Tools for Supporting Community Growth in Open Source

CS463: Final Report Spring 2016

Bruntmyer J. Author, OSU, Goossens M. Author, OSU, Nguyen H. Author, OSU

Abstract

For the past six months our group has been working on a project that is creating tools that gives users the ability to look for open source community leaders that are hosting events. These tools will allow users to have the opportunity to find these events in order to become a contributor to an open source project. This is done in the form of a website that will have features for finding certain events dealing with open source projects so that it can be easily accessible by people with a passion for wanting to contribute to projects. Throughout this document, we look at what this team has accomplished for each of the requirements that have been laid out, discussing problems that have halted our progression through the project, and how we changed our timeline. Also included are important images of the user interface we have decided to use, along with pieces of code that we have completed. By the end of this document, you will get a complete picture of how we reached our version 1.0 release.

I. INTRODUCTION TO THE PROJECT

The Community Driven Development project is sponsored by the Apache Software Foundation under Ross Gardler, Director and President of the company. The project was designed and prototyped by Gardler before being proposed to the senior software project class for Oregon State University. Development for the project hopes to achieve more involvement in terms of helping building upon the Open Source Community and helping it grow and gain more traffic. The prototype was regularly used by Gardler and many of the faculty from Apache and is continued to gain interest to the public. The importance of development of the project will promote more growth and allow more tools for users in the community to find and locate more events and get more involved in open source community groups and projects. The students assigned to work on this project were Justin Bruntmyer, Megan Goossens, and Hai Nguyen. Each member took upon equal and separate roles for the project. No specific roles were assigned as each member took part in doing part of the work for every task. Ross Gardler provided the initial prototype while the project was improved and implemented with the requested features as the term progressed.

CONTENTS

II. Requirements Document

A. Introduction to the Problem

Open source projects are known for the code that is developed from them however that is not what causes these projects to either live or die. Though it is true that an open source group can survive without a viable community in most cases there is a class of open source projects that lives or dies on the strength of its community such as the Apache Software Foundation projects. Existing tools are in place for tracking activity within online communities however these tools focus on identifying customers to the open source projects. The problem that is being focused on is that new tools need to be made to identify community leaders so that community builders know who to engage with.

B. Project Description

The goal of this project is to build connections, collaborations, and identifications between community members and leaders in open source projects. This entails contributing to Apache Software Foundations existing software platform to create tools that identify and support community leaders and members. Better tools will help identify potential community leaders which will then allow community builders to engage with and support those leaders. Over the course of the project, consistent updates check what tools were used as intended and which ones could be improved and expanded. Evaluation of the tools implemented is done in order to measure its significance and effectiveness.

The solution to this problem should be able to identify community leaders to people with a desire to improve the strength of a community around a specific project. This solution includes being able to observe Apache Software Foundation community activity on publicly available information such as meetups.com in order to identify those potential community leaders. The solution is to provide enough information to community builders to contact and engage with the leaders so that they can begin their own contributions to the projects.

C. Requirement

The end goal of our project is to create a tool for identifying community leaders for others to see and given the chance to get involved with by having a web page that will let users go through and view all of this data in one location. This data will be gathered from an algorithm that reaches out to meetups.com and use key terms to find community leaders. As a stretch goal we would like to have this algorithm gain data from other social media sites such as Facebook and Twitter.

We will be starting with a prototype that has been implemented which means we will need to start by fixing what doesn't work and then adding key features that we believe the project needs. In a final delivery there will be a patched version of the prototype with tool fixes along with new features to improve usability.

REQ#	Requirement	Priority	Expected Completion
1	Gain access to source code of the Community Developments page	HIGH	11/30/2015
2	Fix the ”People” page where the list of community leaders are shown	HIGH	01/11/2016
3	Tweet at a person listed i the database	LOW	03/14/2016
4	Add user accounts to the application and track when a user has tweeted an event	MEDIUM	02/15/2016
5	List tweets about events and/or people via the app	MEDIUM	02/22/2016
6	List tweets about events and/or people from twitter, but not via the app	MEDIUM	02/29/2016
7	Export a list of people with information	LOW	03/07/2016
8	Improve hashtag searching of application for better results on relevant events	HIGH	02/08/2016*
9	Alpha Release		02/08/2016
10	Improve the visuals of the tool looks as a whole	HIGH	02/01/2016
11	Beta Release		03/14/2016
12	Implement a system of finding events nearby a location entered or within radius of the user	HIGH	02/08/2016*
13	Add feature to generate a profile for community developers to have contacting information easily view-able	HIGH	01/25/2016
14	1.0 Release		05/20/2016

*We expect this task to be more difficult, so we will work on the task in parallel to other tasks during that time.

