Spring 2005

Design of North Texas Pc Users Group Ecommerce Interface and Online Membership System

Jeir J. Steele
Regis University

Regis University
School for Professional Studies Graduate Programs
Final Project/Thesis

Disclaimer

REGIS UNIVERSITY

SCHOOL FOR PROFESSIONAL STUDIES

MASTER OF SCIENCE
IN
COMPUTER INFORMATION SYSTEMS

Design of North Texas PC Users Group

Ecommerce Interface and Online

Membership System

PROFESSIONAL PROJECT

Jeri J. Steele

December, 2005

Regis University
School for Professional Studies
MSCIT Program

Abstract

**Design of North Texas PC Users Group
Ecommerce Interface and Online
Membership System**

by

Jeri J. Steele

December 19, 2005

The North Texas PC Users Group is a non-profit that is struggling with their existing

membership process. This thesis explores the analysis of a new membership process and

discusses the resulting new architecture and system design to implement it. The

implementation of this new business process will reduce the time to produce the standard

monthly reports and foster future membership retention efforts. The new design supports

Ecommerce instant transactions, online availability of reports for the Board of Directors,

and the access control of a member-only online site. The research for this project

included an analysis of the cost, benefits, and features needed for online credit card

transactions by a small non-profit organization. The presented architectural design

supports an n-tiered distributed application, over an underlying relational database which

will ensure the membership information is safe, accurate, and timely while supporting

future performance, scalability, and reliability needs.

Acknowledgments

My heartfelt appreciation goes to the Board of Directors of the North Texas PC Users Group. They had to live through the many ideas and frustrations of trying to design an Enterprise-strength application part-time, while I continued to work full-time at Texas Instruments. Without the many hours of reviews and design specification drafts, this paper would not have been possible. I thank each and every one of you for your hard work and support during this period. It is my hope that our work will be used by NTPCUG to benefit all members.

I am also very appreciative to my supervisor at Texas Instruments, Chester Larned. He has supported my completion of this degree in many ways. Not only did Texas Instruments pay for the tuition, they provided an environment that allowed me flexible time off and a place to try out some of my newly acquired skills. I couldn't ask for a better place to be employed while working on this Masters degree.

A big "Thank You" goes to my thesis advisor, Professor Tim McKenzie, for his assistance and patience. Mr. McKenzie was very supportive and encouraging, especially when I experienced within the same year, the personal challenge of having my house flooded by a broken hot water heater and a husband hospitalized with heart problems, while I worked on this project and paper. Thank you for all that you have done.

I am indebted to the various Regis University Faculty members that were professors in my online classes. The curriculum challenged my thinking and helped to expand my skills in writing, in addition to teaching me Computer Science.

To my son, Zachary Loafman, your excellent work as a Computer Science major at Carnegie Mellon University was a major inspiration to me going back to finish my graduate work. Having you succeed so well made me value our many hours of 'playing' together on computer systems when you were young.

Finally, to my husband, Bill Dow, thank you is inadequate for cheerfully supporting the many hours I had to spend on my school work. You were always encouraging. I especially valued your help in keeping the house going and representing us at various meetings, while allowing me time to study and write. You are truly my best friend and the love of my life.

Table of Contents

Figures

Tables

Chapter 1 - Introduction

The North Texas PC Users Group (NTPCUG) is a non-profit organization with over 450

members dedicated to education on personal computers (PC) and PC applications.

There are dozens of Special Interest Groups (SIGs) that use online areas to post technical

information in a member-helping-member format. The SIGS meet monthly and this

monthly meeting is where most members pay their yearly dues. The current membership

lists are used to control who receives raffle tickets and other special member-only

promotions. They are also used to control who gets email aliases and is admitted to a SIG

forum area.

The current Membership Director maintains the membership database on a Windows-

based Access database and produces reports for the Board of Directors (BOD) monthly.

Even though input transactions are initiated on the web, information is entered in a

manual process using a Microsoft Access database. This process is error prone and takes

many hours of the Membership Director's time.

The current web page for joining or renewing a membership runs on a Unix-based

Apache web server. Recently NTPCUG was donated a copy of WIN 2003 Server which

includes Windows SharePoint Services (WSS). A team has been formed to implement

expanded member-only services on WSS. NTPCUG will use this new member-only area

to enhance membership benefits by allowing members to upgrade their technical skills

and to exchange technical information. The SharePoint Portal server team will need

membership information to control access to these member-only areas. With all the other

online applications moving to a Windows environment, the BOD has asked if possible to

move the Ecommerce transaction scripts to an IIS Windows server to save money on the

server leases.

Current Process and Problems

The current NTPCUG membership director maintains the membership database and

produces reports for the Board of Directors. Input comes from application forms (hard

copy) and from text in an email generated from PayPal.com[1] via Perl Ecommerce

transaction scripts. Information on new members received in an email then must be

entered into an Access database data entry screen by the Membership Director. Any

membership renewals received in the snail mail are forwarded once a week to the

Membership Director by the Treasurer. The data entry process is error prone due to the

lack of design for the user interface. Similarly any member information updated online

also comes as an email and also has to be entered by hand into the Access database by the

Membership Director.

Once a month the Board needs access to information, such as expiring memberships,

yearly membership retention trends, and the current list of members to track attendance

and send email. Tracking membership trends is crucial to the health of the organization.

Due to the time involved in processing updates and sending out membership cards and

renewal notices, the reports are being generated only once a month, during the first week

of the every month.

Member information updated by online transactions, the generation of ad-hoc reports,

and the updating of membership information needs to be almost instantaneous to better

support the running of the organization. An improved membership process will save the

[1] PayPal is a registered trademark of PayPal, Inc.

Membership Director time in the short term, improve data accuracy, and will expand future services for the members.

Review of Existing Technology

The existing NTPCUG database is implemented in Microsoft Office Access 97 on hardware that is described in Appendix B, "

Future System Requirements". Due to changes Microsoft made in Office Access 2003 software, much of the existing application code will no longer run. The design of the current database tables has evolved over time. There are fields in some database tables that are no longer being used or updated. These tables are not normalized and therefore have not been synchronized with other tables that share the same information.

Project Goals

The highest priority project goal is to revise the current membership process to reflect the current business process and make it more efficient. Additionally, other goals include:

- Moving the North Texas PC Users Group membership process to a new systems architecture to support future membership and organizational goals

- Reducing data entry errors by 20%

- Working with the SharePoint Portal server team to control access to a member-only area

- Providing a way for members to review and update their personal information in a secure and robust way online.

Issues and Limitations

The membership system must allow:

- Any number of payments for each membership. Each member's payments will be tracked over the years and multiple payments may be made within a year.

- Support of multiple different categories of memberships with different dues payment amounts for each category.

- Use of the membership expiration date to prevent non-members from accessing certain areas online.

- On-demand reports for authorized people such as the Board of Directors, or specifically only the Membership Director or Treasurer.

- Members to update personal information such as emails, address, and telephone numbers.

- Members to have multiple emails and phone numbers such as home, work, cell and fax numbers.

- Changes to be queued for review and then authorized by the Membership Director or another duly-designated person.

Software Requirements are documented in Appendix A, "

- Future System Requirements**"** with the cost of any software used to be under $250.

- Development to be done by volunteers who have or can acquire current systems skills on Microsoft Windows.

Project Scope

The scope of this project is to gather the requirements, to review the membership business process, to design the overall architecture for the membership system, to develop the project plan and to write the Membership System Specification including the database design for the North Texas PC Users Group. This design will consider security and reliability issues and be versatile enough to be implemented on any relational database. In review of the business processes various Ecommerce providers will be evaluated against the current provider, PayPal.com, to ensure that the needs of NTPCUG are still being met by PayPal.com.(Paypal.com, 2005)

Definitions of Terms

Throughout this thesis there are a few acronyms used to define terms. Among them are:

ACID – atomic, consistency, isolation and durability; the four critical transaction properties

BOD – Board of Directors

DSL – Digital Subscriber Line

ERD–Entity relationship diagrams

GUI – Graphical User Interface

HTTP–Hypertext Transfer Protocol

IIS – Internet Information Server

MD – Membership Director

MSS –Membership Support System

MHz – Megahertz, Computer Clock Speed

NIC – Network Interface Card

NTPCUG – North Texas PC Users Group (http://www.ntpcug.org)

PC – Personal Computer

PDA – Personal Digital Assistant

RAM – Random Access Memory

SDLC – Software Development Life Cycle

SIGs – Special Interest Groups

SSL – Secure Sockets Layer

WSS – Windows SharePoint Services[2]

Summary

This project is the culmination of over a year of research and work with a team to design and replace the current membership business process and database. The academic research and review of current topic material is discussed in Chapter Two. The third chapter describes the methodology for this project. Chapter Four contains the history of the project and the overall results. Chapter five includes the lessons learned from the project and the recommendations for the future.

---

[2] Microsoft, Windows, Outlook, Access, and SharePoint, are either registered trademarks or trademarks of Microsoft Corp. in the United States and/or other countries.

Chapter 2 - Review of Current Products, Literature & Research

Literature Overview

The literature on Ecommerce design is very prevalent. Even though the field is relatively new, there are a number of good books, current articles and white papers on Ecommerce design. The white papers consulted were primarily to evaluate Ecommerce vendors and the criteria to use in their evaluation.

Research of Relevant Literature

The first book consulted for this project was Barry Boehm's *Software Engineering Project Management*. For software projects without a clear choice of technology, the chapter on the Spiral life cycle model of software development gives a way to plan projects in small, manageable phases. Additionally, Boehm's online paper entitled, "Spiral Development: Experience, Principles, and Refinements" added insight into the use of this model type for Ecommerce and for other types of applications as a risk-driven process. To construct the project plan, Richard Fairley's *Software Engineering Project Management: a Guide for Preparing Software Project Management Plans* was the most useful for the Project Outline and for quality control of the Project Plan.

To review the business process and create an overall Ecommerce project architecture, several books and articles were consulted. Peter Fingar, Harsha Kumar, and Sharma Tarun's *Enterprise Ecommerce* was an excellent book for an overall strategy to implementing Ecommerce for the organization. In particular, the discussion of leveraging

current assets was used. In reviewing the business model changes for NTPCUG, Ravi

Kalakota and Marcia Robinson's Book *E-Business 2.0: Roadmap for Success* was used to

review how n-tiered Ecommerce systems are designed to support data warehousing and

using business intelligence reports to combat customer churn. The problems being

addressed by the organization is the same as that which a company would experience

with customer churn. J. Dowling's article, "The Art and Science of IT Architecture

Design" presented a step-by-step methodology for extracting the business requirements

for an IT system. Luke Hohmann's book *Beyond Software Architecture: Creating and

Sustaining Winning Solutions* presented information on designing systems to manage

dependencies, defining the business models and transactions of the system and managing

the risk through security. Additionally, this book provided a major insight into how to

break the project into implementation phases and how to get the best benefit from testing

in a multi-layered environment. Chapter 5, "Architecture for Electronic Commerce" in

The Business of Ecommerce by Paul May has excellent examples of choices for

architectural layers in Ecommerce system. Also from that same book, Chapter 4, "The

Technology Landscape", provided several models for electronic payments. In particular,

the information on a low volume of transactions was used.

Further research was done online on types of credit card payments that can be done for

small businesses. The best survey article used was James Maguire's article, "Making

Small Payments Practical for Small E-Tailers" from the website,

http://SmallBusinessComputing.com . The PayPal.com website has numerous resources

including merchant payment options, white papers, and technical manuals.

Once the overall architecture of the project was decided, Arthur Langer's *Applied Ecommerce: Analysis and Engineering for Ecommerce Systems* provided a major source of information on applying software engineering techniques to a web-based Ecommerce system. The hybrid software design model, how to gather requirements, how to design relational databases, and how to design transaction processing to support web applications were incorporated from this book. Anup Ghosh's book *E-Commerce Security* provided the bulk of the information on security issues for Ecommerce. For a short review on database modeling and normalization, Lannes Morris-Murphy's Oracle 9i: *SQL with an Introduction to PL/SQL* was consulted. To evaluate the tools to be used for the implementation, Mike and Joline Morrison's *Database-Driven Web Sites* provided excellent details on implementation of Web GUI with the database backend. Especially valuable was the explanation for using dynamic pages on a Windows IIS server, since most of the NTPCUG web developers had previously used Apache on UNIX. Also consulted was Hyatt's online article, "N-Tier Application Development with Microsoft.NET."

Project Summary of Topic Knowledge

Effective businesses and small non-profit organizations of the next decade will be customer-focused and perform efficiently as well as creatively. To accomplish that, we have to translate our business processes from a paper-based workflow to a digital form. Our success depends on how the system is used to deliver services faster, cheaper, and with better quality.(Kalakota, 2001)

The enormous task of extracting the requirements for an IT system is quite large. With Ecommerce systems, the focus is both on gathering the requirements for the business

processes and on the system that is designed to implement them. When exploring the

requirements, first focus on the infrastructure, then on the movement of the data, and

finally on the business transactions themselves. In order to succeed in the design of an

Ecommerce system, it is fundamental to understand what is driving the enterprise to

change and what business process is being automated.(Dowling, 2002)

Ecommerce systems need to be flexible enough to support changing environments and

workloads. The website or server demand can change at any point in time from very low

use to a very high use in a blink of an eye. Several groups may be using the same

database that provides the information to the Ecommerce systems, which can translate

into an uneven or unpredictable usage pattern. The earliest Ecommerce systems were

separated into the client and the server to try to address this need. The client-server

architecture separated the data management from the user interface. However, having

only two layers suffered from many problems, the largest of which is that the business

logic could reside in both layers and therefore was not easily modified or replaced. Now

the most common system architecture for business applications is to decompose it into

multiple layers consisting of at least three separate pieces. This architecture is referred to

as an n-tiered architecture.(Hohmann, 2004)

The advantages of an n-tier model include:

- Supports management of complex problems by limiting the impact of changes through the separation of the data and business rules

- Shares common services among several applications or projects

- Supports migration of the database and the various application layers to future technologies without having to replace the entire system

- Supports addition of processes, hardware processors, or additional servers to support performance, reliability, and scalability of the system by allowing replication

The n-tier model also has some disadvantages. These include:

- Additional effort is required in all stages of the software development to address the design issues

- A disciplined approach to development is needed to produce a good scalable system

- Development teams may not have experience in the technologies chosen since there is very little literature on the development of n-tier systems

- Testing of n-tier systems is more complex (May, 2000)

The key to producing sound, layered system architectures is to have clear defined system responsibilities, stable interfaces between the system layers, and well-defined encapsulation within each layer. Users of the system should not be able to tell which processes are running where and in which layer. All the users should care about is having the system perform the business functions well. To protect the data and the application's

security, the minimum number of interfaces should be exposed in these layers. (Fingar, 2000; May, 2000)

The basic three-tier client-server architecture is composed of the following layers:

- Presentation layer or also known as the Graphical User Interface, GUI, which runs on a client machine such as a PC
- Middle layer which is the business logic running on the application server
- Data layer which is the location for the database or other persistent data store and is usually located on a back-end server

The presentation layer presents information and manages the interaction usually in a piece of software such as a web browser running on a personal computer. Another example of a GUI layer is a handheld computing device such as a PDA. The GUI does not contain any of the business logic. In some Enterprise Applications the user interface may contain two layers, one for the presentation of the information and another that interfaces the workflow between the GUI and the middle layer.

The middle layer is used to implement an Enterprise application logical view of the data. This layer is where the application business rules are implemented and should be independent of how the data is physically represented. This layer can also be further split into more than one layer, such as a services layer and a domain layer. The services layer would define the services that are visible to the rest of the system. The domain layer is entirely optional and would be used when the business rules are too complex to be represented in one middle layer such as when making a logical view from two legacy relational databases.

Finally the last layer is the persistent data layer which is usually a database management system such as a relational database. Ecommerce systems may also create a layer to manage the mapping between Enterprise domain objects in the middle service layer and the objects in the relational database. Triggers and stored database procedures may be included in this layer to encapsulate that mapping.

