinski, 2007).

Linux has gained wide popularity due to its capability and free distribution. It has a large user community and many vendors offer professional support packages for organizations. It is a

direct competitor to Microsoft operating systems and proprietary UNIX operating systems especially in the server market. Major organizations such as IBM, HP, SUN, and Novell support the development and use of Linux (Rosen & Host & Klee & Farber & Rosinski, 2007). This support by large organization and individuals combined with free distribution has helped Linux take a large share of the Internet server market.

### **APACHE**

The Apache web server is an open source project and has been free since its inception. It is developed, maintained, and enhanced by a worldwide group of volunteers. It is administered under the Apache Software Foundation. The project started in 1995 by patching the National Center for Supercomputing Application (NCSA) httpd web server (the first web server) (Rosen & Host & Klee & Farber & Rosinski, 2007). Later versions of Apache were rewritten to eliminate remaining bits of the NCSA httpd web server. Apache can be downloaded from Apache.org and many mirror sites.

Apache became the most popular web server on the Internet shortly after its release in 1995 (Rosen & Host & Klee & Farber & Rosinski, 2007). Currently over 73% of web servers use Apache and less than 20% use Microsoft products (ServerWatch, 2008). Apache is almost always included in distributions of UNIX type operating systems (includes Linux). The most difficult part may be to decide on what distribution of operating system to use since there are hundreds of Linux distribution available. Choosing the right distribution can make installation and configuration much easier since there are likely to be installations that will fit your particular needs.

Apache is a fairly small web server engine to handle static web page display. There are over 30 modules that can be installed with Apache to provide other capabilities. Web server



administrators can choose those modules that fit their needs (Rosen & Host & Klee & Farber & Rosinski, 2007). This keeps the Apache installation at a minimum and keeps it from being a resource hog. An example of a commonly used module is `mod_access` (apache.org-modules, n.d.). `Mod_access` includes `.htaccess`. This is a well known ability to restrict access to parts of the server. It is often used to password protect a directory on the server (apache.org-modules, n.d.). This allows the administrator to restrict access to authorized users. This ability is critical to many organizations that want to work via the Internet but need to protect their data.

## MySQL

MySQL is touted as the most popular multi-user, multi-threaded SQL database server that is free of charge (MySQL, 2010). MySQL does have some licensing issues so it is wise to research the available licenses against anticipated use (FSF, n.d.). A misinterpretation of license policies could be costly to an organization. MySQL was developed by MySQL AB (Sweden) in 1995 (MySQL, 2010). MySQL AB was recently purchased by SUN Microsystems (MySQL.com, 2008). This has caused much concern in the open source community that MySQL will become proprietary. SUN stated they plan to continue MySQL as open source (MySQL.com, 2008).

MySQL is a fully capable database. It includes stored procedures, triggers, cursors, Secure Socket Layer (SSL), numerous functions, libraries for multiple programming languages, and is Atomic-Consistency-Isolation-Durability (ACID) compliant (MySQL, 2010). It is considered easy to use promoting rapid web development. MySQL also offers multiple storage structures that satisfy many different data needs. MySQL runs on most platforms available today including Solaris, Linux, UNIX, HP-UX, and Mac OS X (Rosen & Host & Klee & Farber &

Rosinski, 2007). This makes it a good choice for organizations that may have multiple platforms they want to host a database on.

## **PHP**

“PHP is a scripting language and interpreter that is freely available and used primarily on Linux Web servers. PHP was first developed in 1995 by Rasmus Lerdorf (Rosen & Host & Klee & Farber & Rosinski, 2007). PHP, originally derived from Personal Home Page Tools, now stands for PHP: Hypertext Preprocessor” (SearchEnterpriseLinux (n.d.). Lerdorf released PHP into open source and now many volunteers around the world work on improving PHP.

PHP script is embedded in a web page along with Hyper Text Markup Language (HTML). The web server sends the file to be interpreted by PHP (generally Apache PHP module) and the PHP interpreter sends HTML back to the web server. The web server then sends it on to the user who requested the page. One result of this is that web users never see the actual PHP code even if they use the view > source option in their browser. This helps to maintain security. Using an interpreter instead of CGI also increases security.

TIOBE index shows PHP to be the fourth most popular programming language behind Java, C, and C++ (TIOBE, 2009). Usage stats for April 2007 show 1.2 million IP addresses and over 20 million domains using PHP (cz.php.net, 2007).

## **LAMP Overview**

LAMP is more than the sum of its parts (ServerWatch-LAMP, 2005). The open source parts of LAMP have worked to increase its compatibility. Extensions have been created to improve the cooperation between the components. There are Linux distributions tuned for LAMP applications. Apache has modules that can be added to work with PHP or Perl (ServerWatch-LAMP, 2005). These modules eliminate the CGI problems and use a PHP or Perl

interpreter. MySQL has extensions that allow PHP to access the database easily. This makes it easier to develop data driven apps using PHP. PHP code can be directly embedded in HTML files (ServerWatch-LAMP, 2005).

Apache can only serve static HTML pages (ServerWatch-LAMP, 2005). It needs the PHP language to add dynamic functionality. MySQL is needed to maintain persistent data between sessions and to easily reorganize the data based on user preference. This allows a few simple queries to display data in many different ways. A few simple criteria can be passed to MySQL to allow users to drill down into data. Linux provides a high quality, capable, reliable operating system for all of this to run on (ServerWatch-LAMP, 2005). Last but not least, an organization can download all this software for no cost. They can also change and configure the software to meet their needs.

### **Business Intelligence**

“Business intelligence (BI) allows people at all levels of an organization to access, interact with, and analyze data to manage the business, improve performance, discover opportunities, and operate efficiently” (Howson, 2008). The non-profit coalition has people at many levels including local media, board of directors, volunteers, donors, and other agencies. BI can reveal performance, efficiency, and opportunities (Howson, 2008). BI includes processes and technologies that allow all the users to access and analyze data (Howson, 2008). BI can provide insight for management, provide history and trend analysis, aggregate information based on categories, and provide ad hoc reporting and analysis across functions such as people, family, agencies, services, geography, etc. BI is useless unless people act on the information to provide value to the business (Howson, 2008).

BI can provide information about what is happening within the business (Howson, 2008). Organizations used to have to wait until quarterly reports were run to identify problems. BI allows users to drill down into data, build pivot tables, and slice data many ways helping users to isolate problems and then fix those problems before they cause harm to the organization (Howson, 2008). BI allows business users to run queries without having a software engineer create a new program module.

BI often consists of a data warehouse, an extract-transform-load (ETL) function, front end tools for the user, and a network infrastructure. A data warehouse normally combines data from different systems or agencies to make query and reporting more efficient. ETL is the logic to pull data from the various transaction based systems, transform it so that it can be standardized with other data, and then load it into the data warehouse. On Line Analytical Processing (OLAP) is a term often used for the front end tools (often web based) the user sees to access and analyze data (Howson, 2008). The OLAP tools provide a method for the user to choose what dimensions of data they want to see and what output format they desire.

