By: Colin Harrison, Christian McMurtrie, Timothy Nakhisa, and Zachary Rivera

11/15/2015

CAKE Group

Second Iteration Report



Contents

[Introduction 1](#_Toc437334571)

[Website 1](#_Toc437334572)

[Project Management Plan 1](#_Toc437334573)

[Project Organization 1](#_Toc437334574)

[Risk Analysis 1](#_Toc437334575)

[Hardware and Software Resource Requirements 1](#_Toc437334576)

[Work Breakdown 1](#_Toc437334577)

[Project Schedule 2](#_Toc437334578)

[Project Overview 2](#_Toc437334579)

[Problem and Solution 2](#_Toc437334580)

[Stakeholders 2](#_Toc437334581)

[Scope 3](#_Toc437334582)

[Requirements 3](#_Toc437334583)

[Development, Operation, and Maintenance Environments 3](#_Toc437334584)

[System Model 3](#_Toc437334585)

[Textual Use Cases 3](#_Toc437334586)

[User Interaction 4](#_Toc437334587)

[Functional Requirements 4](#_Toc437334588)

[Primary Requirements 4](#_Toc437334589)

[Secondary Requirements 5](#_Toc437334590)

[Nonfunctional Requirements 5](#_Toc437334591)

[Feasibility 6](#_Toc437334592)

[Quality Assurance Plan 6](#_Toc437334593)

[Document Standards 6](#_Toc437334594)

[Coding Standards 6](#_Toc437334595)

[User Interface Guideline 6](#_Toc437334596)

[Change Control Process 7](#_Toc437334597)

[Testing Process 7](#_Toc437334598)

[Conclusion 7](#_Toc437334599)

[Appendix A 9](#_Toc437334600)

[Use Cases 9](#_Toc437334601)

[Appendix B 12](#_Toc437334602)

[System Models 12](#_Toc437334603)

# Introduction

Our team is Computer Applications: Knowledgeable Engineering (CAKE). The members of this team are Colin Harrison, Christian McMurtrie, Timothy Nakhisa, and Zachary Rivera. We are working with our client, Dr. Yvonne Chueh, to create a website similar to “Meal Train”. This website is intended to help the elderly in planning, organizing, and executing social events.

# Website

<http://caketeamcwu.wix.com/cake>

# Project Management Plan

## Project Organization

We will be using an agile Kanban method for solving our problem to keep organized. This will make it so we can pair program, keep in contact, and have a good developer to quality assurance ratio to constantly push out a refined product each sprint for the client. This will help our client see a consistent growth, and help us keep on track of where are project is at, where it’s going, and what changes might have to be made. Currently we will have three developers (Christian, Tim, and Zac) and one quality assurance tester (Colin). We also have designated our scrum master to be Colin, our documentation manager is Christian, and version control system manager is Zac. We will be rotating these rolls every four weeks to make sure that everyone can experience each roll.

## Risk Analysis

A large risk for our project would be feature overload. To make this website fully functional we must implement a lot of features, but we most focus on them one at a time and not get overwhelmed. The agile Kanban organizational method will help us to not get overwhelmed, and consistent interactions with our client to be able to update, and manage expectations.

|  |  |  |  |
| --- | --- | --- | --- |
| **Issue** | **Impact** | **Feasibility** | **Priority** |
| Compatibility | HIGH | HIGH | HIGH |
| Readable Text | HIGH | HIGH | HIGH |
| Volunteers | HIGH | MED | MED |
| Share Photos | MED | HIGH | HIGH |
| Paid Option | LOW | MED | LOW |
| Languages | LOW | LOW | LOW |

## Hardware and Software Resource Requirements

For our project we will each need a computer with the most widely used browsers installed, such as Firefox, Chrome, Edge, and Safari to be able to test our website on. There are also third party programs we can use that would let us tap into a multitude of different browser and operating system configurations called “Saucelabs.” Our target installation will be at least the main browsers Firefox and Chrome, then hopefully expanding to others as the project goes on. Furthermore, we will be using “GitHub” for our repository and Microsoft Office for reports and presentations. We will also need html editors such as IntelliJ, and MySQL or another way to manage databases of user information.

## Work Breakdown

We will be using an iteration “onion layering” breakdown where we will start by trying to get a homepage set up. After we will populate the home page, and get tab placeholders for the next pages. We will take each feature on one at a time to get the project done that way we have a base to build upon getting closer to our end product the client has requested while maintaining a functional product.

