March 2015

Prospective Member Sourcing System

Amanda Michelle Weis
Worcester Polytechnic Institute

Jiedong Wang
Worcester Polytechnic Institute

Thomas Francis Meagher
Worcester Polytechnic Institute

William Hames Richtmyer
Worcester Polytechnic Institute

Zeng Liu
Worcester Polytechnic Institute

PROSPECTIVE MEMBER SOURCING SYSTEM

A MAJOR QUALIFYING PROJECT REPORT
SUBMITTED TO THE FACULTY
OF THE
WORCESTER POLYTECHNIC INSTITUTE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF BACHELOR OF SCIENCE

BY

ZENG LIU, MIS & CS
THOMAS MEAGHER, MIS
WILLIAM RICHTMYER, MIS
JIEDONG WANG, MIS
AMANDA WEIS, MIS

DATE: MARCH 6, 2015

SPONSORED BY:
COMMUNISPACE CORPORATION

PROJECT ADVISOR:
PROFESSOR ELEANOR T. LOIACONO

Table of Contents

Table of Figures

Table of Tables

Acknowledgements

The team would like to thank the Ms. Laura Naylor and Mr. David Rosenberg for championing the project within Communispace and Mr. Sean Burke, Mr. Jack Bergersen, and Professor Eleanor Loiacono for their continued support and guidance on throughout the project experience. The team would also like to thank everyone who took the time to meet and share their opinions and insight, especially Mr. Justin Hill, Ms. Michelle Fisher, and Mr. Slava Asipenko. Without help from these stakeholders and people, the project would not have been such a successful and enjoyable experience.

Abstract

This Major Qualifying Project (MQP) details a Prospective Member Sourcing system prototype that the team developed for the Communispace Corporation's Client and Consumer Services team. The project objective was to enable more efficient recruitment and resource allocation by providing the Client and Consumer Services team the ability to source community members through data centralization. Throughout this paper, the team overviews the synthesis and application of their academic knowledge, as well as their methodology, for developing the system prototype.

Authorship

The authorship of the project report was spread amongst the various MQP team members. The Acknowledgements section was written by Amanda Weis. Next, the Abstract and Executive Summary were both written by Thomas Meagher. The Introduction sections was written by Zeng Liu, William Richtmyer, Thomas Meagher, and Amanda Weis. Zeng Liu and William Richtmyer wrote the Communispace Overview and the Major Qualifying Project Overview respectively. Thomas Meagher wrote the Problem Statement and Project Objectives, while Amanda Weis wrote the Project Overview.

In the Literature Review section, Zeng Liu and William Richtmyer wrote Communispace Corportation section. Zeng Liu wrote about previous MQPs in the Previous MQPs section. Jiedong Wang wrote the section on IT Business Alignment. Next, Thomas Meagher and William Richtmyer wrote The Systems Development Life Cycle section. Lastly, in the Literature Review, Amanda Weis wrote the Project Methodologies section.

In the Planning Phase section, Zeng Liu wrote the Scope and Objectives for the report. Thomas Meagher wrote the Proposed Methodology. The Systems Development Scenarios and Project Stakeholders were written by William Richtmyer. Next, Jiedong Wang wrote the Project Plan. While Zeng Liu wrote the Systems Request, Amanda Weis completed the Feasibility Study section and Jiedong Wang wrote the Risk Assessment.

The Analysis Phase section was written by Thomas Meagher, Zeng Liu, Amanda Weis, and William Richtmyer. Thomas Meagher wrote the Analysis Strategy, Uses Cases,

Requirements Analysis and Organization, and System Proposal sections. Zeng Liu wrote the Requirements Definition. Next, Amanda Weis wrote the Profile Fields Questionnaire section. William Richtmyer completed the Process and Data Models section.

In the Design Phase section, there were multiple authors. Jiedong Wang wrote the Design Plan. Zeng Liu and Amanda Weis worked together to complete the Alternative Matrix Section. Zeng Liu worked on the System Architecture Design and Program Design sections. Thomas Meagher and Zeng Liu teamed up to write the User Interface Design and User Interface Mockup sections. Lastly, William Richtmyer wrote the Entity Relationship Diagrams sections.

In Implementation Phase section, Jiedong Wang and Amanda Weis wrote the Development and System Iterations sections. Thomas Meagher wrote the Population of the Database and Conversion Plan sections. William Richtmyer wrote the Conversion Plan section. The Support Plan and Documentation section was written by Zeng Liu. Thomas Meagher wrote the Training Plan section.

Lastly, the Recommendations and Conclusion Section was written by Thomas Meagher and Amanda Weis. Thomas Meagher wrote the Recommendations and Conclusions Sections, while Amanda Weis wrote the Project Presentation Section. Lastly, the Lessons Learned section was written by both Thomas Meagher and Amanda Weis.

Executive Summary

Often focusing on projects abstracted from the real world, the Management Information Systems (MIS) curriculum at Worcester Polytechnic Institute (WPI) provides students with the theoretical knowledge and skills to succeed. Through their MQP, Communispace presented the team with the unique opportunity to synthesize their theoretical education during the practical application of their skills towards solving a real-world business problem.

Based on the proposed areas of need, the primary objective of the WPI MQP team was to develop a system prototype that enables the Client and Consumer Services team to recruit members and allocate resources more efficiently and effectively. During the project's completion, the team followed a four-phase methodology to remain focused and stay within the project scope. The methodology aligned with the Systems Development Life Cycle, including planning, analysis, design, and implementation phases.

Beginning with the planning phase, the team gathered system requirements, through meetings and interviews with major stakeholders and potential users. Utilizing these requirements, the team determined their project scope, plan, and timeline for completion. Next, in the analysis phase, the team processed the information gleaned from on-site meetings and interviews, and translated it into formal system requirements. During this phase, the team developed a system proposal and corresponding request to define the project objectives and present their understanding of the opportunity.

After the stakeholders accepted the proposal and documentation, the team segued into the design phase. Comprising two major parts, back- and front-end design, the team acted meticulously in making sure the system requirements were captured in design. The design process was highly iterative, refining the user interface mockup, database architecture, and other components as the team received feedback from their stakeholders.

Concluding with the implementation phases, the team translated the previous four phases in the Prospective Member Sourcing System (PMSS) prototype. The prototype functionality was segmented into different pages (Figure 1): Search Repository, Add Members, Manage Members, and Reports. On these pages, users are able to perform all the requisite capabilities, allowing them to search, add, manage, and view a summary of system data.

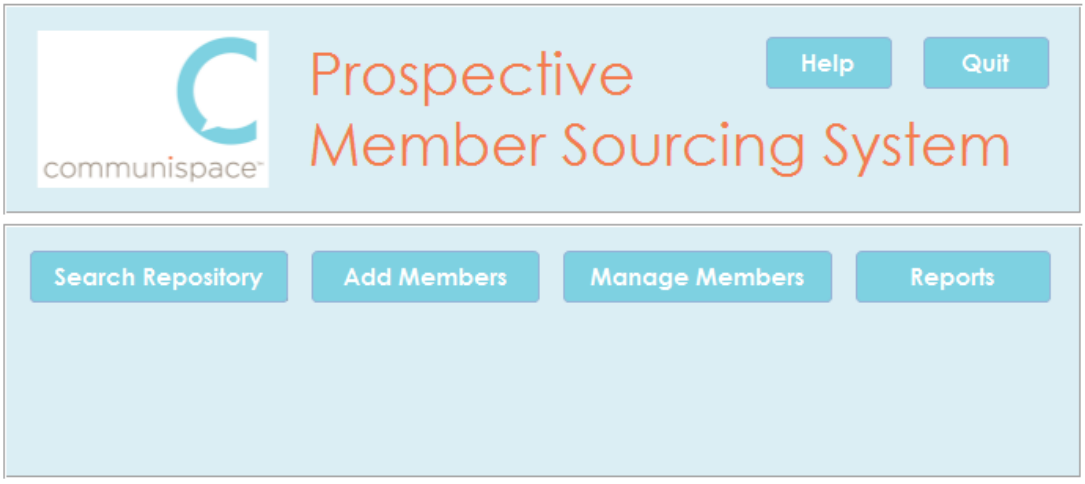


Figure 1 : Prospective Member Sourcing System

1. Introduction

With the introduction, the team presents a brief overview of both the Communispace Corporation and the Major Qualifying Project (MQP), followed by a Problem Statement and Project Overview. This section provides a basic foundation for understanding the context of the project.

1.1 Communispace Overview

Founded in 1999, the Communispace Corporation specializes in helping companies better understand their customers through consumer collaboration. In this capacity, Communispace creates and maintains online private communities to collect customer insights and feedback. Led by the Client and Consumer Services division, the company's expertise lies in community member recruitment and facilitation. The team has recruited members from 96 countries and built more than 700 diverse communities, comprising over 100 Fortune 1000 clients and more than 130,000 active online community members.

In 2007, Communispace underwent a global expansion into Europe, Australia, Asia and Pacific countries with leading consumer brands in the electronic, airline, and food industries. In 2012, Communispace transitioned to a proprietary software platform, Catalyst, to provide a secure environment for consumers to interact with the company's

clients. Catalyst also allows for easier way to generate community-specific reports and analyze results.

Communispace has been an award-winning leader in their industry (Figure 2). In 2009, Communispace was awarded The Leader in Providing Market Research Online Communities (MROCs) by Forrester Research. They have also been included on the Honomichl List of Top 50 Market Research Companies four years in a row.

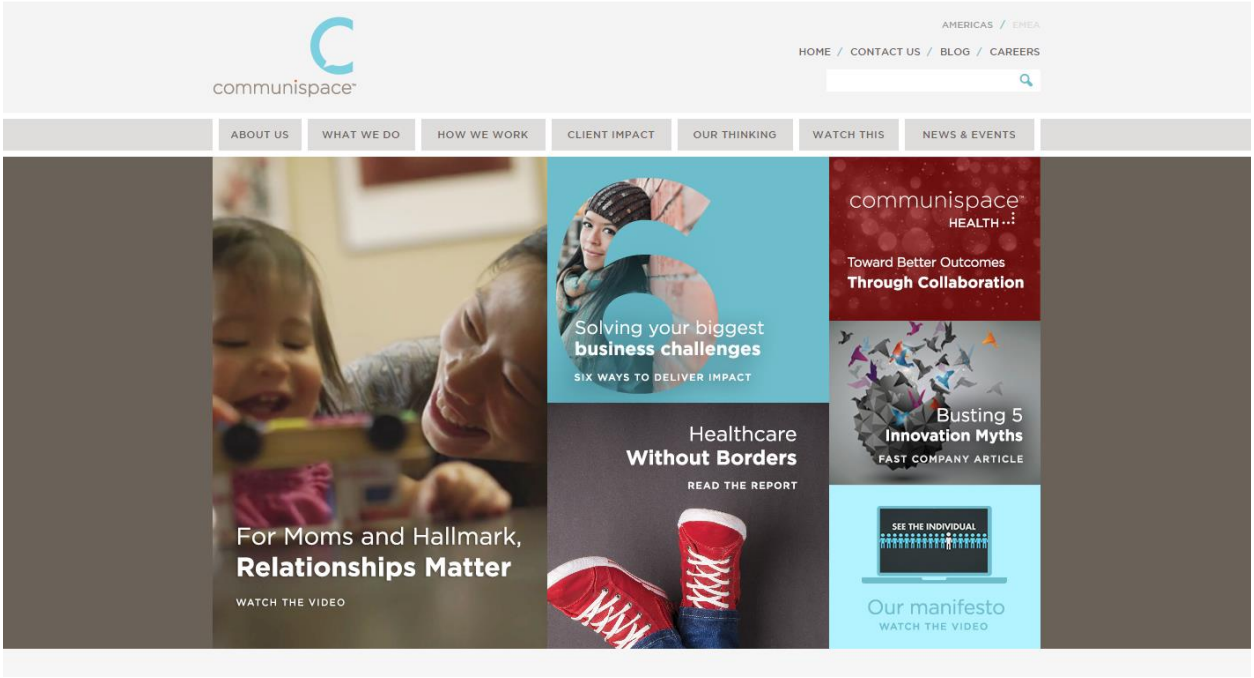


Figure 2: Communispace.com

1.2 Major Qualifying Project Overview

A private engineering college, located in Worcester, Massachusetts, Worcester Polytechnic Institute's motto is *lehr und kunst*, a German-phrase translating to *theory and practice*. Since its origination in 1865, WPI has closely followed that motto, supplementing

theoretical academics with hands-on technical experience. In 1970, WPI created the *WPI Plan*, integrating the Interactive Qualifying Project (IQP) and Major Qualifying Project (MQP) to extend the practical motto into a project-based approach. Unlike most higher educational institutions, WPI's curriculum is four seven-week long terms, rather than the typical two 14-week long semesters.

The MQP is a senior-year capstone project geared towards challenging students to synthesize their academic knowledge during a real-world project. Throughout the project, students have the support of a faculty advisor, within their major, to submit a formal deliverable to the project sponsor and WPI. In the field of Management Information Systems, projects are usually sponsored by companies that have established relationships with WPI. Past project sponsors include General Electric, Fidelity, EMC, and the group's project sponsor, Communispace Corporation. MQPs typically last three terms, taking up a third of a typical course load. In addition to MQP, students usually take two other classes.

1.3 Problem Statement

One of the most important assets of a Communispace community is its members. Depending on their customer's community requirements, Communispace sources prospective members from many different channels—vendors, advertising, social media, and client lists. Within their current methodology, there is an opportunity to optimize how Communispace initially sources and potentially re-uses members for their communities. By

making this process more efficient, Communispace can save time and money, as well as pass on more value to their clients.

Often, Communispace has to discard users that do not qualify for recruitment to a specific community. When these overflow candidates are discarded from a non-vendor source (social media, blog, banner advertisement), their data is no longer tracked within Communispace's existing recruitment system. Since Communispace already purchased the non-vendor media, an opportunity exists for the unqualified overflow candidates to be saved into a repository for future use. Among this, it would allow Communispace to decrease the overall sourcing costs per community member.

Retaining community member data in a central, normalized repository for future use would allow Communispace to create their own independent sourcing channel. A Communispace member repository could capture proven members whom have expressed interest in Communispace communities. Repository members could also enter from multiple other processes, including the aforementioned unqualified overflow scenario. The repository would lower the sourcing cost, as these members would be more accessible and proven. Other mediums, like panel vendors, are more expensive to use and candidate quality is unknown.

Refining the community recruitment process so each individual recruitment event is viewed as interconnected, rather than isolated, would allow Communispace to gain

significant recruiting efficiencies. By analyzing events together, the Client and Consumer Services team can learn from similar instances from a much broader perspective. Moving community recruitment from an event-based to ongoing process across audiences may yield efficiency gains. Instead of waiting for a client-related event to trigger recruitment, prospective members could be continuously recruited, preempting need.

Updating the sourcing process for Communispace would foreseeably increase the value of their communities, since communities could be originated and grown faster than before. Optimizing the sourcing by retaining unqualified candidates, developing a prospective member repository, and integrating ongoing recruitment could directly improve Communispace communities.

1.4 Project Overview

Throughout the project, the team synthesized their academic knowledge in designing a system to be used by Communispace's Client and Consumer Services team to more effectively and efficiently manage their recruits and online community members.

There are three cases in particular on which the team was focused. The first was *saving interested people or overflow* who do not qualify for a particular client. Since they are clearly interested in joining a community, they might be useful in another future situation. The second case was the *retention of former community members*. At present, when a community ends, all community members leave. However, engaged community members,

interested in sharing their thoughts, might be useful in the future. Rather than trying to re-recruit them, it would be easier and more efficient to approach them directly. The final case was sourcing members across several communities during *ongoing recruitment* at once rather than using isolated events to source the recruits for each community. The team's project explored each of these three cases in order to help Communispace optimize their community sourcing.

1.5 Project Objectives

While working on a MQP, it is important to define clear, attainable objectives that help the project progress towards completion. In defining their objectives, the team aligned academic and business concepts with both the standards and expectations of Communispace and the Major Qualifying Project. During completion, the project objectives were further developed as more information was discovered about the team's MQP.

Working collaboratively with Communispace, the team's main objective was to identify their business needs in sourcing and recruitment, and develop a system that provides value to Communispace and the Client and Consumer Services team. At a minimum, the resultant system should provide organizational stakeholders with an actionable system baseline, which can continually be improved by the organization. Upon completion of the MQP, the Communispace team should be able to utilize the MQP team's work and transition to leading future system development. They should also be able to present a compelling case of business justification to internal stakeholders based on the team's recommendations.

From an academic perspective, the MQP should provide the team with an intensive theoretical and practical synthesis of their MIS education. The objective of the team's project work with Communispace was to demonstrate the application of their knowledge in a practical way. Throughout the project, the team utilized relevant resources that allow them to present a comprehensive capstone of their academic experiences. Most of the academic objective was captured in this final MQP report, where the team details their background, methodology, and final solution.

The last objective of the team's MQP was to further develop the relationship between Communispace and WPI. Working with sponsoring organizations offers a valuable and challenging environment for MQP teams and their respective sponsors. Recognizing the importance of these relationships, a goal of the project was to work effectively and efficiently together, maximizing the amount gained from the experience.

2. Literature Review

In this section, the team details information that helped them to understand the daily operations of Communispace. The team also discusses material that provides the underlying knowledge for the development methodology.

2.1 Communispace Corporation

Communispace specializes in consumer collaboration and providing insights to help companies better identify their business objectives. Recruiting community members plays an important part in the company's operation. The process can be optimized to save resources by becoming more efficient. To achieve this refinement, the project will focus on capturing existing user profiles, from recruitment overflow and prior communities, and helping the company move from event-based recruitment to ongoing recruitment across audiences.

2.2 Previous MQPs

Communispace has sponsored two prior MQP teams. The first project was conducted during the 2012-2013 academic year, where a group of students built a Vendor Performance and Relationship Management (VPRM) system prototype to help with the vendor selection process. In the 2013-2014 academic year, another project team extended the VPRM system to enable better recruitment and assist in design decisions, by providing Communispace the ability to evaluate vendors based on their different criteria. The past

Communispace projects, conducted by WPI student teams, have been very successful for all involved parties. Although the team will not be working extensively with the VPRM system, it is good to know the successes of the past WPI Communispace MQPs.

2.3 IT Business Alignment

When developing any system, it is important to consider the alignment of information technology (IT) with business. IT should always work in tandem to help the business achieve its ultimate goals and objectives. In the context of the project, it is important to examine how the system the team is developing will align with the business' objectives and provide them with value. In the context of the project, the team has defined competency and value measurement as an important indicator for alignment; however, other components of alignment do exist. For a greater understanding of IT business alignment and other factors, Appendix B can be referenced.

2.3.1. Competency / Value Measurement

Competency and value measurements are used as a benchmark and balance of IT and business metrics. Three attributes that are particularly noteworthy for IT business alignment are IT, business, and balanced metrics. The IT and business-related metrics tend to be more formal about the surrounding processes and how they are implemented. This means that each metric is specifically targeted towards a defined competency in either IT or business.

The more aligned these metric are with each other, the more mature the alignment of the relationship can be. Conversely, the balanced metrics are less related to specific functions, like IT or business, and are more concerned with the level of integration between IT and business. For competency and value measurement to be a successful indicator of IT business alignment it is important for the metrics to be balanced as well as detailed, relating to the specific competencies. The objective in using value measurements is to benchmark the strengths and weaknesses so that organizational endeavors, like information systems projects, can be accurately assessed.

2.4 The Systems Development Life Cycle

The fundamental four-phase methodology that serves as the basic framework for information system development process is called the *systems development life cycle* (SDLC). The four phases comprise the process of determining how an information system (IS) can support business needs, designing the system, building it, and delivering it to users. Each of the phases: planning, analysis, design, and implementation shown in Figure 3, address one of these questions and can be broken down into a series of smaller steps. When each phase is performed, the smaller step-like series "rely on *techniques* that produce *deliverables*" (Dennis, Wixom & Ross, 2012, p.11).

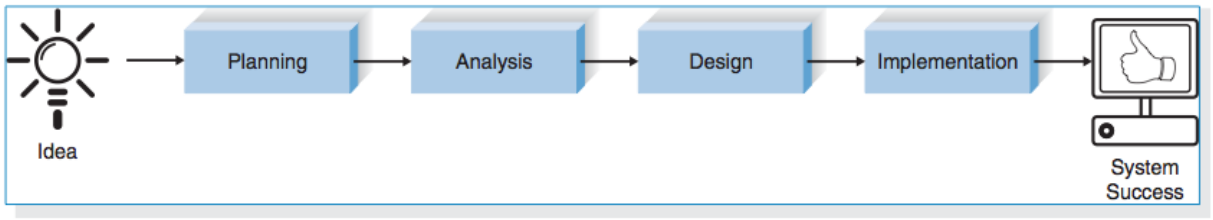


Figure 3: the Systems Development Life Cycle (Dennis, Wixom & Roth, 2012, p.11)

Even though Figure 3 suggests the SDLC follows a linear path from origination to completion, this often not the case. Some projects are able progress logically from phase to phase, while many others might take different routes in a combination of "consecutive, incremental, iterative, and/or other patterns" (Dennis, Wixom & Ross, 2012, p.13). Similar to the varying pattern approaches, information system projects may emphasize different parts of the SDLC or approach the SDLC phases in different ways.

2.4.1. Phase One: Planning

The first phase of the SDLC, where "the project team identifies the business value of the system, conducts a feasibility analysis, and plans the project," is the *planning phase*. Since the planning phase serves as the foundation for completion, there must be a fundamental understanding of the project that not only captures the value and feasibility, but also the project management outline of the project. The planning phase can be further decomposed into three main steps: *project identification and initiation, feasibility analysis, project management* (Dennis, Wixom & Ross, 2012).

Project Identification

For a project to exist in an organization, it must first be identified as a business need. This business need could emerge when a firm must fix an existing problem, pursue an opportunity to gain a competitive advantage, or for many other organizational reasons. Once the need is identified, a *project sponsor*, an individual with a vested interest in the successful completion of the system, manages the initial development that follows the SDLC. The project sponsor works across various departments to gather high-level *business requirements* for the system. These requirements summarize the features the system will have to include and are representative of the different organizational perspectives. If the project sponsor has conducted sufficient work, they should be able to determine the *business value* of the system.

Project Initiation

In many organizations, the project is initiated by the *system request*, which "presents a brief summary of a business need, and it explains how a system [...] will create business value" (Dennis, Wixom & Ross, 13). During project identification, the project sponsor has gained enough insight to formalize their knowledge into the system request document, typically capturing the following elements: project sponsor, business need, business requirements, business value, and special issues. In this format, it is important to note that the project sponsor is no longer the organization, but rather a single contact person that will serve as the organization's project lead. Upon review of the systems request, if all the requisite information needed to proceed is captured, then the project can move forward (Dennis, Wixom & Ross, 2012).

Feasibility Analysis

The system request, specifically the business need and high-level requirements, provides the background for a *feasibility analysis*, where the technical, economic, and organizational viability of the potential project are assessed. In analyzing the feasibility of the project, the organization can make an informed decision on whether to proceed. Furthermore, the feasibility analysis will outline any risks associated with moving forward and how to manage them. Similar to the system request, a *feasibility study* document is often prepared, translating the analysis into a formal deliverable.

Technical Feasibility

The technical analysis of feasibility aims to answer the question: *Can we build it?* The internal or consulting information technology team's familiarity with the application and technology required for the project must be considered. As the project team is more familiar with the requisite technologies, the level of risk decreases. In addition, the project size must also be evaluated—the larger the size, the more risk. Lastly, the compatibility of the project determines whether the system will be able to integrate with the existing system (Dennis, Wixom & Ross, 2012).

Economic Feasibility

Considering the costs and benefits of the proposed system, the economic feasibility analysis strives to answer: *Should we build it?* As a baseline, the development and annual operating costs should be considered, as well as the annual benefits. The *return on investment* (ROI), *break-even point* (BEP), and *net present value* (NPV) are commonly used metrics for determining the value by weighing costs and benefits. Classifying measures as tangible or intangible helps break down complexity in the costs and benefits (Dennis, Wixom & Ross, 2012).

Organizational Feasibility

During the assessment of the organization's feasibility, the question: "If we build it, will they come?" is assessed. Across the organization, the analysis determines whether the project will be successfully received by the different stakeholders and integrated into future operations for its intended use. If the project objectives aligns with business strategy, then there will likely be less resistance and risk involved. A *project champion*, a high-level executive that sometimes acts as the project sponsor, provides senior-level support and resources for the project. Finally, an assessment of the *system users* should be conducted. Since the system users will be using the proposed system the most often, they must be considered during the process (Dennis, Wixom & Ross, 2012).

Project Management

In the second part of the planning phase, the project goes through a comprehensive project management process. The objectives of project management include balancing the optimal completion time-frame with minimizing cost and maximizing completion of planned outcomes. In order to achieve these objectives, staffing, project, and control plans are created.

Staffing the Project

Before any project work can be completed, the team must be assembled. The project manager uses the requirements of the project to determine what the requisite skills and positions are on the project team. As the team is put together, the project manager should review the staffing plan with the stakeholders to make sure the team has the capability to complete the project. Once the project is staffed, the project can be accurately planned.

Creating the Project Plan

As part of the project management section of the planning phase, the specific task and objectives must be outlined in a project plan. Often, the project manager creates the plan along with the team to follow a realistic timeline for the project progression. Deadlines are assigned, which allows for a project pace to be set. Many techniques for creating a project plan are used, like Gantt charting, which distinguishes between individual tasks, while still showing project milestones.

Controlling the Project

As the final part of the planning phase, the project is reviewed by the stakeholders to ensure that it can move forward. A plan for the management and control of the project is created for dealing with anything that might occur during project completion. When creating these plans, it is important to note the gradual refinement, often iterative nature of the SDLC, meaning that planning phase can always be revisited if there is insufficient information to continue.

2.4.2. Phase Two: Analysis

Secondary to the planning phase is the analysis phase. During the analysis phase, the questions of "*who* will use the system, *what* the system will do, and *where* and *when* it will be used" (Dennis, Wixom & Ross, 13) are addressed. In answering these questions, the analysis phase also serves as the main strategy and requirements gathering period. There is a deeper investigation conducted into the functional aspects of the system before any design is performed, which consists of an analysis strategy, requirements gathering, and the resultant system proposal.

Analysis Strategy

An effective analysis strategy will seek to understand the existing *as-is* system, identifying improvements, and define requirements for the new *to-be* system. The analysis strategy serves as the framework for how the team conducts itself through the analysis phase and how requirements will be elicited.

Requirements Gathering

During the planning phase, high-level business requirements for the system were identified. In the requirements gathering section of the analysis phase, those high-level business requirements are translated to statements of what the system must do or what characteristics it needs to have. These statements, which confirm the value to the business, can be classified as either functional or nonfunctional requirements. Requirements can be elicited in many ways, including interviews, surveys, or one-on-one meetings. Later, the requirements can be transformed into the project's data models and processes.

System Proposal

The synthesis of the analysis strategy and the requirements gathering is the system proposal. Much like the preceding system request and feasibility study, the system proposal formally captures the requirements of the system along with the business justification. The system proposal presents the case for moving forward with the project and can be either approved or reworked.

2.4.3. Phase Three: Design

The main objective of phase three is to design the proposed system. After planning and analyzing the project's needs, it is finally time to see how the system will work. Phase three is an important one, as it bridges the gap between the team’s businesses and utilizes the

team's knowledge from a technical standpoint. Careful consideration must be taken to ensure all aspects of the system are designed and in scope, otherwise the development process (phase four) may take longer than expected. The team will contact Communispace regularly to ensure that the client's needs are met, while also avoiding scope creep and staying on a timeline. The primary output will be the system specification, which is broken into five design sections: physical system, architecture, interface, programs, and database and files.

System Design

Designing the physical system will lay out the design strategy for the team so that they are able to conceptualize the foundation for the project. This is where the system specifications, as well as an alternative matrix of all physical options are shown.

Architecture design is at the core of ensuring that the project has the necessary physical and virtual elements to perform its duties. Hardware and software selection is also made during this step.

Interface Design

Designing the user interface (UI) is an important step for the client and the team to communicate the usability and layout of the program. If the software does not have a well-organized and consistent UI, then the users will be adversely affected. Navigation, data consistency, and satisfaction are all issues that can be avoided if designed correctly. The first step is developing use scenarios that cover a range of functions the customer may

want to use the system for. Based on these scenarios, the structure and standards of the program will be evaluated. Eventually a prototype will be developed and the client will evaluate that interface to confirm the system's UI.

Program Design

The software's logic will be based on designing the program. This involves data flow diagramming, a program structure chart, and specifications that allows the team to implement the system with a clear vision.

Database and File Design

Lastly, the project will involve databases with customer and vendor information. Selecting how to store the information, the amount of storage the database will need, as well as performance characteristics all need to be taken into account. The team will provide entity relationship modeling to structure the layout of the database, and establish the performance tuning and size estimations of the system.

2.4.4. Phase Four: Implementation

After the system is designed, it is finally time to focus on delivering and maintaining the completed system. The previous three phases have planned the project's methodology, addressed all customer objectives, and mapped the particular systems in a design. The primary output of this phase will be the final product: the functional system.

System Construction

The system will be constructed through programming, software testing, and performance testing. To ensure the product will be of quality, a plan for testing along with documentation of the software allows the user to test the system and expands the program if needed.

System Installation

After construction, installing and maintaining the system are important in the program's transition from the team to the client. Installing and training the user to become proficient at using the program is something that takes time. The larger the organization, the more stakeholders involved that will need to become acquainted with the system. Meetings and learn sessions should be scheduled in advance to allot the amount of time required for such an important process. Develop a support plan, maintain or make small revisions to the system if necessary. Lastly, a post-implementation audit will be conducted to determine the project's performance and success.

Maintenance

After the client has management of the system, the SDLC may not be finished. Helping the users to use the system and providing support is a part of what is called on-demand training. Online support such as *frequently asked questions* (FAQs) and documentation can provide the user with enough information to solve simple issues. If a more complicated

problem arises, most organizations have a help desk that can help provide hands-on support. Over time, the system may evolve and development will be needed. This process of refining the system can take anywhere from hours to months depending on the complexity of the system. Errors are the most common for system maintenance; however changes because of enhancements and organizational strategy can play a factor in maintaining a system (Dennis, Wixom & Ross, 2012).

2.5 Project Methodologies

While going through the Systems Development Life Cycle, there are many different approaches available to effectively manage the development process. There is no single best approach; each methodology has its own set of strengths and weaknesses. Choosing the right project management methodology is an important step towards creating a successful system. There are seven primary methodologies for system development: Waterfall, Parallel, V-model, Iterative, Agile, System Prototyping, and Throwaway Prototyping (Dennis, Wixom & Roth, 2012).

2.5.1. Waterfall

The waterfall method involves sequential progress through the phases of the SDLC. The project progresses from phase to phase in a highly regimented manner. At the conclusion of each phase, the key deliverables (usually including a large amount of documentation) are presented for approval. Upon completion of the controlled approval process, the project is ready to proceed to the next phase.

The waterfall method is a traditional system development method and waterfall projects are ideally large systems with very clear requirements. Much of the time and emphasis is spent on planning, so it is important that the requirements for the system do not change much throughout the project in order to avoid extensive rework. The following graph Figure 4 gives a visual representation of the waterfall method.