### **Search Tools**

Site search is a common and very helpful tool for collaboration users. Sites can easily grow large and that makes it difficult to find specific content even if navigation is well done. “Sphider” (<http://www.sphider.eu/about.php>) is one example of many search engines that can be used on an individual web site. Sphider is free to download, written in PHP, and uses MySQL as the storage tool for the index.

### **Security**

“Web application security is a young and evolving discipline” (Lerdorf & Tatroe & MacIntyre, 2006). On January 12, 2009 over 30 US and international cyber security

organizations met and released a list of the 25 most dangerous programming errors (SANS, 2009). Some of the involved organizations were Symantec, Microsoft, Department of Homeland Security's National Cyber Security Division, National Security Agency's Information Assurance Division, Oracle, VeriSign, and the University of California (SANS, 2009).

The experts agreed quickly on the major issues and realized that these errors are not well understood by programmers (SANS, 2009). They found the errors are not widely taught by computer science programs and organizations don't test for these errors (SANS, 2009). The SANS director said it is now time to fix these problems. "First we need to make sure every programmer knows how to write code that is free of the Top 25 errors, and then we need to make sure every programming team has processes in place to find, fix, or avoid these problems and has the tools needed to verify their code is as free of these errors as automated tools can verify" (SANS, 2009). A few of the top 25 errors are mentioned below.

**Improper input validation** is "when software fails to validate input properly, an attacker is able to craft the input in a form that is not expected by the rest of the application. This will lead to parts of the system receiving unintended input, which may result in altered control flow, arbitrary control of a resource, or arbitrary code execution" (SANS, 2009). An example of this problem is when an online shopper enters a negative number as a quantity when buying an item. If the program doesn't check for negative numbers, the shopper could have their account credited instead of debited (SANS, 2009). A user could also enter values followed by two dashes which could be recognized as a comment by the system. This could eliminate the remainder of a SQL statement which would disable other security related logic in the statement (SANS, 2009). There are a number of ways to mitigate improper input validation:

- “Understand all the potential areas where untrusted inputs can enter your software: parameters or arguments, cookies, anything read from the network, environment variables, request headers as well as content, URL components, e-mail, files, databases, and any external systems that provide data to the application. Perform input validation at well-defined interfaces” (SANS, 2009).
- “Assume all input is malicious. Use a standard input validation mechanism to validate all input for length, type, syntax, and business rules before accepting the data to be displayed or stored. As an example of business rule logic, “boat” may be syntactically valid because it only contains alphanumeric characters, but it is not valid if you are expecting colors such as ‘red’ or ‘blue’ “ (SANS, 2009).

**Improper escaping of output** can allow an attacker to change commands that are sent to another component and insert malicious commands (SANS, 2009). “If an application uses attacker-supplied inputs to construct a structured message without properly encoding or escaping, then the attacker could insert special characters that will cause the data to be interpreted as control information or metadata. Consequently, the component that receives the output will perform the wrong operations, or otherwise interpret the data incorrectly” (SANS, 2009). An example is a user enters some text followed by a bar. The application treats it all as a string and passes it to the server. The server interprets it as two commands since the bar is a character recognized as separating commands. Some mitigation strategies are:

- “Use stored procedures or other structure mechanisms if available, especially if they use strong typing. Such procedures will typically automatically escape or encode special characters for communicating with the component” (SANS, 2009).

- “Understand how the data will be transmitted between different components. Study all expected communication protocols and data representations to determine the required encoding strategies” (SANS, 2009).
- “Apply the proper encoding at each interface. Use libraries that use the expected encoding or handle it automatically, such as the ESAPI Encoding API. Alternately, use built-in functions, but consider using wrappers in case the built-in functions are discovered to have a vulnerability” (SANS, 2009).

**SQL Injection** “targets software that constructs SQL statements based on user input. An attacker crafts input strings so that when the target software constructs SQL statements based on the input, the resulting SQL statement performs actions other than those the application intended” (CAPEC, 2009). SQL injection is successful when the application does not validate input. “A successful SQL injection exploit can read sensitive data from the database, modify database data (insert/update/delete), execute administration operations on the database (such as shutdown the DBMS), recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system” (OWASP-Guide to SQL Injection, 2009).

“There are two complementary and successful methods of mitigating SQL injection attacks: parameterized queries using bound, typed parameters, and careful use of parameterized stored procedures” (OWASP-Guide to SQL Injection, 2009). Parameterized queries are the easiest to adopt and work with Java, .NET, and PHP. A parameterized query separates the query and data by using placeholders which are called ‘bound’ parameters (OWASP-Guide to SQL Injection, 2009). The query template is built and sent to the database while the parameters are built separately and sent to the database when the query needs to be executed (OWASP-Guide to

SQL Injection, 2009). The database makes sure the parameters are only used as values in the query and prohibits SQL injection attacks. Parameterized stored procedures store the query in a database stored procedures rather than in the PHP code. Using parameterized bound queries in combination with parameterized stored procedures makes it very unlikely that a SQL injection attack will be successful (OWASP-Guide to SQL Injection, 2009).

**Cross-site scripting (XSS)** “is the most prevalent and pernicious web application security issue” (OWASP-Cross Site Scripting 2009). “XSS flaws occur whenever an application takes data that originated from a user and sends it to a web browser without first validating or encoding that content” (OWASP-Cross Site Scripting 2009). This allows the attacker to execute scripts (usually JavaScript) in the victim’s browser which can then allow the attacker to hijack user sessions, deface web sites, insert hostile content and take over the user’s browser (OWASP-Cross Site Scripting 2009). There are three types of XSS attacks: reflected, stored, and Document Object Model (DOM) injection.

Stored XSS attack allows an attacker to store malicious code in the site’s database or a file and then that malicious code can be downloaded or used by many unsuspecting web users. Online forums are a good example since users are allowed to enter HTML to format their posts (OWASP-Cross Site Scripting 2009). Reflected allows the attacker to manipulate web pages sent to unsuspecting users. The attacker may be able to modify the page and convince the user to click on a button or URL and send them to a malicious site (OWASP-Cross Site Scripting 2009). DOM XSS attacks allow the attacker to modify or add JavaScript code that is sent to a user (OWASP-Cross Site Scripting 2009).

It is best to use a combination of the following methods to prevent XSS attacks. Input validation is checking all input data for length, type, syntax, and business rules before accepting



it for use (OWASP-Cross Site Scripting 2009). Output encoding is making sure that all user supplied data is encoded before rendering it as output and set the character encodings for each page (OWASP-Cross Site Scripting 2009). Do not rely on “blacklist” validation to prevent XSS attacks (OWASP-Cross Site Scripting 2009). Some programming languages have libraries that can help deal with XSS attacks. PHP has an OWASP PHP Anti-XSS library and .NET has the Microsoft Anti-XSS Library (OWASP-Cross Site Scripting 2009).

