Software Requirements and Planning: CommunityConnect

Kamron Ebrahimi & Samuel Wilson & Leif Tsang
& Thomas Korsness & Quinton Osborn
ebrahimk, wilsosam, tsangl, korsnest, osbornq

January 29, 2018

Contents

1 Project Description

1.1 Approach

CommunityConnect is a web-based application used to help individuals from diverse ethnic backgrounds build a social support network. CommunityConnect achieves this through a simple matching algorithm. Users will create an account, in which they submit information such as their ethnic background, religion, language, a brief biography, contact information and location, and their accounts information will be cross compared with other user accounts, looking for other individuals with the same cultural background within some given mile radius.

CommunityConnect targets displaced individuals or immigrants living in the United States. A host of scientific inquiries have been made into the effects of support networks among displaced individuals and immigrants. The 2014 study, *Social isolation and perceived barriers to establishing social networks among Latina immigrants* [1], found that perceived social isolation and lack of social support negatively impact health and that many immigrants unaccustomed to US culture report feeling lonely, isolated, closed-in, and less free in the US due to family separation and various obstacles to develop and maintain relationships. This is not the only study to yield such results in fact there is a specific term for this phenomena coined cultural isolationism. The 2010 study *Social support and health: immigrants and refugees perspectives* [2], investigated the effects of social support on the utilization of public services among refugees and immigrants and found that there exist a positive correlation between the two. This study further reinforces the need for some web based application like CommunityConnect.

1.2 Innovation

There are many humanitarian web-based applications which serve to connect immigrants or refugees with local resources or with lost family members and friends. Similarly the web application Meetup allows user to join groups based off of their ethnicity, and designated, user-organized events for large groups of people in metropolitan areas. CommunityConnect would personalize the user experience even more and potentially serve immigrants/refugees who move to more rural locations by finding individuals who share the users cultural background who are physically close by. If a person who lived in a rural community used Meetup they would potentially have to travel sometime to gathering location and then be exposed to a large gathering of people who may already be acquainted with one another. CommunityConnect would connect people on an individual and geographical basis so users can build close knit relations even in rural areas.

1.3 Tools

CommunityConnect would require a variety of different programming languages for both client and server side interactions. On the client side, languages such as HTML, CSS, and Javascript would be used to create an aesthetic webpage interface for the user. PHP will be used to write the matching algorithm on the server side of the process. HTML, CSS, and JavaScript are standard client-side web development languages and all members of the CommunityConnect group are familiar with these programming languages, thus no time will be wasted getting members up to speed. PHP will be the server side scripting language used to develop CommunityConnect. As a team CommunityConnect needs to get up to speed on server side scripting, however we are selecting PHP as our the primary server side scripting language because some members of the team have fundamental skills with the language.

1.4 Requirements

1.4.1 Functional

1. The user will be able to create a profile containing their information containing but not limited to:
   - Contact Information
   - Location
   - Ethnic Background
   - Country of Origin
   - Religion
2. The user will be able to perform a search for others within a certain mile radius of their location.
3. Searches will return the contact information of other uses with identical ethnic backgrounds.

1.4.2 Non-Functional

1. CommunityConnect will securely maintain user profile information.

2. CommunityConnect will return search results within 1 minute of processing a request.

3. Profiles of individuals are only public to the user if they match with that individual.

1.5 Documentation and Features

CommunityConnect will have a home webpage on which will be hosted a brief biography about the developers, a setting option allowing users to swap between the language in which the page is displayed and a brief guide on how CommunityConnect works. The guide will be the users main resource for questions regarding the use of the application.

The four major features of CommunityConnect which make it so useful are as follows:

