UNIVERSITY OF WISCONSIN-LA CROSSE

Graduate Studies

A WEB-BASED APPLICATION TO SUPPORT THE LA CROSSE WELLNESS PROJECT

A Manuscript Style Thesis Submitted in Partial Fulfillment of the Requirements for the

Degree of Master of Software Engineering

Nhan V Than

College of Science and Health
Computer Science

December 2006

A WEB-BASED APPLICATION TO SUPPORT THE LA CROSSE WELLNESS PROJECT

By Nhan V Than

We recommend acceptance of this manuscript in partial fulfillment of this candidate's requirements for the degree of Master of Software Engineering in Computer Science. The candidate has completed the oral examination requirement of the capstone project for the degree.

Dr. Thomas Gendreau Date
Examination Committee Chairperson

Dr. Kasi Periyasamy Date
Examination Committee Member

Dr. David Riley Date
Examination Committee Member

Dr. Vijendra Agarwal Date
Director, University Graduate Studies

ABSTRACT

THAN, NHAN, V., “A Web-based application to support the La Crosse Wellness Project”, Master of Software Engineering, December 2006, (Dr. Thomas Gendreau, Dr. Kasi Periyasamy).

The purpose of the La Crosse Wellness Project is to promote optimal wellness by focusing on health promotion as a process that is learned and ongoing. The La Crosse Wellness Project is comprised of two components: The La Crosse Wellness Inventory (LWI) for individual assessment purposes, and the Wellness Development Process (WDP) for the development of individualized plans of action. In this project, the LWI and WDP are developed as web-based applications which provide participants mobility, flexibility, convenience and data security. Moreover, security of personal information and extended database design are required to be investigated and implemented to provide a more secure transaction and meet the major requirements of this project.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my project advisors Dr. Thomas Gendreau and Dr. Kasi Periyasamy for their valuable guidance. I also would like to thank the project sponsor, Dr. Gary D. Gilmore, who initiated this project and provided the support for this project. I need to express my special thanks to Dr. David Riley for his guidance and also for his valuable explanations. I would also like to express my thanks to the Computer Science Department and the University of Wisconsin-La Crosse for providing the computing environment for my project. I'm also grateful for the advice and guidance from my professor at the University of Natural Sciences – Ho Chi Minh City, Vietnam, Tran Minh Triet, PhD candidate in Computer Science; and my friend, Mr. Luong Vi Minh. Finally, I wish to thank my parents, relatives and friends for their support and encouragement during the duration of this project.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

GLOSSARY

AES-256

Advanced Encryption Standard-256 is a cryptographic symmetric encryption and uses a 256-bit long key.

Apache Tomcat 5.5

A web container that functions as a web server supporting servlets and JSPs.

API

Application Program Interface. The interface through which an application accesses the operating system and other services.

CSS

Cascading Style Sheets. A configuration file for setting common web page properties.

DBMS

Database Management System, software that controls the organization, storage, retrieval, security and integrity of data in a database. It accepts requests from an application and instructs the operating system to transfer the appropriate data. Some of the major DBMS vendors are Oracle, IBM, Microsoft and Sybase.

FrontPage 2003

A Microsoft GUI program for designing web pages and web sites.

GUI

Graphical User Interface, the presentation layer for user to interact with an application.

HIPAA

Health Insurance Portability and Accountability Act (1996), a US Federal regulation that gives patients greater access to their own medical records and more control over how their personally identifiable health information is used. The regulation also addresses the obligations of healthcare providers and health plans to protect health information.

HTML

HyperText Markup Language. A markup language designed for creating web pages and other information to view in a web browser.

IEEE

Institute of Electrical and Electronics Engineers. An international organization whose constitution describes its purpose as “scientific and educational, directed toward the advancement of the theory and practice of several engineering fields including computer science.”

JSP

Java Server Pages is a Java technology that is used to develop dynamic web pages.

MVC

Model-View-Controller is a framework for JSP and Servlet to make a 3 tier web application.

SHA-256

Secure Hash Algorithm 256 is a cryptographic one-way hash function and uses a 256-bit hash value.

Servlet

A Java program adding dynamic HTML or XML content to a web server application

SMTP

Simple Mail Transfer Protocol is the standard way to send emails across a network.

SQL

Structured Query Language is a primary language used to create, modify and query relational databases.

SRS

Software Requirements Specification. A document format supplied by the IEEE for specifying the requirements of a software system.

SSL

Secure Sockets Layer is a cryptographic protocol that provides secure communications on a network over a socket connection.

Stored Procedure

A program that is physically stored within a database that can be run to perform various queries or other database operations.

Web containers

A Web container is a computer program that runs Web applications. Web containers are also sometimes called Web engines.

WAN

Wide areas network. A computer network that usually covers a wide geographical area and involves many computers. A good example is the Internet.

WUI

Web User Interface. A kind of GUI that runs in a web browser.

XML

Extensible Markup Language. A W3C recommendation for creating special-purpose markup language and is widely used to exchange data, store configuration data, along with many other uses.

Verification

A process that confirms a development process or activity or task to be correct.

Validation

A process that confirms that the product (or partial product) meets the expectations.

Testing

A type of validation applied to source code.

LWI

La Crosse Wellness Inventory

WDP

Wellness Development Process

LWP

La Crosse Wellness Project

1. Background Information

Health promotion involves a life-long process in which health-related decisions are made to maximize one's health potential. Health promotion is a way of life which can be chosen by each one of us. Self-responsibility is a key ingredient in this chosen way of life. The purpose of the La Crosse Wellness Project is to promote optimal wellness by focusing on health promotion as a process that is learned and ongoing. One definition of wellness is "the daily striving for the goal of becoming healthier through ongoing assessment, intervention, and reinforcement" [5]. This means that a positive, healthy lifestyle can best be accomplished when a framework is offered. Within this framework are the following components: assessment, intervention, and reinforcement. Assessment refers to reviewing one's past and present health-related influences and actions. Assessment provides an awareness of one's health state and helps point the way toward changes. Intervention involves learning and establishing a process that enable you to take action for healthy changes. Reinforcement rewards a person for those changes.

La Crosse Wellness Inventory (LWI): The La Crosse Wellness Inventory is designed to serve as an educational tool, as well as an assessment process for individuals to examine their current level of wellness. The inventory serves as an educational tool by creating an awareness of activities for developing a healthy lifestyle. In responding to the inventory statements, a participant may make an assessment regarding his/her past and present lifestyle. Furthermore, the inventory results will acquaint participant with selected community resources.

The LWI is a nationally revalidated instrument [14] that assesses one's level of wellness in nine health-related areas, resulting in wellness scores represented as percentages for each area. The nine areas are: rest and relaxation, emotion and mental health, sexuality, personal health habits, fitness, nutrition, drugs, safety, environmental sensitivity. There are a total of 178 items in this inventory. For each item a participant answers, the score will be accumulated based on a weighted scale so that at the end, the result will be calculated as a percentage for each area.

Wellness Development Process (WDP): The WDP, as the next step of LWI, addresses the follow-up processes of intervention and reinforcement through the development of a personalized wellness plan and the identification of personalized reinforcements. It encourages individuals to become fully engaged in the wellness enhancement process. Enhancement in this project means reducing or eliminating unhealthy actions in one's life, and/or maintaining or improving healthy actions in his/her life [6]. Rather than a computer printout specifying the steps an individual needs to take, participants go through a guided session in which they make choices in the development of their next steps for health enhancement. Participants are encouraged to take small, reasonable next steps; steps that can be easily achieved, thus generating a reinforcing sense of success.

In WDP there are several phases: Establishing a wellness area for enhancement, Identifying wellness results, establishing wellness activities, determining small action steps, and then developing at least a one-week personal plan of action that includes a starting and ending date, the action steps to be taken, and sources of reinforcement. In establishing a wellness area for enhancement, each participant needs to review his/her LWI scores and then transfer the scores to WDP. Then the participant needs to consider the top three areas that need to be enhanced based upon one's desire and ability. When identifying wellness results, the participant must establish lifestyle changes for one selected wellness area. In establishing wellness activities, the participant is asked to list healthy activities (he/she is presently doing) that need to be continued and perhaps even improved. Then he/she is asked to list unhealthy activities (he/she is presently doing) that can be reduced or eliminated. Activities are then further refined into action steps that the participant will use to move toward the result he/she has chosen.

Each participant fills out an action plan after going through the phases of WDP. Follow-up sessions are then held in order to determine the degree to which participants are experiencing levels of success. This also provides an opportunity for participants to receive and review additional wellness resources.

The current manual system has the following disadvantages:

Paper intensive: There are about 16 pages for 178 questions in LWI. Thus, for each participant, 16 pages of paper copies are required. It multiplies as the number of participants increases.

Historical record: It is often difficult to find a specific record of a particular participant. Sometimes the paper records are lost or misfiled.

Communication intensive: Currently, there is a need for a good deal of guidance and facilitation by LWP advisors. If the guidance systems can be built into the LWI and WDP, the time investment by all who are involved will be substantially reduced.

Besides the paper versions done by Dr. Gary Gilmore, there were two electronic versions of LWI and WDP. The first electronic version of WDP was written by Donald C. Xiong (1998), and it was written in Java script and HTML. Meanwhile, the first electronic version of LWI was written in Java applet (2000). Both versions didn't have either web server or database to store participant's responses for future research. In order to take part in the project, a participant need to copy the html files, load them on his/her computer. At the end of LWI or WDP, the participant must print his/her responses for later reference. Although these two versions of both LWI and WDP met the requirements of the project at that time, they weren't able to adapt to the new requirements.

2. A Brief Introduction to Software Life Cycle Models

The term “software life cycle” refers to a coherent sequence of well-defined software development processes from conception to eventual retirement over the course of a software product’s lifespan. The various stages and activities performed in each stage may vary depending on the life cycle model chosen. There are a wide variety of software life cycle models reported in the literature [15]. There is no single life cycle that is best suited for every project. Different software life cycles may complement each other on the same project during different stages of the development of a product. Successful use of a particular life cycle model does not ensure success or a best fit for a new project. Choosing the most appropriate model for a particular software project is challenging. Many factors such as team size, project size, available resources, deadlines or milestones, team skills and experience, business policies, and application domain may influence the choice of a life cycle model.