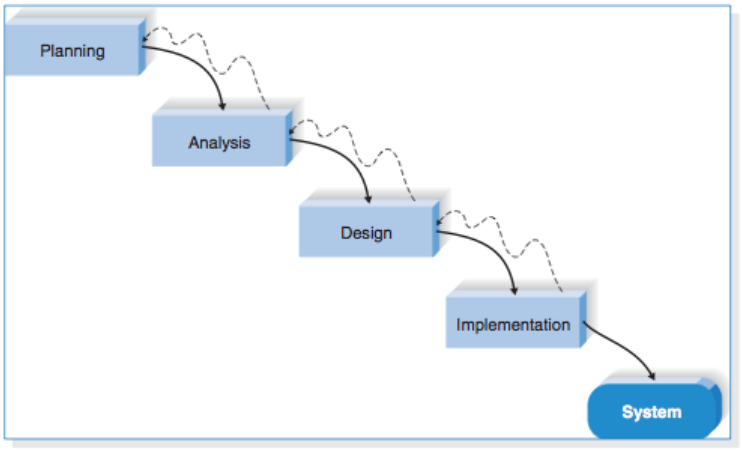


Figure 4: Waterfall Development (Dennis, Wixom & Roth, 2012, p.51)

2.5.2. Parallel

One variation of waterfall methodology is parallel development, which was developed to address one of the waterfall methodology's main drawbacks: its lengthy timeline. In parallel development, the project is broken down into several subprojects at the end of the analysis phase like shown in Figure 5. The design and implementation of the subprojects is then concurrently run to shorten the length of the overall project.

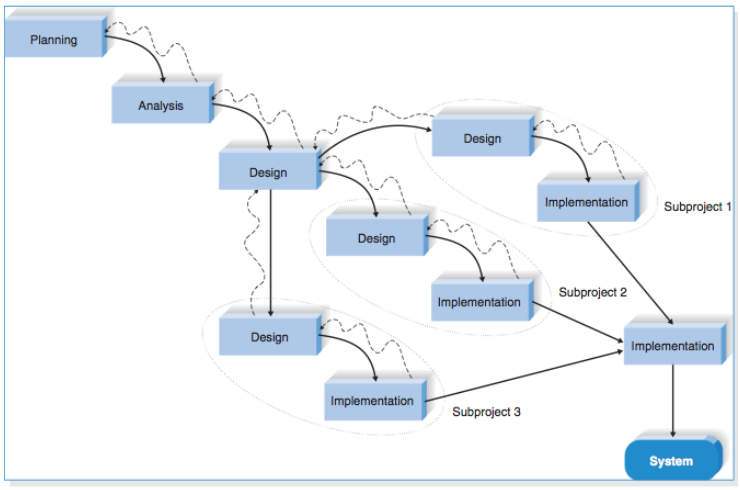


Figure 5: Parallel Development (Dennis, Wixom & Roth, 2012, p.52)

2.5.3. V-Model

Another variation on the waterfall methodology, the V-model places a large amount of emphasis on testing. The project proceeds through the analysis, design, and implementation phases, but also includes numerous additional phases shown in Figure 6: unit testing, integration testing, system testing, and acceptance testing. An important part of the initial analysis and design phases is the design of the various tests, helping to maximize the effectiveness of testing. V-model adds to the waterfall method, but really does not do anything to improve upon the problems that the waterfall method has.

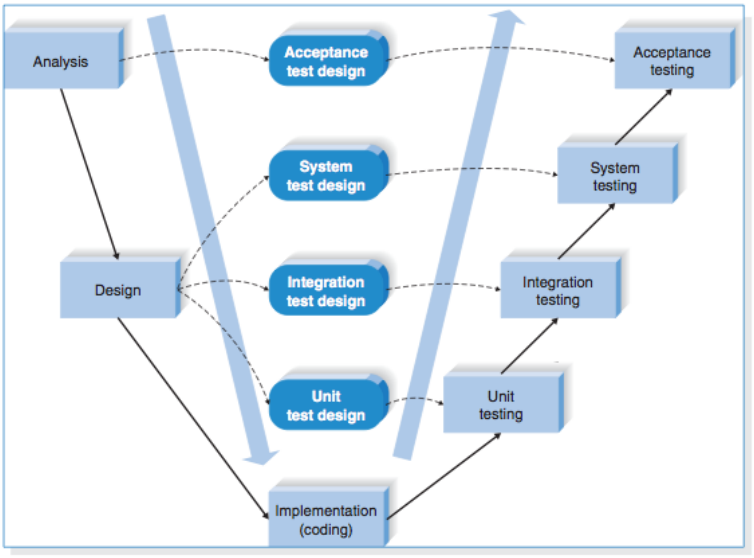


Figure 6: V-Model Development (Dennis, Wixom & Roth, 2012, p.53)

2.5.4. Iterative

Iterative development belongs to a group of methods called Rapid Application Development (RAD). It involves splitting the project up and creating versions visualized in Figure 7; initial versions have only the most important requirements. Then, once feedback is given, the next version of the system is developed. Iterative development allows the end user of the system to get involved with the incomplete project early on, which can have both positive and negative repercussions for the project.

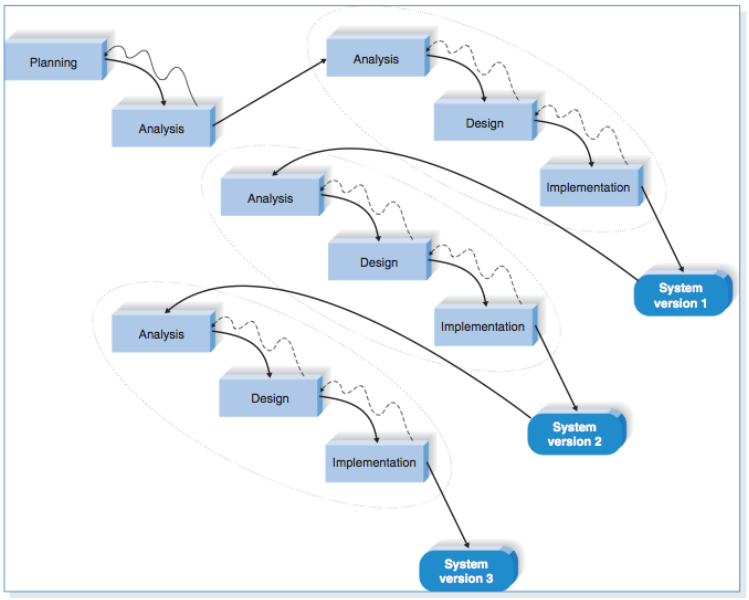


Figure 7: Iterative Development (Dennis, Wixom & Roth, 2012, p.55)

2.5.5. System Prototyping

In the system prototyping methodology, analysis, design, and implementation are all conducted together in order to create a simple prototype. The system prototype has limited functionality and is developed very quickly. After collecting feedback from users, a second prototype with more functionality is developed and the process is repeated. The following Figure 8 gives a clear visual representation of the system prototyping methodology. Eventually, the users, sponsors, and analysts must agree that the prototype includes all necessary functionality and then it can be installed. The system prototyping method works well when user requirements are ambiguous or unknown. The continual feedback on a series of prototypes helps to define the requirements of the project.

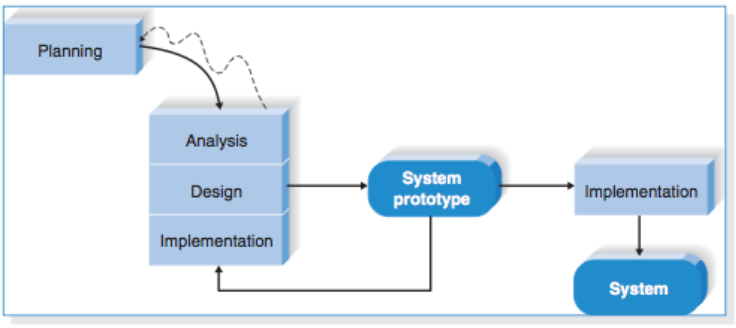


Figure 8: System Prototyping Development (Dennis, Wixom & Roth, 2012, p.55)

2.5.6. Throwaway Prototyping

Throwaway prototyping is another approach that involves prototypes. Figure 9 helps to better understand the throwaway prototyping process in a graphical way. With throwaway prototyping, there is a detailed analysis phase in order to gather ideas for the system. However, if some ideas are not well understood or pose a technical challenge, the team will create a design prototype, which is not a working system but is intended to assist with the problem. Unlike system prototyping, the design prototypes do not evolve into the final system; rather, they are thrown away once the problems are solved. Then, the project can move forward with its design and implementation phases.

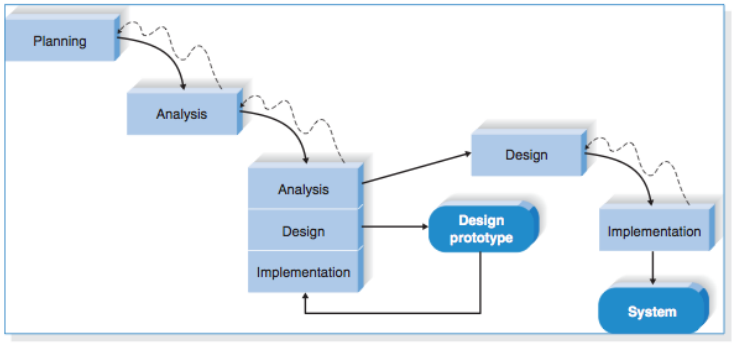


Figure 9: Throwaway Prototyping Development (Dennis, Wixom & Roth, 2012, p.56)

2.5.7. Agile

Agile development is a category of methodologies that are focused on face-to-face communication and programming. Agile projects are iterative and each version of the project is a complete software program just like shown in Figure 10. Iterations are short (two to four weeks) and involve all phases of the SDLC. Agile development requires a strong team that resolves issues mainly through face-to-face meetings rather than through extensive documentation. Agile is a category of development methodologies and includes several popular options including Scrum, extreme programming (XP), and dynamic systems development method (DSDM).

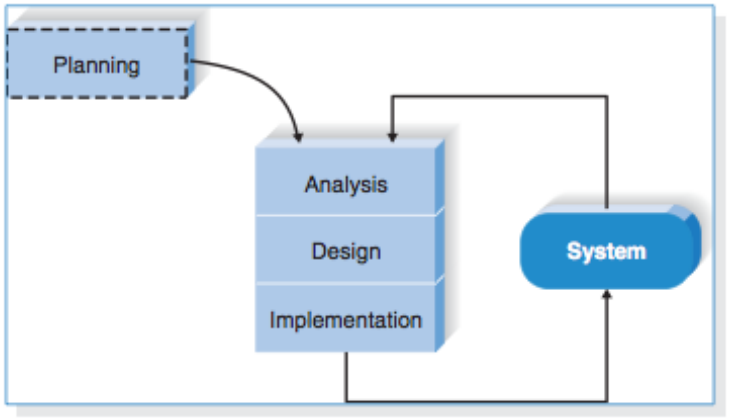


Figure 10: Agile (XP) Development (Dennis, Wixom & Roth, 2012, p.57)

2.6 Methodology Comparison

After researching the various systems development methodologies, the team conducted a brief, yet comprehensive analysis. The team put together a chart (Table 1), comparing the methodologies across multiple component usefulness levels. Each methodology was qualitatively ranked as poor, good, or excellent, and using these ranks the team was able to objectively choose which methodology to follow during the subsequent planning phase, detailed in Section 3.2.

The first usefulness component ranked was "with ambiguous requirements," meaning when requirements are ambiguous, how well does the methodology cope with the ambiguity. The waterfall, parallel, and v-model methods ranked as poor, while agile, system, and throwaway prototyping ranked excellent. Iterative ranked as good. The next component was "with unfamiliar technology," meaning how well the method works when

the team uses unfamiliar technology during development. Again, waterfall, parallel, and v-model methods ranked as poor as well as system prototyping and agile. Throwaway prototyping ranked as excellent and iterative scored good.

Next, the "with complexity" component, waterfall, parallel, v-model methods, and iterative ranked as good, systems prototyping and agile ranked as poor, and throwaway prototyping as excellent. With complexity means as the project complexity changes, how useful does the method become. Similarly, "with reliability" was assessed, systems prototyping scored poor, waterfall, parallel, iterative, and agile ranked as good, and throwaway prototyping was determined to be excellent. Lastly, "with short time schedule," meaning how well the method performs with a short time constraint, was analyzed. The waterfall and v-model were ranked as poor, while the parallel and throwaway prototyping methodologies scored as good. Iterative, agile, and systems prototyping all achieved a rank of excellent.

Table 1: Comparison Chart

Usefulness Component	Waterfall	Parallel	V-Model	Iterative	System Prototyping	Throwaway Prototyping	Agile
with ambiguous requirements	Poor	Poor	Poor	Good	Excellent	Excellent	Excellent
with unfamiliar technology	Poor	Poor	Poor	Good	Poor	Excellent	Poor
with complexity	Good	Good	Good	Good	Poor	Excellent	Poor
with reliability	Good	Good	Excellent	Good	Poor	Excellent	Good
with short time schedule	Poor	Good	Poor	Excellent	Excellent	Good	Excellent

3. Planning Phase

Through the synthesis of their academic knowledge and research into different aspects of the project, the team was able to gain a solid foundation for the rest of the project. The vision of Communispace, to build a sourcing member repository, has remained consistent from the beginning of the team's project work, while the form was originally rather ambiguous. Weekly meetings at Communispace with key stakeholders as well as at WPI with Professor Loiacono were beneficial to planning the project.

In this section, the team focuses on detailing the structure of how the project will be completed. The planning phase includes the scope and objectives, methodology, project plan, feasibility and risk assessment, and proposal conclusion. Also included is a documentation of the stakeholders and systems request. Through this section, the team articulates the requirements and expectations that comprise their project planning and proposal.

The planning phase is important in discovering the inherent need of a project, the feasibility of the venture, and determining the resources needed to execute the plan. After all necessary research was performed, the team developed a proposal to plan out how to accomplish the project.

3.1 Scope and Objectives

To maintain focus during the project, it is important to define the scope and objectives. The scope of this project is to deliver a system that maximizes the efficiency of the community recruitment process. Centralizing prospective member data from past communities and overflow, and shifting from event-based to ongoing recruitment across audiences was identified within the scope of the project. The objective is integrating these three scenarios into a system in a timely manner.

3.2 Proposed Methodology

Based on their initial understanding of the project, the team decided to form smaller, agile groups , sometimes working as individuals, in order for them to cover a large number of focus areas. The parallel operational strategy allowed them to use the five-person project team to flexibly work within the three seven-week terms. Often, segments of the group focused on different parts of the project, while sometimes the group worked together. Group assignments were divided based on how much work needed to be completed and were split up in terms of functionality. Zeng Liu focused on the system's search, while Amanda Weis focused on adding members to the system, through both manual and import processes. Thomas Meagher worked on the manage members functionality, Jiedong Wang worked on the reports, and William Richtmyer worked on database setup and standardization.

With this operational strategy, it was important for them to hold team meetings to update each other so everyone had knowledge of the segmented work. The different groups were determined throughout the project, depending on the amount of expected work and time constraints. The advantage to working on individual segments in small groups is that the team followed an agile-like development process, where each segment was non-sequential and iterative. This allowed the team to constantly develop their approach to the shifting project objectives. When the team needed to spend more time refining a particular project segment, they were able to competently iterate upon it.

The group conducted their analysis, design, and implementation together, using the system prototyping method, which works well when requirements are ambiguous or unknown. Each prototype version was developed quickly so stakeholder feedback could be elicited and integrated. With only one day per week to meet with the stakeholders, the continual feedback for each prototype helped to define the requirements of the project. Prototypes were developed until there was an agreement among the stakeholders that all the requirements were captured.

3.3 System Development Scenarios

During the planning phase, the team developed three possible scenarios shown in Figure 11 in the following page in which the project could be completed. The first scenario is *Recommendations*, in which the team spends a large portion of their time providing a detailed in-depth analysis and requirements gathering. With all of the detailed data

gathered, the team provides Communispace a robust recommendation of a plan of action, and what areas to pursue. The second scenario is the *Design* scenario. This is where the current state of the "system" is analyzed and requirements are gathered. With the research performed, the system is designed by the team and recommendations are made how to build the system. Lastly, *Prototyping* provides the third scenario for system development. After analysis and requirements are gathered, the system is designed and a prototype made. With the prototype produced, documentation for the program, training for employees, and a transition plan for future improvements are produced. The team chose the *prototyping* system development scenario as it provided the most value to Communispace and the MQP team.

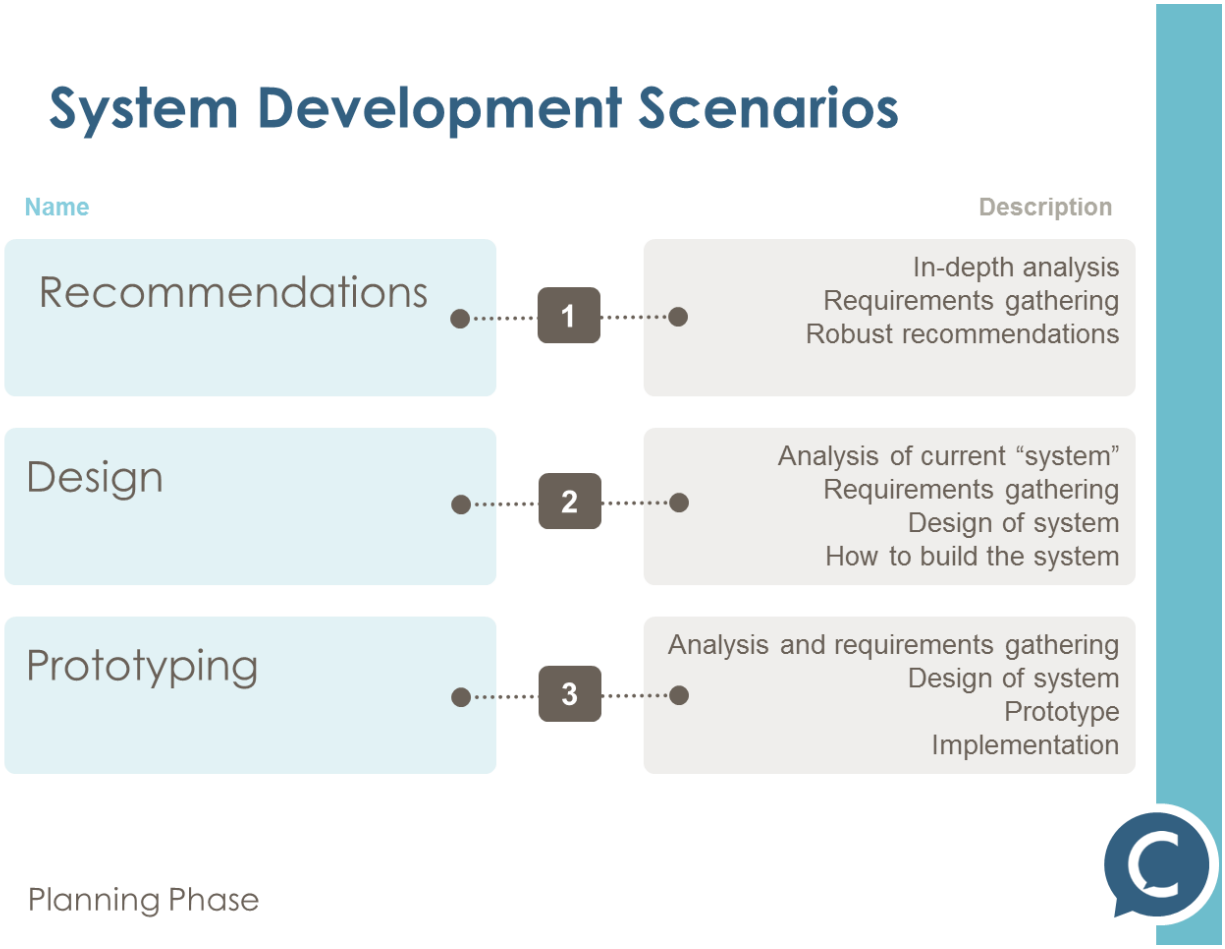


Figure 11: System Development Scenarios

3.4 Project Plan

The project is scoped to be finished in three WPI academic terms, in equivalence of eighteen weeks (not counting three weeks of project preparation at the beginning of A term). With the SDLC approach, there are four phases within the project methodology: the Planning Phase, lasting around four weeks; Analysis Phase, around three weeks; Design Phase, around three weeks; and Implementation Phase, around seven weeks. A complete view of the project Gantt Chart can be found in Appendix G. The entire process will give the

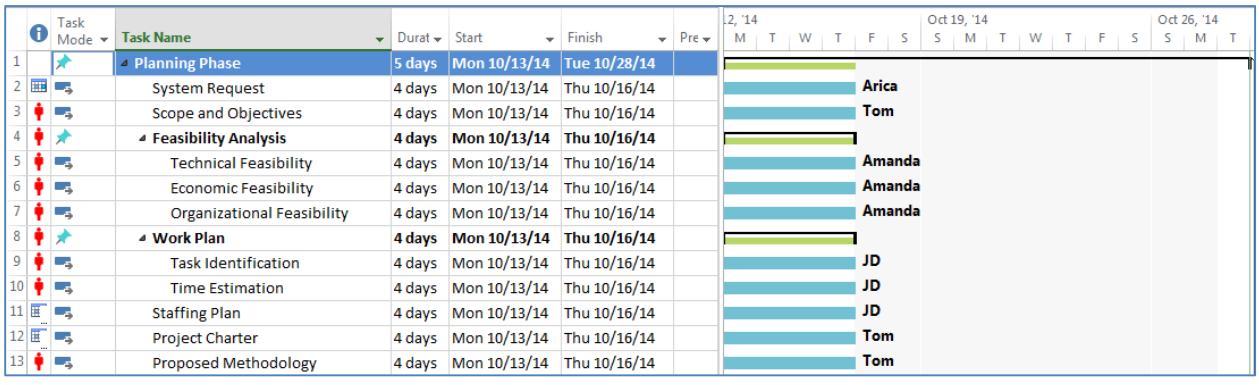
team a comprehensive understanding of the project in order to provide better project deliverables.

3.4.1. Task Identification & Time Estimation

The distribution of tasks with the associated time estimation is shown in the Gantt chart as follows divided by the four phases of the project. For each of the four project phases, there are a number of tasks associated in completing the phase. It is important to note that the estimation of task and time is tentative because of the iterative project methodology—if more time is needed, then the schedule will adjust accordingly.

Planning Phase

The planning phase shown in Figure 12 consists of system request, scope and objectives, feasibility analysis, work plan, staffing plan, and proposed methodology. The tasks of the planning phase are evenly distributed and took about three weeks to complete.



	Task Mode	Task Name	Durat	Start	Finish	Pre
1		**Planning Phase**	**5 days**	**Mon 10/13/14**	**Tue 10/28/14**	
2		System Request	4 days	Mon 10/13/14	Thu 10/16/14	
3		Scope and Objectives	4 days	Mon 10/13/14	Thu 10/16/14	
4		**Feasibility Analysis**	**4 days**	**Mon 10/13/14**	**Thu 10/16/14**	
5		Technical Feasibility	4 days	Mon 10/13/14	Thu 10/16/14	
6		Economic Feasibility	4 days	Mon 10/13/14	Thu 10/16/14	
7		Organizational Feasibility	4 days	Mon 10/13/14	Thu 10/16/14	
8		**Work Plan**	**4 days**	**Mon 10/13/14**	**Thu 10/16/14**	
9		Task Identification	4 days	Mon 10/13/14	Thu 10/16/14	
10		Time Estimation	4 days	Mon 10/13/14	Thu 10/16/14	
11		Staffing Plan	4 days	Mon 10/13/14	Thu 10/16/14	
12		Project Charter	4 days	Mon 10/13/14	Thu 10/16/14	
13		Proposed Methodology	4 days	Mon 10/13/14	Thu 10/16/14	

Figure 12: Project Plan- Planning Phase

Analysis Phase

The analysis phase shown in Figure 13 consists of analysis plan, requirements gathering, process model, data model, project methodology. During the analysis phase, the team will present the system proposal for review by Communispace to ensure the project is on the right track.

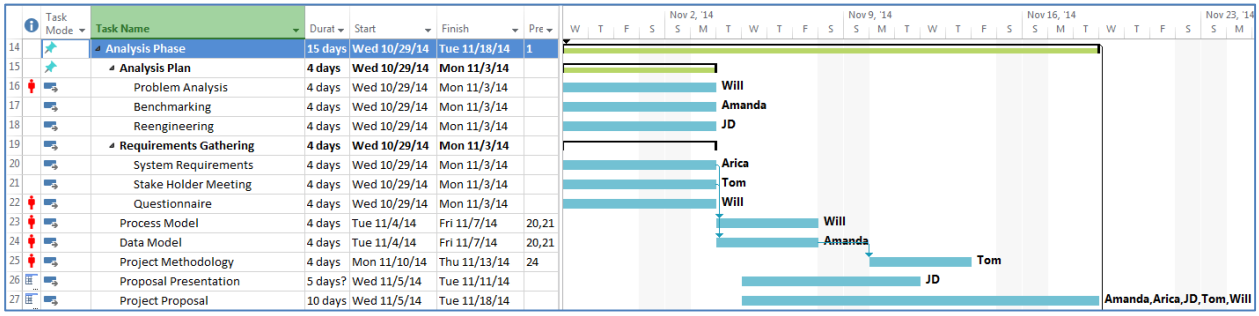


Figure 13: Project Plan - Analysis Phase

Design Phase

The design phase of the project includes physical, architectural, interface, database, and program design. Tasks identification and time estimate for design phase are shown in Figure 14. During each of the design phase processes, there will be interviews/meetings in order to receive timely feedback from project sponsors to better fulfill the project requirements.

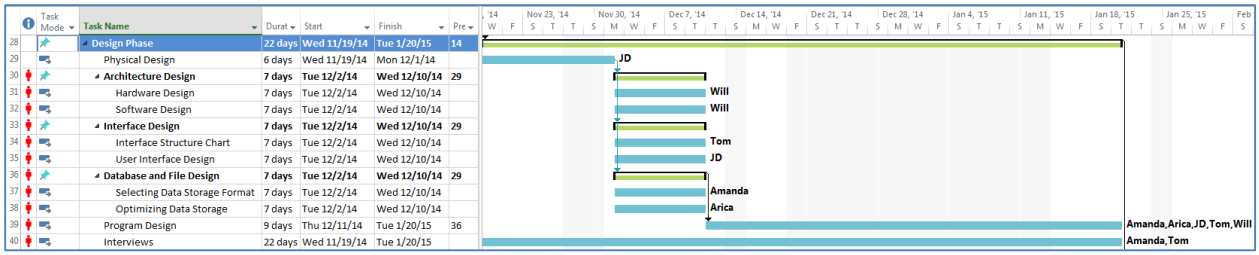


Figure 14: Project Plan- Design Phase

Implementation Phase

The implementation phase took about seven weeks with system construction, test plan, system testing, system documentation, and system refinement visualized in Figure 15. The project presentation would take place on February 25th, 2015. The implementation phase would be run iteratively, with continual improvements to the working prototype.

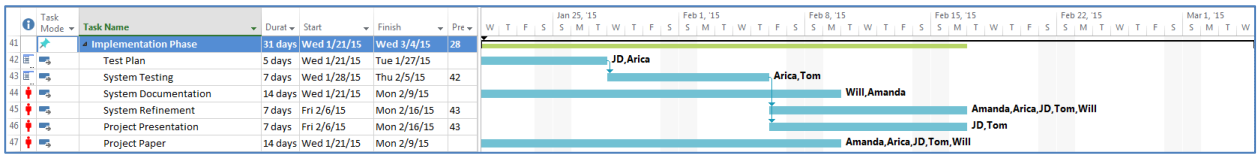


Figure 15: Project Plan- Implementation Phase

3.4.2. Staffing Plan

The team consists of five seniors completing their Bachelors of Science in Management Information Systems at WPI. The project work is evenly distributed across the team based on each member's expertise while still maintaining a level of common work for a better learning experience through real-life practical collaboration.

The project requires knowledge of many areas covered in the WPI MIS curriculum including database design & management, user interface and user experience design, computer programming, human computer interaction (HCI), marketing, accounting,

operations management, leadership, and most importantly, systems analysis & design. The team applied the knowledge learned from WPI experiences to create a successful project for Communispace. The final deliverable was a relational database similar to those that they encountered in their coursework at WPI. In order to build the tool, the team utilized skills from various programming courses as well as user interface and HCI principles that were explored in MIS courses. The marketing courses helped the team to understand the nature of Communispace's business while their accounting coursework assisted them in the creation of the economic feasibility calculations. Operations management principles aided the creation of a streamlined tool to help the Communispace team recruit more efficiently and effectively. Most importantly, the systems analysis and design coursework provided the team with the basis of understanding how to build a valuable system and the ability to successfully plan, analyze, design, and implement the system at Communispace. Detailed individual descriptions of the team members can be found in Appendix C.

3.5 Project Stakeholders

There are three groups of stakeholders that benefit from the member repository: senior leadership, those affected by the member repository, and those working with the member repository on a day-to-day basis. The *Client and Consumer Services* division is where most of the stakeholders reside. Ms. Laura Naylor is the Senior Vice President of *Consumer Relations*, and plays the role of decision-maker. Reporting to her is another decision-maker, Mr. Dave Rosenberg. Mr. Rosenberg is the Director of the *Consumer Acquisition* group,

responsible for two subdivisions. These two subdivisions are *Data Analytic*s and *Digital Marketing*.

Data Analytics works with the member repository on a day-to-day basis, identifying those that may benefit from members already stored in the system. There are three relevant members of the *data analytics* group: Mr. Jack Bergersen, Mr. Sean Burke, and Ms. Michelle Fisher. Mr. Bergersen and Mr. Burke had a day-to-day communication with the team in developing the project's vision. Ms. Fisher was vital in gathering requirements for the team, and the team met regularly with her.

Digital Marketing is a group that will benefit from the member repository. When gathering members for recruitment or performing community maintenance, the *Digital Marketing* team will benefit from sourcing these members from an existing database. Though not using the member repository on a daily basis, it has the potential to save time and money, something that the project hoped to provide. Two members who are a part of this team are Mr. Justin Hill and Ms. Amanda Hartie. The team met regularly with Mr. Hill to gather a perspective from the *digital marketing* team. The group met with Ms. Amanda Hartie a few times to get an example of use-cases, as well as her valuable opinion on the project.

The last stakeholder who will be important in the future is Mr. Slava Asipenko, Director of *Engineering and Development* for Catalyst. While he may not be interacting with the member repository, he will certainly have an impact of potentially integrating the program

with Communispace's proprietary program called Catalyst. In an attempt to be forward looking, the team kept Mr. Asipenko up-to-date on their project's progress and sought his insight.

3.6 Systems Request

This project has been initiated to optimize the company's process to source and re-use members for the communities. Using the proposed system, users will be able to maintain a database that integrates both members from past communities and potential members captured from overflow, as well as searching for usable members through the database. The team expects the system to enable more efficient recruitment and to improve profit margins by decreasing recruiting expenses. A detailed system request overviewing the business need, requirements, value, and special issues and constraints can be found in Appendix A.