There may be some confusion over the number of layers to implement in a system. What drives the separation of a layer is the clear separation of responsibilities and system interfaces. If each separate layer is designed properly then it can be implemented by a team independently and in parallel to the other layers based on those system interfaces. Another major consideration of Ecommerce systems is the management of transactions. Transactions are simply an agreement between a buyer and a seller in exchange for a payment of some kind. How that is implemented in an Ecommerce system is a lot more complicated. A transaction is an event that cuts across all the layers in the system and, if interrupted, can cause an inconsistency in files or databases. All transaction in Ecommerce systems must be logged and a mechanism must be in place to reverse the changes that have been made if the entire operation is not successful. Transaction control will have to be incorporated in not only the database but often in the middle layer.(May, 2000)

Systems involving Ecommerce transactions are usually viewed as an Enterprise application implemented in a layered architecture as described above.(Langer, 2002) Just as managing a company takes organizing people and groups around a business process, taking an Enterprise viewpoint requires organizing the application software around those same business processes. To create an Ecommerce system, start with a good ebusiness

design as the first step. Technology alone will not solve the problem; the Ecommerce technology has to be well-designed and well-executed.(Kalakota, 2001)

Summary and Contribution to the Field

NTPCUG is a non-profit organization which supports leading edge educational opportunities pertaining to information technology. A large portion of the members are also programmers. As a result, there is a wealth of programming talent available as volunteers. As new ideas are generated for giving members more services, there WILL be changes to the membership system. Even though the current business processes has an Ecommerce component using PayPal.com payments, there are still too many manual processes to completely prevent data entry errors. The new system will enhance the membership process for NTPCUG and contributes in two major ways:

- As an example of an implementation approach for a flexible, cost-effective layered architectural design for a small non-profit organization using Ecommerce processes to enhance their member benefits.
- This thesis will also present a methodology and criteria to use in evaluating vendors for low-volume online Ecommerce transactions.

Chapter 3 - Project Methodology

Software Development Life Cycle Model

The traditional software development life cycle, SDLC, is a series of sequential software development steps, each independent of the previous step. No succeeding step can start until the previous one completes. This life cycle is also known as the waterfall development model because the Gantt chart of this approach resembles a waterfall flowing from feasibility, to analysis, to design, to development, to testing and finally, to implementation. The waterfall model is best used with development of systems with known technology.

Many aspects of this project are new for the team or the choice is unknown at the start of the project. Additionally, WSS is a relatively new technology. Interfacing to the SharePoint Portal, in addition to handling a new method of communicating about the Ecommerce transactions from our vendor, creates major risks to the project. To handle these risks, Boehm suggests a spiral model of SDLC. In the Spiral model, team would use prototypes to mitigate some of the project risk.(Boehm, 1988) As each risk is mitigated, part of the unknown becomes known and the development team can move forward in the SDLC. The various teams on the system can be working on different phases of the SDLC without waiting on other teams or on even on the previous phase to be fully completed before starting to prototype in a phase. This type of software development model is best used to guide very risky, software-intensive systems.(Boehm, 2000) The methodology allows the team to incrementally grow the project while decreasing the degree of risk as

the project progresses. To mitigate the risk our project will create prototypes for the

following parts of the project:

- Several scripts for the different transaction types from the preferred

  Ecommerce vendor

- Several scripts to convert the format of the old database tables to the new

  database table format

- Several server configurations to test system performance by measuring

  response time to the most common GUI choices

- Several server configurations to stress test the system. This will be

  accomplished by simultaneously accessing a menu item from a GUI which

  will generate the longest query to the database. The same scenario will

  also be repeated with the scenario that will retrieve the largest amount of

  data from the database

- A small one menu item GUI screen will be generated as soon as possible

  to test the difficulty of implementing the application layer using

  Microsoft's Visual Studio.NET and to explore the Integrated Development

  Environment, IDE.[3]

- A prototype of server-side scripts for .NET will be generated to

  understand the interface needed to implement and run them

The Spiral Model is a lot less formal than a waterfall model in that a complete set of

documents is not needed to start the next phase. Thus the Spiral model is much more

adaptable to change. This will gives NTPCUG the flexibility needed to make the final

---

[3] (Morrison, 2003)

choices on the implementation language and n-tiered architecture. The Spiral model will also allow the various teams to not delay each other's schedules since each phase does not have to wait on the previous one.

For the relational database implementation in the data layer, the logical view of the data is separated from the physical data storage fields. The methodology to do this separation is the old structured analysis and design methodology, better known as the waterfall method. Data Flow Diagrams, DFDs, are constructed to show the flow of data into the system. Then the logical data modeling process produces the needed information to construct the data dictionary and the entity-relationship diagrams, known as ERD, which form the basis of the database tables. Then the data must be normalized. Relational databases must use tables that have been normalized; otherwise, the integrity of the database is not preserved. What is accomplished by the process of normalization is that redundant data is removed. (Langer, 2002; Morris-Murphy, 2003) This is the accepted practice for designing a relational database.

Langer suggests that **both** the waterfall and Spiral methods are useful for Ecommerce systems and refers to using both in the same system as the hybrid life cycle.(Langer, 2002) For this reason we have chosen a hybrid life cycle approach. The database tables will be designed using ERD and the normalization techniques. The Ecommerce transactions methodology will be rapid prototypes, and the overall phases of the development will be the Spiral model. (Fairley, 1988; Fingar, 2000)

The life cycle of this project contains six phases:

- Requirements and Business Analysis

- Domain Analysis and Logical Architecture Design (using prototypes)

17

- Development

- Testing

- Implementation

- System Maintenance and Support

Phased Implementation Details

Phase I – Requirements/Business Analysis

Requirements will be gathered on: membership process, GUI, reporting, Ecommerce transactions, WSS interfaces, data flow diagrams, and security.

Output:

- New business process, data flow diagrams and use case scenarios

- Define subsets of functionality for implementation phases based on the GUI functionality

- Project Plan document

- Preliminary Software Requirements Specification and Architecture document.

Phase II – Domain Analysis and Logical Architecture Design

In phase II, decisions will be made on software tools to be used and the logical architecture. The preliminary Ecommerce transaction and GUI design will be completed. Both the Software Requirements Specification and the Software Development Plan will be reviewed with the stakeholders.

The logical view of the database through ERD will be reviewed and completed. The logical architecture design will be completed. Analysis of the Ecommerce vendor needs and costs will result in a choice of the Ecommerce vendor.

Output:

- New member database schema

- Prototype for the first implementation phase GUI

- Prototype of membership database conversion script

- Ecommerce vendor comparison charts completed

- Ecommerce transaction design prototyped

- Decisions on software tools to be used and the layers for the architecture design

- Final review of the MSS Software Requirements Specification and the Software Development Plan will be held with the stakeholders.

- Completed the MSS Software Requirements Specification document

Phase III – Development Phase

During this phase, code to handle authentication and authorization security, and interfaces to the WSS will be developed. The relational database will be constructed, loaded, and any stored scripts to retrieve information from database will be constructed. Also to be completed in this phase is the member-only services on WSS, the application tier scripts, and online reports. The Ecommerce scripts will be also be finalized.

Output of this phase:

- Relational database tables tested and loaded with live data

- Stored procedures and scripts generated

- Ecommerce scripts unit tested

- Test Plans, Software Quality Specification, User Acceptance Criteria, and the Business Readiness Plan will be complete.

Phase IV – Testing Phase

Output of this phase will be the results of the execution of the Software Test Plan. All scenarios will result in a thorough examination of the database data, report output, application code, and Ecommerce transaction code.

Phase V – Implementation Phase

This phase is the final execution of the User Acceptance Test from the Business Readiness Plan. The output of this phase will be the execution of the final cutover to production.

Phase VI – System Maintenance and Support

This phase is the ongoing maintenance of the MSS including the membership database, stored database procedures, the application code, and the scripts to support Ecommerce transactions.

Each Phase contains some part of all the software development steps:

- risk assessment

- benchmarks, prototypes or models

- designs or code

- testing

- plans for the next phase

When plotted on a Gantt chart, the timelines become almost like stripes on a Barber pole; in a three dimensional form the timelines form a spiral similar to the example shown in Appendix B, "Software Methodology – Spiral SDLC".

Methods, Tools, and Techniques

The project will use all the above research by implementing a GUI client interfaced to a middle tier of application, and a database tier as the data store. Which software will be used for the application and database will be finalized in the prototyping phases. In the database tier, key business rules that enforce the referential integrity will be programmed using stored procedures and triggers.(Langer, 2002) A major decision to be made during the prototyping phase is the security methods to be chosen for authorization of users and authentication of transactions. Privacy and security issues will be addressed throughout the prototyping phase along with any changes to the structure of the architecture. The testing of the system will be accomplished using spiking as a way of thoroughly testing all layers. In implementation, the portion of the functionality to be tested at each phase will be identified. The finished online system will be hosted on NTPCUG servers. These will either be hosted on or interface to the WSS.

Chapter 4 - Project History

The initial start of the project was started in the fall of 2004. The project team consisted of four NTPCUG members who were the subject matter experts on the membership process and the goals of the organization. Requirements were gathered primarily by the Membership Director since she was the primary super-user of the system.

Business Process Review

After reviewing the NTPCUG membership process, major areas for improvement were identified:

1) There is an inconsistent application of business rules in a number of areas:

   ➢ Assigning of membership expiration dates

   Example: a membership that had expired 6/30/05 was renewed on 8/15/05 with the expiration date set to 6/30/06. If the person asked the BOD, the member was given an exception and renewed with the date 8/31/06 which would have been the date as if they had joined new on that date.

   ➢ Assigning of expiration dates when converting from a regular membership to a family membership

Example: a regular membership that expired 6/30/06 was converted to family membership on 9/30/05 by the member paying only the 'difference' between the regular membership and the family membership. The new expiration date for all family members was sometimes set to 6/30/06 and other times to 9/30/06.

➢ Assigning of expiration dates for memberships that expired over 6 months ago.

Example: a membership that expired 9/30/04 was renewed on 03/31/05. The new expiration date was sometimes set to 3/31/06 and other times to 9/30/05.

As a result of the application of these rules, some members were getting a 'free ride' for several months and NTPCUG missed some income. There was ample opportunity for favoritism and errors in posting an incorrect expiration date.

2) Some payments took as long as two weeks to post. A new process needs to be able to post the payments more quickly.

3) The Board of Directors (BOD) wants updated membership retention information more frequently than a couple of times a month.

4) Members enter online information into a form that is then emailed to the Membership Director, who then enters it into the database by hand, which sometimes introduces errors. The new system needs to support a member's profile page, where he/she can see all their personal information and update it

as necessary. In addition, the new system needs to require every member to review their own profile at least once a year.

5) The BOD asked that the team consider a design that will save money by consolidating all the application code on either a Windows or on a UNIX server, so that two different operating systems do not have to be supported on the servers.

6) Many errors have been introduced into the database by manual entry of the data. Members are frustrated when information from the web form is entered into the database incorrectly. The new process needs to have more error checking and less manual entry of data.

Other business needs of the organization were researched. First, the membership has been gradually declining. One way the BOD wants to attract new members is by expanding the membership benefits to include a member-only area online. Members want opportunities to learn new technologies; in response to this need a SharePoint Portal server project has been initiated, but the project requires a current membership list to physically implement access security for the portal. Additionally, the BOD wants inexpensive Ecommerce transactions and instant access to the member-only area; therefore a form of instant transaction notification must be supported. Research will be needed to see if our existing provider, PayPal.com, is the cheapest vendor providing this feature. Since NTPCUG originally started accepting credit-card transactions, there are several new providers of this service.

Proposed New System

To meet the needs of the organization, an online membership process seems to be ideal

for everyone involved. Each member would be able to join, renew, and update their

membership within a few minutes. The members are already familiar with using an

online web browser form. For the Membership Director, the information entered by the

members would not have to be re-entered into a second data entry screen and the reports

could be generated and available to the BOD within minutes. For the BOD, the

membership trends report could be generated ad-hoc and the most recent standard reports

would always be available. This fits the trend of using the Internet to enhance the

relationship with customers (the members).(Daley, 2000)

Other considerations to be discussed about the new system include:

- Analysis of the transaction providers to make sure that PayPal.com is the least expensive while meeting the organizational needs

- Systems architecture to support maintainability, performance, scalability, and reliability issues

- Security to protect the server, the application, the database, and the member's personal information

- Ecommerce transactions in a consistent robust manner

Analysis of Transaction Providers

Online Payment Types

Online merchants today are faced with a myriad of choices when evaluating potential processors of online credit card transactions. Online payments can be accepted through the following ways:

- Micropayment processors

- Low volume transaction processing on the Ecommerce vendor's site

- Low to medium volume transaction processing on the merchant's website

- High volume processors directly hooked into a banking institution.(Daley, 2000) (Maguire, 2004A, , 2004B)

No research was done on the transaction processors that handle offline transactions, although there may be overlap with the merchants evaluated. There also may be more than one provider in each of these categories available; therefore to simplify price comparisons, only one provider per category was chosen.

A micropayment is usually implemented as a stored-value account for the customers (members). Merchants that sell items with less than $2.00 charge per transaction will want to investigate micropayment processors. In that model, each of the customers would place a deposit of say $40 in their account and then individual charges are applied against the escrow held in the account. Merchants that sell digital content, such as news articles or songs or companies that collect fees for a toll road, are examples that would work best in a stored-value account. For comparison purposes, **BitPass.com** was chosen as the representative micropayment provider.

For online transactions greater than $5 and in low volume, there are providers such as

PayPal.com that charge no monthly service fee, but charge a higher per-transaction fee.

Security and customer service are handled by the PayPal site and via the telephone.

PayPal can also handle a more sophisticated interface where the shopping cart resides on

the local website, as well as the one that runs on PayPal's website. (Paypal.com, 2005)

A new provider, **Free-merchant-accounts.com,** was also considered in the comparison.

It is a good choice for greater than $5 per charge with more than five but less than fifty

transactions per month. The drawback to this vendor is that no information was available

about the reputation of this provider. (Free-merchant-accounts.com, Unknown)

Large volume processors of online credit card transactions usually charge a monthly

and/or a Gateway fee along with a very small per-transaction charge. The setup time of

these accounts is longer than the others mentioned, but they also support offline

transactions and handle much larger volumes. For this comparison **Yahoo! Merchant**

was chosen as the large volume vendor. (Yahoo!_Merchant, Unknown)

A comparison was first done on the transaction fees. In Appendix C, "Table C-1 Online

Transaction Fees" is the chart that was constructed to show the various transaction fees

charged by each of the four vendors being evaluated. On the comparison of transaction

fees alone, Free-merchant-accounts.com seems to win. However when considering the

cost of the monthly fees the picture changes. Also, some of the vendors have different

payment fees based on the volume of transactions. For a more realistic comparison, a

chart was constructed on the yearly cost of all the vendors, using several different

payment volumes. See Appendix C, "Table C-2 Transaction Processors Yearly Cost" for

this table. It was still not easy to compare the vendors, so a graphical chart was

constructed. The graphical chart below shows the cost of regular $55 membership

transaction, at various volume levels with the four vendors being compared.[4]

ECommerce Providers Comparison

Yrly Cost — Bitpass, Free-merchant-accounts.com, Paypal.com, Yahoo! Merchant — Transactions per Month (5 / mon, 10 / mon, 20 / mon, 30 / mon, 50 / mon)
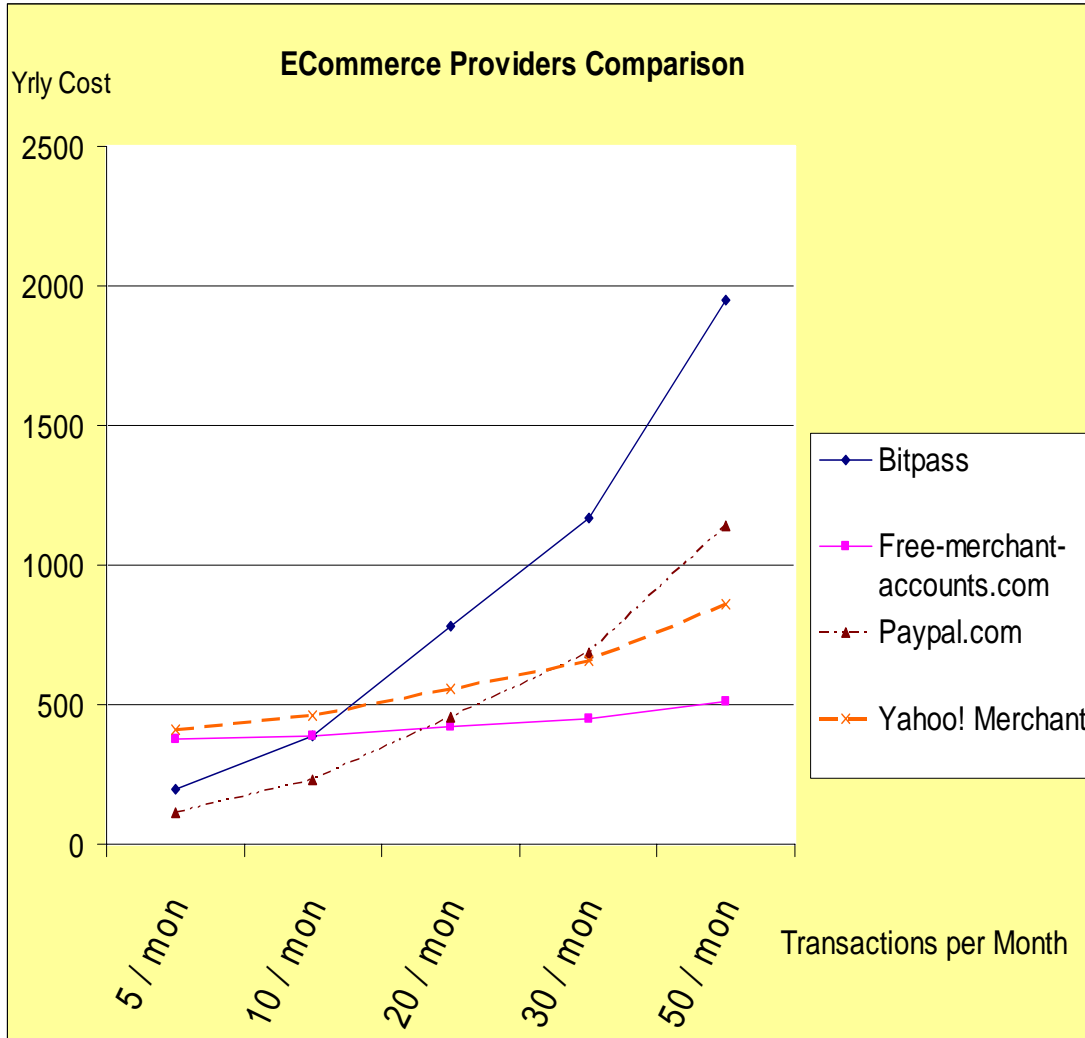
The yearly cost was calculated using the monthly transaction fees (for a regular

membership) at the quantities graphed which were added to the monthly and gateway

fees for one year. By plotting the cost of 5 to 50 transactions for each service analyzed,

we see which service is the most cost effective based on various volumes. **Paypal.com** is

the most cost effective for less than twenty transactions per month. For greater than

