

**2007**

**University of North Carolina Wilmington**  
**Master of Science in**  
**Computer Science and Information Systems**  
**Proceedings**

<https://csbapp.uncw.edu/mscsis>

**MS CSIS Capstone Research Project**  
**A Solution for Community Development**

**By**

**Christopher K. Holtsford**

**The University of North Carolina Wilmington**

**Wilmington, North Carolina**

**May 2007**

**Project Committee**

**Thomas N. Janicki (chair)**

**Douglas M. Kline**

**Ron Vetter**

**Charles Laymon (external professional)**

**Submitted as a portion of the requirements for the  
Master of Science Degree in Computer Science Information Systems**

# Table of Contents

<b>Introduction</b> .....	1
<b>Project Background</b> .....	4
Client Benefits.....	4
<b>Project Description</b> .....	6
System Design Overview.....	6
Membership Overview.....	6
Messaging System Overview.....	8
Web Content Management Overview.....	9
Implementation Requirements.....	10
Technical Approach.....	11
Presentation Layer.....	12
Business Logic Layer.....	14
Data Access Layer.....	16
<b>Systems Analysis</b> .....	17
Methodology.....	17
Inception Phase.....	19
Interviews.....	20
Problem Definition.....	21
Background.....	21
Audience.....	21
Organizations.....	23
Current Problems.....	23
System Objectives.....	24
Overview.....	24
Membership.....	24
Messaging System.....	25
Limited Web Content Management System.....	25
Elaboration Phase.....	26
Use Case Diagrams.....	27
Construction Phase.....	27
Transition Phase.....	31
Alternate Analysis Methodologies.....	32
Spiral Model.....	32

Waterfall Model.....	34
Technologies.....	35
Software and Programming Languages.....	36
Membership Implementation.....	38
Messaging System Implementation.....	40
Limited Web Content Management Implementation.....	42
<b>Implementation.....</b>	<b>45</b>
Database.....	45
Design Patterns.....	45
Testing.....	46
<b>Lessons Learned.....</b>	<b>47</b>
New Technologies, Techniques, and Skills.....	47
What Worked Well.....	49
Areas of Improvement.....	50
<b>Conclusion.....</b>	<b>52</b>
<b>Acknowledgements.....</b>	<b>54</b>
<b>Appendix A - Client Interviews.....</b>	<b>56</b>
<b>Appendix B - Use Case Diagrams.....</b>	<b>61</b>
<b>Appendix C - Unified Process Tasks.....</b>	<b>71</b>
<b>Appendix D - Application Screen Shots.....</b>	<b>79</b>
<b>Appendix E - Database Diagrams.....</b>	<b>93</b>
<b>Appendix F - System Sequence Diagrams.....</b>	<b>109</b>
<b>Appendix G - Project Metrics.....</b>	<b>114</b>
<b>References.....</b>	<b>116</b>

# MS CSIS Capstone Research Project

## A Solution for Community Development

### Abstract

*This report will detail the analysis, design, and implementation of a community development solution for two non-profit organizations: the Wilmington Family YMCA and Y-Tri Club. The solution consisted of implementing role based membership, a messaging system to increase community communication, and a Web content management system.*

### Introduction

To satisfy the capstone project requirements for the Masters of Science in Computer Science Information Systems I have analyzed, designed and implemented a solution that has greatly enhanced the functionalities and usability of two related web sites. They are the Wilmington Family YMCA and Y-Triathlon (Y-Tri) Club's Web applications. The project consisted of three key enhancements to their online capabilities:

- An authentication and registration system for members seeking information from their "Intranet" sites. Specifically the introduction of roles, role verification and enhanced password protection.
- A messaging system to provide a means for employees and members to communicate with one another

- A limited web content management system to provide administrative personnel a means to easily update the content on the two web sites

This project not only benefited the YMCA as will be described later, it also personally challenged me. The project also enabled me to achieve a higher degree of technical experience with web applications and security issues; provide a service to the community, and meet community development goals. I have accomplished these goals through the YMCA and Y-Tri Club Web applications. My personal knowledge increased in the following areas:

- Advanced programming techniques in the .NET 2.0 Framework
  - .NET 2.0 Framework libraries and development environment
  - How to expand the functionality of the .NET 2.0 components to fit the needs of the user by inheriting from and expanding the .NET intrinsic classes
- Implementing proper Web application design utilizing n-tier application development
  - User interface layer (presentation)
  - Data access layer
  - Database layer
- Increased database design and security knowledge
  - Provide secure transmissions via SSL
  - Encrypt sensitive user data
  - Normalized relational database tables
    - Super type paradigm
- Project management experience
  - Developing a timeframe and then delivering project components to meet these deadlines

- Working with a real client
- Increase communication skills through meeting with clients on a regular basis
- Meeting client expectations

The project was also a great opportunity to provide the community and two non-profit organizations (Wilmington Family YMCA and Y-Tri Club) with new technologies that will assist their service to the Wilmington community. Their specific goals and benefits include:

- Provide YMCA community members with a service in which they can receive electronic messages on YMCA and Y-Tri Club event updates and messages from the CEO and other YMCA officials
- Provide a medium for which YMCA and Y-Tri Club members can communicate with one another. These members can easily collaborate and coordinate team practices among many other activities (peer to peer)
- Allow YMCA designated personnel with administrative privileges to upload and update YMCA web content without having to know any HTML programming language requirements.
  - Allows for real-time changes to be made to increase content accuracy
  - Allows time sensitive updates to be posted quicker

## **Capstone Project Background**

I first got involved with the YMCA and Y-Tri Club while interning with Charles Laymon of StepQuest, Inc. in the summer of 2006. The purpose of the internship was to gain knowledge of proper web and database development techniques and best practices within the .NET 2.0 Framework. As a result of this internship, it was determined a need existed at the YMCA for new applications. These applications would also advance my analysis, design and web development skills. The final internship deliverables proposed were:

- An integrated solution to allow all Y-affiliated web sites to run under a single code base
- Use proper design and development techniques such as n-tier web application design

After above enhancements had been made to the YMCA and Y-Tri Club web applications during the internship, the clients at the YMCA identified three new requirements. These evolved into the foundation of my capstone project:

- Membership capabilities (authorize and registration of a member profile)
- A messaging system for member collaboration
- A limited content management system

### **Client Benefits**

These three system enhancements will provide the following key benefits to the client:

- Better communication from management to YMCA and Y-Tri Club members
  - CEO messages
  - Registration announcements (i.e. classes)

- Important changes to programs (i.e. schedule changes)
- Better communication between YMCA and Y-Tri Club members (a new member to member communication system) to assist to:
  - Organize team practices
  - Organize group workout/training sessions
- Faster updates to YMCA and Y-Tri Club web content
  - No workflow to a third party or to a 'web guru', updates are made directly to the content by YMCA or Y-Tri Club management as needed
  - HTML editor allows for non-technical personnel to manage content, eliminating the dependence of a third party to manage content
- More accurate updates to YMCA and Y-Tri Club content
  - Updates are carried out by YMCA or Y-Tri Club management, not a third party, therefore updates will be more accurate
  - Content will stay up-to-date as the clients can manage content themselves
    - Community events
    - Registration dates
    - YMCA and Y-Tri Club related event dates

## **Project Description**

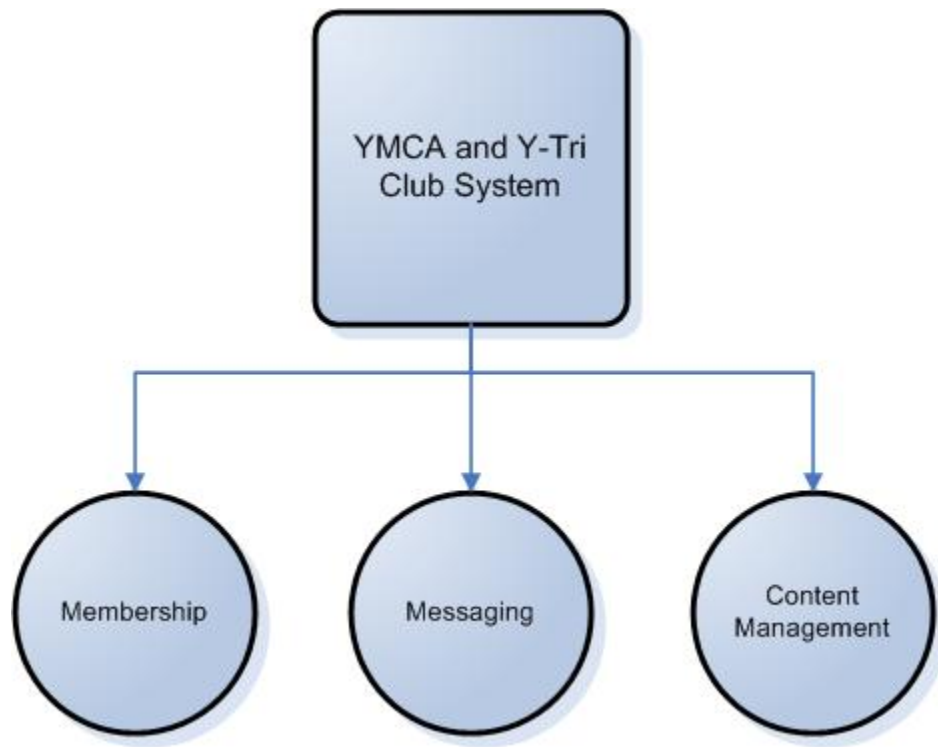
This section will describe in more detail the three key enhancements to the YMCA communication system.

### **System Design Overview**

The new system requirements (membership, messaging, content management) are built off the existing database and web system of the YMCA and Y-Tri Club that was created in part by myself during the internship with StepQuest. The majority of the system however, was previously created by Charles Laymon of StepQuest, Inc.

### **Membership Capabilities Overview**

- Membership Capabilities
  - Security
    - *Authentication* – The user must prove who they are by submitting a set of credentials such as a user id and a password. Once they have proven to be a valid user, they will be admitted into the system..
    - *Encryption* – Sensitive user information being transmitted from the user to the server must be encrypted. 128 bit SSL encryption will be used to ensure that the user information exchanged is safe.
    - *Authorization* – Once a user is authenticated, the application must determine user permissions via role based security. This is key to know which users may modify or access what data in the system. In effect roles are assigned with specific permissions.



*Figure 1.1 – The new system architecture*

- Activation – To provide the YMCA management with member accountability, all members are initially loaded into the system by YMCA management. A member of the YMCA should be able to activate their account with the site. Only YMCA or Y-Tri Club members will have an account in the system pending activation. After activation the users will gain access to members only content on the web sites.
- User Profile – A user should be able to access their profile that contains their personal information. Furthermore, the user needs to be given the permission to change their information if they wish.
  - Privacy information
    - Allow other members to view personal information such as contact information

- Contact information
  - Address
  - Telephone
  - Email Address
- Change messaging preference
  - Opt in/out of receiving messages from management or other members
  - Subscribe/unsubscribe to messaging channels

### **Messaging System Overview**

- Messaging System
  - Event Announcements – Allow for better communication to registered members of upcoming events within the YMCA/Y-Tri Club community
  - Messages from the CEO – Inform members of the YMCA of any critical information and updates to programs or member services (e.g. a membership promotion)
  - Community Communication – Allow the YMCA community to better communicate with each other and organize events among the membership.

Specifically the messaging system is only available to registered members. When a user activates their account, they are presented with the option of choosing to 'opt-in' and receive messages from the messaging system. A user who composes a message will select the recipients. The recipients of the message will be the subscribers of the channels addressed by the sender (i.e. bicycle interest channel). When the user posts the message, an email with the message is sent to the channel subscribers. Attachments may also be submitted with the message. Channel

subscribers will receive the message at their primary email address, which is selected in the user's profile. Every user will have a primary email address. Users will be able to reply directly to the original message sender only.

### **Limited Web Content Management System Overview**

Privileges to the limited web content management system are only given to administrators. Overall this system enables administrators to modify the existing web site without knowing detailed HTML instructions. For the limited content management system, I have implemented a third party HTML editor that StepQuest, Inc. elected to use. A limited content management system is defined in part by the following criteria:

- Content editing – Administrators will be able to modify content directly from the user interface
- Version control – For YMCA purposes, previous versions of the edited page will not be saved for future retrieval
- Workflow – Updates do not have to be approved by anyone but the administrator making the updates.

A challenge faced by many website owners is keeping the content current. Periodically, the YMCA and Y-Tri Club have updates to their site's content. A limited content management solution allows the Y-staff to make real time updates as new content becomes available. The limited content management system provides the administrator with a list of HTML pages that they can edit. If they have a page they need to edit, they will be able to select it from the list and edit the page and submit the newly updated page to the database. Administrators may also navigate to the pages that need updating and will be presented with the HTML editor on the page.

## **Implementation Requirements**

To implement the three major system components (membership, messaging, limited content messaging system) several preliminary requirements must be addressed.

Refer to Appendix E for the database modifications that were necessary.

- Membership
  - Appropriate modifications to the database tables must to be made
    - Fields for account activation and deactivation
    - User role information
    - Fields to hold password information
      - Password
      - Security question
      - Security question answer
  - The column which holds the member's password information must be encrypted
  - Additional stored procedures must be written
    - Activate and deactivate a member account
    - Change and recover passwords
    - Manage user roles
    - Update, create, delete contact point types
    - Change primary contact point
- Messaging System
  - New tables must be added to the database
    - The main table which holds the messaging notification channel information
    - The table which defines the types of channels

- The table which holds user channel subscriptions. It relates to the another 'person' table, which is the main table for user information
  - New stored procedures must be written
    - Stored procedures for which administrators can manage channel information
      - Change channel display attribute
      - Change channel active attribute
      - Create a new channels
      - Update the name of a channel
      - Update the descriptions of a channel
    - Stored procedures for member subscriptions
      - Add a subscription
      - Remove a subscription
    - Stored procedures for managing messaging types (i.e. Email types)
  - Implement "NotificationEmail" class previously written by Charles Laymon of StepQuest, Inc. to handle message operations
    - Attachments
    - SMTP
- Limited Content Management System
  - Investigate an HTML editor (purchased or freeware) to update static web HTML content

### **Technical Approach**

This project was developed in the Microsoft .NET 2.0 Framework. It was programmed in the C# language and ASP.NET, emphasizing strong object oriented

design. Details of alternate development platforms and languages will be explored in the "System Analysis" portion of the paper.

## N-tier Development Pattern

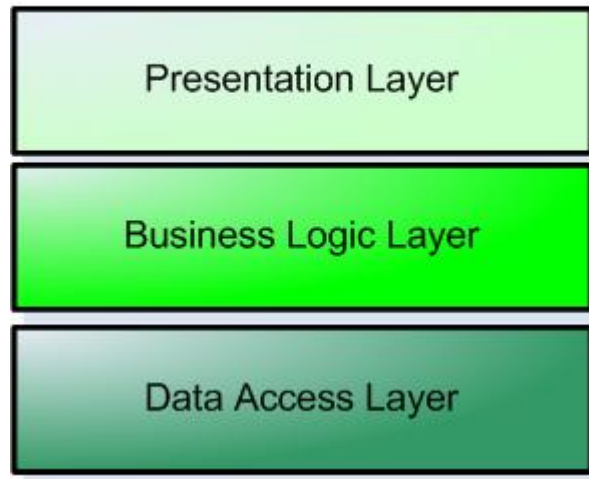


Figure 1.2 – Tiers utilized in the n-tier development pattern

The new system was implemented with an n-tier development approach as shown in Figure 1-2. The tiers were as follows:

- Presentation layer
  - *Style sheets and themes* – The consistent look and feel of the new system is accomplished through cascading style sheets (CSS) and themes.
  - *User friendly navigation* – The consistency of the pages in the system provide for user friendly navigation among pages for both administrators and non-administrators. This is accomplished through navigation bars and user controls for signing in or out that persist on all pages.

- *Security* – A portion of the security measures are exercised in the presentation layer. The navigation bar menus are enabled or disabled based on the roles the user is in (See Appendix D).
  - If a user (non-administrator) is on the system and not logged in, they will *not* be presented with a link to a profile or to search for other members.
  - If a non-administrator is logged in they will only be presented with links to their profile and the member directory
  - An administrator who is logged in will be presented with all links a non-administrator would see, and menus for administrative functions:
    - Manage member roles
    - Manage messaging channels
    - Manage content
- *Accessibility and standards* – The new Wilmington Family YMCA and Y-Tri Club system is built on the XHTML standard. The W3C defines XHTML compliant pages as [1]:
  - “Documents that are XML conforming and are readily viewed, edited, and validated with XML tools”
  - “Documents can utilize applications (i.e. scripts and applets) that rely on either the HTML or XML Document Object Models”

The new system complies with Priority One of the W3C Web Content Accessibility Guidelines. Some of the guidelines for the W3C WCAG P1 are as followed [2]:

  - If images display important information, make sure it is represented in another way

- If a certain color of text is used to represent important information, make sure it is represented another way
  - If style sheets are ignored by the client, make sure content can still be read
  - Provide alternative for information that is conveyed in scripts or applets
  - Ensure all images contain the "alt" attribute
- Business logic layer
  - *Microsoft .NET* – The new Wilmington Family YMCA and Y-Tri Club system was developed with Microsoft .NET 2.0 Framework using Microsoft Visual Studio 2005 (analysis of other development platforms considered in the "Systems Analysis" section). The Microsoft .NET 2.0 Framework was a requirement of StepQuest, Inc. It provided for seamless integration with Microsoft SQL Server Express Edition, rapid application development and debugging support.
  - *C# programming language* – For the development language in the business logic layer, C# (a programming language developed by Microsoft) was utilized. C# was chosen for its strongly-typed capabilities. C# provides tighter type control because unlike VB.NET, it does not allow developers to implement undeclared types. Also C# does not allow late binding without explicitly using Reflection. This makes for a far more controlled development environment where the compiler can be far more effective in catching compile time errors.
  - *Design patterns*
    - Page Cache – The site implemented an overall page caching pattern by utilizing the "MasterPage", so that pages

implementing the “MasterPage” will not be allowed to cache their content and will always post to the server when a request is made [10].

- Intercepting Filter – In order for the application to move users seamlessly between Secured (HTTPS) and Unsecured (HTTP) pages the application exploits the HTTP pipeline to intercept http requests and make determination as to whether the request is for a secured or non-secured resource and format the users request accordingly without unnecessary client messages [10].
  - Adapter – Data access is performed through an Adapter Pattern utilizing “TableAdapters” to perform functions of Insert, Update, Delete, and Select on specified tables. This functionality essentially abstracts away the complexities of making such updates to a database table (or set of tables) and allows business tier logic to “adapt” to the data tier logic [10].
  - Layered Application – N-Tiered architecture was used to create a logical separation between presentation, business, and data access logic [10].
- *Security* – Security measures were taken in the business logic layer to protect the system from unauthorized access. Every page and user control in the new system that requires membership (e.g. profiles, directories, administrative pages) verifies that:
    - The user is logged in
    - The user is in the appropriate role to view the page
- If a user who is trying to access a page fails one security check, they are denied access to the page/user control. With addition to checking

roles on the page, FormsAuth (Forms Authentication) is also implemented in the Web.Config file. FormsAuth allows for extra security by specifying pages of the system and what role is allowed to access that page.

- Data access layer
  - *SQL 2005 Express* – The database for the project was designed and developed with Microsoft SQL 2005 Express Edition. Alternatives will be explored in the "System Analysis" section of the paper.
  - *Database design* – The design of the database developed for the new system was based on the relational database model. The database is in third normal form.
  - *Security* – Several security measures were involved on the data access layer:
    - Encryption – Database fields which hold user passwords are encrypted
    - Stored procedures – Stored procedures were used for all Transact-SQL statements. By encapsulating processes, stored procedures hide the complexity of the database. Furthermore stored procedures allow for parameters thus protecting against SQL injections [3]. Additionally, stored procedures allow us to assign permissions to the stored procedures and not directly to the tables. This prevents situations where a user could launch an application, like Microsoft Access, and look directly at the tables (assuming they had access to the database and database server).

# Systems Analysis

## Methodology

There are many different system development methodologies. The methodology for system development used for this project was the Unified Process. The Unified Process (UP) was determined to be the best fit for the Wilmington Family YMCA and Y-Tri Club community development project because the project was to be broken into phases. With the UP, I was able to define unique disciplines, iterations, and phases that specifically pertain to the project. Shown in Figure 1.3 is the an overview of the Unified Process Model.

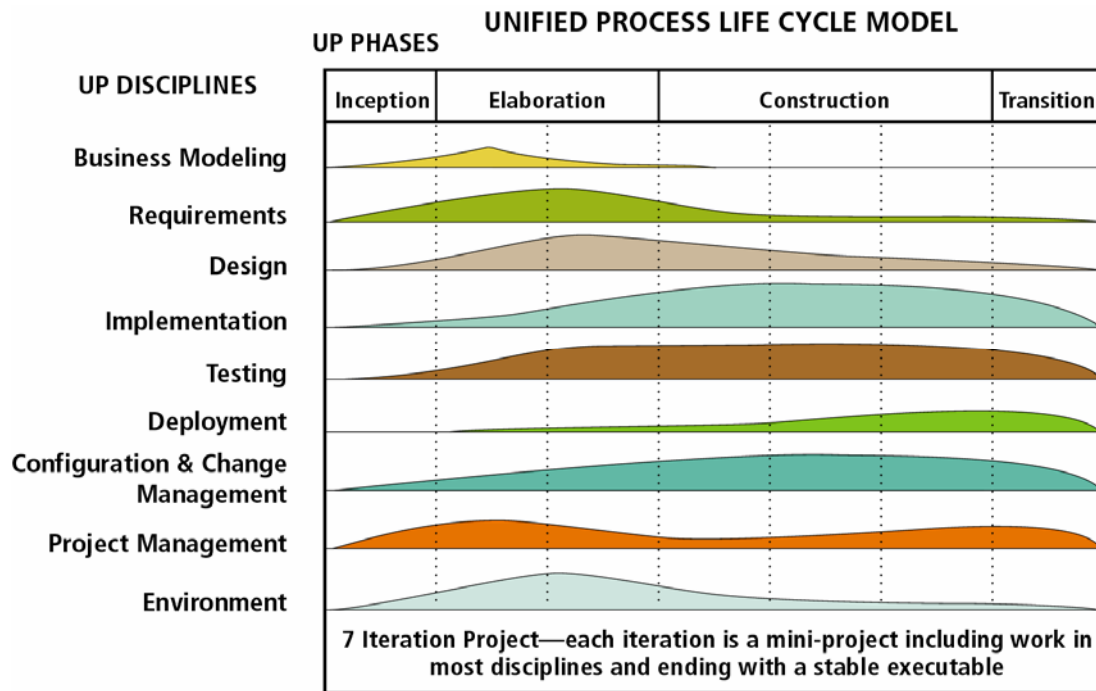


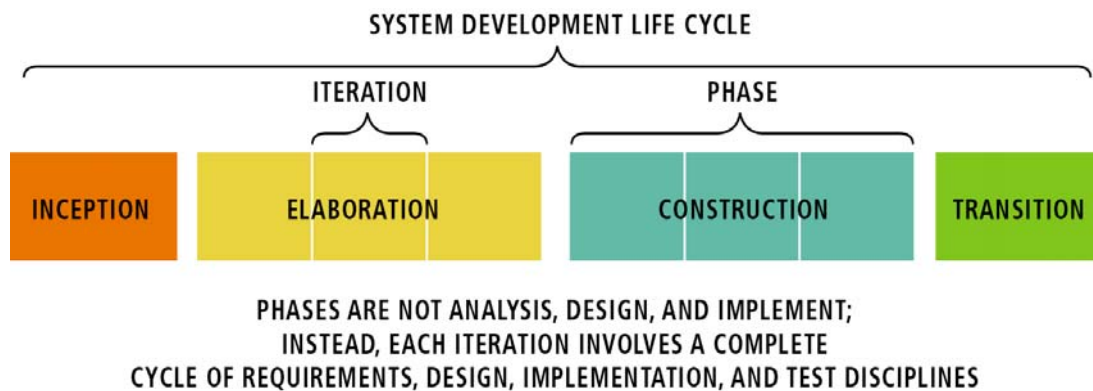
Figure 1.3 – Sample of the UP Life Cycle showing disciplines [4]

The key concept in the UP is the idea of iterations within phases. “Iterations are system development process in which work activities – analysis, design,

implementation – are done once, then again, and yet again on different system components; they are repeated until the system is closer to what is ultimately needed [4].” The UP consists of four phases:

- Inception
- Elaboration
- Construction
- Transition [4]

Within each phase are iterations involving analysis, design, and implementation for part of the system as shown in Figure 1.4 [4].