## Project Schedule

Our goal is to reach each milestone every sprint (which is every two weeks), and updating our assignments during our scrum meetings (a few times a week). We will hopefully be continuing to build upon each previous sprint so our assignments would be changed if we completed our goals on our previous sprint. We would have a retro after a sprint and document what got done to help project our next sprint and if we are on track. With Kanban we could tackle our project two ways: having separate developers each working on one task at a time or by using paired programming for features to continue the “conveyor belt” while having quality assurance continuously checking the developers last feature.

# Project Overview

This project will aim to help elderly persons facing social isolation due to the lack of nearby family members. A website will be created promoting social activities as well as more personal outings for seniors. It will allow for members of the local senior community and their distant families to stay connected.

## Problem and Solution

Rapidly advancing technology has allowed people to remain connected even when half a world apart. A person in Seattle can speak with and see a person in London in real time. Or fly from New York to Los Angeles in just a few hours. These advances have allowed people to move freely around the world and still remain in touch with family and friends from their hometowns.

These changes have also led to a group of increasingly isolated individuals; the elderly. As more families choose to relocate for work or personal reasons, the parents and grandparents often stay put in their hometowns. As they age and mobility diminishes, socializing becomes more difficult if not nearly impossible. A lack of socializing can lead to reclusion which can contribute to deteriorating health, both mentally and physically (John T Cacioppo, 2003).

The solution is to promote social outings for the aging population, whether by organizing group outings or having younger community members step in when family is not available. The vision of Dr. Chueh is to have an online site where such activities can be arranged. This would include having a system where families away from older relatives can contribute to an account funding events or meal outings that allow the older family member to socialize outside of their own home.

While software promoting social activities is not unique, it often misses some of the most vulnerable demographics. This software would solely focus on those who may not be familiar with or able to use computers or the internet well and who may have trouble getting out on their own.

## Stakeholders

The client is Dr. Chueh. She is a Professor of Mathematics at Central Washington University in Ellensburg, WA. She was inspired to do something for the aging community during a trip to the east coast. During this trip, she attended a funeral where the family of the departed received food and condolences from members of an online site called “Meal Train”. She recognized the good that can come from a community banding together in times of need and thought of her own family. She saw the limitations of the Meal Train site and wanted to create something that was proactive.

The elderly in communities across the globe are also stakeholders. For this project, we will limit this to the elderly in Ellensburg. They currently cope with the problem of isolation by attending (when possible) small gatherings at church’s or sometimes hosted by local retirement homes. There is no prevailing solution.

The families of the elderly who do not live near-by and are unable to visit regularly is another group of stakeholders. These people care for their family members but are rooted in homes across the country making their ability to help impractical with current resources. This site would give them a tool that will allow them to help their relatives maintain a healthy social life.

## Scope

The software will help alleviate the problem by providing a dedicated site to increasing social health among the elderly. It will provide an easy to use, senior friendly interface that will help elderly persons connect to other members of the community.

It will also create a place where distant family members can help their loved ones stay healthy and connected to society. This should reduce some of the stress that can be caused worrying about an older family member.

The site will provide a much needed tool to promote healthy social living, but the weight of utilizing the tool will still rely on people. Family members will need to donate to elderly without sufficient income to pay for their own outings. Younger members of the community will need to volunteer their time, and family members will need to use good judgement when accepting a request to take out their loved ones.

# Requirements

## Development, Operation, and Maintenance Environments

The hardware and software that will be necessary to build and maintain the project would be a computer that is “decent” enough. A computer that has an Intel core i3+ plus processor or an equivalent processor from a different manufacturer with at least 2GB RAM with a hard drive/solid state drive of at least 100GB free is recommended. In terms of software, one would need at a minimum a Notepad++ for basic HTML, JavaScript, and CSS coding. Another software that one could use is Sublime Text2. Having multiple web browsers and internet connection is a must.

## System Model

### Textual Use Cases

Create Account

1. Create Profile

* 1. Username not available, try new username
  2. Password does not meet requirements, choose different password

2. Create Connections

2.1 Send out connection invites

2.2 Accept invited connections

2.3 Reject invited connections

Create Event

1. Select Create Event button
2. Select Event Type
   1. Select type potluck
   2. Select type outdoor activity
   3. Select type indoor activity
   4. Select type restaurant
3. Create Name
4. Create Date(s)
5. Invite
   1. Invite connections
   2. Make public

Login

1. Open site
2. Enter Password
   1. Incorrect password
      1. Reenter password
      2. Account Locked after 3 attempts
   2. Incorrect username
      1. Reenter username
3. Login

Donate

1. Login
2. Select recipient
   1. Select from connections
   2. Select donate to site
3. Donate funds
4. Recipient receives funds

(See Appendix A for use cases and Appendix B for system models)

## User Interaction

The program for the user is straight forward. The user will be able to interact with the website by seeing other people in their area that want to participate in the meal plan. Whether they be the host or looking for a family or a group to have a meal with. This information will be on the main page of the site for fast accessibility. The user will also be able to upload pictures or videos that they desire, that is related to organizing meals or giving other users an idea of what outdoor activities they are in to or content that has to do with their PIE experiences.