---

[4] Note: No set up fees were considered; the monthly and gateway charges were included in the calculations

twenty and less than thirty per month **Free-merchant-accounts** is the best but needs

further research because this seems to be an introductory price. The website for that

company noted that the fee amount used in this calculation was a 'promotional' fee and

was subject to change. When the monthly volume of payments reaches over fifty, **Yahoo!**

Merchant seems to be the processor to consider, since it is the second cheapest cost.

BitPass is not cost effective for monthly amounts analyzed. This service would be best

for micropayments such as songs for $.99.(BitPass.com, Unknown)

Online Payment Vendor Choice

All requirements for payment vendors were gathered and evaluated. This included

requirements for customer service and future needs features such as shopping carts.

Reliability, security, and cost were the major deciding factors. The conclusion was that

our payment provider will remain **PayPal.com**. The information used to come to this

conclusion is summarized in a requirements comparison chart in Appendix C, "Table C-3

Providers Features."

Proposed Application Architecture

The new membership system will be implemented as a client /server layered architecture.

This allows us to place the data and programs in the most efficient configuration. There

will be a minimum of three tiers in the architecture; however the final choice of how

many layers will be made after prototyping is done for the system. The three tiers will be

composed of the same basic ones described above: GUI, the application middle layer, and

the data layer. The business rules will reside in that middle application layer. Keeping the

application code on the middle tier makes it easier to administer and isolates the data

storage to the back-end server placed behind a firewall. This tiered design also supports

security needs that will be discussed later.

During the architecture prototyping, the scalability of the application will be tested. The

basic three layers can be split onto several physical boxes, so that a larger load can be

handled. If the loading does not demand this split, then more than one layer can be

implemented on the same machine. In order to have the flexibility of separating the

layers, several of the three basic layers will likely be expanded during the system design

as shown in Appendix B, "Figure B-2-Future System N-tiered Architecture".

One likely layer to be split will be the data layer. It would be split into three sub-layers.

The lowest layer would be the basic data store, in our case the relational database. The

second sub-layer consists of the set of stored procedures. The third sub-layer will be the

classes that will do the select, insert, update, and delete operations on the database. These

classes would shield the basic data and handle the concurrency issues. This also isolates

any code changes to this layer in case the database changes.(Hyatt, [Unknown year])

Caution will have to be exercised in the design of these sub-layers however. Database

programmers without exposure to n-tiered architecture will want to place the business

rules in the stored procedures and triggers. This was considered a good programming

practice in the past. A better approach with a multi-layered system is to define a set of

services to connect to the database using a standard programming language such as SQL.

If you change the relational database vendor, then the SQL dialect, stored procedures and

triggers are likely not portable. Therefore, stored procedures or triggers will be used only

if absolutely needed.

However, there are times that placing stored procedure into the database makes good sense in this architectural design. The first reason would be to support a performance need. One such example would be when the system must iterate over large volumes of data quickly from another layer. It simply is not practical to move a lot of data out of the database to the application layer. Thankfully, this does not seem to occur in the current business process. A more likely need in this system would be to enforce the integrity of a set of operations. An example that will occur is deleting a payment transaction from the pending queue and posting it to an individual membership. To ensure the consistency of the database, these operations must be done as one atomic operation.(Hohmann, 2004) Another layer that might be split is the presentation layer. The presentation layer is where everything related to the presentation of the windows running on the client PC or the web form resides. It perhaps will be split into both a GUI and a sub-layer to handle the workflow. An example of where this might apply is a user selecting to enter the application to record volunteer hours or selecting to update personal information. Both of these choices will invoke a different application which then operates on the same database. Other potential uses for a workflow controller would be in handling messages to and from a messaging queue, web services or code to communicate to another system. All of these would reside in this layer. The system logical architecture and how these many layers might be mapped on to only a client and two servers is in Appendix B, "Figure B-2-Future System N-tiered Architecture".

Membership Database Redesign

The current membership database was custom code written in Microsoft Access on an ad-hoc set of tables. The new system should be based on a logical database design and

needs to be implemented on a relational database. (Morris-Murphy, 2003) This requires a

redesign of the current database tables because there is repeated information and the table

information is not normalized.

Cost/Benefit Analysis of Proposed Method

Benefits to the organization:

- Minimizes manual processes to lower the data-entry errors

- Removes the last process required for the UNIX server to allow the Board of Directors to authorize removal of the UNIX server. Cost savings to NTPCUG is $480 per year. This cost will likely increase to $600 next year.

- Provides one interface for the member to join, edit, and renew membership in the organization.

- Provides "instant" issuance of membership numbers and authority to access member-only area of the website.

- Provides Membership reports faster and removes dependency on email delivery of the files.

- Provides BOD ad-hoc queries to determine membership trends quickly.

- Breaks client dependency on a specific database package without affecting all of the layers (user interface, business logic or the database structure).

Risks to the organization include:

- Web server may be less reliable since it will be running Windows Server 2003 instead of UNIX.

- Increased security risk, since many Virus and Worm attacks are aimed at Windows.

- Increased hardware requirements.

- Database may be less protected since it will reside on the Internet.

Costs:

- For software, Visual Perl plug-in for .NET Studio, $40.

- Server previously purchased for SharePoint Portal server project. Software was donated by Microsoft.

Security Considerations for Ecommerce

The four main types of security considerations highlighted by Luke Hohmann include the following:

- Digital identity management

- Transaction security

- Software security

- Information security (Hohmann, 2004)

Each of these categories requires various tools and techniques. According to Ghosh the number one rated concern in using Ecommerce is the potential for loss of assets and privacy due to breaches in security. Security is very, very important to the NTPCUG system. No system is 100% protected from all forms of attack. Each risk that is mitigated

adds cost and decreases performance. The key to a successful system is to balance the

risk, costs, and performance degradations. In the MSS the areas where we need to focus

our security concerns include:

- Member's profile data

- The web forms used to update Member's profile data

- Credit card information in Ecommerce transactions

- Ecommerce vendor's payment response

- Online reports that contain any member's personal information

- Ecommerce transaction queues

- Database tables

- Access to member-only area on the Windows SharePoint Portal server.

The first area of security that we need to address is digital identification. This will be

followed by other areas of concern which include the client-side software, the data

transaction protocols, the Web server software, and the server operating system. All these

areas must be equally addressed to have a flexible and robust application. (Ghosh, 1998;

Hohmann, 2004)

Digital Identification

In Enterprise applications there are different services provided for people in different

roles. For instance, in the MSS, the administrator of the system has more access to data

and reports than an individual member who can only access their own information. Each

person using the system must enter a user-id and password to verify their identity. All

usage of the system will be tracked by this identity. Some new ways of doing the

authentication include a digital signature on a smart card and using biometrics such as a fingerprint scanner.

There are two parts to handling a digital identity, authorization and authentication. Authorization defines who gets access and what rights they get. There are a number of ways this can be implemented. One way this can be implemented is through LDAP, Lightweight Directory Access Protocol. Another would be through Microsoft's Active Directory Services. The second part of identity confirmation is authentication, which is the proof of identity. This is most widely known piece of digital ID management since users have to submit a password as part of authentication. Another way the identity might be confirmed is through recording the client hostname and IP address. To achieve a stronger version of an authentication, a system would operate not only on something you know such as a password, but it could also use something that you physically possess such as a smart card or your computer serial number, or something that identifies you such as a voice print, retina scan, finger print, or the shape of a finger.

Client-side security

As commercial web sites build in more executable content, this increases security vulnerabilities. Generally the HTTP protocol is considered stateless because every subsequent request to a web browser has no memory of a previous request. To solve this problem, state information is passed using "cookies". For example, cookies are often used to retain the authentication information during a web session. Cookies do work to pass the information; however, there are privacy and security concerns when passing information like this. On the positive side, adding intelligence in the client provides a richer online experience by processing data locally with languages such as ActiveX or

Java. To accomplish this, the web server accesses local files and thus it is possible that the web server can read and write your files on your local PC. This also means your personal information might be still resident in a "cookie" for every website to see.

Data Transaction security

The subject of security for Ecommerce payments is quite large and includes the choice of processing a credit card verses using a stored-value account such as **BitPass** discussed above in Online Payment Types. The other area not previously discussed is the protocol used by web-based transactions which is SSL, Secure Sockets Layer. It does not handle security of the payment itself, but rather wraps the messages sent back and forth from client to server in a secure digital envelope. Other more sophisticated encryption techniques may be used depending on which Ecommerce vendor is used. In the MSS the transaction logs, credit card information, and personal identifying information, such as phone numbers and addresses, all have to be protected to the best of our ability.

The most important security for transactions is audit trails. Auditable systems require authentication and a method of behavior tracking. Providing audit trails gives forensic evidence, in case there is suspicious behavior. Additionally, the audit trails may be required by the government and can be used to audit trends over time, such as looking at the number of credit card transactions in each month or the number of people that start to fill in an order form but never complete the transaction with a payment.

Server security

On servers, the areas of security that need to be addressed include the web server, any application software, and the database. First, the web server can be secured with commercially available software to prevent virus and hacking attacks. For basic access to

the computer via FTP protocol, other measures can be made such as limiting privileges to only a few accounts and setting file permissions, so that there are limited areas of exposure on the server. Relational databases also provide another layer of security by user-ids, roles, and limiting rights. For individual documents, you can even go further and restrict the access to a file by an IP address, user-id and password authentication, or by digital certificates which are discussed below. On the server, CGI scripts can also be isolated to an area that is not reachable by FTP or by the execution of the client-side code. By separating the data and the scripts, someone can be prevented from altering the executable code from the web.

Another way privacy can be achieved is to place a hardware firewall between the trusted portion and the untrusted portion of a network. A hardware firewall is almost essential to preventing break-ins to servers due to the number of worms and other attacks that can be initiated from the Internet. Hardware firewalls can also be used internal to corporate networks to keep unauthorized personnel out of databases and application servers. Firewalls are perhaps the easiest way to secure a system.

Security holes can be present in all layers of the architecture including the web server, the database, or in the application software. All aspects of the software design will be reviewed to make sure that adequate protection is provided. All code implemented must also be reviewed for security holes. The 'off the shelf' software must be reviewed for the latest patches, and the system must be tested over and over again for security holes to protect the information on the server.(Ghosh, 1998)

Operating System Security Issues

For operating systems security, keeping the latest patches applied and regularly reviewing the vulnerable configurations and settings is advisable. The weakest link in any operating system is usually in authentication of the users. See the discussion above entitled "Digital Identification". Other problems in operating system security are often 'disasters waiting-to-happen,' such as problems with buffer over-runs. The administrators of the operating system have to stay informed about the latest problems and apply patches as fast as feasible to keep their systems protected. Another way to handle this risk is to hire outside consultants to test your system for problems. Using code analysis and testing, they will identify potential problems and recommend actions for your organization or company to take to plug any security holes.

Support of Ecommerce 'Instant' Notifications

Luke Hohmann suggests that the first step in designing the transaction is to define the unit of work, the transaction, then determine how it fits in the business model.(Hohmann, 2004) For the MSS every transaction needs an audit trail. Therefore, the member making the payment, when we received it, the amount of the transaction, and what type membership dues is being paid will all be logged. Each of the transactions in this system must also be uniquely identified to ensure data integrity. When the member makes a payment from a form, they are taken to the Ecommerce provider's site. The money transaction is performed and then they are returned back to our site with a receipt. Additionally, a notification of a payment is sent to our server (it may be a separate server from where the web-form resides) to a particular URL where a script will process the payment matching it to the pending transaction. The unique identification on the

transaction assures that we match the particular payment made to the proper member and the payment receipt is authentic. Each web form completion generates a 'pending' transaction to NTPCUG's database. When the payment confirmation is received, it is also recorded. If required by the Ecommerce provider, an acknowledgement of the payment is also sent. The Membership Director will later review the pending transactions and then approve them to be posted to the MSS database. (Instant Transaction discussion here)

To support Ecommerce transactions, particular interest will be paid to supporting the four critical transaction properties: atomic, consistency, isolation and durability, also know as the acronym, ACID (the first letter of each property). Atomic means that the transaction either finishes completely or not at all.

The complete states and scenarios for handling the Ecommerce transactions will be thoroughly documented so that detailed tests can be constructed. At each phase of the implementation, testing of all the paths through the application is essential especially during the User Acceptance testing prior to production cutover.

Approach for design and testing of multi-tiered systems

All of the tiers of the architecture will be developed simultaneously by several teams. This presents a problem for coordinating the design and testing of the various tiers. Luke Hohmann suggests when designing a system with substantial amounts of existing data to start the design with the database. This will allow the database design to drive what gets implemented first in the application and GUI layers. One approach that he suggests is to define a small core subsystem of the architecture to implement first and then define all the subsystem interfaces for this small core. He refers to this build process as 'spiking.' A spike should be a user-visible piece of functionality described by a single use case. The

important feature of the spike is that it drives functionality through all the tiers of the architecture.(Hohmann, 2004)

Using spiking to implement the system is an incremental approach. From the first small core, continue to add functionality in phases. Spiking thus becomes a way to manage risk in the implementation of multi-tiered system. As each spike is implemented and tested, the system becomes more complete and stable. We will use periodic builds of the entire system to keep all the interfaces in sync.

In parallel development there will be many times when the different layers are at different levels of development or testing. The MSS will use the GUI layer to drive the other layers. During the implementation of a specific use-case if a missing attribute is discovered the GUI team will work with the middle-tier team to make sure that the functionality is implemented there. Then the middle-tier team will work with the database team to add attributes or stored procedures if they are missing in the data layer.

Since NTPCUG was donated a copy of Microsoft Server 2003, the implementation languages for the middle application tier will likely be on Visual Studio .NET 2003. The way this could be accomplished is a fairly complex topic. Without getting into the subject too deeply, start with the database model. This defines the lowest tier of data. Then define a dataset object which will serialize to XML and handle the communication to the application middle layer. Then, in the middle tier, these application/domain objects are tied back to the data store tier. The actual business rules then are written as methods to the objects. Since you can construct a collection to encapsulate all the other tables, to instantiate the objects at execution time, all that is needed is for the presentation layer to start the one collection on the server. (Hyatt, [Unknown year])

Alternate Tools and Processes

The database design has to be capable of being implemented on any relational database. At this time the assumption is that NTPCUG will use Microsoft SQL Server since we were donated this software. If that is not acceptable there are other solutions at no cost to the organization. For instance, the MySQL database can provide the same functionality and with excellent performance for no cost.