3.7 Feasibility Study

This feasibility study is meant to justify the development of the project by observing in detail three different segments. Through discussions with Mr. Jack Bergersen and Mr. Sean Burke, the team has discussed the viability of the project technically, organizationally, and economically. Technical feasibility explores the technological obstacles that the team faces. Economic feasibility provides estimates about the cost and economic benefit that they system will provide. Finally, organizational feasibility outlines the support of the

organization and any problems that may arise organizationally with the implementation of the system. An overview of the feasibility study can be found in Appendix D.

3.7.1. Technical Feasibility

Technical feasibility is an analysis of the ability to create a system and the risks associated with the building of the system. The technical feasibility of the project is limited by two factors: the technical expertise of the team to build the system and the technical abilities of the Communispace team who will be using the final product.

The team has varying levels of technical ability; some members of the team have more experience with programming through internships and WPI coursework, while others have fewer skills in this area. Because of the tight timeframe of the project, the team chose to consider options with which the team is more familiar. The team is relatively comfortable with programs such as Microsoft Access, Microsoft Excel, and SQL databases, so the team focused on these when considering potential options for building the system.

The group also needed to consider the ability for the Client and Consumer Services team to utilize the final product. The levels of technical knowledge within the group vary as well, so it was important for them to consider that the system must be usable and easy to learn in order for it to provide the most value to the Communispace group. Members of the team are fairly comfortable with programs like Microsoft Access but are not necessarily able to

write SQL queries without the aid of an interface. This was another point which the group took into consideration when selecting options for the member repository.

3.7.2. Organizational Feasibility

Organization feasibility is an analysis of how well the proposed system will work within the operations of Communispace and how well it is supported by the stakeholders. The proposed system has the potential to offer high value to its users. The member repository will be a useful option for recruiting for future communities. The new repository will increase the efficiency of the recruiting process and the productivity of the recruiting team at Communispace. From the discussion with members of the Client and Consumer Services team, the member repository is a well-supported idea. Because there is no current system for utilizing past or overflow recruits, there will be an adjustment period in which people get used to the idea of using the repository as a potential recruitment source. There will also be a learning curve associated with learning how to best utilize the new system.

External to the Client and Consumer Services team, the team also explored the organizational feasibility from the perspective of the internal Technology team. The Technology team currently utilizes their proprietary software platform, Catalyst, for managing client communities and there was an opportunity for the team to integrate a solution with it. Looking further into integration with Catalyst, the group met with Mr. Slava Asipenko, Director of Engineering and Development, to understand the feasibility of the option and discuss their project. While the insight from Mr. Asipenko was relevant,

integrating the resultant system with Catalyst is not in scope because of time, organizational, and technical constraints. The Catalyst team does not have the time to train the team nor support the development as well as they do not have the technical skills to implement a Catalyst-based solution.

Overall, the project has two project sponsors who strongly support the project. Ms. Laura Naylor, the Senior Vice President of Member Experience and Operations and Mr. Dave Rosenberg, the Vice President of Client and Consumer Services, have both provided the team with guidelines and objectives in order to make this project a success for them Sand for Communispace.

3.7.3. Economic Feasibility

Economic feasibility is an analysis of the project's ability to be completed within budget constraints, as well as an analysis of the economic benefit the system will provide.

During the team's interviews with various stakeholders at Communispace, the Client and Consumer Services team expressed the possibility for a member repository system to provide several tangible benefits to their team. The main economic benefit of the system will be a reduced expense for recruiting. Because the Communispace team will have the option of recruiting members for communities from within the member repository, there will be less of a need to spend money on other forms of recruiting like website ads and the purchase of email lists.

The analysis of the economic feasibility of the project is both qualitative and quantitative. The team has a strong understanding of the benefits that the system could provide to the team in terms of cost reductions. The team found that the cost of hiring the WPI team for 21 weeks of work at the cost of [THIS SECTION HAS BEEN REDACTED] is significantly less than the average recruiting budget for a community. According to Mr. Jack Bergersen, a single community will usually spend between [THIS SECTION HAS BEEN REDACTED] to recruit members, though some have spent over [THIS SECTION HAS BEEN REDACTED] in the past to recruit members. From this, it is clear to see that if the member sourcing system is used to recruit people to even a few communities, it will provide a very high return on investment.

The team first conducted an analysis to compare three scenarios: utilizing the WPI project team, using a current Communispace employee, and using an outside consultant. The details of this analysis were obtained through information from Mr. Jack Bergersen and independent research into employee and consultant compensation. The analysis is outlined in Table 1, and clearly shows that the WPI project team is the most economically sensible direction for the project.

Table 2: Economic Comparison of Development options

[THIS SECTION HAS BEEN REDACTED]

Based on data from 2014, the team was then able to calculate the total recruiting budget for one year. This includes both recruiting for new communities and for "refreshing" a community with new members while it is still operating. The total amount spent on recruitment in 2014 was approximately [THIS SECTION HAS BEEN REDACTED].

The team conducted a complete cost-benefit analysis based on the WPI team development scenario outlined above. In all cases, the system was assumed to only return half of the calculated yearly benefits during 2015. This assumption was made because the system will not be totally complete until mid-way through the year and will take some time before it begins returning benefits to the team. The cost-benefit analysis was split into three separate analyses: the low estimate is shown in Table 2 and assumes that the PMSS will be used to recruit five percent of community members for ten percent of communities. In this instance, the return on investment (ROI) over three years is 300% with a break-even point occurring in 2.88 years.

Table 3: Cost-benefit analysis for low estimate

[THIS SECTION HAS BEEN REDACTED]

The medium estimate is shown in Table 3 and assumes that the PMSS will be used to recruit nine percent of community members for ten percent of communities. In this instance, the return on investment (ROI) over three years is 620% with a break-even point occurring in 1.43 years.

Table 4: Cost-benefit analysis for medium estimate

[THIS SECTION HAS BEEN REDACTED]

The high estimate is shown in Table 4 and assumes that the PMSS will be used to recruit 13 percent of community members for ten percent of communities. In this instance, the return on investment (ROI) over three years is 940% with a break-even point occurring in 1.24 years.

Table 5: Cost-benefit analysis for high estimate

[THIS SECTION HAS BEEN REDACTED]

After speaking with Mr. Jack Bergersen, the team understood that the system would definitely provide economic benefits to the Client and Consumer Services Team, most likely saving somewhere between $24,000 and $43,000 over three years. A summary of the cost-benefit analysis can be found in Table 5.

Table 6: Cost-Benefit Analysis

[THIS SECTION HAS BEEN REDACTED]

3.8 Risk Assessment

Risks associated with the project were identified and analyzed in this section with details including what the risk was, how the risk was going to impact the project, and what actions the team would take to mitigate the risk. Following sections are divided into Risk Identifications and corresponding risk mitigations.

3.8.1. Risk Identifications

It is important for the team to identify potential risks so that everyone is aware of what may come across moving forward with the project. Identifying the risks also helps the team to foresee coming challenges to be prepared thus overcome the risks. The risks identified by the team include the following three:

Project Ambiguity

The first risk is project ambiguity. Based on the fact that system should fulfill business and technical needs from various stakeholders, the project requirements gathering process will take time for the team to understand and digest on an organizational level and a tactical level in order to provide best project deliverables to Communispace.

Scheduling

Scheduling would be challenging as well since the team being consisted of 5 full-time college seniors and the project having various stakeholders involved at Communispace from different teams and disciplines. Additionally, given the fact that the team can only

commute to Communispace on Wednesdays, scheduling can be a challenging issue when it comes to large group meetings involving various stakeholders especially across different disciplines.

First attempt for problem solution

This is the first time the issue of data centralization and reuse was brought up to find a solution. The fact that whether or not the system prototype created by the team would solve the problem is a risk itself. Furthermore, how well the prototype would help solving the problem is a risk, too.

3.8.2. Risk Mitigation

After identifying potential risks, the team came up with brief risk mitigation plans corresponding to each of the three risks identified in previous section. These specific mitigation plans would allow the team to minimize the impacts of the potential risks on the project.

Project Ambiguity

The team would take notes and use critical thinking to best understand the requirements by various stakeholders from both technical and business perspectives. Additionally, the team would distribute project work reasonably to have each of the five members taking

lead on different matters and requirements to maximize the values of all five members so that the project can consistently stay on a high level of clarity.

Scheduling

The project phases are not strictly taken place one by one, in other words, the four phases could sometimes overlap providing a level of flexibility to the project. The project plan was scoped to be finished one week prior to last Wednesday, March 4, 2015, that C term ends to get a one-week room for any delay and administrative work that may occur.

First attempt for problem solution

The team would work closely with two primary contacts: Mr. Sean Burke and Mr. Jack Bergersen to constantly being able to understand and evaluate the project progress. In addition, the team would meet three times per week to update each other with any progress or thinking on the project. Having the ability of working closely dedicated to the project provides great focus and high efficiency in building the solution.

4. Analysis Phase

After the planning phase was complete and the team established a methodology for completing the project, the team moved on to the analysis phase. The team performed a substantial amount of analysis, through stakeholder interviews and other information gathering and continued to refine the system requirements throughout the analysis phase. As a whole, the analysis phase encompassed an analysis strategy, requirements definition, uses cases, system proposal, and process and data models.

4.1 Analysis Strategy

The analysis strategy of the project enhanced the team's understanding of requirements gathering, possible use cases, system data and process flows. Clarifying the validity and effectiveness of the analysis strategy through interviews with different stakeholders also gave the team a more comprehensive view and deeper insight into how the system should function and how it could bring a more valuable use to fulfilling the business needs.

Analogous to the project methodology, the team also used a methodical analysis strategy. The team attempted to standardize the format for conducting the analyses that examined different responses from varying organizational and technical perspectives. Even though the analysis strategy followed a broadly rigid structure, the team was flexible whenever it was advantageous to the project.

4.2 Requirements Definition

After several meetings with Mr. Jack Bergersen, Mr. Sean Burke, and other major stakeholders, the following requirements were identified to outline the various objectives of the project.

4.2.1. Business Requirement

The overarching business requirement for the project was to enable a centralized member repository to optimize the company's recruiting process, thus saving valuable company resources (Meeting notes 10/01/2014).

4.2.2. User Requirements

Based on the team's meeting with stakeholders, it was determined that the users of the proposed system should be able to:

- Add people directly from "overflow" as well as people who are not qualified for a screener survey; (Meeting notes 10/01/2014)
- Import alumni members from closed communities; (Meeting notes 10/01/2014)
- Look up people who might be qualified to participate in a community, given certain user-defined profile fields; (Meeting notes 10/01/2014)
- Maintain a blacklist of low-performance members; (Meeting notes 09/24/2014 and 10/01/2014)
- Delete members from the repository. (Meeting notes 09/24/2014 and 10/01/2014)

4.2.3. Process Oriented

Process-oriented functionalities to support the user requirements were also considered by the team, including:

- Display a list of members given one or several search criteria. (Meeting notes 10/01/2014 and 10/29/14)

4.2.4. Information Oriented

The team also considered information-oriented requirements of the system in order to support the aforementioned processes, including:

- Handle data normalization; (Meeting notes 09/24/2014 and 10/01/2014)
- Batch check qualifications of potential members as well as the recruiting limit of the community; (Meeting notes 10/01/2014 and 10/08/2014)
- Export data into users' desired formats. (Meeting notes 10/01/2014)

4.2.5. Non-Functional

Non-functional requirements describe the characteristics that the system should have, broken down into four sub-categories. Operational requirements describe in which physical and technical environments the system operates. Performance requirements give the expected speed, capacity, and reliability of the system. Security requirements define which users have authorized access to the system under what circumstances. Finally, cultural/political requirements are cultural, political, and legal factors that affect the system (Dennis, Wixom & Roth, 2012).

Operational

- The database maximizes the efficiency of data normalization; (Meeting notes 10/01/2014)
- The system limits the number of times that a member can be exported; (Meeting notes 10/01/2014)
- The system facilitates a standardized naming convention for survey fields; (Meeting notes 10/08/2014)
- The system runs on Windows 7.

Performance

- The database queries are optimized to allow for fast searches;
- The repository is viewable and editable by multiple users at the same time, while handling potential scheduling issues.

Security

- The members' profile data is secured;
- Only certain users are allowed to access the repository.

Cultural and Political

- The behaviors of the system do not conflict with existing processes or working norms.

4.3 Profile Fields Questionnaire

In order to better understand the importance of various profile fields, the team created a questionnaire, which was sent to all members of the Digital Marketing and Data Analytics teams. The questionnaire was conducted through Google Forms and distributed to the employees in an email. The central question on the survey asked respondents to select the profile fields that they felt were most important to include in a general screener survey, limiting the selection to six fields. The team provided 12 fields to choose from: Country of Residence, Age, Male/Female, Education Level, Employment Status, Children (yes/no), Income, Ethnicity, Marital Status, Fluent Languages, Job Industry, and Other. The responses that the team gathered were meant to help the team prioritize common fields for the system and to aid in the creation of a general screener survey.

The survey received a 43% response rate, with nine responses from the 21 emails distributed. A high level overview of the survey results are shown in Figure 16. There were

several interesting results to note. Of the responses, there were two profile fields that were unanimously selected: age and gender. Other fields with a high number of responses were ethnicity (seven selected) and country of residence (five selected). Three fields received no votes from respondents: education level, fluent languages, and job industry. Finally, two respondents chose "other" and listed client specific behaviors or traits. The full results of this question are shown in Tabel 5.

Table 7: Summary of Results of Profile Fields Questionnaire

[THIS SECTION HAS BEEN REDACTED]

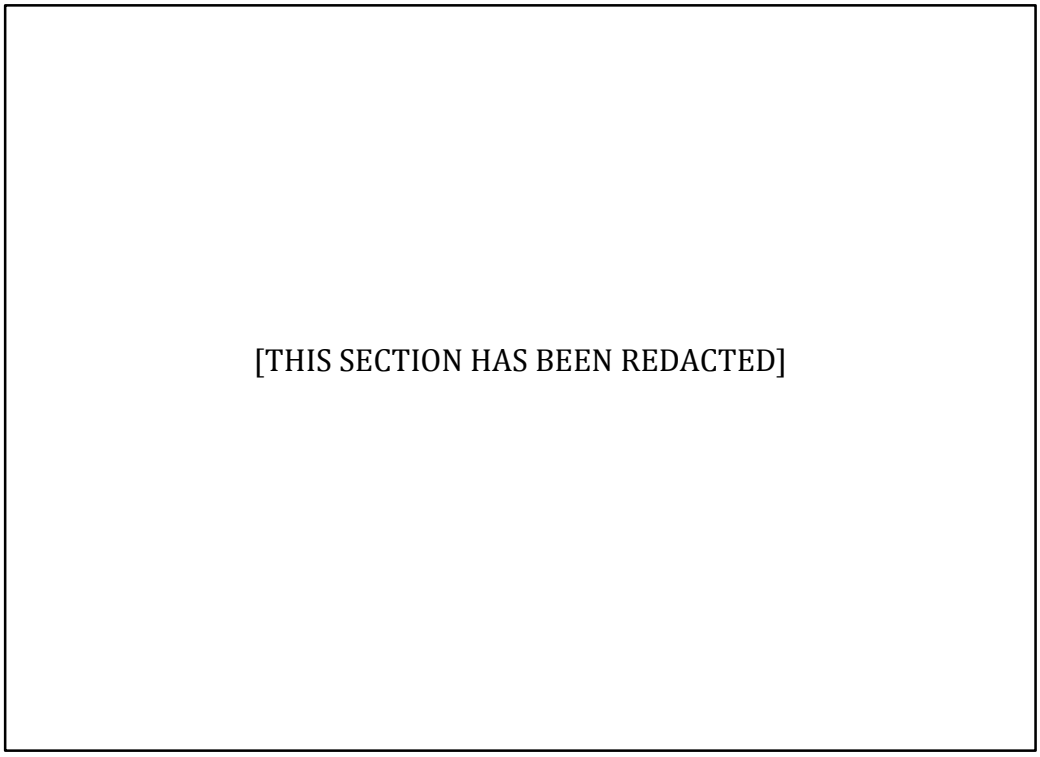


Figure 16: Profile Fields Questionnaire Results

The results from the survey gave the team a good understanding of which fields are the most important to include in the system. A full copy of the questionnaire and its results can be found in Appendix E.

4.4 Use Cases

[THIS SECTION HAS BEEN REDACTED]

4.5 Process and Data Models

Over several weeks of interviews with key stakeholders, the team identified five basic steps in the community sourcing process. The overall process, as well as the detailed information about each of these steps, is outlined in the following section.

[THIS SECTION HAS BEEN REDACTED]

4.6 Requirements Analysis and Organization

In order for the team to formalize the system requirements, the team utilized a graphical method to segment and organize the system requirements. The team created multiple diagrams using a whiteboard and sticky notes to represent the as-is and to-be systems. Each sticky note is meant to represent a different requirement of the system. In addition, the team placed the requirements along a user interaction timeline, where they could align with the chronological structure of the previously developed process flow diagrams.

4.6.1. As-Is System

In the as-is system diagram in Figure 23, the team captured 'Excel Dump File.' Since Communispace had no formal system in place for capturing potential members, the Excel files that the Client and Consumer Services team used were the closest thing to an existing system. For the purposes of this exercise, the Excel files served as the as-is system, but it is important to note that there was a high degree of variability among team member files.

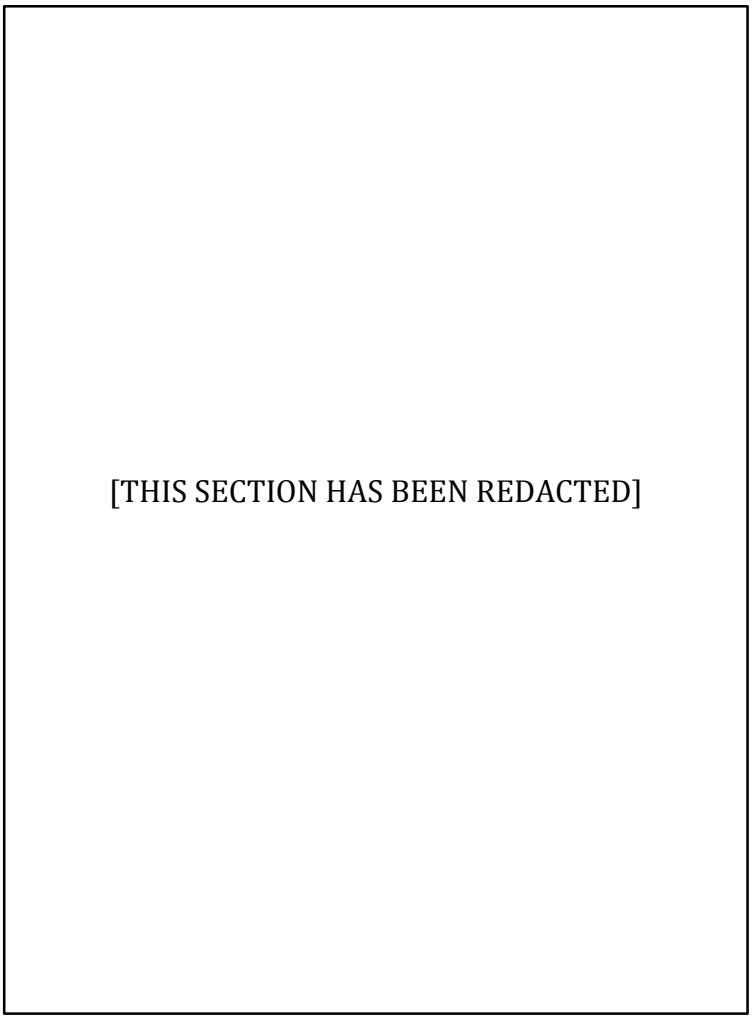


Figure 17: As-Is System

4.6.2. To-Be System

The team captured the requirements gathered during stakeholder interviews and research. The to-be system contains two sections, members before, on the left, and after entering the system, on the right.

On the left of Figure 24, where members have yet to enter the repository, the team identified the following requirements: survey (15 questions max), dupe check (quality),

normalize data (profile fields), track sources (PCID), small business and pharmaceutical tags, check for duplicates, and fields for erroneous data. The survey (15 questions max) notes captures the requirement that the member screener survey should be as short as possible, with 15 questions being the current number the recruiting team aims for. The dupe check (quality) note meets the requirement for prospective members to be checked against the dupe check list Communispace maintains in order to block scammers from entering the system. Next, the normalize data (profile fields) note deals with the requirement to standardize data in the repository, capturing relevant profile fields from the many different client communities. Checking for duplicates is a requirement to make sure the same user is not in the member repository multiple times. Finally, fields for erroneous data captures the requirement to store member data so it is accessible and searchable, while in an unstructured format.

On the right of Figure 24, the after entering part of the to-be system captures two requirements, the ability to track people in use currently and keep in contact with members. The ability to track people currently in use meets the requirement for members' status to be tracked in the system so that they are not used more than once simultaneously. Next, the requirement to keep in contact with members is for Communispace to know which members to reconnect with so their data does not become stale.

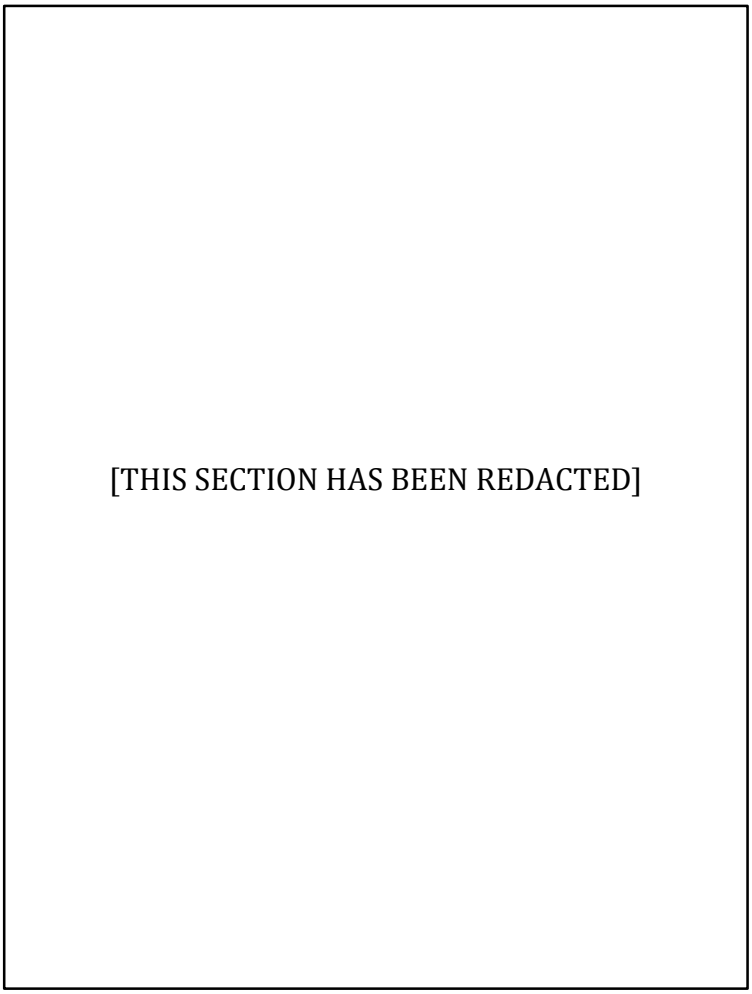


Figure 18: To-Be System

4.6.3. System-Wide Requirements

In the requirements diagram of the entire system view (Figure 25), the team took the requirement stickies and abstracted them to fit along the previously defined process flow diagram for community development. The five different sections, survey, entry, in database, exiting info, and overall are further described in this section. It is important to note the different color requirement sticky notes, yellow and blue. Yellow notes represent system requirements directly gleaned from the requirements gathering the team conducted, while

blue notes represent requirements that the team defined beyond those standard requirements.

[THIS SECTION HAS BEEN REDACTED]

Figure 19: System-Wide Requirements

4.6.4. Survey and Entry Requirements

Zooming in on the survey and entry sections, the team selected seven yellow and one blue requirement sticky notes shown in Figure 26. In the survey section, notes were placed if they were relevant to the screener or general surveying process that all community members must go through. The survey section notes were: survey (15 questions max),

normalize data (profile fields), small business and pharmaceutical tags, check for duplicates, and fields for erroneous data. In the entry section, the team placed: dupe check (quality), track sources (PCID), and check for duplicates. The team also defined an import function so that the data can be input into the system from other locations and formats.

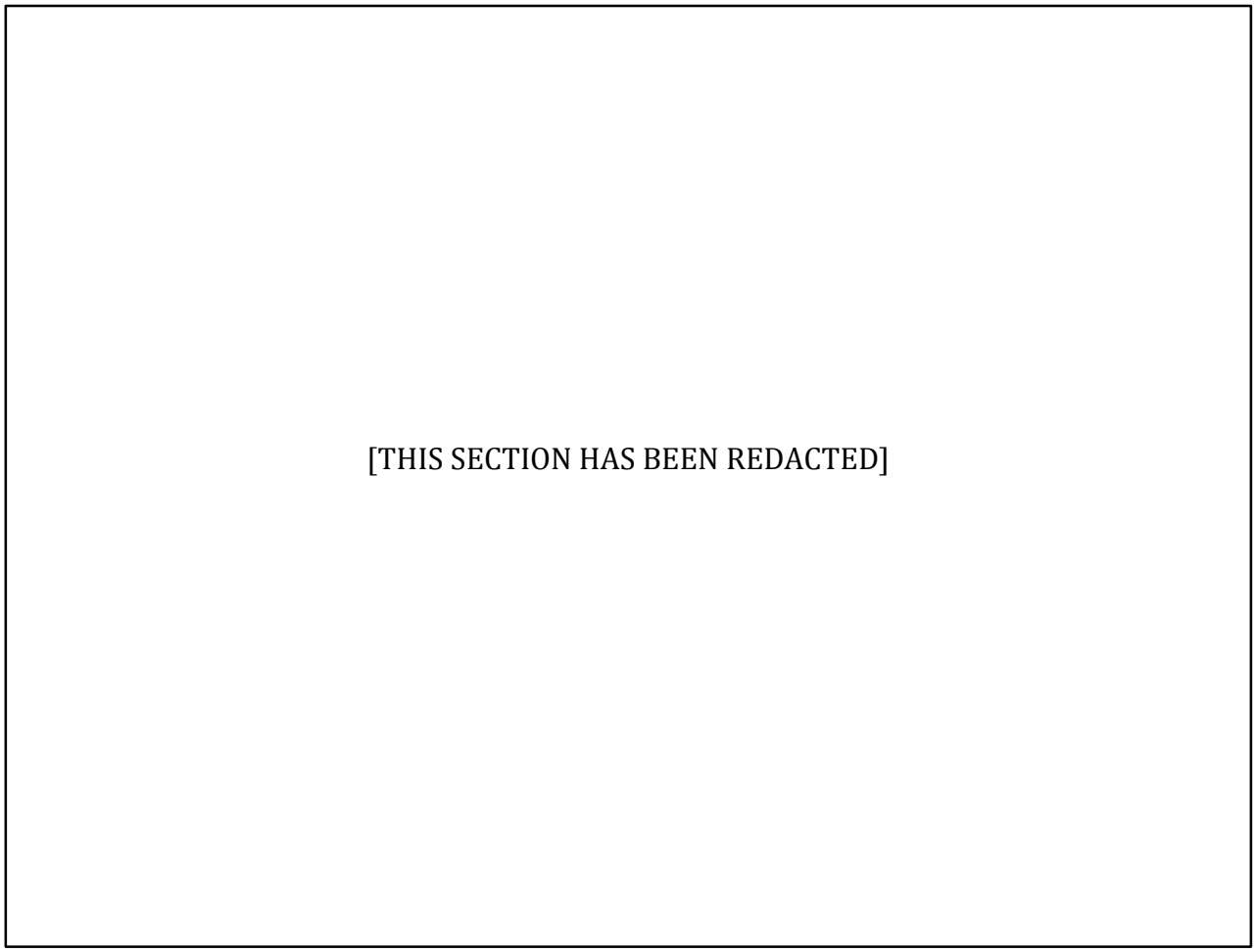


Figure 20: Survey and Entry Requirements

4.6.5. In Database and Exiting Information

In the *'In Database and Exiting Information'* sections (Figure 27), the team captured requirements related to community members being in the repository as well as leaving the

repository. The in database section contains two yellow and five blue requirement notes: track people in use currently and keep in contact with members. Beyond those standard requirements, the team identified five others: keyword search, filters to sort, tags, edit members, and reporting. These pieces of functionality defined by the team are important for users processing the member data. In the exiting information section, the team created two blue notes: removing members and export, so that members can be removed from the system and their data can be exported.

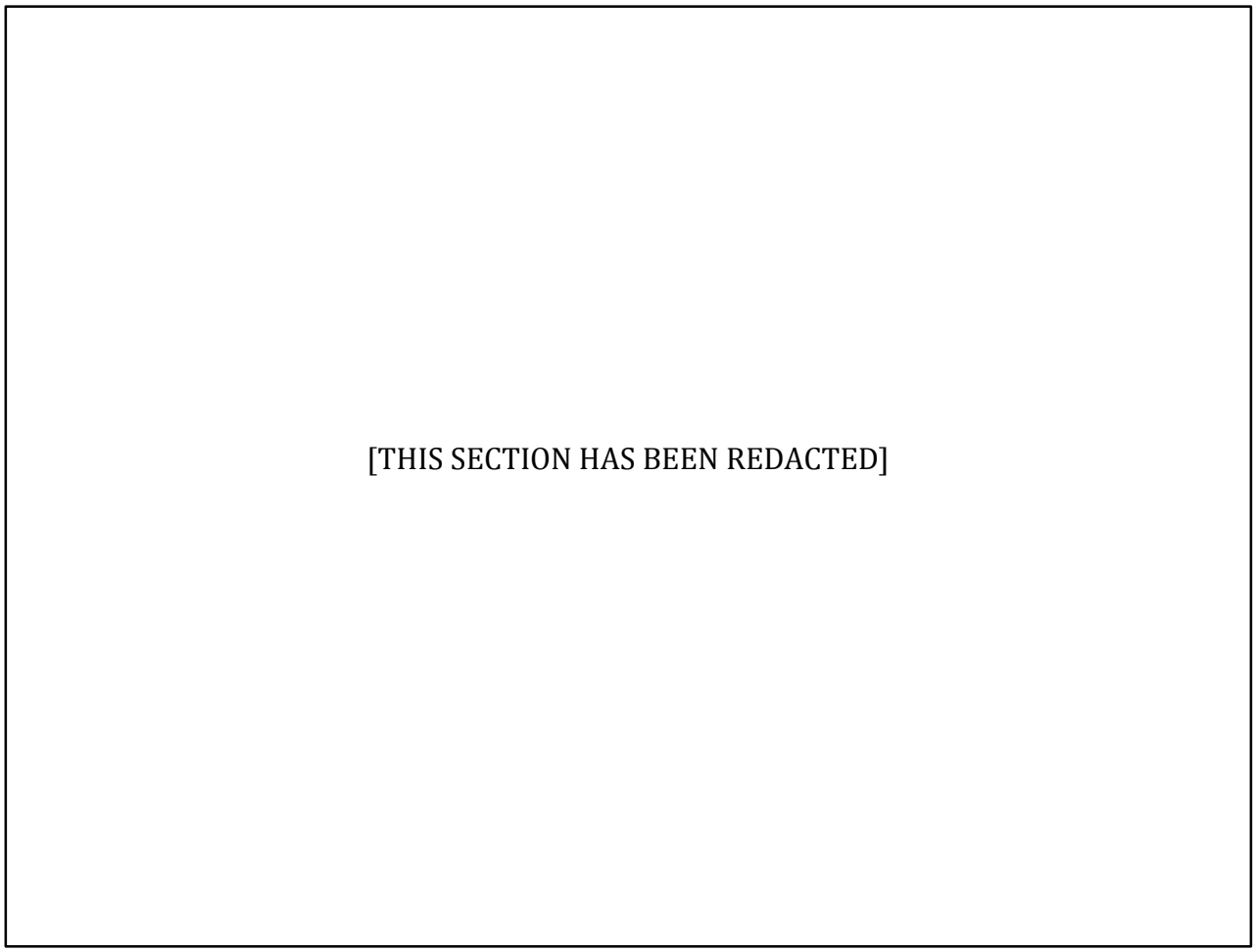


Figure 21: In Database and Exiting Info Requirements

4.6.6. Overall (Overarching) Requirements

In the last section of the requirements diagram (Figure 28), the team put notes that capture overarching requirements for the project. These overarching requirements were defined during the requirements gathering process and deal with multiple areas of the recruiting process. The first yellow note is scalability because the system should be scalable in the future to accommodate an increasing number of members. The next yellow note is the ability to integrate with Catalyst, Communispace's proprietary software system for managing communities. After continued discussion, the team included it in their requirements diagram, but it was determined to be out of scope, as it was not technically feasible based the constraints of the project. The team identified three blue note requirements: secure access to the repository, integration with the Vendor Performance Relationship Management (VPRM) system, and user concurrency.