D. Alpha/Beta/1.0 Releases

We plan to release the Alpha version of the project during week 6 of winter term where we will have multiple fixes complete for the tools that are already implemented in to the project that we are starting with. This will mean that the Alpha version will be a major patch on what is created already in order to improve the functionality of the current implementation. We plan to release the Beta version of the project during finals week of winter term where we will implement new tools for community leaders to manage their posts along with some other useful tools. Here we plan to have almost all of the functionality ready. Lastly version 1.0 will be released on May 16, 2016 which is the Monday prior to the engineering expo. This version will be our final release and should be bug free in the implementation of our project.

E. Specifications of Requirements

1) Gain access to source code of the Community Developments page

1.1. Gain access to the source code for the prototype to begin working.
1.2. This will be given by client as we will be collaborating with him on testing the functionality of the prototype.

2) Fix the People page where the list of community leaders are shown

2.1. Currently takes a large amount of time in order for the ”people” page to load as it shows the current people who are identified as community leaders.
2.2. Figure out why the page takes such a long time to load. Could be too much data and the algorithm for pulling up this page needs to be changed.
2.3. Need to debug the causes of crashes by some browsers.

3) Tweet at a person listed in the database.

3.1. There will be a button that a user can select to send a tweet to a person directly.

3.2. This will be sending the user to the twitter API in order for them to construct and send a tweet.

4) Add user accounts to the application and track when a user has tweeted an event

4.1. This is so they don't inadvertently do it twice
4.2. Track date of tweet, tweet text and tweet ID
4.3. This enables the prototype to branch out further than meetups.com
4.4. Allows user to see what tweets they have tweeted about

5) List tweets about events and/or people via the application

5.1. This should work for the current user and others
5.2. Allows retweet or sending as new tweet based on existing one
5.3. Have a button that will list the tweets that were tweeted via the application.

6) Export a list of people and email addresses

6.1. The emails corresponding with the people will be exported as well
6.2. Will be in JSON, CSV and test format for easy consumption in other applications

7) Improve hashtag searching of application to improve finding relevant events

7.1. The searching algorithm is currently returning data from meetups.com with searching for tag ”apache” however this is not working correctly as it is pulling information that has nothing to do with open source projects and is very random.
7.2. The searching algorithm will need to be fixed so that it find the correct tags from meetups.com and pull the accurate data.
7.3. The stretch goal here is the be able to use this algorithm on multiple social media networks.

8) Alpha Release

8.1. Alpha to be released the 6th week of winter term.
8.2. This version will have some fixes for the improvement on algorithm, mark event action, mark group action, import members action, import meetups action, and the people action.
8.3. If not completely fixed/implanted the project will at least handle them by throwing a message to the user saying this tool is under construction.

9) Improve the visuals of the tool looks as a whole

9.1. Make the user interface look sharper and more appealing
9.2. Have the events and people information that is displayed when a user decides to look into more details about a person or even be more appealing as well.

10) Beta Release

10.1. The Beta of the project will be released on finals week of winter term.
10.2. In the Beta the fixes from the fist Alpha should all be fixed correctly, and all of the tools that we wanted to add should be implemented but not necessary working correctly, at least has some functionality.

11) Implement system of improved sorting of finding events by nearby location within radius of the user

11.1. The program will be able to get the location from the user and based on this location the nearest community leader events will be shown to the user.

12) Add feature to generate a profile for community developers to have contacting information easily viewable

12.1. Based on the data collected from meetups.com to identify the community leader not only will the events be posted but the community leader will have a section where the user can easily view the contact information pulled from meetups.com. A link with their name.
12.2. This is not a collaboration communication tool but just a way to view the contact information.

13) 1.0 Release.

13.1. This will be the final release where all of our implementations should be working correctly with no bugs.
13.2. This will be released on May 16, 2016 which is the Monday prior to the Expo.