Visual.NET on IIS will likely be one of the development tools. After a prototype is constructed, if the performance is not good, there are several choices that can be made. These may include:

- Getting help to improve our system from Microsoft support and the user group liaison

- Tuning the webserver

- Exploring hardware options of adding another processor or server

- Moving back to various open-source tools is always a fall-back solution.

The n-tiered architecture and design is flexible enough that other tool choices can easily be made.

Key Criteria for Project Success

The following must be implemented in order for the system to be a success:

- All membership functions in the old system must be present in the new system design. These will be verified during user acceptance testing.

- The architecture approach and systems design has to be accepted by the NTPCUG BOD.

- Support for a reasonable security policy implementation must be present.

- Support for a complete audit trail of all transactions and monies must be provided.

The documents to be produced include a Project Plan, a Software Specification and a Risk Analysis, and a Test Plan.

Overall Project History

The project team all contributed to the MSS System Requirement Specification document. The resulting document also includes the initial development plan and was presented on the 29th of August, 2005 at a review. Out of the review the over-all design was accepted, the MSS Specification was updated, several report layouts were added, and a glossary was added.

Managing the implementation project is the President-Elect of NTPCUG. The software development plan outlines all the pieces that can be assigned to other members of the project to develop. The prototyping and development phases have now started. Although several changes were made to the original plan, the initial phased approach is still valid. At first, the Windows SharePoint Portal server team was uncertain how and

when they would be ready to put the server into production. The original project proposal was to prototype an online database using MySQL running on UNIX then to generate a second prototype on an SQL Server running on Windows. The SharePoint Portal server project progressed much faster than the Membership Support System project, leaving the logical choice of implementing on Windows using SQL Server.

The problems encountered on the project were primarily related to limited resources. First, it took considerable time to generate the requirements document. Then getting everyone available to review the specification, architecture, and approach was a real scheduling challenge.

Another variable that changed over the duration of the project was the Ecommerce vendors. When the project was conceived, there were primarily two types of vendors supplying credit card processing online: those that processed payments with no setup fees and traditional processors that charge a setup fee and a per-month fee along with smaller transaction fees. Now there are vendors that specialize in micropayments and PayPal.com offers a wide variety of service levels to go with the levels of processing transactions as choices or extra paid options.(Paypal.com, 2005) Even after all the analysis, the online credit card vendor will not change, since NTPCUG is such a low volume user. However, now we aware that if we approach the monthly transaction level of around twenty, then NTPCUG needs to once again start evaluating vendors. By the time we achieve twenty-five transactions a month, NTPCUG would need to be prepared to move to a new vendor. The design produced is flexible and multi-layered so that many changes can occur over time in response to changes in software, database design, user loads, security, and hardware without impacting the rest of the tiers of the system. The BOD will be able to

obtain valuable business information faster from the server. The members will be able to

keep their information current easier. The Membership Director will regain a large chunk

of time spent in data entry and reporting. Overall, the choice of a distributed, layered

Ecommerce system with a database backend will serve the organization for many years to

come.

Chapter 5 - Lessons Learned

The biggest challenge presented in this project was designing a system for an organization of programmers. Everyone exposed to the project process had a different view on scope and how it should be designed based on their previous experience. Getting everyone involved to agree on a direction was challenging at times, much like herding cats. At work the project manager or the technical team lead generally sets the tone of the project plans. Here the purpose was to achieve a consensus rather than dictate a direction. The lessons learned from this project include:

- The analysis of the Ecommerce transaction processor confirmed my 'gut feel' that Paypal.com was a good choice for a small non-profit as the credit card processor. The cost of the processing is highly dependent on the volume of transactions and the level of customer service expected of the vendor.

- Separating the system into layers of the GUI, the application tier, and the data tier allowed the choice of software to be postponed and eventually coordinated with the SharePoint Portal server project, saving time and effort in not prototyping a second implementation which was a complete surprise to me.

- Numbering the requirements will help to trace the design requirements to the test cases in the test plan. We are constantly referring back to the requirements list.

- Determining the output from each phase of the Spiral model can be a challenge. Since development is taking place on many of the n-tiered layers at the same time, each portion is in a different phase of development at various times. Determining what to use as the output criteria of the phase was the defining factor to determine when a phase was complete.

- Focusing on modularizing the design allows more than one programming team to be operating in the implementation phase.

- In the project plan, the schedule was missed. There was not a high enough factor in the project estimating to allow for the resource risks. With all the people involved, including myself, having limited availability, the schedules were way under-estimated. With family commitments, family disasters, and work time encroaching, the resources had a difficult challenge even getting the project started. Even though this is a normal risk for professional projects, this project being staffed by volunteers made estimation of deadlines next to impossible.

- Initiating a design project for an organization of programmers brings forward many critics. They all have strong opinions on the languages, software to be used, and how they would do the project differently. The support of the BOD was crucial to being able to focus the group on one implementation path.

Project Scope Change/Items Done Differently

Trying to be the development manager and a technical contributor led to a resource bottleneck. Many times others were waiting on me for information. When I had family

health problems, this caused a major bottleneck to the project timeline. I learned that as a leader, I need to delegate more, especially when working on volunteer projects.

First, I've only estimated a few project plans of this size from start to finish. Usually, I review project plans or only contribute as an individual contributor or as one of many team leads. Next time, I would have an experienced project manager review my estimates and project plan to get help in allowing for the various risks.

Second, I would establish a strong structure for the project. The Project Lead for any system to be developed for an organization of programmers has to have strong control of the project. The scope can get out of bounds very, very quickly if not constantly reviewed for fit to the project goals.

Meeting Project Plan Initial Expectations

Initially the plan was to prototype two database implementations on different hardware platforms. The selection of the best one was going to be based on performance and cost. Along the way, I discovered that the cost of maintaining the UNIX platform for the web server in addition to the Windows database server had increased substantially. This led to narrowing the architecture choice to Windows.

The goal of revising the membership process to support the current business practice and make it more efficient is complete. A design specification for the new Membership Support System has been written. The design incorporates an online database utilizing real-time Ecommerce transactions. The secondary goal was also addressed in that the WSS will have control of the members-only portion of the website. Another goal that was accomplished is that the design supports a member's ability to review and update their own personal information in a secure and robust way. The prototyping and

implementation phases have been started. The goal of reducing data entry errors will have to be measured six months after implementation.

Project Next Steps

The next step of this project is for the team to continue with the prototyping, coding, testing and implementation. The organization is proceeding with this project plan. The database design included tables to support the tracking of volunteer time and corporate memberships. There is no plan at this time to actually design the application that will track the volunteer time or support other classes of memberships. Additional design work could be continued on these other sub-systems.

Conclusions

This project verified the initial assumption that a layered n-tiered architecture for Ecommerce was a good choice. This choice gave us the flexibility to change the hardware and software to be used during design and execution without adversely affecting the project quality. By taking time to consider the many ways the membership data was to be used a thorough analysis of the business process was done. This was of significant impact to this project in that it helped us clarify and improve that business process for the organization. This analysis also produced solid system architecture consisting of a GUI, an application layer, and a data layer.

Our analysis further yielded that the choice of credit card processing vendor for Ecommerce was very clear due to the small transaction volume. However, this could have easily changed due to the fact that even during the time the project was being implemented, the Ecommerce solutions available and transaction costs changed. Next

time the organization has to evaluate their online Ecommerce transaction processor, the features checklist and costs analysis spreadsheets will be available to make this process easier. This point stressed how important a business needs analysis is to the system design. Although the Ecommerce vendor analysis did not result in a new vendor, it points out how quickly the vendor offerings can change online these days. This surprising speed of change strengthened the conclusion that the single most important design choice in an Ecommerce system is to design using the n-tiered architectural model.

Appendix A - References and Works Cited

BitPass.com. (Unknown). BitPass Professional Get Started Guide.
http://corp.bitpass.com/docs/professional-get_started.pdf

Boehm, B. W. (1988). *Software Engineering Project Management: a Spiral Model of Software Development and Enhancement* (1st Edition ed.): IEEE Computer Society Press.

Boehm, B. W. (2000). Spiral Development: Experience, Principles, and Refinements. 2005, http://www.sei.cmu.edu/cbs/spiral2000/february2000/SR08html/SR08.html

Campbell, A. (Unknown). Bitpass Micropayments User Survey http://www.smallbusinesses.blogspot.com/2004/01/bitpass-micropayments-user-survey.html

Daley, W. M. (2000). *The Business of E-Commerce*. New York, NY: Cambridge University Press.

Dowling, J. (2002). The Art And Science Of IT Architecture Design. 2005, online at http://www.umsl.edu/~sauter/analysis/dowling/

Fairley, R. E. (1988). *Software Engineering Project Management: A Guide for Preparing Software Project Management Plans*: IEEE Computer Society Press.

Fingar, P., Kumar, Harsha, & Sharma, Tarun. (2000). *Enterprise E-Commerce*: Meghan-Kiffer Press.

Free-merchant-accounts.com. (Unknown). Free-merchant-accounts.com Information page. 2005, http://www.free-merchant-accounts.com/index_alt.html

Ghosh, A. K. (1998). *E-Commerce Security*: John Wiley & Sons, Inc.

Hohmann, L. (2004). *Beyond Software Architecture: Creating and Sustaining Winning Solutions*. Boston: Addison-Wesley.

Hyatt, K. ([Unknown year], 2005). N-Tier Application Development with Microsoft.NET. http://www.microsoft.com/belux/nl/msdn/community/columns/hyatt/ntier1.mspx and http://www.microsoft.com/belux/nl/msdn/community/columns/hyatt/ntier2.mspx

Kalakota, R. R., Marcia. (2001). *E-Business 2.0: Roadmap for Success*: Addison-Wesley.

Langer, A. M. (2002). *Applied Ecommerce: Analysis and Engineering for Ecommerce Systems*: Wiley Computer Publishing.

Maguire, J. (2004A). Big Changes in Small Payments. 2005, http://www.ecommerce-guide.com/solutions/secure_pay/article.php/11836_3330031_1

Maguire, J. (2004B). Making Small Payments Practical for Small E-Tailers 2005, http://www.smallbusinesscomputing.com/emarketing/article.php/3332801

May, P. (2000). *The Business of E-Commerce: From Corporate Strategy to Technology*. New York: Cambridge University Press.

McPhee, L., Drucker, Peri, Cormia, Robert & Hammer, Cathy (2000). E-Commerce Product-Comparison Matrix from Tips & Tactics for Conducting E-Commerce. 2005, http://www.inc.com/tools/2000/09/20007.html

Morris-Murphy, L. L. (2003). *Oracle9i: SQL with an Introduction to PL/SQL*: Thomson: Course Technology.

Morrison, M. M., Joline (2003). *Database-Driven Web Sites* (2nd ed.). Boston: Thomson.

Paypal.com. (2005). Compare Key Features of Each Payment Solution. Accessed 2005, https://www.paypal.com/us/cgi-bin/webscr?cmd=_feature-and-pricing-comparison

Yahoo!_Merchant. (Unknown). Yahoo! Merchant Solutions. 2005, http://smallbusiness.yahoo.com/merchant/ and http://smallbusiness.yahoo.com/merchant/faq.php#p1

Appendix B – System Design Notes

Existing Hardware

Hardware specifications for the existing database are:

- Pentium processor of 800 MHz

- CD R/W for backup

- 128 megabytes RAM

- 180 megabytes for the Microsoft Access 97 database and 1megabyte for reports
  and emails

- Software: Microsoft Word and Publisher with 400 MB for the software space.

- Monitor resolution of 640 x 480; 256 colors minimum

Hardware specifications for the existing Unix Web server at the ISP are:

- Pentium processor of 800 MHz

- 512 megabytes RAM

- CD ROM for backup

- 100 megabytes Disk for logs and transaction logging

- Dumb terminal or access to Unix server via *ssh* utility

- 10/100 Ethernet NIC

- DSL line with static IP address (domain name already present)

- Perl 5 and Apache web server software installed
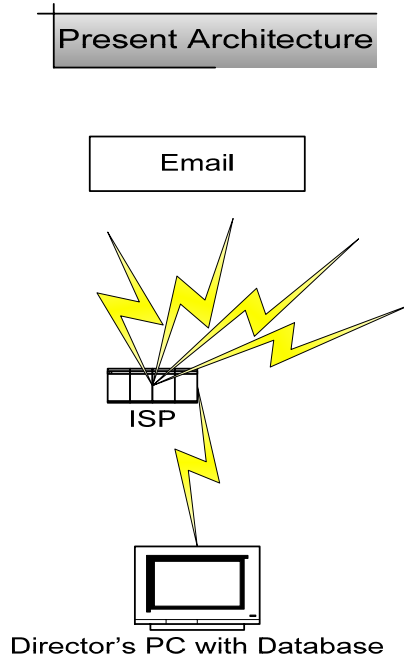
Figure B-1 Existing Architecture

Future System Requirements

The target hardware will be at least:

- 800 MHz Pentium Class Processor

- 512 MB Memory

- 575 MB Disk Space

- Server will be connected to the internet on a high-speed phone line, DSL,

  or LAN

The systems software to be used:

- OS – Windows Server 2003

- Application Software – Microsoft SQL Server and scripting language: PHP or

  ASP.NET

- Ecommerce application code must run on the same system or a second server of

  similar configuration

53

Figure B-2-Future System N-tiered Architecture (Hyatt, [Unknown year])
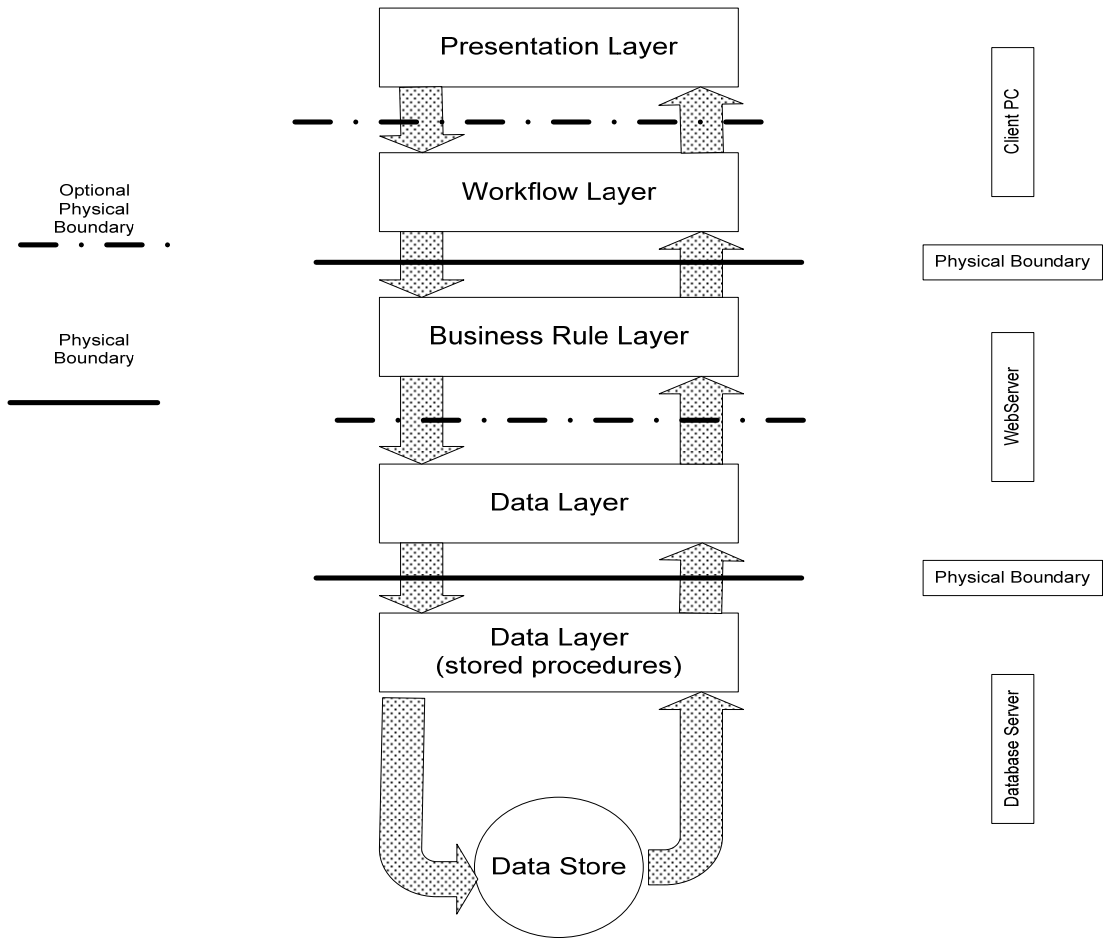
Deliverable	Date
Gather Requirements for overall system	10/07/04
Data flow of Membership Process	11/08/04
Research of Ecommerce vendors	01/14/05
Risk Analysis and Risk Management Plan	02/28/05
High-level Architecture Design	08/31/05
Software Design Document	10/31/05
New Membership Database Schema	05/05/05
Scripts to Convert Existing Database to New Format	06/15/05
Testing & SCM Plan	10/31/05
Prototype of Web Interfaces	10/31/05
System Architecture Finalized	08/31/05
Configuration of Server and Tools to be used	08/31/05

Software Methodology – Spiral SDLC
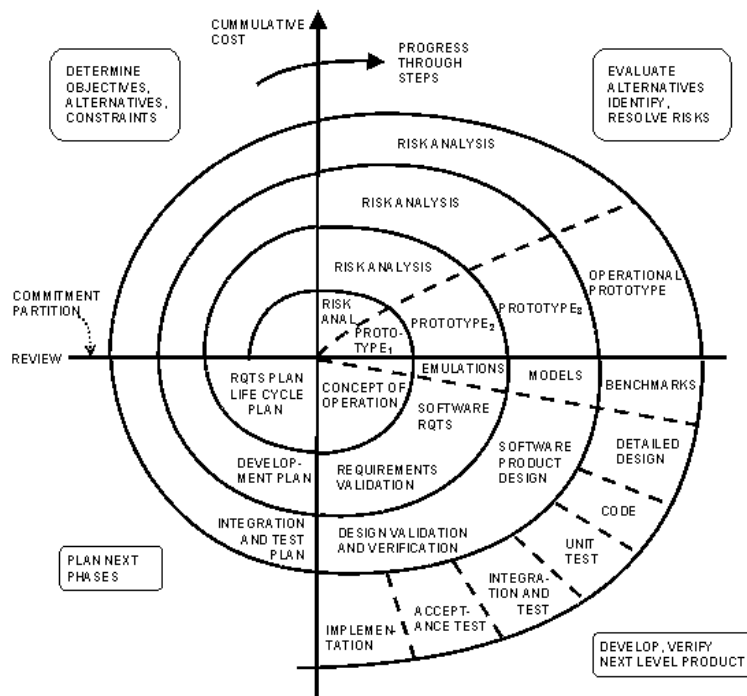
Diagram from Boehm's online paper on how the Spiral model is currently being

used.(Boehm, 2000)

Appendix C – Ecommerce Vendor Requirements

The chart below was constructed to show the various fees charged by each of the four

vendors being evaluated.