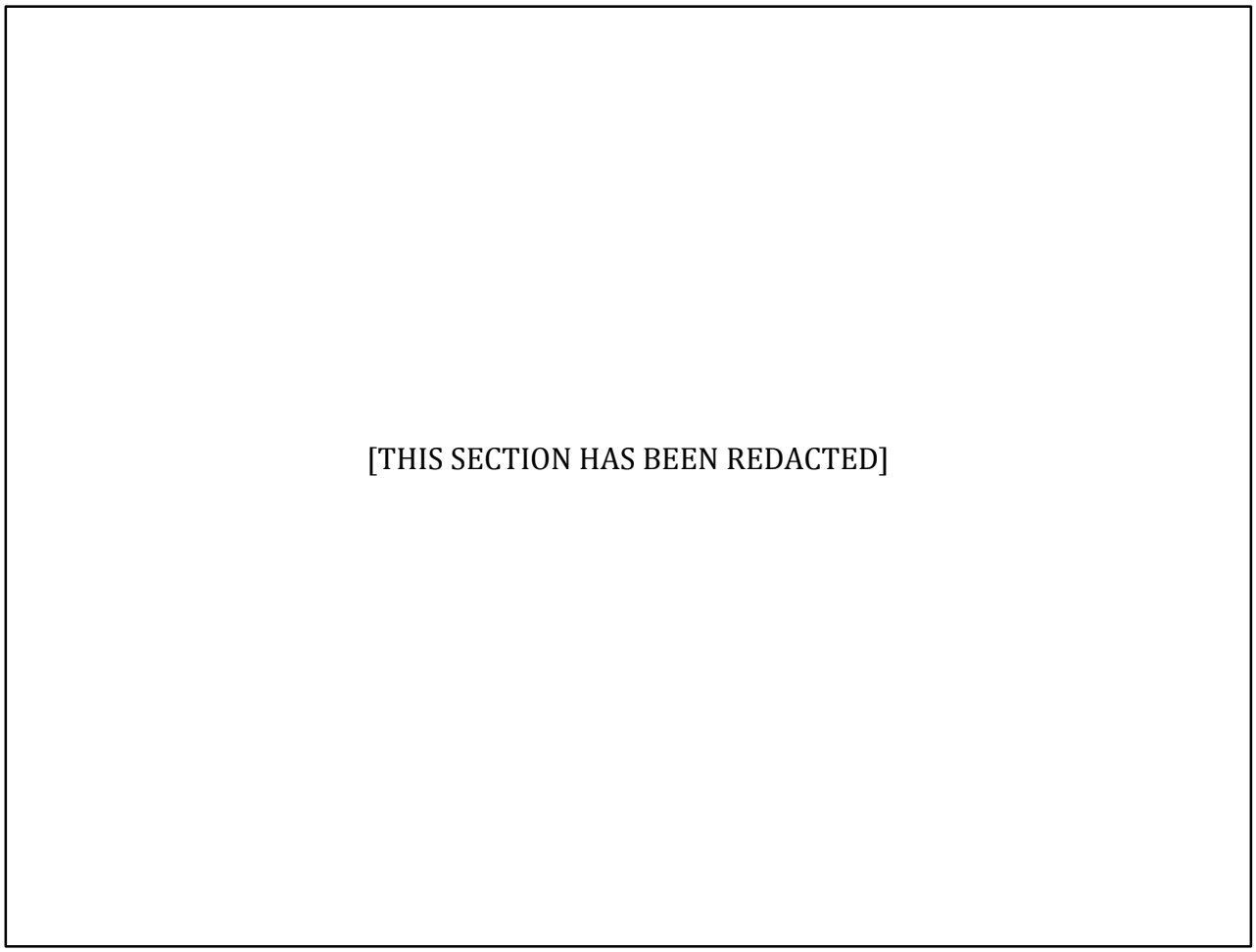


Figure 22: Overarching Requirements

The process of diagramming the requirements, using yellow and blue sticky notes, was beneficial as it allowed the team to visualize how the requirements aligned with the business process. By following this process, the team was also able to extend the requirements beyond what was explicitly stated in the information gathering phase and create a more comprehensive system.

4.7 System Proposal

In order to help the Communispace Client and Consumer Services team recruit members for their communities more efficiently and effectively, the team proposed the creation of a database system prototype. The system allows users to input potential future community members at various points during the recruitment process, including during the screening process and after communities sunset and end. It also allows users to search the database using several different criteria and/or keywords in order to find potential recruits for new communities. Users are able to edit the member information stored in the database and have the ability to remove people in the event that members do not want to be contacted further.

5. Design Phase

Using their insight from the analysis phase, the team conducted the design phase, where system requirements and functionality were adapted into the final system design. The team provided the Client and Consumer Services team with several designs and iteratively integrated feedback throughout the process. In designing the system, the team addressed system architecture, user interface, and program design. Subtopics relating to those areas are addressed and detailed in the following section.

5.1 Design Plan

The design phase is highly dependent upon analysis of the project. The main source of input for design is the requirements from analysis phase. The design phase sets up how the system will be constructed theoretically.

There are three types of design methods: Custom Development, Packaged software, and Outsourcing. The custom development method is for large enterprise systems with highest level of customization. Packaged software is usually suitable for medium size companies with a limited level of customization. Outsourcing is when for systems that are not essential to the business. Figure 29 summarizes the three methods in detail.

	When to Use Custom Development	When to Use a Packaged System	When to Use Outsourcing
Business need	The business need is unique.	The business need is common.	The business need is not core to the business.
In-house experience	In-house functional and technical experience exists.	In-house functional experience exists.	In-house functional or technical experience does not exist.
Project skills	There is a desire to build in-house skills.	The skills are not strategic.	The decision to outsource is a strategic decision.
Project management	The project has a highly skilled project manager and a proven methodology.	The project has a project manager who can coordinate vendor's efforts.	The project has a highly skilled project manager at the level of the organization that matches the scope of the outsourcing deal.
Time frame	The time frame is flexible.	The time frame is short.	The time frame is short or flexible.

Figure 23: System Acquisition Strategy

Based on each of five parameters shown in Figure 29 above, the best system acquisition strategy for this project is packaged system method considering the 21-week project time frame, common business need, and certain level of existing In-house functional experience. The benefits of choosing packaged software include a level of customization and low cost.

5.2 Alternative Matrix

To design the Prospective Member Sourcing System, the team analyzed the possible development environment alternatives to build the system. The team settled on five grading criteria for development options: scalability, cost, technical abilities, learning curve, and security. *Scalability* is an important factor when choosing a program. The team wanted to provide a prototype that will provide instant value to a company that is growing rapidly like Communispace. *Cost* is another factor when developing the prototype. Purchasing licenses for software programs for all of the users of the system can be a costly

endeavor. Research favored already-existing Microsoft software such as Access and SSRS because of the price and availability to team members. *Technical abilities* were also assessed when deciding upon a program. The development of a program was scheduled to take a few weeks; if the stakeholders already knew how to use the software, then it would be easier for them to build on to the program. If there was a steep *learning curve*, the developers may have needed to spend extra time in training. This ultimately would have hindered the project's deadline. *Security* was an important consideration because the system contains sensitive data about potential members.

The team applied these criteria over several different options, as seen in Table 7: Access, SQL Server Reporting Services, Wakanda and NuBuilder. Access, a popular database software developed by Microsoft, is a convenient tool that allows for quick development and ease of use based on drag-on-drops of UI elements. It has limited scalability since it has a file size limit and does not quite solve for concurrency. However, this can be elevated if it is used with SQL server. Another development tool provided by Microsoft is SQL Server Reporting Services. Using this tool, developers can easily create reports. Front-end development skills are needed to support more complex functionalities of system. Wakanda is a web application development toolkit. The database languages it supports are MySQL, ODBC (Open Database Connectivity) and its own language, WakandaDB. It provides a free license, but the team would have to upgrade to a commercial/enterprise license for private deployment. NuBuilder is also a tool for building web applications. It is open-source and completely free. The database it uses is MySQL.

Table 8: Development Options Summary

[THIS SECTION HAS BEEN REDACTED]

The team further created a comparison matrix to help make the final decision, which is shown in Table 8. In the alternative matrix, the team assigned a numerical value to each of the aforementioned criteria for each development option. Values ranged from one (worst or most difficult) to five (best or easiest). This numerical assessment allowed the team to take a more objective approach when deciding upon a development method. Microsoft Access scored highest in the matrix, with high scores for technical skills (all team members had prior experience), learning curve (Communispace employees had prior experience), and cost.

Table 9: Development Options Matrix

[THIS SECTION HAS BEEN REDACTED]

The team decided to build the system prototype in Microsoft Access since it satisfied the project requirements while accommodated the team's ability and time scale.

5.3 System Architecture Design

In designing the new system, the team structured the architecture so it focused on centralizing member information and making it accessible. The objective of creating the

member repository was to shorten the length of time to recruit new users for communities. By centralizing all the data, the team provided the Client and Consumer Services team with a single authoritative location for candidate information that could be used in this process. Lastly, combining the centralization of data and accessibility factors, the system allows users to retrieve information easily by abstracting that process into an easy-to-use interface.

5.4 User Interface Design

The User Interface (UI) design of the system strictly followed commonly-used design principles as outlined in the System Design & Analysis book. The UI layout was designed for ease of navigation, providing an intuitive understanding of page hierarchy and function. Also, individual pieces of functionality, like buttons and other interface elements, act as the user would expect based on their labeling and context. Furthermore, colors and fonts were selected in accordance with the company's brand and style guides. The system's User Experience (UX) was designed and developed to be easy and intuitive for every user, by catering to the ability of the potential lowest common denominator. Consistency was also maintained throughout the UI and UX to help users quickly learn the system. With all the preceding guiding components, the team designed an elegant, yet simple system, minimizing complexity whenever possible.

In the rest of this section, an interface structure along with its corresponding interface template, are detailed. The interface structure and template were reviewed and approved

by project stakeholders before moving forward with development. It is important to note that the UI design was iterated upon during the development phase as the team continued to seek feedback from the system users and incorporated it into subsequent versions.

5.4.1. UI Structure and Hierarchy

The system's interface structure was created after the major stakeholders approved the team's project proposal. The interface structure is a comprehensive hierarchy of the various screens of the system and how they relate to each other. In Figure 30, the UI structure diagram is presented.

Figure 24: Potential UI Structure Hierarchy

The diagram consists of two component trees, which include different pieces of functionality in the system. The first tree captures the authentication process, where the

user logs in to the system. Based on their role, administrator or user (default), the user is presented with their associated functionality. Role is determined by the system administrators. After the user is authenticated into the system, the next tree becomes active. This authentication process was deemed unnecessary during the implementation phase, as authentication is now handled simply by restricting access to the application on the server.

In the second component tree, the homepage is divided into four options: search repository, manage members, add member(s), and reporting. These options segment system functionality into clearly understandable, sub-trees. The first subtree, search repository, allows the user to search by keyword or advanced search. When the results are returned, different actions can be performed. Next, the add member(s) subtree contains import from file and manually add member functions.

In the last two subtrees, the system is divided into manage members and reporting. The manage members page includes functionality for editing, removing, and exporting member data from the system. Lastly, in the reporting subtree, users are able to generate a number of different reports, using the repository data.

5.5 UI Mockup

Based on the interface structure, the team created an interactive UI mockup, which was presented to project stakeholders. System layout, design, and functionality were included in the interactive mockup so that the stakeholders understood how the system would work

when it was completed. In this section, the team has taken screenshots of the interactive prototype to outline the UIs. While the interactive mockup inherently explained how the system worked by physically performing the actions, the screenshots are fully explained in the next section.

5.5.1. Home

Similar to the interface structure, the top-level interface, or home screen, was segmented into four sections with the four buttons: Search Repository, Add Members, Manage Members, and Reports. By selecting one of the buttons, the user was taken to the page that was described by the button name. For example, the Search Repository button directed the user to the search repository page. A visual representation of the Home mockup is displayed in Figure 31.

It is also important to note that the top navigation bar was consistent across all the pages in the system. On the left side of the navigation bar, there was a "bread crumb"-style menu that displays the user's current location. In Figure 31, it was set to *Home* (because the user was viewing the home page); if the user selected the Search Repository button, the navigation menu would update to *Home > Search Repository*. On the right side of the navigation bar, the user could use the Log Out button, taking the user out of the system, or access the Help button.

Figure 25: Home Page UI Mockup

5.5.2. Search

Immediately when the search repository page loads, the user was presented with two options: Keyword Search or Advanced Search (as seen in Figure 32). For explanation purposes, the former will be detailed before the latter.

Figure 26: Search Page UI Mockup

In the next Figure 33, the search results were displayed below the search boxes. The results could have been selected for export or edited directly. By searching again, the user would have cleared the results and new ones would have been returned. If the Export button was selected, a user would be prompted to choose a name for the resultant exported file and save location.

Figure 27: Search Page UI Mockup - Continued

Lastly, the Advanced Search function was more complex than Keyword Search, but was still intuitive for users to understand. In Figure 34, a user selected a field from the dropdown, an operator, and a value. For example, a user could select the Gender field, Is operator, and Female value, thus returning search results that meet that criteria. Below the value field, the plus button allowed users to add multiple criterions, stacking them on top of each other to create ranges and boundaries. For example, a salary range on the field income, operator greater and less than, and values of 50,000 and 75,000.

Figure 28: Advanced Search UI Mockup

5.5.3. Add Members From File

When the Add Members button was selected on the home screen, a dropdown was displayed to the user, with the options From File and Manually shown in Figure 35. By selecting the From File option, the users was guided through the import members process, and by selecting Manually, the user was directed to fill out a form manually.

Figure 29: Add Members Dropdown List UI Mockup

When the user selected the From File option, the user saw the Add New Members From File page (Figure 36). On this page, the user clicked the Browse button, which prompted them to select a file. There is also a warning, specifying that the file cannot be larger than 2GB. Once the user selected a file in the file browser, the user then pressed the Submit button to continue the import process.

Figure 30: Add Members from File UI Mockup

After selecting Submit, the user next saw the Add New Member(s) Confirmation page (Figure 37). Here the user checked how the system performed when importing the file. The user had two options at this point: Go Back, taking the user to the previous page, or OK to Continue, importing the file. Before selecting Ok to Continue, the user selected the records that should be imported by selecting the checkboxes. This is important because the system might not be able to read all the records in the imported file, resulting in corrupted or incomplete data. In this case, the user could quickly scan the records and only import the ones that meet their quality standards. There was also an option to Select All.

Figure 31: Add Members Confirmation Page UI Mockup

Finally, when the user selected OK to Continue, if the import process was successful, the user was shown a Success page (Figure 38) and could view the records that were imported. At this point, the user could return home or import another file.

Figure 32: Add Members Success Page UI Mockup

5.5.4. Add-Manually

If the user selected the Manually option from the Add Member dropdown on the home page, the user was shown the Add New Member Form (Figure 39). Here the user could add new members one at a time, by filling out all the requisite fields and submitting the form. For convenience, the user was prompted for any errors and was given date pickers for the date-related fields. The two main actions on this page were Cancel and Submit.

Figure 33: Add Members Manually UI Mockup

After the user submitted the form, they would be shown a success message (Figure 40) and the member data that was just added. The user could choose to return home by pressing the Home button or add another member manually by pressing Add More.

Figure 34: Add Members Manually Success Page UI Mockup

5.5.5. Manage

By selecting the Manage Members button on the home page, the user was brought to the Manage Members page shown in Figure 41, where they could perform management actions on all the members in the repository. The user could view members by status by selecting one of the four tabs: All, In Use, Not In Use, or Recently Added. By selecting row checkboxes, the user could export to Microsoft Excel or remove the selected members by pressing their corresponding buttons. Lastly, the user could edit member data inline by selecting the cell-like field and updating its value. Any changes made by the user were updated whenever the user changed their selection or left the page.

Figure 35: Manage Members UI Mockup

5.5.6. Reports

The report section was not included in the team's UI mockup, since the team did not have enough information to proceed with creating reports. The team's first impression of the report section was that it would contain high-level summary measures that decision-makers in the business would be able to reference. The team gathered more information from the project stakeholders and implemented it in later versions of the system.

5.5.7. Use-Scenarios

In an effort to better understand what the User Interface would look like, the team developed Use-Scenarios. As defined in the System Analysis and Design textbook, "A use scenario is an outline of the steps that the users perform to accomplish some part of their

work" (Dennis, Wixom & Ross, 2012). By establishing these scenarios through analysis of the to-be system, mainly the data-flow diagram and the "sticky notes" of designing the system, the team was able to navigate through a process and apply it when designing the user interface.

[THIS SECTION HAS BEEN REDACTED]

5.6 Program Design

Physical process models were created to show implementation details and to explain how the final system works. The team also provided a structure chart that showed all the components of the program at a high level to assist in designing the system. The components are arranged in a hierarchical format to enforce a sense of sequence, selection and iteration. To achieve its best quality, the structure chart was designed with high cohesion, loosely coupled modules and high fan-in as possible.

A series of detailed program specifications for different modules and processes was created to provide a comprehensive documentation for both the ease of learning and for further development in the future.

5.7 Entity Relationship Diagram

The Prospective Member Sourcing System was designed to store member, community, and user information. With its connection to the existing 'Dupe Check' and 'Vendor Performance Relationship Management' systems, Communispace employees are able to

search prospective members, manage members, view reporting metrics, and add new members to the system. There are eight main tables in the PMSS database: [THIS SECTION HAS BEEN REDACTED]. A screenshot of the entity relationship diagram for the system is shown in Figure 42.

Figure 36: Entity Relationship Diagram

[THIS SECTION HAS BEEN REDACTED]

6. Implementation Phase

During the implementation phase, the team had multiple levels of focus as they integrated the system into the Client and Consumer Services team and operations. Building off the preceding analysis and design phases, implementation included system development, testing, database population, and support and documentation plans. The team split up work to complete the different system implementation parts and the methodology that was used will be detailed throughout this section.

6.1 Development

While in development, the team primarily used their background in analysis and design to translate from a theoretical to a practical system. Based on the team's results from the design phase, the platform and technologies were determined. The development process was conducted by the team using the system prototyping method. This methodology allowed the team to iterate and include all the requisite capabilities for the system. Once in development, there was refinement of the stakeholder system requirements and these were integrated as necessary, where they aligned with the project scope.

6.1.1. Development Challenges

Search

The Search function, divided into “Keyword Search” and “Advanced Search”, was technically very challenging and time-consuming to develop. Initially, the search function

was able to search keywords across all data fields, which then based on feedback, was changed to exclude names and emails. The advanced search took about six weeks to fully function as expected because it required an intense amount of programming and debugging using Visual Basic in Access.

Import from file

The import from file function was technically beyond the team's knowledge when we started working on this function. The team spent weeks researching and implementing various methods, which ranged from building Macros to coding in Visual Basic (VB) and Structured Query Language (SQL) to importing manually. Even when methods did work, there were significant problems with the business flow; for instance, one attempted method required system users to significantly manipulate the Excel files before importing them to the system. After several weeks of trials, the team approached Mr. Sean Burke, who encouraged the team to look at some past work of his own that had VBA (Visual Basic for Applications) code which the group was able to modify and use in the final system.

System Integration

Another challenge the team ran into is system integration. For the system to maximize its usability and usefulness, it needs to work with other systems at Communispace such as VPRM and DupeCheck. Luckily, the project's onsite liaisons shared their experiences of integrating different systems built in Access for them to communicate with each other. The team was finally able to integrate past systems with the Prospective Member Sourcing System.

6.1.2. Development Refinements

Universal Design

The system was refined visually with a universal design concept. This includes unified graphic alignment, consistent fonts and colors, and careful spacing. The team was able to design the system based on Communispace colors: light blue, grey, and orange. The design of the user interface focused on simplicity while providing the necessary functionalities. The team also created a responsive interaction style for ease of navigation.

Coding Revision

In terms of system functions, the Search Repository, Add Members From File, and Manage Members functions were developed using VB code. Programming requires a debugging on a regular basis as the system grows during development. The team kept revising the code for those functions in all iterations of the development phase to refine the functionality, usability, and stability of the code.

6.2 System Iterations

During the implementation phase, the team created several iterations of the system. Each iteration took one to two weeks to complete and was shared with Mr. Sean Burke and Mr. Jack Bergersen, as well as occasional other stakeholders. Following these meetings, the team worked to incorporate additional features and suggestions into the next iteration of the system.

6.2.1. First Iteration

The first iteration of the system was competed over the team's Winter Break period and shared with Mr. Bergersen and Mr. Burke in mid-January. The first iteration focused on the creation of all basic functionalities of the system.

Homepage

The homepage was designed based on the four functionalities the system should handle: Search Repository, Add Members, Manage Members, and Reports as shown in Figure 43. Each of the four functional buttons, once clicked, directed the user to its corresponding functional form. The quit button simply closed the system once clicked. The design used in the home page came from Communispace's requirement for simplicity and is consistent with Communispace's corporate color scheme: light blue, grey, and orange.

Figure 37: Homepage - 1st Iteration

Search Repository

The Search Repository function contained two different search areas: Keyword Search and Advanced Search. The Keyword Search searched across all fields for the entered value. For example, searching for Female returned all the records that contained the text “Female”. Advanced Search allowed users choose their desired search field with two available parameters as shown in Figure 44.

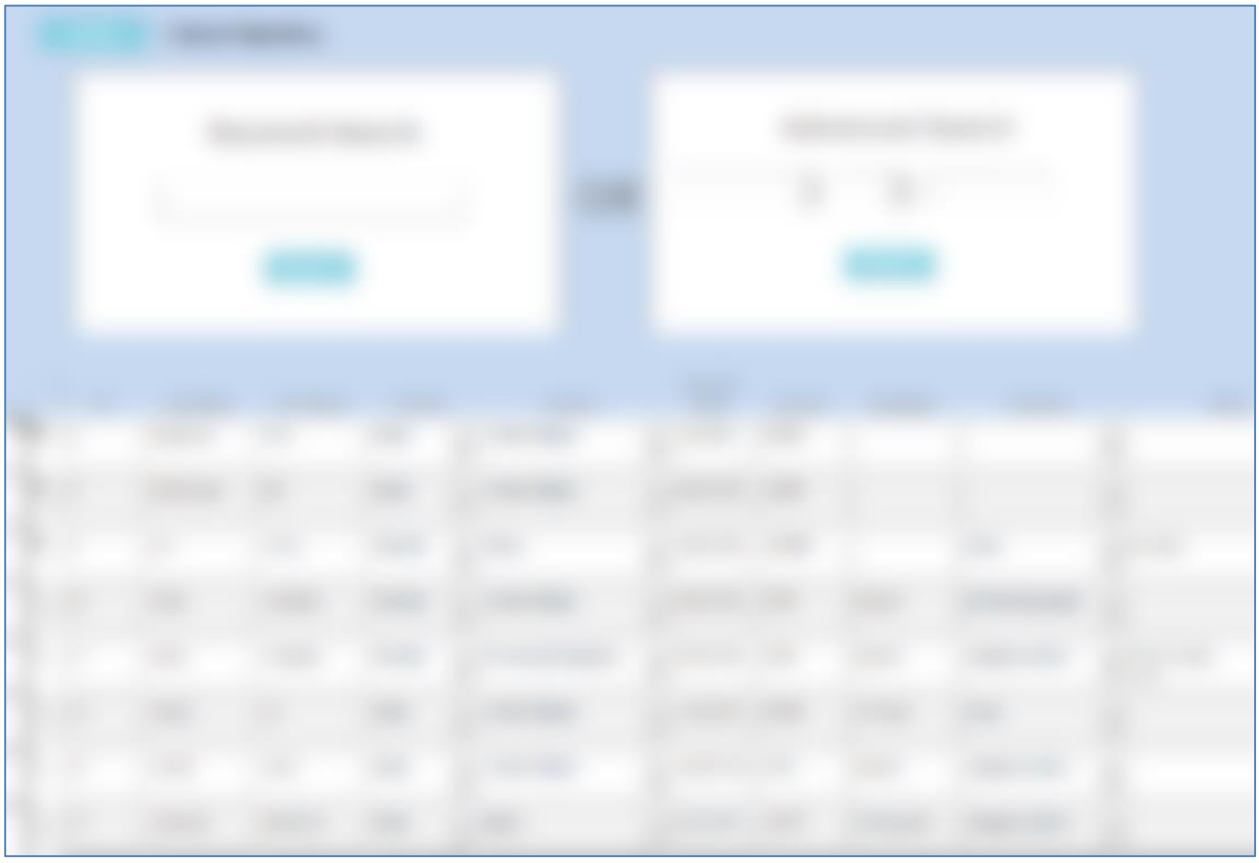


Figure 38: Search Repository Page - 1st Iteration

Advanced Search

The Advanced Search function, as of the first iteration, was able to perform searches across 11 profile fields listed in Figure 45 with parameters including “contains” and “is” shown in Figure 46. After discussing the first iteration with Mr. Burke and Mr. Bergersen, the team received the requirement of adding more search fields, thus allowing the user to narrow the search field further.

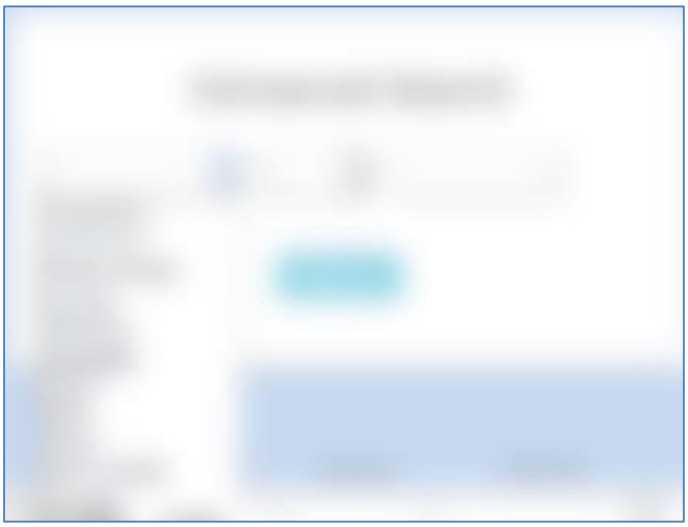


Figure 39: Advanced Search Options - 1st Iteration

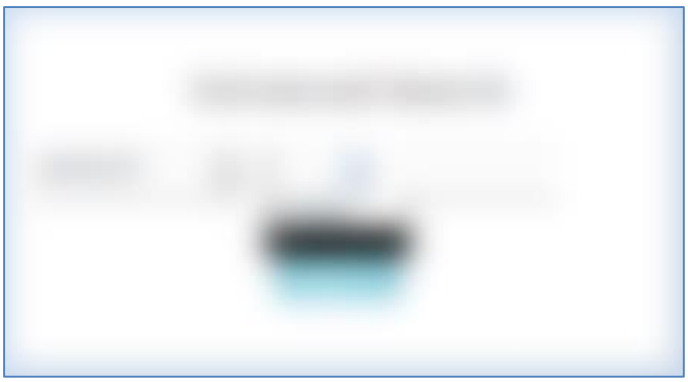


Figure 40: Advanced Search Parameters – 1st Iteration

Add Members

When the user clicked on the “Add Members” button, a drop-down of two buttons appeared below in order to allow users to choose between adding members from an external file or manually. A visual representation of this function is displayed in Figure 47. Each of the two sub buttons directs the user to its corresponding page.

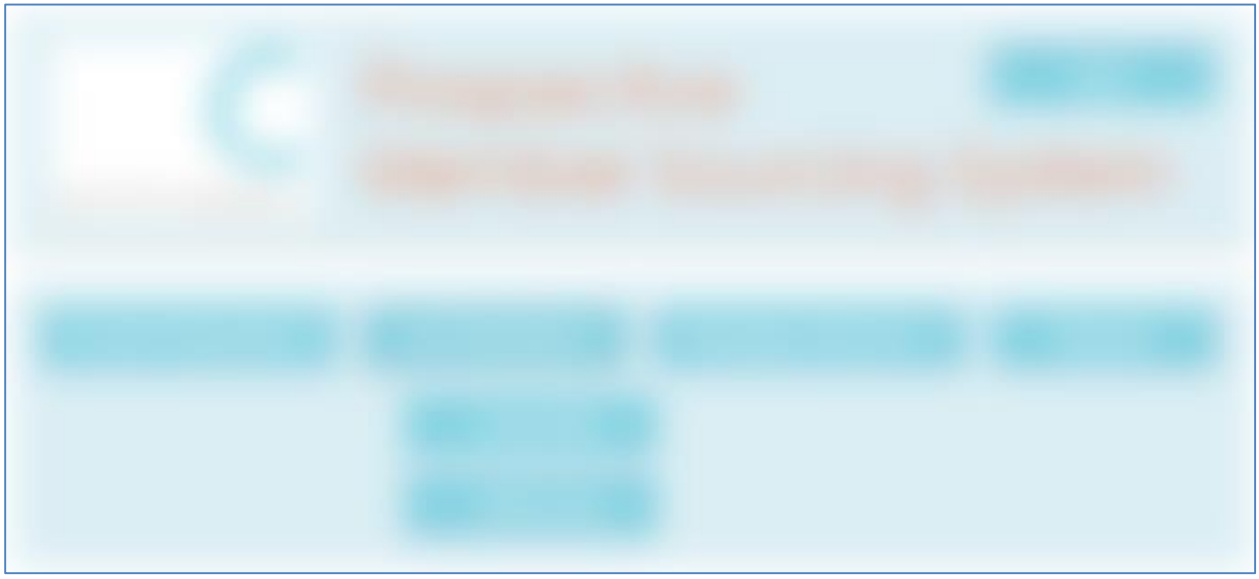


Figure 41: Add Members Options - 1st Iteration

Add From File

At the first iteration of the system prototype, the Add Members from File function was not yet built nor designed. Therefore there was nothing except breadcrumb navigation and the title on the Add From File page as shown in Figure 48. During the first iteration, the team researched different methods for importing from an Excel file but was not able to implement any of these methods.