Inappropriate images or posts will not be tolerated neither do we condone such actions, this is not a dating or chatting site. The user will also have the ability to fill out a form for when they want to volunteer and host a meal or vice versa.

## Functional Requirements

### Primary Requirements

* We need to host the website on a server
* We need a database to store information for the forms
* We will have a calendar that will be used for clients to book days where they will be organizing meals or other outdoor activities.
* The website will be free for the time being until further development.

### Secondary Requirements

* Determining how users will pay for the services the website offer. WePay, Amazon pay, etc.
* Translating to different languages. For restaurants that are in different language.
* Get people to join the website.
* Testing yourself by booking you and your friends for software development
* We plan on having a third party service that we can implement to check the background of users/volunteers who get added on the site. This will be for security purposes.

## Nonfunctional Requirements

A user will be able to use our website for their needs anywhere in the world since it is in the internet. Our project is based to be used in the United States as far as the physical environment is concerned. Specifically, we will focus in Ellensburg for testing purposes but the functionality will not differ by city or state, except for the restaurants in that particular town and we can adapt to that by embedding a location service/GPS API. Currently our project does not have any other systems that it will interface on the actual website itself. We will have external links that will refer a user to either a restaurant or a third party website/program for background checking.

Conceptually, how efficient it will be in countries that are not as advanced as the United States is something that we cannot fully determine currently. Some of the factors that might hinder for our project to be something of value to others across is the world would be things such as: background checks for new users, paying system, and restaurants. But, if someone out of the country would like to assist their friend/family member in the United States, they would be able to do that with minimum limitations.

For nonfunctional requirements here is how the system will adapt to some of those instances:

* **Efficiency**- The system itself will be quite efficient in terms of providing the service that the user expects the program to provide as our website is catered to a few specific things that ensure the vision statement is met without too much jargon. The user interface is designed with the concept of simplicity and no learning curve. The text and diagrams are large enough so elderly people will have little to no problem to navigate through the site.
* **Reliability**- This will not be an issue for the user since our implementation is streamlined concisely with what is necessary. The only factor that may affect the system’s reliability is whether users actually sign up and interact with other users and arrange meals and outdoor activities, because if this is not accomplished, the site serves no purpose for the user.
* **Portability**- This won’t be a major issue since majority of users have access to internet. Whether that be via a cellphone or a computer. For those who may not have either, libraries, school labs, and cyber cafes are other considerable option. The only downside with a user accessing it on a cellphone is that all of the content may not fit on the screen or be displayed as it would have been on a computer. Further down the road, this will be something to consider, mobile version for our website.
* **Problem Size**- The problem size is big enough to make the project fun and challenging. For instance, we have to figure out how we will approach background checks without making the user feeling uncomfortable. We have to find a way of translating language in a reliable-efficient manner. We are certain that as we implement some of our core requirements, the problem size will be affected, hopefully not too drastically.

## Feasibility

In terms of feasibility, we are certain that we can finish our project by the end of winter quarter with no problem, but we are aware that some minor set backs are inevitable for various reasons. For example, feature overload is a possibility. Our client wants the website to have a lot of functions, but realistically, we will finish the primary requirements before working on the secondary to ensure a working product by the end of winter quarter.

# Quality Assurance Plan

## Document Standards

To ensure consistent and correct documentation occurs in all phases of the project, we have a role called “Documentation”. This role is assigned to one individual who is responsible for maintaining the documentation throughout the current iteration. This role is kept through the entirety of the project and will rotate among team members.

Currently, this role includes: documenting all meeting notes, posting them to the repository, ensuring that the repository folders are kept orderly, and creating meeting agendas. However, the PowerPoint document and the Iteration reports are the responsibility of all team members. They are a group effort where each member contributes.

The template for the Iteration reports were selected by the group and is the template we will use for all iteration reports. The meeting notes format has had minor changes in layout, but the general format has been consistent; noting start and end time, brief mention of topics discussed and details about the discussion of each topic.

## 

## Coding Standards

Our coding standards list is fairly basic. We developed the list after researching coding standards for other software projects and including rules that we believed would make the files easier to read. The following list contains the coding standards for our project. An example of code following these rules is included below.