1. Personalized profile creation containing contact information, biography, cultural background.

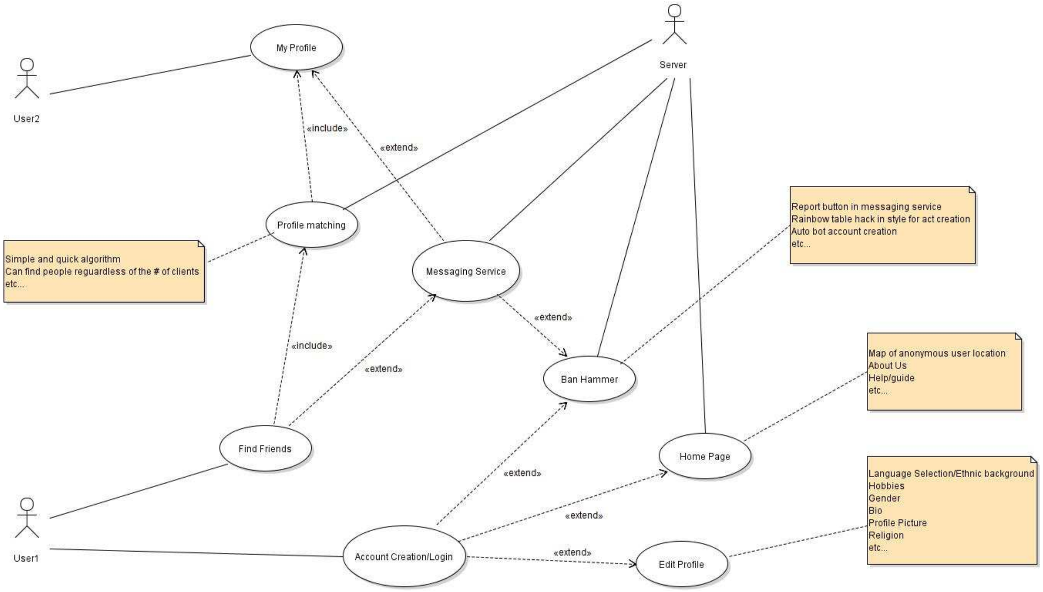
2. Intricate profile matching algorithm which generates matches based off user location and ethnic background.

3. A simple reporting system for protecting users from harassment, and a system for banning/holding and an account.

4. A homepage with a user guide and statistics/graphic on the applications traffic in different locations.

2 Use Cases

2.1 Diagram



2.2 Use Case 1

Use Case Name:
Friend Matching

Goal Level:
Create an algorithm which allows people to find others near them with the same cultural background.

Actor:
The server and the user

Preconditions:
The user has created an account with their personal information and has decided to use the matching algorithm.

Postconditions:
The system returns the profiles of individuals, with the same cultural background who are in the same location.

Flow of Events:
The user must decide to find my friends and then the server must provide matching profiles that are flagged as similar to the user.

Quality Requirements:
The matching algorithm must return user profiles in under ten seconds and must account for high database usage.

Importance:
This use case is very important because without friend matching, CommunityConnect has no functionality. The matching system utilized in the application is the selling point of the entire system and thus needs to be well organized, efficient and accurate.

Error Scenario:
The friend matching algorithm could encounter a few issues including the possibility of long searching times and dead results (No results coming up in the area). The first can be fixed with efficient search algorithms however the second issue comes with a factor that cannot be controlled. This issue is a byproduct of user traction in given areas.

2.3 Use Case 2

Use Case Name:
Profile Creation

Goal Level:
Users can create a profile which contains the users email address or phone number, a brief biography, interests, cultural background, religion, and fluent languages.

Actor:
The user is the primary actor, though the server must store profiles in a dynamic array or heap.

Preconditions:
The user accesses the CommunityConnect website and decides to make a profile.

Postconditions:
The user has a profile containing their personal information, which is stored in a remote server.

Flow of Events:
After the user logs in or creates an account they are met with the option to edit profile. This will allow them to make and changes they deem necessary to keep it updated. Alternatively, the user now has the option to utilize the friend matching function of CommunityConnect.

Quality Requirements:
Ideally the profile creation aspect of CommunityConnect will be able to access some geographical data and determine whether or not a user entered a valid location within the United States.

Importance:
This use case is critical to the function of CommunnityConnect because without a profile, the friend matching algorithm has not data through which to generate matches.

Error Scenario:
Profiles could potentially have issues such as users providing too much info or putting the wrong info into the wrong fields. This can be circumvented by putting pattern recognition in some fields as well as a once over check by a system to verify that the information was entered correctly.

2.4 Use Case 3

Use Case Name:
Home Page (Application Information)

Goal Level:
The CommunityConnect home page will provide a visual containing information on the amount of user traffic in different areas, as well as a guide on how to utilize the application and the applications mission statement.

Actor:
The actor is the user, or individual accessing the home page of the website. The server must also store the contents of the website.