*Figure 1.4 – The Unified Process Life Cycle [4]*

The UP is designed for large systems projects and may be modified for smaller products. As this project did not involve many different professionals I used a modified model of the following UP disciplines:

- Requirements gathering
- Design
- Implementation
- Testing

Deployment for the project, I defined 5 iterations (of the above disciplines), one iteration for every two months of the project life cycle; August 2006 to May 2007. The phases of the UP for this project are detailed in Figure 1.5.

<b>UP Iterations</b>					
<ul style="list-style-type: none"> <li>• Merge data into one database</li> <li>• Implement two systems under one code base</li> <li>• Gather new requirements</li> </ul>	<ul style="list-style-type: none"> <li>• Understand the business environment</li> <li>• Create the system vision</li> <li>• Create business models</li> </ul>	<ul style="list-style-type: none"> <li>• Membership analysis</li> <li>• Membership development</li> <li>• Informal testing</li> </ul>	<ul style="list-style-type: none"> <li>• Messaging analysis</li> <li>• Messaging construction</li> <li>• Web content management analysis</li> <li>• Web content management development</li> <li>• Informal testing</li> </ul>	<ul style="list-style-type: none"> <li>• Modify UI</li> <li>• Secure communications</li> <li>• Proper error handling</li> <li>• Formally test</li> </ul>	<ul style="list-style-type: none"> <li>• Formally test</li> <li>• Move system to production</li> <li>• Send invitations to all YMCA members</li> </ul>
<b>Summer 2006</b>	<b>First Iteration</b> <b>Inception Phase</b>	<b>Second Iteration</b> <b>Elaboration Phase</b>	<b>Third Iteration</b> <b>Elaboration Phase</b>	<b>Fourth Iteration</b> <b>Construction Phase</b>	<b>Fifth Iteration</b> <b>Transition Phase</b>

Figure 1.5 – Events which took place during each phase.

### **Inception Phase**

The inception phase of the project is the first phase. Several key processes are completed during this phase.

- Understand the business environment [4]
  - Describe the problem or need facing the Wilmington Family YMCA and Y-Tri Club
  - Evaluate existing architecture
  - Determine stakeholders – Define managers, officers and number of community members
- Create the system vision – Define, develop, and refine the vision for the new system [4]

- Create detailed plans to show how the new system would improve current operations and solve existing problems
  - Inconsistent and inaccurate information
  - No convenient way to make updates in house
  - No communication mechanism
  - Little to no community collaboration
  - Lack of community development
- Define benefits [4]
- Create a list of system objectives [4]
- Create business models – Develop detailed use cases for each of the three main project deliverables [4]
  - Membership use cases
  - Messaging system use cases
  - Limited content management use cases

Additionally, the inception phase outlines the scope of the project and makes a clear definition of what the project will accomplish [4]. With a project of this size, it is important to outline the scope, such as to be aware of a scope creep.

### **Interviews**

Interviews were conducted with key personnel affiliated with the YMCA and Y-Tri Club. To gain a better perspective of system requirements and design, different stakeholders (managers and members) were interviewed. An overview of the questions and answers from the interviews conducted may be found in Appendix A.

The following stakeholders were interviewed:

- Dick Jones – YMCA CEO: Dick Jones has been the CEO of the YMCA since 2003. He directs the daily operations of the YMCA and provides

administrative oversight and management of 12 full time employees and more than 130 part time employees.

- Shannon Berg – Member Services Director: Shannon Berg serves as the YMCA Member Services Director and is the founder and liaison of the Y-Tri Club.
- Lorie Lucas – YMCA Member: Lorie Lucas has been a member of the YMCA for several years. She and her children are active users at the YMCA and participates regularly in athletics and classes offered.

### **Problem Definition**

After the interviews this section identifies the problems as defined by the stakeholders, in addition an early goal statement was developed for the project. This was:

***The non-profit organizations of the Wilmington Family YMCA and Y-Tri Club require a web application upgrade that will be a driving force in increasing communications for all stakeholders.***

### **Background**

Currently the Wilmington Family YMCA and Y-Tri Clubs web sites coexist under the same code base and reside in the same database. Current Y-related information needs are met by these two sites. There is currently no convenient membership or content management needs being met at the present.

### **Audience**

The audience that will benefit from this new service consists of the following:

- YMCA Management Personnel

- Dick Jones: **YMCA CEO** – Dick Jones will be able to keep the YMCA members informed up important news, events, and the future goals of the organization.
- Shannon Berg: **Member Services Manager** – Shannon Berg will benefit by keeping members informed of important content updates via a limited content manager and messaging system.
- **YMCA Departments** – Quickly get important updates to the website and via the messaging system, communicate with members.
- **YMCA members** – Better communication will be provided via the messaging system. Easy organization of practices will be provided as well as shared workout and nutrition tips. Members will be able to self-form practice teams with other members.
- **Y-Tri Club officers** – Keep members informed of important updates and events. Update and manage content through a limited content management system.
- **Y-Tri Club coaches** – Communicate with members and organize practices
- **Y-Tri Club members** – Communicate with one another and share workout tips

Figures 1.5 and 1.6 detail the management structure of the YMCA and Tri-Club Organization.

- The Wilmington Family YMCA consists of many departments, each department containing many officers and coaches..
- The YMCA currently has over 5,000 members in the community
- The Y-Tri Club consists of several officers and many coaches
- The Y-Tri Club currently has over 100 members in the community

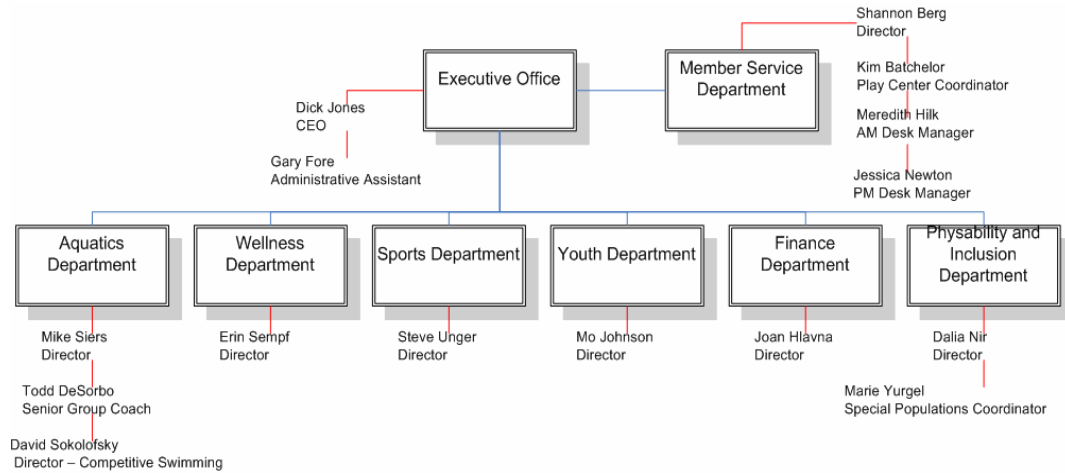


Figure 1.5 – the YMCA Organization

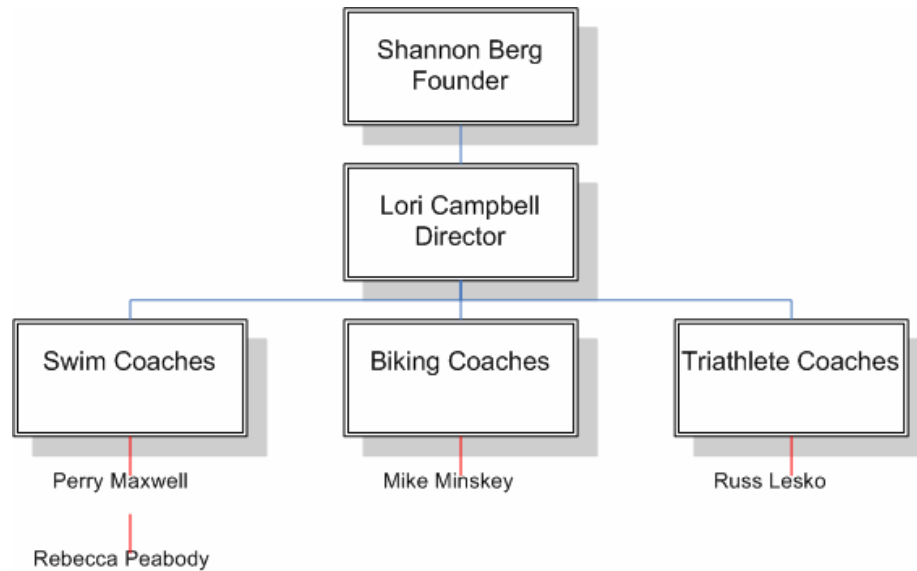


Figure 1.6 – the Y-Tri Club Organization

Current Problems

- Keeping web content updated – The problems both organizations (YMCA and Y-Tri Club) face with the current system is limited means to keep content updated without going through a third party. Therefore it is important for the managers of both organizations to keep content updated.

- Untargeted communication between members – Current communications from the club directors to YMCA and Y-Tri Club members are implemented via email or in the case of the YMCA, via print media. While this has been somewhat effective in the past, problems exist where users cannot get information targeted to their needs. Also, there is no current way for members to form special topic/inter community communications (e.g. running) to other interested members. Phone, word of mouth, and emails only reach so many people. They do not reach all parties interested.

## **System Objectives**

### Overview

The system objectives for the Wilmington Family YMCA and Y-Tri Club are to:

- Provide an integrated and seamless system both organizations can use to increase service and value to members of the community
- Allow members to have an account on the new system
- Allow members and managers to message one another
- Implement a limited content management system for YMCA and Y-Tri Club managers and officers

### Membership

The overall objectives for the membership elaboration phase of the project are as followed:

- Allow Wilmington Family YMCA and Y-Tri Club members to have a personal account
- Allow these members to have access to premium web information
- Allow members to have and share personal contact information in their profile

- Phone information
- Email information
- Address information
- Provide members an option to share information with other members
- Allow members to search for other members with activated accounts

### Messaging System

The overall objectives for the messaging system phase are as followed:

- Allow members to subscribe to “messaging channels” of personal interest, such as:
  - Running channel
  - Biking channel
  - Nutrition/Diet channel
- Provide a vehicle in which members can message one another with common interests
- Allow members to send attachments in messages
  - Attachments must meet certain size and file type requirements
- Provide an option for members to choose whether they are to receive messages or not (opt in or opt out)

### Limited Web Content Management System

The limited content management system objectives are:

- Provide both organizations with a tool that will enable them to easily manage YMCA and Y-Tri Club content without having to go through a third party
- Integrate as implemented an off-the-shelf limited content management suite that meets the needs of the YMCA and Y-Tri Club
  - Freeware

- Purchase
- Provide stakeholders with current and accurate information

### **Elaboration Phase**

The second phase, the elaboration phase, is the biggest phase during the UP. Because major project components are developed during the elaboration phase, it consumed the majority of iterations (months) of the project. The elaboration phase was an opportunity to further define requirements, schedules, and design of the entire project as programming on the three key deliverables commenced [4].

A major reason for selecting the UP methodology was because it is a more adaptive approach to system analysis than the alternatives (detailed later). Because of this, I was able to further break down the elaboration phase to better define each major component individually. Each “sub phase” was completed in order of importance, having membership the most important because messaging and content management relies on the completion of the membership phase. The subsequent phases (messaging then content management respectively) did not begin until the one before it was complete.

Membership phase – The membership phase was the first sub phase of the elaboration phase to begin. It was the most important of the three phases as it was the backbone of the new system requirements. Please refer to Appendix C for the tasks which were accomplished during the membership elaboration sub phase.

Messaging System phase – The messaging phase was the second phase completed in the project life cycle. Please refer to Appendix C for the tasks which were accomplished during the messaging elaboration sub phase.

Limited Web Content Management phase – The limited web content management phase was the final phase of the project life cycle. Please refer to Appendix C for detailed steps taken during this phase.

### **Use Case Diagrams**

Please refer to Appendix B to view the detailed use case diagrams relating to:

- Messaging system functions
- Account activation
- Member functions
- User profile functions
  - Setting profile privacy
  - Setting messaging preference (opt in/out)
  - Inserting/Editing/Deleting contact point types
    - Address
    - Phone
    - Email
- Member directory
- Role management
- Web content management

### **Construction Phase**

The third phase in the UP is the construction phase. After the main foundation of the new system was put into place during the elaboration phase, the construction phase was an opportunity to accomplish the following tasks:

- Clean up and modify the user interface with the addition of technologies such as Asynchronous JavaScript and XML (Ajax)
- Provide for proper error handling

- Secure member communications with SSL encryption
- Formally test the system with Nunit testing
- Prepare to move the system to production
- Develop page that sends out account activation initiations to all members via Email

Modification of the User Interface (UI) – The priorities of the previous phase, the elaboration phase, was to simply get the system working correctly. Emphasis was placed on functionality and correctness as opposed to how the components were presented in the UI. During the construction phase, however, actions were taken to improve the look of the UI and improve the experience the user has with the system. Such actions that were taken were:

The system also incorporates the Microsoft .NET Ajax components into the UI. With the use of Ajax, the user experience is greatly improved:

- Pages become more sophisticated
- Interactions with the system by the user are more responsive and interactive
- Client-server communications are more effective
- Positioning pointers are maintained so that the user does not lose context of where they are at on the page

Ajax was used to upgrade the UI that was initially developed in the elaboration phase in several areas:

- Membership profile – Ajax upgraded the profile by grouping the various contact point types into Microsoft Ajax Control Tabs. This provided better organization of the controls which were previously presented side by side with no separation. The key benefit to the user is easier navigation.

- Administration pages:

Role Management – Allow two functions, manage roles and assign user roles, to exist on the same page separated by Ajax tabs

Channel/Messages Management – Allow two functions, managing channels (hidden from non-administrators) and message composition, to exist on the same page separated by Ajax tabs

Content Management – Provide a page for which administrators could manage YMCA and Y-Tri Club related content

- Provide edit buttons on the “Offering Content” which will easily take administrators to the limited content manager control for editing
- Enable links in the “Membership” menu item based on the login status and on the member’s roles. Displaying these links in the menu item provides for easier navigation to the system’s pages for the user.
- Display links to the directory, the user’s profile, manage roles, channels, and offerings in the “Membership” menu item if the user logged in is an administrator.
- Display links to the directory and the user’s profile if the user logged in is not an administrator

### Error Handling

In the elaboration phase, errors were not handled in an acceptable manner for a production grade system. Sometimes errors were caught and ignored. Other times error handling was ignored all together. The main idea of the elaboration phase was to get the foundation of the system functional. During the construction phase, however, proper error handling was completed.

- Error handling was set up properly in the Web.config file (a XML file that maintains the specific configurations of this web system)

- The Application\_Error method was setup in the Global.cs class (available to all classes in ASP.net)
  - Receives the last error message generated, the inner exception, and the source. Then the Application\_Error method redirects to a custom error page (Error.aspx) which informs the user an error has occurred and provides a link to return to the home page, hiding the error message from them and a potential code leak.
  - Error.aspx logs error content to the "Event Viewer." This will allow the developers, to know what happened, where, and why. This in turn makes debugging the problem much easier.
  
- Secure Communications – In phase two, no measures were taken place to secure user communications of sensitive data across the wire. User communications with the server should to be protected. The construction phase was the first in which steps were taken to secure communications. The steps taken to secure communications in the construction phase will not be the final steps, as the certificates used were not purchased from a trusted third party (such as GoDaddy).
  - The certificates used for development and testing purposes were created by "SelfCert," a Windows program in which you can create your own certificates without cost. As previously mentioned, to insure our clients are well protected, StepQuest, Inc. is going to purchase a certificate from a trusted certificate authority (CA) for implementation in the final (implementation) phase.
  - In order for the application to move users seamlessly between Secured (HTTPS) and Unsecured (HTTP) pages the application exploits the HTTP pipeline to intercept http requests and make determination as to

whether the request is for a secured or non-secured resource and format the users request accordingly without unnecessary client messages.

- Formal Testing with Nunit – In the previous phase, informal testing was conducted which consisted of personally logging into the system and testing all functions to verify everything worked as desired. In the construction phase more formal testing with Nunit was conducted. “Nunit is a unit-testing framework for all .NET languages [5].” “The goal of Unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy [6].”
- Activation Email Invitations – A list of users had to be obtained from the YMCA management. This list would contain the names and email addresses of all Wilmington YMCA members. Scripts must be constructed to take these members and put them in the database (activation field is initially set to false). A page was then constructed that would email all members with a link to activate their account. See Appendix D for screen shots of the “Activation” page. Invitations were not sent in the construction phase, they will be sent in the transition phase of the UP.

### **Transition Phase**

The transition phase of the UP is the final phase of the process [4]. This process consisted of moving the system and database from development to production. A final step of this phase is that invitations were sent to members to activate their accounts. In addition SSL Certificates were obtained and the key users were trained on the use of the system.

## Alternate Analysis Methodologies

Alternate analysis methodologies were known and were considered before the initial phase began. Many analysis methodologies exist; however, there were only two additional alternate mythologies that were deemed appropriate for this system. The two alternate approaches are: the spiral model and the Waterfall model. In the following section I will detail each one and explain why choosing the Unified Process was the right choice.

### Spiral Model

“The spiral model, like the UP, is an adaptive SDLC (System Development Life Cycle) approach meaning the project activities, including deliverables and plans, are adjusted as the project progresses [4].” The spiral model shown in Figure 1.8 emphasizes developing prototypes of a system component in iterations.

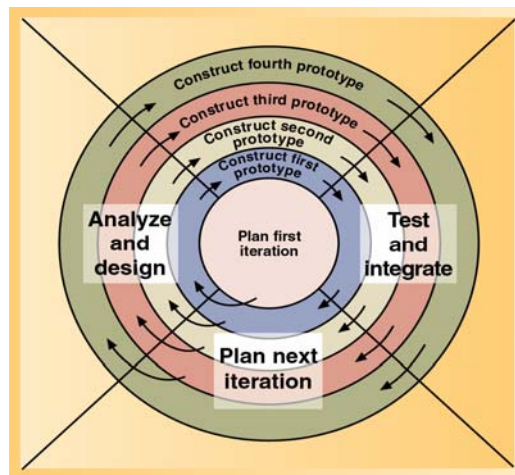


Figure 1.8 – The spiral life cycle model [4]

As previously mentioned: “An iteration is a system development process in which work activities – analysis, design, implementation – are done once, then again, and yet again on different system components [4].” A prototype is a working model of a larger, more complex system. A prototype should be:

- *Working* – It should be functional and represent the functionality of the final system
- *Focused* – It should be focused on one component of the final system. In my case, the three components would be membership, messaging, and content management.
- *Quick* – The prototype should be built and modified quickly to allow testing and demonstration [7]

The idea of developing a prototype in iterations is to quickly develop a preliminary model of a component and demonstrate it to the clients to get their input on whether it is acceptable or not. If not, the developer(s) must then start a new iteration and quickly develop another prototype and present the new version to the customer for approval.

- *Advantages of Prototypes:*
  - Provide clients with a detailed idea of how the final system will look and work
  - Provide for open and constant communication between the developer(s) and clients
  - Increased system development speed
  - System is easier to understand and learn
- *Disadvantages of Prototypes:*
  - May cause systems to remain unfinished
  - Possibility of implementing systems before they are finished
  - Loose focus on what is important on the system
  - Could create a scope creep
  - It can be expensive
  - Difficult to control [7]

The spiral model was considered when the requirements of the three components (membership, messaging, content management) were defined by the clients. However, a developer or development team should use the spiral model when requirements are not fully understood or are ambiguous. In the case with the Wilmington Family YMCA and Y-Tri Club, requirements for their three new components were not ambiguous and were understood and easily defined. Development would have been considerably slower and the quality of the final product would have been less than it is if this methodology was considered. Since I had detailed requirements for all three components, I was able to choose a methodology which would allow me to be thorough in each phase of the software development life cycle, which the Unified Process allowed me to do.

### Waterfall

Another SDLC approach considered was the waterfall model or “top-down” approach [4]. The waterfall model shown in Figure 1.9 is a very basic and traditional SDLC model. “The waterfall approach assumes that the various phases of a project (planning, analysis, design, implementation, and support) can be carried out and completed entirely sequentially [4].” In the waterfall when a development phase is completed, there is no going back [4].

- Advantages of the Waterfall Model:
  - Better understanding of the requirements [8]
  - Defined methods for requirements gathering [8]
- Disadvantages of the Waterfall Model:
  - Once a phase is exited, there is no revisiting it
  - Developers must be error free

- o Less focus on risk than a model such as the Spiral Model [8]

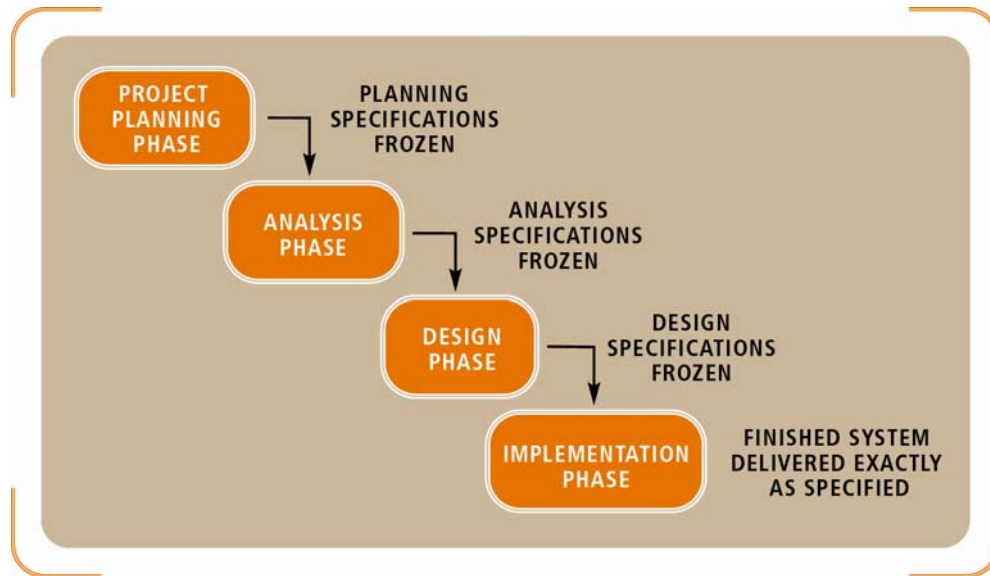


Figure 1.9 – The waterfall software development lifecycle [4]

### Technical Specifications / Requirements

Many different technologies were utilized to construct this new system. The following will detail each technology used for implementing:

- The project as a whole
- The membership requirement
- The messaging system requirement
- The limited web content management system

Additionally, I will provide possible alternative technologies for implementation considered for each requirement during the analysis process.