GSA says “identity theft is one of the fastest growing crimes in the country and concern by the public is growing rapidly with each new data breach exposed” (GSA). “Identity theft occurs when someone uses your personally identifying information, like your name, Social Security number, or credit card number, without your permission, to commit fraud or other crimes” (Identity Theft, 2009). ). Gender, date of birth, and 5 digit zip is enough to identify 87% of the US population (Dr. Sweeney). Identity theft happens to as many as nine million Americans a year and takes many forms. Thieves rent apartments, obtain credit, or establish utilities in someone else’s name (Identity Theft, 2009). Identity information is stolen in many different ways. ITRC shows data breaches have increased 47% from 2007 to 2008. ITRC reports only 2.4% of these breaches had encryption or other protection methods in place. Only 8.5% of breaches had password protection. It is clear that unprotected data is at the most risk (ITRC). There are three main security processes working together to provide access to assets in a controlled manner (Todorov, 2007). There are authentication, authorization, and accounting (Todorov, 2007).

Authentication is the method used to verify a person. Computers need to model a real world person or organization and they often do this with a user ID and password. The person must have initially registered with a computer application and provided some personal details.

The application owners must have also approved the person to access the application. The user ID is the identification and the password acts as the authentication. A successful login is used as proof of identity and lets the person access the computer application but does not necessarily allow them access to any information (Todorov, 2007).

Authorization is the process of determining whether an already authenticated user is allowed to access information (Todorov, 2007). It is used to provide granular control to the owners of the data (Todorov, 2007). Access control lists are normally maintained that determine which users can access data (Todorov, 2007). Online banking is a good example. Thousands or millions of users are authenticated when they log in but are only authorized access to their own accounts. This protects individual user financial assets and personal information.

Accounting holds users responsible for their actions (Todorov, 2007). Computer applications should maintain audit trails so they can track who modifies data. This can help maintain data integrity.

### **Change Management**

Change Management is a process that maintains integrity of the IT environment of an organization while executing changes to that environment (SkillSoft, 2006). The Information Technology Infrastructure Library (ITIL) is an International standard and library of best practices for the IT community. ITIL divides Change Management into six steps: Change request, change classification, change authorization, change development, change review, and change closure (SkillSoft, 2006).

1. Change request is often made through a standard document called Request for Change (RFC). This step includes an initiator asking for a change to software,

submitting the RFC to the change manager for approval, and ensuring the RFC is complete.

2. Change classification specifies a priority and categorizes the RFC. This helps to determine the effect of change on the IT environment. Priority levels can be: emergency, high, medium, or low. Categories can include:
  - Major – changes which affect all users.
  - Significant – changes which affect multiple departments.
  - Minor – changes which affect a group and risk is low.
  - Standard – changes which follow a predefined path and contain a predetermined solution.
3. Change authorization is the step where the RFC is authorized for work or not. This step includes reviewing the RFC for missing information, reviewing to make sure the priority and category are correct, and ensuring the change is authorized.
4. Change development is the step that develops a plan for implementing the RFC. This step includes scheduling the change, appointing a change owner to manage the changes, and identifying the necessary steps to implement the change.
5. Change review monitors the changes in the working environment and conducts post implementation reviews to ascertain differences between plans and actual effects.
6. Change closure ensures the implemented change meets specifications, determines if the change needs to be backed out, and closes the RFC if specifications were met.



## **Chapter 3 – Methodology**

### **Work Plan, Methods and Procedures**

The research began with an introduction of the research area and a literature review. The researcher used a qualitative study using single case study methodology based on a coalition of social service agencies. The researcher interviewed key personnel from some of the social service agencies, observed operations at the agencies, participated in operations at some of the agencies, and collected documents used by the agencies in their daily routine. The investigation concluded with an analysis and results section using information from the literature review compared to the interviews and collected documents. The goal was to provide a practical methodology for a coalition of social service agencies to implement collaborative information sharing.

Qualitative research allowed the researcher to dig deep to get an understanding of the phenomenon in this study. Qualitative research focuses on phenomena that occur in natural settings and involve studying them in all their complexity (Leedy & Ormond, 2005). The issue here was a social phenomenon and didn't lend itself to objective methods. The researcher had to use his ability to interpret and sense of what he saw (Leedy & Ormond, 2005). Qualitative studies can serve several purposes. "They can reveal the nature of situations, settings, processes, relationships, system, or people" (Leedy & Ormond, 2005). They can assist a researcher to gain new insights about a particular phenomenon (Leedy & Ormond, 2005). They can provide a way for the researcher to judge the effectiveness of a practice or innovation (Leedy & Ormond, 2005). The coalition of agencies that wanted to collaborate formed a complex relationship system.

This problem had unique properties since it focused on specific social agencies in one County of Indiana. The researcher chose the single case study research design to investigate this poorly understood situation. This method allowed the researcher to collect extensive data including observations, interviews, participate in routine operations, and collect documents about the social agencies. The case study allowed the researcher to include information surrounding the case such as the physical environment, economic factors, and social factors that have an important bearing on the problem.

The research problem is a unique case study but other communities may benefit from the data collected and the solution implemented. Most communities have homeless task forces, food pantries, and churches that assist individuals. These communities have similar problems but may have different resources, different levels of volunteers, and lesser or greater level of problems in these areas. New York City has over 10 million people and is a very different scale than Johnson County, Indiana that has about 130,000 people.

There were some factors that limit the possible solutions for this collaborative effort. The homeless task force has very limited funding and has to deny some people assistance because they routinely run out of money. They can only afford to spend a very small amount of money on technology if anything at all. The ministerial association of churches has many benevolent funds but they also routinely run out of money assisting people. The food pantry has better funding but they also want to focus the vast amount of their resources directly on their clients. The economic downturn during this research period also impacted the coalition ability to invest money in a solution. Most of the individuals running the agencies were volunteers which affected the agencies. Their funding was mostly from local donors with some grant money occasionally infused into their agency.

The researcher collected multiple forms of data including observations, interviews, participation, and various documents from the agencies. This allowed the researcher to construct a more meaningful picture of the coalitions' problems (Leedy & Ormond, 2005). This allowed the researcher to analyze many separate pieces of data to look for a convergence of the data (Leedy & Ormond, 2005).

The researcher spent considerable time at the IFPJC observing daily operations and interactions among those running the agency. The researcher had also previously created a desktop software program to allow IFPJC to keep track of clients, meals distributed, food stamp users, homeless individuals, and location data such as town, zip code, and school district. This software program was written in Microsoft FoxPro and operated on one computer. It had over three years of data the researcher was able to analyze. The researcher also observed one of the churches benevolent committee operations.

The researcher conducted in depth interviews with the IFPJC manager, the CHI manager, and two of the principal officers of the ministerial association. The IFPJC manager volunteers about 30 hours per week, has a Masters Degree in Social Work, and coordinates extensively with other community agencies that help those in need. She has been working this effort for over three years. The CHI manager has managed the agency for many years and is a trained social worker. The Ministerial officers interviewed were respected clergy for local churches.

The researcher collected numerous documents from all three agencies. The IFPJC supplied the entire database of clients for the past three years including the database structure. The ministerial association provided several forms from different churches that were used to capture client information when they came to the churches for assistance. The researcher

collected blank and sample intake documents from CHI as well as some of the reporting requirements they have for federal money they receive to help clients.

The researcher participated in distributing food at the IFPJC and in many of their administrative board meetings where they discussed their major problems and issues. The researcher also participated in data collection and reporting activities for the IFPJC. The database with three years of data provided a wealth of information. The researcher also had some participation with one church's benevolent fund. This church was part of the ministerial association. The researcher also participated in multiple meetings where two or more of these agencies were discussing the IT solution they needed that would allow them to collaborate and share data and resources to improve their agencies operations.