Table C-1 Online Transaction Fees

Merchant	Setup Fee	Gateway Fees	Monthly Fees	Transaction Fee Regular Dues $55	Transaction Fee Family Dues $85
BitPass	0	0	0	3.25	4.75
Free-merchant-accounts.com		19.95	10	0.25	0.25
PayPal.com	0	0	0	1.9	2.77
Yahoo! Merchant	50	0	29.96	0.83	1.28

Table C-2 Transaction Processors Yearly Cost

Merchant	Yearly Cost - 5 per month	Yearly Cost - 10 per month	Yearly Cost - 20 per month	Yearly Cost - 30 per month	Yearly Cost - 50 per month
BitPass	195	390	780	1170	1950
Free-merchant-accounts.com	374.4	389.4	419.4	449.4	509.4
PayPal.com	114	228	456	684	1140
Yahoo! Merchant	409.32	459.12	558.72	658.32	857.52

This table shows the cost of $55 membership transactions at various volume levels with

the four vendors begin compared. No set up fees were considered. However the monthly

and gateway charges were included in the calculations.

Table C-3 Providers Features

Feature	Need	Nice to Have	BitPass.com	Free-merchant-accounts.com	PayPal.com	Yahoo! Merchant
Shopping Cart And Payment Processing						
Shopping cart	√		√	√	√	√
Softgoods-capable (e.g., sale of files)		√	√	√	√	Note 4
International currency capable			?	√	√	√
Calculate sales tax		√	?		√	√
Supports all major credit cards	√		√	√	√	√
Calculate shipping charges		√	?	?	√	√
Supports SSL	√		√	√	√	√
Customer Service Administration						
Live chat-room or phone help		√	?	√	√	√
E-mail order-verification to customer & merchant	√		√	?	√	√
Detailed monthly statement	√		?	√	√	√
'Instant' order-verification to merchant website	√		√	?	√	Note 4
Merchant help w/store		√	?	√	√	√
Credit Card information private to member and vendor (NTPCUG doesn't handle card numbers)	√		√	X	√	√
Site-Activity Administration						
Online payment administration	√		?	√	√	√
Online transfer to merchant bank account	√		ACH or PayPal	?	√	√
Allows web server-side coding		√	Note 1	?	√	Note 4
Costs of payment processing			Note 1	Note 2	Note 3	Note 4
No one-time setup fee	√		√	√	√	X
No Monthly Fees	√		√	X	√	X
Windows & Unix support	√		√	?	√	Note 4

Table design modified from *Tips & Tactics for Conducting E-commerce* (Maguire, 2004B; McPhee, 2000)

General Note:

Peppercoin and Yaga, both mentioned by James Maguire in "Big Changes in Small

Payments", specialize in micropayments similar to BitPass. They had no information

openly posted on cost at the websites so were not included in the comparison.(Maguire,

2004A)

Note 1: BitPass costs and notes:
    Fees:
    - For items priced $0.01-$5.00, the transaction fee is 15%.
    - For items priced $5.00 and higher, the transaction fee is 5% + $0.50.
        Fees for BitPass Studios
    - For items priced $0.01-$5.00, the transaction fee is 30%.
    - For items priced $5.00 and higher, the transaction fee is 10% + $1.00.
    For our $40 dues, BitPass would be $6.(BitPass.com, Unknown)

    Website handles the verification and hand-off to a website without having to do the code required for
    PayPal. Customers can pay with all credit cards AND PayPal. However, NTPCUG currently has a
    PayPal account and the expertise to do the code which is not that complicated. (Campbell, Unknown)

Note 2: Free-merchant-accounts.com costs and notes:
    Transaction Fee .25 cent per transaction; Monthly Statement Fee $10 per month; Gateway Fee $19.95
    per month (Free-merchant-accounts.com, Unknown) Details on implementation is not posted to the
    website and requires signing up for the service. (Free-merchant-accounts.com, Unknown)

Note 3: PayPal costs and notes:
    For items priced $0.01-$3000, 2.9% + $0.30 USD
    For $40 dues the costs are: $1.46

    PayPal can direct the member to a URL after the transaction. The URL can be accessed directly
    without paying, but the required access control is a small amount of code.(Paypal.com, 2005)

Note 4: Yahoo! Merchant costs and notes:
    They host the website. You are unable to use your own server at the 'Starter' level of cost. Perl and
    PHP are available at this level though. At the 'Standard' or 'Professional' level you may have real-time
    transactions and your own server.(Yahoo!_Merchant, Unknown)

    'Starter' level - Setup fee: $50
    Transaction fee: 1.5% per transaction
     Monthly Hosting Fee $29.96 per month

    'Standard' level - Setup fee: $50
    Monthly Hosting Fee $74.96 per month

Appendix D – Glossary

Term	Definition
Apache Web Server	a Web server program that was built by a group of open-source programmers
BOD	Acronym for Board of Directors
DSL	Acronym for Digital Subscriber Line
Ecommerce	e-commerce is any kind of transaction done partly or completely over a network
Entity relationship diagrams	Entity Relationship Diagrams are used to pictorially represent database entities, the attributes used to describe the entity, and the relationships between various entities
ERD	Acronym for Entity Relationship Diagrams
GUI	Acronym for Graphical User Interface
HTTP	Acronym for Hyper Text Transfer Protocol
Hyper Text Transfer Protocol	The communications protocol that is used for Web browsing
IIS Server	*See Windows IIS Server*
Instantiate	To create an instance of an object of a particular type
MD	Acronym for Membership Director
Member-only area on Web	Only NTPCUG members in good standing can access any of the NTPCUG SharePoint sites.
Membership Support System	The Membership Support System includes all of the functions of the Membership Director plus some additional enhancements.
MSS	*See Membership Support System*
NIC	Acronym for Network Interface Card
NTPCUG	Acronym for North Texas PC Users Group
PDA	Acronym for Personal Digital Assistant
Perl	Acronym for Practical Extraction and Reporting Language. PERL is an open source server side programming language extensively used for web scripts and to process data
RAM	Acronym for Random Access Memory
SDLC	Acronym for Software Development Life Cycle
SecureSockets Layer	This layer is a protocol for securing a message transmission over the Internet
SharePoint Services	*See Windows SharePoint Services*
SIG	Acronym for Special Interest Groups
Snail Mail	US Postal or other physical mail
SSL	Acronym for Secure Sockets Layer
Windows IIS Server	Internet Information Server, a Microsoft Web server that runs on Windows
Windows SharePoint Portal Server 2003	Enables enterprises to develop a portal that seamlessly connects users, teams, and knowledge

Windows SharePoint Services	A free software component, which can be downloaded and installed on Windows 2003 Server to implement and manage a team collaboration infrastructure.
WSS	Acronym for Windows SharePoint Services

Appendix E – MSS Software Requirements Specification

< This page intentionally left blank >

North Texas

PC Users Group

Membership Support System

Systems Requirements Specification

December 12, 2005

Version 2.0.5

By member, Jeri J. Steele,

Student at Regis University, Denver, Colorado

Preface

The purpose of this document is to provide the requirements for the North Texas PC Users Group Membership Support System. This document contains goals and objectives, Project scope, and requirements. It has been written for the stakeholders and system developers to utilize in further analysis and development of the system. Revisions will be made as changes to the requirements occur.

Note: This is a work in progress. Parts that are not complete will be finished later in the Software Design Life Cycle (SDLC).

Summary of Changes to Document

Figure 1   Version Change Table

Date	Version No	What Changes were done	Page No	Initials
12/16/04	1.0	Initial Document created	ALL	JJS
1/02/05	2.0	Updated to latest table design	ALL	JJS
4/30/05	2.0.1	Updated for E-Commerce transaction design	ALL	JJS
08/13/05	2.0.2	Updates for Stored Commands layer	ALL	JJS
09/18/05	2.0.3	Updates for Business Rules	ALL	JJS
09/27/05	2.0.4	Updates after System Review	ALL	JJS
12/05/05	2.0.5	Formatting changes to publish with Thesis	ALL	JJS

Table of Contents

Introduction

The North Texas PC Users Group (NTPCUG) is a non-profit organization with roughly 400 members dedicated to education on PCs and PC Applications. The current NTPCUG membership director maintains the membership database and produces reports for the Board of Directors (BOD). Even though input transactions are initiated on the web, information is entered in a manual process using an Access database. This process is error prone. Additionally, the Board of Directors needs timely membership information. NTPCUG also has been working to enhance membership benefits by developing a Share Point Portal for collaboration of the members to upgrade their technical skill-sets and exchanging technical information. The implementation of the SharePoint Portal is an undergoing project for 2005 however currently the list of authorized members must be generated from an Excel spreadsheet. The new Members Support System (MSS) will be a means for tracking the member's dues transactions, producing online membership reports, and to authenticate members using the SharePoint Portal. MSS is an important part of the strategy to enhance membership benefits and to continue to grow the membership.

These requirements will mention several parts of the system that could be implemented in several ways. First is a relational database which will likely be implemented with Microsoft SQL Server database. However, it could just as easily be implemented with MySQL or any other relational database. Also, access to various parts of the system needs to be controlled through authentication and authorization security. This security control will likely be implemented via Microsoft's Active Directory Group Policies.

1.0 Project Goals and Objective

The project objective is to produce the Membership Support System and reduce the errors in membership transactions by 20%. Additionally, to make the issuing of membership numbers automatic by the system after verification of the transaction, and to make the monthly membership reports available weekly right after processing new memberships. All reports must be available from the Internet with controlled access. All of this must be accomplished with no increase in membership processing time needed by the Membership Director. Another main objective is to control membership only areas online with the latest MSS data.

1.1   Summary of Objectives

Figure 2- Objectives and Business Champion Table

Project Objective	Purpose (Business need being fulfilled)	Business Champion	Priority
Reduce the errors in data entry	Easily available online membership information on a member's profile page, where each member can update their MSS data.	Jeri Steele, Linda Moore	High
Reduce time to issue membership numbers	Issue membership numbers for people who have paid with a properly cleared online transaction for people wanting to join the credit union or access the member-only area online	Linda Moore	High
Make reports available online each time the database is updated	Current information for BOD in a timely manner	Birl Smith, Linda Moore	Medium
Control member-only area of NTPCUG MSS SharePoint sub-site by using Security Policies	Increase interest in Member Only area of Website, but allow only valid people access to NTPCUG only files	Art Yasui	High
Reduce processing time of credit card transactions	Decreases time to clear credit card transactions	Jeri Steele	High
Maintain low cost of credit card processing	Minimal or no increase to processing costs	Linda Moore	High
Clean up bad data in database	Remove old members to archive area & clean up free form fields to standardized formats	Jeri Steele	Medium
Automate More of MSS Processing Flow	Standardize output formats	Jeri Steele	High
Allow ordering of discounted training or other items by valid members	Future Need- would include an inventory list, prices and validation against current membership list; also purchases maybe limited to a certain number of inventory items and only certain types of members such as SIG leaders may order..	Linda Moore	Low

68

1.2   Summary of NTPCUG's Technology architecture plan

The business applications maybe distributed across several physical computer systems for a secure system that will allow for an incremental growth in capacity. The BOD also intends in the future to offer some Member benefits by not only being able to pay dues online, but also to support direct ordering on the Web site, and any other additional membership related functions.

2.3 Summary of Current Membership System Context

- Access Database with use of Microsoft Publisher, Word & Excel on the comma delimited files exported from the database.
- Automated online membership form in Perl & confirmation via Email.
- Manual process for generating the membership cards, renewal letters, and email files.

2.4 Statement of Scope and Constraints

The MSS will provide a membership tracking system which allows NTPCUG to provide many membership benefits at a low dues cost. The scope of this project includes:

- An automated credit card processing that will give a membership number and access to members-only sections of the web 'instantly'.
- A new relational database for membership information will be created from cleansed current data and moved to a NTPCUG online server.
- Members must be able to join, update or renew via a manual process or on the http://www.ntpcug.org website.
- After posting all membership transactions, a standard 'set' of extracts or reports will be uploaded to the NTPCUG MSS SharePoint sub-site.
- Security access to online reports will be restricted to the BOD or authorized MSS administrators.
- An updated membership list will update the files that implement the security access policies to the NTPCUG Enterprise Applications.
- Online updates and new member's information will be placed in a pending transaction queue for review before they are applied to the relational database tables to allow for correction of data-entry errors.
- Database tables and/or other mechanisms will be provided to isolate code changes when changes occur in dues amounts, membership expiration policies, membership types and services, and other policy rules of the NTPCUG.
- All changes to online credit card processing must be at no or minimal change in card processing fees for a volume processing.
- Create report extract when snail mail and email notices are sent to members, whose membership is about to expire or has expired. Extract needs to include date sent and other member details.
- Implementation must be done with volunteers and existing hardware.

The primary business benefit for NTPCUG includes:

- Reduced errors caused by manual processing

69

- Rapid processing of online membership transactions
- Maintaining or reducing the time to process monthly transactions and reports
- Increased use of online updating of member information
- Automatic access to member-only area for new members
- Increased membership loyalty (and therefore renewals) through increased member benefits
- Increase members knowledgebase of leading edge software, which keeps their skill set up-to-date.
- NTPCUG members working on this development project can also add their NTPCUG project experience to their resume.