In this project, the developer applied both rapid prototyping model and incremental model. Initially, rapid prototyping model helped developer not only to discover the requirements of the project but also to meet the deadline stated by the sponsor. The sponsor wanted to have the electronic versions of both LWI and WDP available as soon as possible to support a graduate student in Health Promotion in Community Health Education in completing her thesis. After discovering all possible requirements, the developer chose to integrate the LWI and WDP electronic versions to preserve the nature of the project. At that time the sponsor decided to add new functionalities to the project. In order to adapt with the sponsor’s changes, the incremental model is a better choice than the waterfall model, because the developer can reuse some existing modules from the prototypes in rapid prototyping model. This also saved development time, since the developer just needed to build the new modules and integrate with existing ones.

Rapid prototyping is used as a means to better facilitate the requirement gathering process from a customer. During the requirement phase, a cursory prototype can be constructed and given to a potential user or customer in order to (1) determine the feasibility of a requirement, (2) validate that a particular functionality is really necessary, (3) uncover missing requirements, or (4) determine the viability of a user interface. Armed with the experience of having used the prototype, the team completes the SRS with increased assurance that the right system is being specified [6]. The prototype is presented to the customer as quickly as possible. Since the prototype is going to be thrown away, quality standards are lessened, documentation may not exist, performance may not be a large concern, and well understood functionalities may be completely left out of the prototype. The prototype is done this way because its advantages exist only if results from using it are available in a timely fashion [3]. When used correctly, *rapid prototyping* has been shown to help create software that more closely meets user's needs and expectations. However, it does have some potential disadvantages when misused. One incorrect use is to let the prototyping activity go on for too long of a time period. The goal is to develop the prototype as quickly as possible, capture requirement information by interacting with customers frequently, and then throw it away. Another danger in using *rapid prototyping* is that some non-functional requirements, such as performance, may not be represented or discovered by the prototype. Also, the prototype might not be running in the same environment as the final product. Once a prototype is completed and the customer has given feedback, the requirements for the product must further be developed using a more formal SRS document. The SRS is a valuable communication tool that provides input into other phases, such as design and testing.

Incremental model, another life cycle model, is also investigated. In this model, software is constructed step by step, in the same way that a building is constructed. While a software product is in the process of being developed, each step consists of additional pieces that are added to what has gone before. The construction of the complete product proceeds incrementally in this way until the product is finished. The product is designed, implemented, integrated, and tested as a series of incremental builds, where a build consists of code pieces from various modules that will interact together to provide a specific functional capability. At each stage of the incremental model, a new build is coded, and then integrated into the structure which is tested as a whole. The process stops when the product satisfies all its requirements.

3. Development of The La Crosse Wellness Project

LWP consists of two main components: LWI and WDP. Although the LWI and WDP are different in functionality, they are built for the same purpose: storing and retrieving health information in a secure manner. In addition, WDP depends on LWI. In other words, the user must first have LWI in order to work with WDP. Thus, the LWI was first developed and then served as a reference structure for the development of WDP. Rapid prototyping was chosen as the life cycle of LWP. The first phase of development involved collecting software requirements from the sponsor and also from investigating the current WDP product. This was done to produce the software requirement specification for the LWP. In the second phase, requirements were then analyzed and a detailed design document was written. The third phase was about developing prototypes for both the LWI and WDP in order to communicate with the sponsor and more quickly discover the additional requirements as well as changes to existing requirements. This phase was repeated until all requirements were met. In the next phase, both LWI and WDP were integrated. At the final phase, due to new requirements come from the sponsor, it dealt with adding new functionalities to the integrated version.

3.1. Collecting software requirements

The sponsor, Dr. Gary Gilmore, and his graduate assistants (project clients) served as the application domain resource for answering various questions during this phase. After several meetings and interviews with the project clients about the new LWI product and the WDP, the project's functionalities were identified and grouped into two categories: (i) security- related functionalities and (ii) non-security related functionalities. Security-related functionalities are those requiring encryption and decryption mechanisms, while non-security related functionalities are those relating to the logic of the software:

calculating results for LWI, adding, saving, and retrieving information for both the LWI and WDP. In addition to those functionalities, a careful study of the technical expertise of the various types of users revealed that the user-interface for the system must be easy-to-use and user friendly. At this point in the project, the developer worked with the sponsor to determine if a web application was the best fit for the users of the new system. A more detailed examination of the definition of web-based applications versus traditional "rich client" applications was done to help in the decision process. Following are some of the important characteristics of a web-based application as reported from the developer's experience:

Advantages of using web-based user interface

- Deployment – eliminating client administration is possible to achieve if all processing is handled by the server.
- It is portable for client, since only HTML is returned to client browsers typically
- A web-based application can be run from anywhere over Internet, Intranet, or local machine – This ensures availability of the system all the time.
- Ease of use even for the least technical users who understand how to use a browser. This would be the case for a majority of users of LWI and WDP.
- Support and compatibility - usability issues are less important, especially if there are a large number of heterogeneous clients.
- It is more convenient for remote administration or application hosting for the user

Disadvantages of using web-based user interface

- There are cross-browser issues since not all browsers render everything in the same way, especially older browser versions
- There are no real standards that currently exist for web-based applications.
- There is no drag and drop without additional programming effort, intelligence, auto-completion, context-sensitive help (in general)
- It is more difficult for a web-based application that needs lots of editing and ease-of-use features

- Responsiveness from servers can be an issue depending on bandwidth and graphics, and other factors
- It is inherently stateless
- Disconnected operations are a problem if the network is unavailable, for example

According to the sponsor’s requirements, the developer agreed to develop this project as a web-based application because of the following reasons:

- The rapid development of internet requires to port stand-alone application to web-based applications.
- A user does not need to install the whole system on his/her machine.

The following additional requirements were added to LWP:

- Security – Authentication. All users need to login with a password. They should be able to logout and to change passwords.
- Security – Authorization. The system administrator must be able to set the application policies, such as: time constraint for retaking LWI, or minimum number of records for producing aggregate report. An administrator should also able to see aggregate reports. Administrators are not allowed to see any participant’s information directly from the application or database. A participant can only view his/her own LWI or WDP record.
- Security – Confidentiality. The system supports storing encrypted passwords and the ability to encrypt the data stored in database and data during network transfer.
- Security – In general, the system must defend against at least some common forms of web application attacks such as SQL injections and cross-site scripting attacks.
- The system needs to work with a DBMS (Database Management System) such as: Oracle, MySQL, or MS Access.
- Scalability **-** The system must support several hundreds of concurrent users.
- Extensibility – The system must facilitate adding new modules in the future without major re-work.

Besides collecting requirements from the clients, the developer needed to spend more time using the current WDP and also looking through its source code in order to (i) understand the WDP in detail from a user perspective, including the application domain, and (ii) to understand it from an architectural and development perspective.

3.2. Rapid Prototyping for LWP

First Prototype of LWI

At the beginning of this project, the developer and project sponsor had agreed that prototyping a subset of functionality would be of great value to both the sponsor and the developer. It helped in refining requirements, and evaluating the feasibility of the product. The first prototype of the LWI was developed in the Spring of 2005. This prototype was used to support the thesis of a Master of Public Health candidate [11]. The requirement of this prototype was set up as a client-server LWI so that participants were able to take the LWI survey, and visualize their health scores as percentages. The participants' health scores were then used in WDP, and also used to produce an aggregate general report, as well as aggregate reports based on any specific characteristic (gender, age, education). First, a literature survey was performed to examine various technologies that could be used to create a web-based application. Technologies including JSP/Servlet, and ASP.NET were examined. Each technology is a server-side scripting language that uses code embedded in, or logically separated for the application code and HTML for the presentation layer. Each of the languages is capable of creating a cross platform, cross browser web-based application as they return naïve HTML code the clients after all the server-side code has been processed and replaced by the engines that process the server script. The developer found ASP.NET was not suitable for the project although it was known as a popular technology for any web application on market. The reason for this was that ASP.NET was not a free technology, and there were limited free tools supporting ASP.NET. JSP and Servlet, on the other hand, is a free technology, and there are many free tools supporting JSP such as: NetBeans, Eclipse and Tomcat Apache. Java is used mostly in both JSP and Servlet. In addition, this technology also supports MVC (Model-View-Controller) Framework which is highly object-oriented, so that it was chosen as the main technology in both components LWI and WDP.

For this prototype, the developer used Java script for calculation and JSP/Servlet for storing information to and/or retrieving information from database. Because the prototype was used to support a graduate student in Public Health, the developer was asked to finish this prototype in a one and one-half month time period to meet the thesis's timeline, and there was not enough time to determine a full security solution. In this case, a simple authentication was incorporated. Microsoft Access was chosen to develop the database for this simple prototype, so there were no security features on the database side either. In this case, in order to protect participant's information, a participant was asked to choose a username different from his/her real name. Inside the database, the application does not store the participant's real name. This guarantees that just by looking at the database, no one could guess the owner of a specific record. After a few pilot tests in Onalaska, and the La Crosse County, with positive feedback and recommendations offered, the sponsor asked the developer to start the first WDP prototype.

First prototype for WDP

Since a working version of WDP was already in use, the developer began by investigating this application. This was done by the developer's using the current WDP and studying its source code.

The previous WDP was developed as a web-based, client-side-only application. It was written in HTML and Java script, with no specific security features to support authentication, authorization, and validation. The application was required that all HTML and Java script code be downloaded to the user's machine, and the program was run locally. In addition, there was no database associated with WDP so that the user could not save nor retrieve his/her information in the future.

The developer had to build a new WDP in order to address the new requirements from the first electronic version, including designing database and authentication. Once again, JSP and Servlet were the technology used in this prototype development. The first prototype was also used to support the Public Health graduate student's thesis. The lack of time did not allow the developer to construct any security solution, and thus, only a simple authentication was implemented. Similar to that of LWI, the first prototype of WDP used Microsoft Access 2003 as a primary database platform to process simple queries, so that there was no protection for the database at this time. This prototype was tested by volunteers at the university, at the La Crosse County and at the School District of Onalaska.

Second prototype for LWI

In this prototype, the developer was required to finalize a security solution for the LWI. Many solutions have been thought of to meet the privacy and security requirements, and also to match with those of HIPAA standards [2]. Solutions including protecting the data on database, and the data traversing over the internet were examined.

The first prototype contained vulnerabilities inherent in Java script. Java script is easily revealed with the use of "view source" functionality provided by every browser. This gives a user the ability to see the Java script code embedded in the page. To mitigate this security threat, the developer decided to use a JSP/Servlet for processing data, leaving Java script to controlling the web user interface for client-side validation.