F. Risk Assessment

1) Miscommunication on can happen in between the Community Development team currently working on the project.
2) Security issues, unwanted changes to source code from unauthorized individuals
3) Algorithms created cause even slower computation time for certain aspects of the website
4) If certain implementation is added and is miscommunicated to a function that was not originally discussed
5) Pulling data from meetups could be corrupted
6) Code does not match with already implemented coding standards on the project

G. Stretch Goals

1) Create a notification system to alert users of important events that the user may be interested in.
2) Create profile system for users and community developers
3) Create a notification system to alert users of important events that the user may be interested in
4) The project will have a tool that will allow a user to enter their email and they will be notified when a project event is added that they may be interested in.

4.1. This could also be implemented to send emails to users that sign up for them to a certain community leader for a certain project rather than trying to figure out what they might be interested in.

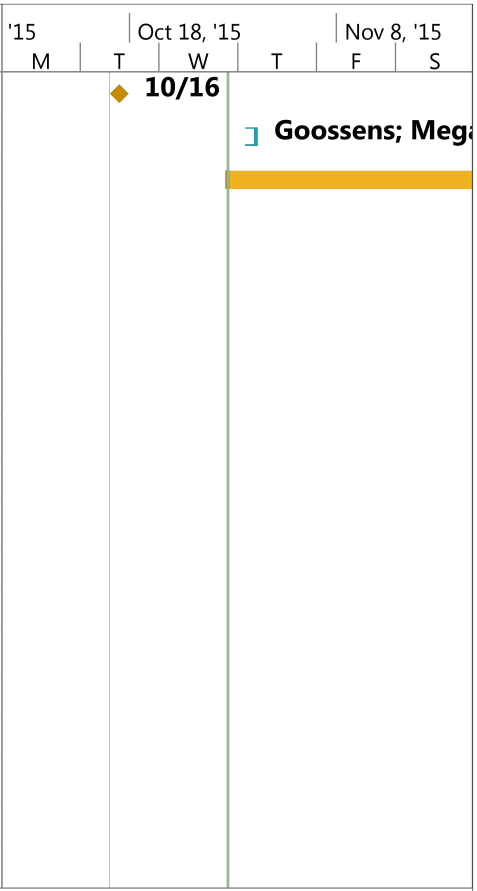
5) Fix ”Import Members to allow importing of members
6) Allow community developers to edit their own events for specific messages
7) Create profile system for users and community developers

H. Time Table Gant Chart

The gantt chart below shows the timeline we have set for our group to meet the requirements proposed in this document. We will be trying to fulfill this timeline as much as possible in a timely manner as it gives us a great path towards completing this project.

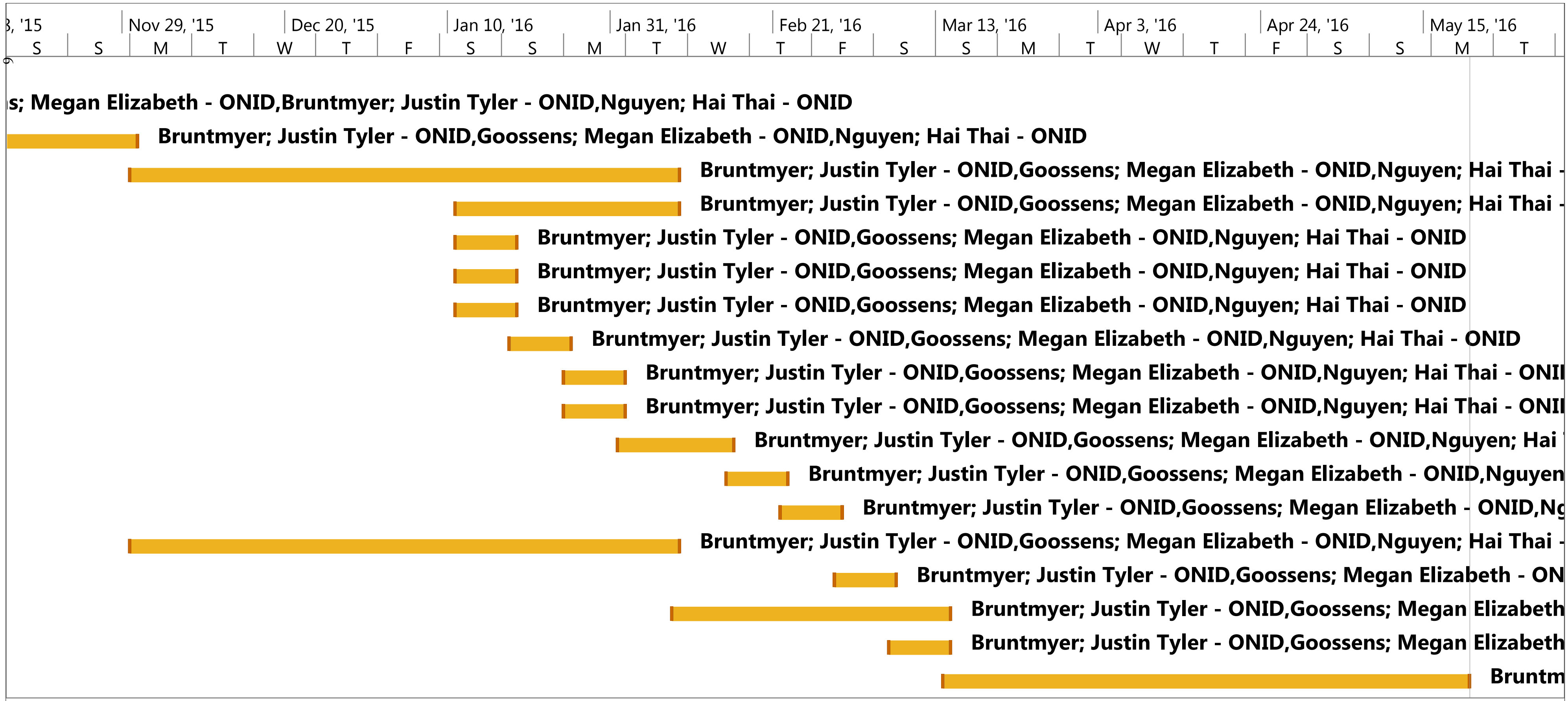
See gantt chart on next page.