## Chapter 4 – Data Analysis and Results

The researcher started with interviewing the IFPJC manager since she was the one who conceived the idea of a non-profit coalition sharing information. The IFPJC manager had already discussed the idea with the CHI manager and MA clergy. They had all agreed to work with the researcher toward the goal of a collaborative shared IT platform. The researcher had previously worked with the IFPJC manager on the desktop application used by the IFPJC since 2006. This previous working relationship made it easy to discuss in depth the goal of the three organizations.

The interview started with a broad over view of the economic situation of the agencies. The IFPJC manager stated that her agency had enough money to keep up with the needs of their clients but not much left for anything else. They are able to pay rent and utilities, buy necessary equipment, pay for insurance, and repairs, but did not pay any employees. The IFPJC manager is a volunteer spending about 30 hours a week running the agency. The IFPJC previously agreed to pay for the web hosting site mentioned earlier in this paper. She felt the agencies could not afford to pay for software development of the magnitude they would require. They needed a volunteer which is the method they use to accomplish almost all their work.

The researcher already had access to the IFPJC desktop database application that was used to keep track of client information. The IFPJC manager discussed with the researcher how the three agencies wanted to expand on that capability. She stated that the three agencies often assisted the same clients and it would be very helpful if they could share client and family information. It could prevent many duplicate forms and input time. The current desktop application is limited to one computer so it wouldn't be acceptable for all three agencies to use.

A new IT platform would have to allow multiple users in different geographic locations to use the application simultaneously.

She gave an example of a family of five that came to the IFPJC. IFPJC volunteers would sit down with the client and fill out a form or directly input the family information into their computer application. The information consisted of all family member names, date of birth of each member, marital status, address, phone number, date of visit, were they on food stamps, were they homeless, county, and school district. The family might then go to CHI or the MA to get assistance on housing or utilities they couldn't afford. Those agencies would record much of the same information. The IFPJC manager said all three agencies felt it would be helpful if they could look up the family in a computer application and if the family had already been entered by any agency, then they could just select and modify the family information instead of recreating it.

The IFPJC manager provided some yearly statistics for their agency and how they use some of the information. During 2009, over 8,000 families with over 30,000 members visited the IFPJC. They distribute nine meals per person so they handed out 270,000 meals during 2009. At the end of each day, the IFPJC manager would run a report from their computer application and balance it against their sign-in log. At the end of each month she would run a report and provide it to the board of directors and other interested parties. The local newspaper would often call her to ask about her statistics and thoughts since it was a good indicator of the economic health of the county. They also use the information to keep their donors informed of their needs and to apply for grants. She stated that without donors they would be out of business in a very short time.

The IFPJC stated that privacy was very important for their clients. Much of the information they acquired in their database was client personal information and had to be protected from any unauthorized access. Also, each agency needed to have the capability to choose what information they would share with other agencies. She felt it was very important that her agency have the ability to own and control their data. She felt that other agencies might want to join the coalition at a later time and so the new platform would need to accommodate new users. Also, some agencies might decide to leave and they should be able to extract their data from the shared platform without causing any harm to the other agencies.

The IFPJC manager stated that their computer environment was acceptable but she knew the agencies had a variety of computers with some old, some new, different operating systems, and different networking capability. She felt the new IT platform needed to support low end or old desktop computers and couldn't require them to have any particular software. She did say that IFPJC would at least have an Internet connection and a browser installed on the computer. She said they could utilize a specific browser if necessary but felt it would be easier for them if the platform could support any common browser. She also felt the platform should support Windows, MAC, or a Linux operating system for the users. After some discussion she understood that this might limit the software development environment. Specifically, she mentioned that the current application provides a Microsoft Excel spreadsheet and she realized that some operating systems might not support that type of output.

The researcher inquired about types of users and the IFPJC manager responded that they would have several users that could input client information and that they would normally work at the IFPJC site but sometimes she and others would take home paper intake sheets and input the data at home. She said they also wanted a public front end to the application. She further

elaborated that she meant a public web site available over the Internet that anyone could look at aggregated information that was similar to what she provided to the local newspapers and all her donors. She wanted the new IT platform to dynamically populate the public site with information they entered. She strongly stated that they had to be careful with the aggregated information so that no individual client could be identified. She was adamant that client confidentiality was paramount.

The IFPJC manager went on to explain that the agencies wanted a way to collaborate with each other about client needs. She gave an example to explain using the family of five previously mentioned. They might come to any one of the agencies first but have needs that take all three agencies to fulfill. Her example assumed they came to the IFPJC first. The IFPJC would do an intake and start filling their food order. The IFPJC manager often acts as a social worker for families in need and this routinely leads to her recommending other agencies that can help with needs other than food. She wants the new IT platform to provide a means for her to refer a family to the other agencies. Since she has already done an intake and talked to the family, she wants to send a notification through the new IT platform to the other agencies so they know the family is coming, have the information already inputted for them, thus saving everyone time.

Furthermore, the IFPJC manager wants the new platform/application to have a list of agencies available to intake personnel so they can easily refer the client to these other agencies. She said it would be great if it was more than a list; that it would categorize or tag the agencies with key words so that intake personnel could look for agencies that provide utility assistance, child care, etc. She mentioned that the agency information would need to include phone,

address, points of contact and descriptive information so the intake worker could make an intelligent referral for the client.

The IFPJC manager mentioned that it would be helpful if the platform provided a way for agencies to discuss issues or specific clients. She mentioned that something like a chat program or discussion forum that might allow the agency managers to work on a problem together. She noticed that Internet forums allow you to start a thread and keep the comments for later reference. She felt this would be helpful for the three agencies.

The researcher collected various documents from the IFPJC. A copy of the entire application and database was made available to the researcher. This was easy since he was the developer and had been maintaining the application for a few years. A copy of the output report used by the IFPJC manager with her board of directors, donors, and the local newspaper was obtained. A copy of the sign-in sheet was obtained. A copy of the paper intake form was obtained.

The researcher had previously written the desktop database application in Microsoft Visual FoxPro. He was able to review the requirements in depth and create a software application that had met the IFPJC needs for the last four years. This application was not suited to geographically disbursed users. This experience helped the researcher to understand the data structures that supported the IFPJC. The data structure keys on families, their members, the visit date, and their status on the visit date. A review of multiple database records along with confirmation by the IFPJC manager revealed that many of these families were dynamic. This meant that their address might change every few months, the members of a household would change often due to divorce or extended family members moving in with each other, they might be homeless one month but not the next, they would be on food stamps during one visit but not

the next, phone numbers changed often, and changing addresses sometimes meant changing school districts.

The key to understanding the data was that “family” information was only good for the day they visited the IFPJC. It could easily change the next time they visited the IFPJC. The same family identification number was used in the database from one visit to another but clearly the family could be different during each subsequent visit. The new platform needed to change the way families, visits, and clients were maintained in the database. The IFPJC manager and researched discussed this in depth and agreed that the key element are the visit and the clients involved in that visit. This is slightly different that the current system which incorporates the idea of a family. The new platform needs to connect clients to a visit and not try and create a family. A visit would need to include the attributes that are the same for all clients in that visit such as date, address, school district, and phone numbers. The client information would include name, date of birth, social security number, drivers license number, etc. A third set of information would include information that is specific to a client and that specific visit such as dependent children, homeless, food stamps, and marital status.