## Software and Programming Languages

- Microsoft .NET 2.0 Framework

This project was built in the Microsoft .NET 2.0 Framework with Microsoft Visual Studio 2005, using the C# language. The Microsoft .NET 2.0 Framework with C# was chosen for several reasons:

- It was a StepQuest, Inc. requirement (the consulting firm)
- It provides for rapid application development
- It works well with Microsoft SQL Server 2005 Express, another client requirement

### VB.NET vs. C#

Other languages native to ASP.NET were considered for implementation such as using VB.NET in the .NET 2.0 Framework. VB.NET and C# utilize the same libraries in the .NET 2.0 Framework and can interface with Microsoft SQL Server 2005 Express. I selected C# as I was more familiar with the C++ language of which C# is a close relative. This allows C# to provide tighter type control. Additionally, C# does not allow late binding without explicitly using Reflection, which makes for a far more controlled development environment where the compiler can be more effective in catching compile time errors.

### PHP

PHP was an option for implementation; however, it was not chosen to implement the system for several reasons:

- Using .NET was a requirement from Charles Laymon of StepQuest, Inc., everything Charles Laymon implements and supports is written in .NET

- .NET is superior in a broader reach of application types (e.g. Windows forms, mobile, web services, etc.)
- Microsoft SQL Server 2005 Express

The database which holds all YMCA related information (member accounts and content) was developed with Microsoft SQL Server 2005 Express. Factors which resulted in the use of Microsoft SQL Server 2005 Express are:

  - It was a requirement of Charles Laymon of StepQuest, Inc.
  - It is free. The YMCA and Y-Tri Club are non-profit organizations, purchasing a license for database server applications is not feasible for what they are going to use it for.
  - It provides seamless integration with Microsoft Visual Studio 2005 and the Microsoft .NET 2.0 Framework.

#### Microsoft SQL Server 2005

Other database management software were considered for the new system, one of which was Microsoft SQL Server 2005. Microsoft SQL Server 2005 provides the same services and more than Express Edition. However, a license for the full SQL Server 2005 implementation is expensive, and in the case of the YMCA, unnecessary. For example, the full implementation of SQL Server 2005 provides for more storage room than Express Edition provides. However, the amount of storage Express Edition supports is sufficient for the YMCA and Y-Tri Club.

#### MySQL

MySQL was not an option as it was not a requirement of Charles Laymon of StepQuest, Inc. Microsoft Visual Studio 2005 and the .NET 2.0 Framework works best with Microsoft SQL Server. For example, "TableAdapters" in the

.NET Framework do not integrate with MySQL tables. Since Laymon only uses the Microsoft .NET Framework, Microsoft SQL Server is always chosen to be the database in his projects.

- **Membership Requirement Implementation**

There are many possible ways to implement the membership requirement for this system. ASP.NET 2.0 has a built-in administration tool for assigning membership and identity; there are open source solutions; and there is the option to create a custom membership implementation. For this project, it was elected to create a custom membership by overriding the default role and membership providers in the .NET Framework. Implementing membership in this manner was for the following reasons:

- Roles and passwords for members are stored in and derived from the database (Appendix E)
- Potential future requirements will be best fit by a custom system
- Creating custom role and membership providers worked best with the design patterns used:
  - Adapters, Caching, and n-tier development
  - Interfacing with the database with "TableAdapters"
  - Caching reusable user information such as the user's role. This prevents additional trips to the database.
- By overriding the default role and membership provider methods, it was able to have control over what role and membership methods were initialized and of their operations.
- It was much easier to implement a custom membership system as I was developing off an existing database

- Finally it was implemented this way because Charles Laymon of StepQuest, Inc. wanted it implemented this way so that I could better understand and utilize the way the .NET 2.0 Framework handles role and membership functions.
  - How to implement .NET Framework intrinsic web controls

#### Microsoft ASP.NET 2.0 Membership and Identity Administrative Tool

As previously mentioned, there are alternatives that could be used when implementing membership and roles on a web application. First, Microsoft ASP.NET 2.0 contains a built-in membership and identity administration tool. Unless otherwise specified in the Web.Config file the system will automatically verify login information and role information. As this implementation provides users with an organized and easy to read UI, using this implementation was impractical for the new YMCA and Y-Tri Club system for several reasons:

- The administration tool doesn't easily support the multiple roles the system needs for one particular user
- This tool is primarily for managing smaller groups of people. The YMCA and Y-Tri Club have a combined membership of over 5,000 members
- This implementation would be less flexible for future requirements

#### Non-Profit Membership Software

There exists software designed specifically for non-profit organizations to manage membership. Such membership tools can be found under the QuickBooks Solutions Marketplace. The non-profit membership solutions found here, such as "One Vision – Church Membership Software" allows organizations to keep track of their members. The solutions offered under the QuickBooks Solutions Marketplace can be useful under certain circumstances. For example, it would work well when building a

system around a non-profit membership solution. They do not work well however, when you try to make one of these solutions fit and work in an existing system, which has more needs than tracking members and their dues payments. Additional reasons these solutions will not work well with the new system are:

- Lack of support for required membership roles
- Expansion restrictions. Future requirements will not be supported well by these solutions.

### **Messaging System Implementation**

As it was with both membership and web content management, several options were available in choosing how the messaging system requirement was to be implemented. The following options were available:

- Develop a new messaging system
- Obtain a class which was previously written by Charles Laymon of StepQuest, Inc. that provides SMTP functions
- Find a solution: Open source, freeware, or purchase

For this implementation I chose to utilize the previously written messaging class done by Charles Laymon. Several factors led to this decision.

- Time requirements. The system had to be done in one year. It would have taken much more time to develop an entire messaging system. Additionally, much time would have been devoted trying to implement a solution from an outside source (open source, freeware, or purchased solution).
- It has been utilized in previous projects by Charles Laymon of StepQuest, Inc. Therefore, there were no questions on whether if it would work correctly and I will not be supporting the system in the future and Charles Laymon will be. It

is better to utilize the SMTP class he created so he can better support the system in future additions to the system.

The class consisted of methods for handling attachments, and sending emails. I had to develop the UI to interface with this class (Appendix D for screenshots). The UI consisted of a list of checkboxes for the messaging channels. The messaging channels are what users subscribe to (i.e. with a "Running" channel subscription, you will receive all messages related to running). The UI contained a "Subject" textbox, a text area for the body of the message and a file upload control.

### *Developing a New Messaging System*

My initial plans were to develop the messaging system from ground up due to the unique requirements of the system (e.g. messaging channels). However, with the addition of the class obtained from Charles Laymon, developing the entire system became unnecessary. Only developing the UI and the business logic layer of the UI functions to interface with the messaging class was needed. Time constraints also dictated that I not build the system from ground up. This benefited the project for the following reasons:

- It was exactly what was needed for handling email attachments and sending emails
- Time was saved from not having to develop that class
- The time freed up was better spent on testing and ensuring all client requirements were being met
- Reusability for Charles Laymon

### Finding a Solution in Freeware/Open Source/Purchasing

It was difficult to find a solution to meet the unique client requirements for the messaging system. The following reasons detail why I did not pursue finding a third-party solution:

- It would be possible to incorporate the messaging channels with freeware or open source solutions. In this case however, it was much quicker to use the solution handed down from Charles Laymon.
- Purchasing a solution was not feasible as it is a non-profit organization
- The time saved by purchasing a solution is not justified by the prices of the messaging systems

### **Limited Web Content Management Implementation**

Many options were available for the implementation of the web content management system requirement. The web content management system needed to provide management of the YMCA and Y-Tri Club to make real-time changes to web content without having prior knowledge of HTML. Such options for the web content management system included:

- Developing a system in house that would fit the needs of the client
- Finding open source/free software online that fit the needs
- Purchasing a content management program

### Developing a Web Content Management System

Developing a web content management system in house presents several benefits:

- Create a solution that fits the needs of the clients
  - Leave out unnecessary features (and complexity that might confuse the user)

- Make the system simple as possible for unskilled users which would allow for fast training
- Provide a system that will be flexible to support future requirements.

Although developing a limited web content management system poses some benefits to the client, it is impractical as it presents many draw backs.

- Not feasible to create an entire web content management system
- Time requirements are a factor. Since other solutions are available time is better spent working on the functionality of the system to ensure everything works property.
- Creating a HTML editor for the limited web content management requirement could take a long time, and chances are when it is complete it will not be as good as ones you could buy or get for free.

#### *Freeware Solution or Purchasing a Solution*

Since developing the limited web content management system was considered to be impractical for this implementation, looking for an open source solution or purchasing a solution became the course of action. It was ultimately elected to find and implement a freeware solution called "FreeTextBox" (for screenshots refer to Appendix D). By going with this option, the system gave up some features that would have included if purchased solution had been made. The following reasons are why "FreeTextBox" was implemented rather than purchasing a solution such as the "WebHtmlEditor" by Infragistics:

- HTML editors that are for sale are often very pricy. "WebHtmlEditor" can cost from \$795.00 - \$1,290.00
- Come with more features than necessary
- Too complex/high learning curve

For the features and content the YMCA and Y-Tri Club management will be managing, it is not desirable to purchase this type of tool. These tools allow for entire web applications to be built. A solution for managing content on existing pages or creating similar pages is all that is needed. By implementing "FreeTextBox" we do give up some features that would be nice to have (compared to editors such as "WebHtmlEditor" by Infragistics). However, it was chosen for the following reasons:

- It is inexpensive (Free!)
- It is easy to use
- It serves the purpose of editing web content without prior knowledge of HTML

## Implementation

For more details on many of the features in this section please see the screen shots starting in Appendix D.

### Database

As previously discussed many new fields and tables were added to an existing database. For screenshots of relational database tables please refer to Appendix E.

### Design Patterns

Use of design patterns permit easier implementation and more importantly upgrades to the existing system when needed. Examples of design patterns used are:

- Page Cache – We implemented an overall page caching pattern by utilizing the “MasterPage”, so that pages implementing the “MasterPage” will not be allowed to cache their content and will always post to the server when a request is made [10].
- Intercepting Filter – In order for the application to move users seamlessly between Secured (HTTPS) and Unsecured (HTTP) pages the application exploits the HTTP pipeline to intercept http requests and make determination as to whether the request is for a secured or non-secured resource and format the users request accordingly without unnecessary client messages [10].
- Adapter – Add data access is performed through an Adapter Pattern utilizing “TableAdapters” to perform functions of Insert, Update, Delete, and Select on specified tables [10]
- Layered Application – N-Tiered architecture was used to create a logical separation between presentation, business, and data access logic [10].

## **Testing**

Two types of tests were conducted on this project: Informal testing, and formal testing.

- Informal Testing – Informal testing was carried out during the elaboration phase of the Unified Process. Informal testing refers to testing without the aid of a tool such as NUnit. Informal testing consisted of the developer and typical users testing all components of the system to ensure:
  - Desired results were given
  - Proper functionality
  - The system was error free
- Formal Testing – Formal testing was carried out primarily during the construction phase of the Unified Process. Formal testing refers to testing with the aid of a unit testing tool such as NUnit.

## **Lessons Learned**

Throughout the year long process of implementing this new community development system for the YMCA and Y-Tri Club, many new important technologies, techniques, skills, and lessons were learned. These lessons learned will help me in future projects by knowing what to do and how to do it. Additionally, and perhaps more importantly, I now know what not to do.

## **New Technologies, Techniques, and Skills Learned**

### Microsoft ASP.NET 2.0

When the internship started with Charles Laymon and StepQuest in the summer of 2006, I knew nothing of creating web applications with Microsoft ASP.NET 2.0 and the C# programming language. Furthermore, I did not know how to involve other useful technologies, such as XML, to better accomplish a task. Valuable advanced ASP.NET techniques learned that will benefit me and future projects include:

- Allowing two different applications to operate under the same code base
- Overriding role and membership provider methods
- Creating an application that will have thousands of users with many different roles
- SMTP operations. Sending emails with attachments
- Incorporating outside tools, such as a content management program or Microsoft Ajax Controls, into a system
- Securing pages from unauthorized access under Secure Authentication
- Persisting state using session variables and cookies
- Creating custom error pages and logging the error in the Event Log to provide for proper and effective debugging

- Knowing how to work with ASP.NET 2.0 controls and how to make them work together
  - “RadioButtonLists” controlling the content presented in a “FormView”
  - “GridViews” controlling the content presented in a “DetailsView”
- Working with master pages and user controls
- Working with SSL certificates
- Exploit the HTTP pipeline to intercept http request and determine if the request is for a secured or non-secured resource and format the users request accordingly without unnecessary client messages.

### Database Development

When the project started, I had little experience in database design, development, stored procedures and best practices. I had not used a database of this size on a project in the past. Database techniques and skills acquired during this project include:

- Proper relational database design
- Supertype and subtype architecture
- Experience with complex stored procedures for all CRUD (Create, Read, Update and Delete) operations
  - Utilizing a try/catch block within a stored procedure
  - Joins
  - Conditional statements
- Utilizing XML data types as stored procedure parameters
- Utilizing stored procedure functions
- Database security
  - Encrypting fields
  - Give table permissions through stored procedures

## N-tier Application Development

As I was not new to the concept of n-tier application development, I was new to n-tier development and best practices in the ASP.NET 2.0 Framework. The ASP.NET 2.0 Framework provides great tools for developing efficient n-tier applications. I was introduced to the following ASP.NET 2.0 n-tier development tools:

- The Data Access Layer (DAL)
- DAL "DataTable" and "TableAdapter" objects
- Increased usage of a business logic layer.

## What Worked Well

- The Unified Process methodology (especially the idea of reviewing system needs and modifying as the process evolves)
- Microsoft ASP.NET 2.0 supported the needs of the project
- Microsoft SQL Server 2005 Express
- Database flexibility
  - Add new membership roles
  - Add new notification channels
- N-tier development
  - DAL objects: TableAdapters and DataTables
  - Stored procedures
- Utilizing "FreeTextBox" rather than programming the web content management requirement

## Areas of Improvement

### Systems Analysis

There are a few aspects of the project which could have been improved. The first area for improvement is the analysis portion of the project. The methodology in which I chose worked great. Even though the requirements were well understood and I experienced success with the Unified Process, some systems analysis processes should have been done earlier in the project life cycle.

- Present the client with detailed use cases before development of a requirement began. This would have made sure that all parties involved understood what the system was to do, and redefine requirements if necessary.
- Conduct formal interviews with system users (YMCA members) earlier. Lorie Lucas was formally interviewed after development had begun. Her suggestions, which were great, may not have been out of the scope of the project if I had interviewed her before development began. I depended a little too much on the consultant (Charles Laymon) early in the project.

### Scope Creep

The second area for improvement is to better identify and stay away from the scope creep, whether the scope creep is the client or myself. Scope creep refers to uncontrolled changes in the project scope which results in the project team drifting away from the original purpose of the project [9]. With all of the technologies learned during this project, it was very easy for me to get sidetracked from the main objective by trying to implement a very neat, but unnecessary feature. For example, I wanted to implement a lot of Ajax features into the system. The system objective did not rely on these features being added however.

One must exercise self discipline to not drift away from the project as it does not help accomplish the main task: the client's requirements. Sometimes, however, I lacked self discipline and Charles Laymon of StepQuest, Inc. had to keep me focused.

It is important to be aware of scope creep during the systems analysis process, especially during interviews. Lorie Lucas, a YMCA member interviewed, gave some great ideas for the system during our interview. Although they were great ideas, they were not part of the requirements for the project, and time should not be devoted to implementing unnecessary features. The interview with Shannon Berg, the YMCA member services director and Y-Tri Club founder, displayed an example of a scope creep. Shannon Berg was very excited about the new system, as were all managers at the YMCA. During the interview, Shannon often drifted away from the main topics of increasing communication and managing content to how she would like the UI to look. While she gave good suggestions of what she wants for future improvements, changing the UI was not in the scope of the project. The way in which I dealt with this is by politely telling her we would focus on her suggestions to the UI at a later date.

## **Conclusion**

My coursework at UNC Wilmington provided for a strong background and guidance in successfully developing the new system for the Wilmington Family YMCA and Y-Tri Club. The following courses played a significant role during the project:

- Database Management Systems – This class provided a great foundation of database management principles, such as proper design and proper use of stored procedures, which I was able to build off of and expand in this project.
- Project Management – Project management introduced a greater concept of time management. The concepts learned in this class came to great use when managing a project of this size.
- Analysis, Modeling, and Design – This class introduced the Unified Process methodology which was used during this project. Additionally, this course helped in creating diagrams which are helpful in conveying system functionality to the client.
- Internship with StepQuest, Inc. – The internship I did with StepQuest, Inc. was perhaps the best learning experience I have had at UNC Wilmington. Everything I learned during the course of the internship assisted in the successful completion of the project, such as the .NET 2.0 Framework, .NET design patterns, and advanced database principles.

## **Future Work**

During the interview process with the YMCA management, Y-Tri Club officers, and members of both organizations, many additional features were mentioned which, at the time of the interview, were out of scope. Many of these suggestions were very good and will be included in future releases. Such suggestions included:

- Provide members the ability to register for classes or sign up for a workout session with an instructor over the internet. Members expressed that it can be difficult at times to secure a workout session with an instructor. The instructor's schedules are only posted at the facility, and members must call in or visit the facility not knowing if they can get the time they would like. Members suggested that it would help if the instructor had their schedule posted on the site and have members register for a time slot through the new system. Additionally, CEO Dick Jones expressed a desire for the system to be a tool for registration.
- Provide members with a calendar of events for each month.
- There is a third site for the YMCA swim team which doesn't conform to the new look and feel of the YMCA and Y-Tri Club sites. Dick Jones, the Wilmington Family YMCA CEO, expressed a need for uniformity among all sites.
- Shannon Berg requested that the site be "condensed," so to provide for easier navigation for herself and members.

## Acknowledgements

The purpose of this section is to acknowledge the people that helped and supported me throughout the project.

First I would like to thank Charles Laymon for giving me the opportunity to work with him on one of his projects for the past year. I have learned so much in the time in which I worked with Charles. Charles not only showed me new technologies, but he served as a great mentor and supporter for many other aspects of my career. I am very fortunate for having worked with Charles.

I also would like to thank my project committee: Dr. Thomas Janicki, Dr. Douglas Kline, and Dr. Ronald Vetter. They gave me continuous support for which I am very grateful. In particular, I would like to thank my committee chair Dr. Thomas Janicki for all of the help and support he gave me. Dr. Janicki took many hours out of his schedule to work with me on all aspects of the project such as the presentation and the technical paper.

Finally I would like to thank my parents Wayne and Janice. This was a very demanding, time consuming, and a mentally exhausting capstone project. They provided continuous support and encouragement throughout the project, and throughout my academic career.

## **Appendices**

## Appendix A – Stakeholder Interviews

This section will detail the prime questions and responses for each of the major stakeholders in the project

### **Dick Jones – Wilmington Family YMCA CEO**

- **Q:** What do you primarily use the YMCA Web site for?  
**A:** “I use the site to make sure that the information is current and accurate. Specifically, I often check to ensure that YMCA events are scheduled correctly.”
- **Q:** Do you like the idea of having a messaging system?  
**A:** “Yes.”
- **Q:** What would you use the messaging system for primarily?  
**A:** “I would use the messaging system for our members who frequently visit the Web site for YMCA information. I would use it as a communication vehicle for those members. It would be very useful for sending emails and news letters about important YMCA events and dates to online registered members. I would also like to use it as a means for gathering information. For example, I could use it for surveys to discover member satisfaction levels on YMCA events such as our basketball league. ‘How did you like this year’s basketball season?’ This would help us optimize our programs.”
- **Q:** What features do you like about the messaging system?  
**A:** “I like that our members will be able to opt in and out of receiving messages. I am excited that it will bring our members with similar interests together. Finally, I like that the gathering of information will be made much easier. We

will not have to go find information as we would have in the past, information will come to us.

- **Q:** Are there any features that you do not like?

**A:** "I am concerned with privacy. Our members should have the ability to hide their information from other members if they wish. I am concerned about the confidentiality and privacy of our younger members (children). We also need to ensure the communication between our members and our server is secure. Passwords need to be heavily secured too."

- **Q:** Where would you like the website to be in the future?

**A:** "I would like our websites to be very interactive for both our members and our staff. I would like it to be a great tool for communication for members sharing information with other members, and for our staff and myself to give members important announcements. I want the sites to be a 'one-stop-shop' for information, information on events, registrations, and communications. Additionally, we currently have three sites: the Wilmington Family YMCA site, the Y-Tri Club site, and a site for our swim team (which is maintained by a swim team member). It would be really nice to integrate have all three of these sites to have the same look and feel. This would provide easier navigation for our members."

### **Shannon Berg – Member Services Director**

- **Q:** do you primarily use the YMCA Web site for?

**A:** "I use the site quite often for checking event dates, registration dates, to see what's going on in programs and to make sure our 'member specials' are up-to-date and accurate."

- **Q:** Do you like the idea of having a messaging system?

**A:** "Yes, I love it."

- **Q:** What would you use the messaging system for primarily?  
**A:** "I would use it to help get the word out to our members regarding Y-related news. I would use it to help keep the members informed of what is going on in our facility. A communication tool is something that we would really love."
- **Q:** Are there any features that you do not like?  
**A:** "I have no problems with the current specifications."
- **Q:** Do you like the idea of a content management system?  
**A:** "A content management system is something that we really need. Right now we have to rely on a third party for updates to our site's content. A content management system would allow us to remove the dependence on the third party for minor updates, which we have so often. For example, recently we had a change to our T-Ball registration date. However, it did not get updated on our website. We have a few people come into the office wanting to register their children for T-Ball when it wasn't the correct registration date. A tool such as a content management system would allow us to make those minor (yet important) changes when they arise."
- **Q:** How often do you have to make updates?  
**A:** "A lot. Programs get delayed, and other unforeseen events occur all the time. We have major updates to our events quarterly in our 'Program Guide.' All of the information in that gets updated to the website. The banner on top of the Wilmington Family YMCA site gets updated every month. And there are always spur of the moment events and updates that need to be done that we could do if we had such a tool."
- **Q:** Where would you like your website to be in the future?  
**A:** "I would like the site to be navigated through more easily. You have to click so many places to get where you want to go right now. The site could be

condensed to allow this to happen. I would like to be able to update our content easily without having to rely on a third party.”

**Lorie Lucas – YMCA Member**

- **Q:** Do you currently use the website? If so, how often?  
**A:** “Yes. Occasionally I use the site, about once a month.”
- **Q:** What is the reason you use it so rarely?  
**A:** “The content doesn’t change enough. Additionally, it is hard to find things like a specific class or schedules.”
- **Q:** In your opinion, what could be added to the site to make it better?  
**A:** “(With respect to the messaging system) I would like to see more channels. I would want to have a ‘Yoga’ channel. A channel or page that informs me of YMCA related news. Also, a calendar of events would be very helpful.”
- **Q:** What do you currently use the site for?  
**A:** “I use it for two things: to check the schedules of workouts and group classes.”
- **Q:** Do you like the idea of a messaging system?  
**A:** “Yes. If there is going to be an event like a 5K run, I want to know about it. I would love to receive monthly tips for running and nutrition tips.”
- **Q:** What else would you like to see on the messaging system?  
**A:** “As mentioned before, I would like additional channels like a ‘Yoga’ Channel. I would also like a general message channel that would inform me of important updates to the site, to programs or schedules. Also, a channel for team sports would be really great. Additionally, right now at the YMCA, if you want to sign up for a personal training session etc. you have to call the office and if the instructor isn’t there, you have to call back later. If they are there, then you have to manually sign up for a spot, not knowing what is currently

taken up. It would be great to message individual instructors to schedule sessions and have their schedule online.”

○ **Q:** What feature do you like about the messaging system concept?

**A:** “I like the ability to opt in/out of receiving messages.”

○ **Q:** Do you have any concerns about member profiles?

**A:** “Not if you can change your privacy information, I don’t have a problem.”

## Appendix B – Use Case Diagrams

The following contains detailed use case diagrams for the following system functions:

- Messaging system functions
- Account activation
- Member functions
- User profile functions
  - Setting profile privacy
  - Setting messaging preference (opt in/out)
  - Inserting/Editing/Deleting contact point types
    - Address
    - Phone
    - Email
- Member directory
- Role management
- Web content management

### **Messaging System:**

The following use case diagrams will detail the functions of the messaging system.

Figure B.1 details how a user would manage their subscriptions to a messaging channel. Users subscribe to messaging channels such as a “Running” channel to receive messages that pertain to running events.