The database used in the first prototype also represented vulnerability. Although using Microsoft Access as a primary database platform is easy to set up, it contains many limitations in extending and storing encrypted data, along with restrictions to scalability, stability and reliability. As a result, Oracle was chosen to replace MS Access. Even though Oracle itself was much more expensive, and harder to set up in comparison to other free platforms, it was already available and properly configured. Additionally, the database no longer resided on the same machine with the application server, which enhances performance. Other reasons for selecting Oracle as the database include the following: support for encrypting data and BLOB (binary large object) type to store huge encrypted data. Multiple formulations were used on the Oracle-encrypted database in order to determine how to best apply this technology to the project. Finally, the developer decided not to use the encrypting data mechanism Oracle provided based on information privacy and confidentiality. If developer used the Oracle encrypting mechanism, this would give the database administrator, or anyone who could access the database, the authority to decrypt all information from database. In addition, Oracle can only encrypt information with one key for information in the same table, which does not absolutely ensure confidentiality of the information. Establishing a virtual private database, another solution from Oracle, was considered. Once again, it was determined not to be the best choice, since data was not encrypted in the database so that it would be "private" for application users only, and not for all of those who gained access to the database. The best solution, after all, was not using Oracle security features, instead using the developer's own security mechanism. With this approach, the application (not the database management system) takes care of the encryption/decryption operation, while the database management system manages storage of the already-encrypted data.

Finding the best security solution was the most challenging problem. It required thorough understanding of information security, not only because there were so many choices available, but also because the developer had not experienced this kind of problem before. Initially, asymmetric encryption was considered. Although it met the privacy and confidentiality of information, it didn't satisfy the performance requirement. In addition, storing public and private keys, as well as key exchange, contributed to the complexity of this approach. By comparison, symmetric encryption is relatively easy to implement, maintain and satisfies the performance goals of the project. In order to satisfy the major requirements: information privacy and confidentiality, the participant's password is used as the key for symmetric encryption. The password is chosen by participant at the registration stage, and it is protected with SHA-256 (Secure Hash Algorithm), a one way encryption, in the database. Breaking the hashed value to retrieve the original is extremely difficult, and requires a great deal of time and effort, along with a very powerful server. In addition to the protection of password from the database side, the participant's personal information is protected with AES-256 (Advanced Encryption

Standard), a highly secure encryption algorithm. With an authorized given username and password, the participant can only work with his/her own record

From the application aspect, in order to protect not only LWI, but also the whole LWP process, there must be a strong authentication for the users of the project. Additionally, there is a need to protect data traversed in the internet. SSL (Secure Socket Layer) was chosen to prevent data interception, eaves dropping, and other man-in-the middle attacks.

Second prototype for WDP

The requirements for the second prototype were also information privacy and security. This prototype was started after the second prototype for LWI was achieved. It didn't take a long time to develop this WDP, since it inherited a lot from the second prototype of LWI and the first prototype of WDP, especially security solution: encryption algorithms. Additionally, encrypted data in database was actually a well organized XML string.

3.3. Incremental Prototyping for LWP

The second prototypes of WDP and LWI from rapid prototyping were the input for incremental prototyping. The developer needed to integrate both the WDP and LWI prototypes. LWI and WDP were unit-tested, integrated-tested and system-tested independently before integration. The test cases were reapplied after integration. Following the integration stage, the developer was built five new modules: an authentication module, an authorization module, a registration module, a confirmation email module and an administration module. At the completion of each module, developer integrated the module with the prototype and retested. This process was repeated until all new modules were integrated and tested.

4. The LWP Application

This section explains the completed high-level architectural design of LWP application, detailed design architecture, database design, security design, and web-based user interface along with some detailed usability issues that were factors in its construction.

4.1. High Level Architectural Design

The high-level architectural design of the LWP application is presented in Figure 4.1. Users interact with LWP through JSP/Servlet web pages with a browser from a client machine for all functionalities of the system. These functionalities include navigation, requesting information, submitting data, and performing various actions. The web pages reside on a web server running the Tomcat 5.5. The web pages contain the code for various LWP functionalities, and many of them interact with an Oracle database.

The LWP application is designed as a three-tiered application. In a three-tiered system, the client, the database, and the web server are on three separate machines. With this design, it requires an administrator for the server to monitor activities on the server, as well as incoming and outgoing packets through the server. In addition to monitoring the packets traversed through the firewalls, the server administrator must also be responsible for making the server secure against attacks. For this scope of the project, the database administrator’s role is primarily setting up the database structure for the application to run. In the future, the need for database administrator is important. For example, in case of system failure, the database administrator would be in charge of recovering the database.

4.2. Detailed Architecture of LWP

As a web-based application, users have to interact with LWP through web pages. Each page stands for a distinct functionality of the system. The LWP application consists of numerous of modules that are listed in the following table.

	Name of module	Purpose
1	Registration module	This module will let the user to register into the system.
2	Confirmation email module	This module will confirm a successful registration to the user, and provide a link for user to activate his/her account.
3	Activation module	This module will activate the user’s account when he/she clicks the link from the confirmation email
4	Authentication module	This module authenticates user’s account in order to grant or deny a user to access the system.
5	Authorization module	Based on the status of user’s account (activated/not activated) or status of LWI and status of WDP, user will be in one specific type. According to the user’s type, there are a collection of actions that user can interact with the system.
6	Administration module	This module will allow the administrator to change the settings of the system, such as: minimum dates between two adjacent LWIs, or minimum records for displaying aggregate data (at least five records).
7	LWI module	This module will let the user to interact with LWI: create a new LWI, save an incomplete LWI, continue an incomplete LWI, review a complete LWI
8	WDP module	This module will let the user to interact with WDP: create a new WDP, save an incomplete WDP, continue an incomplete WDP, review a complete WDP, create a new WDP based on an existing WDP, edit an existing complete WDP.
9	Log out module	This module will safely log the user out.

Table 1. Core Modules in Chapter 4

The LWP application itself is composed of multiple layers, including a presentation layer, an application logic layer, and a database layer.

The various types of LWP user and associated conditions and capabilities are listed in the following table:

Type	Condition	Capability
1	• Not registered user	- Register a new account
2	• User has a registered account, but has not activated it.	- Activate account
3	• User's account is registered and activated. • No LWI record found associated with this account	- Create a new LWI
4	• User's account is registered and activated. • An incomplete LWI record found associated with this account	- Continue with the incomplete LWI
5	• User's account is registered and activated. • A complete LWI record found associated with this account	- Review complete LWI. - Start a new WDP for this LWI.
6	• User's account is registered and activated. • A complete LWI record found associated with this account. • No WDP record found associated with LWI	- Create a new WDP for LWI - Review complete LWI
7	• User's account is registered and activated. • A complete LWI record found associated with this account. • An incomplete WDP record found associated with LWI	- Continue with incomplete WDP. - Review complete LWI

8	• User’s account is registered and activated. • A complete LWI record found associated with this account. • One incomplete WDP found associated with LWI • One or more WDP record(s) found associated with LWI	- Review complete LWI. - Review complete WDP(s). - Continue incomplete WDP.
9	• User’s account is registered and activated. • A complete LWI record found associated with this account. • No incomplete WDP found associated with LWI • One or more complete WDP record(s) found associated with LWI. • None of the complete WDPs’ complete dates are no longer valid	- Review complete LWI. - Review complete WDP(s). - Create a new WDP. - Create a new WDP based on an existing complete WDP.
10	• User’s account is registered and activated. • A complete LWI record found associated with this account. • No incomplete WDP found associated with LWI • One or more complete WDP record(s) found associated with LWI. • One of the complete WDPs’ complete dates is still valid	- Review complete LWI - Review complete WDP(s). - Edit a complete WDP.

Table 2. Type of users in Chapter 4

4.3. Database Design

During the design, it was important to decide how to protect information from being read by unauthorized individuals. Users must be prevented from reading information which belonging to other users, and the database administrator must be prevented from reading user information. The solution that was chosen was to use a different encryption mechanism from that provided by the DBMS (in this case Oracle). Another important database design decision was what subset of data to encrypt. Encrypting all of the data is not reasonable since this could downgrade the performance of the system. Additionally, encrypting all of the data is not much more secure than encrypting private data and leaving public data unprotected.

In schema LWP_User which stores all users of this system, the developer decided to encrypt user's password with SHA-256 bit. SHA was chosen because it is a hash function also known as one-way encryption which is considered to be reasonably secure for password confidentiality. Before hashing the password, a random string is attached to the password, in order to prevent guessing the password in case more than one user has the same password. With this approach, if there is more than one user with the same password, the hash values are still different.

The LWI and WDP schemas are more complicated because there are so many data to store and protect such as: there are 180 fields in LWI and 21 fields for WDP. The developer decided not to have that many fields in the schema, because the size of schema would be very big and in addition, it would reduce the performance of retrieving data for decrypting each field for each record. Instead, the developer considered to treat the 180 fields in LWI as one encryption block, and 21 fields in WDP as another encryption block The blocks store the encrypted XML string for LWI and WDP respectively. So in this approach, the performance is much faster because it only needs to decrypt/encrypt once. The advantage of using XML is that it is well-organized and is easy to refer to any field when needed.

The following structure shows the XML structure of both LWI and WDP.