The researcher spent several days working at the IFPJC doing several jobs. The researcher spent time working in the warehouse section where all the food is received, unloaded, and sorted. He worked in the food distribution area where they volunteers receive orders from the intake person and select the correct amount of food, bag it, and help the client out to their car if necessary. The researcher spent most of his time working in the intake area where the client is interviewed, data collected, and orders sent to the distribution area. Reports are also created here at the end of each day. The intake area was of most interest since it encompassed most of the tasks that the new IT platform will fulfill. The researcher was able to validate and cross check

the interview information from the IFPJC manager against the documents obtained and the database research. The results confirmed that the database structure should be based on visits and clients with the information divided into three areas; visit, client, and client-visit information.

The researcher also attended a few board of director meetings to hear what kind of information they found useful in directing the pantry. The board of directors for the IFPJC consisted of approximately eight personnel. They were mostly interested in the monthly reports produced by the current computer application. They were also concerned that client information was well protected.

The researcher interviewed the CHI manager and one of the CHI case managers next. She provided two different intake documents and a few hours of in-depth interview. She echoed much of what the IFPJC manager provided with one key difference. That difference was the concept of a “case” in addition to the “visit” utilized by the IFPJC.

The CHI manager explained that most of the clients or client groups that request assistance from their agency are not the same as a visit to the IFPJC. A client or client group visits CHI and often requires more than one type of service. It may also require CHI to spend hours or days of effort over a longer period of time (weeks or months) to meet the client’s needs. The IFPJC usually completes their service of providing food in the same day. CHI refers to this as managing a case. A case manager is assigned to the case and they will work all the issues for that client group until completion. Sometimes a case is transferred to another case manager for a variety of reasons. The case includes all the members of a client group (which is commonly thought of as a family or household).

The case manager conducts an assessment of all the client's needs and must be able to input that information into the software application. The information includes details of every client member of the client group, all the services that are required, all the services that are provided, date the services are provided, who or what agency is providing the services, results of the services provided, name of the case manager, notes about the case, documents that pertain to the case, messages and referrals that are sent to other agencies, an assessment of the group, and when the case is opened and closed.

The CHI manager provided more detail about the information they need to track. It includes: name, date, new client or past client, zip code, township, family type (single, single with dependent children, married, married with dependent children, widow, divorced), homeless, number of children, mental illness, disability, domestic violence, felony, incarcerated, and race. In addition, the two intake forms that were provided specify even more detailed information that CHI uses.

The CHI manager listed some other information needs she had. She would like the capability to run reports at many different levels. She needs detailed reports on cases, clients and client groups. This information will contain private client information and is only available to authorized users of CHI. She needs reports that aggregate information based on clients, services provided, referrals, zip code, township, homeless, children, and marital status. The aggregate reports must protect the privacy information of clients since it will be available to the public and used to obtain grants and inform donors.

The CHI manager stated that the application must allow case managers to type in free flow text as part of a case. Case managers routinely receive a lot of information that is pertinent to a case but not supported by a database structure. This information is often the "why"



information. For example, a client comes to CHI and asks for rent assistance. The client explains that their apartment was flooded, they didn't have flood insurance, the landlord wouldn't refund any of their deposit or rent, and they don't have enough money for deposit and the first month's rent. The case manager needs this information as part of the case for justification and that information should be available in the application.

The researcher interviewed a MA member next who provided a common intake form (in appendix D) used by many other MA members. This form needed client information such as name, address, assistance needed, income, expenses, and other agencies contacted. A copy of the form is in appendix D. The member mentioned that their organization worked similar to CHI. They opened a case when the client requested assistance. They might work with the client for a few hours or many days depending on their needs. Each church usually has a person or two designated to spend their benevolent funds to assist clients. They are equivalent to CHI's case manager.

The MA churches do want to collaborate with other agencies. They are more comfortable spending their funds on clients that are referred from a trusted agency. They wanted to share information with the other agencies to better serve the client and conserve resources. They wanted a reporting capability similar to that mentioned by the IFPJC and CHI but they needed to add how much was spent on a client. This would allow them to understand what needs their resources were meeting.

The researcher then analyzed all the information and sketched out a rough database and user interface design. The researcher then conducted a meeting with all three agency representatives (IFPJC, CHI, and MA). This meeting was to review the major premises of the

required system to make sure the basic analysis was accurate. The meeting helped clarify the most important aspects of the required system.

It became clear during the discussion that all the agencies create a case when a client or client group comes to their agency for assistance. The IFPJC does an intake when the client/client group comes in and then provides food and that normally ends the case. The IFPJC does make referrals to other agencies and sometimes adds notes or comments to the client's record. So, although the IFPJC does usually provide a service, they often need to add other elements that would be considered a case. CHI definitely opens a case with each client visit and routinely provides multiple services. The MA also agreed they open a case for each client even if they only provide one service during that visit. The three agencies agreed that the case was the central piece of this application.

The group discussed the major elements of a case. It was agreed that client information such as last name, first name, middle initial, suffix, date of birth, gender, race, income and expenses, health data, and ethnicity needed to be captured for each client in the group. Family or group information such as address, phone numbers, email, homeless status, family type, and food stamps was collected. Address includes items such as street, city, zip, state, county, township, and school district. Phone includes home, work, cell, and fax. Email includes personal or work email addresses.

The groups decided that an assessment of the client group was a key driving factor. The assessment notes all the needs the group has such as housing, rent assistance, food, medical care, dental care, counseling, job assistance, etc. A more detailed list is in appendix B. The case manager would review this needs list with the family to determine which needs apply to them. The group's needs are what drive the services the case manager will strive to provide. This

brought up the discussion that the application needs a list of agencies and what services they could provide. This would allow the case manager to select a need and locate an agency that could provide the service.

The group spent some time discussing the needs listed in Appendix B in detail. All three agencies expressed a strong desire for the user interface to make it simple for a user to find the agencies that could meet each need. The group described their vision of a user friendly interface. The case manager checks a need such as “shelter”, a service line is automatically generated to solve that need, and then there is a drop down menu in the service line that is populated with those agencies that are capable of fulfilling the need. The case manager then selects an agency and writes any necessary notes. The case manager can then click a button to send the agency a message/referral asking them if they will assist the client. The case manager can do this for every need they selected for the client.

The group stated that the case manager needed to add notes or comments on the case into the application. A case manager would expect to have several cases open simultaneously and some cases could last for months. The notes would assist the case manager to remember conversations with the client, conversations with other agencies, and items still to be done.

The group wanted the application to keep track of all the services that were provided to the client throughout the life of the case. They wanted to see if the client was a previous client or new client. They wanted to upload some documents for the case so that everything they need would be in the case application. The group discussed keeping track of all their appointments. They wanted the application to allow them to make and schedule appointments. They wanted the calendar to send or synchronize with the tool they each used.