## Messaging System Overview: Manage Subscriptions

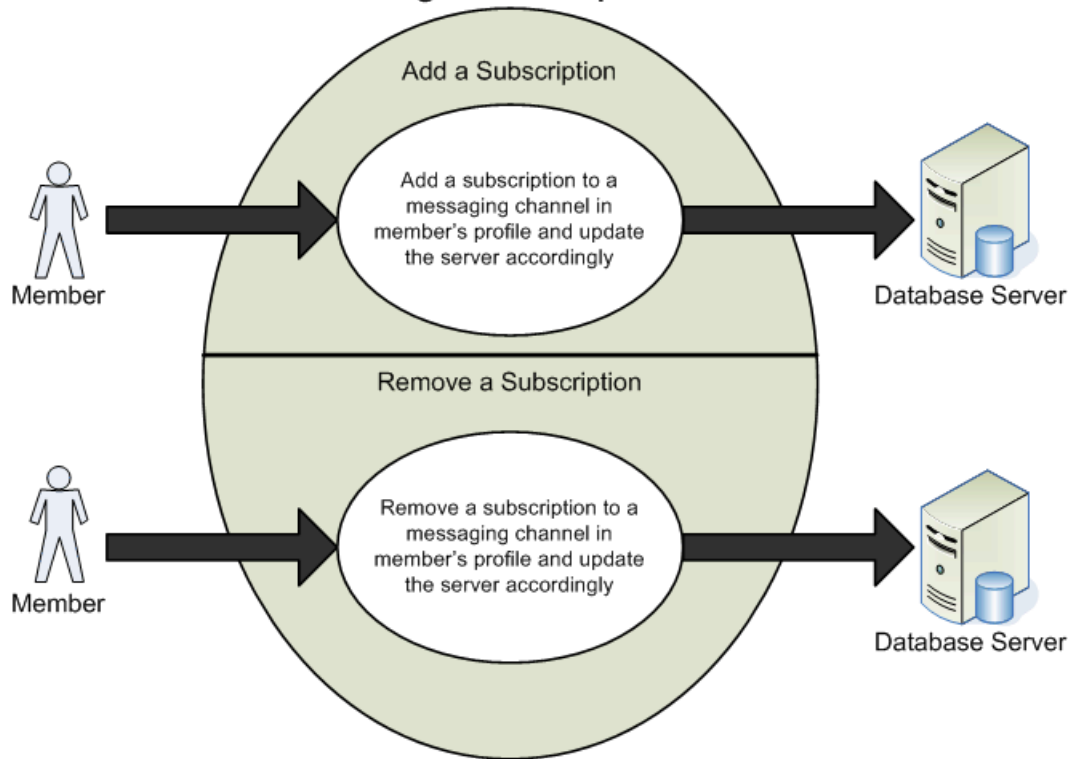


Figure B.1 – Managing channel subscriptions

Figure B.2 details the operations involved in the messaging system when a user sends messages to other users.

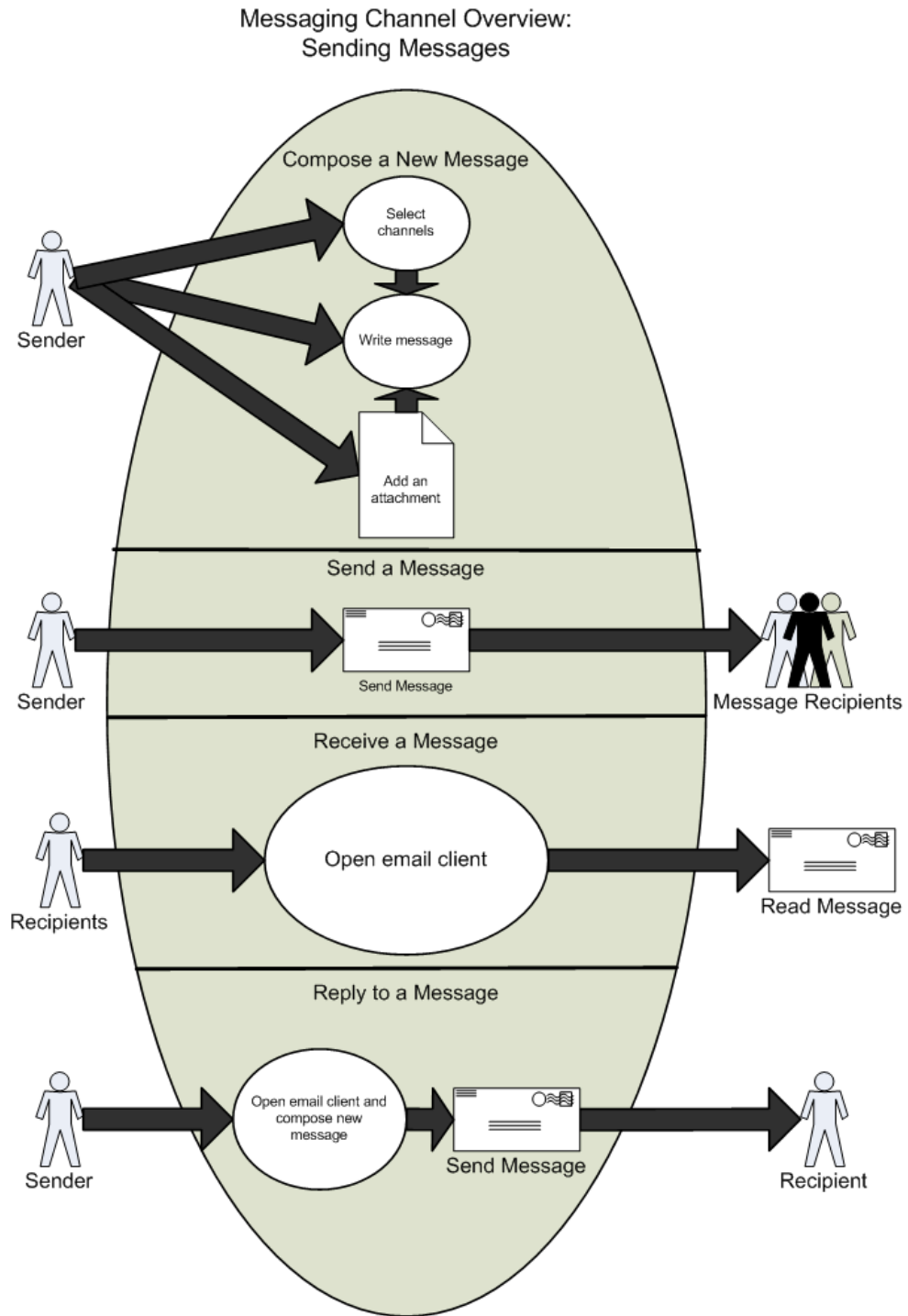


Figure B.2 – Operations of sending a message

Administrators are given the authority to create and edit channels, display or hide channels, or disable and enable a channel. Figure B.3 details the operations involved in each operation.

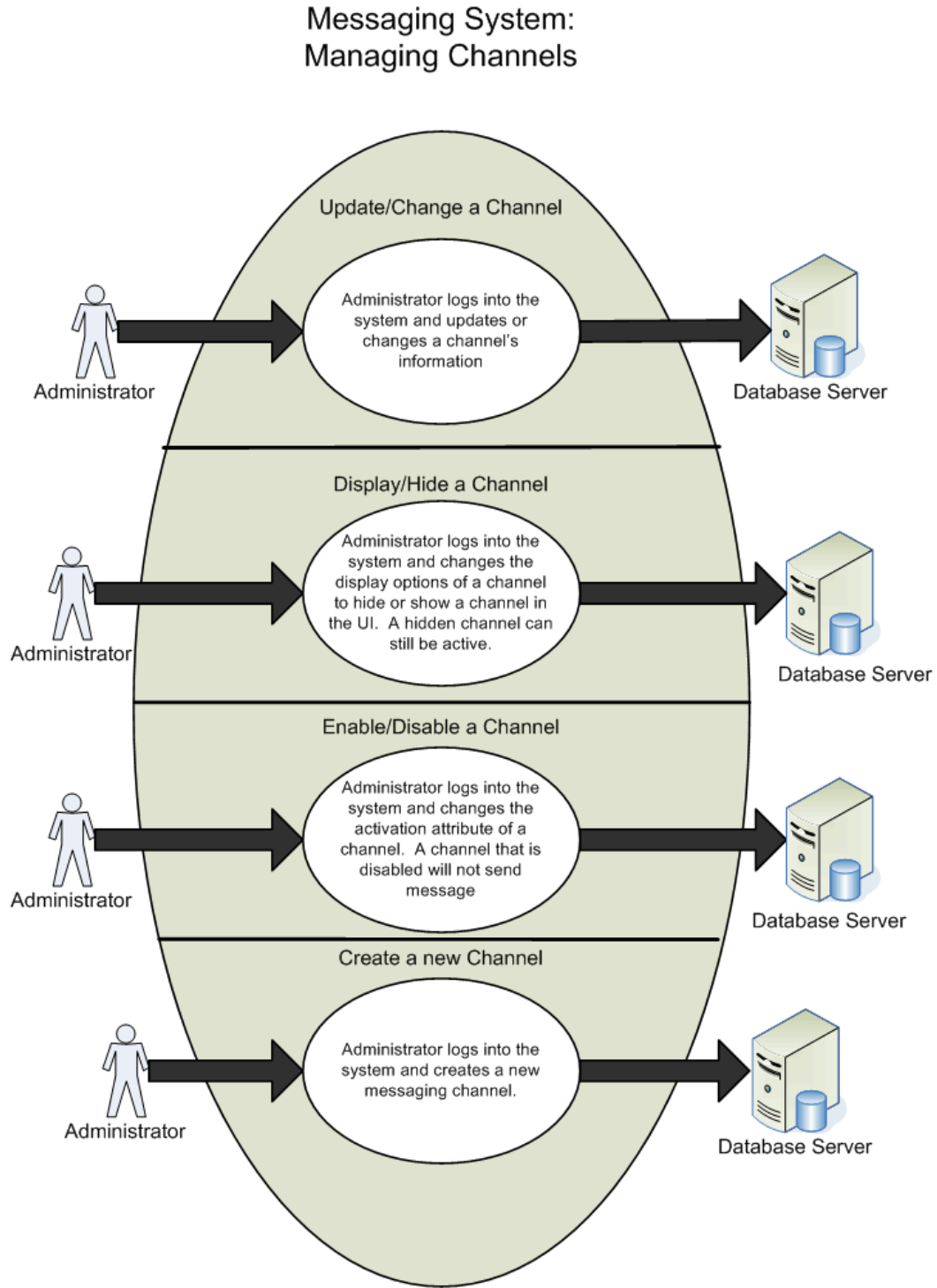
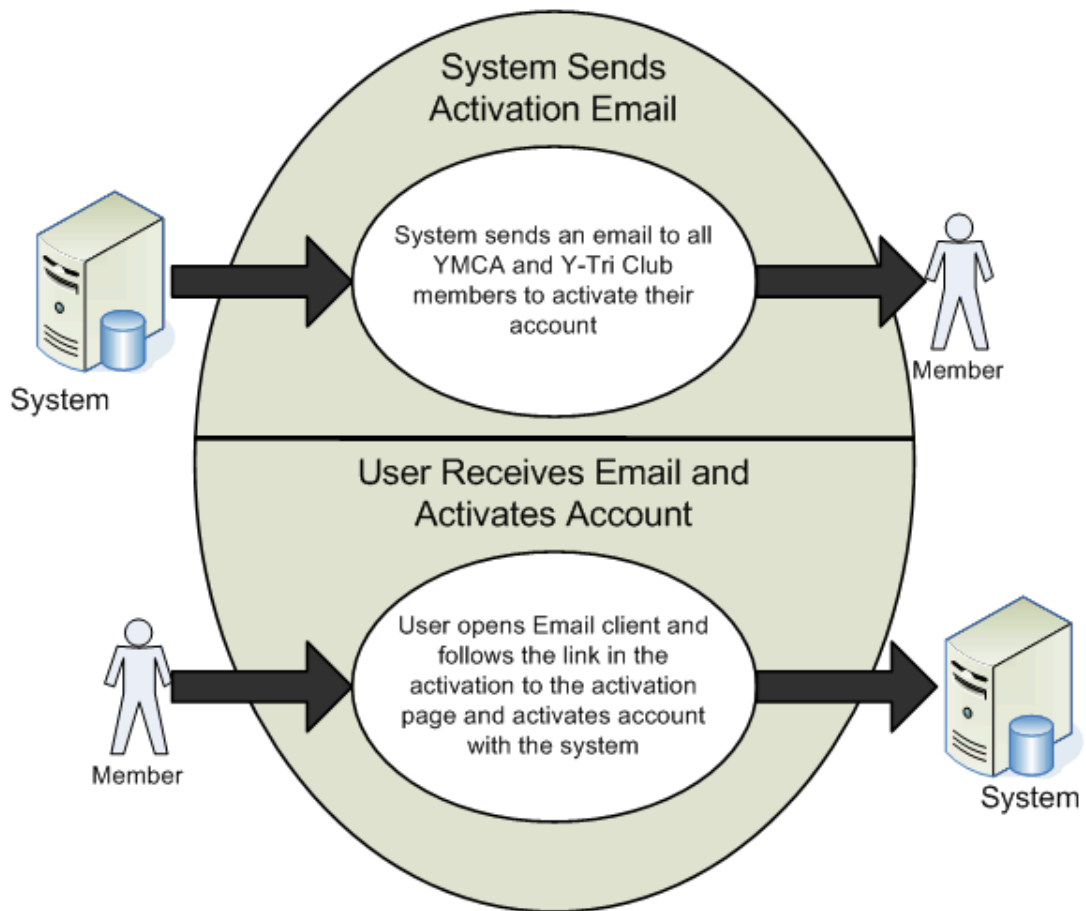


Figure B.3 – Administration of messaging channels

**Member Account Activation:**

In order to have control over who has an account in the new system, invitations to have an account are sent to registered YMCA and Y-Tri Club members. This is done so that non YMCA or Y-Tri Club personnel cannot have an account. Figure B.4 details the events Involved in member account activation.

### Account Activation: User Activates an Account



*Figure B.4 – Member account activation*

## Member Functions

Figure B.5 details all of the member functions that exist in the new system.

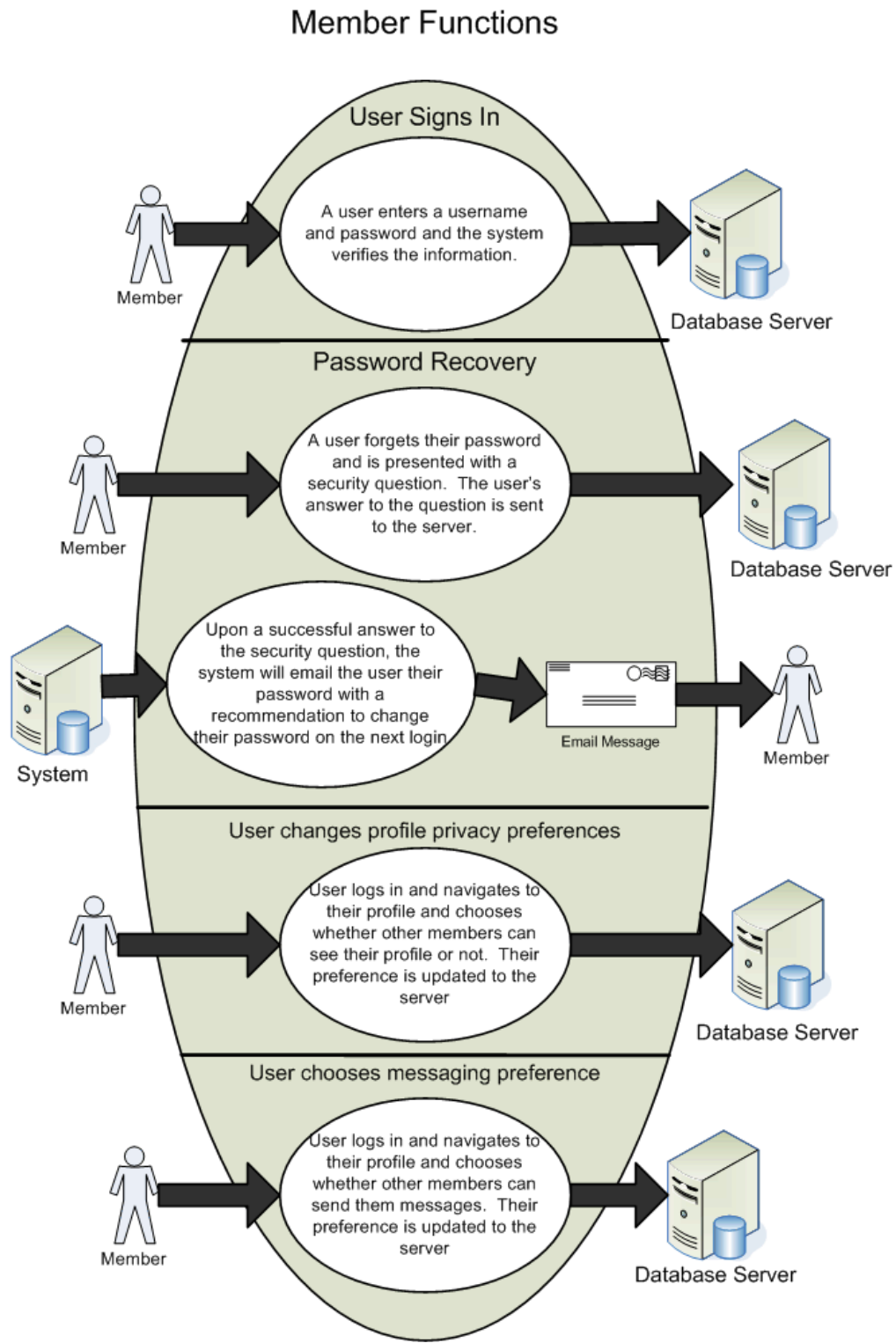


Figure B.5 – Member functions

Members have the ability to manage their “contact points.” A contact point refers to the users email addresses, phone numbers, or addresses. A user must have a primary contact point for each contact point type. Users may elect to not enter a phone or address contact point type however, in which case they would not have a primary contact point for those types. All members must have a primary email address (for this is the email their account invitation is sent to).

## Member Contact Point Type Management

The following applies to all three contact point types: Address, Email, Phone

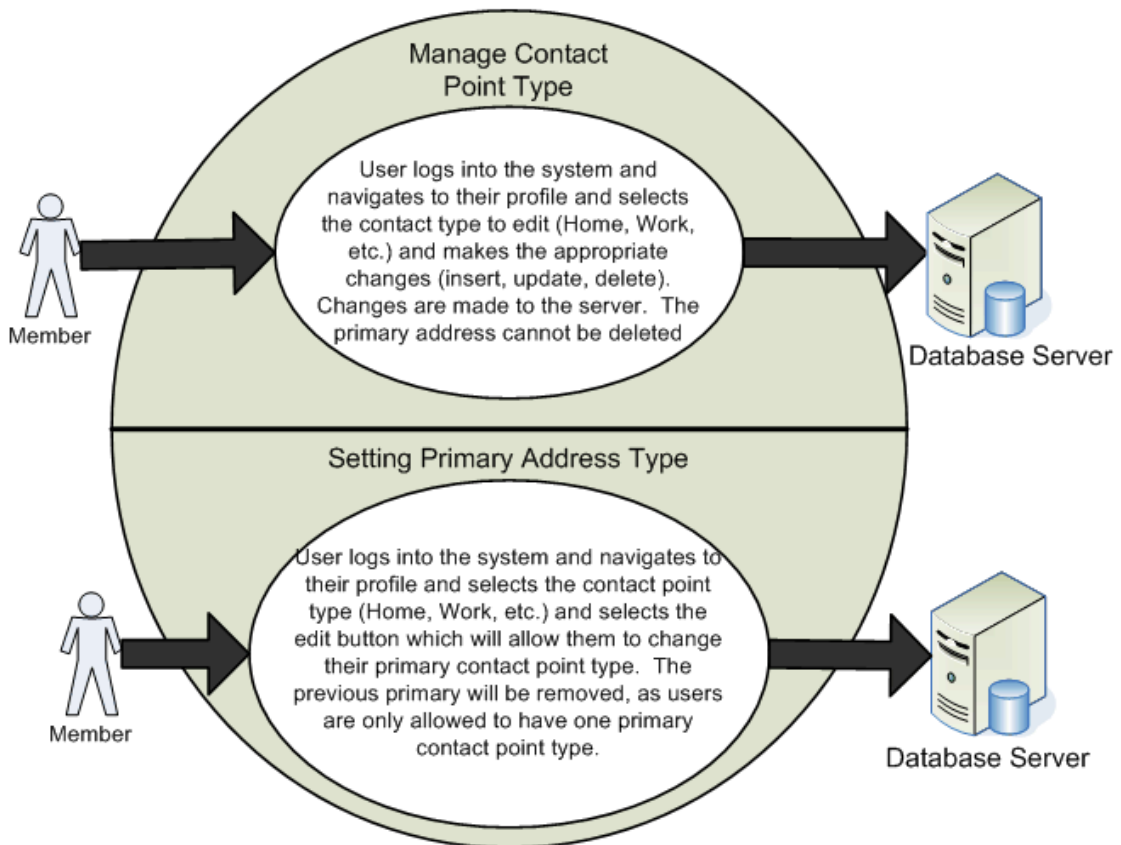


Figure B.6 – Contact point management

A part of the member functionality is a member directory. A member directory allows for the sharing of contact information. A user may elect to not share their information however. With the member directory, users may search for other members in the system. Figure B.7 details the functions of the member directory.

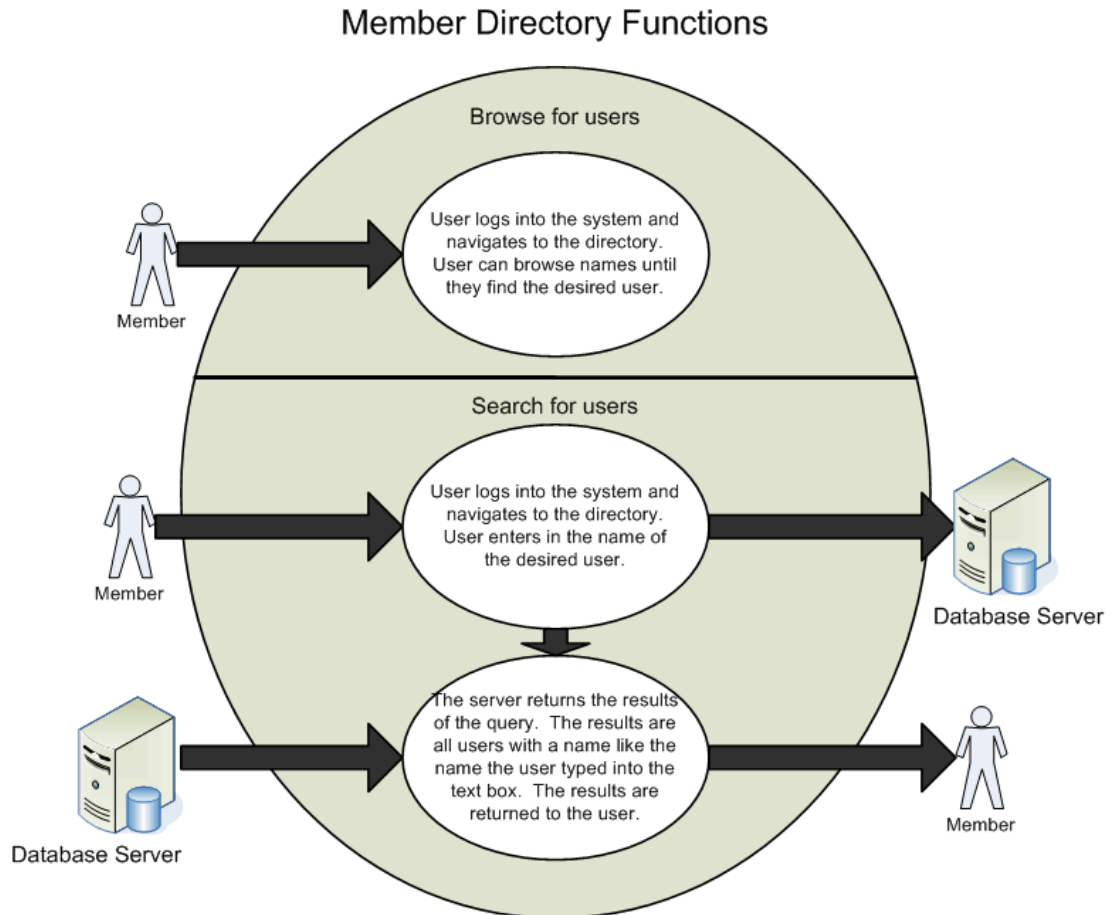


Figure B.7 – Member Directory

## Administrative Functions

Administrators are given the rights to manage user roles. Administrators can create, edit, and delete a user role. Additionally administrators have the ability to assign or remove a role from a user in the system as detailed in figure B.8.