LWI

```
<root>
        <gender><value>x</value></gender>
        <age><value>x</value></age>
        <education><value>x</value></education>
        <section1>
                <q4><value>x</value></q4>
                …
                <q174><value>x</value></q174>
        </section1>
        …
        </section9>
        <percent1><value>x</value></percent1>
        …
        <percent9><value>x</value></percent9>
        <status><value>y</value></status>
        <current><value>x</value></current>
        <submit_date>x</submit_date>
</root>
```

WDP

```
<root>
        <complete>
                <area1><value>x</value></area1>
                <area2><value>x</value></area2>
                <area3><value>x</value></area3>
                <selected-area><value>x</value></selected_area>
                <first-result><value>x</value></first-result>
                <second-result><value>x</value></second-result>
                <third-result><value>x</value></third-result>
                <final-result><value>x</value><final-result>
                <healthy-activity><value>x</value></healthy-activity>
                <unhealthy-activity><value>x</value></unhealthy-activity>
                <selected-activity><value>x</value></selected-activity>
                <action-step1><value>x</value></action-step1>
                <action-step2><value>x</value></action-step2>
                <complete-date><value>x</value></action-step3>
                <chosen-action><value>x</value></chosen-action>
                <support><value>x</value></support>
                <current><value>x</value></current>
                <submit_date><value>x</value></submit_date>
                <id><value>x</value></id>
        </complete>
        ….
        <incomplete>
                <area1><value>x</value></area1>
                <area2><value>x</value></area2>
                <area3><value>x</value></area3>
                <selected-area><value>x</value></selected_area>
                <first-result><value>x</value></first-result>
                <second-result><value>x</value></second-result>
                <third-result><value>x</value></third-result>
                <final-result><value>x</value><final-result>
                <healthy-activity><value>x</value></healthy-activity>
                <unhealthy-activity><value>x</value></unhealthy-activity>
                <selected-activity><value>x</value></selected-activity>
                <action-step1><value>x</value></action-step1>
                <action-step2><value>x</value></action-step2>
                <complete-date><value>x</value></action-step3>
                <chosen-action><value>x</value></chosen-action>
                <support><value>x</value></support>
                <current><value>x</value></current>
                <submit_date><value>x</value></submit_date>
                </incomplete>
        </root>
```

The following figure showing the relationship and structure of each schema

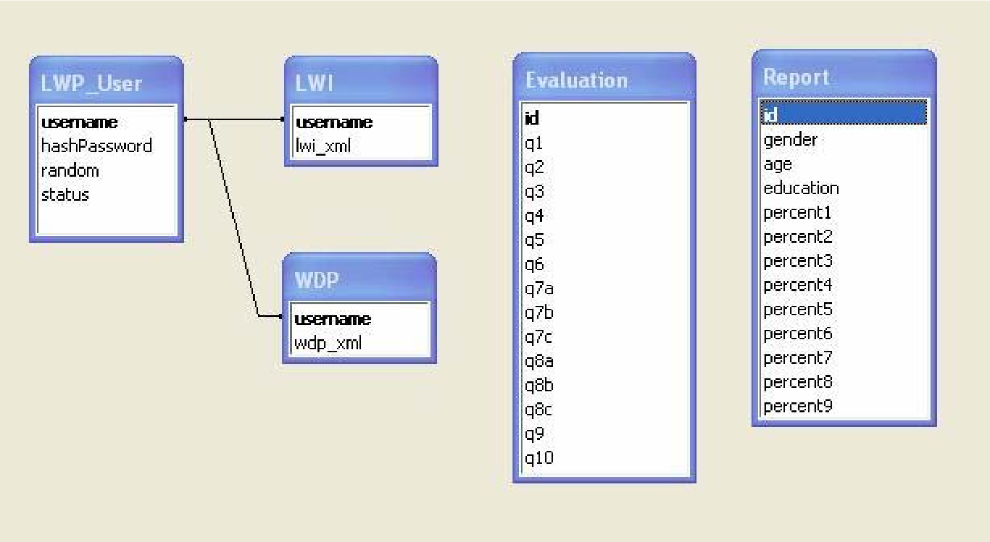


Figure 1. Database schema in Chapter 4

Column name	Description
username	Store value of username
hashPassword	Store hash value of combination password and random string
random	Store value of random string (created at registration)
status	Store the status of user (activated or not activated)

Table 3. LWP User Schema Description in Chapter 4

Column name	Description
username	Store value of username
lwi_xml	Store encrypted xml string of LWI

Table 4. LWI Schema in Chapter 4

Column name	Description
username	Store value of username
wdp_xml	Store encrypted xml string of WDP

Table 5. WDP Schema in Chapter 4

Column name	Description
id	Store value of record’s index
q1	Store value of question 1’s response
q2	Store value of question 2’s response
q3	Store value of question 3’s response
q4	Store value of question 4’s response
q5	Store value of question 5’s response
q6	Store value of question 6’s response
q7	Store value of question 7’s response
q8	Store value of question 8’s response
q9	Store value of question 9’s response
q10	Store value of question 10’s response

Table 6. Evaluation Schema in Chapter 4

Column name	Description
id	Store value of record’s index
gender	Store value of gender
age	Store value of age range
education	Store value of education level
percent1	Store value of percentage of section 1
percent2	Store value of percentage of section 2
percent3	Store value of percentage of section 3
percent4	Store value of percentage of section 4
percent5	Store value of percentage of section 5
percent6	Store value of percentage of section 6
percent7	Store value of percentage of section 7
percent8	Store value of percentage of section 8
percent9	Store value of percentage of section 9

Table 7. Report Schema in Chapter 4

4.4. Web User Interface Design

The LWP application contains a web-based graphical user interface, also known in some literature as a WUI (Web User Interface). This interface consists of many JSP/Servlet web pages through which the users interact. A JSP/Servlet page is basically a HTML page with associated code that is to be processed by the server before it is returned to the client. The code is specified in Java, a JSP/Servlet language. The code for the web page is written the same way as for any other object-oriented programming language using classes, methods, attributes, and the like. NetBeans takes care of associating the code with the associated HTML used for the actual presentation layer. Code may also be embedded directly into HTML. The code-behind methodology was selected by the developer instead, since it allows for better encapsulation, easier maintenance, and better readability at the price of some small amount of performance.

Regardless of the coding method used for the web pages, all JSP/Servlet code is processed on the web server and is replaced with HTML before the response is sent back to the requesting client. This allows multiple browsers like Internet Explorer and Netscape Navigator, as well as multiple versions of the browsers to interact with the system, even though the client browsers may be on non-Windows platforms or older versions of the Windows operating system.

4.5. Security Design

As a web-based application in the health area, security is of paramount concern. Because all information transmitted through this application is health-sensitive information related to individuals, security is considered the highest priority to meet some acceptable levels of HIPAA requirements. The most difficult problem for developer was how to make sure confidentiality and privacy were preserved for any individual's information. A solution had to be developed to satisfy the security needs of not only the application web server, but also the database.

How HIPAA (Health Insurance Portability and Accountability Act) **is applied in this project?**

Much of the current Health Insurance Portability and Accountability Act is concerned with the administrative simplification standards; however, there is much more to this law. HIPAA is a multifaceted piece of legislation covering the following three areas:

- Insurance portability – implemented
- Fraud enforcement (accountability) – implemented
- Administrative simplification – implementation pending

The first two components of HIPAA, portability and accountability, have been put into affect. Portability ensures that individuals moving from one health plan to another will have continuity of coverage and not be denied coverage under pre-existing condition clauses. Accountability significantly increases the federal government's fraud enforcement authorities in many different areas. The third component, administrative simplification, is arguably the most significant piece of the legislation, even though it is not mentioned in the title and received very little attention when the law was first enacted because its implementation date was not imminent. However, the implementation of HIPAA administrative simplification is quickly approaching [2].

With the scope of the project, the developer focuses on the last two areas. Among those areas, administrative simplification is the most difficult. It primarily focuses on Privacy rule and Security rule. The privacy rule is a standard stipulating that any health plan, health care clearinghouses, and certain health care providers have to apply in order to protect the privacy and confidentiality of patients' personal health information. While Security rule is another standard for health plans, health care clearinghouses, and certain health care providers to assure their customers that the confidentiality and privacy of health care information they electronically collect, maintain, use, or transmit is secure. Security of health information is especially important when health information can be directly linked to an individual.

Why HIPAA must be applied in this project?

There is time constraint that requires all health plans, health care clearinghouses and health care providers (covered entity) to apply HIPAA since October 2002 (according to the law signed by President Bush on December 27, 2001) [9]. The LWP application stores and transmits its customers' health information, so it is a covered entity as a health care provider (according to HIPAA glossary).

What is covered in the scope of this project?

Applying HIPAA Privacy rule

The privacy rule concerns about the privacy and confidentiality of health care customers. The LWP ensures the privacy of its customer by providing the registration and activation mechanism. The user is required to register with a valid LWP account in order to be able to use the system. After the registration stage, an activation reminder email is sent to user to remind user to activate his/her account. This confirmation email is sent only after user has registered to LWP to make sure that user is not an unintended party. To strengthen the privacy, the LWP was designed in such a manner so that no one else besides the owner of the account knows the username and password. The user’s password goes through a SHA-256 hash function before it is stored in database. The LWP ensures that the user’s identity is not disclosed to any third party by asking the user not to use his/her name or any information that can identify him/her. The user’s personal information including the LWI and WDP is protected with an encryption mechanism. This encryption guarantees that no one else besides the owner can see the information. Although user’s information is encrypted in the same procedure, function for every user, AES (Advanced Encryption Standard) 256 bits, this mechanism uses user’s hashed SHA-256 password as the key to encrypt/decrypt.

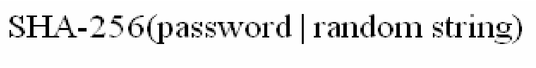

SHA-256(password | random string)

Figure 2. Hashed password and random string in Chapter 4

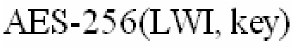

AES-256(LWI, key)

Figure 3. Encrypted LWI in Chapter 4

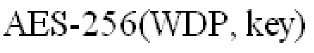

AES-256(WDP, key)

Figure 4. Encrypted WDP in Chapter 4

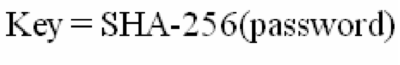

Key = SHA-256(password)

Figure 5. Key used in encrypt/decrypt LWI/WDP in Chapter 4

Applying HIPAA Security Rule

Web server

Sygate firewall software is used to monitor all incoming and outgoing traffic through the web server for this project.

Application

In this project, the developer enabled validation and verification in both client-side and server-side of the application. It is apparent that client-side validation is not secure, since a hacker can see the Java Script code and changes the script to try different values until something unauthorized is retrieved. Another security feature enabled by the developer was to prohibit the user from submitting HTML or XML through any of the input controls that could contain malicious script if processed. This prevents a web-based attack known as “a cross-site scripting attack” in which a hacker tries to steal session.

Moreover there are also authentication and authorization to protect the application from hackers. The user must register and be activated to use the application. After registering, there is a confirmation email sent to user’s email address asking the user to activate his/her account by clicking on the link residing inside the email. There is a random string associated with the link to make sure that user cannot activate his/her account without the confirmation email. This approach prevents hackers from creating a spoof confirmation email with the link direct to hacker’s web server. In that case, hacker might have the user’s username and password, but cannot activate the account, because hacker doesn’t have the confirmation email. A user’s password is only stored in the database in encrypted. This kind of encryption will ensure that even database administrator cannot guess or retrieve the password, because it is a one- way encryption. The user is also asked to be careful when choosing his/her username to make sure that no one else can guess the identity of the user through username.