ID	ℹ	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1	✓	📌	Problem Statement	0 days	Fri 10/16/15	Fri 10/16/15		Bruntmyer; Justin
2		📌?	Requirements Docum			Fri 10/30/15		Goossens; Megan
3	👤	📌	Gain access to source	24 days	Wed 10/28/1	Mon 11/30/15		Bruntmyer; Justin
4	👤	📌	Implement system of	51 days	Mon 11/30/1	Mon 2/8/16		Bruntmyer; Justin
5	👤	📌	Fix the "People" actio	21 days	Mon 1/11/16	Mon 2/8/16		Bruntmyer; Justin
6	👤	📌	Add user accounts to t	6 days	Mon 1/11/16	Mon 1/18/16		Bruntmyer; Justin
7	👤	📌	List tweets about ever	6 days	Mon 1/11/16	Mon 1/18/16		Bruntmyer; Justin
8	👤	📌	List tweets about ever	6 days	Mon 1/11/16	Mon 1/18/16		Bruntmyer; Justin
9	👤	📌	Export a list of people	6 days	Mon 1/18/16	Mon 1/25/16		Bruntmyer; Justin
10	👤	📌	Add feature to genera	6 days	Mon 1/25/16	Mon 2/1/16		Bruntmyer; Justin
11	👤	📌	Tweet at a person list	6 days	Mon 1/25/16	Mon 2/1/16		Bruntmyer; Justin
12	👤	📌	Improve viewing meth	11 days	Mon 2/1/16	Mon 2/15/16		Bruntmyer; Justin
13	👤	📌	Improve searching alg	6 days	Mon 2/15/16	Mon 2/22/16		Bruntmyer; Justin
14	👤	📌	Fix "Import Members'	6 days	Mon 2/22/16	Mon 2/29/16		Bruntmyer; Justin
15	👤	📌	Alpha Release	51 days	Mon 11/30/1	Mon 2/8/16		Bruntmyer; Justin
16	👤	📌	Allow community dev	6 days	Mon 2/29/16	Mon 3/7/16		Bruntmyer; Justin
17	👤	📌	Beta Release	26 days	Mon 2/8/16	Mon 3/14/16		Bruntmyer; Justin
18	👤	📌	Create profile system	6 days	Mon 3/7/16	Mon 3/14/16		Bruntmyer; Justin
19	👤	📌	1.0 Release	50 days	Mon 3/14/16	Fri 5/20/16		Bruntmyer; Justin



Project: Project3
Date: Wed 10/28/15

Task	▬	Inactive Summary	▭	External Tasks	▬
Split	┈	Manual Task	▬	External Milestone	◇
Milestone	◆	Duration-only	▬	Deadline	⇩
Summary	┌┐	Manual Summary Rollup	▬	Progress	─
Project Summary	┌┐	Manual Summary	┌┐	Manual Progress	─
Inactive Task	▭	Start-only	[		
Inactive Milestone	◇	Finish-only	]		