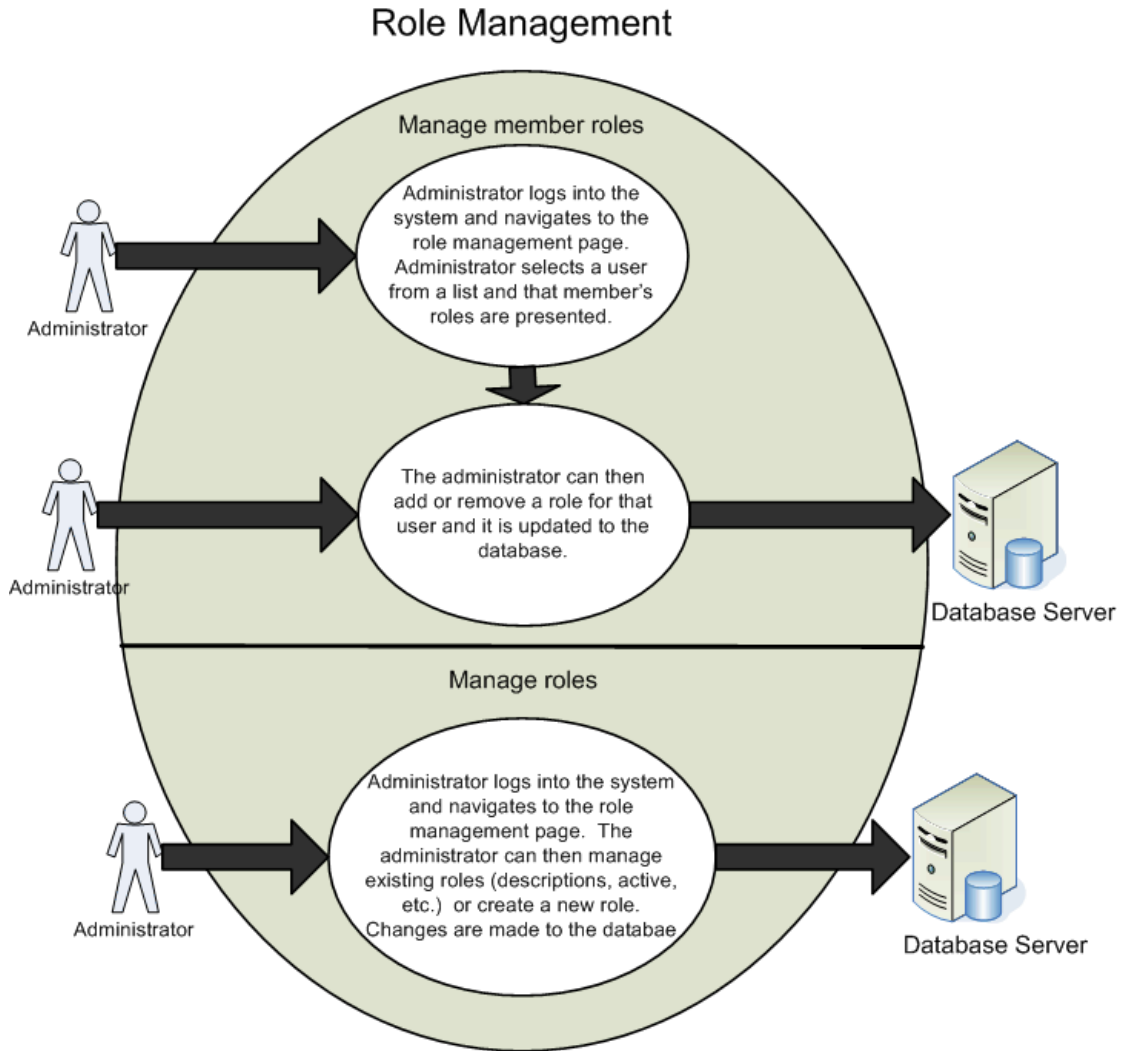


Figure B.8 – Role management

Users with administrative privileges have the ability to manage Web content. Administrators do not need prior knowledge or HTML to do updates, as the HTML editor "FreeTextBox" was implemented. Figure B.9 details how administrators manage Web content.

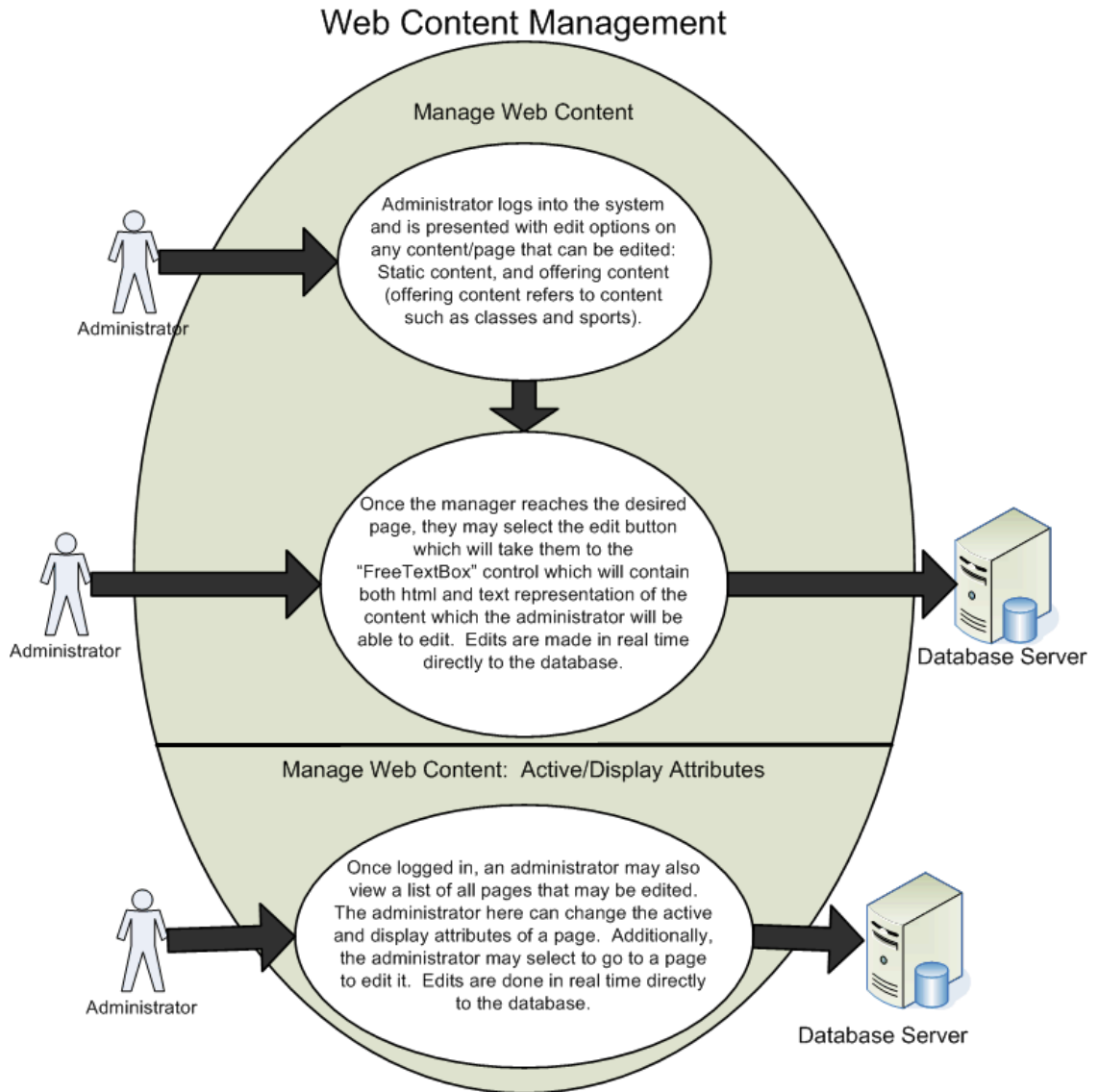


Figure B.9 – Web content management

## Appendix C – The Unified Process

The following details the specific steps which took place during the phases of the Unified Process.

### **Elaboration Phase**

#### Membership Phase

- *Further analysis and requirements gathering*
  - Define how YMCA members were to activate their account
  - Determine modification requirements for database tables
  - Determine necessary stored procedures for membership implementation
  - Define proper security measures
- *Make necessary modifications*
  - Database
    - Additional fields to dbo.AccessLogin
      - Password
      - Password recovery question
      - Password recovery answer
      - Account activated
      - Account active
    - Additional member roles to dbo.AccessRole
    - Table relationships
  - Stored Procedures - Create, Read, Update, Delete (CRUD)
    - Activating accounts
    - Password encryption
    - Password recovery

- Validate users on login
- Change password
- Manage member roles
- Create member roles
- Assign roles to members
- Member profiles
  - Privacy settings
  - Messaging settings (opt in/out of messages)
  - Address types (Home/Work)
    - Setting primary address contact point
  - Email types (Home/Work)
    - Setting primary email contact point
  - Phone types (Home/Work/Mobile)
    - Setting primary phone contact point
- Member directory search
- *Develop* – Membership and profile development was carried to near completion during the elaboration phase. As the project required a custom membership API, the default .NET membership management API had to be overwritten. To override the default .NET membership provider the following steps were taken:
  - Make the necessary Web Membership Provider modifications to the Web.Config file
    - Define the name of the file which would hold the membership provider operations
    - Create a subclass for the membership provider to separate and manage user validation: “WebUserProvider.cs”
  - Develop relevant Web Membership Provider methods:
    - Validating a member login

- Get a member password for password recovery
- Compare and verify member information for password recovery
- Change a member password
- Create data access layer table adapters:
  - Web Membership Provider methods
  - User account activation
  - Profile functions
  - Membership directory
- Develop the business logic:
  - Member profile
    - Profile privacy
    - Address information
    - Email information
    - Phone information
    - Message preferences
  - Member directory
    - Navigate to a person's profile through the .NET components
- Secure code in business logic layer by restricting access to the business logic layer controls based on:
  - Who is accessing the data?
    - If it is not the owner of the profile
      - Check the owner's privacy settings
      - Hide controls such as changing passwords
      - Remove edit and delete features for contact point types
  - Is the user logged in?
  - Is the user part of the correct domain in the database?



- Kept conversations stored in the database
- Instead of sending the message to the person directly, the email would provide the user with a link to go to the site to read the message
- Replying to an email would be done on the site rather than through the users email client

These requirements outlined were deemed unnecessary for the following reasons:

- As time was a factor and no messaging system was in place, getting a working messaging system was more important. Developing this would result in myself becoming the scope creep
  - It is inconvenient for the user to receive an email that states there is a message waiting for them. It is better to present the message in the email, rather than a link to the message
  - Since the Wilmington Family YMCA and Y-Tri Club are non-profit organizations, we opted to use Microsoft SQL Server 2005 Express: a free, less robust implementation of SQL Server 2005. It was unnecessary and impractical to store said messages in the database, when the limited space is needed for content
- *Make necessary modifications*
    - Database – New tables and stored procedures were needed to accomplish the messaging component.
      - Tables
        - NotificationChannel – The main table which holds the messaging notification channel information:
          - Channel type
          - Channel name
          - Channel description

- Channel active attribute – A channel not active will not allow any messages to be sent to that channel
  - Channel displayed attribute – A channel can be active, yet not displayed
- NotificationChannelType – The table which defines the types of channels. For this implementation there is only one type of channel: an “Email” channel type
- NotificationChannelAccess – The table which holds user channel subscriptions. It relates to the Access table, which is the main table for user information
- Stored Procedures – Create, Read, Update, Delete (CRUD)
  - NotificationChannel stored procedures for which administrators could use to manage channels
  - NotificationChannelType stored procedures
  - NotificationChannelAccess stored procedures for which be used to add/remove channel subscriptions for users
- *Develop* - For the development portion of the messaging sub phase, instead of creating a new class for sending emails, I interfaced the business logic in the messaging component with a class previously written by Charles Laymon of StepQuest, Inc.
  - Create data access layer table adapters:
    - Get subscribers for sending an email
    - Updating subscriptions
    - Modifying channels for administrators
  - Develop user interface layer:
    - Administration of messaging channels.
      - Update names

- Update description
    - Update channel active attribute
    - Update channel display attribute
  - Send messages
    - Select channel(s) in which the message is intended for
    - Subject line and message body
    - File upload dialog
- Develop business logic layer:
  - Interface messaging system logic with previously written “NotificationEmail” class
    - Handling attachments
    - Creating the necessary parts of the email such as To, From, BCC, etc.
  - Determine the recipients of the message via XML
- Secure code in the business logic layer – Secure the page based on who is logged in
  - If the user is not in the send messages role, do not show the page
  - If the user is not in the Admin role, do not allow access to the control message administration page
- *Non-formal testing* – During the Elaboration phase, formal testing of messaging functions with a tool such as NUnit was not exercised. The extent of testing during this phase consisted of logging into the system, subscribing to various channels, and testing if emails would be sent out properly.
  - Ensure I received emails sent from the page
  - Confirm that the message format was correct
  - Test attachments

- Met system size and extension constraints, system should not add attachment if not
- Ensure that if attachments met constraints that they were delivered correctly

Limited Web Content Management phase – The limited web content management phase was the final phase of the project life cycle.

- *Further analysis and requirements gathering*
  - Determine what method of implementation was feasible based on requirements.
    - Develop the system in house.
    - Find a solution: freeware, open source, or purchase.
- *Make necessary modifications* – It was decided that it was not feasible to develop a web content management system in house. Therefore, the only modification needed was installing the web content management control. The “Offering” tables required no modifications.
- *Develop* – Not much development was necessary as a web content management was not developed. “FreeTextBox,” a free HTML editor was chosen. All that was necessary was installing the “FreeTextBox” control on the system.
- *Non-formal testing* – During the elaboration phase, no formal testing with a tool such as NUnit was not exercised. Non-formal testing which took place during this phase consisted of accessing pages with “FreeTextBox” and making changes to pages and ensuring the updates were successfully executed

## Appendix D – Application Screenshots

### Logging Into the System:

Figure D.1 is of the homepage before a member is logged in. Notice the lack of menu options for a user not logged in. Figure D.2 is the application login screen.

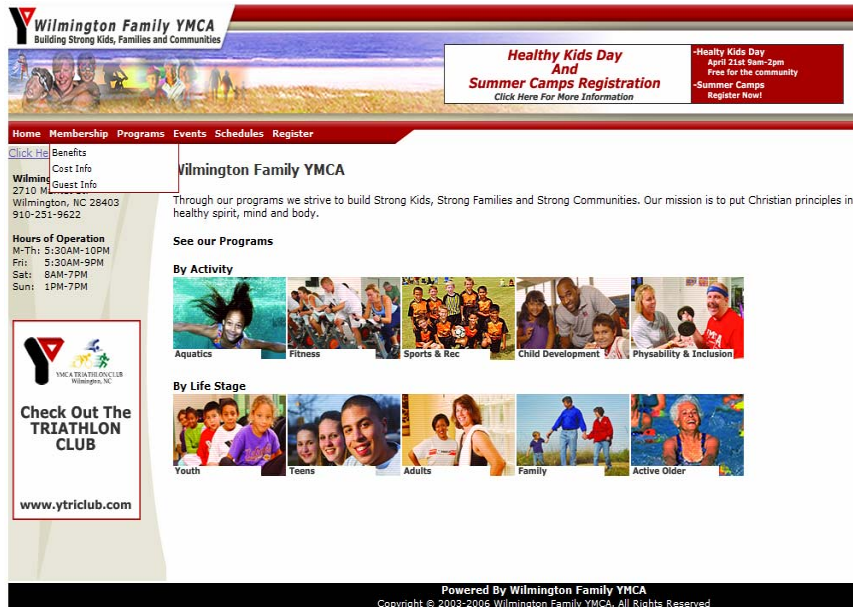


Figure D.1 – Homepage of the new system

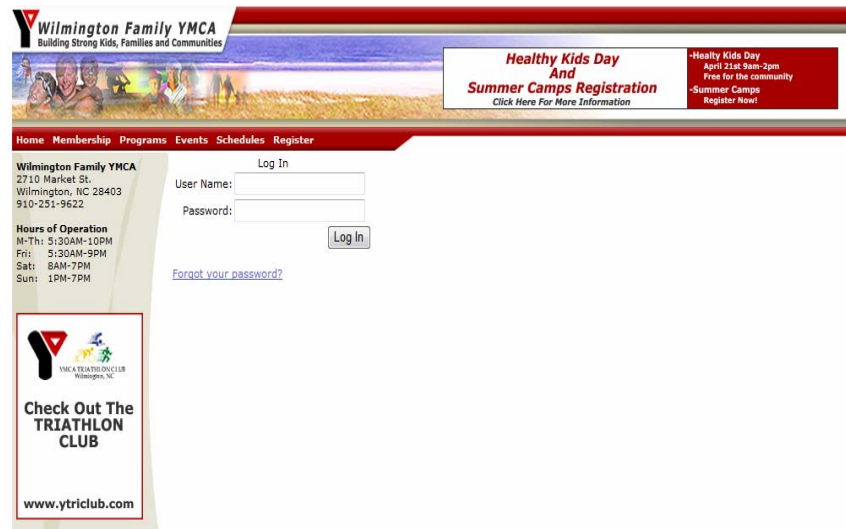


Figure D.2 – The login screen

The following screen shots detail the user's ability to recover their password. Figure D.3 demonstrates that the user must first enter their username. Upon entering a valid username, the user is then presented with a security question which they must provide the answer (which was set on activation of the account). A successful answer will result in the system emailing the user their password.

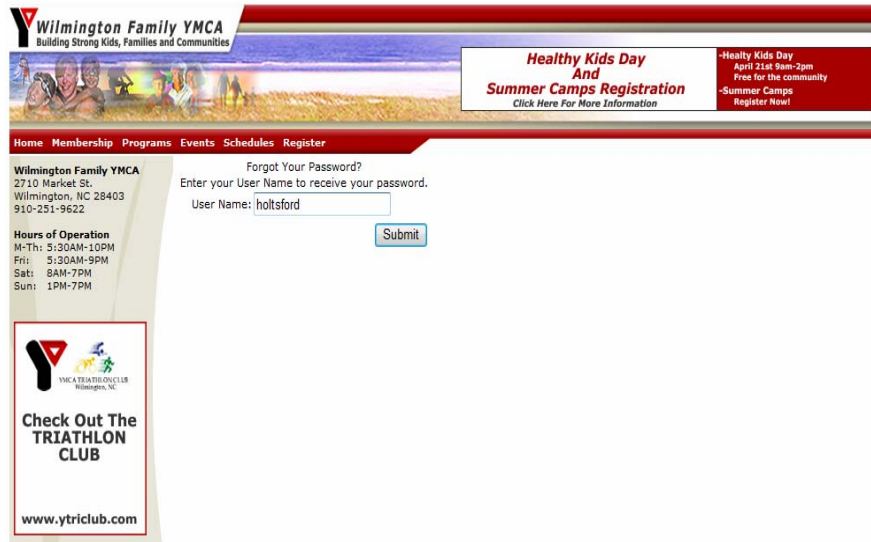


Figure D.3 – Enter a valid username to retrieve a password

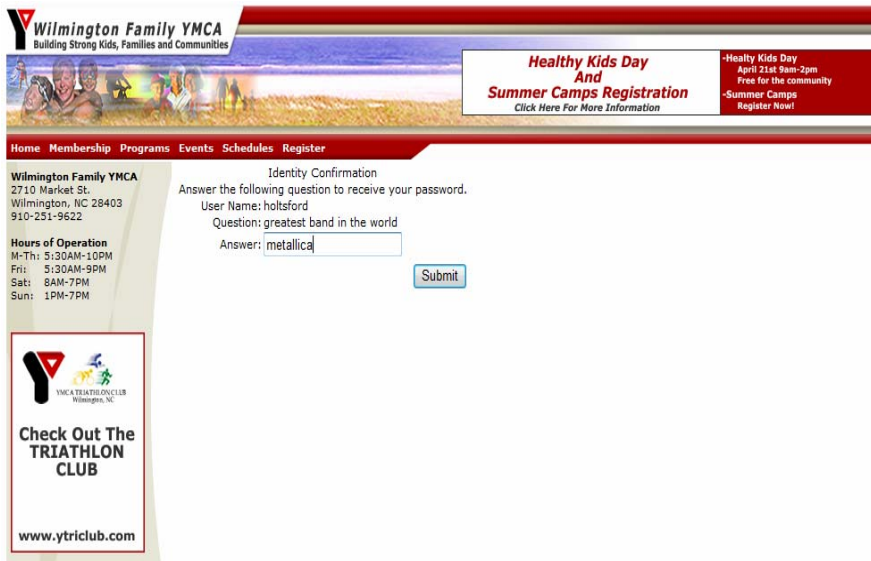


Figure D.4 – A correct answer to the user's security question must be provide

## Member Profile:


Upon a successful login by a user, they will be presented with access to their profiles and other areas of the system depending on their roles. In this particular screenshot, the user logged in has administrative roles. Notice the extra menus options, and the options to edit content.



Figure D.5 – The homepage when a user successfully logs into the system

Figure D.6 shows the user profile after a user is logged in. A member's profile consists of:

- The option to change their password
- The ability to opt in/out of receiving messages from other members
- Channel subscriptions
- A link to composing a message
- An option to change their profile privacy
- Options for editing their contact points



**Wilmington Family YMCA**  
Building Strong Kids, Families and Communities

**Healthy Kids Day  
And  
Summer Camps Registrat**  
[Click Here For More Information](#)


---

[Home](#) [Membership](#) [Programs](#) [Events](#) [Schedules](#) [Register](#)

holtsford [Logoff](#)

**Wilmington Family YMCA**  
2710 Market St.  
Wilmington, NC 28403  
910-251-9622

**Hours of Operation**  
M-Th: 5:30AM-10PM  
Fri: 5:30AM-9PM  
Sat: 8AM-7PM  
Sun: 1PM-7PM



**Check Out The  
TRIATHLON  
CLUB**

[www.ytriclub.com](http://www.ytriclub.com)

**Profile information for:** Holtsford, Chris

[Change your password](#)

**Contact Information:**

Would you like to receive email messages via Email?

[Edit](#)

**Messaging Channel Subscriptions:**

Run  Bike  Swim  Nutrition

[Compose Messages](#)

**Privacy Information:**

Would you like other members to view your profile information?

Check the box if you would like your information to be public.