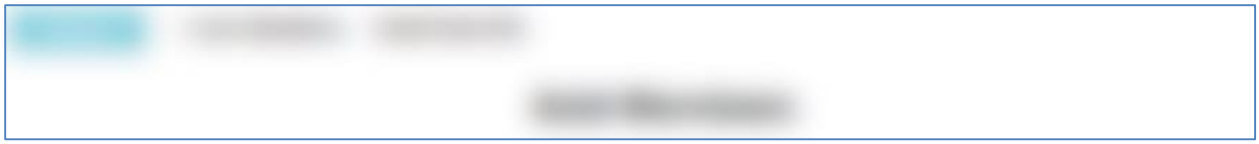


Figure 42: Add Members From File - 1st Iteration

Add Manually

Add Members Manually directed the user to a form of 11 input fields with input data types ranging from text, date, number, and drop down list. After a user typed in or selected information, a click on the "Add Record" button would store the data inputted into the database. At the first iteration, one problem that the team faced was that upon opening, the form displayed the first record in the database instead of blank fields as shown in Figure 49. None of the fields were required in order to add a record to the PMSS.

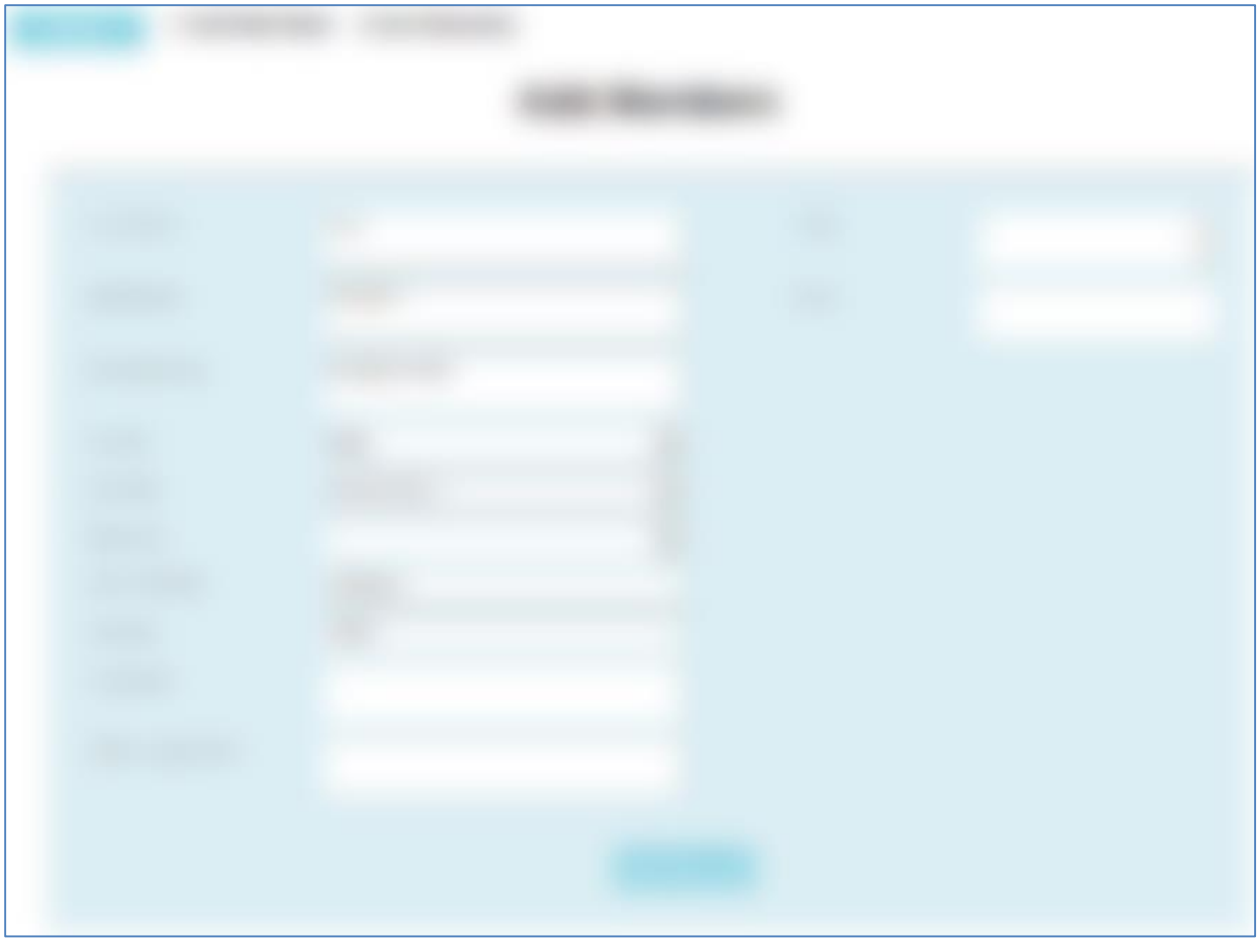


Figure 43: Add Members Manually - 1st Iteration

Manage Members

The Manage Members function allowed user to select members to either remove them from the system or export the members for reuse. A screenshot of the Manage Members page is shown in Figure 50. As of this stage, the Manage Members page was not integrated with the Search Repository function. Rather, a user could select members using the checkboxes on the left-hand side and perform basic functions of exporting to an Excel file or removing the selections from the system.

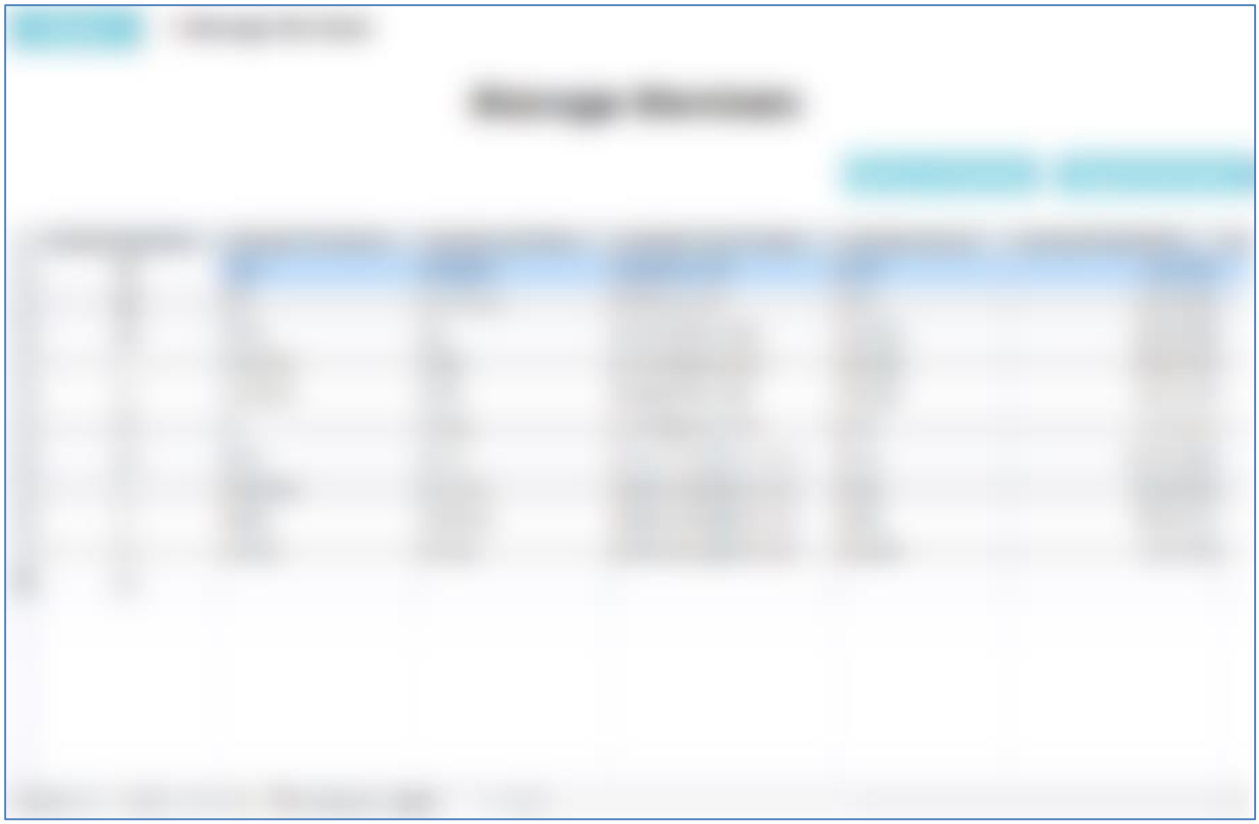


Figure 44: Manage Members - 1st Iteration

Remove Selected

The Remove Selected button, once clicked, directed the user to a Confirmation page where the user could double check before actually removing selected members from the system. The confirmation page is shown in Figure 51. On the confirmation page, once the button "Confirm Delete" was clicked, there would be a popup window (Figure 52) warning the user in order to prevent users from accidentally deleting members from the system.

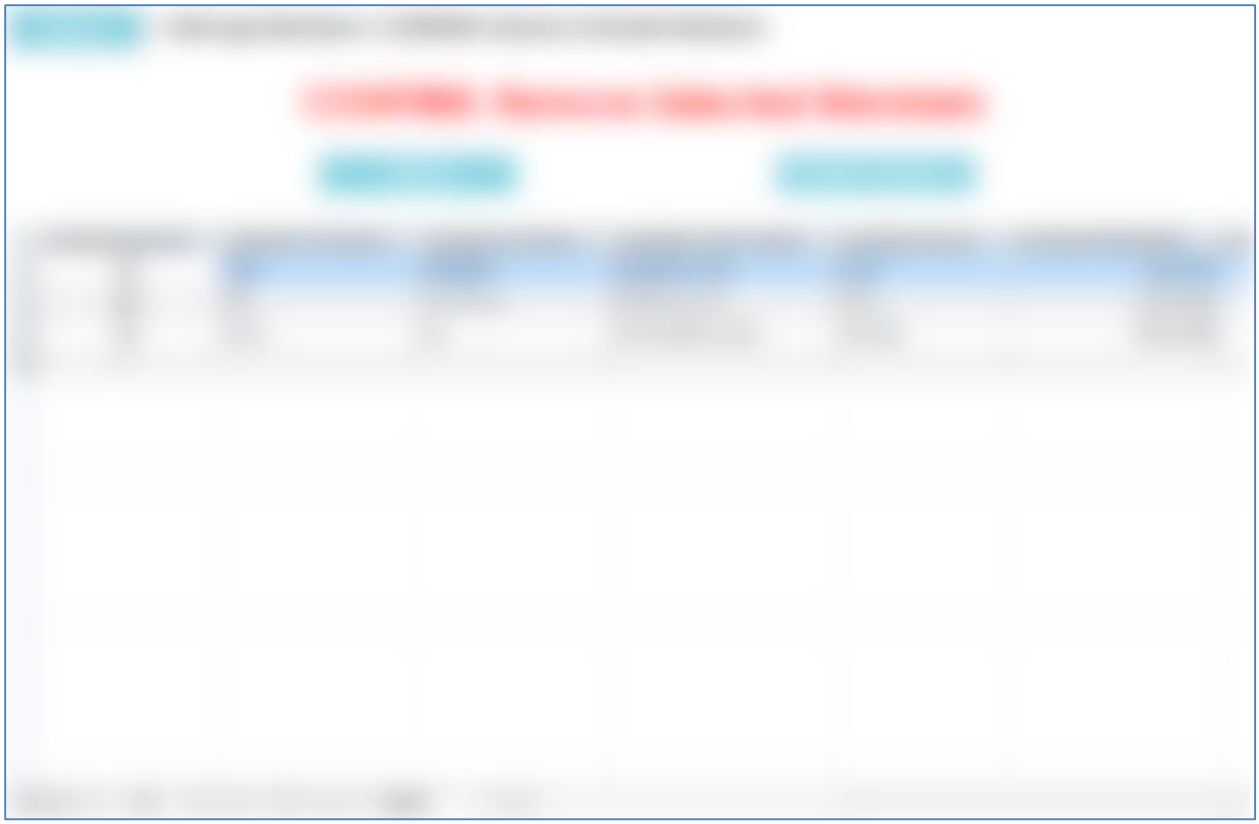


Figure 45: Remove Members Confirmation Page - 1st Iteration

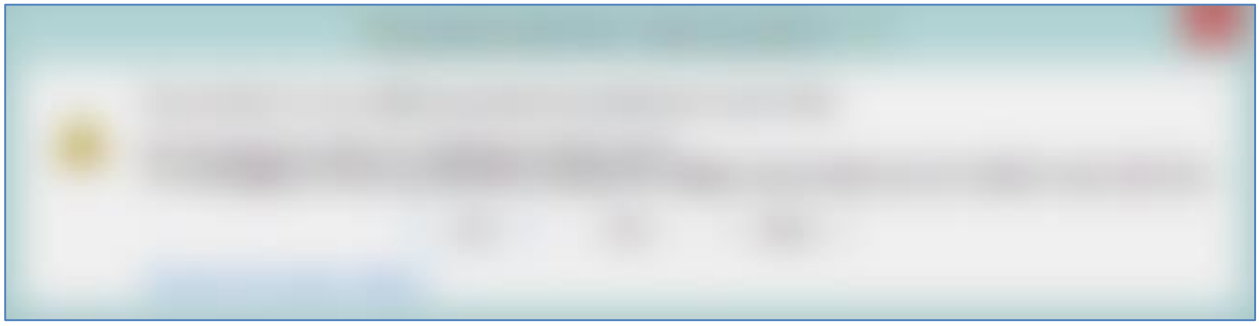


Figure 46: Remove Members Warning Popup – 1st Iteration

Export Selected

The “Export Selected” button, once clicked, directed the user to a pop up window (Figure 53). This pop up window allowed the user to save the exported members’ data in a desired file directory for further analysis and reuse.

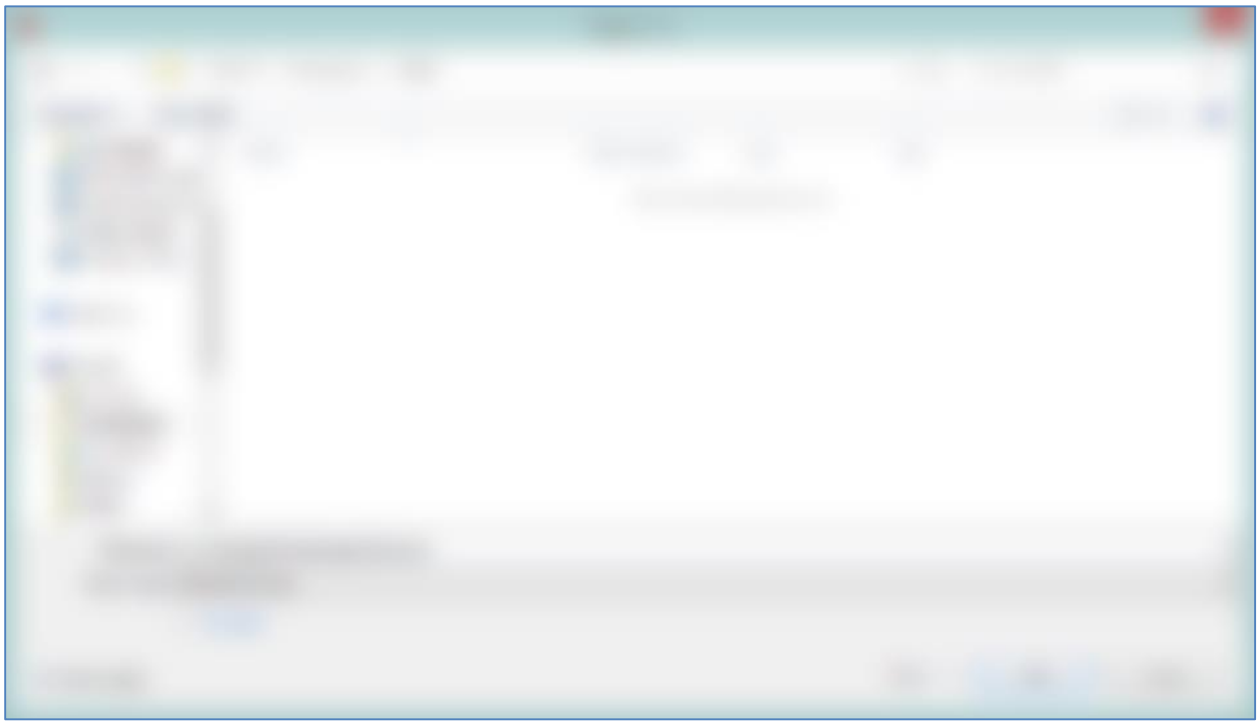


Figure 47: Export Members File Directory - 1st Iteration

Reports

As the screenshot displayed in Figure 54 shows, the Reports page was not designed as of this iteration. The reporting function was deemed low priority compared to other functionalities in the system. This page was intended to show some high level statistics of the system.

Figure 48: Reports Page - 1st Iteration

6.2.2. Second Iteration

The second iteration of the system was completed at the end of January. It incorporated feedback from stakeholder meetings and expanded upon several system functionalities. The changes made in the second iteration are detailed in the next section.

Homepage

The feedback from the project sponsors about the system UI was quite positive so there were not any changes made to the look of Homepage. One addition to this page, compared to its first iteration, was the addition of a date and time stamp at the bottom of the screen, shown in Figure 55. The time refreshes any time a button is clicked in the system.

Figure 49: Homepage - 2nd Iteration

Search Repository

The Search Repository page had some major changes in the second iteration. There were three buttons added on the top of the page: Reset, Export, Select All, and Deselect All. Furthermore, the Advanced Search was able to add up to three search parameters with the new “plus” and “minus” buttons. All of the changes that were made to the Search page are show in Figure 56.

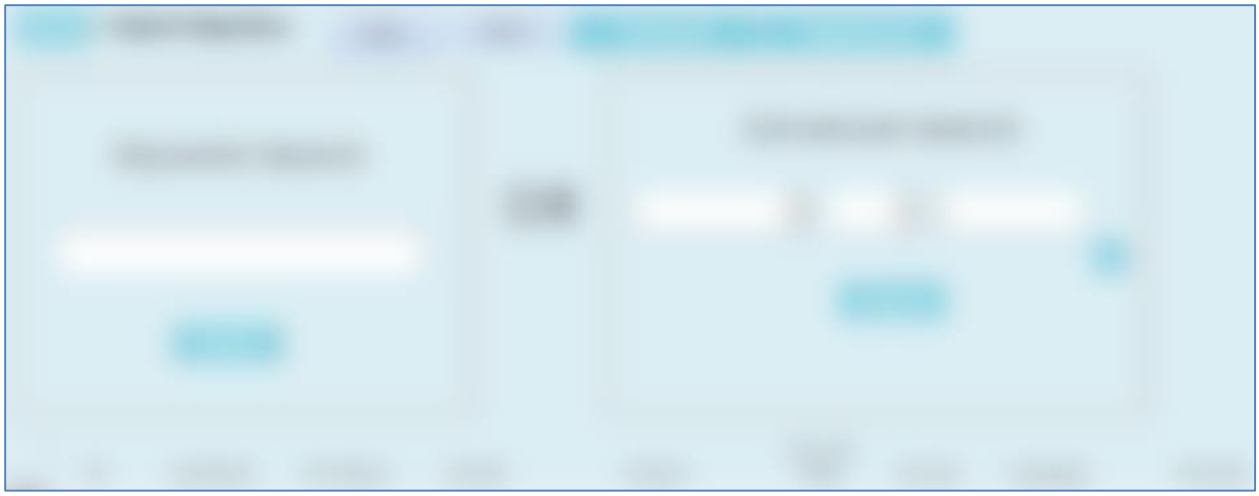


Figure 50: Search Page - 2nd Iteration

Advanced Search

The Advanced Search function added up to three search parameters based on the feedback from the first system iteration as shown in Figure 57. However, the team was given feedback on this iteration which suggested that the team should not include members' first name, last name, or email address as search parameters as these fields don't provide much value. Removing these parameters would also decrease the work load of the search function.

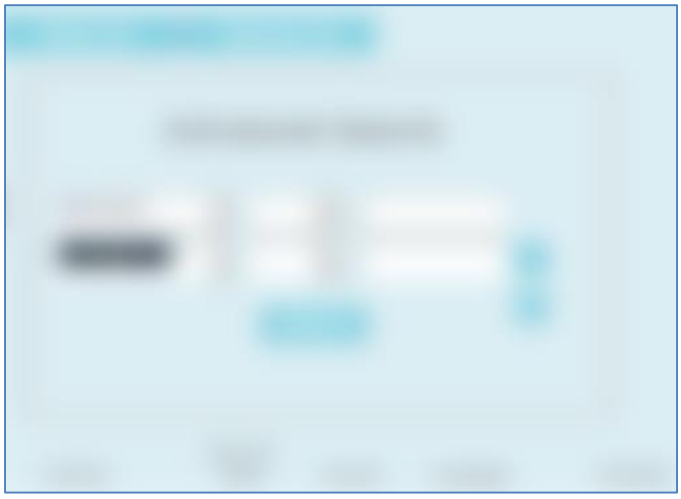


Figure 51: Advanced Search - 2nd Iteration

Add Members

As in the first iteration, the user can select from two ways of adding members on the Home page by clicking the "Add Members" button (Figure 58) then clicking one of the two sub buttons: "… From File" and "… Manually". There were no changes made to the homepage.

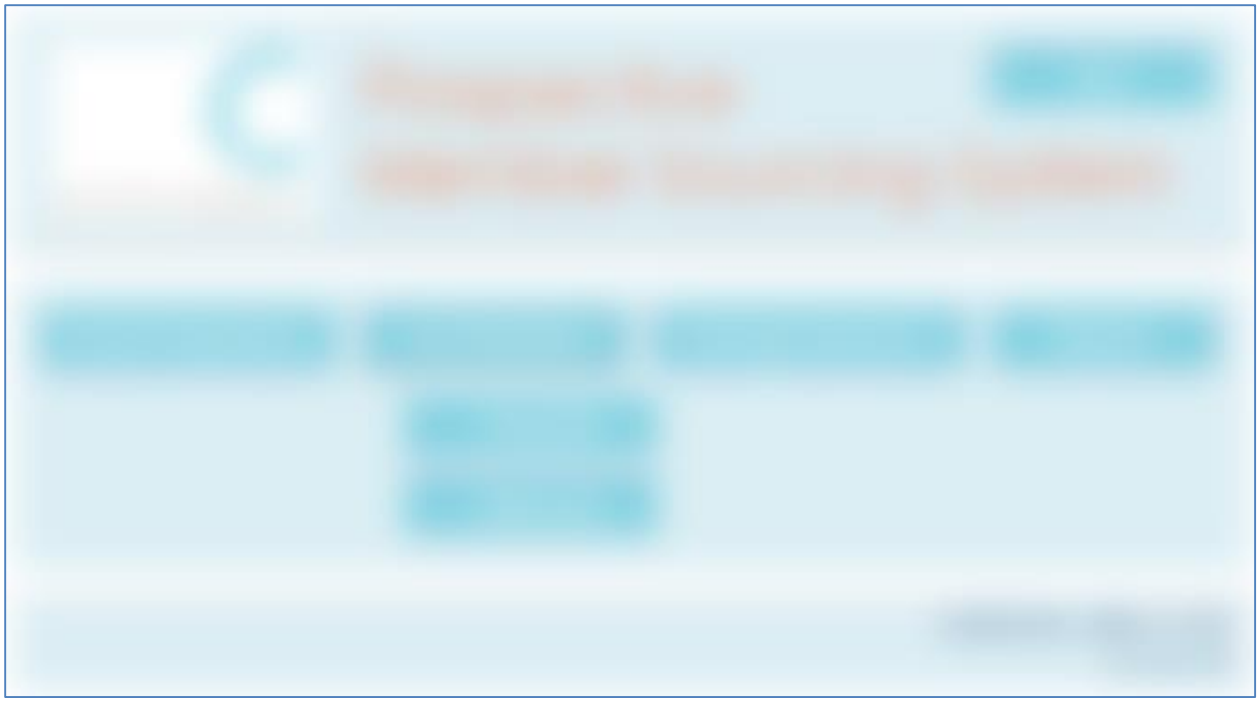


Figure 52: Add Members Options – 2nd Iteration

From File

The Add Members From File page was not developed at the second iteration as the team ran into technical issues achieving this function. The team tried several approaches including using VB to program the function and using a Macro to link an import template. Unfortunately, the team was met with errors for both methods. The team continued to study and conduct research on how to best implement this function. The screenshot in Figure 59 shows the Add Members From File page with universal elements including breadcrumb navigation, date, time, and title.

Figure 53: Add Members From File - 2nd Iteration

Manually

For the Add Members Manually page, the team was able to fix the problem in the first iteration of having to display the first record of user data in the system. The eleven input fields are defaulted to be blank in this iteration, as can be seen in Figure 60. Additionally, the Language field became a drop-down menu, the date and time stamp was added to the top right of the page, and two new buttons were added to the bottom of the page. The first, Clear Form, clears any data inputted by the user in the fields, while the second, Cancel, returns the user to the Home screen.

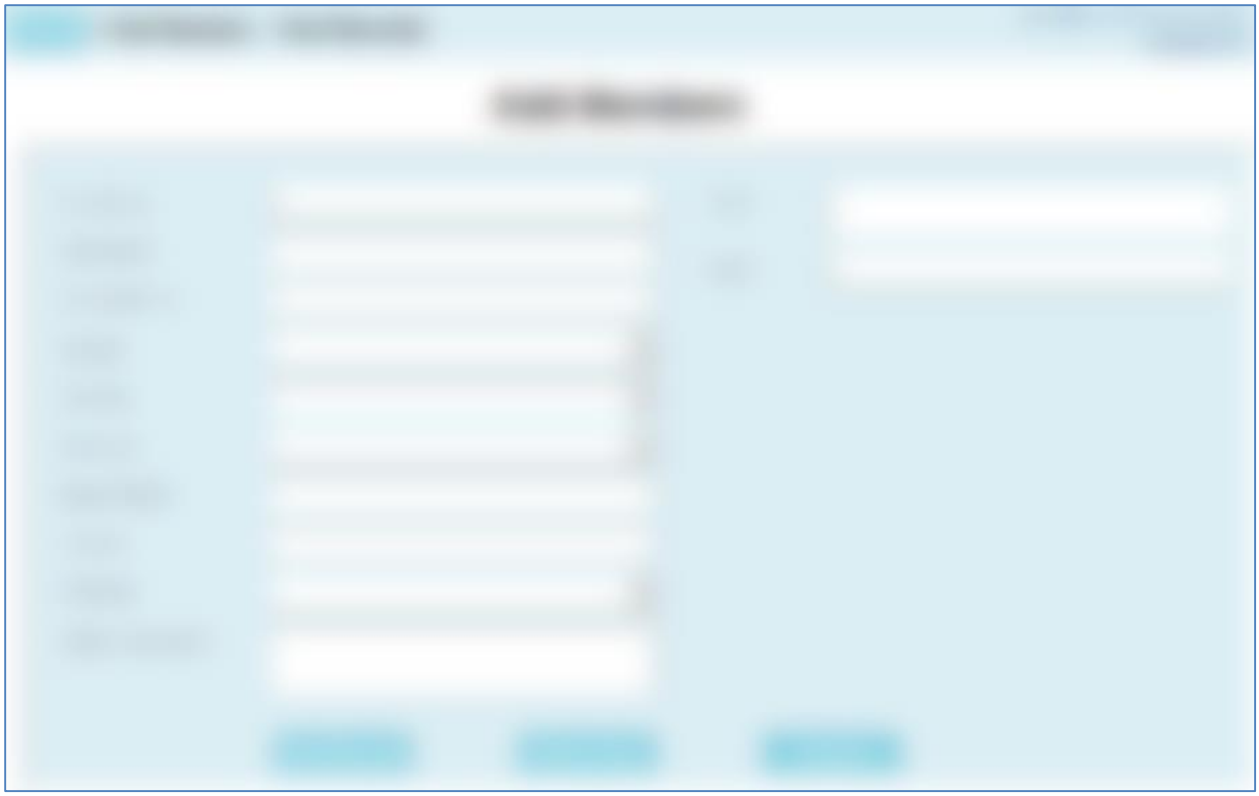


Figure 54: Add Members Manually - 2nd Iteration

Manage Members

The Manage Members page experienced some major renovation during the second iteration (see Figure 61). There were two buttons added: Select All and Deselect All. The team also changed the way member data was displayed to match the format of the Search Repository page for better integration between the two functions.

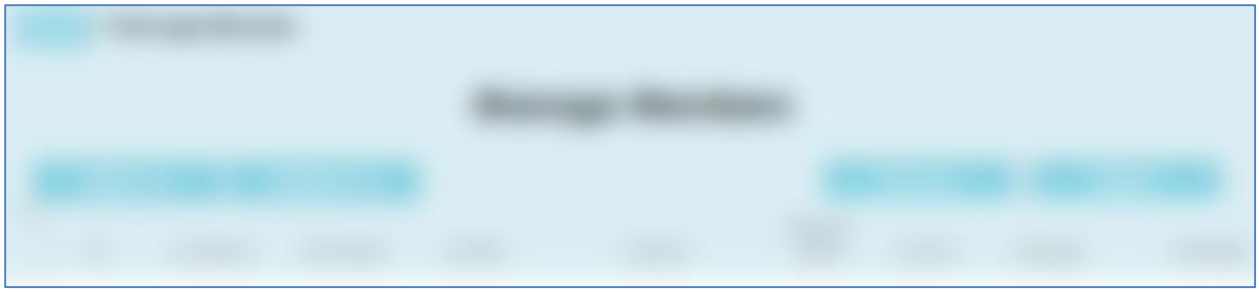


Figure 55: Manage Members - 2nd Iteration

Select All

The newly added Select All button selected every member displayed in the page. In the second iteration, by default, everyone is automatically selected in the database. As mentioned in the previous section, the page style was changed to match the format of the Search Repository function for better integration, which would be achieved in the next system iteration. A screenshot in Figure 62 gives a general idea of the Select All function.

Figure 56: Manage Members Select All - 2nd Iteration

Deselect All

The Deselect All button deselected every member who was selected. This provided the user with the flexibility of resetting the selection. Figure 62 above and Figure 63 clearly show the difference between the screen when Select All and Deselect All have been clicked respectively.