* Semi-colon on end of all JavaScript lines.
* Use brackets to encapsulate nested JavaScript
* Close connection to database after use.
* Handle exceptions appropriately.
* Always use alt attribute with images.
* Avoid long code lines, max characters for one line about 80. Should not need to scroll page left-right to read code.
* Declare language at the head of all files.
* Use comments as needed. Do not overuse.
* Save with the right extension - .html, .css, .php, etc.
* Indent all lines four spaces, all nested code four additional spaces.
* Separate all code blocks with single empty line.
* Example code following guidelines from about.html
* All file names start with lowercase.
* No spaces in filenames
* If filename is multiple words, first word is lowercase additional words are uppercase. (i.e. signUp.html)

|  |
| --- |
| <head>  <title>About</title>  <link rel="stylesheet" type="text/css" href="template.css">  <link rel="stylesheet" type="text/css" href="buttons.css">  <div id='cssmenu'>  <ul>  <li><a href="home.html">Home</a></li>  <li><a href="about.html">About</a></li>  <li><a href="helpPage.html">Help</a></li>  </ul>  </div>    <h1>Making Outings as Easy as PIE</h1>  </head> |

Figure 2

## User Interface Guideline

The guidelines for our user interface were developed with our target demographic in mind. We are focusing on easy to see text, large buttons, and minimal side content. The site will be as easy to use as possible, while still providing extensive features.

The consistency of the website will be accomplished by using minimal templates, and reusing the basic ones (such as navigation bar at top of page) on each page. This will allow the user to be comfortable navigating each page of the site, as the key buttons/links will be in the same place no matter where they are on the site.

Since we are developing a site with a senior demographic in mind, we must assume that internet familiarity is minimal. Even more, we must assume that a significant portion of the users will have at least moderately impaired vision. These two assumptions have significantly shaped our user interface guidelines. They are the driving reason behind large buttons, large text and minimal unnecessary content.

## Change Control Process

Our group will continue to use GitHub as the control change as we find it easy to use and there is little to no learning curve with it.

The process our group will use to protect us from having unseen or creeping requirements is by continuing using Trello to set our priorities right. We have been setting deadlines on our core deadlines and keeping a tight communication to make sure that those requirements are taken care of. Every once in a while the requirements get altered due to unforeseen roadblocks that make it impossible to finish a particular requirement. In such instances we help each other to make sure that we have resolved everything and that the person who was in charge of that particular requirement can finish it.

## 

## Testing Process

Our team will have a multi-step process for testing including unit testing, automation testing, and user testing. This will ensure that we cover functional testing, do not add any bugs that previously existed, and that the product will work on multiple different platforms. We will first start by user testing with quality assurance’s personal computer and browser configurations. We will write down our configuration, and steps we did to produce a bug so that the developers can reproduce the bug and try to fix the code. After we have done this we will add the unit test to check and make sure that that bug is not introduced again after we add more features in our code. Then, once we have finished a unit test we will add it to our automation sweet so that every time we check in new code, our build agent (Teamcity) will run the unit tests against the new code using “Saucelabs” as the host the run the unit test among many different browser and operating system configurations. Doing this will make sure that we have covered unit testing, integration testing and system testing. We will continuously do these steps while developing our project to make sure that our program covered all aspects of testing.

In order to validate that our site meets the standards of our client Dr. Chueh, we would like to put our site through a series of acceptance tests. However, we do not have guidelines clearly defined. We were given free rein on design without clear standards for the site’s functionality. The only major requirement was to keep her vision intact. To ensure that our design is not straying from this, we will demo our prototype to Dr. Chueh every two weeks during our meetings.

There were however, some features that she asked be included, such as the ability to share photos and create events. These functions will be included in the site, but the testing for them is rather straightforward. Their functionality will be tested during unit and integration testing.

# Conclusion

For next term we would like to integrate a calendar to our website. The calendar will be able to show the days that there are events going on. This could range to simple things such as someone wants to do some social activities like walk downtown and wants a companion or a couple of people are going to have lunch at a local restaurant. We also want to be able to use our database to store user’s information such as usernames and passwords since we haven’t been able to fully implement this tool. We are planning on implementing a way to make the site safer for users, we are considering having some sought of background check to make ensure security of all users who will be meeting other people online and to weed out fake profiles. We will also like to start conversations with local restaurants and see if we can partner up and promote their pages on our website.

## Appendix A

### Use Cases









## Appendix B

### System Models

Create Connections



Create Event

Create Profile



Donate



Login