[Edit](#)

**Your Email Address Information: Your Phone Information:**

Home  Work
  Home  Work  Mobile

Email Address: ckh0299@uncw.edu      Number: 9109999998

Primary Email Address:       Primary Phone Number:

[Edit](#)      [Edit](#)

**Your Address Information:**

Home  Work

Address Line 1: 131 charter dr  
Address Line 2:  
Address Line 3:  
Address Line 4:  
City: ilm  
Province: nc  
PostalCode: 28403  
Country: usa

Primary Address:

[Edit](#)

Figure D.6 – The user profile

Inside the member profile, users may change their password as demonstrated in Figure D.7:

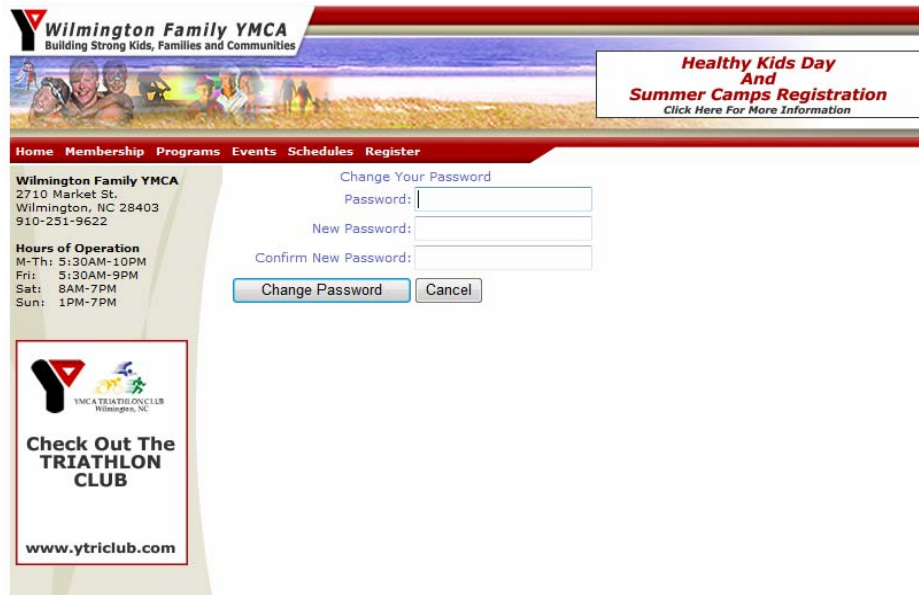


Figure D.7 – Changing a password in the user profile

A user can set their preference for receiving messages from other members through the messaging system as shown in Figure D.8:

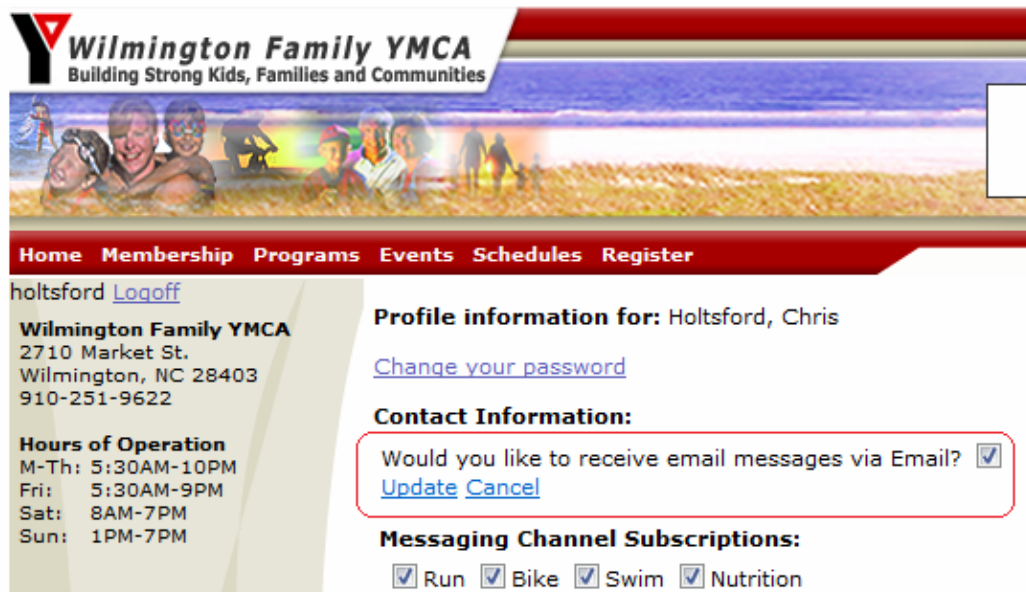


Figure D.8 – Changing messaging preferences

The following screenshot shows the ability of the user to change their contact information (outlined in red). Notice that when in *editing* a primary contact point, you are not allowed to remove the primary contact attribute. Also, you are not allowed to delete a primary contact point. The second screenshot on this page demonstrates that. Notice in Figure D.9 that users are not presented with a delete button for a primary address, because a user must have a primary address, email and phone contact point type.

**Your Email Address Information:**  
 Home  Work  
 Email Address:   
 Primary Email Address:   
[Update](#) [Cancel](#)

**Your Phone Information:**  
 Home  Work  Mobile  
 Number: 9109999998  
 Primary Phone Number:   
[Edit](#)

**Your Address Information:**  
 Home  Work  
 Address Line 1: 131 charter dr  
 Address Line 2:  
 Address Line 3:  
 Address Line 4:  
 City: ilm  
 Province: nc  
 PostalCode: 28403  
 Country: usa  
 Primary Address:   
[Edit](#)

Figure D.9 – Changing “Contact Points.” Primary contact points may not be deleted

**Your Email Address Information:**  
 Home  Work  
 Email Address:   
 Primary Email Address:   
[Edit](#)

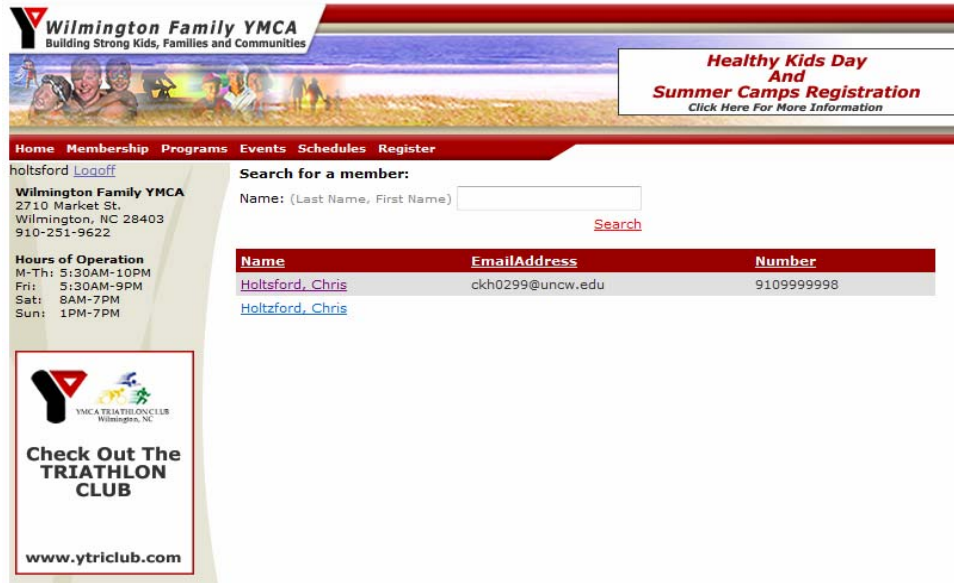
**Your Phone Information:**  
 Home  Work  Mobile  
 Number: (910)-326-4647  
 Primary Phone Number:   
[Edit](#) [Delete](#)

**Your Address Information:**  
 Home  Work  
 Address Line 1: 131 charter dr  
 Address Line 2:  
 Address Line 3:  
 Address Line 4:  
 City: ilm  
 Province: nc  
 PostalCode: 28403  
 Country: usa  
 Primary Address:   
[Edit](#)

Figure D.10 – Users may only delete a non-primary contact point

## Member Directory:

The following shows the member directory. On the initial page load, all members listed in the system are displayed (only two for this example). Click the name of a person and if their profile is public, you will be able to view their profile. Figure D.12 demonstrates that the results will narrow as you search for members.



**Wilmington Family YMCA**  
Building Strong Kids, Families and Communities

**Healthy Kids Day And Summer Camps Registration**  
Click Here For More Information

Home Membership Programs Events Schedules Register

holtsford [Logout](#)

**Wilmington Family YMCA**  
2710 Market St.  
Wilmington, NC 28403  
910-251-9622

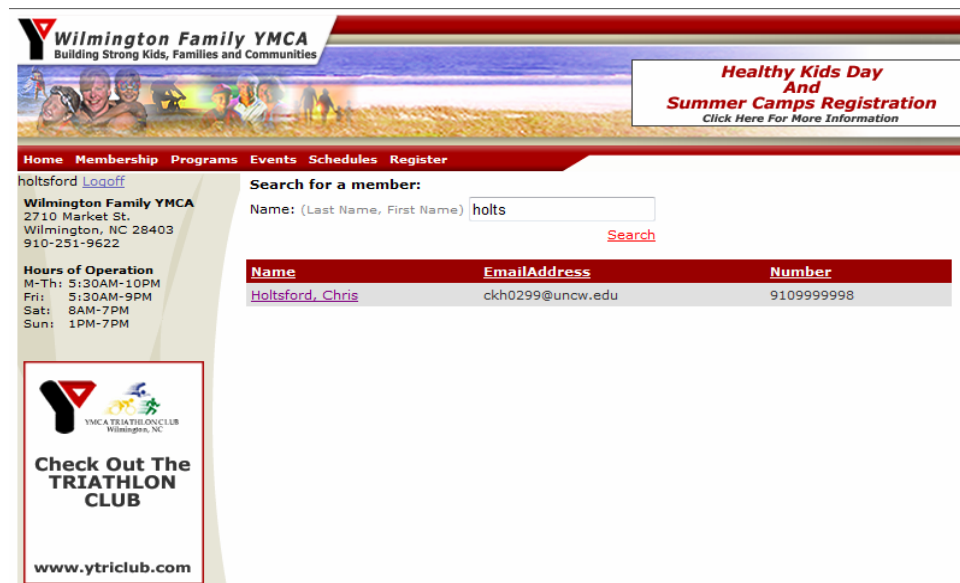
**Hours of Operation**  
M-Th: 5:30AM-10PM  
Fri: 5:30AM-9PM  
Sat: 8AM-7PM  
Sun: 1PM-7PM

**Check Out The TRIATHLON CLUB**  
www.ytriclub.com

**Search for a member:**  
Name: (Last Name, First Name)   
[Search](#)

Name	EmailAddress	Number
<a href="#">Holtsford, Chris</a>	ckh0299@uncw.edu	9109999998
<a href="#">Holtzford, Chris</a>		

Figure D.11 – Member Directory



**Wilmington Family YMCA**  
Building Strong Kids, Families and Communities

**Healthy Kids Day And Summer Camps Registration**  
Click Here For More Information

Home Membership Programs Events Schedules Register

holtsford [Logout](#)

**Wilmington Family YMCA**  
2710 Market St.  
Wilmington, NC 28403  
910-251-9622

**Hours of Operation**  
M-Th: 5:30AM-10PM  
Fri: 5:30AM-9PM  
Sat: 8AM-7PM  
Sun: 1PM-7PM

**Check Out The TRIATHLON CLUB**  
www.ytriclub.com

**Search for a member:**  
Name: (Last Name, First Name)   
[Search](#)

Name	EmailAddress	Number
<a href="#">Holtsford, Chris</a>	ckh0299@uncw.edu	9109999998

Figure D.12 – Results are narrowed when searching for members

## Administrative Functions – Messaging Channel Management:

Administrators are given the ability to add to system components, such as adding messaging channels. In Figure D.13, there are two tabs: “Manage Messaging Channels” and “Send Messages.” Only administrators will be given access to the “Manage Messaging Channels” tab. Figure D.14 shows channel management in “edit” mode.

Wilmington Family YMCA  
Building Strong Kids, Families and Communities

Home Membership Programs Events Schedules Register

Wilmington Family YMCA  
2710 Market St.  
Wilmington, NC 28403  
910-251-9622

Hours of Operation  
M-Th: 5:30AM-10PM  
Fri: 5:30AM-9PM  
Sat: 8AM-7PM  
Sun: 1PM-7PM

Check Out The TRIATHLON CLUB  
www.ytriclub.com

Manage Messaging Channels | Send Messages

Name	Description
Select Run	Running notification channel for YMCA
Select Bike	Biking notification channel for ymca
Select Swim	Swimming notification channel for ymca
Select Nutrition	Nutrition notification channel for ymca

Name: Run  
Description: Running notification channel for YMCA  
Enabled:   
Displayed:   
Edit Delete New

Figure D.13 – Manage “messaging channels”

Wilmington Family YMCA  
Building Strong Kids, Families and Communities

Home Membership Programs Events Schedules Register

Wilmington Family YMCA  
2710 Market St.  
Wilmington, NC 28403  
910-251-9622

Hours of Operation  
M-Th: 5:30AM-10PM  
Fri: 5:30AM-9PM  
Sat: 8AM-7PM  
Sun: 1PM-7PM

Check Out The TRIATHLON CLUB  
www.ytriclub.com

Manage Messaging Channels | Send Messages

Name	Description
Select Run	Running notification channel for YMCA
Select Bike	Biking notification channel for ymca
Select Swim	Swimming notification channel for ymca
Select Nutrition	Nutrition notification channel for ymca

Name: Bike  
Description: Biking notification channel  
Enabled:   
Displayed:   
Update Cancel

Figure D.14 – Channel management in “edit” mode

Figure D.15 shows the messaging system. There is a user role called “MessagesAdmin” in which a user must be assigned in order to send messages. Administrators may assign or remove roles which will be detailed later.

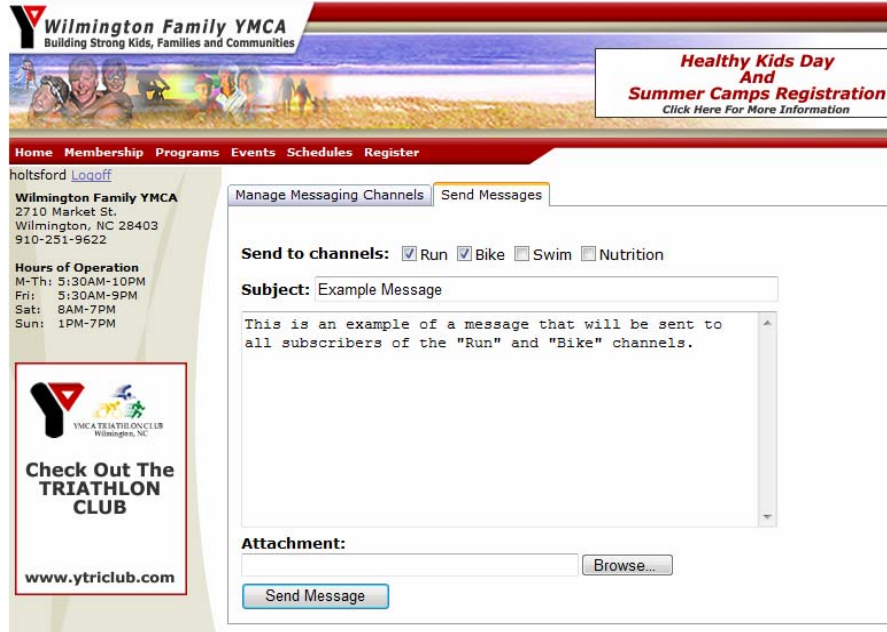


Figure D.15 – The Messaging System

### Administrative Functions – Role Management

Administrators are given the ability to manage roles. They may create a role, edit a role, delete a role, assign a role to a user, or remove a role from a user. Figure D.16 demonstrates the user interface the administrator is presented with when managing roles. Figure D.17 shows the role management function in “edit” mode.

**Wilmington Family YMCA**  
Building Strong Kids, Families and Communities

**Healthy Kids Day And Summer Camps Registration**  
Click Here For More Information

**Healthy Kids Day**  
April 21st 9am-2pm  
Free for the community

**Summer Camps**  
Register Now!

Home Membership Programs Events Schedules Register

holtsford [Logout](#)

**Wilmington Family YMCA**  
2710 Market St.  
Wilmington, NC 28403  
910-251-9622

**Hours of Operation**  
M-Th: 5:30AM-10PM  
Fri: 5:30AM-9PM  
Sat: 8AM-7PM  
Sun: 1PM-7PM

**Check Out The TRIATHLON CLUB**  
www.ytriclub.com

Manage User Roles [Assign User Roles](#)

Name	Description	Modify Roles
<a href="#">Select</a> Admin	Role has full admin privileges (YMCA)	Name Admin Description Role has full admin privileges (YMCA) <a href="#">Edit</a> <a href="#">Delete</a> <a href="#">New</a>
<a href="#">Select</a> ContentAdmin	Role can change site content (YMCA)	
<a href="#">Select</a> User	Role has no administrative privileges (YMCA)	
<a href="#">Select</a> MessagesAdmin	Role has admin privileges for YMCA messages	
<a href="#">Select</a> SendMessages	Role allows users to send messages for the ymca	
<a href="#">Select</a> UpLoadAdmin	user can upload files	

Figure D.16 – Role management UI

**Wilmington Family YMCA**  
Building Strong Kids, Families and Communities

**Healthy Kids Day And Summer Camps Registration**  
Click Here For More Information

**Healthy Kids Day**  
April 21st 9am-2pm  
Free for the community

**Summer Camps**  
Register Now!

Home Membership Programs Events Schedules Register

holtsford [Logout](#)

**Wilmington Family YMCA**  
2710 Market St.  
Wilmington, NC 28403  
910-251-9622

**Hours of Operation**  
M-Th: 5:30AM-10PM  
Fri: 5:30AM-9PM  
Sat: 8AM-7PM  
Sun: 1PM-7PM

**Check Out The TRIATHLON CLUB**  
www.ytriclub.com

Manage User Roles [Assign User Roles](#)

Name	Description	Modify Roles
<a href="#">Select</a> Admin	Role has full admin privileges (YMCA)	Name <input type="text" value="Admin"/> Description <input type="text" value="Role has full admin privileges (YMCA)"/> <a href="#">Update</a> <a href="#">Cancel</a>
<a href="#">Select</a> ContentAdmin	Role can change site content (YMCA)	
<a href="#">Select</a> User	Role has no administrative privileges (YMCA)	
<a href="#">Select</a> MessagesAdmin	Role has admin privileges for YMCA messages	
<a href="#">Select</a> SendMessages	Role allows users to send messages for the ymca	
<a href="#">Select</a> UpLoadAdmin	user can upload files	

Figure D.17 – Role management in "edit" mode

Figure D.18 (shown below) shows the administrator's ability to assign or remove a role from a user. To do so, an administrator must simply select a user from the system (only two for this example). When the user selects a user, they are presented with all of the roles the user belongs to. A check in the checkbox beside a role means the user belongs to that role. To add or remove a role from a user, the administrator must simply check or uncheck the box and the system will update accordingly.

**Wilmington Family YMCA**  
Building Strong Kids, Families and Communities

**Healthy Kids Day And Summer Camps Registration**  
Click Here For More Information

Home Membership Programs Events Schedules Register

holtsford [Logout](#)

**Wilmington Family YMCA**  
2710 Market St.  
Wilmington, NC 28403  
910-251-9622

**Hours of Operation**  
M-Th: 5:30AM-10PM  
Fri: 5:30AM-9PM  
Sat: 8AM-7PM  
Sun: 1PM-7PM

**Check Out The TRIATHLON CLUB**  
www.ytriclub.com

Manage User Roles Assign User Roles

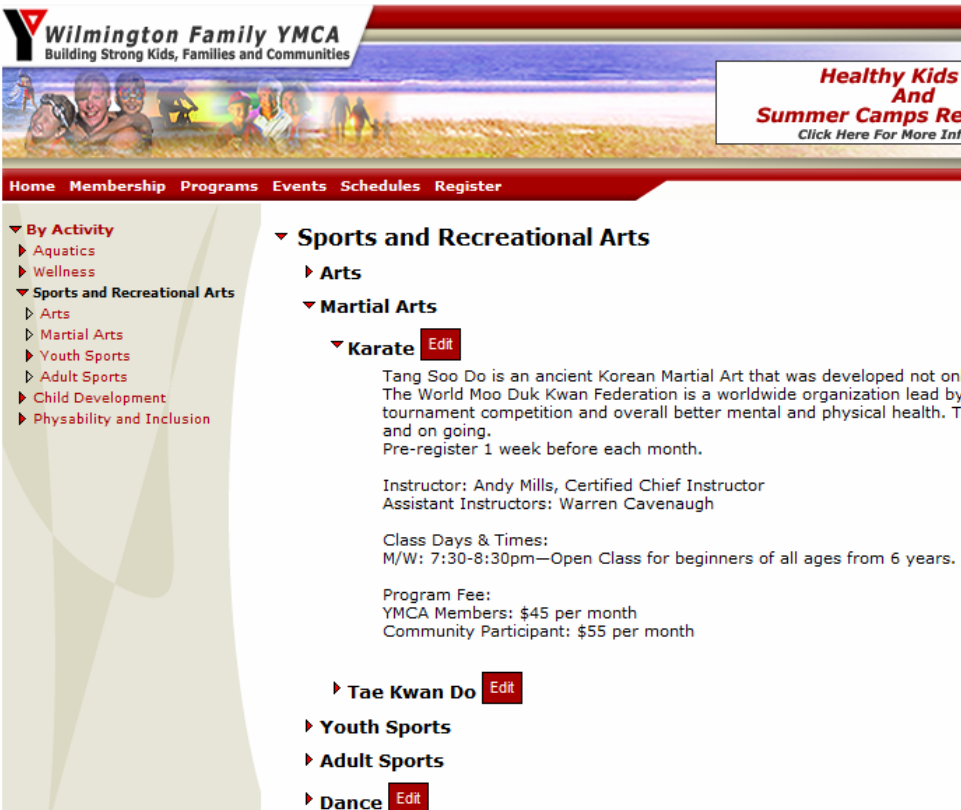
	LoginId	FirstName	LastName	EmailAddress	Roles
<a href="#">Select</a>	holtsford	Chris	Holtsford	ckh0299@uncw.edu	<input checked="" type="checkbox"/> Admin <input checked="" type="checkbox"/> ContentAdmin <input checked="" type="checkbox"/> User <input checked="" type="checkbox"/> MessagesAdmin <input checked="" type="checkbox"/> SendMessages <input checked="" type="checkbox"/> UpLoadAdmin
<a href="#">Select</a>	christest	Chris	Holtzford		

Figure D.18 – Assigning/Removing roles

## Administrative Functions – Managing Content

The database for the two applications contains over 200 content records.

Administrators are given the right to change these records. For this example, a content record is “Karate.” When *administrators* navigate through the site, they are presented with “Edit” buttons such as those in Figure D.19 when there is a record which can be edited. Once a record is selected to be edited, the administrator is taken to a page such as Figure D.20 where more information on the record is provided, such as the administrator who updated the page last. If the administrator elects to update the record, they must select the “Edit” button as shown of Figure D.20 which will take them to the “FreeTextBox” HTML editor control, shown in Figure D.21. This control allows the administrator to manage content without prior knowledge of HTML.



The screenshot shows the Wilmington Family YMCA website. The header includes the logo and tagline "Building Strong Kids, Families and Communities". A navigation menu lists Home, Membership, Programs, Events, Schedules, and Register. A sidebar menu is titled "By Activity" and includes categories like Aquatics, Wellness, Sports and Recreational Arts, Child Development, and Physability and Inclusion. The main content area is titled "Sports and Recreational Arts" and lists "Arts" and "Martial Arts". Under "Martial Arts", "Karate" is listed with an "Edit" button. Below the "Karate" entry, there is a detailed description of Tang Soo Do, including information about the instructor (Andy Mills), class days and times (M/W: 7:30-8:30pm), and program fees (\$45 per month for YMCA Members, \$55 per month for Community Participants). Other activities listed include "Tae Kwan Do", "Youth Sports", "Adult Sports", and "Dance", each with an "Edit" button.