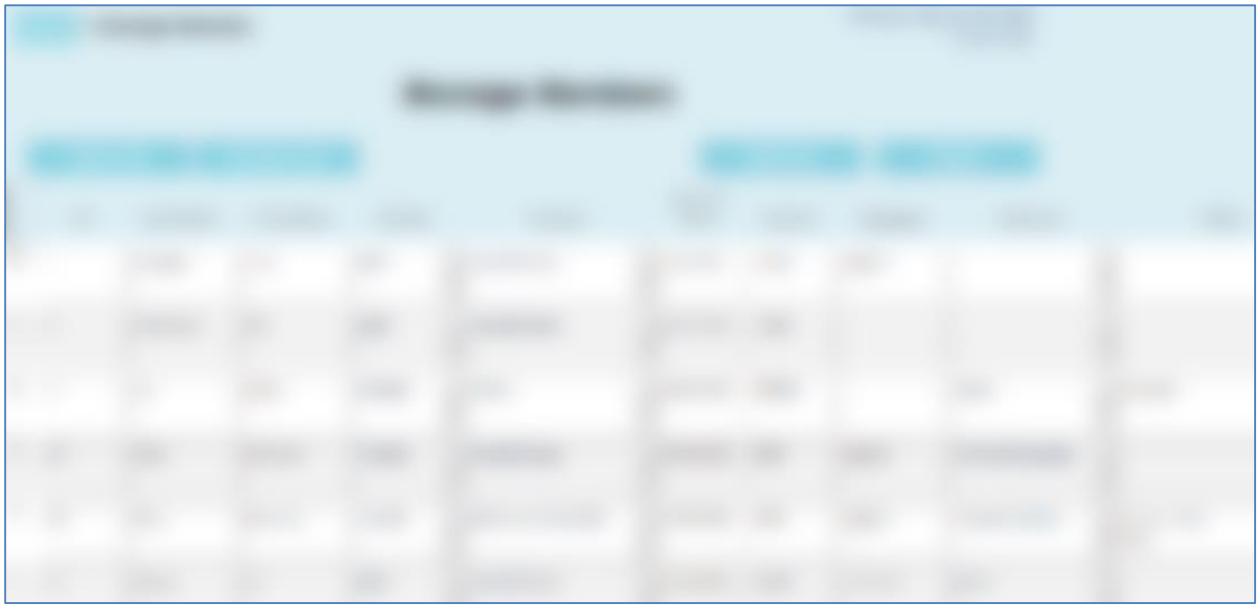


Figure 57: Manage Members Deselect All - 2nd Iteration

Remove

The Remove button on Manage Members page did not change compared to the first iteration as this function was already well developed. The Remove button, once clicked, would take the user to the confirmation page like shown in Figure 64.

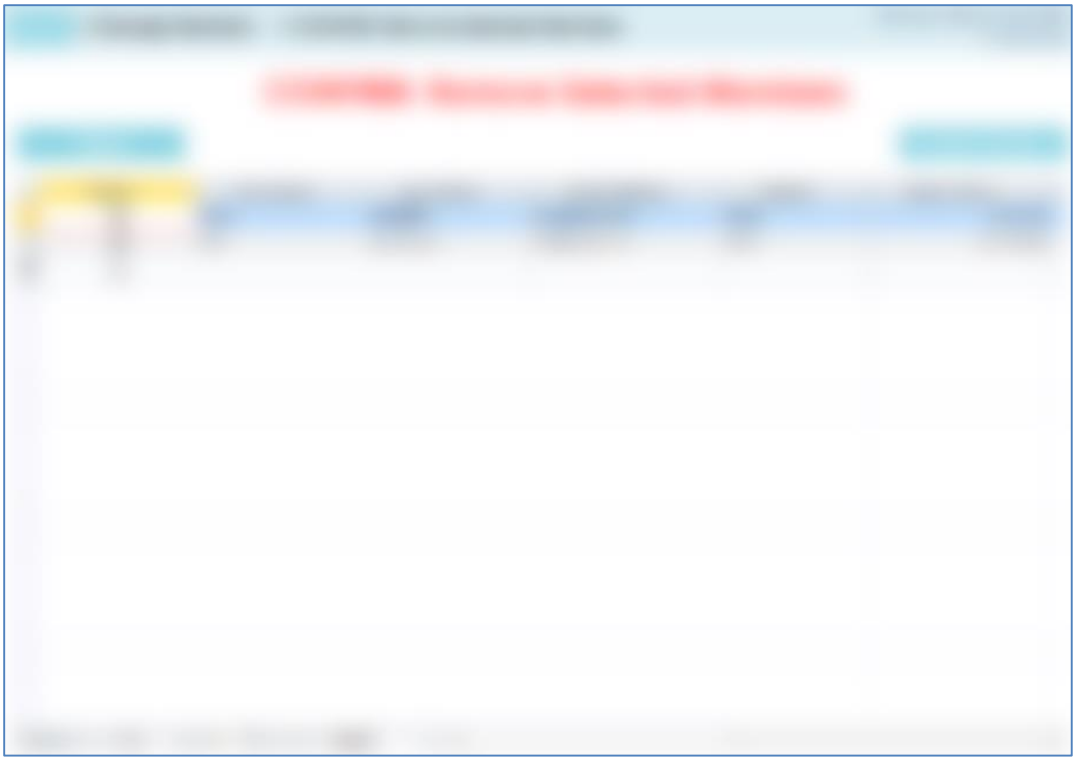


Figure 58: Remove Members Confirmation Page - 2nd Iteration

Export

Choosing the Export button would still take the user to a file directory popup window as shown in Figure 65. The Export function stayed the same as the first iteration since this function was also well developed.

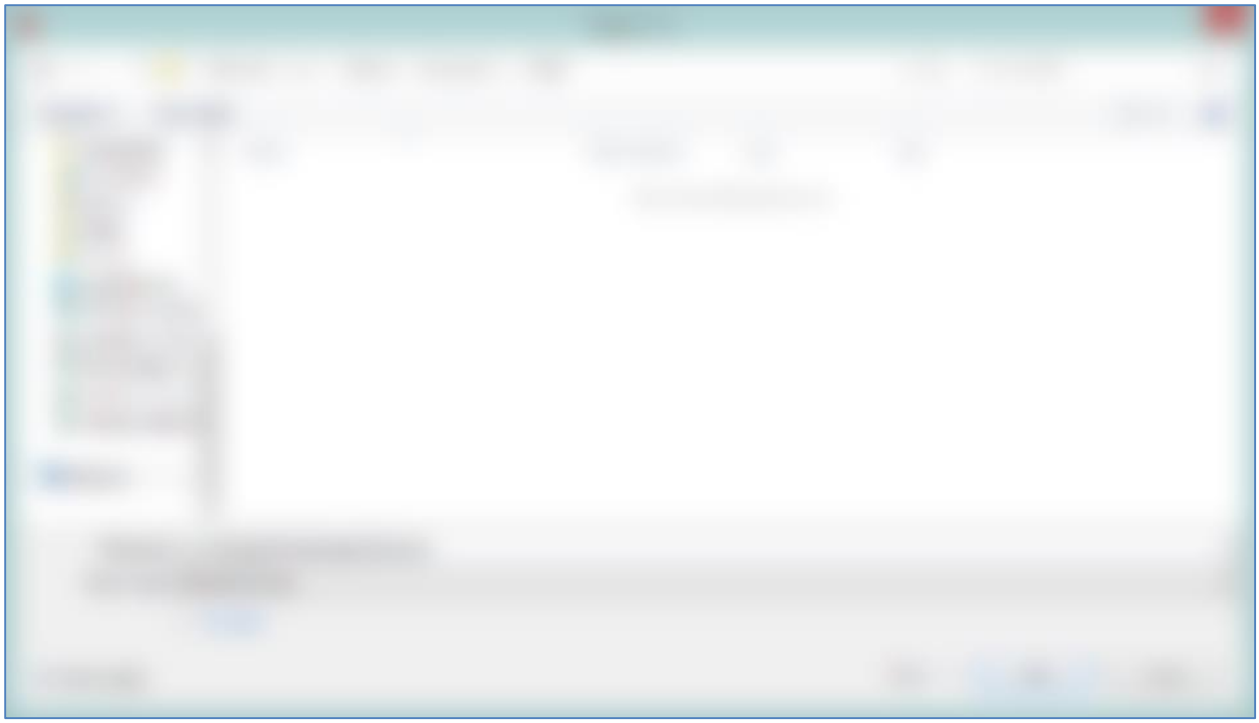


Figure 59: Export Members File Directory- 2nd Iteration

Reports

The Reports page, as of the second iteration, was still in an undeveloped stage shown in Figure 66. The team tried to run queries to display several different statistics in the system but Access was only able to show one visualized query at a time. The team continued its effort to achieve this function through alternative methods in future iterations.

Figure 60: Reports Page - 2nd Iteration

6.2.3. Third Iteration

The third iteration of the system was completed in mid-February. Again, the iteration incorporated feedback that the team received during the team's meetings with Mr. Burke

and Mr. Bergersen. The third iteration cleaned up the user interface and continued to expand system functionality.

Homepage

The team decided to add a "Help" button to the third iteration Homepage (Figure 67) based on the need for a user manual for the system. The Help button, once clicked, directed the user to a Help page with a user manual. Everything else on the Homepage remained unchanged.

Figure 61: Homepage - 3rd Iteration

Search Repository

There were several changes to the Search Repository page as of the third system iteration. First, the buttons, once at the top of the screen, were moved beneath the search boxes. Second, the "Manage" button was added, serving as a bridge between the Search Repository

page and Mange Members page. The Manage button, once clicked, directed the user to the Manage Members page, which was populated with the members from search results. This allowed user to manage only the members he/she intend to target from search. A screenshot of the Search Repository is shown in Figure 68.

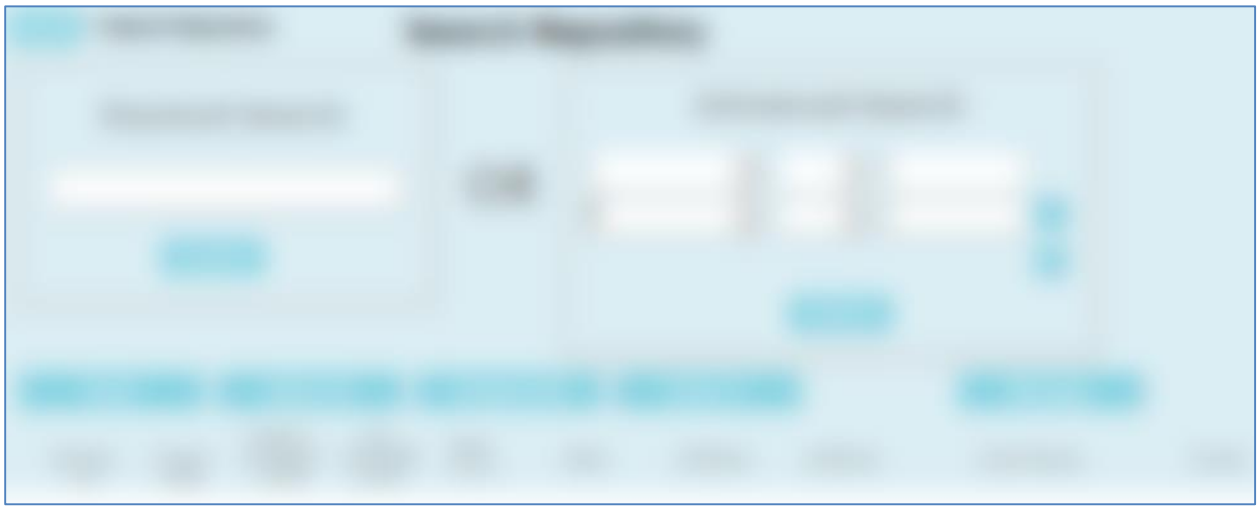


Figure 62: Search Repository Page - 3rd Iteration

Add Members

As in the first and the second iterations, the user had two ways to adding members: Add From File and Add Manually. There were no changes needed here at the homepage (Figure 69) for this function.

Figure 63: Add Members Options - 3rd Iteration

Add From File

After seeking alternatives to achieve Add Members From File function, the team was able to implement an approach using a Macro in Access, shown in Figure 70. The user was able to click the "Import from Excel" button and follow along with the instructions above it to import data. This macro worked by opening an Excel template and clearing it of all existing data. The user pasted in the appropriate member data to the template Excel sheet, saved it, and closed Excel. Then, back in the PMSS, the user clicked on the "Confirm Import" button, which appeared in a pop-up window, in order to finalize the import of the data to the system. This function worked well as an alternative but after showing this approach to Mr. Burke and Mr. Bergersen, the team was informed that there were previous attempts of similar function at Communispace that could be implemented to better achieve this system function.

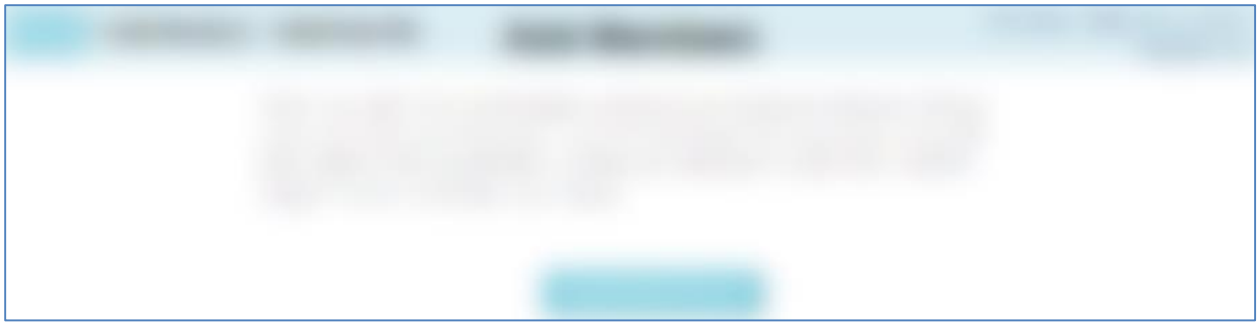


Figure 64: Add Members From File - 3rd Iteration

Add Manually

There were some design changes for the Add Manually function during the development of the third system iteration as shown in Figure 71. The Input fields are better aligned to one column for a feeling of consistency and to minimize confusion. The team also added several new profile fields as they became necessary. The status of a member could be "In Community", "Available", "Reserved", or "Do Not Contact". When inputting a member manually, the default value for his/her status would be "Available" as shown below. The Birth Year field was also changed from a specific date to a 4-digit year because there is little need to be as specific as a date for members' births. The team decided that several fields needed to be required when entering data manually, and such fields were marked with an asterisk after the field title.

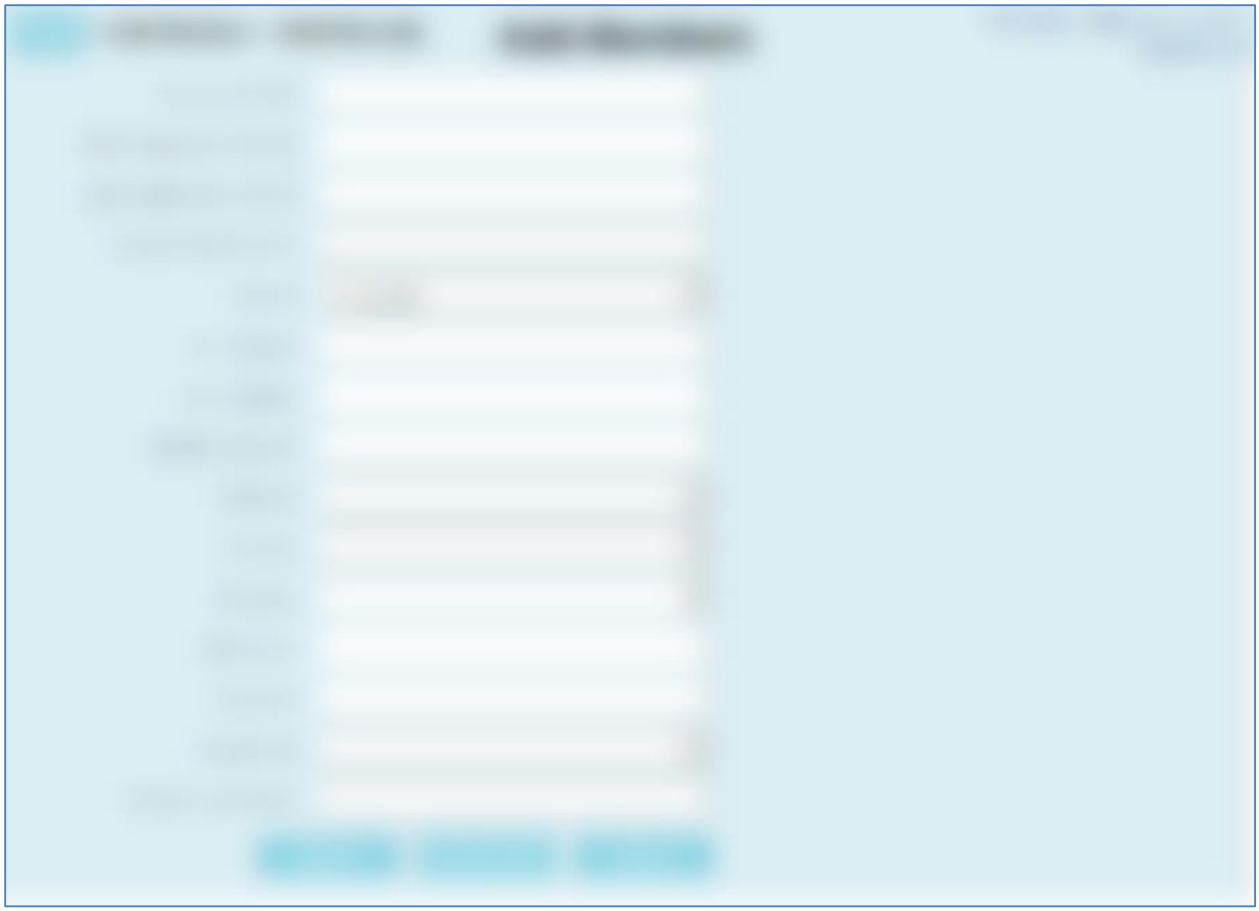


Figure 65: Add Members Manually - 3rd Iteration

Manage Members

The Manage Member was renovated by refining the layout and redesigning the buttons. The page featured six buttons: Reset, Select All, Deselect All, Select 'n', Update Status(es), and Export as shown in Figure 72. After discussion of the previous iteration, the team deleted the "Remove" button, instead opting to allow users to have a status of "Do Not Contact". In this way, members' data remains in the system even if they do not wish to be a part of communities. The buttons were better aligned as well to improve user experience.

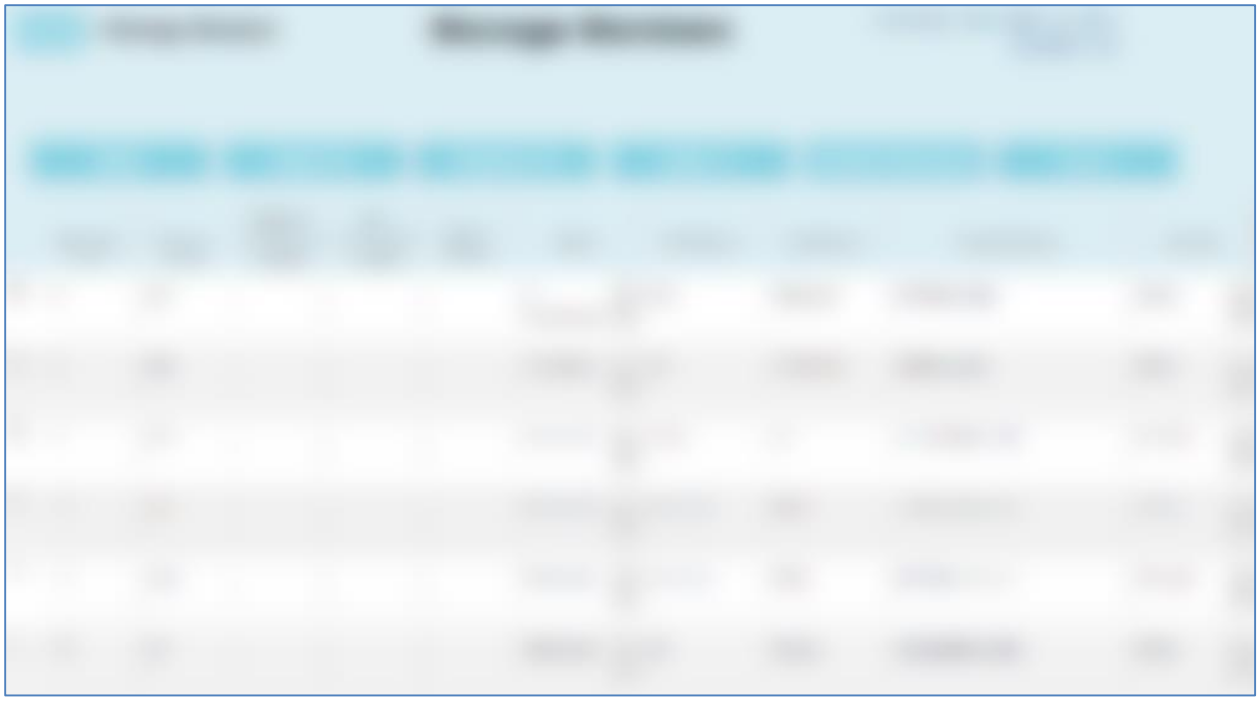


Figure 66: Manage Members - 3rd Iteration

Reset

The Reset button cleared the filtered results from search as well as deselected the selections whenever the user finds it necessary. This allowed the user to clear any accidental actions in the system. A screenshot of the Reset button is in Figure 73.

Figure 67: Manage Members Reset – 3rd Iteration

Select All

The Select All button worked in the same way as in the first and second system iterations. It simply selected all users displayed in this page. However, since the Search Repository function and Manage Members function were linked during this iteration, the Select All button selected every member out of the filtered results from search. An example is shown in Figure 74.

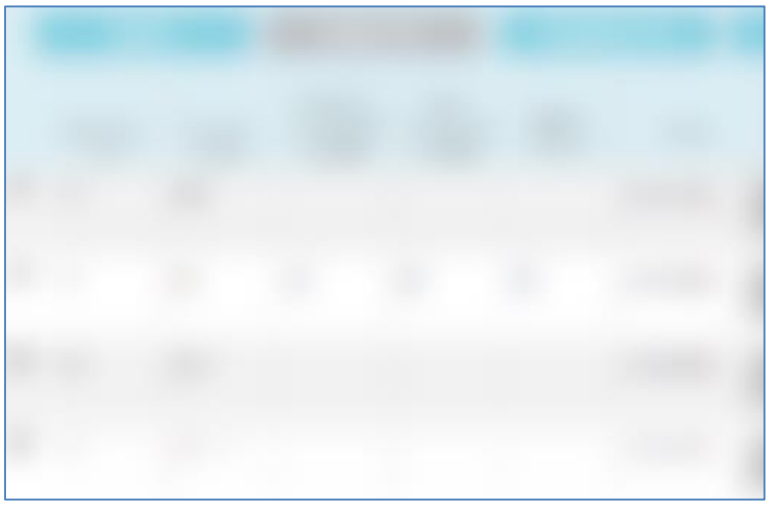


Figure 68: Manage Members Select All - 3rd Iteration

Deselect All

The Deselect All button, similar to the Select All button, did not change compared to previous iterations. An example shown in Figure 74 above and Figure 75 gives a clear representation of this function of deselecting the selected members.

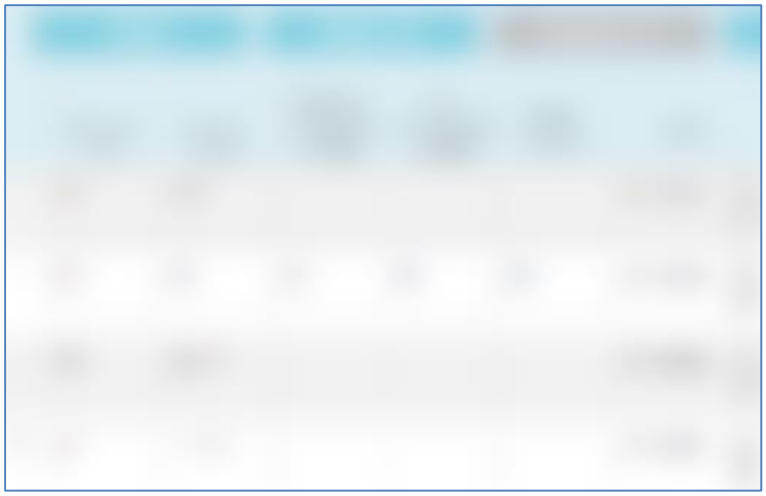


Figure 69: Manage Members Deselect All - 3rd Iteration

Select "n"

After the second iteration, the project sponsor requested the feature "Select 'n'" in the Manage Members page so users could easily manage a certain number of members for reuse/analysis. Once the Select "n" button was clicked, a pop up window asked the desired number of members the user wanted to select. After a valid number was inputted, clicking OK or simply hitting the Enter key on the keyboard selected the number of members the user wanted. An example of selecting three members is shown in Figure 76 and 77.

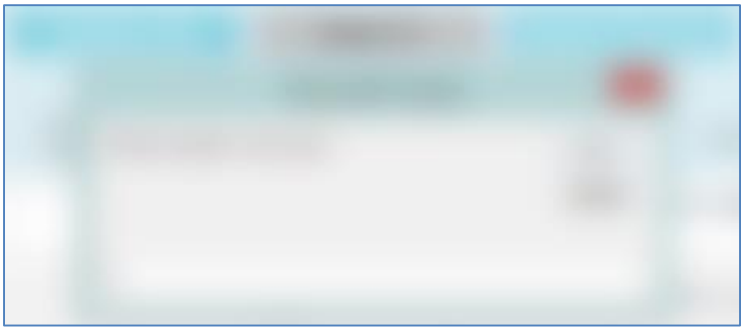


Figure 70: Manage Members Select 'n' - 3rd Iteration

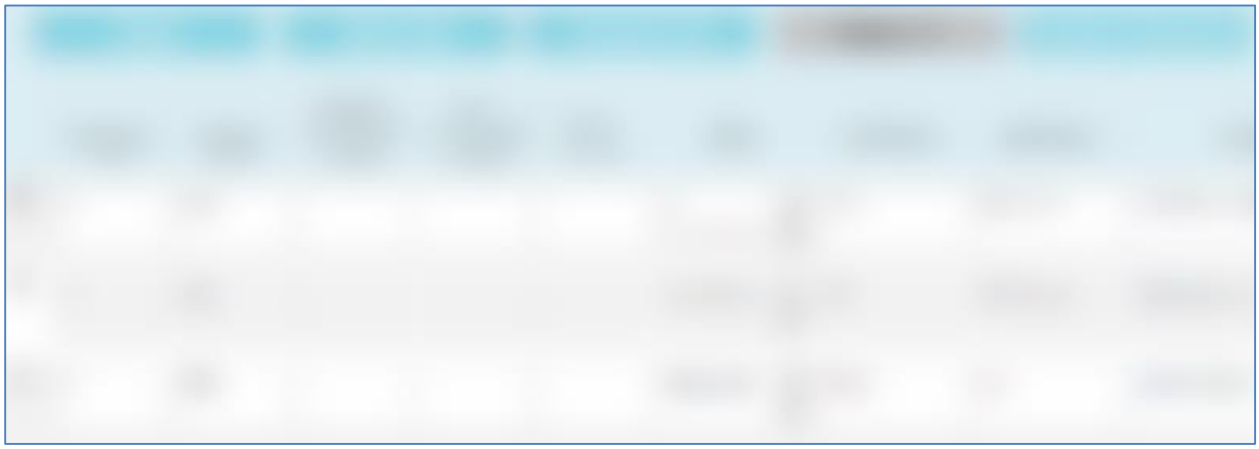


Figure 71: Manage Members Select 'n' Results - 3rd Iteration

Update Status(es)

The Update Status function (Figure 78) allowed user to update members' status from "Available" to "Reserved" to prevent members from being reused for multiple purposes. The reverse also worked for allowing the members to be "Available" again. Another status, "Do Not Contact", was available for rare cases in which a member does not want to be contacted again by Communispace.

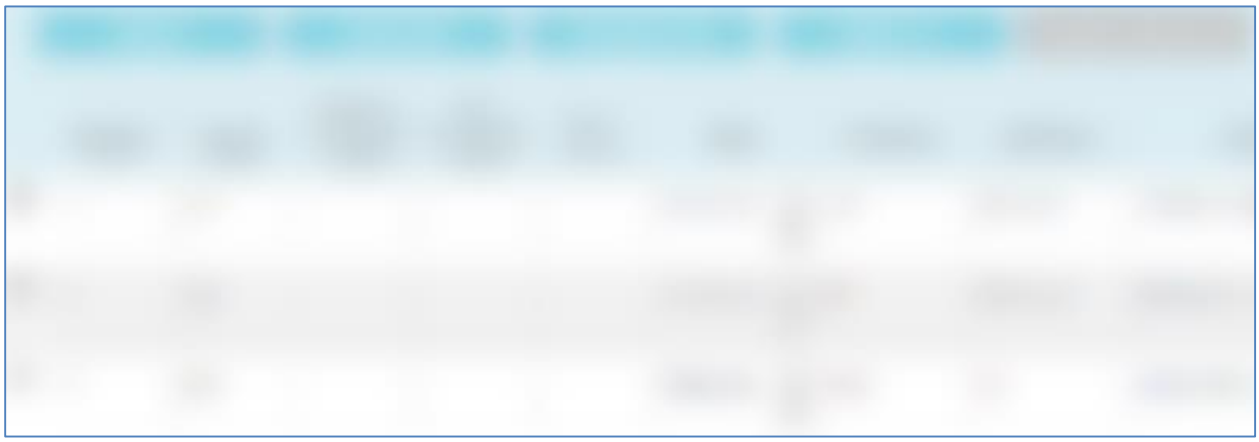


Figure 72: Manage Members Update Status(es) - 3rd Iteration

Export

The Export function remained the same as the previous iteration since it was a finished function as shown in Figure 79. This function allowed a user to easily export members' data for business purposes.

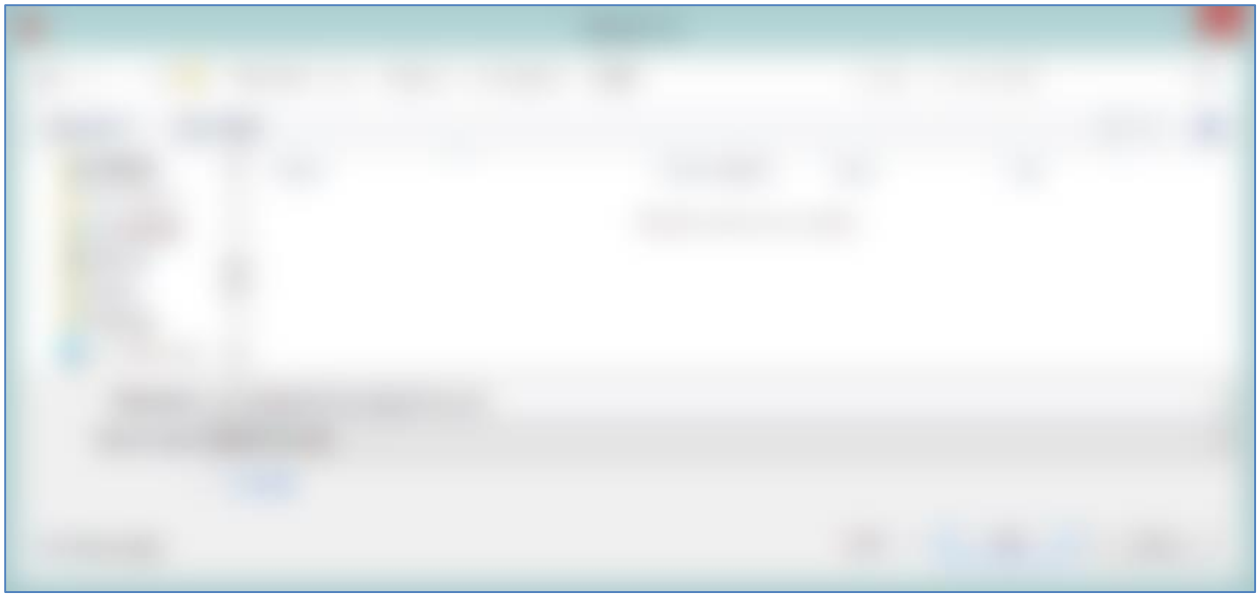


Figure 73: Export File Directory - 3rd Iteration

Reports

The Reports page was finally added during this iteration, featuring some visualized statistics of the system including member status distribution, member gender distribution, member language distribution, member birth year distribution, and member ethnicity distribution as shown in Figure 80. As of this iteration, the Reports page had a few small bugs to fix; for example, the “Slice 1” labels counted the number of blank cells and the Language labels showed numbers instead of countries because of the way countries were referenced.