## **Chapter 5 – Recommendations and Conclusions**

The research showed that the agencies needed an IT platform that would enable them to collaborate on client needs by creating cases and sharing information among the agencies that would meet those client needs. They also needed to aggregate information from all the agencies to help them understand what the needs are in their community, what needs are being met, and which needs are not being met. Those needs not being met would show the service gaps for the community. Research shows that an IT platform could also help them collaborate on larger issues not tied to one family. The platform needed to be accessible simultaneously by multiple users who are geographically disbursed. No one agency could accomplish this alone, so a collaborative environment is necessary.

### **Web Hosting Site**

IFPJC already had a suitable web hosting site for the collaboration effort costing less than \$10 dollars a month, providing 10 GB of database storage, unlimited web storage, unlimited email accounts, PHP programming language support, 15 dedicated IP addresses, unlimited data transfer, FTP accounts, free shared SSL, and 24/7 support. This web hosting platform allows the coalition to create custom applications and would allow multiple users to connect to the platform simultaneously from disbursed geographic locations.

### **Database**

Research shows that open source database management systems such as MySQL or PostgreSQL can be used as the main database tool for the coalition's shared database and custom applications. All agencies can share this database so that information is easily consolidated and aggregated for coalition wide reports and analysis. Research showed that the database must be

structured so that users can only modify or create information for their agency. The database must also allow one agency to withdraw from the collaboration effort without adversely affecting the remaining members. The agency that withdraws must be able to extract their data so they can continue their operation outside of the shared database environment. Research shows that MySQL provides all the necessary functionality needed by the coalition including stored procedures, triggers, ACID compliance for transactions, and a choice of storage engines. The InnoDB storage engine can be used for all coalition input screens due to its ACID compliance and referential integrity.

Research showed that MySQL accomplishes user account management by allowing multiple accounts to be created. These accounts can then be granted the specific privileges such as insert, create, update, and read. Accounts can be used in applications and individual users can be allowed to use those accounts based on their authorizations. Users can log in with a username, password, and agency name via an application web page. The application web page can query the database which contains tables controlling usernames, passwords, and access to specific MySQL accounts. This method limits access to data to those authorized by the agency administrator.

Based on research a possible database structure to support the coalition is listed in appendix A. The key concept driving this database structure is the idea of “case management”. A case includes clients, family information, needs, services, notes, uploaded documents, and an internal message system. Clients will visit an agency to request services and the agency will open a case. The case may stay open for only a few moments or many months. Multiple agencies can provide services to one client group but there will be only one case manager. The

system allows the transfer of cases from one manager to another as well as from one agency to another.

Research shows that the database must be designed to maintain an audit trail so the agencies know who created, modified, or deleted data in the database. The audit table keeps the table name, primary key field name, primary key id, field name, old information, new information, userID, and a timestamp. Multiple changes in the same record would require multiple entries in the audit table. The audit table stores the original input for a record, any changes to an already existing record, and any deletes. The audit tables are populated by creating triggers on tables.

The application has a backup and recovery plan. The hosting company provides local database backups twice per week and remote backups every two days. This results in database backups between one and four days old. Although this prevents the coalition from losing more than a few days worth of data, it is insufficient to meet coalition needs. The coalition needs a point in time recovery so that no more than 30 minutes of input are lost. The application provides a custom back up plan that transmits changed data every 30 minutes to an offsite data storage location (external hard drive at one agency location). The application also provides the user an option to create a back up at any time. There is a table called “backups” that keeps track of the last backup taken by timestamp. All records that have a timestamp later than the last backup are included in the new backup.

### **Programming Language**

Research shows that the PHP programming language is sufficient to be used on the server side and provide the middle tier business logic. PHP can be mixed with HTML and JavaScript to produce the graphical user interfaces (GUI) on the client side seen by the users in their web

browsers. JavaScript can be used for client side validation when necessary and to capture the user actions such as button clicks. HTML and CSS can be used for formatting the web pages.

The users must allow JavaScript to run in their browsers.

Research shows that the LAMP platform allows custom applications to be built. The main effort of custom applications would be based on the case management needed by the coalition. Research shows that the case management elements include client information, family/group contact information, family needs, services provided, case notes, case messages, referrals, documents, and related cases. Each agency has different needs and would require a customized user interface but much of the functionality can be programmed once and used many times. An example is client information. CHI needs to keep race and ethnicity information but IFPJC does not. The database can have fields for this information, CHI's screen can allow race and ethnicity input but IFPJC's screen does not allow race and ethnicity input. The database fields just do not have to be mandatory.

### **Security**

Research shows that client information must be limited to authorized users. HTTPS (TLS or SSL) should be used for all the client user interfaces to protect client personal information. The database server and web server must reside in a protected computer center administered by the website hosting company. Authorized users must have usernames and passwords to login to the application. Usernames and passwords should be maintained in the database. Users must authenticate successfully to access protected web pages. The login module should work for every agency so that only one security module needs to be created. Usernames and passwords should be associated with specific MySQL accounts so users are limited to specific functions in the database. Each agency should have an administrator that is responsible

for authorizing access to their protected pages and periodically checking that those users are still authorized. All records in the database must maintain the user ID of the last user to change data in the record. The audit table should maintain a history of the changes to each record.

### **Collaboration Tools**

Research shows that the agencies need two kinds of collaboration. One is specific to a case and is part of the case management user interface discussed previously. Collaboration by multiple agencies on a case is accomplished by using the internal notes, documents, messages, calendar, and client information built into the case management user interface. The second is an opportunity for the agencies to collaborate on more basic or general issues such as funding, statistics, community problems, etc. Research shows that the web hosting vendor provides free tools such as WordPress or b2evolution for creating blogs, phbBB or XMB Forum for creating Internet forums, WebCalendar for creating multi-user or single user calendars, and DokuWiki for creating Wiki's. These tools can be set up to allow all the agencies access.

### **Change Management**

Based on research the coalition needs the the application to have a trouble ticket system to allow users to easily report problems with the platform. There should be a menu item throughout the application that allows users to describe a problem they are having with the application. This menu function should insert the problem in a database table as well as emailing the coalitions technical support. The function should also record the user's username as well as a timestamp in addition to the user's comments.

The coalition should have a change management process in place to authorize any software changes to the application. Anyone can make a recommendation for a software change. The recommended changes are submitted via the trouble ticket system listed above. The



application allows the agency administrators and software developers to look at the list of trouble tickets. The agency administrators and software developers agree on what software changes will be implemented in the next release. The developers then make the changes in a dedicated development environment. The changes are then moved to a dedicated testing environment where the agency administrators test the changes. The developers correct any errors. When the agency administrators are satisfied with the changes, they give approval for the developers to put the software changes in the production environment.

### **Business Intelligence Tools**

Research shows a strong need for a business intelligence function for both the public and the private side. The public side should provide data that is aggregated to a level that protects client personal information. The public BI web pages should allow visitors to request data by city, county, school district, zip code, and township for a specific date range and cross referenced to service provided, age, number of clients, gender, marital status, dependent children, food stamp recipients, and homeless status. The private side should allow authorized agency users to pull detailed reports that include client's personal information. It also should include all the information provided on the public side. Data should be available in multiple formats such as Microsoft Excel, text files, or PDF.