Project: Project3
Date: Wed 10/28/15

Task	▬	Inactive Summary	⌐¬	External Tasks	▬
Split	┈	Manual Task	▬	External Milestone	◇
Milestone	◆	Duration-only	▭	Deadline	↓
Summary	⌐¬	Manual Summary Rollup	▬	Progress	─
Project Summary	⌐¬	Manual Summary	⌐¬	Manual Progress	─
Inactive Task	▭	Start-only	[		
Inactive Milestone	◇	Finish-only	]		

III. CHANGES TO THE REQUIREMENTS

After continuous work on this project, the perspective on what this community development tool wants to accomplish became clear. After taking more time to read and understand the code that was previously written for the prototype, we can see clearly how the data is organized and connected using Django and its tools. First and foremost, the purpose of the community development tool remains the same. The purpose is to gather information from meetup.com which is the website we are pulling information about events and people from. Next we parse that information into a list of upcoming events related to Apache and open source projects so that developers in the open source community have an easy way to access an environment where they can hope to participate in those events. The ultimate goal of this project is to create a set of tools that eager developers can use to find events, view community leader profiles, and get involved.

REQ#	Requirement	What Happened To It	Comments
1	Fix the ”People” page where the list of people are shown from groups	The People page currently takes all of the people in the database and lists them onto the page. Normally, if the prototype is hosted on a local machine and the database is relatively small, then the page loads fine in a minimal amount of time. The issue is nested in the actual hosted site by Apache where hundreds of thousands people are imported into the database daily and dramatically slowing down the loading time of the page. With our current progression of the project, we have not made significant progress into improving the loading time of the page. We use our own local host to import a small amount of members at a time and that requirement is set to be worked on shortly for Beta implementation. The guideline for working towards accomplishing this requirement is to limit the amount of people loaded at a time onto the page. For Beta, we have rearranged where the table is generated for the people page in the function within views.py. This change specifically was introduced because a bug was found where when the table is generated, then if the amount of people were too many, then the table would crash and not build. With the rearrangement, now the table does not break through a large build and now tends to load faster. Unfortunately, the implementation of the fix for the People page is local. We have yet to test it on the host that Apache is using to run the prototype currently. We predict that the fix will work but it will need to be approved and patched into the live site for clarification.	The biggest change made for this requirement is realizing that the problem was in the prototype as the people page was trying to display a large amount of tables that was causing the browser to crash. The bug was found and taken out of the code and the people page now loads normally.
2	Tweet at a person listed in database	Each person who is imported into the application is generated their own profile page based off of their Meetups ID. Information from Meetups about the persons profile is also parsed in the community development tool. Those profiles include displaying the twitter handle of the person. This was done by changing the Meetup API request so that we could get the correct information and then store that information in our database. This can be seen in code snippet 1 located below with the URL shown along with the ’if’ statements to locate the twitter handle. This allows the user to get in contact with the person in a profile. Above the twitter handle is a button that has the Twitter symbol which allows the user to click and send a tweet at the person via Hoot-suite. The hoot-suite app is given the twitter handle of the person the user wants to tweet at and the URL of the persons page on our application for reference. The user signs in to compose the tweet and sends it under their Twitter account. The code snippet 2 located below shows the HTML encoding of the button used to create the Hootsuite connection and shows the retrieval of the twitter handle form the database with ’person.service’. Not all users have a Twitter handle registered with Meetups thus the tweet button does not have any use. To handle this case the tweet at button only appears on profiles of imported people that have a registered Twitter handle.	blah