Figure 74: Reports - 3rd Iteration

Help

The Help Page, as of the third iteration, was roughly designed as a dialog box with placeholders for different contents as shown in Figure 81. The team later altered this page and decided to not have the FAQs and Contact System Admin function as these weren't very necessary. However, the System Manual was a high priority and was fully implemented in the final iteration of the system.

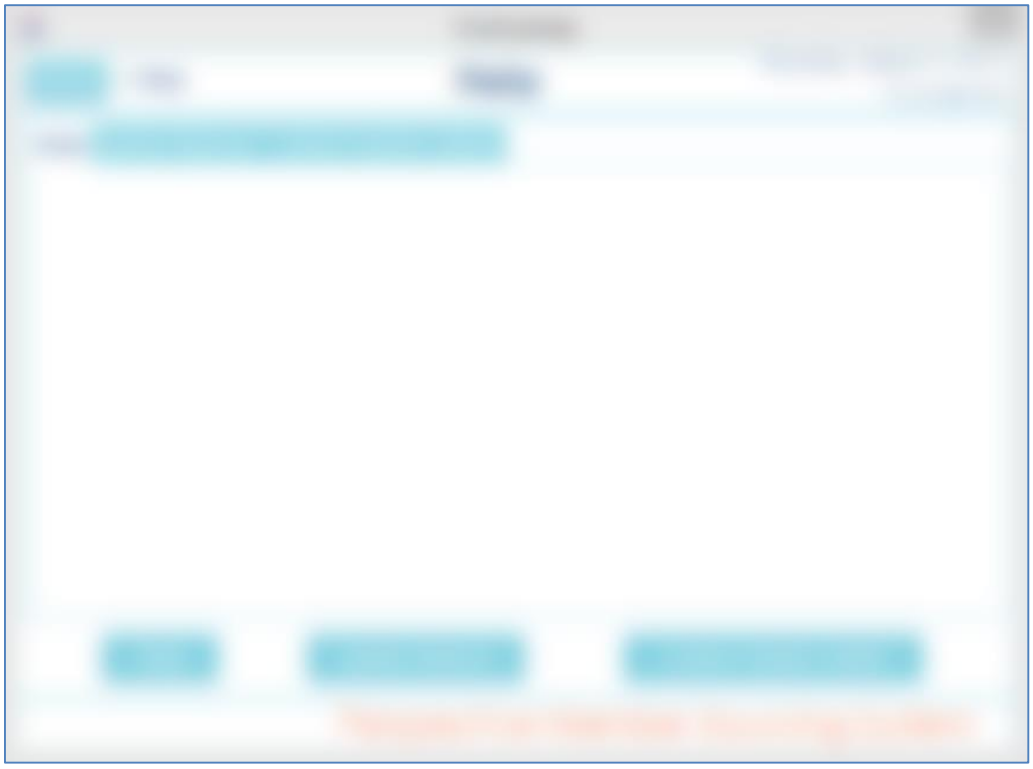


Figure 75: Help Page- 3rd Iteration

6.2.4. Final Iteration

The final iteration of the system was developed at the end of February and included a few minor changes in the first week of March. This iteration included feedback from Mr. Burke and Mr. Bergersen as well as feedback from the team's final presentation to all major stakeholders. It should be noted that the final iteration of the system is still a prototype and is not a completely finished product. There were still improvements to be made to the system, many of which are outlined in a later chapter of this report. However, the final iteration includes all of the requested major functionalities of the system.

Homepage

The Homepage (Figure 82), as of the final system iteration, did not change as the functions and design of this page was fairly mature after going through three iterations. The team decided to keep the simplicity of this Homepage as well as few functional buttons.

Figure 76: Homepage - Final Iteration

Search Repository

The Search Repository page experienced some final changes during the final iteration, mostly relating to the Advanced Search feature. The UI was finalized as shown in Figure 83. All buttons functioned in the same way as in the third system iteration and were evenly sized and aligned. The Search Repository also went through some final revisions for VB code.

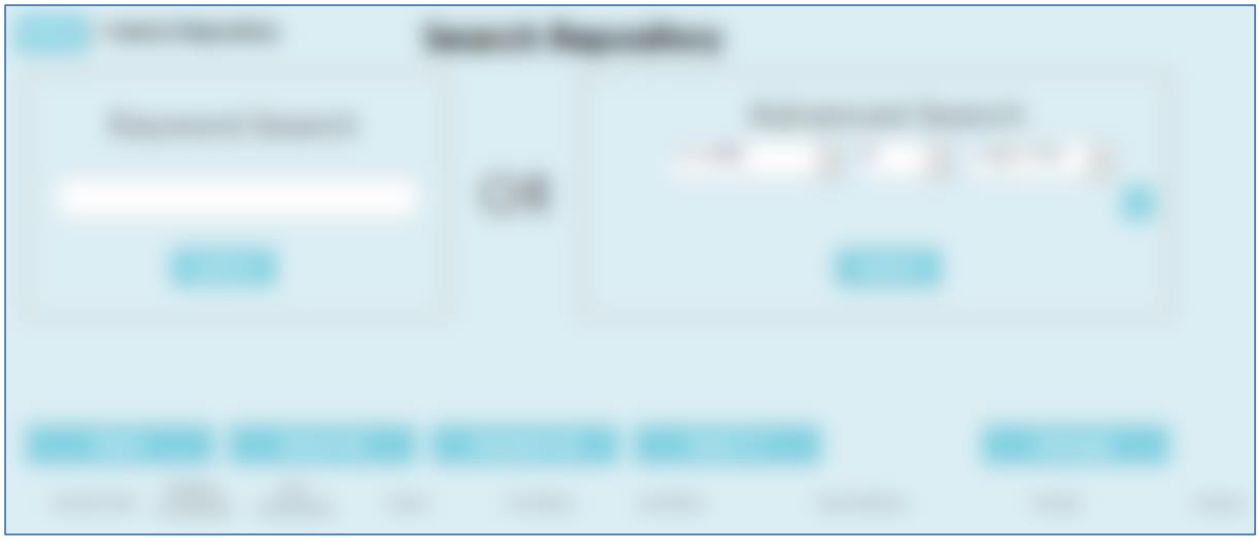


Figure 77: Search Repository - Final Iteration

Advanced Search

After reviewing the third iteration, the team decided on several changes taking into account all feedback. Those changes included increasing the number of advanced search parameters up to five, adding an “and” and “or” logic for each of the definitions, finalizing logic for each of the input fields, adding in the logic of resetting parameters once the “minus” button was clicked, and adding animated interaction when the user changed search parameters. As of the final iteration, all required functions for Search Repository were implemented and refined and can be seen in Figure 84.

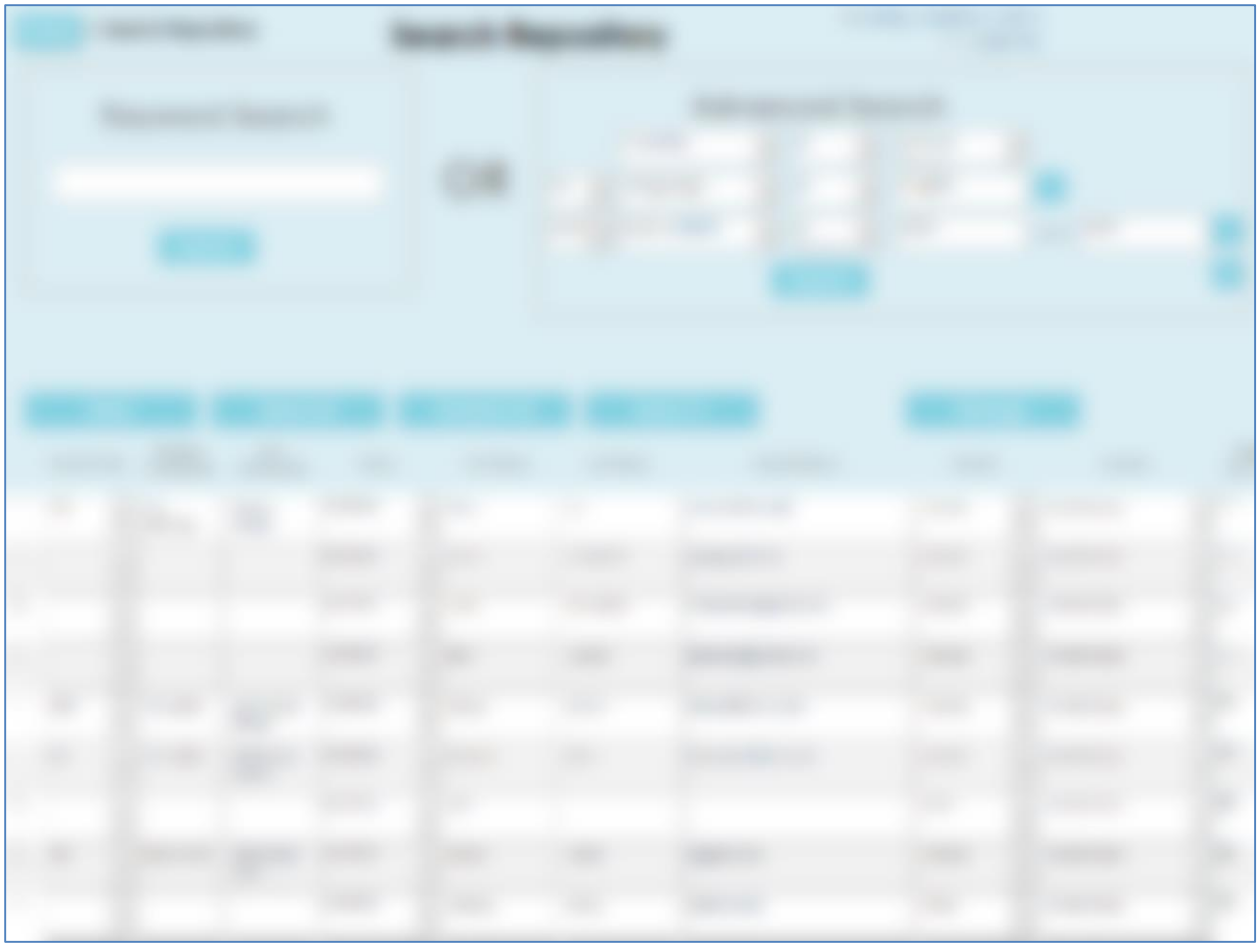


Figure 78: Advanced Search - Final Iteration

Add Members From File

By learning from previous Communispace practices, the Add Members From File function was finalized as shown in Figure 85. It allows the user to open up a Member Data Report (MDR) file by locating it with the file browser. The system matched fields automatically based on their field names and the user manually matched the fields that were not automatically matched using “Associate Selected” button. This process could also be undone by clicking the “Unmatch Selected” button. The Add Members From File function

was able to be finalized to work well with existing Communispace data files by the final iteration.

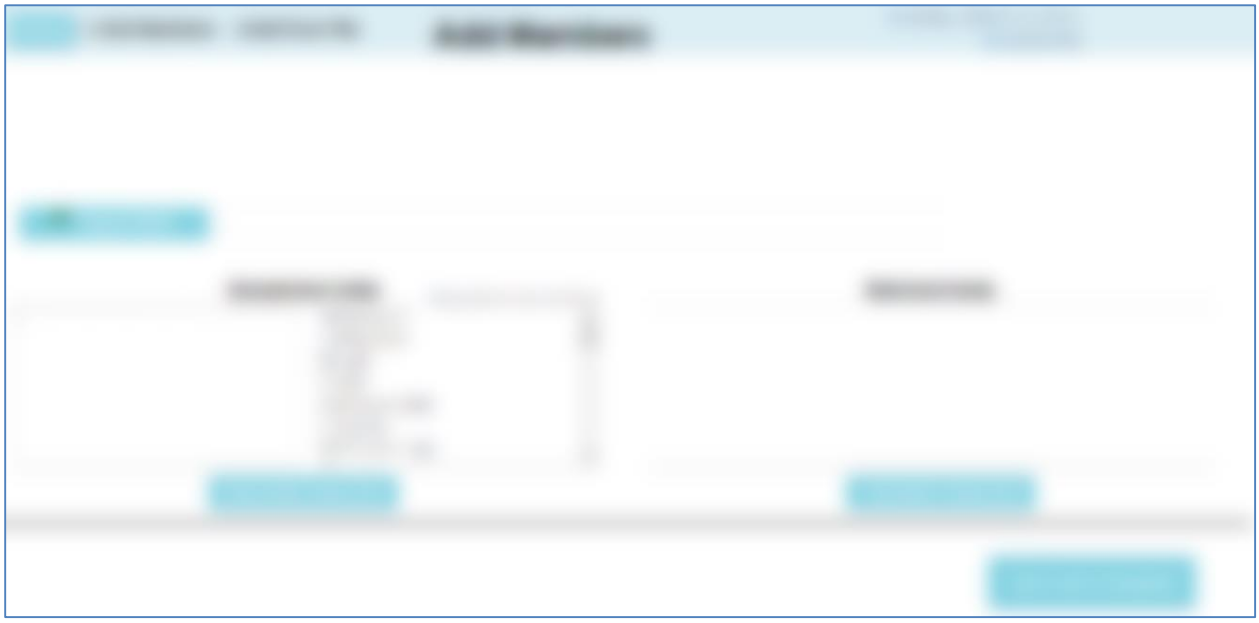


Figure 79: Add Members From File -Final Iteration

Add Members Manually

The team went through a very restrict revise for the final iteration and decided to perform a major reconstruction in terms of UI for the Add Members Manually page. As shown in Figure 86, every profile field was no longer the same length; instead, each had a corresponding input type and length so that the user could easily input member data manually. This improved the user experience of this function by a significant amount.

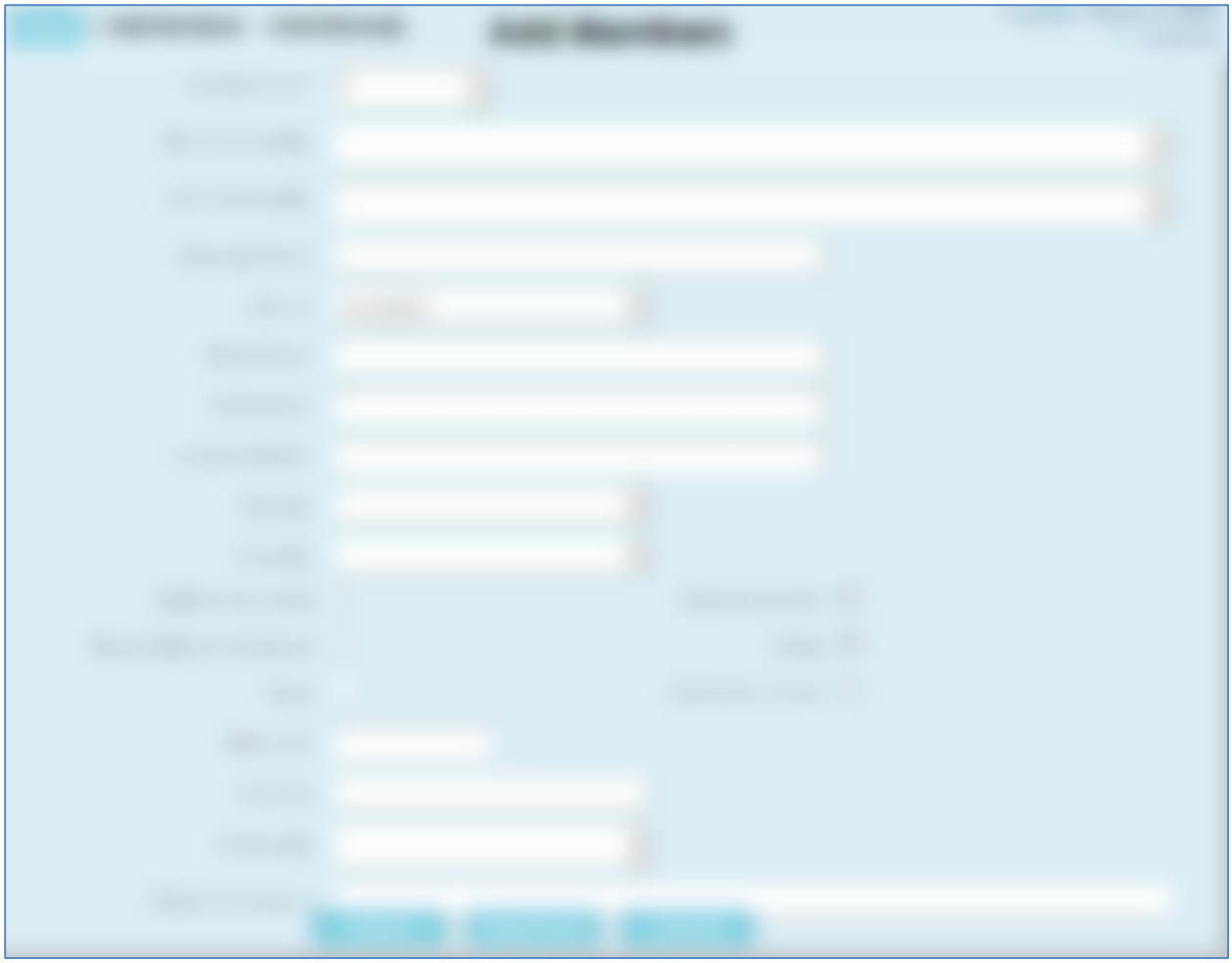


Figure 80: Add Members Manually - Final Iteration

Manage Members

The Manage Members page, as it was already well developed during previous iterations, did not experience many changes in the final iteration, which is apparent in Figure 87. However, there were some coding refinements performed on the page. All buttons worked the same way they worked in the third iteration and the integration with the Search Repository function worked well as it was supposed to.

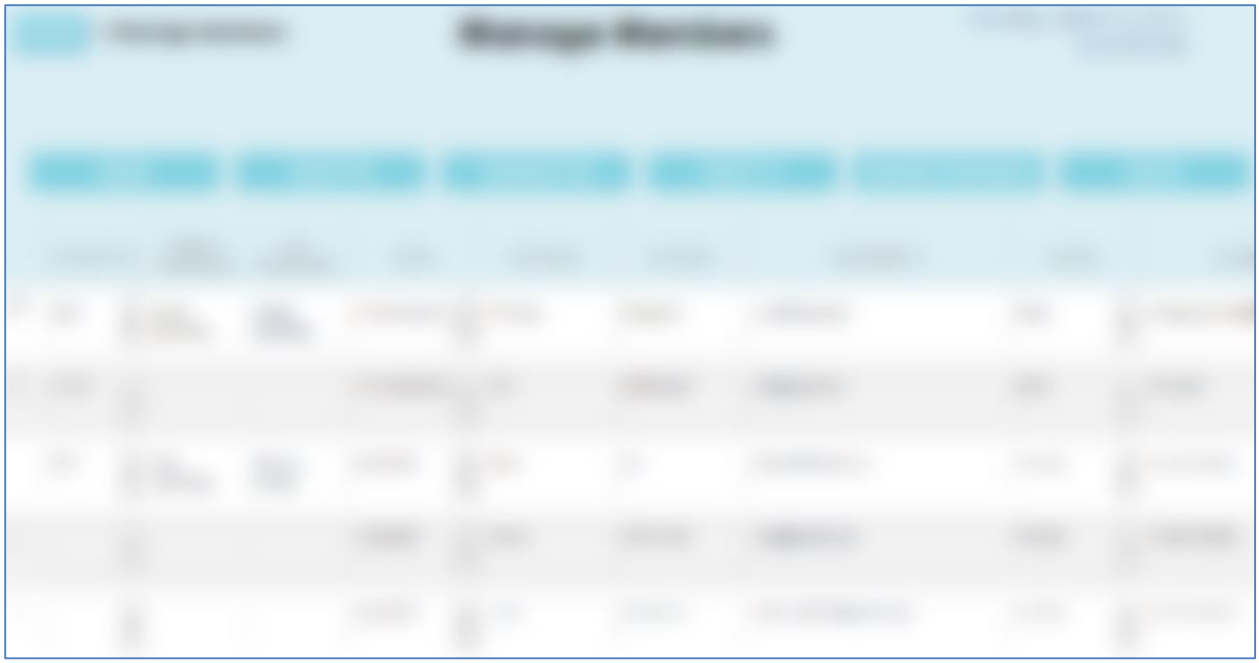


Figure 81: Manage Members - Final Iteration

Reports

The Reports function was refined as well for the final iteration. The Slice 1 for blank fields issue was resolved and some general bug fixes were performed as well. The Reports page gave the user a high level overview of some system statistics including member status distribution, member gender distribution, member language distribution, and member birth year distribution as shown in Figure 88.

Figure 82: Reports - Final Iteration

Help

The Help button, once clicked, directed the user to a PDF file of the system manual (Figure 89) which included all major use scenarios for the system and clear explanations of each of the functions in the system.

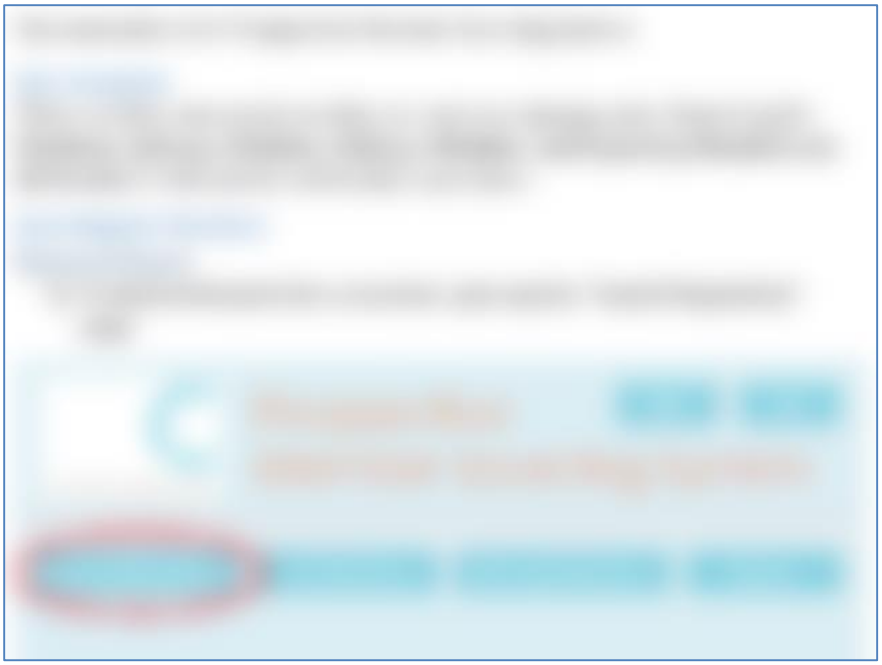


Figure 83: System Manual - Final Iteration

6.3 Testing

After components of the system were completed, the team tested the specific components to verify that their functionality aligns appropriately with the requirements and their predetermined purpose. Testing was conducted by two main groups: the WPI MQP and Client and Consumer Services teams. The MQP team tested all the system functionality multiple times throughout system development. Once the core functionality was complete and the MQP had ended, the Client and Consumer Services team tested the system. This was the most important group for testing because the Client and Consumer Services team will integrate the system into their operational processes on a day-to-day basis.

The MQP team planned to apply multiple types of testing to the system, including unit, integration, system, and user acceptance testing. After the team tested each *unit* of functionality, the team identified and squashed any bugs that existed in the code (unit testing). Next, once the team confirmed the individual modules worked correctly, they moved on to integration testing, making sure the modules functioned correctly together. For example, the team tested the connection between the Search and Manage Members pages to make sure the search result data could seamlessly carry over from one to the other. System testing was briefly conducted to verify that the system meets the business, usability, and performance requirements. However, the team was unable to test performance under a heavy load, which is unlikely to be an issue based on the expected usage.

Unfortunately, other priorities took precedence and there was not enough time to perform comprehensive user acceptance testing of the system, but after reviewing it with Mr. Jack Bergersen and Mr. Sean Burke the performed user testing was deemed sufficient. Testing was an important process for identifying improvement opportunities in the system and subsequently implementing the improvements in system revisions.

6.4 Population of the Database

In order for the Client and Consumer Services team to migrate to the system, it needed to be seeded with relevant data. The team accounted for this population of the system's database in the implementation phase so the users could begin using the system almost

immediately. In order to seed the data, the team used the Import From File function to add information from several MDRs to the PMSS. The MDRs came from communities that were sunsetted in the past few years. This allowed the team to quickly seed the database and make it ready for use. The ability to populate the PMSS with relevant data so quickly was a valuable asset to the team.

6.5 Conversion Plan

Within the scope of the project, it was not possible to supervise the migration to the new system, which is why the team created a support plan and documentation to help the migration run smoothly. In order to make the system fully-functional, it must undergo a conversion into Communispace's existing technical and business processes. The technical processes allow the system to integrate into the existing SQL servers and Excel MDR files. Having the most up-to-date information will make the system the most effective at importing and exporting data. This in turn will drive the business processes. Users in Communispace's Members Services and Client Communities will import and export the live data to make a self-sustaining prospective member sourcing system.

The first step in converting the technical processes is to create a permanent location for the system and establish connectivity. Currently, the system is only linked to local tables. This was purposefully done as to not disturb any live connections with Communispace's SQL servers. Establishing the live connection may be easy, but certain business logic must be applied. As mentioned in the Recommendation Section (7.1), the system must display only PCIDs that Communispace can use. PCIDs that are not suitable must either be hidden or not

stored altogether. The same technical logic must also work for the different Status fields. See the Recommendations Section (7.1) for more information.

The second step is to establish the business logic that will work along with the technical process. Users will need to import information to make the system more valuable. Over time with growth, the system will gain more value for Communispace. At first, the users will spend more time importing data, rather than searching and exporting data. As time continues and the system grows there will be more members and the users will be able to export members for their own use. As Communispace grows, their business process may change and the system will need to adapt.

6.6 Support Plan and Documentation

Throughout the project, the MQP team has documented its work, which will be used as support documentation for the system. The documentation has two parts: technical and user documentation. It also covers detailed explanation of the underlying algorithms behind the code as well as instructions on how to call and extend the functions. The technical documentation contains an overview of the code structure and function groups. The complete technical documentation for PMSS can be found in Appendix J. This documentation will be helpful for future programmers that intend to extend the system or migrate the system to a different environment. This is especially applicable since there may be a future need to integrate with Catalyst. The code sources themselves are also well documented with comments.

Besides the technical documentation, the team also created dedicated user documentation for troubleshooting the system in the event it is needed. This user documentation provides a comprehensive overview of the system's functionality as well as steps which walk users through common use scenarios in a tutorial-like format. The user documentation can be found in Appendix I. An embedded help menu and useful tooltips are also implemented within the system to facilitate the learning process.

System documentation is very important for successfully transitioning the system to the Client and Consumer Services team and was a focus of the MQP team at the end of the project. A support plan for the system was also detailed, including the potential for future WPI MQP teams or Communispace employees to expand upon the system in other projects, because the team will not be able to continue its contribution after the conclusion of the project. The System Documentation can be found in Appendix I.

6.7 Training Plan

Before new users start using the system, they can be briefly trained on its functionality and different purposes. As with any system, there is a learning curve before a user is completely comfortable integrating it into their day-to-day processes. To ease this transition, the team compiled comprehensive training documentation, which runs through different use cases in a tutorial-like format. This documentation can be referenced in Appendix I.

7. Recommendations and Conclusions

Based on the WPI MQP team's comprehensive research and on-site development over their twenty-one week MQP, this section provides the Client and Consumer Services team at Communispace with actionable recommendations for future development and implementation. In addition to detailing recommendations, the team also outlines their lessons learned and conclusions from working on the project experience with the Client and Consumer Services team.

7.1 Recommendations

While these recommendations did not fit within the project scope due to their relative complexity and low prioritization, they are derived from various stakeholder meetings and research, and would be valuable extensions for the system to capture. The recommendations mainly focus on implementing logic into the system backend and include system business logic, tighter integration with external data sources, and data normalization.

7.1.1. Business Logic

While the project team was able to implement some business logic into the system prototype, more work can be done to streamline the system so that it is aligned with the

Client and Consumer Services team's business processes. The team integrated business logic in the context of exporting members from the database, where their status changes based on the recruitment process. Employees managing the member recruitment process can update member statuses in the repository to reflect their internal recruitment status: Available, Reserved, In Community, or Do Not Contact. This type of business logic can develop further throughout the system, making it more valuable for Communispace. For example, the PCID field (source of member) can take into account if the member is from a client list, panel, or social media placement, and then display that member to users accordingly. Even though the team did not have enough time to implement all the complex logic that powers the recruitment process, the system was designed for ease of extensibility.

7.1.2. Tighter Integration with External Data Sources

In order to make the system prototype more useful, the team integrated external data sources from SQL Server that link it to systems currently used by the Client and Consumer Services team. The current links, to the Vendor Performance Relationship Management and Dupe Check systems, are a foundational step to incorporating the Prospective Member Repository into the technological stack the recruitment team uses on a day-to-day basis. In the future, the system can link to other external sources, making the connection more seamless between everything.

7.1.3. Data Normalization

As businesses grow there are opportunities to streamline processes and make them more efficient. In the context of the recruitment process, member data can be normalized to make it more useful for everyone that uses it. The team attempted to look into data standardization at the end of the project, but was unable to spend an adequate amount of time working on it because of various constraints. An example of normalizing data to make it more useful is in regards to member income, a field that the team attempted to work on.

Currently, member income is captured in a text field that contains highly variable ranges, where the increments differ from community to community, making it difficult to query. Normalization either in process, where the increments are standardized, or in the system, where it parses and standardizes data on its own, could be very valuable to the end users. For example, search results across income would return all users that fit the criteria more accurately than before. Overall, data normalization in the system, whether across income or another field, helps make existing data more valuable to the end users. The team did include a recommended algorithm to implement normalization for the income field in their technical documentation, which can be referred to in Appendix J.

7.2 Project Presentation

The team presented the Prospective Member Sourcing System (PMSS) prototype to several stakeholders on February25, 2015. At the conclusion of the presentation, there were several discussion points, which are detailed in the following section.