Preconditions:
The individual must a have some means of accessing the internet.

Postconditions:
The individuals is directed to a home page where they can get helpful hints on using the application or just simply learn why the application was created and its intended use.

Flow of Events:
The user accesses CommunityConnect via an internet connection. Unfamiliar with the application, the user accesses the home page to learn how to create a profile and get started using CommunityConnect.

Quality Requirements:
The home page must display a real time visual with statistics on user traction in different areas. The tentative plan was to display

Importance:
The home page is the central resource for new users to gain insight into the purpose of CommunityConnect and also get a general idea whether or not their are individuals in their area who use the application.

Error Scenario:
The home page visual may not account for stale profiles, additionally because CommunityConect is targeted toward immigrant and refugees, the home page may be unsuccessful at conveying information about the application. Thus from the home page the user has the ability to change the language in which the application is displayed.

3 Planning

3.1 Milestones

All milestones are listed in the order in which they must be completed. Each succeeding milestone is dependant on the proper functionality of the preceeding milestone.

3.2 Home Page

Create Home Page consisting of clickable links to different pages within the website

3.2.1 Tasks

- Create HTML file for webpage layout
- Create CSS file for webpage aesthetics
- Create JavaScript file for webpage interactivity
- Write user help guide

3.2.2 Resources

- Some text editor: atom, sublime text, vim etc. etc.
- Github
- Database/Server on which to push the HTML, CSS and JavaScript files

3.3 HTML Form

Complete the HTML form for user account creation. Users should be able to fill out a form which sends data to a remote server. The form will have all specified fields such as ”name”, ”country of origin” etc.

3.3.1 Tasks

- Create HTML file for form layout, specifying the proper fields of user input
- Create CSS file for an aesthetic form design

3.3.2 Resources

- Some text editor: atom, sublime text, vim etc. etc.
- Github
- Database/Server to which the HTML form is submitted

3.4 Server Architecture

Complete server architecture for storing profiles in a dynamic array or heap.

3.4.1 Tasks

- Access/identify remote server on which profiles will be stored
- Generate a system which neatly organizes profiles in some efficient data structure (most likely a dynamic array)
- Create server client socket such that profiles can be readily accessed and manipulated on the client side.

3.4.2 Resources

- Mongodb
- PHP libraries
- Some text editor: atom, sublime text, vim etc. etc.
- Github

3.5 Sorting Algorithm

Write server side sorting algorithm which cross compares user profiles and identifies matches.

3.5.1 Tasks

- Design an efficient means of sorting through a given data structure (dynamic array)
- Write pseudo-code for the algorithm
- Implement algorithm using some server side scripting language, most likely PHP
- Push algorithm onto some remote server where profiles will be stored

3.5.2 Resources

- PHP libraries
- Some text editor: atom, sublime text, vim etc. etc.
- Github

3.6 Reporting Function

Include a reporting function on profiles to mitigate harassment.

3.6.1 Tasks

- Write pseudo-code for reporting algorithm
- Include a ”report” button on webpage layout
- Implement algorithm, which reports individuals who harass users, after two report notices, the users account will be deleted

3.6.2 Resources

- PHP libraries
- Some text editor: atom, sublime text, vim etc. etc.
- Github

3.7 System Testing

Begin testing CommunityConnect.

3.7.1 Tasks

- Perform load testing
- Test to see if application generates matches properly
- Verify that home page visual displays real time where users are, and in what numbers
- Get feedback from public on the system
- Document all user feedback

3.7.2 Resources

- Code/systems for load testing
- People willing to try the system out and provide feedback

3.8 Launch

Launch CommunityConnect and brainstorm some advertising and marketing strategies.

3.8.1 Tasks

- Fix issues found in testing, this may take some time

3.8.2 Resources

- Some text editor: atom, sublime text, vim etc. etc.
- Github
- Customer/customer feedback

3.9 Schedule/Timeline

timeanddate.com

February 2018 (United States)