3	Add user accounts to the application and track when a user has tweeted an event	The purpose of adding user accounts to the applications is to be able to track when users tweeting about events or people. With the completion of this requirement, the account creation is working along with being able to sign in successfully with a confirmation of signing in by displaying a welcome message along with the user's username. There is also a login and logout button located in the top right section of the website on all pages allowing the user to login or logout at anytime. Everything is also backended with features of the website only existing if the user has an authorized account. This means that action functions within the application which include importing meetups, importing members, marking events as not applicable, and marking groups as not applicable to be behind being signed in. This means if you are not logged in with the authorized account, you can't perform these actions. Note that the accounts created are allowed access to these features, but do not have access to the administrative page that deals with the Django database.	blah
4	List tweets about events and/or people via the app	We were unable to track the precise tweets made from our application, but we have found an alternative that mostly works. As shown in the code snippet, the website makes a call to Twitter's search API, requesting all tweets that contain a certain hashtag as well as the hashtag #Meetup. The hashtags are the same as the ones used to get events from meetup.com. Once it has the tweets, it uses the id from each one to make another call to Twitter's OEmbed API, which sends back HTML that is used in the page's template to present embedded tweets to users. This still pulls a few tweets that are unrelated, but bit of filtering would work. Unfortunately this would be difficult, and is outside the scope of our project.	blah
5	List tweets about events and/or people from twitter, but not via the app	The website now has a tweet parser. As shown in the code snippet, it sends a call to Twitter's search API, requesting all tweets with a certain hashtag. The hashtags are the same as the ones used to get events from meetup.com. Once it has the tweets, it uses the id from each one to make another call to Twitter's OEmbed API, which sends back HTML that is used in the page's template to present embedded tweets to users. The search results currently contain all instances of the hashtag requested, even when they are not relevant to any event, or even open source. The search needs refinement, however, this will be difficult, and is not within the scope of the project	blah
6	Export a list of people with information	When taking on this requirement we quickly realized that the Meetup API would not provide email address for its users which was understandable. We then looked at what other information would be useful to extract about the people that were loaded into the database. This lead us to export information such as name, twitter handle, bio, Meetup ID, URL, country, state, and city. The export can be executed by clicking on the 'Export Info' button located in the top left of the people page and creates a file in a XLSX format which can be directly opened or saved.	blah
7	Improve hashtag searching of application for better results on relevant events	The solution to this issue was to change one word in the call to the meetup.com API. In the API there are many different restrictions you can use to request events. Two of them come into play here: "text" and "topic." The text query searches through the content of the events, while the topic query looks at the topics related to the group. The original query was looking at text, which resulted in a lot of unrelated events. The query was fixed to use "topic," which has reduced unwanted events significantly. A thourough examination of events imported resulted in no unrelated events pulled in.	blah

8	Implement a system of finding events nearby a location entered or within radius of the user	This tool creates a list of all events parsed through the application and displays a marker for each even onto the Google Map displayed to the right of the lists of events. A user can search for a specific event by looking through the list and then seeing where this is on the map. This feature also asks the web browser of the user for the geo-location of the user in order to place a special marker on the map showing where the user is in reference to the rest of the markers. The user can decide weather or not to accept giving the their location to the application. The map is generated through Google Maps by using the Google Map API calls. This map was implemented with a search bar allowing the user to search for a location and see what events are in that location. With each search the map jumps to the searched location and is given a 200 mile radius circle to show what events are within 200 miles of the user. In order to get the markers of each event to show up on the map the latitude and longitude of each event needed to be stored and accessed by the map. This was done by a Meetups API call as shown in code snippet 3. Once the API call is made the json object is returned and parsed to obtain information on the event including name, longitude, and latitude. Once the information is stored it is accessed in the JavaScript for the Google Map which can be seen in code snippet 4 below. This is done by looping though a list of events and gather the name, longitude, and latitudes in order for the markers to display. When a user hovers over a marker on the map the name of the event is shown. When there are no events imported a message stating ”No Events Available” is displayed.	blah
9	Add feature to generate a profile for community developers to have contacting information easily visable	The main goal of the tool is to promote community development in the open source scene. What this feature tackles is a way to display important information about people that are already involved with projects to those who would like to get involved. This feature generates profiles for event hosts along with members of groups that have been imported by the user. These profiles consist of the of selected person, Twitter handle, location, link to Meetups profile, last activity date, group associated with, topics they are interested in, a biography, and a picture of themselves. In order to gather the information for these we had to make another API call to Meetups. First, we had to adjust the current API call for importing events to also gather the ID of the event hosts. Once this ID was obtained the next step is another API call gathering information on each event hosts while searching with the ID’s gathered. Once this information was obtained the profiles for event hosts and imported group members could be displayed in their own sections of the application. With users having the options to see group members and event hosts there are more opportunities for these users to get involved.	blah