7.2.1. Scalability and constraints

One concern that was mentioned by Mr. Slava Asipenko was the scalability of the PMSS. This was of particular concern for him given the development in Microsoft Access. The team explained that the intention was to have the tables in the database stored on Communispace's SQL Server. The SQL Server backend ensures that the PMSS will not be hindered by the addition of thousands of records. Mr. Jack Bergersen and Mr. Sean Burke agreed that the Client and Consumer Services team has several other systems stored on the SQL Server that each contains tens of thousands of records. Unfortunately, the team was unable to test the capacity of the PMSS but remains assured that the system will scale well for the foreseeable future.

7.2.2. Data Normalization

Another discussion point was poised by Mr. Dave Rosenberg, who inquired about the issues surrounding data normalization. Everyone present agreed that data normalization was a problem company-wide and beyond the scope of the project. However, the team did offer up their observations on the subject; the team recognized that Communispace was a growing company that collected large amounts of data. The team felt that the most sensible albeit difficult path would be to standardize data across all business processes including screener surveys (ConfirmIt), communities (Catalyst), and additional applications. Though this would be a daunting task, it would provide Communispace with many benefits, especially in light of the company's continued growth. As far as the Potential Member

Sourcing System, the team attempted to normalize fields where possible (i.e. Country, Gender, and Language) but left some fields (including Income) without normalization.

7.2.3. Integration into daily processes

One concern that was brought up by Mr. Jack Bergersen was the need for a conversion or integration plan in order to help everyone understand how the PMSS would eventually be rolled out. The team had not spent much time considering this but in the week following the presentation, a conversion plan and training plan were written in order to aid the roll out of the new system in the future. These plans can be found in 6.6.

7.2.4. "Notes" field

Following the presentation, Mr. Dave Rosenberg was curious about the "Notes" field and which fields from the MDRs would potentially map to it. The team had not previously considered this very thoroughly, so there were several minutes of discussion in order to devise the best plan of action, which was the creation of another field. This new field, entitled memberBlob, would collect all of the fields from an imported Excel sheet in a blob- or json-like format, concatenating the headers to the fields, and separating each with a pipe delineator (|). In the week following the presentation, the team added the memberBlob field to the system. Unfortunately, there was not enough time to fully develop the import of all fields, but the team was certain to set up the system to be able to handle this extension in the future.

7.3 Lessons Learned

As with any project, there are constraints that affect the way the project is completed. The team ran into two main constraints during the project: timing and scope creep. Time was a limiting factor during the project's completion because it determined the level of detail the team could focus on during each phase. The project timeline officially covered 21 weeks, but was somewhat less because of external factors. Also, when certain phases took longer than expected, others had to be shortened to fit within the timeline. For example, the team was unable to test for the planned amount of time because system development took longer than initially expected. The lesson learned from working with a shifting timeline was to plan realistically and focus on accomplishing the core of the system with what remained.

Change in project scope was another constraint for the team. As various stakeholders were informed about the team's progress, more requirements were sometimes appended to the project's scope, causing it to become more complex. The team realized that everything was not going to be accomplished in the system prototype, nor should it be, and instead focused on only the most important requirements that would serve as the system's foundation for future development. Related to project scope, there was a fair amount of ambiguity when the team started the project. Because of the ambiguity, the team had to be cautious to not add too much to the prototype since this was the first pass at the system.

7.4 Conclusions

Throughout the team's work on their MQP, their objective was to work collaboratively with Communispace, identify their business needs in sourcing and recruitment, and develop a system prototype that provides value to Communispace, specifically the Client and Consumer Services team. In accomplishing this objective, the team synthesized their four years of academic experience and applied it towards a real-world experience. Since the project was conducted over a 21-week period, it was important for the team to stay focused on the alignment between business and information technology because the resultant prototype was handed off to them at the end.

Although originally focusing on three components related to the recruitment and sourcing of members: overflow capture, alumni member retention, and ongoing recruitment, the team ultimately worked almost exclusively on the first two components. While ongoing recruitment was not explicitly worked on since it was determined to be out of scope, the team felt as though the system could be used as an auxiliary for any specific ongoing recruitment processes. Overall, the team captured all of the foundational business requirements in the system prototype that was delivered to the Client and Consumer Services team.

In completing their MQP, the team felt that their Systems Analysis and Design course prepared them very well for the process that was followed throughout the project. Even though prototype development began later than expected, the team understood that it was

important to accomplish a lot ahead of time, including writing this project report and understanding Communispace's business model, which allowed them to accelerate their development process later on.

Overall, the project experience taught the team a great deal about managing complexity and project scope, in addition to real-world project management and system development. The benefit of the capstone-like project experience was that it allowed the team to apply and adapt their theoretical knowledge not only during their planned processes, but also when unexpected events occurred. Including the aforementioned lessons learned, the team learned a lot about how projects actually work inside a growing company, which is an important concept for the team to understand in their upcoming careers.

Appendix A: Systems Request

[THIS SECTION HAS BEEN REDACTED]

Appendix B: Additional Literature Review

This appendix provides additional information from our literature review, specifically relating to IT and Business Alignment.

Does IT Matter?

Does IT matter? Nicholas Carr's article "Why IT Doesn't Matter" published in May 2003 Harvard Business Review proved his point that information systems had become a commodity. That had been true for a while at the time being, and even now in a lot of the cases. However, the fact that IT generally doesn't offer a high level of competitive advantage doesn't mean that IT is not important. In fact, IT is enabling a lot of the fundamental works within organizations nowadays. With good IT, the efficiency that can be improved, the transparency that could be raised, and the opportunity either internally or externally that might be taken advantage of are just limitless. Therefore building a good IT within an organization is becoming more and more essential.

Strategic Alignment Maturity

Luftman, in his book Strategies for Information Technology Governance, introduced the model of strategic alignment maturity which consists of 6 parts: partnership, scope and architecture, skills, governance, communications, competency/value measurement. This section introduces how this model could enhance an organization's IT Business strategic alignment. Communications, Competency/value measurements, governance, partnership,

scope and architecture, and skills together build an IT Business Maturity model. Every single one of the criteria could affect the alignment in its unique aspects. The Figure 90 gives a visual explanation of the model.

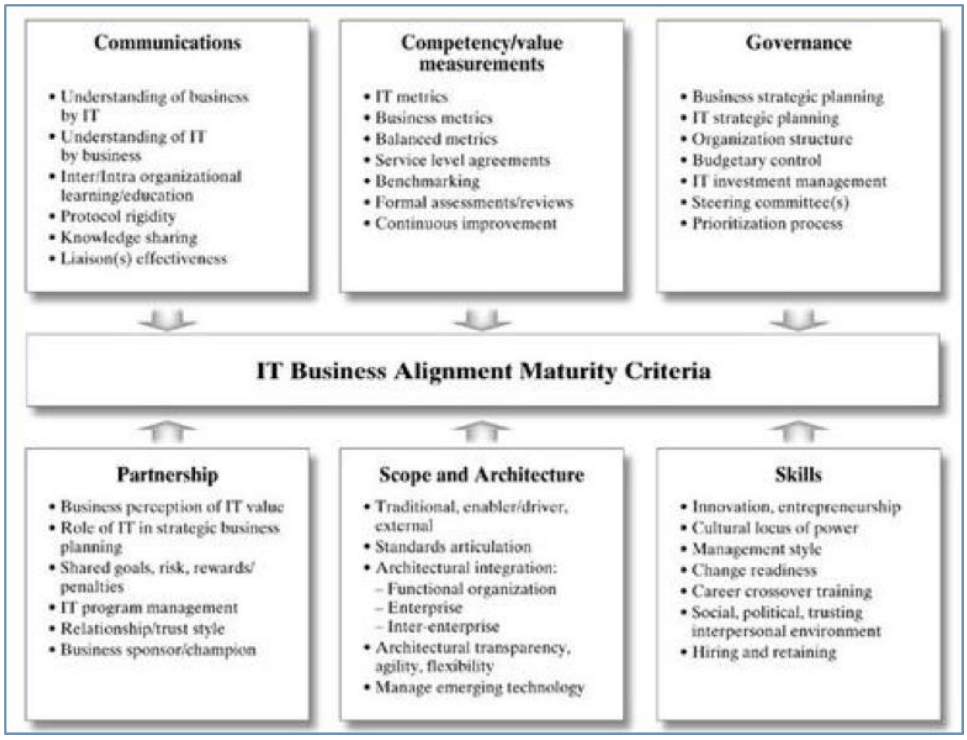


Figure 84: IT Business Alignment Maturity Criteria (Luftman, 2000)

Communication

IT could maximize its value by effectively working with the other divisions through good communication.

IT can help improve the communication efficiency within an organization significantly by providing fundamental support of IT tools. Beyond that, it could also bring additional internal value by providing information such as transparency, evaluation, and simulation. These values would lead to a better collaborative effort and more efficient communication across different functional areas.

On an organizational scope, communication across different levels of functional divisions is also essential. Generally there are three levels within an organization: strategic level, tactical level, and operational level. The most effective approach towards good communication is to establish a shared understanding. Once people from different departments/levels with various backgrounds reach a shared understanding, there will be less cost, either time wise or financial wise, wasted through communication. More efficient communication gives people a better opportunity to collaboratively do work together, not spending time talking and debating while speaking different languages.

Governance

"IT Governance is the term used to describe the selection and use of organizational processes to make decisions about how to obtain and deploy IT resources and competencies."

- Henderson and Venkatraman, IBM Systems Journal, 1993

IT Governance provides the fundamentals for better prioritizing and planning between IT and Business. Its definitions include:

- Operating model for how organization will make decisions about use of IT.
- Involves external relationships for obtaining IT relationships.
- Involves authority, control, accountability, roles, and responsibilities.
- Involves processes and methods for making decisions.
- Involves judgments about how well use of IT enables strategic direction.

The Figure 91 shown is the international standard of IT governance, called “ISO/IEC 38500”. “While the IT governance standard is a brilliantly short and straight-forward international standard, actual implementation of an IT governance framework can be challenging”. (Calder, 2006)

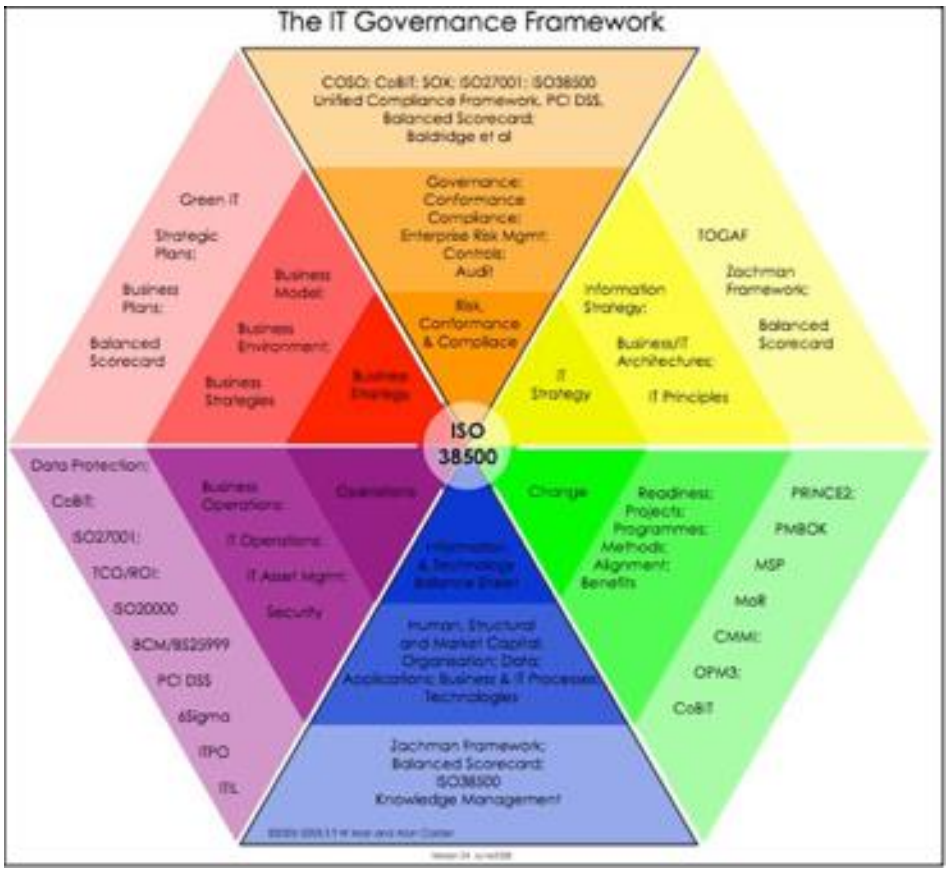


Figure 85: The IT Governance Framework (Calder, 2006)

Partnership

Partnership allows better shared definition, goal, and management with an emphasis on collaboration between IT and Business. Good partnership allows better cooperation and coordination for cross discipline work in an organization.

Scope and Architecture

Good scope and architecture offers a company architectural transparency, agility, and flexibility with management of emerging IT. Building a right scope with the correct architecture gives am organization the chance to gain better management advantages other organizations.

Skills

Innovative approach of management leads to cultural power and change responsiveness, thus good IT skills bring a great potential in enhancing a company's business and culture. These skills also bring an organization various benefits including more efficient communication.

Business - IT Strategic Alignment

Good strategic alignment bridges the gap between IT and Business. As the model shown in Figure 92, it takes effort to build a process to better align Business strategy and IT strategy. The five-level approach explains how strategic alignment could be affected by the different levels of the process. A goal which every organization should aim at is Optimized Process as the alignment gap between Business and IT strategy is minimized at Level 5. "A shared understanding of how IT applications, technologies and services will contribute to business objectives – today and in the future." (infotech.com, 2014) A good Business – IT alignment is usually built upon four steps:

- Set Conditions to Achieve Alignment,

- Scan for Potentially Enabling Technologies,
- Determine IT Value Imperatives,
- Develop IT Vision and Mission.

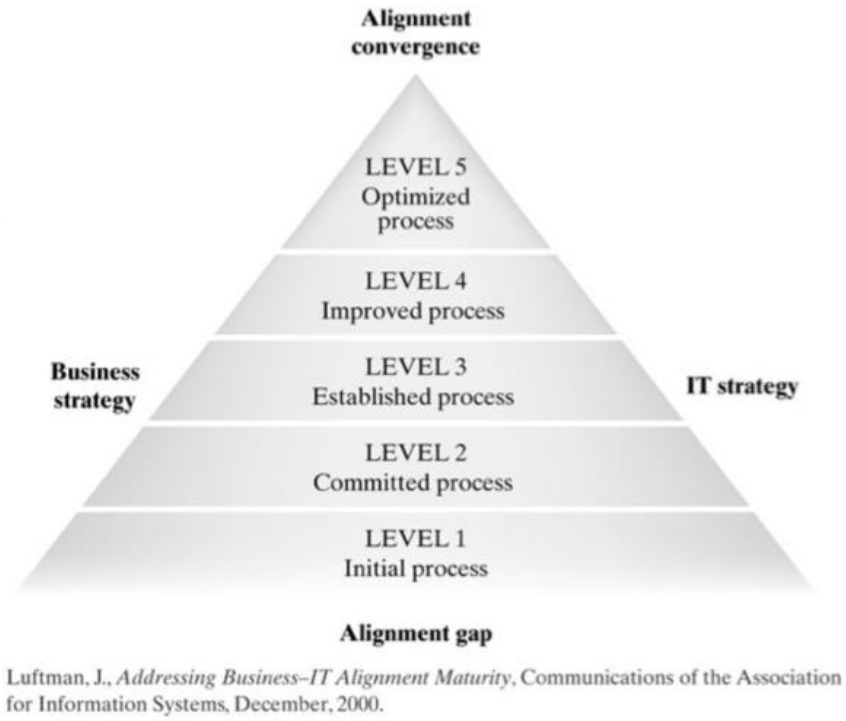


Figure 86: Addressing Business-IT Alignment Maturity. (Luftman, 2000)

As the key of the IT-Business strategic alignment being to lower the gap between Business and IT strategies, the role of Chief Information Officer (CIO) in a firm becomes extremely essential in the process.

Emerging Technologies

Emerging technologies such as Big Data is creating unlimited possibilities of how IT can enhance the way people work. Among the many technologies that IT could offer to an

organization's internal customers, Big Data is the big trend. Other technologies including Business Intelligence and Cloud Computing is creating even more innovative approaches to let people get the most out of IT and ultimately improve the their businesses. Technology is changing so fast that the team never knows which technology is going to revolutionize the way they are used to live and work. The fast development in the industry is creating more opportunities for IT to be utilized in a business environment. Depends on the type of the organization, the technologies that are useful might vary in lots of ways.

Appendix C: Detailed Staffing Plan

Zeng “Arica” Liu

Zeng is double majoring in Management Information Systems and Computer Science. She completed her Interactive Qualifying Project in London, England and will be completing her Computer Science MQP in Silicon Valley. She worked as an Information Technology Intern at Liberty Mutual. Next year she will be working as a Technology Consultant at Sapient Global Markets. She enjoys solving for complex algorithms and coding.

Thomas Meagher

Thomas is majoring in Management Information Systems. He completed his Interactive Qualifying Project in Zurich, Switzerland and worked as a Systems Analyst Intern at Monster.com. Next year he will be working as a Technology Consultant for Sapient Global Markets and pursuing a Master’s in Data Science at WPI as part of the combined Bachelor's/Master's program. His favorite class at WPI was Systems Analysis and Design.

William Richtmyer

William is majoring in Management Information Systems. He completed his Interactive Qualifying Project in Venice, Italy and worked as an EITS Account Management Intern at CVS Health. Next year he will be living in Providence and working in CVS Health’s rotational program as a project manager and business analyst. He enjoys the challenges of connecting business with technology.

Jiedong "JD" Wang

Jiedong is majoring in Management Information Systems with a minor in Industrial Engineering. He completed his Interactive Qualifying Project in Washington DC and has worked as an IT Business Consultant Intern at EMC and a Data Analyst Intern at Fitivity. Next year he plans on attending graduate school while working part-time. His favorite classes at WPI were Systems Analysis and Design and User Experience.

Amanda Weis

Amanda is majoring in Management Information Systems with minors in Computer Science and International Studies. She completed her Interactive Qualifying Project in Venice, Italy and has worked as an IT Quality Assurance and Testing Services Intern at EMC. Next year she plans to attend graduate school. She enjoys topics surrounding information security and user experience.

Appendix D: Feasibility Analysis

Member Repository Major Qualifying Project Feasibility Analysis

After examining the systems request (Appendix A) our team conducted a feasibility analysis to study the options for creating a member repository system. The detailed feasibility study can be found in chapter three of this document; the highlights of the feasibility analysis are outlined below:

Technical Feasibility

The creation of a member repository is feasible from a technological standpoint, however the system is limited by a few factors:

- Technical expertise of the development team
- Technical expertise of the team that will ultimately be using the system

These factors limit the use of development tools for the member repository to options that will be comfortable for our development team as well as the end users of the system.

Organizational Feasibility

The creation of a member repository is feasible from an organizational standpoint; the following should be noted:

- There is no system currently in place to store members
- The creation of a system is highly supported by various stakeholders including Mr. Sean Burke, Mr. Jack Bergersen, Mr. Dave Rosenburg, and Ms. Laura Naylor (the Senior Vice President of Member Experience and Operations)

Economic Feasibility

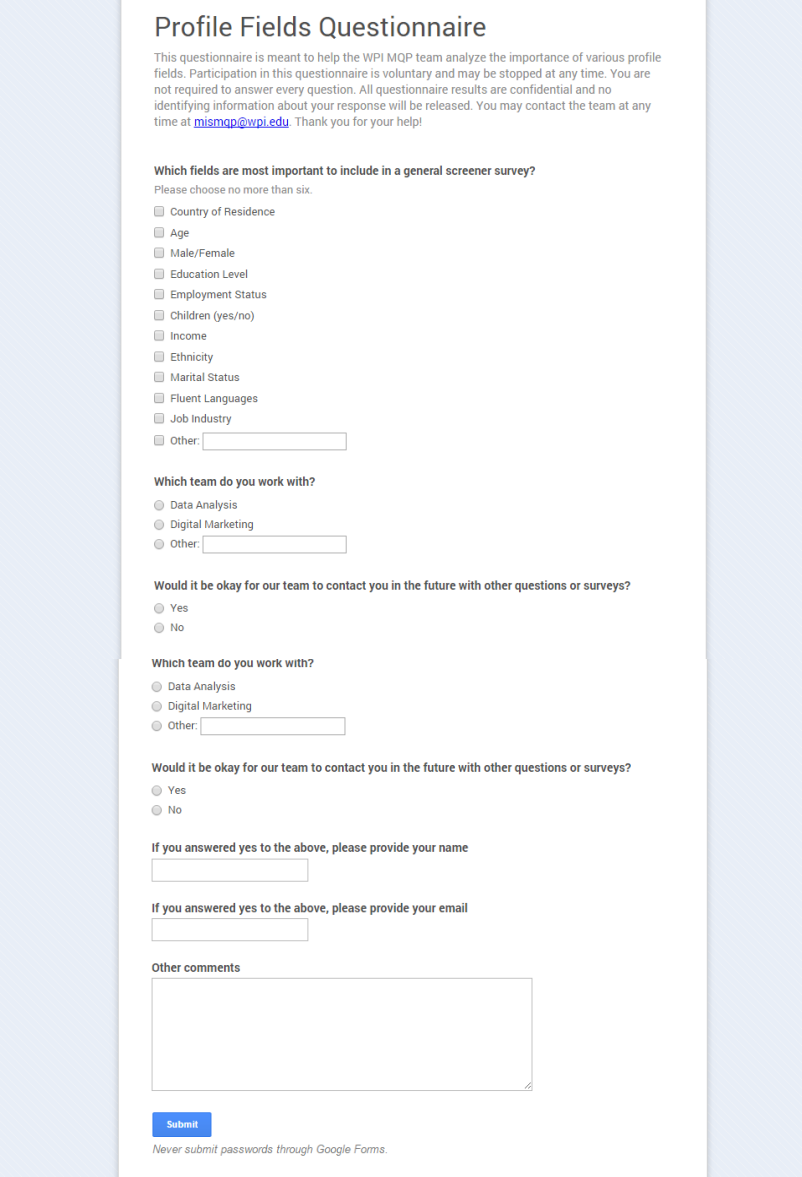
A cost–benefit analysis was performed; our analysis shows that using a team from WPI is economically far superior to using an internal employee or another external consultant.

Overall, the system stands to bring a large amount of economic value to the team because it will more efficiently use the budget for recruiting and will provide a “free” source of members for communities.

Intangible Costs and Benefits include an increase in employee satisfaction because the system will help them more easily recruit members for communities.

Appendix E: Profile Fields Questionnaire

This appendix includes a copy of the complete profile fields questionnaire that the team sent to the Digital Marketing and Data Analytics teams, as well as the results of the questionnaire.



Profile Fields Questionnaire

This questionnaire is meant to help the WPI MQP team analyze the importance of various profile fields. Participation in this questionnaire is voluntary and may be stopped at any time. You are not required to answer every question. All questionnaire results are confidential and no identifying information about your response will be released. You may contact the team at any time at mismqp@wpi.edu. Thank you for your help!

Which fields are most important to include in a general screener survey?

Please choose no more than six.

- [ ] Country of Residence
- [ ] Age
- [ ] Male/Female
- [ ] Education Level
- [ ] Employment Status
- [ ] Children (yes/no)
- [ ] Income
- [ ] Ethnicity
- [ ] Marital Status
- [ ] Fluent Languages
- [ ] Job Industry
- [ ] Other:

Which team do you work with?

- ( ) Data Analysis
- ( ) Digital Marketing
- ( ) Other:

Would it be okay for our team to contact you in the future with other questions or surveys?

- ( ) Yes
- ( ) No

Which team do you work with?

- ( ) Data Analysis
- ( ) Digital Marketing
- ( ) Other:

Would it be okay for our team to contact you in the future with other questions or surveys?

- ( ) Yes
- ( ) No

If you answered yes to the above, please provide your name

If you answered yes to the above, please provide your email

Other comments

Submit

Never submit passwords through Google Forms.

Figure 87: Questionnaire

Results

[THIS SECTION HAS BEEN REDACTED]

[THIS SECTION HAS BEEN REDACTED]

Appendix F: Meeting Minutes

[THIS SECTION HAS BEEN REDACTED]

Appendix G. Project Gantt Chart

The project Gantt Chart is shown in Figure 9. Each of the four project phases were discussed in detail previously in section 3.4 of the paper.

[THIS SECTION HAS BEEN REDACTED]

Figure 88: Project Gantt Chart

Appendix H. Project Presentation

[THIS SECTION HAS BEEN REDACTED]

Appendix I. Project Documentation

[THIS SECTION HAS BEEN REDACTED]

Appendix J: Technical Documentation

[THIS SECTION HAS BEEN REDACTED]

Proposal Sign-Off

Project Name: Prospective Member Sourcing System

WPI MQP Team: Zeng Liu, Thomas Meagher, William Richtmyer, Jiedong Wang, Amanda Weis

Sponsors: Laura Naylor, David Rosenberg, Jack Bergersen, Sean Burke

WPI MQP Advisor: Eleanor Loiacono

Date: November 12, 2014

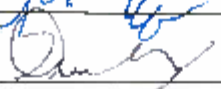
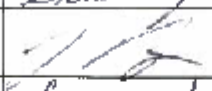
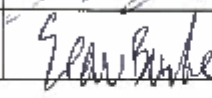
Project Proposal Sign-Off

This indicates approval to proceed to the next phases and indicates an understanding and formal agreement that the project is ready to proceed to the next phase of the initiative.

In signing this document, the signatory agrees that the WPI MQP Team should further invest in delivery of this project.

Phases Completed: Planning and Proposal

Next Phases: Analysis, Design, and Implementation

Approver Name	Signature	Sign Date	Comments
Laura Naylor		11/19/14	
David Rosenberg		11/19/14	
Jack Bergersen		11/19/14	
Sean Burke		11/19/2014	

Final Project Sign-Off

Project Name: Prospective Member Sourcing System

WPI MQP Team: Zeng Liu, Thomas Meagher, William Richtmyer, Jiedong Wang, Amanda Weis

Sponsors: Laura Naylor, David Rosenberg, Jack Bergersen, Sean Burke (until 2/28/15)

WPI MQP Advisor: Eleanor Loiacono

Date: March 4, 2015

Final Project Sign-Off

This indicates approval to conclude the project. It is an understanding and formal agreement that the MQP team has presented Communispace with adequate deliverables.

In signing this document, the signatory agrees that the WPI MQP Team has concluded their work with Communispace.

Phases Completed: Planning, Analysis, Design, Implementation, Recommendations, and Formal Report

Approver Name	Signature	Sign Date	Comments
Laura Naylor		3/16/15	
David Rosenberg		3/16/15	
Jack Bergersen		3/16/15	

References & Bibliography

Communispace Website. (2014). from http://www.communispace.com/

Competency and Value of IT: The Second Dimension of the SAMM (2014), from http://businessitalignment.wordpress.com/2011/02/01/competency-and-value-of-it-the-second-dimension-of-the-samm/

Dennis, A., Wixom, B. H., & Roth, R. M. (2012). *Systems Analysis and Design*: Wiley.

Leung, D. K., Doyle, E. J., Carey, D. A., & Loiacono, E. T. (2013). *Vendor Performance and Relationship Management System*. Worcester, MA: Worcester Polytechnic Institute.

Luftman, Jerry (2000): Assessing Business-IT Alignment Maturity, Communications of AIS, Volume 4, Article 14

Reporting Services (SSRS). (2014). from http://msdn.microsoft.com/en-us/library/ms159106.aspx

Stone, D., Jarrett, C., Woodroffe, M., & Minocha, S. (2005). *User Interface Design and Evaluation*: Elsevier Science.

Taylor, A. G., Snoddy, S. W., Mannheim, G. D., Karp-Neufeld, G. G., & Loiacono, E. T. (2014). *Extension of Vendor Performance and Relationship Management System*. Worcester, MA: Worcester Polytechnic Institute.

Worcester Polytechnic Institute (WPI) Website. (2014). from http://www.wpi.edu/

Glossary

- **Break-even Point (BEP)**

  The point at which total cost and total revenue are equal.

- [THIS SECTION HAS BEEN REDACTED]

- [THIS SECTION HAS BEEN REDACTED]

- **Dynamic Systems Development Method (DSDM)**

  An agile project delivery framework, primarily used as a software development method.

- **Extreme Programming (XP)**

  A software development methodology which is intended to improve software quality and responsiveness to changing customer requirements.

- [THIS SECTION HAS BEEN REDACTED]

- **Human-computer Interaction (HCI)**

  A discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.

- **Information System (IS)**

  A system composed of people and computers that processes or interprets information.

- **Information Technology (IT)**

  The application of computers and telecommunications equipment to store, retrieve, transmit and manipulate data, often in the context of a business or other enterprise.

- **Interactive Qualifying Project (IQP)**

  A WPI project that explores the intersection of science, technology and society.

- **Major Qualifying Project (MQP)**

This is a senior-year capstone project at WPI, where students will be able to gain real-world design or research experience within their major field.

- **Management Information Systems (MIS)**
  A program provided by WPI School of Business to train MIS professionals who use computer technology to solve problems and meet the ever-changing needs of business.
- **Market Research Online Communities (MROCs)**

  A technique for gathering real-time, qualitative market research insights.

- [THIS SECTION HAS BEEN REDACTED]

- **Net Present Value (NPV)**

  The present value of the cash flows at the required rate of return of a project compared to its initial investment.

- **Open Database Connectivity (ODBC)**

  A standard programming language middleware API for accessing database management systems.

- [THIS SECTION HAS BEEN REDACTED]

- **PDF**

  A file format that provides an electronic image of text or text and graphics that looks like a printed document and can be viewed, printed, and electronically transmitted.

- **Prospective Member Sourcing System (PMSS)**

  A system developed by the writers of this project report to help Communispace's Client and Consumer Services team to more effectively and efficiently manage their recruits and online community members.

- **Rapid Application Development (RAD)**

  An approach to software development that puts less emphasis on planning tasks and more emphasis on development.

- **Return On Investment (ROI)**

  The benefit to the investor resulting from an investment of some resource.

- **Structured Query Language (SQL)**

A programming language that is designed to manage data in a relational database management system.

- **SQL Server Reporting Services (SSRS)**
  A server-based reporting platform that provides comprehensive reporting functionality for a variety of data sources.

- [THIS SECTION HAS BEEN REDACTED]

- **Systems Development Life Cycle (SDLC)**
  The process of planning, creating, testing and deploying an information system.

- **User Experience (UX)**
  UX involves a person's behaviors, attitudes, and emotions about using a particular product, system or service. User experience includes the practical, experiential, affective, meaningful and valuable aspects of human–computer interaction and product ownership.

- **User Interface (UI)**
  The space where interactions between humans and machines occur.

- [THIS SECTION HAS BEEN REDACTED]

- **Visual Basic (VB)**

  A third-generation event-driven programming language and integrated development environment (IDE) from Microsoft for its COM programming model first released in 1991.

- **Visual Basic for Applications (VBA)**

  An implementation of Microsoft's event-driven programming language, Visual Basic 6, and its associated integrated development environment.

- **Worcester Polytechnic Institute (WPI)**
  A private technological university located in Worcester, MA.