Comment [DR1]: This is about the fifth time you’ve said this. A better organization wouldn’t be so repetitive.

To prevent eavesdropping and man-in-the-middle attack, the developer chose SSL as a secure protocol to protect communication between client and server. Since there is an encryption and decryption mechanism behind the scene in SSL protocol, there is no need to build another mechanism which might downgrade the performance. For current academic purposes, the ITS (information technology services) from the University of Wisconsin-La Crosse provided a free certificate to enable SSL for this project.

There are both advantages and disadvantages of using this free certificate in comparison with that of a commercial CA companies such as: Thawte, Veri sign. One of the obvious advantages is that commercial certificate costs at least a few thousands of dollars, while the certificate from the University is free of charge. There are some limitations with the current certificate: there is no guarantee that all internet browsers can recognize and run the application on client side, and there is no full commitment for technical support from the ITS (Information Technology Service) since it is not their responsibility. In the future, if this application is commercially used, there must be a certificate from well-known CA companies to replace the current certificate.

Database

The project sponsor asked the developer to draw a solution so that unauthorized person is prohibited from seeing authorized information. This ensures privacy and confidentiality of the project. In the database, information must be stored with an encryption. There are many possible solutions found, but only a few are reasonable to meet both the security and performance requirements. Using the encryption provided by the Oracle DBMS gives the DBA the right to decrypt and see the information which is not supported by the project. The developer had to use a separate encryption mechanism to protect the privacy of information. As a result, AES-256 seems to be a good match, because it is a symmetric encryption that does not reference public/private key management or transmission. Information collected from user’s form is combined to form an XML string encrypted with user SHA-256 hashed password as the key. With this approach, the user must be responsible for his/her privacy and confidentiality by not giving the username and password to anyone.

What is not covered in the scope of this project?

- There is no LWP contingency plan covering 1) data backup, 2) preparing critical systems that would be used to continue operations, or 3) recovering from a system disaster.
- There is no actual plan to secure the server physically. The server is resides in MSE Lab which can be accessed by any MSE student or friend of MSE student. Although only the developer can access the server, anyone who gains access to this Lab can be a threat for the availability of LWP.
- The database server is shared with the university's server, which may represent vulnerabilities if in the future more users assume management responsibilities.

4.6. Deploying LWP

LWP is deployed by installing the application to a web server running Tomcat 5.5. The web server must also have the JVM (Java Virtual Machine) installed on it. There are some predefined classes or jar files relating to security and database connectivity that must be placed in the application package before installing the application. Installing the application to the web server requires application distribution jar file and is achieved by using the deployment function provided by Tomcat.

5. Limitations

This section enumerates some of the limitations of the current project.

- The user’s ability to create his/her own configurable profile is not supported in this version. For example, a user may want to change the current font in the application to his/her favorite, or even change the appearance of each page, user might want to change the number of records appeared at a time. These options are not supported.
- Another limitation is the database platform supported by the application, since Oracle is the only platform supported.
- HIPAA standards are not fully preserved due to budget and room limitations.
- There is no graph report with the current LWI and evaluation reports.
- According to users’ feedbacks as well as the sponsor’s opinion, there are some limitations with the product:
    - Although, LWP’s GUI is easy to use, and simple, it is not quite attractive to users.
    - There is no type of chat session application or forum supported with the current version in order to decrease the high volume of direct communication between participant and LWP facilitators or instructors.

6. Continuing work

This section lists some of the planned work to continue from the current version:

- For the current version, the GUI is not very attractive, though it is user-friendly and easy to use, this could discourage a user to use the system more often.
- Talking agent is not included in the current version. With the help of talking agent, the user will be encouraged to interact more with the system.
- The future version should provide a type of location/organization so that report based on location/organization can be issued. This functionality will no longer restrict application's scope for only individuals but organizations.
- There are no importing, exporting and backing up mechanism for database through GUI supported in the current version. Providing this option in the future will help in emergency situations.
- Web Server, Database Server or any resource associated with this project must be placed in a highly secured place. These places must be well protected and restricted to unauthorized people. (HIPAA)
- Although there is a good and secure protocol between client and server, it needs a separate CA (Certificate Authority) to strengthen the security of this application rather than that of the university.
- At present, Oracle is the only database platform allowed. The application should support a variety of database platforms such as DB2, SQL server and so on in the future.
- Crystal report should be included to enhance the report functionality.

- With the rapid popularity of mobile phone, WAP technology should be included in order to make LWP available to mobile phones.

7. Conclusion

LWP is a multi-user web-based application. It consists of two main components: LWI and WDP. The aim of LWI component is to gather information from participants about their wellness levels based on the nine areas: Rest and Relaxation, Emotional and Mental Health, Sexuality, Personal Health Habits, Fitness, Nutrition, Drugs, Safety, and Environmental Sensitivity. In the wellness perspective, LWI and WDP support each other. Once knowing his/her wellness level through LWI, participants interact with the WDP in order to build individual action plans to improve one's wellness level. After a period of time since the last LWI result, the participant is enabled to interact with LWI again to evaluate the impact of action plan on his/her health level. This process is repeated as many times as a participant wants.

The LWP has been used as an educational tool in HPR-105 and HED-205 courses at this university where there were approximately 1000 students during the Fall semester of 2006. Besides being used in the university, the LWP was also tested by the staff in the School District of Onalaska, School District of Westby and the employees of La Crosse County. The feedback from the participants contributed to the improvement of the process.

8. Bibliography

1. V. Chopra, A. Bakore, J. Eaves, B. Galbraith, S. Li, C. Wiggers, *Professional Apache Tomcat 5*, Wrox, 2004.
2. C. E. Coleman, Andrew H. Joseph, *HIPAA Self-Assessment And Planning, A Guide to the Privacy and Security Standards* (Second Edition), Opus, 2001.
3. A.M. Davis, *Software Requirements Analysis and Specification*, Prentice Hall, 1990.
4. B. Galbraith, P. den Haan, L. Lavandowska, S. N. Panduranga, K. Perrumal, E. Sgarbi, *Beginning JSP 2.0 Build web Applications using JSP, Java, and Struts*, Wrox, 2003.
5. G.D. Gilmore, *Health Education*, 1979.
6. G. D. Gilmore, *La Crosse Wellness Project – Wellness Development Process (original version)*, 1989.
7. M. Hall, L. Brown, *core Servlets and JavaServer Pages Volume 1: Core Technologies* (Second Edition), Prentice Hall, 2004.
8. E. R. Harold, *Processing XML with Java A Guide to SAX, DOM, JDOM, JAXP, and TrAX*, Addison-Wesley, 2003.
9. P. Keohane, *HIPAA Regulatory Manual* (Updated For 2002), The Medical Management Institute, 2002.
10. P. M. Lewis, Arthur Bernstein, Michael Kifer, *Databases And Transaction Processing An Application-Oriented Approach*, Addison Wesley, 2001.
11. J. Price, *Oracle9i JDBC Programming Write JDBC Programs for the Oracle Environment*, Mc Graw Hill Osborne, 2002.
12. E. J. Schafer, *The Development And Evaluation Of An Electronic version Of The Wellness Development Process As Part Of The La Crosse Wellness Project*, May 2006.

13. R. Schreiber, H. Schreiber, *Java Server & Servlets Building Portable Web Applications*, Addison-Wesley, 2000.

14. K. Squire, G. D. Gilmore, D. Duquette, and D. Riley, “The National Content Revalidation of the La Crosse Wellness Inventory as a Part of the La Crosse Wellness Project”, *American Journal of Health Education*, ?, May/June 2001, Volume 32, No.3, pp. 183-188.

15. I. Sommerville, *Software Engineering 6th Ed*., Addison Wesley, 2001

16. M. A. Weiss, *Data Structures & Algorithm Analysis In Java*, Addison-Wesley, 1999.

17. R. A. Wyke, J. D. Gilliam and C. Ting, *Pure JavaScript A Code-Intensive Premium Reference*, Sams, 1999.

18. D. C. Xiong, “Software development for the Wellness Development Process, La Crosse Wellness Project”, *Master’s thesis*, University of Wisconsin-La Crosse, 1998.