## **Chapter 6 – Areas for Further Research**

The IT environment that supports a collaboration effort for multiple social agencies will definitely grow in size over time. Research into data warehouses could provide a methodology that would assist the coalition in improving application performance over time.

## References

- Author Todorov, Dobromir (2007). Mechanics of User Identification and Authentication: Fundamentals of Identity Management. Auerbach Publication. Books24x7. Retrieved January 19, 2009 from:  
[http://common.books24x7.com.dml.regis.edu/book/id\\_26453/book.asp](http://common.books24x7.com.dml.regis.edu/book/id_26453/book.asp)
- Identity Theft, (2009). Fighting Back Against Identity Theft. Federal Trade Commission. Retrieved Jan 22, 2009 from  
<http://www.ftc.gov/bcp/edu/microsites/idtheft/consumers/about-identity-theft.html>
- ITRC (2009). Security Breaches. Identity theft Resource Center. Retrieved Jan 22, 2009 from  
[http://www.idtheftcenter.org/artman2/publish/lib\\_survey/ITRC\\_2008\\_Breach\\_List.shtml](http://www.idtheftcenter.org/artman2/publish/lib_survey/ITRC_2008_Breach_List.shtml)
- IX Web hosting (2009). Hosting plans. Retrieved Jan 24, 2009 from  
<http://www2.ixwebhosting.com/index.php/v2/pages.hostingPlans>
- Db Net Solution (2009). Benefits of Web Based Apps. Retrieved Jan 24, 2009 from  
<http://www.dbnetsolutions.co.uk/Articles/BenefitsOfWebBasedApplications.aspx>
- Rosen, Kenneth & Host, Douglas & Klee, Rachel & Farber, James & Rosinski, Richard (2007). Unix: The Complete Reference, Second Edition. McGraw-Hill Companies, New York, New York.
- JK Design (2009). Why we Recommend the LAMP Solution Stack. Retrieved October 24, 2010 from

<http://www.jkdesign.com/jkdesign-lamp-solution-stack.pdf>

Linux.org (July, 2007). What is Linux. Retrieved April 16, 2008 from

<http://www.linux.org/info/index.html>

LinuxJournal (n.d.). What is Linux? Retrieved April 18, 2008 from

<http://www.linuxjournal.com/xstatic/community/whatislinux>

ServerWatch (April, 2008). March 2008 Security Space survey. Retrieved April 18, 2008 from

<http://www.serverwatch.com/stats/article.php/3739201>

Apache.org-modules (n.d.). Apache modules. Retrieved April 18, 2008 from

<http://httpd.apache.org/docs/1.3/mod/>

MySQL.com (February, 2008). MySQL and Sun. Retrieved April 16, 2008 from

<http://www.mysql.com/news-and-events/sun/>

FSF (n.d.). Free Software Foundation: Frequently Asked Questions about the GNU Licenses.

Retrieved April 18, 2008 from

<http://www.fsf.org/licensing/licenses/gpl-faq.html>

MySQL, 2010. MySQL. Retrieved October 24, 2010 from

<http://mysql.com/>

TIOBE (January, 2009). TIOBE Programming Community Index for January 2009. Retrieved

January 24, 2009 from

<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

cz.php.net (July, 2007). Usage Stats for April 2007. Retrieved April 16, 2008 from

<http://cz.php.net/usage.php>

SearchEnterpriseLinux (n.d.). PHP. Retrieved April 16, 2008 from

[http://searchenterpriselinix.techtarget.com/sDefinition/0,,sid39\\_gci334246,00.html](http://searchenterpriselinix.techtarget.com/sDefinition/0,,sid39_gci334246,00.html)

IXWebhosting (April, 2008). Expert Plan. Retrieved April 18, 2008 from

<http://www.ixwebhosting.com/index.php/v2/pages.planExpertplan>

ServerWatch-LAMP (December, 2005). Understanding LAMP. Retrieved April 19, 2008 from

[http://www.serverwatch.com/tutorials/article.php/10825\\_3567741\\_3](http://www.serverwatch.com/tutorials/article.php/10825_3567741_3)

SANS (2009). CWE/SANS Top 25 Most Dangerous Programming Errors. Retrieved Jan 31,

2009 from <http://www.sans.org/top25errors/>

Lerdorf, Rasmus & Tatroe, Kevin & MacIntyre, Peter (2006). Programming PHP, Second

Edition. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

CAPEC (2009). Individual CAPEC Dictionary Definition: SQL Injection. Retrieved February

7, 2009 from: <http://capec.mitre.org/data/definitions/66.html>

OWASP-Guide to SQL Injection (2009). Retrieved February 7, 2009 from:

[http://www.owasp.org/index.php/Guide\\_to\\_SQL\\_Injection](http://www.owasp.org/index.php/Guide_to_SQL_Injection)

OWASP—Cross Site Scripting (2009). Retrieved February 8, 2009 from:

[http://www.owasp.org/index.php/Top\\_10\\_2007-A1](http://www.owasp.org/index.php/Top_10_2007-A1)

Dr. Sweeney comments about HIPAA and identifying an individual

<http://privacy.cs.cmu.edu/dataprivacy/HIPAA/HIPAAcomments.html>

Federal Trade commission “Protecting Personal Information A guide for Business”. 600

Pennsylvania Avenue, NW Washington, DC 20580. Retrieved February 9, 2010 from:

<http://www.ftc.gov/bcp/edu/pubs/business/idtheft/bus69.pdf>

GSA pamphlet “protecting personally identifiable information”. Retrieved February 10, 2010

from:

[http://www.gsa.gov/gsa/cm\\_attachments/GSA\\_DOCUMENT/USA%20Services%20New  
sletter%2005-07\\_R2-v-x9-c\\_0Z5RDZ-i34K-pR.pdf](http://www.gsa.gov/gsa/cm_attachments/GSA_DOCUMENT/USA%20Services%20New%20letter%2005-07_R2-v-x9-c_0Z5RDZ-i34K-pR.pdf)

Straus, David (2002). *Unix: How to Make Collaboration Work: Powerful Ways to Build  
consensus, Solve Problems, and Make Decisions*. Berrett-Koehler Publishers,  
Books24x7.

Linden, Russell (2002). *Working Across Boundaries: Making Collaboration Work in  
government and nonprofit Organizations*. Jossey-Bass, Books24x7.