10	Improve the visuals of the tool and how it looks as a whole.	The application itself began as a fairly organized piece. The navigation bar implementation really helps the user keep track navigating between each page. When viewing the list of events, the events are listed in chronological order starting with the most recent. There is a search bar available for the user to type in for a certain event that they wish to view. There are other sorting mechanisms to view those events in another sorting order. Our focus in this requirement were to mainly to improve how data is displayed. This specifically applies to to the event page and people profiles. Along with the addition of the Host objects, the generated host profiles would have a visual update as well. As displayed in Figure 9, the improvement of the profile page is shown with categories specifically labelled and displayed in concrete areas of the page. If the certain variable does not exist, then the user can clearly see where that detail would go on the page. Topics are more clearly organized have are configured under a scrollable table as well. Overall, this addition makes the people profile page much more organized and legible. In addition to the updated people profile page, the event page is also upgraded in Figure 10. Similar to the profile upgrade, the event page now has categories that specifically detail where objects belong. This is much more particularly important where the description is displayed for the event. Previously, it was much more difficult for the user's eye to view where the venue of event would be. Now in this update, that portion is clearly labelled which allows the user to spend less time reading and to quickly see the information that they want to.	blah

A. Updated Gantt Chart

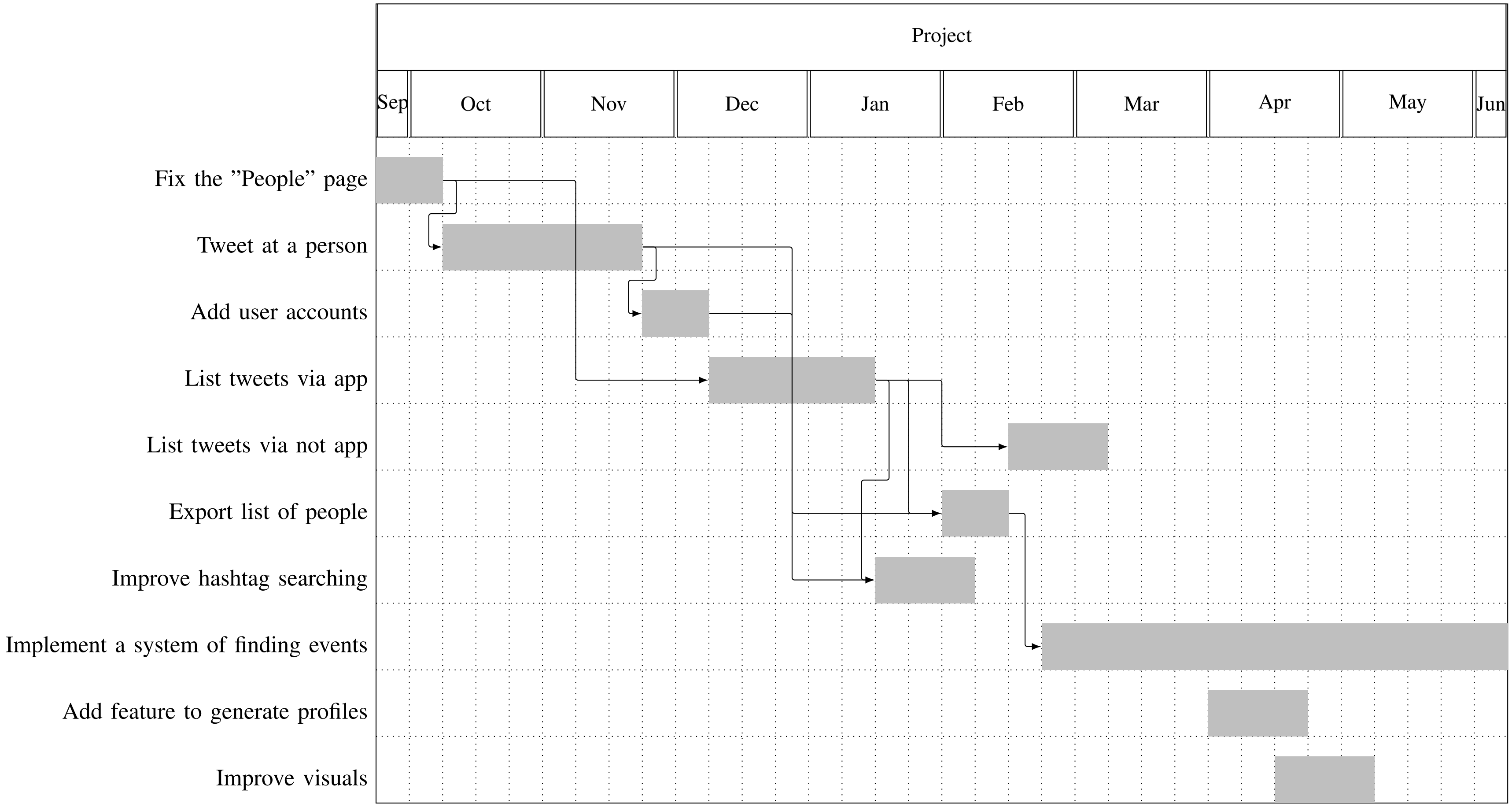


Fig. 1: Gantt Chart - Workplan

IV. Design Document

A. *Frontpiece*

1) Date of Issue and Status
The design of the project was issued on October 4 th , 2015. The status of the project was already in production with a prototype.

2) Issuing Organization
The Apache Software Foundation originally began development of the project and issued the proposal to Oregon State University for continued development. This entails to contributing to Apache Software Foundations existing software platform to create tools that identify and support community leaders and members.

3) Authorship
Current developers contributing to the community tool include Ross Gardler, Justin Bruntmyer, Hai Nguyen, and Megan Goossens. Ross Gardler, president of Apache, is the original developer for the project.

4) Change History

B. *Introduction*

1) Purpose
The primary purpose of this document is to present a detailed description of the design elements of the Getting Connected: Tools for the Open Source Community project. This will guide the group in the design of the application.

2) Scope
This document will provide details on the design of the various technologies of the web application, in particular the Django framework, PostgreSQL, HTML user interface, and the user and group authentication system provided by Django.