2.5 System Context

The Membership Support System interacts with other departments/systems in the following ways:

- Provides Current Membership Lists to BOD, Treasurer, and Membership Director and to the NTPCUG SharePoint Portal.
- Provides Membership Reports on renewals, expired, and new members for the month to the BOD.
- Creates a membership transaction to the credit card processor.
- Interacts with the Member, Membership Director, and BOD by:
  Providing online transaction details in the pending queue, which is saved into the MSS transaction file for future reference.
  Retrieving details from the database for the update transactions
  Sending a new Member a welcoming message and details of the transaction

Figure 3 - System Context Diagram

71
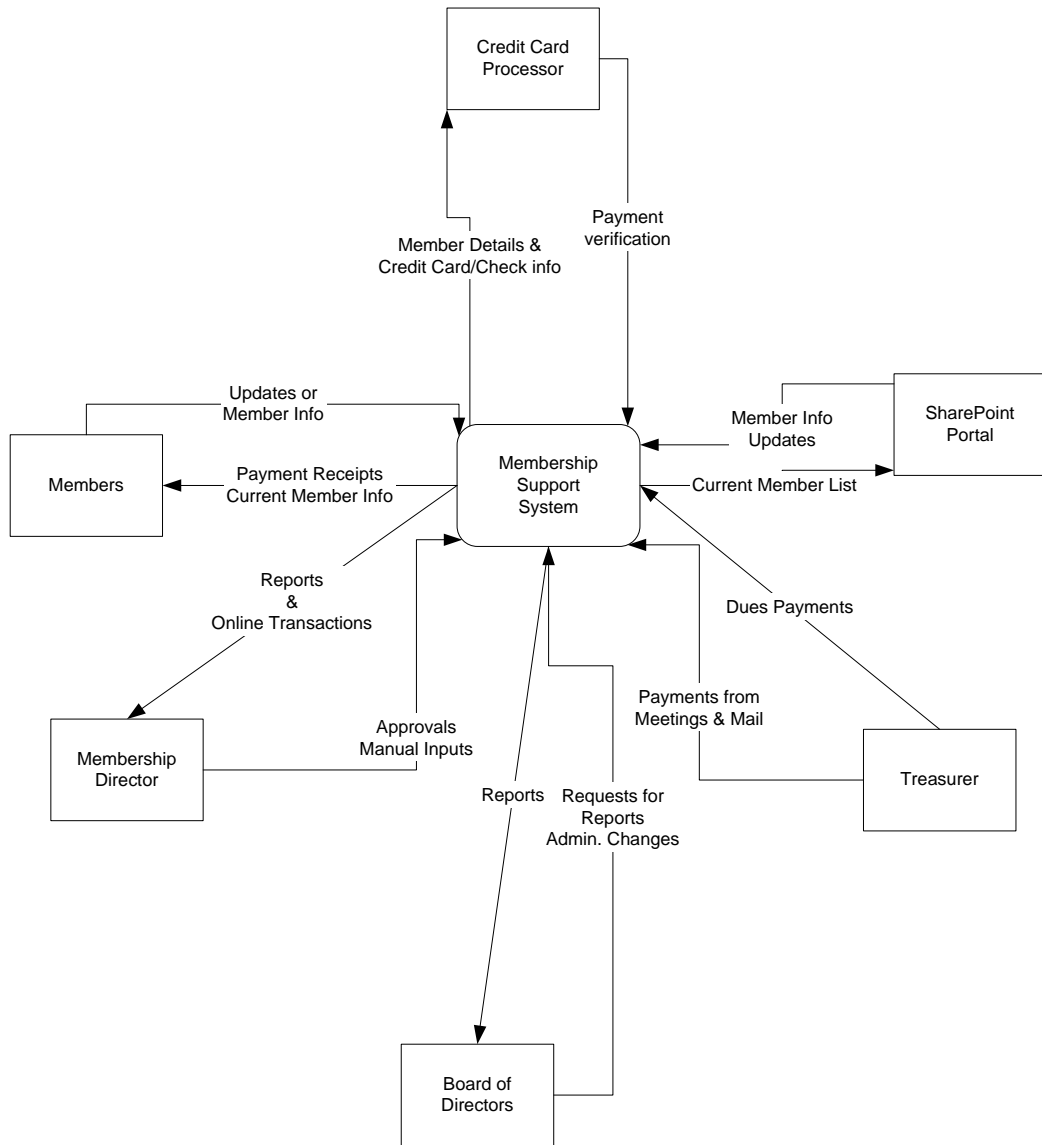
3 Business Process/Usage Scenario

3.1 Stakeholders

The following stakeholders have been identified to be champions of any business process changes needed. Each stakeholder will be interviewed. Any special requirements and concerns will be added to the requirements document as needed. [1]

Figure 4 - Stakeholders

Stakeholders
Birl Smith, President
Linda Moore, President-Elect
Doug Gorrie, Manager of Email & Unix Sys Admin
Marty Mills, Treasurer
Jeri Steele, Membership Director
Art Yasui, System Administrator, Webmaster, SharePoint Team SQL Server developer, W2003 Active Directory
Mark Phelps, System Administrator
Steve Scudder, Webmaster
David Williams, advisor on e-commerce systems
Jack Atkinson, Membership Retention
Robert McNeil, Info Desk Manager

3.2 User profiles

The following types of users have been identified initially. Additional requirements will be added by type of user as further analysis is completed. These requirements will include any special requirements for performance or UI usability issues.

---

[1] John W. Satziner, Robert B. Jackson, Stephen D. Burd, *Systems Analysis & Design* (Boston, MA: Course Technology, 2002), 116-117.

Figure 5   - User Profiles

User	Activity
Member (applies to reg., student or family membership categories)	Update member's personal information; Join as new member or renew membership
Treasurer	Receives payment; verifies membership as needed to receive payments
Membership Director	Post membership updates; Post new members; Post membership Renewals; Generate New Member Cards; Generate updated Email List; Generate New Members for the month List; Generate Renewal notices; Generate membership lists; Generate Expired Member list, Generate Membership Retention Status Report
Board of Directors	New Member Reports, Membership Lists, Retention reports; Volunteer lists, Expired List, online search for member by ID or last name, first name
Email, Web based verification scripts, SharePoint System	Check for valid member in Membership Lists
Visitor	Visitor to website; unregistered
Visitor Registered	A registered visitor

Note: Due to other system requirements other roles maybe defined by the Security team.

3.3 Membership Standing Rules

For the North TX PC Users group there are several standing rules relating to memberships. This section documents the current rules and proposes a new set of standing rules to be presented to the Board of Directors.

3.4 Proposed Membership Standing Rules

3.4.1   **NEW MEMBERS:** New members get a membership ID number assigned, first come, first served. The membership renewal date is set to the end of the current month if the person joins on or before the Third Saturday meeting. If they join after the Third Saturday meeting then their membership renewal date is set to the end of the next month. In the case of new member via snail mail, join is defined as the postmark date.

3.4.2   **FAMILY MEMBERS:** Family members can be up to three people at the same address. There will be only one snail mail renewal notice per family membership and only one envelope sent with the dues cards. Family Members will all get the same expiration date. Fields that can be different are names, email, telephone numbers, occupations, and company. The other family member's membership ID is recorded in the comment field to help with renewals.

3.4.3   **RENEWAL NOTICES:** Members receive email notices two months and one month before their membership expire. If the person has no email address then these notices will be mailed. The month the membership expires the member will receive both an email notice AND a snail mail notice of expiration. Expired members will also receive snail mail notices that their membership has expired for two months after the expiration. No further notice will be sent.

3.4.4 **POSTING RENEWALS:** Renewals are posted at least once per month to the database. No matter when the transaction is posted to the database, the date of the transaction is used. With snail mail the postmark date is used. For PayPal transactions the date of the email sent will be used as the transaction date. If the person's expiration date is less than six months in the past, then the new expiration date is posted as one year greater than the member's current expiration date. If the person has expired over six months ago, then post the membership as if the person joined new on that date including issuing a new member number. Members are allowed to renew and pre-pay several years in advance, if they so choose.

Figure 6 Example Renewal Dates

Current Expiration Date	Date Posted	New Expiration Date	Explanation
10/31/2005	10/01/2005	10/31/2006	on-time renewal
09/30/2005	10/01/2005	09/30/2006	renewal within 6 months
11/30/2006	10/01/2005	11/30/2007	renewal a yr in advance
11/30/2004	10/01/2005	10/31/2006	renewal over 6 months expired – process as new
11/30/2004	10/22/2005	11/30/2006	After 3rd Sat & over 6 months expired

3.4.5 **CONVERTING REGULAR MEMBERSHIP TO FAMILY MEMBERSHIP:** To convert an existing regular member to family membership, look up the expiration date. If the joining member has paid only the difference between regular membership and family membership then the regular member has to have renewed within the past two months, otherwise the member must pay a full year's dues at the family membership rate. If the membership conversion happens at renewal time or within the past two months of expiration, then set the expiration date for all members to one year from the expiration date. If the membership expiration date is less than or equal to the current month plus two, then set the expiration date of ALL family members according to the rules defined in 3.4.4. The current rule for the membership expiration date over two months in the future is to then add all family members at one year over the expiration date, thus giving the new family members extra months. The new proposed rule is to only do a family membership conversion within two months of the renewal date.

Figure 7 - Example Conversion of Renewal Dates

Current Expiration Date	Amount Paid	Date Posted	New Expiration Date	Explanation
10/31/2005	Full renewal	10/01/2005	10/31/2006	on-time renewal
09/30/2005	Full renewal	10/01/2005	09/30/2006	renewal within 6 months
11/30/2006	Full renewal	10/01/2005	11/30/2007	renewal a yr in advance
11/30/2004	Full renewal	10/01/2005	10/31/2006	renewal over 6 months expired
11/30/2004	Full renewal	10/22/2005	11/30/2006	After 3rd Sat & over 6 months expired
04/30/3006	Full renewal	10/01/2005	10/31/2007	Conversion over 6 months in future; Current Rule
04/30/3006	Do not convert	10/01/2005	_____	Conversion over 6 months in future; Proposed Rule

3.4.6 **CONVERTING STUDENT MEMBERSHIP TO REGULAR MEMBERSHIP:** To convert an existing student member to regular membership, look up the expiration date. Set the expiration date according to the rules defined in 3.4.4 Conversion of student membership to regular membership can only occur at renewal time.

3.4.7 **MEMBERSHIP REFUNDS:** IT IS THE POLICY OF NTPCUG TO NOT REFUND MEMBERSHIP DUES.

3.5 External System Events

The Membership Support System has external, temporal, and state events that trigger actions within the system.

Figure 8 - External System Events

Event	Type
Create new member	External event initiated by Membership Director
End of posting Membership Forms	External event initiated at the completion of posting a batch of membership transactions to generate new extracts
End of Month	Temporal event to initiate End of Month Reports
Post Membership Payment	External event initiated upon receipt of payment
Generate Renewal Notices	External event initiated by Membership Director

Any other state events will be identified later in analysis.

3.6 Use-cases

3.6.1 Create a New Membership

Main Flow: A person fills in a membership form online or gives a paper form is given to the Information Desk along with the dues payment to the InfoDesk Manager. The InfoDesk Manager writes a receipt in his receipt book. He gives one copy to member and retains the carbon copy in his receipt book. At the end of the 3rd Saturday meeting, the InfoDesk Manager or the Attendance Check-In Manager will update the new member-renewal pending queue using internet access at the 3rd Saturday meeting. The InfoDesk Manager will give the application forms and membership dues to the Treasurer at either the 3rd Saturday meeting or at the BOD meeting on the following Tuesday.

The Treasurer receives the money. The Treasurer gives (or mails) the application form(s) to the Membership Director, so that she can review and approve the pending queue transactions. The Treasurer deposits the monies received into the NTPCUG bank account with a separate entry for each member.

If the member's application information has not been posted to the pending queue, then the Membership Director will update the pending queue. The Director looks up

the member. Then, the Director will post the payment to the database and update the renewal date according to the Standing Rules. The scenario ends.

Exception Conditions: If Membership Director can not read information on the form, information needed must be obtained via phone.

3.6.2 New or Renewal Membership at Meeting

Main Flow: Member gives a membership form to the membership desk along with the payment. The Treasurer receives the money. The Information Desk, Treasurer or the Membership Director posts the payment. The Director looks up the member. The system queries the database to check if membership currently exists. The system returns membership ID and renewal date if it exists. Then payment is posted to the database and the renewal date updated according to the **Error! Reference source not found.** Membership Standing Rules. A new member will execute the 3.6.1 Create a New Membership scenario. The scenario ends.

Exception Conditions: If a renewing member is not found in the database, then member is looked up by last name, first name, and phone number. If the member is still not found, Membership Director must phone the member and correct any information needed or post the transaction as a new member.

3.6.3 New or Renewal Membership online

Main Flow: A person goes to the membership web page. A current member will retrieve their current information via their information via their Member's Profile webpage and make and make any updates. A payment will be made via the web page. The Treasurer receives information as to the status of the payment. The Membership Director after review will post the payment to the database and the renewal date is updated according to the Standing Rules. Finally, execute the Create a New Membership scenario. This scenario ends.

Exception Conditions: If the errors are found in the information such as an invalid email, Membership Director must phone the member and correct any information needed.

3.6.4 Update Membership Information

Main Flow: Member or authorized Board member goes to 'Update Membership' page and enters their password. The MSS system looks up the existing Member's Profile page and that specific member's information is displayed. The member updates their Member's Profile Page as required and confirms that all of the membership data is accurate. The system gives a confirmation that the updates have been made and they are placed in a pending queue for review by the Membership Director. The scenario ends.

Exception conditions: Member loses their password.

3.6.5 Look up a Member

Main Flow: Member or authorized Board member goes to 'Look up Membership' page and enters their password. The system looks up the existing member's information by entering the member's membership number, or their last name and first name, then the member's information will be displayed. The scenario ends.

Exception conditions: Member loses their password. (This scenario TBD)

3.6.6 Member Forgets Password

Main Flow: Member or authorized Board member forgets their password to access membership information. The system can look up a membership number if you have the first name, the last name, 5-digit zip code and their email. A URL to reset the password will be sent to their email. If their email has changed, then the page will also ask for their phone number. The scenario ends.

> **Exception conditions:** Member can't match enough information to reset their password. If this happens the system will send a message to the two designated system administrators, so that one of the system administrators can process this request to reset the member's password.

3.6.7 Add a Family Member to a Family Membership

Main Flow: A person has filled in a membership form online or the Membership Director received a form from the Information Desk along with the payment. The membership type selected as 'Family' type membership. The membership for the first member is entered the same as any other new membership, then a second form is brought up on the page with the same address and telephone numbers and renewal dates. The additional member's name is filled-in along with additional occupation, employer, email addresses, and work phone numbers; then the membership is added. Up to three members information can be added on the Family Membership. The scenario ends.

Exception Conditions: If Membership Director can not read information on the form, information needed must be obtained via phone.

3.6.8 End of Month

Main Flow: The Membership Director reviews all pending transactions for accuracy and makes any changes needed. All transactions are then applied. The month-end scenario is applied is the following:

- All members that expire at the end of the month are moved to inactive status and the member's ID will be removed from the security access list.
- A new membership list is generated and a member's report is made available online on the MSS SharePoint sub-site.
- A new email list is generated and posted to the web.
- An updated a renewal list is generated and the extracted posted to the MSS SharePoint sub-site.

- An updated membership retention report is generated for the last 2 years and posted MSS SharePoint sub-site.

The scenario ends.

Exception Conditions: None

3.6.9 Convert a Regular Member to a Family Membership

Main Flow: A regular membership can be converted to a family membership within 60 days of the date that they joined or renewed. After entering the new renewal date for the primary member, then a second form is brought up with the same address and home telephone number as the primary member. The additional member's fields such as name, email, occupation, and company are filled-in and the membership added. Up to three members information can be added on one Family Membership. The scenario ends.

Exception Conditions: If Membership Director can not read information on the form, information needed must be obtained via phone.

3.6.10 Payment Transaction Received

Main Flow: System receives a transaction from the processor. The status can be one of the following:

   Completed
   Denied
   Expired
   Failed
   In-Progress
   Pending
   Refunded
   Reversed
   Voided
   Processed

(See explanation in PayPal document for status explanations)
After getting a transaction a notification is sent in email to the administrator and matches the transaction notification with the pending membership form. The scenario ends.

Exception Conditions: The handling of all the different conditions needs to be automated eventually. At first, just send an email to the Membership Director/Administrator who will then bring up the transaction queue for review.

3.6.11 Payment Chargeback from Credit Card Processor

Main Flow: A person requests a chargeback on a credit card transaction. The Treasurer works with the credit card processor to verify that the chargeback is a valid request. If the member has not received a check as a refund and the chargeback is valid, then remove the membership renewal or delete the membership from the database.

Exception Conditions: If Membership Director can not read information on the form, information needed must be obtained via phone.