Figure D.19 – Administrators are presented with “Edit” buttons on content

**Wilmington Family YMCA**  
Building Strong Kids, Families and Communities

**Healthy Kids And Summer Camps Reg**  
Click Here For More Info

Home Membership Programs Events Schedules Register

**Wilmington Family YMCA**  
2710 Market St.  
Wilmington, NC 28403  
910-251-9622

**Hours of Operation**  
M-Th: 5:30AM-10PM  
Fri: 5:30AM-9PM  
Sat: 8AM-7PM  
Sun: 1PM-7PM

**Check Out The TRIATHLON CLUB**  
www.ytriclub.com

Name: Karate  
Description:

Tang Soo Do is an ancient Korean Martial Art that was developed not only for self-defen  
The World Moo Duk Kwan Federation is a worldwide organization lead by Grandmaster J  
competition and overall better mental and physical health. This art will help you to impr  
Pre-register 1 week before each month.

Instructor: Andy Mills, Certified Chief Instructor  
Assistant Instructors: Warren Cavenaugh

Class Days & Times:  
M/W: 7:30-8:30pm—Open Class for beginners of all ages from 6 years.

Program Fee:  
YMCA Members: \$45 per month  
Community Participant: \$55 per month

Previously modified by user: Holtsford, Chris On 12/29/2005 4:17:54 PM

[Edit](#)

Figure D.20 – Content record in “edit” mode

Home Membership Programs Events Schedules Register

**Wilmington Family YMCA**  
2710 Market St.  
Wilmington, NC 28403  
910-251-9622

**Hours of Operation**  
M-Th: 5:30AM-10PM  
Fri: 5:30AM-9PM  
Sat: 8AM-7PM  
Sun: 1PM-7PM

**Check Out The TRIATHLON CLUB**  
www.ytriclub.com

Name: Karate  
Description:

Tang Soo Do is an ancient Korean Martial Art that was developed not only for self-defense, but also  
to condition the body and mind.  
The World Moo Duk Kwan Federation is a worldwide organization lead by Grandmaster Jae Joon  
Kim. Students will train and condition their body and mind for self-defense, tournament competition  
and overall better mental and physical health. This art will help you to improve your balance,  
flexibility, and self-confidence. Classes are monthly and on going.  
Pre-register 1 week before each month.

Instructor: Andy Mills, Certified Chief Instructor  
Assistant Instructors: Warren Cavenaugh

Class Days & Times:  
M/W: 7:30-8:30pm—Open Class for beginners of all ages from 6 years.

Program Fee:  
YMCA Members: \$45 per month  
Community Participant: \$55 per month

Design HTML

Previously modified by user: Holtsford, Chris On 12/29/2005 4:17:54 PM

[Update](#) [Cancel](#)

Figure D.21 – “FreeTextBox” HTML editor control

Administrators may also view a list of all manageable content and alter the “Active” and “Display” attribute. They may navigate to the page of the content by selecting the one they wish to edit. This is shown below in Figure D.22.

**Wilmington Family YMCA**  
Building Strong Kids, Families and Communities

**Healthy Kids And Summer Camps Re**  
*Click Here For More Inf*

[Home](#) [Membership](#) [Programs](#) [Events](#) [Schedules](#) [Register](#)

holtsford [Logoff](#)

**Wilmington Family YMCA**  
2710 Market St.  
Wilmington, NC 28403  
910-251-9622

**Hours of Operation**  
M-Th: 5:30AM-10PM  
Fri: 5:30AM-9PM  
Sat: 8AM-7PM  
Sun: 1PM-7PM

**Check Out The TRIATHLON CLUB**  
[www.ytriclub.com](http://www.ytriclub.com)

<a href="#">Edit</a>	<a href="#">Select</a>	Offering Type	Name	Active	Display
<a href="#">Edit</a>	<a href="#">Select</a>	Service	Parent/Child Classes	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<a href="#">Edit</a>	<a href="#">Select</a>	Service	Pike (Beginner I)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">Edit</a>	<a href="#">Select</a>	Service	Eel (Beginner II)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">Edit</a>	<a href="#">Select</a>	Service	Ray (Adv. Beginner I)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">Edit</a>	<a href="#">Select</a>	Service	Starfish (Adv. Beginner II)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">Edit</a>	<a href="#">Select</a>	Service	Pollywog/Guppy I (Beginner)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">Edit</a>	<a href="#">Select</a>	Service	Pollywog/Guppy II (Adv. Beginner)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">Edit</a>	<a href="#">Select</a>	Service	Minnow/Fish	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">Edit</a>	<a href="#">Select</a>	Service	Adult Swim Lessons	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">Edit</a>	<a href="#">Select</a>	Service	Private Lessons	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">Edit</a>	<a href="#">Select</a>	Service	Early Bird	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">Edit</a>	<a href="#">Select</a>	Service	Aquacise	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">Edit</a>	<a href="#">Select</a>	Service	Water Blast	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">Edit</a>	<a href="#">Select</a>	Service	Rusty Joints	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">Edit</a>	<a href="#">Select</a>	Service	Aqua Moves	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

1 2 3 4 5 6 7 8 9 10 ...

Figure D.22 – Directory of content records

## Appendix E – Database Diagrams

The following are diagrams of the database used in the project. Not all tables were used however as I was working on an existing database. Following the diagrams I will outline what was heavily used in the project and what I contributed.

### Access Tables

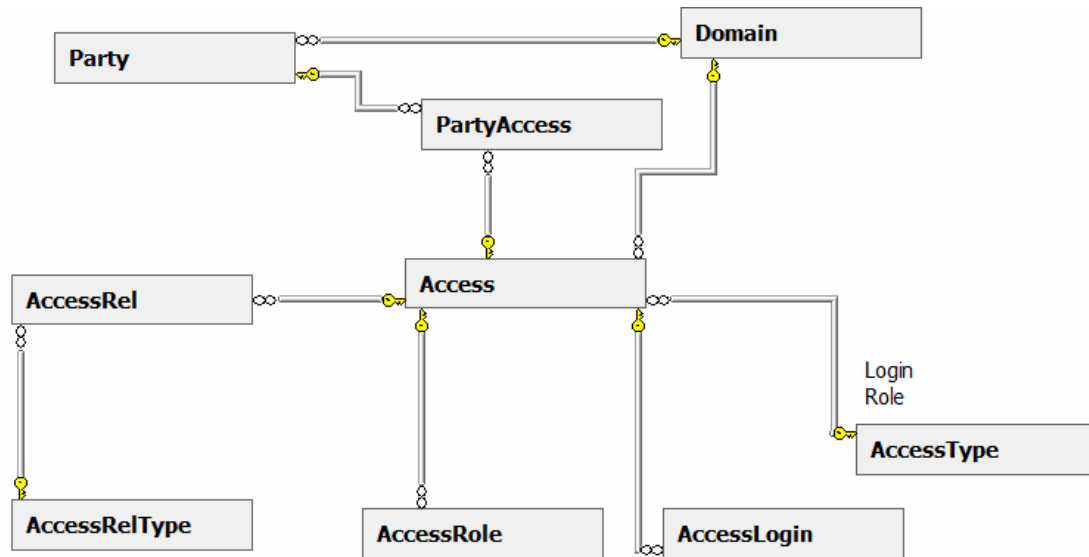


Figure E.1 – Access tables

### Contact Point Tables

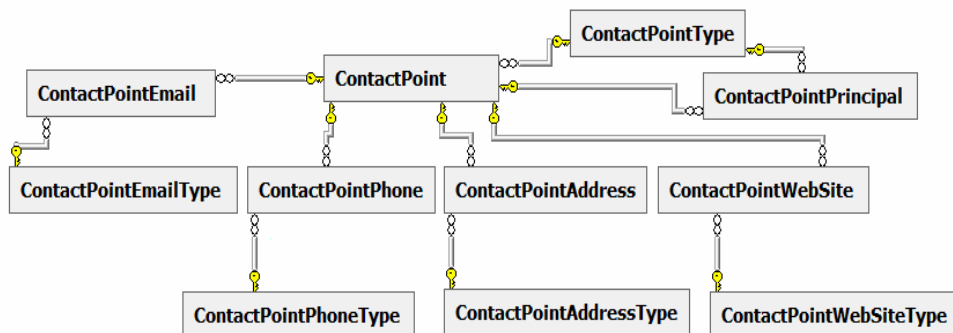


Figure E.2 – Contact Point tables

### Domain Tables

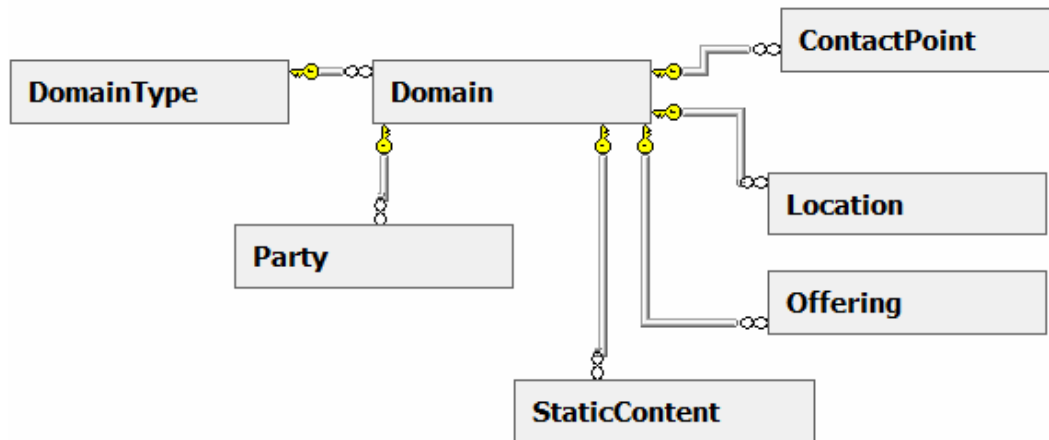


Figure E.3 – Domain tables

### Notification Channel Tables

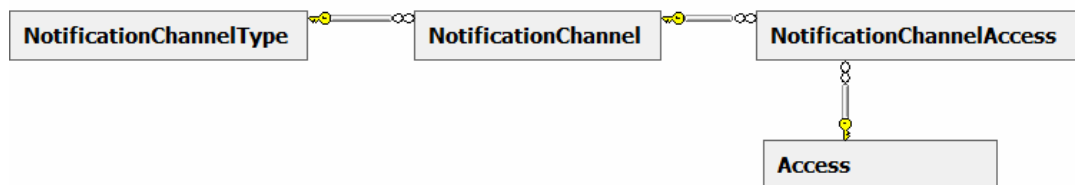


Figure E.4 – Notification Channel tables

## Offering Tables

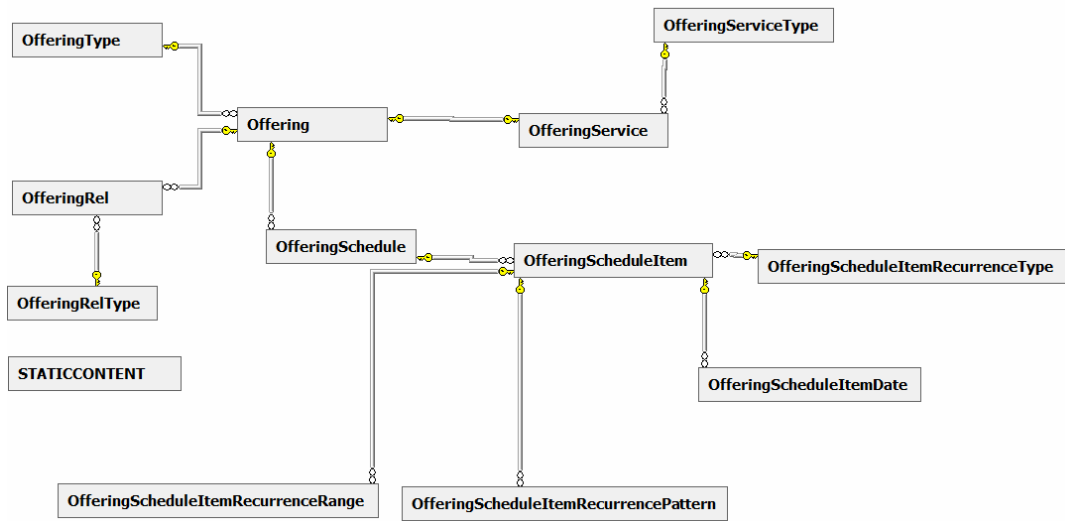


Figure E.5 – Offering tables

## Party Tables

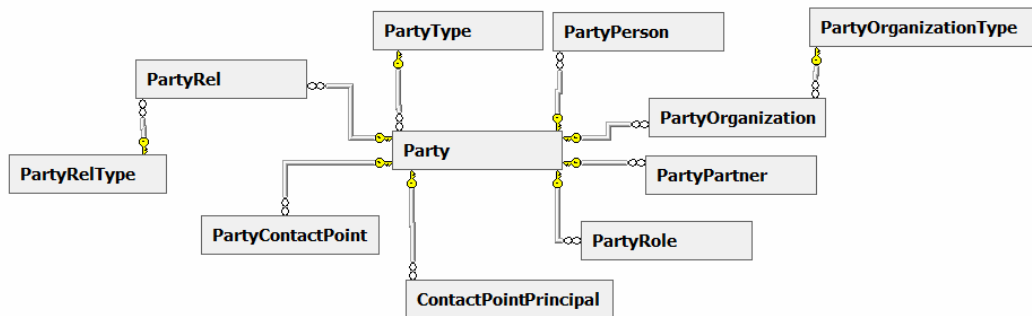


Figure E.6 – Party tables

## Access Entity

The Access tables were used heavily in this project. For the most part, these tables were already in existence at the beginning of the project. Several fields had to be created in the “AccessLogin” table however.



Figure E.7 – Access tables used

- Access – The main Access table. Significant fields include:
  - AccessId – Records created for a LoginName and AccessRole will contain an AccessId.
  - AccessType – Foreign key which indicates what type of Access record it is:
    - Login
    - Role
  - DomainId– Foreign key to the Domain table which domain the AccessId record belongs to (YMCA or Y-Tri Club).

- AccessType – The table which defines what the “Access type” of an Access record is. For example, currently in the system are two Access types:
  - Role – A “Role” Access type means that the Access record in the Access table is a Role. A Role refers to member roles such as “Administrator” and “User.”
  - Login – A “Login” Access type defines a record in the Access table as being a member account. Every *member* in the system has an Access type of Login.

Since Access Roles and Access Logins reside in the same Access table, the AccessType table is beneficial when querying the database for records of a certain type. An additional benefit of this design is flexibility. To create a new AccessType, all that is required is an extra field in the AccessType table.

- AccessRole – The table which holds and defines user roles. Every user must be assigned a role (detailed later). Significant AccessRole fields include:
  - AccessId – The foreign key to the Access table which is used to determine what domain the role belongs to, and to assign a role to a user.
  - Name – This field defines the name of the role.
  - Description – The Description field gives the description of the Role, such as what a member with a particular AccessRole may access.
- AccessLogin –The table which holds user login information. Significant fields include:
  - AccessId – Foreign key to the Access table which will be used to determine what domain the user belongs to, and will be used to assign roles to a member.
  - LoginName – What the user logs into the system with.
  - Password

- PasswordQuestion – This field allows nulls. During account activation, the user can choose to provide a security question in the event they forget their password.
- PasswordAnswer – The answer to the security question
- IsActivated – Indicates (true or false) if the member account has been activated.
- RecvEmails – Indicates (true or false) the member’s messaging preference. A false value means the user does not want to receive emails.
- PublicProfile – Indicates (true or false) if the member wishes to share their information with other members.

AccountActive – Not to be confused with IsActive, this field indicates if the account is active. An inactive account could mean the user is no longer a YMCA or Y-Tri Club member, or if privileges were abused.

- AccessRel – This table performs holds, and assigns roles to users. The fields are defined as followed:
  - AccessId1 – This field represents the AccessId of the *AccessRole*.
  - AccessRelTypeId – A foreign key to the AccessRelType table. This field is used to define the relationship between AccessId1 (the role) and AccessId2 (the user). “RoleContainsUser” is an example of “AccessRelType.” If a role and a user are defined by this AccessRelType then the user is of a certain role, such as “Administrator.”
  - AccessId2 – This field represents the AccessId of the user in the system.

- AccessRelType – This table defines the relationship between a role and a member. This table provides flexibility for creating new types of relationships.
  - Name – The name of the relationship type. For example, a relationship type could be “RoleContainsUser.” This relates a role to a member.

### **Contact Point Entity**

The following “ContactPoint” tables were utilized in this project. These tables were already in existence when the project began. They were created by Charles Laymon of StepQuest, Inc.

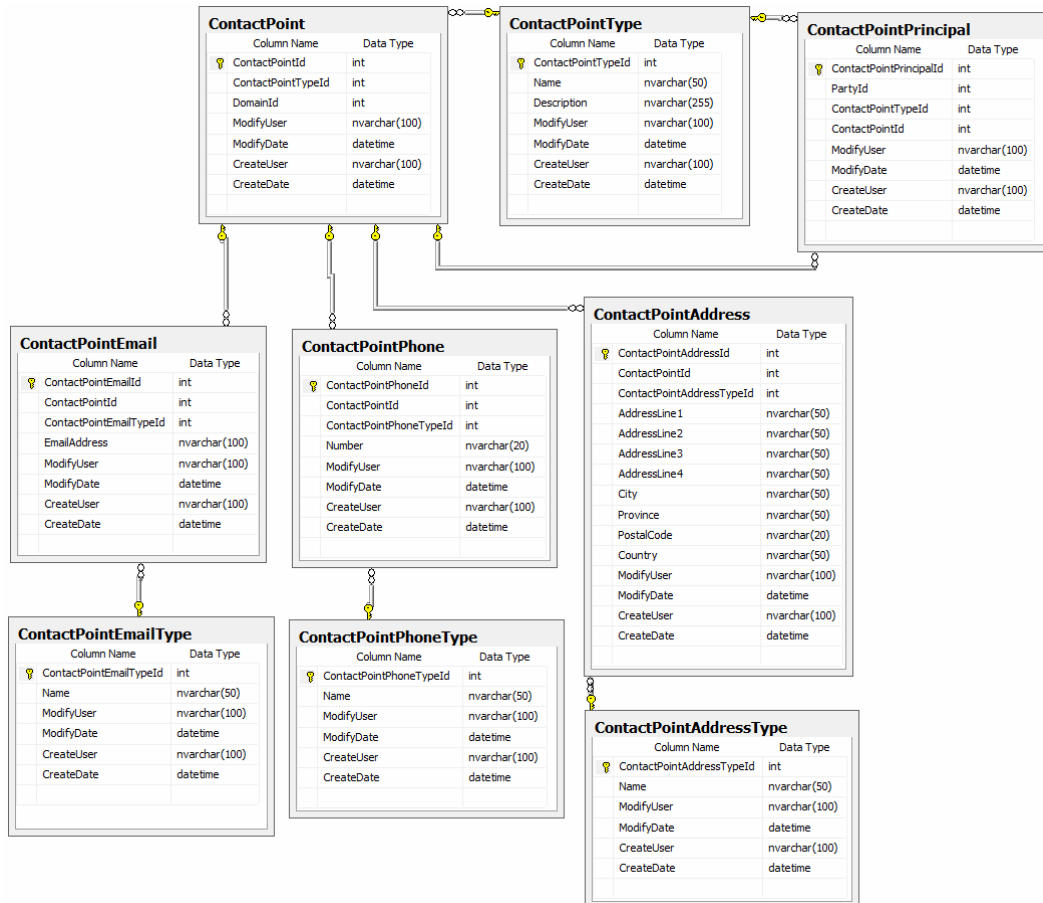


Figure E.8 – Contact Point tables used

- ContactPoint (CP) – This is the main CP table. All “ContactPointTypes” must have a ContactPointId. CP fields are detailed below:
  - ContactPointId – The primary key.
  - ContactPointType – This field details the type of the CP. For example a CP types include addresses, phones, and emails. All CPs must be of a particular type.
- ContactPointType – This table defines the CP types. CP types used in this system are Address, Email, and Phone types.
  - Name – the name of the CP type, for example “Address.”
  - Description – The description of the CP type.
- ContactPointPrincipal (CPP) – This is an important table. This table sets the principal CP for a user. A user may have a principal Address, Email, and Phone type. In the messaging system, messages will be sent to the users primary email address.
  - PartyId – This field is used to relate a CP to a member. All members must be in a “Party.” A Party type can consist of a “Person” or an “Organization.” More details on the “Party” entity will be detailed later.
  - ContactPointType – This is used in the CPP table because a member is only allowed to have one primary CP of type Address, Email, and Phone.
  - ContactPointId – Foreign key to the ContactPoint table. This defines what CP is to be the principal CP for the user.
- ContactPointEmail – This table holds the Email CPs for a user.
  - ContactPointEmailTypeId – The foreign key to the ContactPointEmailType table, which is detailed below.
  - EmailAddress – The user’s email address.

- ContactPointEmailType – For this system, members are allowed to have two types of emails: “Home” and “Work”. “Home” being the user’s home email address, and “Work” being the user’s work email address. “Home” and “Work” are represented in the Name field.
- ContactPointPhone – This table holds the Phone CPs for a user.
  - ContactPointPhoneTypeId – The foreign key to the ContactPointPhoneType table, which is detailed below.
  - Number – the user’s phone number.
- ContactPointPhoneType – For this system, members are allowed to have three phone types: “Home,” “Work,” and “Mobile.” The Name field represents each type.
- ContactPointAddress – This table holds the Address CPs for a user.
  - ContactPointAddressTypeId – The foreign key to the ContactPointAddressType table, which is detailed below.
  - AddressLine1 – The user’s address.
  - City – The city for the user’s address.
  - Province – The state for the user’s address.
  - PostalCode – The postal/zip code for the user’s address.
  - Country – The country for which the user’s address is in.
- ContactPointAddressType – For this system, member are allowed to have two address types: “Home” and “Work.” The Name filed represents each type.

### **Domain Entity**

The following “Domain” tables were utilized for this project. These tables were in existence at the beginning of the project, and required no alterations.

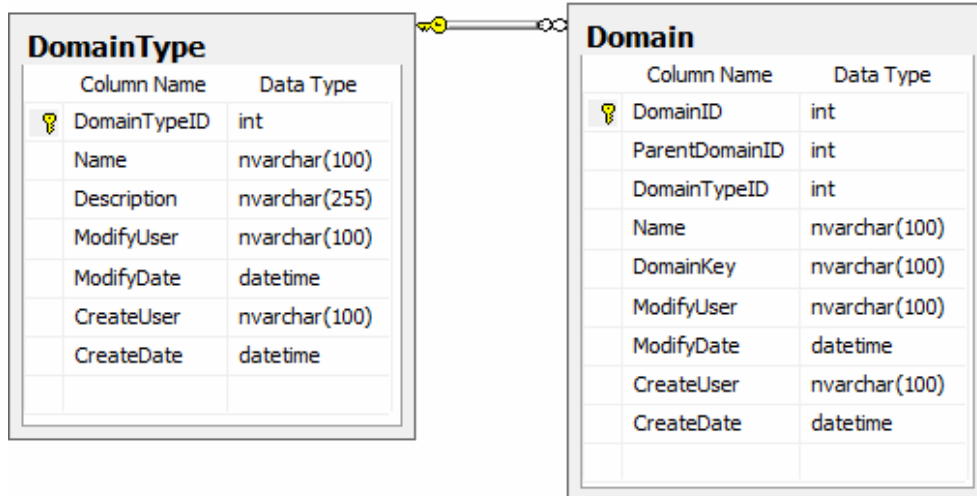


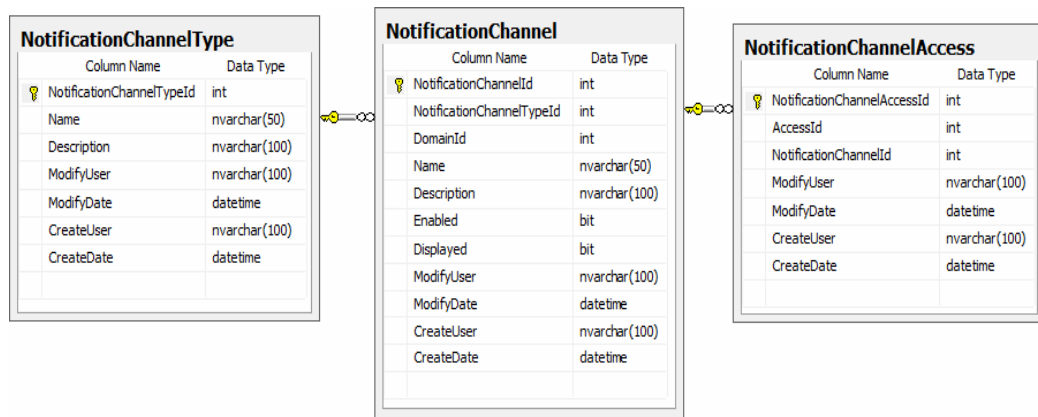
Figure E.9 – Domain tables used

- Domain – A “Domain” table is required because the database holds content for both the Wilmington Family YMCA and Y-Tri Club organizations.
  - ParentDomainID – It is possible to have sub domains, such as “ytriclub.wilmingtonfamilyymca.org.” ParentDomainID defines the parent domain for a sub domain. For example the parent domain for “ytriclub.wilmingtonfamilyymca.org” is “www.wilmingtonfamilyymca.com.” All domains must have a ParentDomainID however. If a domain is a parent domain, then the ParentDomainID field should be the value of the DomainId for that domain. For example, the ParentDomainID value for “www.wilmingtonfamilyymca.com” is the DomainID for “www.wilmingtonfamilyymca.com.”
  - DomainTypeID – The foreign key to the DomainType table, defined below.
  - Name – The name of the domain, for example “YMCA” or “Y-Tri Club.”
  - DomainKey – The address for the domain, for example “www.wilmingtonfamilyymca.org.”