APPENDIX A: Sample LWP Use-case Diagram

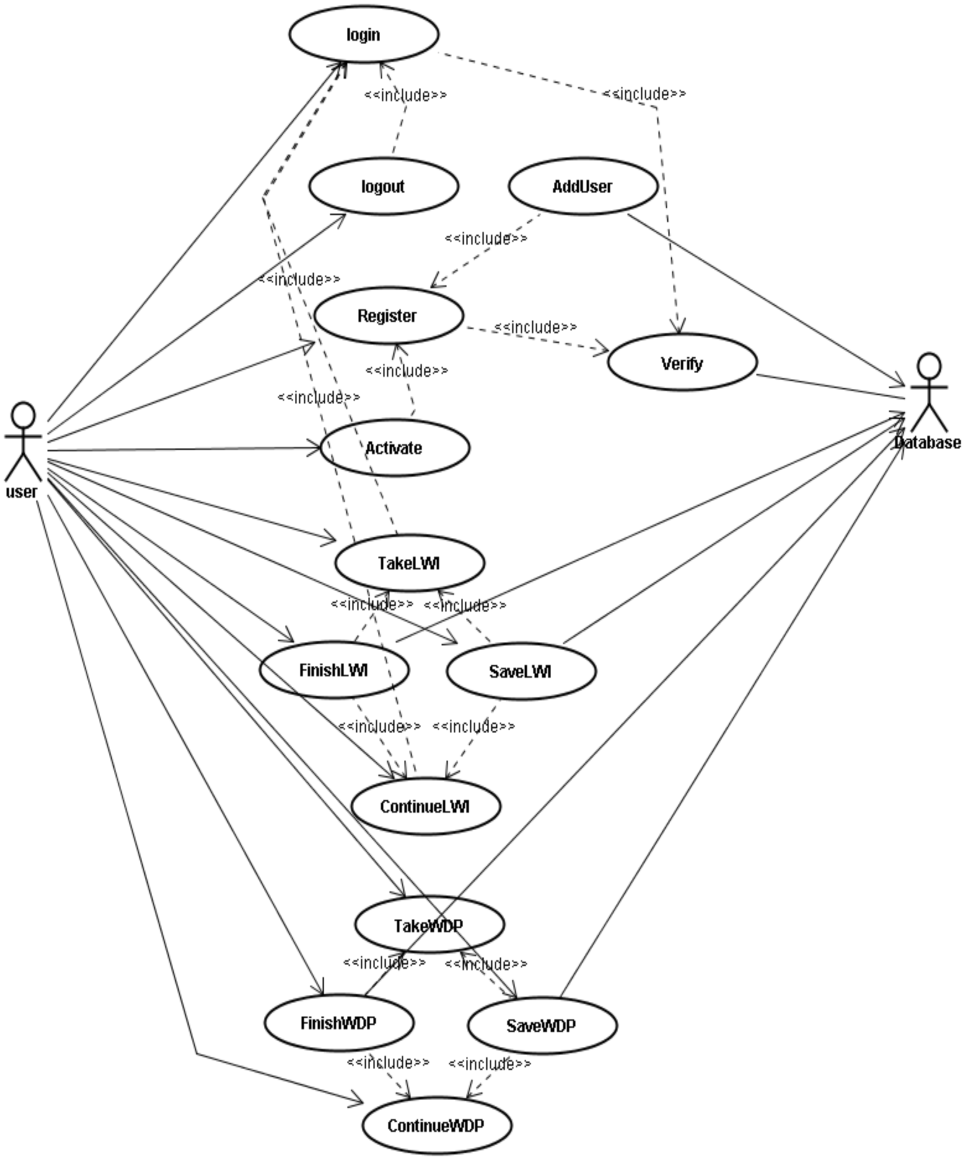


Figure 6. LWP Use-case diagram in Appendix A

APPENDIX B: Sample LWP Class Diagram

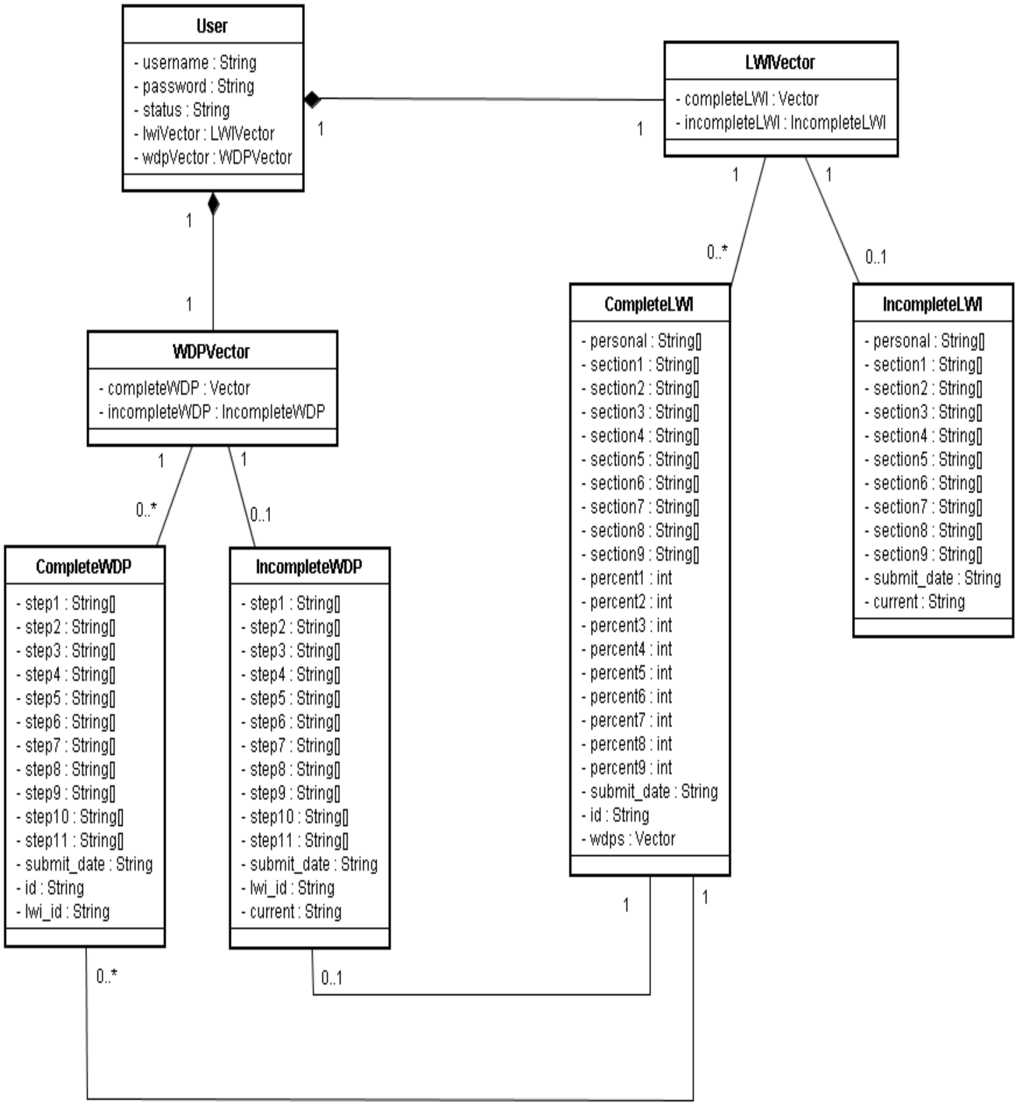
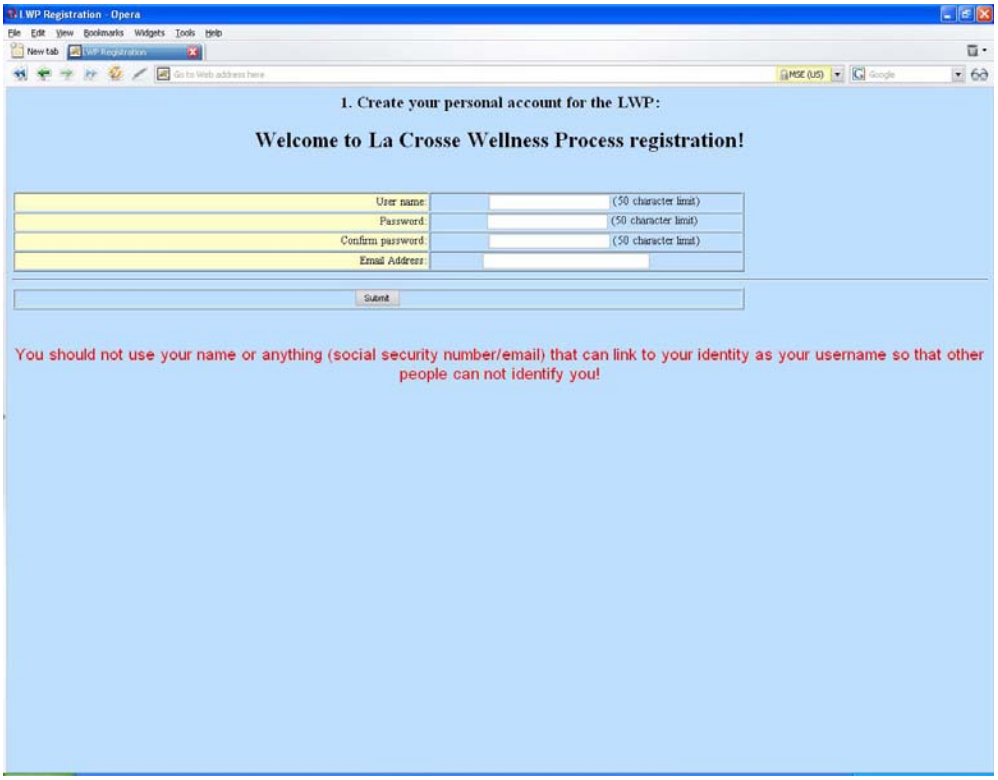


Figure 7. LWP Class Diagram in Appendix B

APPENDIX C: LWP Registering Page



1. Create your personal account for the LWP:

Welcome to La Crosse Wellness Process registration!

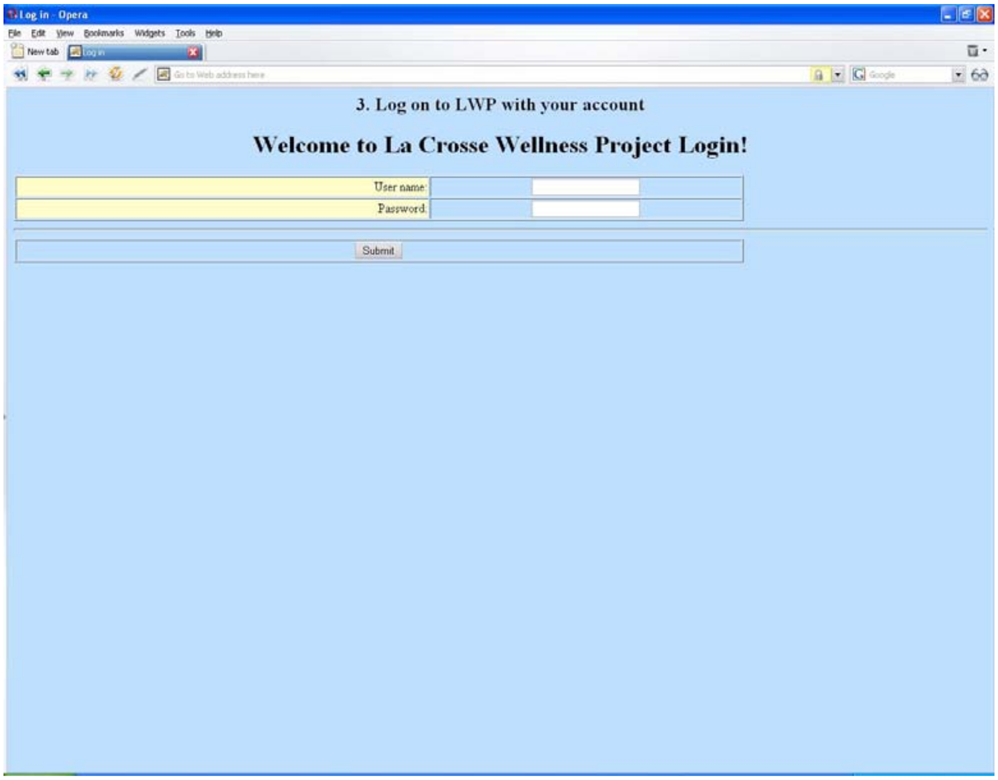
User name:		(50 character limit)
Password:		(50 character limit)
Confirm password:		(50 character limit)
Email Address:		

Submit

You should not use your name or anything (social security number/email) that can link to your identity as your username so that other people can not identify you!