March 2018

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31 ○ Full Moon	1	2	3
4	5 Create Home Page. Home page will consist of clickable links to different pages within the website	6	7 ◑ 3rd Quarter	8	9 HTML form for account creation deadline. Users should be able to fill out a form which sends data to a remote server. The form will have all specified fields such as "name", "country of origin" etc.	10
11	12 Complete server architecture for storing profiles in a dynamic array	13	14	15 ● New Moon	16 Prototype server side sorting algorithm, which cross compares user profiles	17
18	19 By this point users should be able to create a profile, perform a search and have results displayed back to them	20	21	22	23 Include a reporting function, to mitigate harassment ◐ 1st Quarter	24
25	26 Begin system testing	27	28	1 ○ Full Moon	2	3

timeanddate.com

March 2018 (United States)

April 2018

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Sun	Mon	Tue	Wed	Thu	Fri	Sat
25	26 Begin system testing	27	28	1 ○ Full Moon	2	3
4	5 Add any extraneous details, complete bug testing	6	7	8	9 ◑ 3rd Quarter	10
11	12 Make all user interface aesthetic and visually pleasing	13	14	15	16	17 ● New Moon
18	19	20	21	22	23	24 ◐ 1st Quarter
25	26	27	28	29	30	31 ○ Full Moon

Our team has decided to convene on a weekly basis to discuss how we want to tackle each milestone. Thus we are not automatically partitioning specific task to individuals but rather approaching each new task as a group. We decided on this method to facilitate group communication. CommunityConnect is not large in scope that tasks need to be divided, instead by approaching each milestone as a group, all members of the team gain a holistic understanding of the system as a whole. The above calendars provide a tentative outline for when each milestone should be completed and when system testing should begin.

3.10 Project Tracking

Progress will be tracked on a weekly basis during team meetings. All team communication will be done through the CommunityConnect slack channel, allowing the team to work/communicate remotely. All team members are required to update the README.md files for each GitHub, do display what work they accomplished on specific tasks and what needs to be done next.

3.11 Risk Management

3.11.1 Inconsistent Tool Selection

Team members using different methods of page creation may conflict.

1. **Solution:**
2. **Make sure everyone knows what software is being used for each coding task.**

3.11.2 Collaboration

Team members not contributing or communicating (not attending the weekly in-person meetings) when working on critical system components could lead to system failure because the team will not understand code changes or worse what still needs to be accomplished.

1. **Solution:**
2. **Keep our project simple in scope and communicate regularly via the slaack channel.**

3.11.3 Feature creep

If the team keeps adding functionality to the application the scope of the idea will become blurred leading to a poorly defined system.

1. **Solution:**
2. **Make sure core features are complete before expanding.**

4 Meeting Report

This week was spent organizing our project and brainstorming exactley what features we want CommunityConnect to have by the end of the term. No coding has been accomplished yet, but a tentative outline of what need to be done, and by when has been specified in the planning section of this report. Additionally, all members of the team are active on the team slack channel and have agreed to meet on a weekly basis on Saturdays. This will provide the team time to communicate and work on important tasks which need to be accurately understood by all members of the team. This week we laid the foundation off which CommunityConnect will be built and solidified a concrete idea of what each member wants to see in the final product. Next week we aim to begin working on the home page for the application using HTML, CSS and JavaScript. By the end of next week we all want to see a well formatted, easily understood home page for CommunityConnect, as well as an HTML form which allows the user t submit their personal information and parses said information to a remote server. We are thinking about using one of the Oregon State engineering servers, so we have an easily accessible location to prototype our design.

Each member of the group has a made an effort to communicate and contribute to the over all planning and requirements of CommunityConnect. Samuel Wilson, Thomass Korsness, and Quinton Osborn all worked on planning the developement of CommunityConnect. The use cases and use case diagrams were generated by Leif Tsang and finally the project description, meeting report and writing of the official LaTex document was completed by Kamron Ebrahimi. The initial idea of CommunityConnect was brainstormed by Kamron and Samuel (customers) and both members of the team have communicated all of the details of CommunityConnect with the other team members and the team as a whole has come to solid understanding of exactly what creating the CommunityConnect applicaiton will involve and what the systems requirements are.

References

[1] Hurtado de Mendoza A, Gonzales FA, Serrano A, and Kaltman S. *Social isolation and perceived barriers to establishing social networks among Latina immigrants.* American Journal of Community Psychology, 2014.

[2] Miriam J Stewart, Edward Makwarimba, Morton Beiser, Anne Neufeld, Laura Simich, and Denise Spitzer. *Social support and health: immigrants and refugees perspectives.* Diversity and Equality in Health Care, 2010.