- DomainType - This table defines the type of domain. There are two types of domain types for this project:
  - Primary
  - Sub

### **Notification Channel Entity**

The following details the “NotificationChannel” tables used in the project. I created these tables as they did not exist at the beginning of the project and were needed for the messaging requirement.



*Figure E.10 Notification Channel tables used*

- NotificationChannel – This table is the main “NotificationChannel” table. “NotificationChannel” fields are detailed below.
  - NotificationChannelTypeId – A foreign key to the “NotificationChannelType” table, which defines the type of channel. For example, a notification channel type is “Email,” defining that the message will be an email.
  - DomainId – The domain in which the channel belongs to.
  - Name – The name of the channel, for example “Nutrition.”
  - Description – This field describes the purpose of the channel.

- Enabled – If the channel's enabled field is set to "false," no messages will be allowed to be sent over the channel. A "true" value allows the channel to be active and messages to be sent over the channel.
  - Displayed – Not to be confused with "Enabled," "Displayed" just means that the channel is active, just not displayed on the user interface.
- NotificationChannelType – This table defines the type of channel. For example, the only channel type implemented is an "Email" type. This means the message will be sent through email.
  - Name – The name of the channel type such as "Email."
  - Description – A description of the channel type.
- NotificationChannelAccess – This table relates to the "Access" table. The purpose of this table is to relate a channel to a user.
  - AccessId – This represents the user who is "subscribed" to a messaging channel.
  - NotificationChannelId – This field represents the channel that the user (AccessId) is subscribed to.

### **Offering Entity**

An "Offering" refers to a class or service that the YMCA *offers* to members. The following are the tables used in the Offering entity. These tables were already created by Charles Laymon of StepQuest, Inc. when the project started.

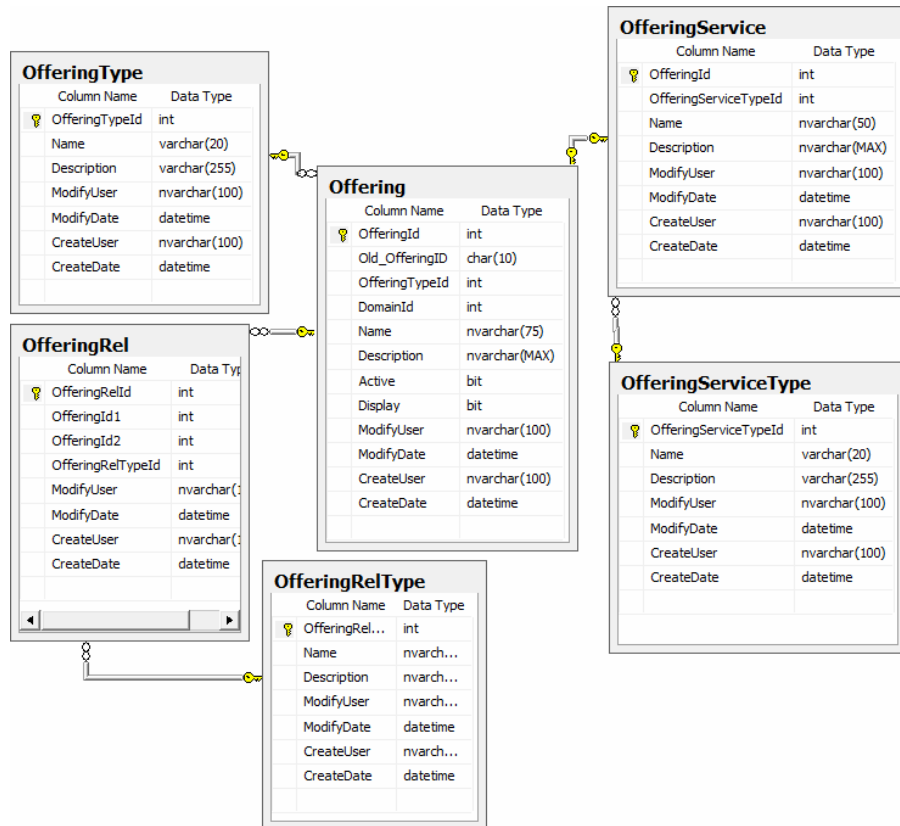


Figure E.11 – Offering tables used

- Offering – The main Offering entity table. Significant fields are detailed below:
  - OfferingId – Each Offering has an OfferingId.
  - OfferingTypeId – Foreign key to the “OfferingType” table (defined later).
  - DomainId – Used to determine what offering belongs to what domain.
  - Name – The name of the offering.
  - Description – The offering described.
- OfferingType – For this system, there are two types an offering could be, they are defined in the “Name” field.
  - Name – The names of the four types of offering types:

- Service – Services are members of a Class. For example, “Karate” is a service in the “Martial Arts” class.
    - Class – A group of a certain type of service, such as “Sports.”
  - Description – Provides a description to the different types of offering types.
- OfferingRel – This table is used to group offerings together by relating services with classes.
  - OfferingId1 – Represents the OfferingId of an offering class type.
  - OfferingId2 – Represents the OfferingId of an offering service type that is categorized under a class.
  - OfferingRelTypeId – Foreign key to the OfferingRelType table, which defines the type of relation (more detail to follow).
- OfferingRelType – This table defines the type of relation an offering has on another, such as relating a service with a class.
  - Name – The name of the type of relation, for example “Contains.”
  - Description – This describes the relation. For example, the description for “Contains” is “One offering contains another.”
- OfferingService – Services represent the majority of the Offering entity. This table holds the information for each service offered.
  - OfferingServiceTypeId – The foreign key to the “OfferingServiceType” table, which defines what type of service it is such as a “Program.” More detail on the “OfferingServiceType” will be provided later.
  - Name – The name of the service offered.
  - Description – The description of the service offered.
- OfferingServiceType – This table further defines the types of services.
  - Name – Provides the name of the service type offered. An example of this is a service of “Program” type.

- o Description – Further details the type of service type.

### **Party Entity**

The Party entity is used for designating a member as being a “person” or as a part of an “organization.” The Party entity was already created by Charles Laymon of StepQuest, Inc. before the project began.

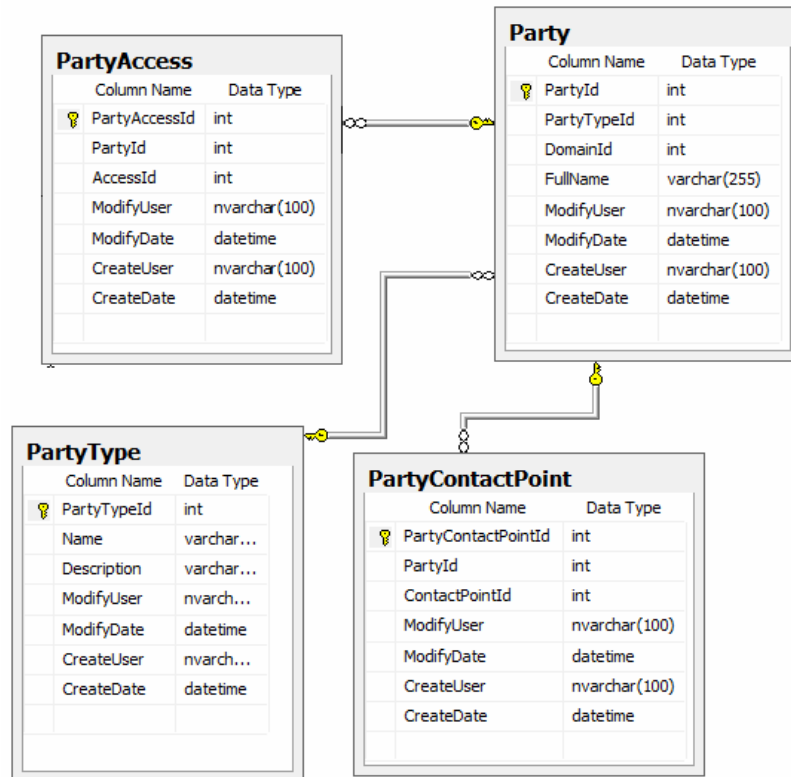


Figure E.12 – Party tables used

- Party – The main Party entity table. Significant “Party” fields are detailed below:
  - PartyTypeId – The foreign key to the “PartyType” table. This defines what type of party type the person belongs to (e.g. “Person” or “Organization”).
  - FullName – The member’s full name (e.g. “Holtsford, Chris”).
- PartyType – This table defines what type of party the member belongs to. There are two Party types: “Person” and “Organization.”
  - Name – The name of the party type.
  - Description – This details the party type.
- PartyAccess – This table is used to bridge the “Access” entity with the “Party” entity.
- PartyContactPoint – This table is used to bridge the “Party” entity with the “ContactPoint” entity.

## Appendix F – System Sequence Diagrams

To better detail some components of the system which could be confusing on how they function I will provide a better description through system sequence diagrams.

The two system components which I will detail are:

- User account activation
- Sending messages through the messaging system

### **User Account Activation**

The Wilmington Family YMCA and Y-Tri Club required that they have full control over who has an account in their system. To provide for such a requirement, we opted to pre-load all members in the registry of both organizations into the system, and send an email to all members with a link to activate their pre-existing account. This way, only paying customers of each group can have an account, not an unknown person in the community. Figure F.1 details the events which take place when a user activates their account. The sequence is as followed:

- The system queries the database for all pre-existing accounts loaded into the system. Once the query is returned, the system sends an activation email to all members. Part of the query returned is the AccessId of the members, which is used to relate their account information. There is a link within the email that uses the AccessId as a parameter. The user follows this link to the system.
- The user follows the link to the activation screen.
- Upon the user's arrival at the activation page, the system checks the account information based on the AccessId parameter.

- The system will verify that the user has not previously activated their account. If the user has previously activated their account, they are forwarded to the login page. If not, then they may proceed to activating their account.
- For activation to be successful the user must:
  - Enter a password twice
  - Enter a security question and answer (optional) if they want to be able to receive their password in the event they forget it.
  - Select the “Activate Account” button.
- The system then will update the user’s activation status in the database, based on AccessId.
- Upon a successful activation, the user will then be forwarded to the login page where they may log into the system for the first time.

### **Sending Messages**

Figure F.2 details the sequence of events which take place when a user wants to send a message to other members. The events of the messaging system are as followed:

- To send messages through the messaging system, a user must first be logged in.
- Upon a successful login, the user may then navigate to the messaging system either through the navigational bars, or via the link in their profile.
- When composing a message the user may select the channels to send the message to, additionally they may include an attachment.
- To send the message the user must select the “Send Message” button (see Figure D.15 on p. 87).

- Once the user selects the "Send Message" button the system loops through the channel subscription check box list, constructing a structured XML data object based on which channels are selected.
- The system uses this XML object as a parameter to a stored procedure in the data access layer. The XML object acts as a table of messaging channels in the stored procedure. We use this XML "table" object in the query by selecting all members' email addresses who are listed as subscribers of the channel.
- The query will return the primary email address of each member that subscribes to a selected channel. *Note: If a member is subscribed to more than one channel, they will only be selected once, so as to not send them multiple messages.*
- Once the database returns all email addresses the system then sends the message out.

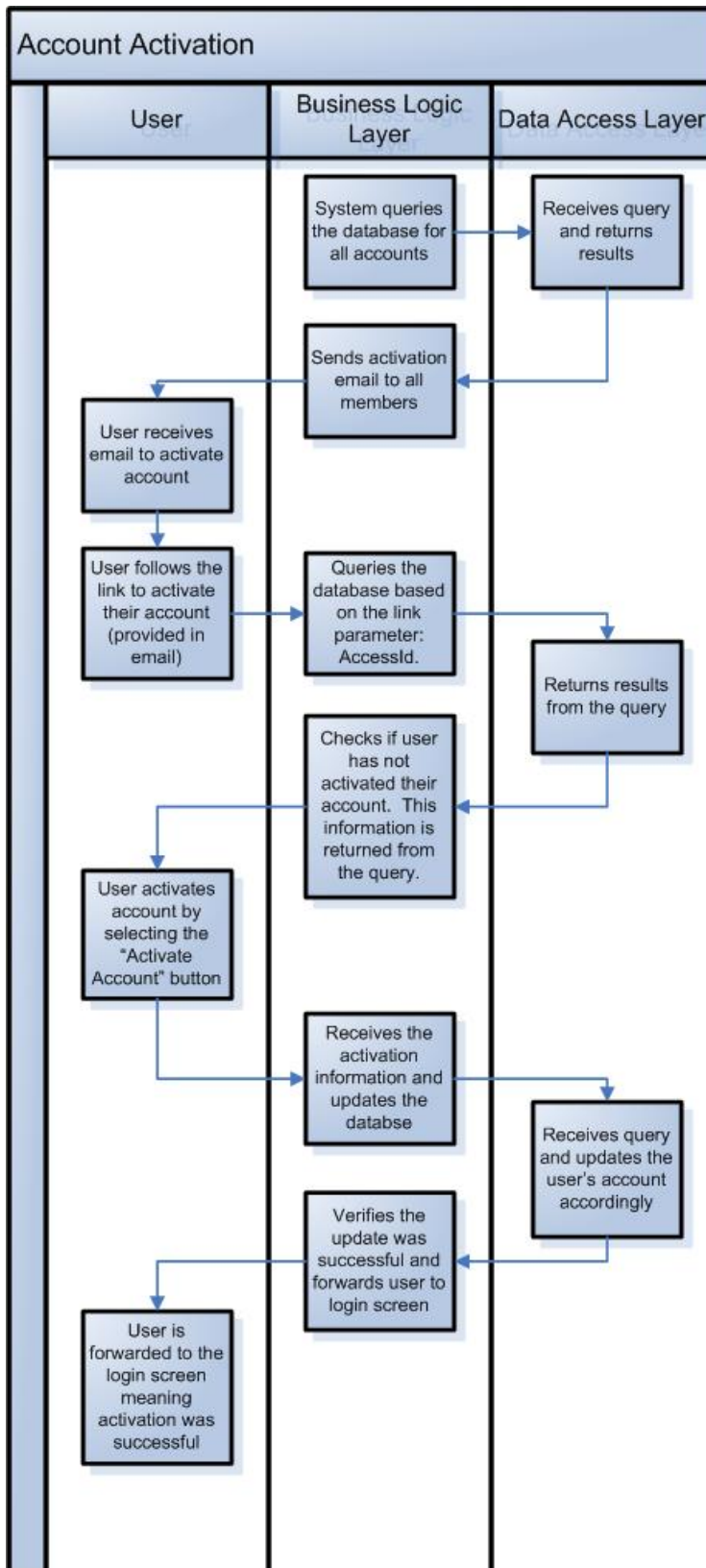


Figure F.1 – User account activation sequence

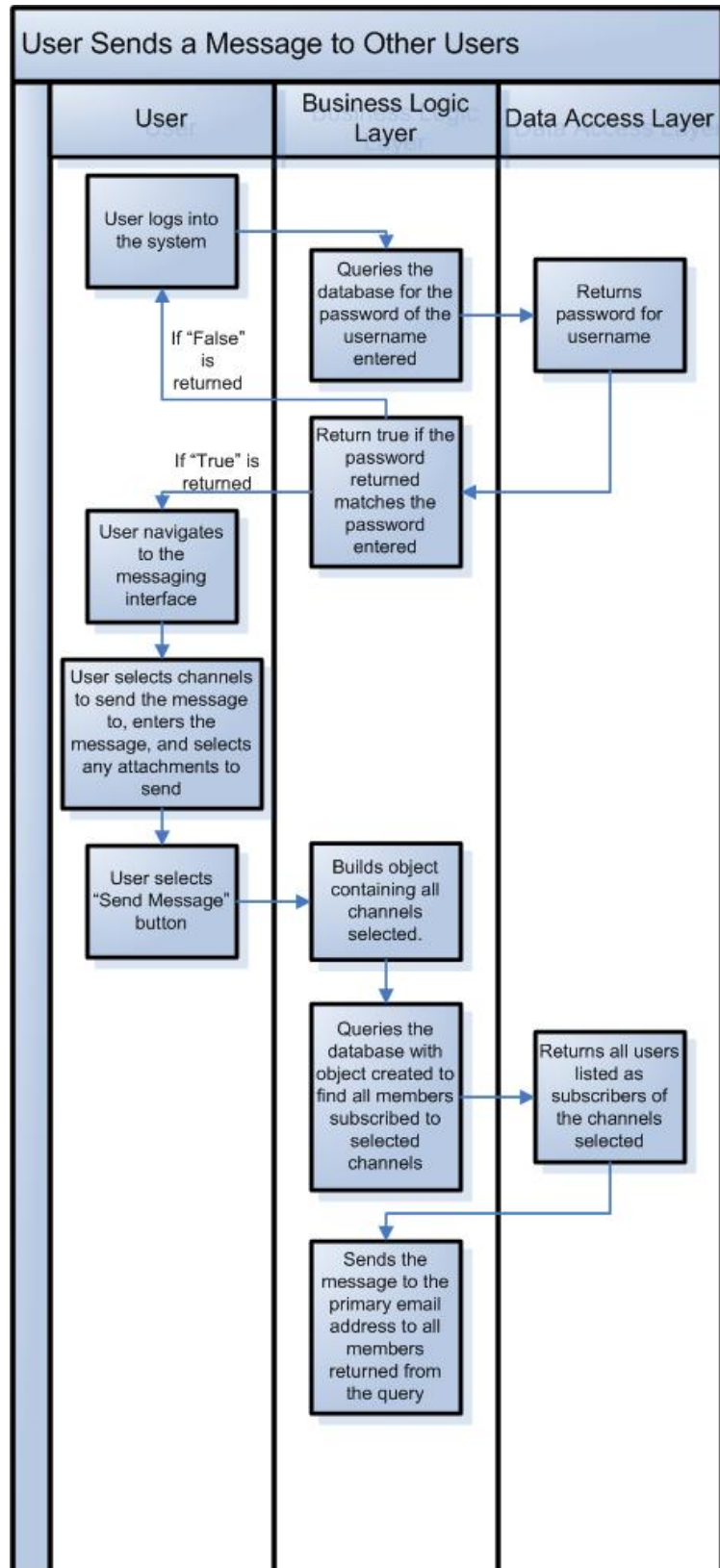


Figure F.2 – Sending messages through the messaging system

## **Appendix G – Project Metrics**

The purpose of this section is to detail the amount of work that is involved in successfully implementing a project of this size.

### **Time**

To successfully implement the new system in the time frame which I had to work in (August 2006 – May 2007), a great amount of time had to be devoted to this project on a weekly basis. The average time spent each week working on the project was approximately 15 - 20 hours.

### **System Metrics**

The following provides numbers for the objects which are all needed to obtain proper functionality. The number of lines of code is not listed because the .NET 2.0 Framework generates thousands of lines of code for each table adapter created, and an accurate count cannot be given.

#### **Database**

- Total number of tables used: 50
- Total number of stored procedures used: 266
- Total number of functions: 2

#### **Application**

- Total number of user controls: 17
- Total number of Data Access Layer Objects: 15. Each contained an average of 4 adapter objects in each
- Number of Active Server Pages: 18
- Classes: 5

- XML files used for navigational menus: 5

### **System Scalability**

There is a threshold on the amount of data stored in the system. This is a drawback to using Microsoft SQL Server 2005 Express. Microsoft SQL Server 2005 Express was used because the non-profit organizations, the Wilmington Family YMCA and Y-Tri Club, cannot afford to purchase the full distribution. Even though Express has less functionality than the full version of SQL Server 2005, Express can be used and redistributed without licensing fees [11]. The following are the limitations of Microsoft SQL Server 2005 Express:

- It supports one CPU at a time. Queries cannot be run in parallel [11]
- It cannot exceed 1 GB of RAM at a time for queries [11]
- There is a maximum database size of 4 GB [11]
- There is no support for analysis or reporting services [11]

Express supports the needs of the two organizations for now. If the systems are to grow significantly, we will need to purchase a license for the full version of SQL Server 2005.

## References

- [1] *XHTML 1.0: The Extensible Hypertext Markup Language (Second Edition)*. August 1<sup>st</sup>, 2002. W3C. February 9<sup>th</sup>, 2007. <<http://www.w3.org/TR/xhtml1/>>
- [2] *Watchfire WebXACT*. 2004. Watchfire Corporation. March 13<sup>th</sup>, 2007. <<http://webxact.watchfire.com/>>
- [3] Vieira, Robert. *Beginning SQL Server 2005 Programming*. Indianapolis, Indiana: Wiley Publishing, Inc., 2006.
- [4] Satzinger, Jackson, and Burd. *Object-Oriented Analysis & Design with the Unified Process*. Boston: Thompson, 2005.
- [5] *NUnit*. 2006. NUnit.org. April 15<sup>th</sup>, 2007. <<http://www.nunit.org/>>
- [6] *Unit Testing*. April 27<sup>th</sup>, 2007. Wikipedia.org. April 15<sup>th</sup>, 2007. <[http://en.wikipedia.org/wiki/Unit\\_testing](http://en.wikipedia.org/wiki/Unit_testing)>
- [7] Alavin, Maryam. "An Assessment of the Prototyping Approach to Information Systems Development." ACM. Volume 27. Number 6 (1984). pp. 557-563.
- [8] Guimaraes, Luciano and Vilela, Plinio. "Comparing Software Development Models Using CDM." ACM. 2005. p. 345.
- [9] *Scope Creep*. April 26<sup>th</sup>, 2007. Wikipedia.org. April 15<sup>th</sup>, 2007 <[http://en.wikipedia.org/wiki/Scope\\_creep](http://en.wikipedia.org/wiki/Scope_creep)>
- [10] *MS Patterns & Practices*. Microsoft Corporation. 2007. April 17<sup>th</sup>, 2007. <<http://msdn2.microsoft.com/en-us/practices/default.aspx>>
- [11] *SQL Server 2005 Express*. April 25<sup>th</sup>, 2007. TechTarget. May 1<sup>st</sup>, 2007. <[searchsqlserver.techtarget.com/tip/1,289483,sid87\\_gci1082707,00.html](http://searchsqlserver.techtarget.com/tip/1,289483,sid87_gci1082707,00.html)>