Figure 8. LWP Registering Page in Appendix C

APPENDIX D: LWP Login Page



3. Log on to LWP with your account

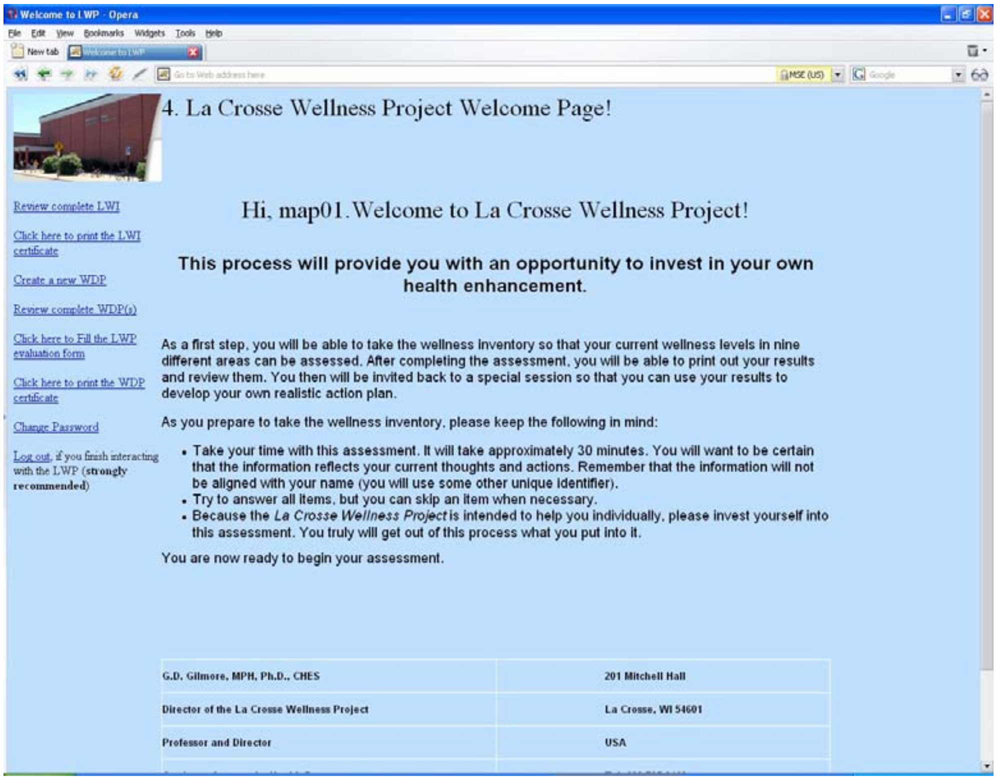
Welcome to La Crosse Wellness Project Login!

User name:	
Password:	

Submit

Figure 9. LWP Login Page in Appendix D

APPENDIX E: LWP Welcome Screen



4. La Crosse Wellness Project Welcome Page!

Review complete LWI

Click here to print the LWI certificate

Create a new WDP

Review complete WDP(s)

Click here to Fill the LWP evaluation form

Click here to print the WDP certificate

Change Password

Log out, if you finish interacting with the LWP (**strongly recommended**)

Hi, map01.Welcome to La Crosse Wellness Project!

This process will provide you with an opportunity to invest in your own health enhancement.

As a first step, you will be able to take the wellness inventory so that your current wellness levels in nine different areas can be assessed. After completing the assessment, you will be able to print out your results and review them. You then will be invited back to a special session so that you can use your results to develop your own realistic action plan.

As you prepare to take the wellness inventory, please keep the following in mind:

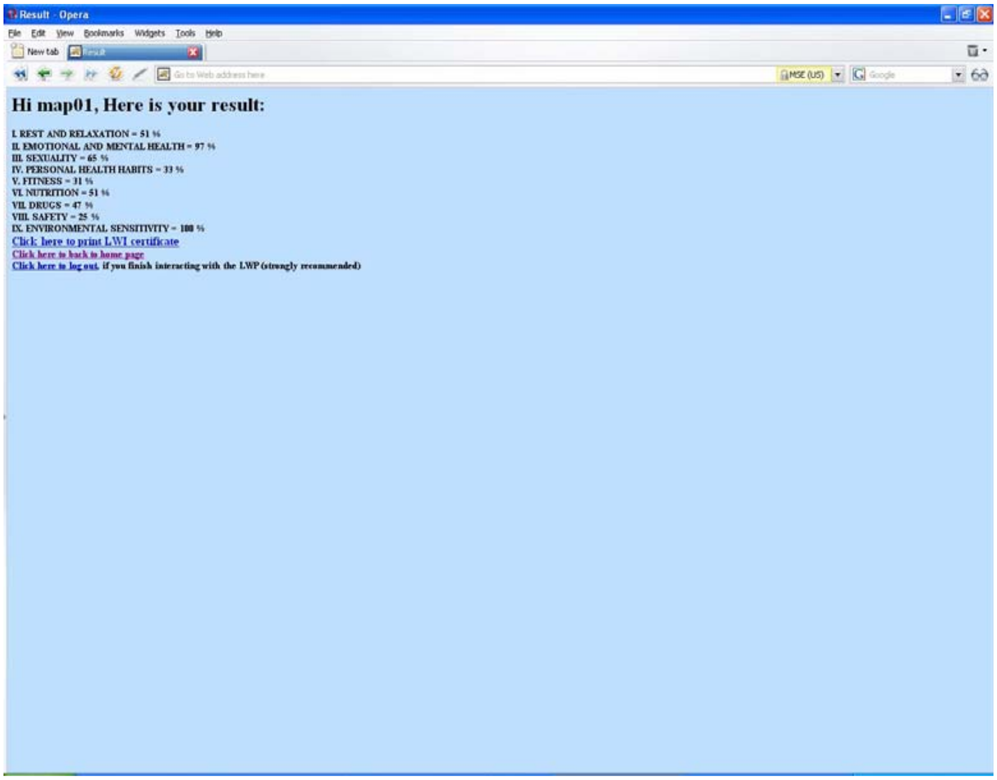
- Take your time with this assessment. It will take approximately 30 minutes. You will want to be certain that the information reflects your current thoughts and actions. Remember that the information will not be aligned with your name (you will use some other unique identifier).
- Try to answer all items, but you can skip an item when necessary.
- Because the *La Crosse Wellness Project* is intended to help you individually, please invest yourself into this assessment. You truly will get out of this process what you put into it.

You are now ready to begin your assessment.

G.D. Gilmore, MPH, Ph.D., CHES	201 Mitchell Hall
Director of the La Crosse Wellness Project	La Crosse, WI 54601
Professor and Director	USA

Figure 10. LWP Welcome Screen in Appendix E

APPENDIX F: Sample LWI Result Page

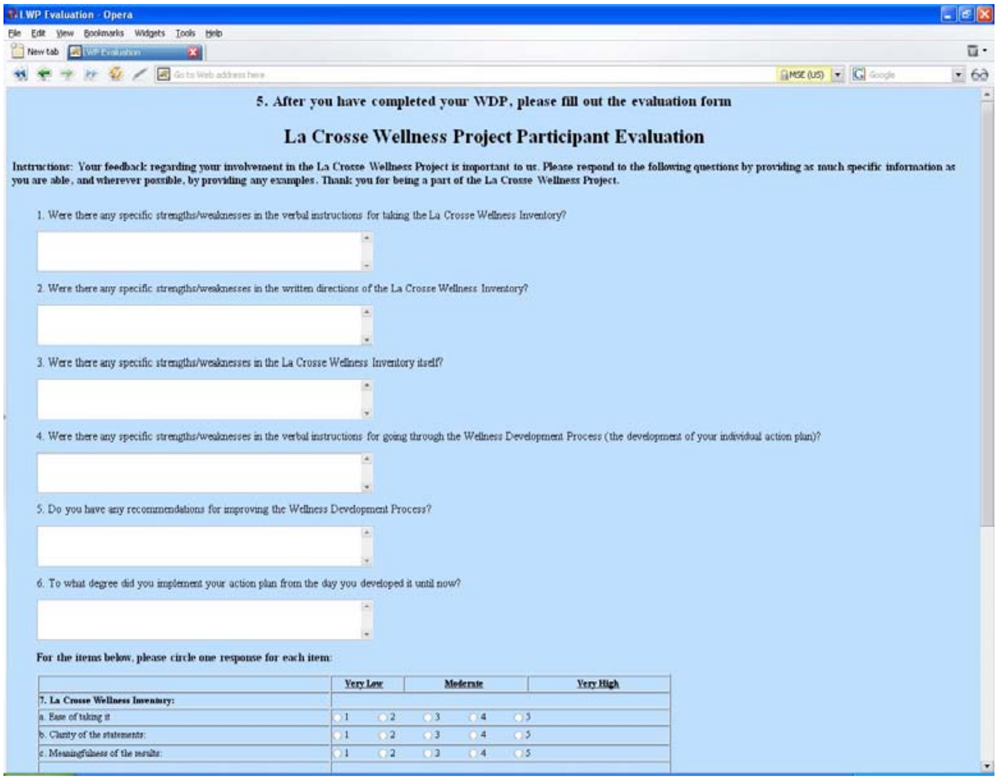


Hi map01, Here is your result:

I. REST AND RELAXATION = 51 %
II. EMOTIONAL AND MENTAL HEALTH = 97 %
III. SEXUALITY = 65 %
IV. PERSONAL HEALTH HABITS = 33 %
V. FITNESS = 31 %
VI. NUTRITION = 51 %
VII. DRUGS = 47 %
VIII. SAFETY = 25 %
IX. ENVIRONMENTAL SENSITIVITY = 100 %
Click here to print LWI certificate
Click here to back to home page
Click here to log out, if you finish interacting with the LWP (strongly recommended)

Figure 11. Sample LWI Result Page in Appendix F

APPENDIX G: LWP Evaluation Page



5. After you have completed your WDP, please fill out the evaluation form

La Crosse Wellness Project Participant Evaluation

Instructions: Your feedback regarding your involvement in the La Crosse Wellness Project is important to us. Please respond to the following questions by providing as much specific information as you are able, and wherever possible, by providing any examples. Thank you for being a part of the La Crosse Wellness Project.