Brown, Katherine M., Huettner, Brenda, and James-Tanny, Char. *Managing Virtual Teams:  
Getting the Most from Wikis, Blogs, and other Collaborative Tools*. Wordware  
Publishing. © 2007. Books24x7. Retrieved February 3, 2010 from:  
[http://common.books24x7.com/book/id\\_18222/book.asp](http://common.books24x7.com/book/id_18222/book.asp)

Nemiro, Jill, Michael Beyerlein, Lori Bradley, and Susan Beyerlein (eds). "Chapter Thirteen - Design Concepts for Virtual Work Systems". *The Handbook of High Performance Virtual Teams: A Toolkit for Collaborating Across Boundaries*. Jossey-Bass. © 2008. Books24x7. Retrieved February 6, 2010 from:  
[http://common.books24x7.com/book/id\\_24270/book.asp](http://common.books24x7.com/book/id_24270/book.asp)

Zelenka, Truitt, Anne, and Sohn, Judi. "Chapter 4 - Rethink Your Relationship with Email". *Connect!: A Guide to a New Way of Working from GigaOM's Web Worker Daily*. John Wiley & Sons. © 2008. Books24x7. Retrieved February 8, 2010 from:  
[http://common.books24x7.com/book/id\\_23447/book.asp](http://common.books24x7.com/book/id_23447/book.asp)

Zheng, Pei, and Ni, Lionel M. "Chapter 1 - Introduction to Smart Phone and Mobile Computing". *Smart Phone and Next-Generation Mobile Computing*. Morgan Kaufmann Publishers. © 2006. Books24x7. Retrieved February 13, 2010 from:  
[http://common.books24x7.com/book/id\\_17930/book.asp](http://common.books24x7.com/book/id_17930/book.asp)

Godwin-Jones, Bob (2003). *Emerging Technologies Blogs and Wikis: Environments for On-line Collaboration*. Retrieved February 11, 2010 from: <http://llt.msu.edu/vol7num2/emerging/>

Phonescoop (2010). *Text Messaging*. Retrieved February 15, 2010 from  
<http://www.phonescoop.com/glossary/term.php?gid=387>

Press-feed (2010). *RSS Feeds – A tutorial*. Retrieved October 24, 2010 from  
[http://www.press-feed.com/howitworks/rss\\_tutorial.php](http://www.press-feed.com/howitworks/rss_tutorial.php)

Woods, Dan, Peter Thoeny, and Ward Cunningham. "Chapter 1 - Understanding Wikis—From Ward's Brain to Your Browser". Wikis for Dummies. John Wiley & Sons. © 2007.

Books24x7. Retrieved February 15, 2010 from:

[http://common.books24x7.com/book/id\\_18441/book.asp](http://common.books24x7.com/book/id_18441/book.asp)

Apple FAQs:For Podcast Fans. Retrieved February 15, 2010 from:

<http://www.apple.com/itunes/podcasts/fanfaq.html>

WebCalendar. Retrieved February 15, 2010 from: <http://www.k5n.us/webcalendar.php>

Project Management Institute. "Chapter 1 - Introduction". A Guide to the Project Management Body Of Knowledge (PMBOK® Guide), Fourth Edition. Project Management Institute.

© 2008. Books24x7. Retrieved February 16, 2010 from:

[http://common.books24x7.com/book/id\\_29854/book.asp](http://common.books24x7.com/book/id_29854/book.asp)

WorkZone. Tracking Projects. Retrieved February 16, 2010 from:

<http://www.workzone.com/product-tour/tracking-projects.jsp?loc=head>

Wikimatrix. Retrieved February 16, 2010 from: <http://www.wikimatrix.org/show/PhpWiki>

Gunasekaran, Angappa. "Chapter IV - Software Architectures and Requirements for a Web-

Based Survey System". Modeling and Analysis of Enterprise Information Systems. IGI

Global. © 2007. Books24x7. Retrieved February 16, 2010 from:

[http://common.books24x7.com/book/id\\_20765/book.asp](http://common.books24x7.com/book/id_20765/book.asp)

Howson, Cindi. "Chapter 3 - The Business Intelligence Front-End". Successful Business

Intelligence: Secrets to Making BI a Killer App. McGraw-Hill/Osborne. © 2008.



Books24x7. Retrieved February 28, 2010 from:

[http://common.books24x7.com/book/id\\_23651/book.asp](http://common.books24x7.com/book/id_23651/book.asp)

Ezinearticles. The Evolution of Audio Video Conferencing. Retrieved October 24, 2010 from:

<http://ezinearticles.com/?The-Evolution-of-Audio-Video-Conferencing&id=71307>

**Appendix A – DB structure**

Below is a list of tables in the database:

TABLE NAME	DESCRIPTION
Client	Client information such as names, birth date, race, ethnicity
Prev_name	Previous names of a client
Case	Case start and end date, case manager, homeless, family type, medical, violence, felony, foodstamps
Case_client	Junction table to link clients to a case
Case_note	Multiple notes for a case.
Case_service	Junction table between case and services provided to a case allowing multiple services for a case.
Case_doc	Multiple documents can be uploaded for each case.
Case_msg	Multiple messages sent and received about a case.
Services	Multiple services that are provided to a case.
Need	List of needs used to assess a client
Agency	List of agencies that provide services
Agency_need	Junction table for need and agency table. Shows all the needs that an agency can fulfill
User	All users that have access to the system
User_agency	Junction table linking user and agency so you can determine the agency that a user works for
Address	Street, city, zip, etc. Has an other_id field that links to multiple tables. This requires all primary keys in the entire database to be unique. Can link to case, user, and agency table.
Phone	Phone numbers for agency, user, and case. Multiple types such as home, work, mobile, fax
email	Email addresses for case, user, and agency tables.
Audit	All database changes are logged here
Backup	Last backup timestamp

Database Table detail Information

<b>CLIENT</b>
Client_id
Lname
Fname
Mi
Suffix
Gender
Dob
Ethnicity
Native
Asian
Black
Pacific
White
Otherrace
createdt

<b>CASE_CLIENT</b>
Case_id
Client_id
Startdt
Enddt
Maritalstatus

<b>CASE</b>
Case_id
Startdt
Enddt
Tag
User_id
Homeless
Familytype
Mentalillness
Disability
Domesticviolence
Felony
Jail
Foodstamps

<b>PREV_NAME</b>
Client_id
Lname
Fname
Mi
suffix

**Appendix B – List of Needs for Client Assessment**

A key part of the coalition case management is a list of needs used to assess the client’s problems:

Shelter	Electricity	Domestic Violence	Elder Care
Rent Assistance	Dentist	Baby Needs	Respite Care
Mortgage Assistance	Mental Health Care	Transportation	
Heat	Pregnancy	Gasoline	
Water/Sewer	Clothing	Vehicle Repairs	
Household Goods	School Supplies	Vehicle Payments	
Food	Telephone	Burial	
Medications	Health Care	Child Care	

**Appendix C – IRB Approval**



Academic Affairs  
Academic Grants

3333 Regis Boulevard, H-4  
Denver, Colorado 80221-1099

303-158-1206  
303-964-3647 FAX  
[www.regis.edu](http://www.regis.edu)

IRB – REGIS UNIVERSITY

August 19, 2010

Larry Noonan  
6330 N 200 E  
Whiteland, IN 46184

**RE: IRB #: 125-10**

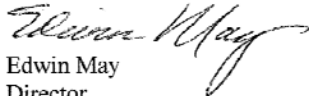
Dear Larry:

Your application to the Regis IRB for your project "Collaboration Methods for Social Service Agencies: A Case Study of Johnson County, Indiana," was approved as exempt on August 19, 2010.

The designation of "exempt," means no further IRB review of this project, as it is currently designed, is needed.

If changes are made in the research plan that significantly alter the involvement of human subjects from that which was approved in the named application, the new research plan must be resubmitted to the Regis IRB for approval.

Sincerely,

  
Edwin May  
Director

cc: Charles Thies