4 Information Flow

Figure 9 - Membership Join Information Flow

Member Join / Renew
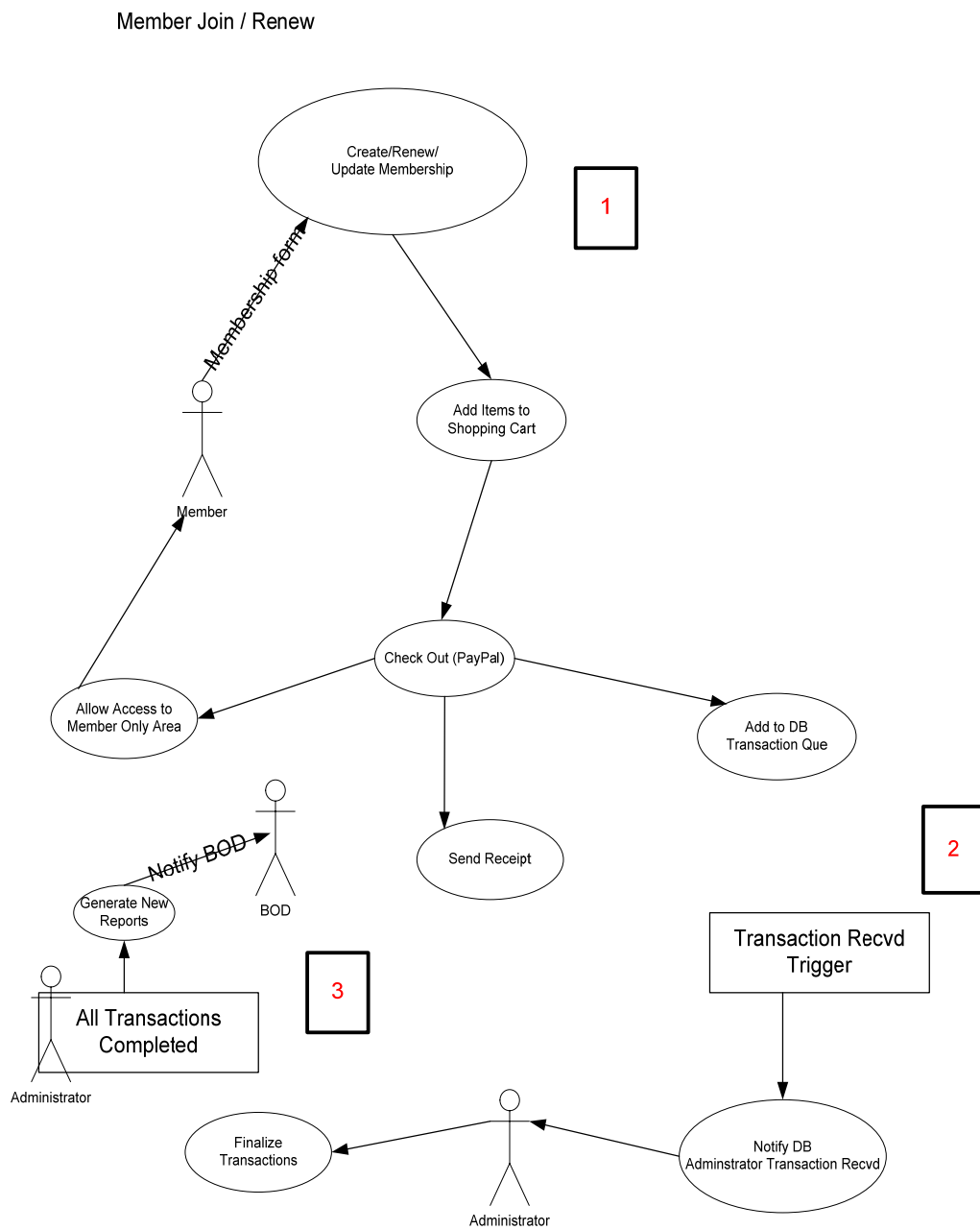
4.1 Membership Join Sequence Diagram

Figure 10 - Membership Join Sequence Diagram
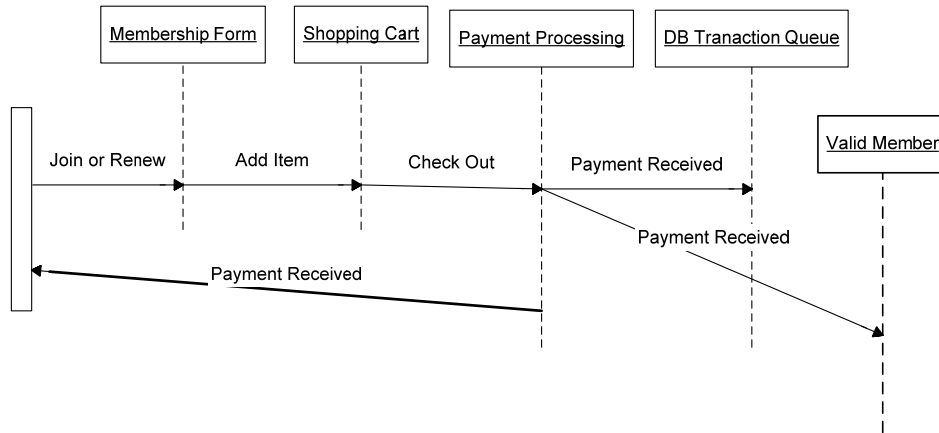
4.2   End of Month Information Flow

Figure 11 - End of Month Information Flow

End of Month

Create EOM Reports

Request EOM Reports or Trigger on Last Day of Month

Copy files to Web

1

Member

Triggers

Notify BOD of New Reports

Renewal Notice Trigger

2

Email Members Renewal Notice
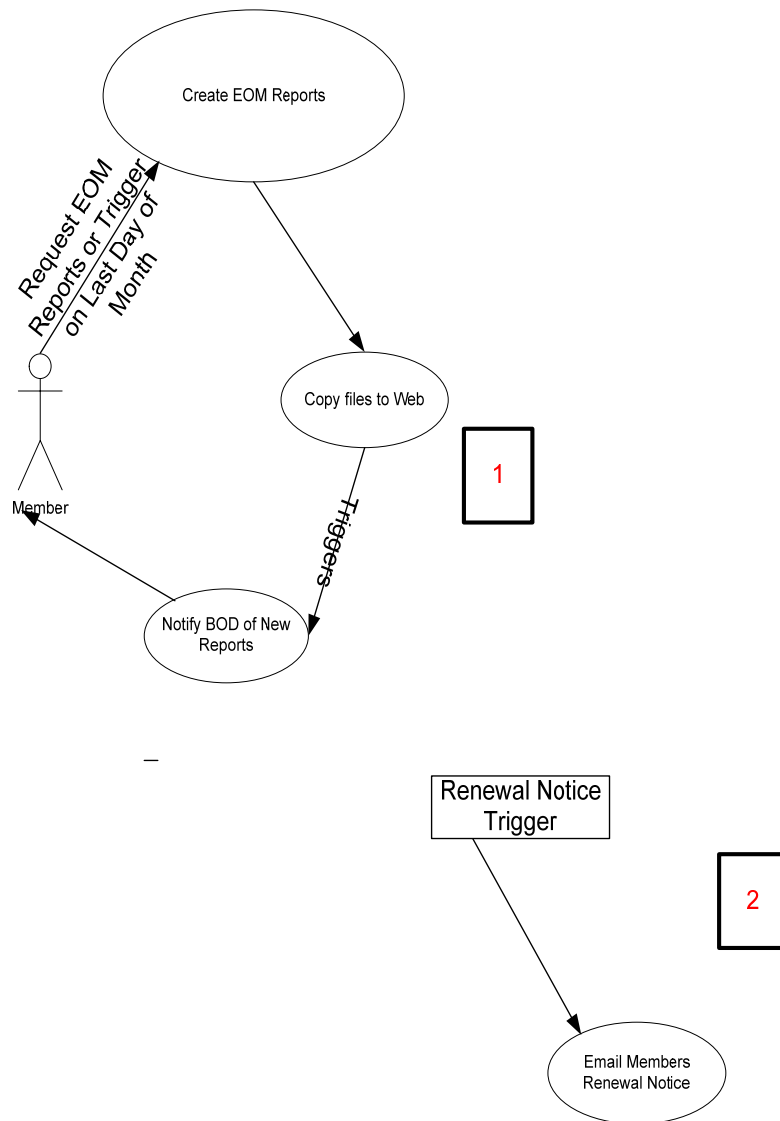
4.3 Screen Prototypes & Store Procedures

The following prototype pages are given for general functionality and flow only. Color and design of the web pages will need to be done by the webmaster. Each Functional area is described with a page showing all the functionality accessible then the procedure accessed is described functionally. The implementation of the procedure may be accomplished with scripting, stored procedures, or other database interface mechanisms.

Implementation will proceed with phases. Phase I first, followed by Phase II, then Phase III, etc. This will get a small subset of functionality up and running and allow a productive use of the data early in the process.

Note: Security Policies such as Username and Password composition is being handled by a different working group. Either a reference to their documents will be placed in this document or a new section will be added later.

4.3.1 *Member Page*

<Need business rules for User Name format and password from security policy team (minimum length of password containing alpha, numbers and special characters).>

Figure 12 - Member Login Prototype

Login to Member-Only Area	Username [ ]
	Password [ ]

What's New	**Forgot Your Password?**
Benefits	**Join NTPCUG**
F.A. Q.	**Update Member Profile Page**
Contact Us	**Renew NTPCUG Membership**

< Update Member Profile Page: Need capability for Admin to send out an automated email to members to check and confirm that their membership data is accurate and up-to-date.**>**

>

[Note: When Users login, they automatically will be displayed a page based on the person's role defined in the database. The roles needed for use by the membership database are: individual, MSSAdmin, Info Desk, Treasurer or BOD.]

4.3.2 *Individual Page*

4.3.2 Individual Page

Figure 13 - Individual Page Functions

	Update My Information
	Renew My Membership

Note: This is just a list of functions to be preformed. Actual visual design will be done by webmaster
<Other selections in the future such as adding volunteer skills>

4.3.3 Individual Update Page

<Retrieve the membership information into the membership form like that at: http://www.ntpcug.org/joinNTPCUG.shtml; when 'submit' button is clicked transfer to update acknowledgement>
Note: Needs Security Policy to prove identity before updates allowed

Figure 14 - Update My Information DB Functions

Phase	Procedures Available
IV	Retrieve Member's Information by selectable fields
IV	Send existing Member their membership ID and password by their email (With enough identifying information to prove they are the correct person)
IV	Submit updated information to pending queue
IV	BOD or MSS Admin retrieve Members pending transactions

4.3.4   Individual Renew Page

<Retrieve the membership information into the membership form like that at: http://www.ntpcug.org/joinNTPCUG.shtml; when 'submit' button is clicked transfer to PayPal to make payment>

Figure 15-   Renew My Membership DB Functions

Phase	Procedures Available
II	Retrieve, Edit, & Update Member Information by ID
I	Submit updated information to pending queue
II	Renew a member

Submit renewal information to pending queue

4.3.5   MSS Admin Page

Figure 16- Admin DB Functions

Phase	Procedures Available
I	List of members for Security Implementation (LDAP extract)
I	Current Members CSV Extract (Same as above)
I	Generate Email Renewals Extract
I	Generate Snail Mail Renewals Extract
I	Generate Expired Members Extract
I	Generate SLMail Extract
I	Retrieve Member by ID
I	Retrieve List of Members by Last Name
I	Retrieve List of Members by First Name
I	Retrieve List of Members by Zip Code or Town
I	Search Log files
I	Archive DB & Log files (temporal or manual)
II	Retrieve, Edit, & Update Member Information
II	Renew a member
II	Add new member
II	Review, Edit & Update Members Information Received
II	Update Reports
III	Generate Email Renewals
III	Generate Snail Mail Renewals
III	Generate Retention Report
III	EOM Processing (Archive & All reports)
IV	Ad hoc reports (parameterized)
IV	Search Archived Membership
IV	Archive Expired Members
IV	Move Archived Member to Active

<Other selections in the future such as Volunteer related>

Figure 17 - MSS Admin I/O for Functions

Phase	Procedures Available	Input	Output
I	List of members for Security Implementation (LDAP extract)	Date	CSV List
I	Current Members CSV Extract (Same as above)	Date	CSV List
I	Generate Email Renewals Extract	Date	CSV List
I	Generate Snail Mail Renewals Extract	Date	CSV List
I	Generate Expired Members Extract	Date	CSV List
I	Generate SLMail Extract	Date	CSV List
I	Retrieve Member by ID	ID	Member Record
I	Retrieve List of Members by Last Name	Last Name	Scrollable List
I	Retrieve List of Members by First Name	First Name	Scrollable List
I	Retrieve List of Members by Zip Code or Town	Zip Code or Town	Scrollable List
I	Search Log files	Search Word or phrase	Scrollable Text
I	Retrieve a copy of the log files		Text File
I	Archive DB & Log files (temporal or manual)	Name of Archive?	Status
II	Retrieve, Edit, & Update Member Information	Member ID	Member Record
II	Renew a member	Member ID	Status
II	Add new member	Member Record	Status & new ID
II	Review, Edit & Update Members Information Transactions Received	Pending Updates	Updates to member records in DB
II	Review, Edit & Update Members Information Transactions Received	Pending Renewals	Updates to member records in DB
II	After Update Reports		Status
III	Generate Email Renewals	Date	Summary or Detailed List
III	Generate Snail Mail Renewals	Date	Summary or Detailed List
III	Generate Retention Report	Date	Report
III	EOM Processing (Archive & All reports)	Date	Archiving & Reports
IV	Ad hoc reports (parameterized)	TBD	TBD
IV	Search Archived Membership	Name or ID	Scrollable List
IV	Archive Expired Members		Scrollable List
IV	Move Archived Member to Active		Status

89

4.4 BOD Page

Figure 18 - BOD Functions

Phase	Procedures Available
I	Retrieve Member by ID
I	Retrieve List of Members by Last Name
I	Retrieve List of Members by First Name
I	Retrieve List of Members by Zip Code or Town
I	Search Log files (Treasurer only)
II	Retrieve, Edit, & Update Member Information (Treasurer or Membership Desk)
II	Renew a member (Treasurer or Membership Desk)
II	Add new member (Treasurer or Membership Desk)
II	Review, Edit & Update Members Information Received ??
IV	Ad hoc reports (parameterized)
IV	Search Archived Membership

4.5 Report File Formats

In order to implement the membership database as fast as possible. All reports are comma delimited extracts. Later new reports will be generated so that other software does not have to be used to generate the renewal letters and forms.

4.5.1   Membership List

MemberId
LastName
FirstName
MiddleName
Address1
Address2
City
State
Zip
PhoneHome
PhoneWork
EmailAddress1
Company
Occupation
DateFirstJoined
Status
ExpirationDate
Last RenewalDate
MembershipType

4.5.2   Renewal List

This is a list of all people that need to be notified about renewal. The single Asterisk fields are required for Snail Mail renewal; Double asterisk for email renewal; only Active status people to be pulled.

DB Fields	Email	Snail Mail
MemberId	√	√
LastName	√	√
FirstName	√	√
MiddleName	√	√
Address1	√	√
Address2		√
City		√
State		√
Zip		√
PhoneHome		√
PhoneWork		√
isValidAddress		
DoNotContact		
EmailAddress1	√	√
IsValidEmail		
Company		√
Occupation		√
DateFirstJoined		
Status		
ExpirationDate	√	√
MembershipType	√	√

4.5.3   Members Expired

MemberId
LastName
FirstName
MiddleName
Address1
Address2
City
State
Zip
PhoneHome
PhoneWork
isValidAddress
DoNotContact
EmailAddress1
IsValidEmail
Occupation
DateFirstJoined
Status
ExpirationDate
MembershipType

4.5.4   Email List

One line for each email defined for every active member in the DB and only valid emails. This will replace the current extract known as the 'SLMail' list.

Email
First Name LastName
MemberId
ExpirationDate

5   Logical Data Modeling

The following are the normalized tables for the database.