1. Were there any specific strengths/weaknesses in the verbal instructions for taking the La Crosse Wellness Inventory?

2. Were there any specific strengths/weaknesses in the written directions of the La Crosse Wellness Inventory?

3. Were there any specific strengths/weaknesses in the La Crosse Wellness Inventory itself?

4. Were there any specific strengths/weaknesses in the verbal instructions for going through the Wellness Development Process (the development of your individual action plan)?

5. Do you have any recommendations for improving the Wellness Development Process?

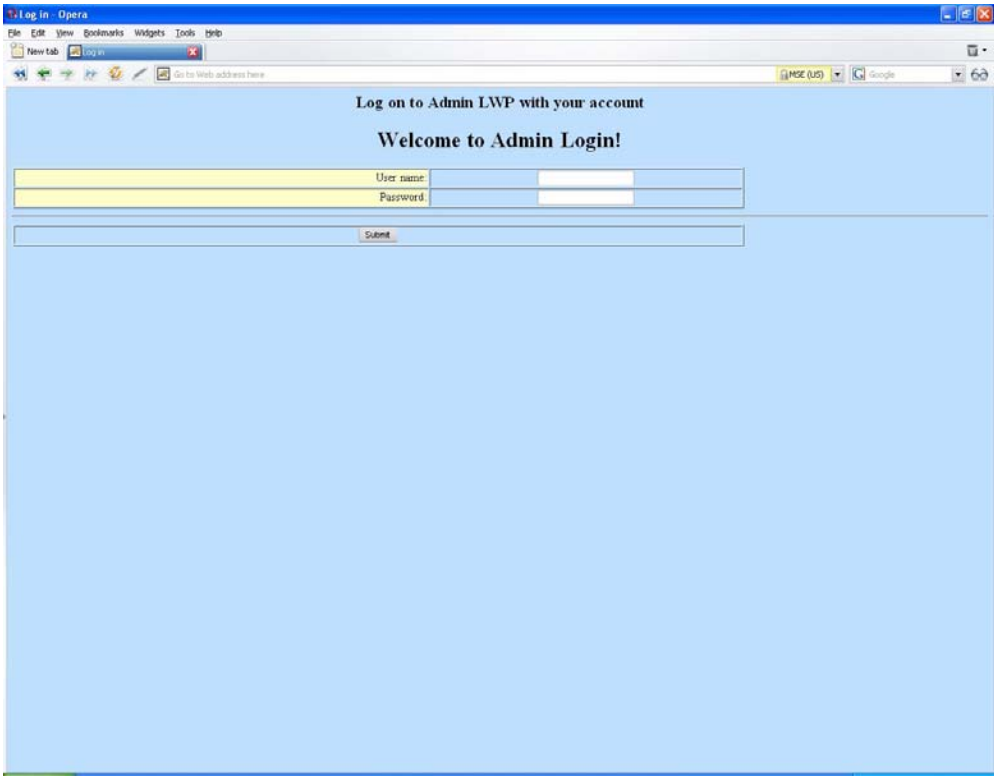
6. To what degree did you implement your action plan from the day you developed it until now?

For the items below, please circle one response for each item:

	Very Low		Moderate		Very High
7. La Crosse Wellness Inventory:					
a. Ease of taking it	○ 1	○ 2	○ 3	○ 4	○ 5
b. Clarity of the statements:	○ 1	○ 2	○ 3	○ 4	○ 5
c. Meaningfulness of the results:	○ 1	○ 2	○ 3	○ 4	○ 5

Figure 12. LWP Evaluation Page in Appendix G

APPENDIX H: LWP Admin Login Page



Log on to Admin LWP with your account

Welcome to Admin Login!

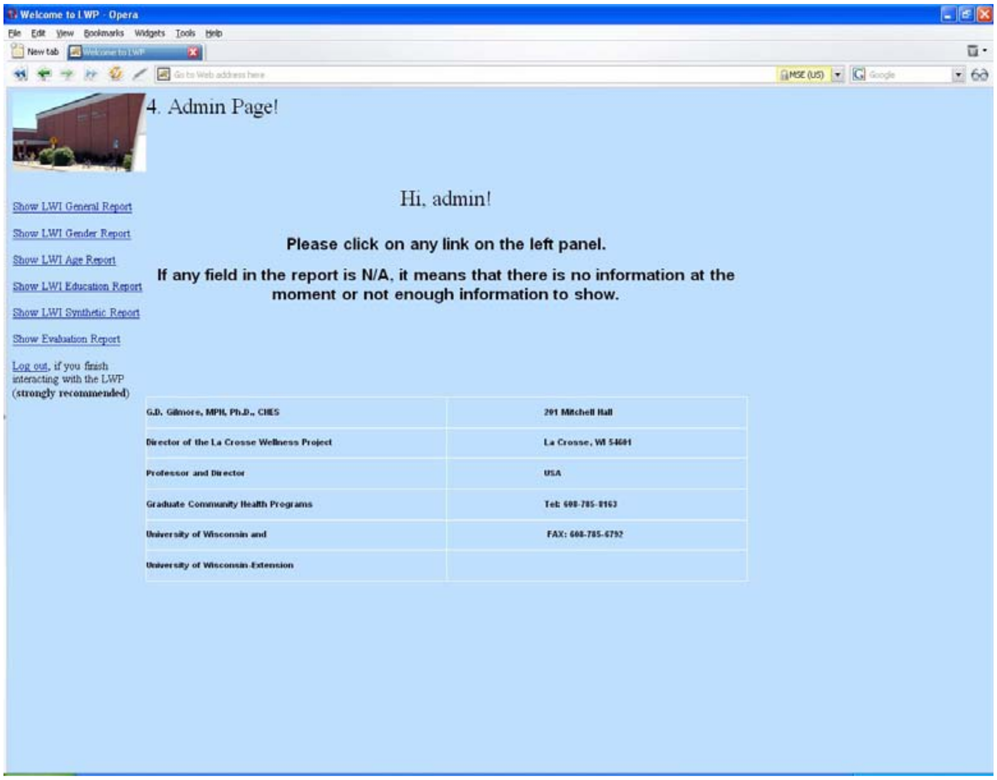
User name:

Password:

Submit

Figure 13. LWP Admin Login Page in Appendix H

APPENDIX I: LWP Admin Page



4. Admin Page!

Show LWI General Report

Show LWI Gender Report

Show LWI Age Report

Show LWI Education Report

Show LWI Synthetic Report

Show Evaluation Report

Log out, if you finish interacting with the LWP **(strongly recommended)**

Hi, admin!

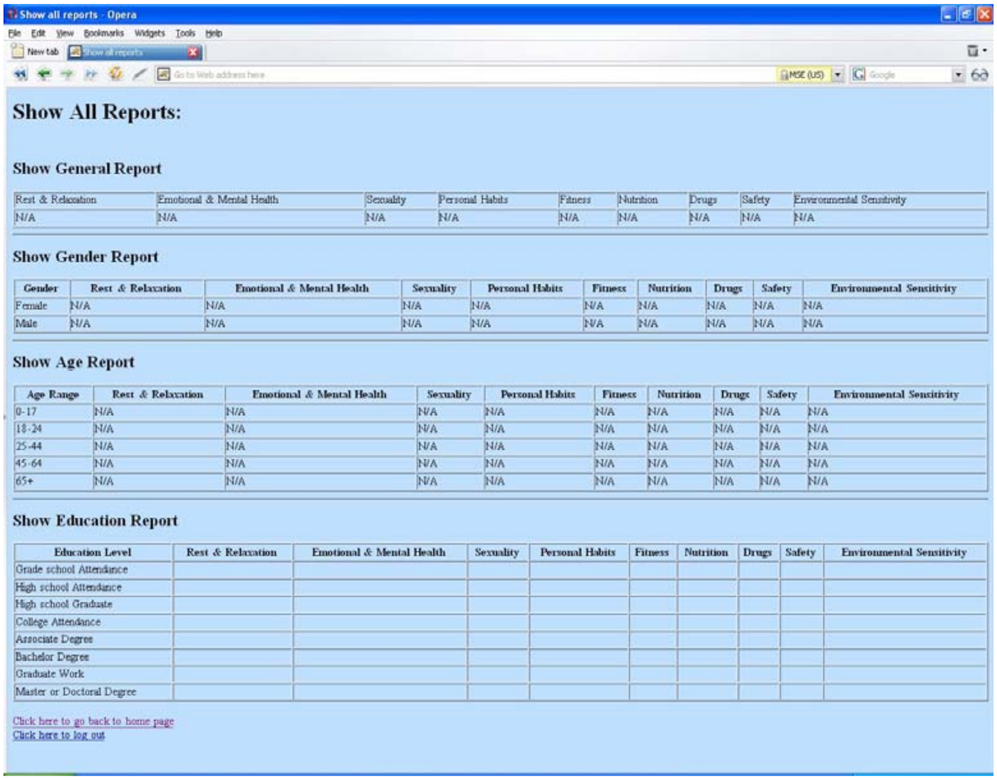
Please click on any link on the left panel.

If any field in the report is N/A, it means that there is no information at the moment or not enough information to show.

G.D. Gilmore, MPH, Ph.D., CHES	201 Mitchell Hall
Director of the La Crosse Wellness Project	La Crosse, WI 54601
Professor and Director	USA
Graduate Community Health Programs	Tel: 608-785-8163
University of Wisconsin and	FAX: 608-785-6792
University of Wisconsin-Extension	

Figure 14. LWP Admin Page in Appendix I

APPENDIX J: LWP All Report Page



Show All Reports:

Show General Report

Rest & Relaxation	Emotional & Mental Health	Sexuality	Personal Habits	Fitness	Nutrition	Drugs	Safety	Environmental Sensitivity
N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Show Gender Report

Gender	Rest & Relaxation	Emotional & Mental Health	Sexuality	Personal Habits	Fitness	Nutrition	Drugs	Safety	Environmental Sensitivity
Female	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Male	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Show Age Report

Age Range	Rest & Relaxation	Emotional & Mental Health	Sexuality	Personal Habits	Fitness	Nutrition	Drugs	Safety	Environmental Sensitivity
0-17	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
18-24	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
25-44	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
45-64	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
65+	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Show Education Report

Education Level	Rest & Relaxation	Emotional & Mental Health	Sexuality	Personal Habits	Fitness	Nutrition	Drugs	Safety	Environmental Sensitivity
Grade school Attendance									
High school Attendance									
High school Graduate									
College Attendance									
Associate Degree									
Bachelor Degree									
Graduate Work									
Master or Doctoral Degree									

Click here to go back to home page
Click here to log out

Figure 15. LWP All Report Page in Appendix J