5.1 MembershipType

<div align="center">Purpose: Defines valid membership types</div>

MembershipTypeID	smallint [Primary Key]
MembershipTypeName	varchar(30) NOT NULL
AnnualDues	money NOT NULL
MaximumMembers	Smallint NOT NULL

5.2 Valid Membership Types

MemberTypeID	MemberTypeDescription
1	Regular [old R]
2	Student [old S]
3	Family [old F]
4	Exempt (Life) [old E]
5	Corporate (future)
6	Mail Exchange [old M]

5.3 Memberships

Purpose: Defines valid current membership entities. It maybe a family or corporate membership as well as an individual membership

MembershipID	int IDENTITY(1,1) NOT NULL [Primary Key REFERENCES dbo.MembershipTypes(MembershipTypeID)]
MembershipTypeID	smallint NOT NULL
MembershipDescription	varchar(50) NULL
MembershipStatus	char(1) NOT NULL ; I for Inactive or A for Active
Expiration	smalldatetime NOT NULL
LastModified	smalldatetime NOT NULL DEFAULT
Comment	nvarchar(255) NULL

94

5.4 MembershipArchive

Purpose: Archive non-current memberships entities

MembershipID	int IDENTITY(1,1) NOT NULL [Primary Key REFERENCES dbo.MembershipTypes(MembershipTypeID)]
MembershipTypeID	smallint NOT NULL
MembershipDescription	varchar(50) NULL
Status	char(1) NOT NULL
Expiration	smalldatetime NOT NULL
LastModified	smalldatetime NOT NULL DEFAULT
Comment	nvarchar(255) NULL

5.5 Members

Purpose: Record individual members information

MemberID	int NOT NULL; PRIMARY KEY CLUSTERED (MemberID)
MembershipID	int NOT NULL; REFERENCES dbo.Memberships(MembershipID)
IsPrimaryMember	bit NOT NULL; flag for the primary member of a membership
Title	nvarchar(10) NULL
FirstName	nvarchar(50) NOT NULL
MiddleName	nvarchar(50) NULL
NickName	nvarchar(50) NULL
LastName	nvarchar(50) NOT NULL
Postnominal	nvarchar(12) NULL
Address1	varchar(50) NOT NULL
Address2	varchar(50) NULL
City	varchar(50) NOT NULL
State	char(2) NOT NULL
Zip	char(10) NOT NULL
Occupation	varchar(60) NULL
JobTitle	varchar(60) NULL
Company	nvarchar(60) NULL
Source	varchar(50) NOT NULL;-- source of record information
IsPrivate	bit NOT NULL DEFAULT 1
IsValidAddress	bit NOT NULL DEFAULT 1
IsWillingToVolunteer	bit NOT NULL DEFAULT 0
DoNotContact	bit NOT NULL DEFAULT 0
NoFollowupRequired	bit NOT NULL DEFAULT 0
Comment	nvarchar(255) NULL
LastModified	smalldatetime NOT NULL

Note: Clarify need to clarify the difference in DoNotContact vs. NoFollowupRequired

95

5.6 EMails

Purpose: Record individual members emails

MemberID	int NOT NULL; FOREIGN KEY REFERENCES Members(MemberID)
EmailAddress	nvarchar(64) NOT NULL
PreferenceLevel	tinyint NOT NULL; -- For multiple user e-mails; encodes which are preferred over others
IsValidEmail	bit DEFAULT 1 NOT NULL

5.7 PhoneNumber

Purpose: Member phone numbers by MemberID

MemberID	int NOT NULL; FOREIGN KEY REFERENCES Members(MemberID)
PhoneNumber	char(12) NOT NULL
Extension	int NULL
TypeCode	tinyint NOT NULL; -- For encoding home business cell pager fax
PreferenceLevel	tinyint NOT NULL; -- For multiple user phone numbers; preferred number
IsValidPhone	bit DEFAULT 1

5.8 PaymentType

Purpose: Lookup table for valid payment types -- cash check PayPal

PaymentTypeID	smallint IDENTITY NOT NULL
PaymentTypeDescription	varchar(20) NOT NULL

5.9 Valid Payment Types

PaymentTypeID	PaymentTypeDescription
1	Cash
2	Check
3	PayPal
4	Old Credit
5	Other

5.10   PaymentSource

Purpose: A lookup table for valid payment sources -- meeting, mail, web, etc.
 Has one-to-many relationship with Payments and PaymentsArchive tables.

PaymentSourceID	smallint NOT NULL primary key
PaymentSourceDescription	varchar(10) NOT NULL

5.11   Valid Payment Sources

PaymentSourceID	PaymentSourceDescription
1	Meeting
2	Post Office
3	Electronic/Web (PayPal)
4	Other

5.12   Payments

Purpose: Dues Payments.

PaymentID	int NOT NULL ;Key
MembershipID	int NOT NULL; REFERENCES Memberships(MembershipID)
Amount	money NOT NULL
PaymentTypeID	smallint NOT NULL;REFERENCES PaymentType(PaymentTypeID)
PaymentSourceID	smallint NOT NULL;REFERENCES PaymentSource(PaymentSourceID)
Payment Date	smalldatetime NOT NULL
IsPending	bit DEFAULT 0-- For PayPal transactions
NewExpiration	smalldatetime NOT NULL

5.13   PaymentsArchive

Purpose: Dues Payments Archives.

PaymentID	int NOT NULL key
MembershipID	int NOT NULL; REFERENCES Memberships(MembershipID)
Amount	money NOT NULL
PaymentTypeID	smallint NOT NULL;REFERENCES PaymentType(PaymentTypeID)
PaymentSourceID	smallint NOT NULL; REFERENCES PaymentSource(PaymentSourceID)
PaymentDate	smalldatetime NOT NULL
IsPending	bit;-- For PayPal transactions
NewExpiration	smalldatetime NOT NULL

5.14   MemberNotice

Purpose: Record all notices sent to members from NTPCUG. Has a many-to-one
relationship with Members table

NoticeID	int NOT NULL key
MemberID	int NOT NULL; REFERENCES Members(MemberID)
NoticeType	int NOT NULL; -- see table below
NoticeDate	smalldatetime NOT NULL; -- when the notice was generated
NoticeInitiator	nvarchar(50) NOT NULL; -- who initiated this notice
NoticeDetail	nvarchar(255) NULL

5.15   Member Notice Types

MemberNoticeType	NoticeTypeDescription
1	e-mail 60 days before expiration
2	snail mail 60 days before expiration if no email
3	e-mail 30 days before expiration
4	mail 30 days before expiration
5	e-mail the month of expiration
6	snail mail the month of expiration
7	snail mail 30 days after expiration
8	snail mail 60 days after expiration

5.16   VolunteerActivity

ActivityID	int NOT NULL; Key
ActivityDescription	varchar(100) NOT NULL
ActivityDate	smalldatetime NOT NULL
EnteredDate	smalldatetime NOT NULL

5.17   VolunteerActivitiesMember

ActivityID	int NOT NULL; REFERENCES VolunteerActivity(ActivityID)
MemberID	int NOT NULL; REFERENCES Members(MemberID)
IsActivityLeader	bit DEFAULT 0

5.18   VolunteerSkill

MemberID	int NOT NULL; REFERENCES Members(MemberID)
Skill	varchar(20)

6 Solutions Architecture

Ecommerce systems need to be flexible enough to support changing environments and

workloads. The earliest Ecommerce systems were separated into the client and the server

to try to address this need. The client-server architecture separated the data management

from the user interface. However, having only two layers suffered from many problems,

the largest of which is that the business logic could reside in both layers and therefore

was not easily modified or replaced. Now the most common system architecture for

business applications is to decompose it in multiple layers consisting of at least three

separate pieces. This architecture is referred to as an n-tiered architecture.(Hohmann,

2004)

Users of the system should not be able to tell which processes are running where and in

which layer. The basic three-tier client-server architecture is composed of the following

layers:

- Presentation layer or also known as the Graphical User Interface, GUI,
  which runs on a client machine such as a PC

- Middle layer which is the business logic running on the application server

- Data layer which is the location for the database or other persistent data
  store and is usually located on a back-end server

The presentation layer presents information and manages the interaction usually in a

piece of software such as a web browser running on a personal computer. Another

example of a GUI layer is a handheld computing device such as a PDA. The GUI does

not contain any of the business logic. In some Enterprise Applications the user interface

may contain two layers, one for the presentation of the information and another that

interfaces the workflow between the UI and the middle layer.

The middle layer is used to implement an Enterprise application logical view of the data.

This layer is where the application business rules are implemented and should be

independent of how the data is physically represented. This layer can also be further split

into more than one layer, such as a services layer and a domain layer. The services layer

would define the services that are visible to the rest of the system. The domain layer is

entirely optional and would be used when the business rules are too complex to be

represented in one middle layer such as when making a logical view from two legacy

relational databases.

Finally the last layer is the persistent data layer which is the relational database.

Ecommerce systems may also create a layer to manage the mapping between Enterprise

domain objects in the middle service layer and the objects in the relational database.

Triggers and stored database procedures may be included in this layer to encapsulate that

mapping.

To determine the number of layers to implement in this system, we will discuss the

design with the team and prototype our choice. What drives the separation of a layer is

the clear separation of responsibilities and system interfaces. If each separate layer is

designed properly then it can be implemented by a team independently and in parallel to

the other layers based on those system interfaces.

Another major consideration of Ecommerce systems is the management of transactions.

Transactions are simply an agreement between a buyer and a seller in exchange for a

payment of some kind. How that is implemented in an Ecommerce system is a lot more

complicated. A transaction is an event that cuts across all the layers in the system and if interrupted can cause an inconsistency in files or databases. All transaction in Ecommerce systems must be logged and a mechanism must be in place to reverse the changes that have been made if the entire operation is not successful. Transaction control will have to be incorporated in not only the database but often in the middle layer.(May, 2000)

All of the tiers of the architecture will be developed simultaneously by several teams. This presents a problem for coordinating the design and testing of the various tiers. Luke Hohmann suggests that when starting with substantial amounts of existing data to start the design of the system with the database. This will allow the database design to drive what gets implemented first in the application and GUI layers. One approach that he suggests is to define a small core subsystem of the architecture to implement first and then define all the subsystem interfaces for this small core. He refers to this build process as 'spiking'. A spike should be a user-visible piece of functionality described by a single use case. The important feature of the spike is that it drives functionality through all the tiers of the architecture.(Hohmann, 2004)

Using spiking to implement the system is an incremental approach. From the first small core continue to add functionality in phases. Spiking, thus, becomes a way to manage risk in the implementation of multi-tiered system. As each spike is implemented and tested the system becomes more complete and stable. We will use periodic builds of the entire system to keep all the interfaces in sync.

In parallel development there will be many times when the different layers are at different levels of development or testing. The MSS will use the UI layer to drive the other layers.

During the implementation of a specific use-case if a missing attribute is discovered the GUI team will work with the middle tier team to make sure that the functionality is implemented there. Then middle-tier team will work with the database team if attributes or stored procedures are missing in the data layer.

Since NTPCUG was donated a copy of Microsoft Server 2003 the implementation languages for the middle application tier will likely be on Visual Studio .NET 2003. The way this could be accomplished is a fairly complex topic. Without getting into the subject too deeply, start with the database model. This defines the lowest tier of data. Then define a dataset object which will serialize to XML and handle the communication to the application middle layer. Then in the middle tier these application/domain objects are tied back to the data store tier. The actual business rules then are written as methods to the objects. Since you can construct a collection to encapsulate all the other tables, to instantiate the objects at execution time, all that is needed is for the presentation layer to start the one collection on the server. (Hyatt, [Unknown year])

Figure 19 - System Architecture Layers
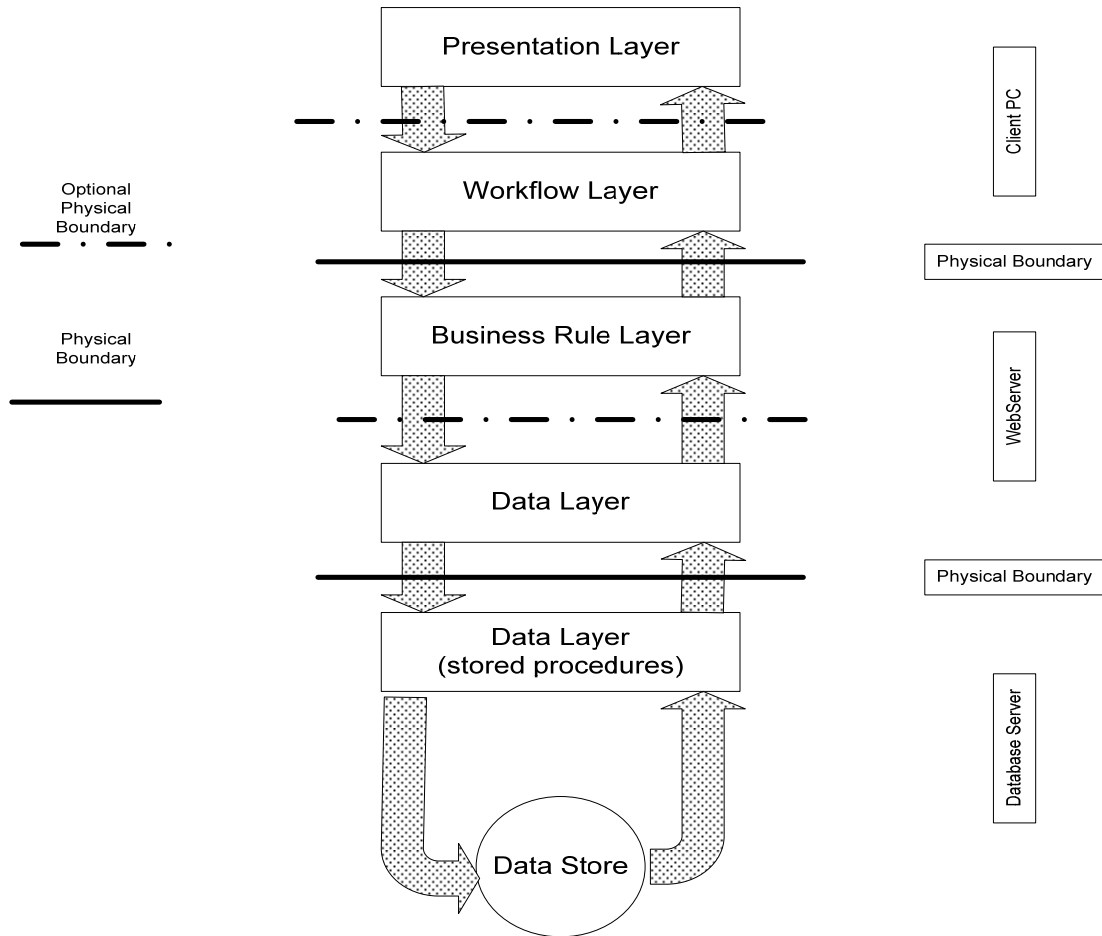
7 Web & Transaction Design

Web Navigational Design <TBD by Web Master>

Transaction Design <TBD after decision on type of PayPal Transaction>
Security/Transactional Integrity <TBD after decision on type of PayPal Transaction>
Transaction Processing Communications <TBD after decision on type of PayPal
Transaction>

8 Business Readiness

8.1 Stakeholders & Use-cases Responsibility for Testing

9 Implementation Plan

This system will be implemented in phases to get small subsets of the transactions, the database, and web pages working as early as possible.

9.1 Phase 1- Convert Legacy Database & Design new relational database

Convert the old database; Build the new database

9.2 Phase 2- Define Business Process, GUI functionality, and transactions

9.3 Phase 3-1ˢᵗ Pass of Adminstrator & BOD Pages

1. Generate EOM Reports (Membership List, IMail list, New Members, and Expired Members)
2. Generate CSV Email Renewal List
3. Generate CSV Snail Mail Renewal List
4. Update Security access list (Active Directory)

Note: An interim security control will need to be in place to prevent accidental exposure of member information

9.4 Phase 4-Member Updates & SharePoint Interface; DBA Review & Post Updates

1. Authenticate member and grant privileges according to Security Policy (includes a way for a member to retrieve their member ID or password)
2. Retrieve Member's Information by selectable field
3. Validate Data Entry fields (emails, addresses, zip, phone)
4. Submit updated member information to pending queue
5. MSS Administrator Review update queue
6. MSS Administrator post updates
7. MSS Administrator or Authorized Person review transaction logs

9.5 Phase 5-Member Join, Renewal, Review Pending Transactions; Post Payments and memberships

8. Retrieve a copy of Financial logs and MSS access logs
9. Search Financial Logs; Search MSS access logs
10. Submit renewal membership to pending queue
11. Submit new membership to pending queue
12. Retrieve pending queue
13. Finalize pending transaction
14. Receive automated payment transaction and place in pending queue

9.6 Phase 6-Generate Reports

1. Generate Renewals via Email
2. Generate Renewals for Snail Mail
3. Generate Retention Reports
4. Generate automated New Member Report
5. Generate automated Imail List
6. Generate 1-6 in one step for EOM

9.7 Phase 7-Parameterized Reports

1. Parameterized Reports

10Testing Plan

10.1 UAT environment

10.2 Classes of tests

10.3 Expected software response

10.4 Performance bounds

11 Appendix

11.1Terms

11.2 Traceability Matrix

11.3 Supplementary information

12 Bibliography

Hohmann, L. (2004). *Beyond Software Architecture: Creating and Sustaining Winning Solutions*. Boston: Addison-Wesley.

Hyatt, K. ([Unknown year], 2005). N-Tier Application Development with Microsoft.NET.
http://www.microsoft.com/belux/nl/msdn/community/columns/hyatt/ntier1.mspx
and
http://www.microsoft.com/belux/nl/msdn/community/columns/hyatt/ntier2.mspx

May, P. (2000). *The Business of E-Commerce: From Corporate Strategy to Technology*. New York: Cambridge University Press.