   The user will have the ability to view the community leaders that are posting events on social media sites based on tags fetched by the tools. This will allow the user to see the activity and contact information of the community leader to potentially contact if the user is interested in contributing. The user will be able to view a summary of the event as well as have a link to the source of the event hosted by the community leader.

3) Context
This project will be free to access for everyone. Development and maintenance will have not cost as this project is open source. Future development plans will be based on the features that do not make it in the 1.0 release of the application. There are several stretch goals that will potentially be incorporated. These features are not covered in this document.

4) Summary
This document will go over the design for the aspects of the project including web framework, database design, user interface, and authentication system.

C. *References*

1) IEEE Standard for Information Technology–Systems Design–Software Design Descriptions, 1st ed. IEEE, 2009.

2) Comdev1-us-west.apache.org, 'Community Development: Events', 2015. [Online]. Available: http://comdev1-us- west.apache.org/events/. [Accessed: 03- Dec- 2015].

3) Djangoproject.com, 'The Web framework for perfectionists with deadlines — Django', 2015. [Online]. Available: https://www.djangoproject.com/. [Accessed: 03- Dec- 2015].

4) W3.org, 'HTML5', 2015. [Online]. Available: http://www.w3.org/TR/html/. [Accessed: 03- Dec- 2015].

5) Postgresql.org, 'PostgreSQL: The world's most advanced open source database', 2015. [Online]. Available: http://www.postgresql.org/. [Accessed: 03- Dec- 2015].

D. Glossary

Term	Definition
API	Application Programming Interface
Query	Request sent to a database
Query Parameter	Argument sent to a database to refine a search
Community Leader	Someone who is in charge of a project or organization
Tags	Keywords - e.g. ”apache,” ”open source,” etc.
HTML	HyperText Markup Language
Django	Web framework
PostgreSQL	Database management system
Meetup	Meetup.com

E. Body

1) Identified Stakeholders and Design Concerns

For the stakeholders, the tools support and help the Apache Software Foundation. Specifically, Ross Gardler is the main stakeholder that is part of the design.

Concerns regarding the design of the project include completions of certain implementations with the requirements listed. Troubleshooting problems or possible encounters that may appear during implementation are listed as such.

2) Algorithm Viewpoint

2.1. Fix the ”People” page where profiles are not implemented

As a high priority, fixing any type of design bug that was already implemented in the prototype is important. The page listed currently takes a very long time as a link from extracting all data from the PostgreSQL database causes difficulty in the time frame. Possible ways to approach in this solution include:

- Taking a look at how the data is sorted
- Looking at the users list and what is included in ”People”
- Changing the algorithm in the time frame to extract the data.

2.2. Design View

Our fixing the people page viewpoint shows how the application will be interacting with the Django framework along with the database. We know that the button on the webpage will be an HTML button that will be selected by the user. Once selected the Django framework will be activated to query the database for the current list of community leaders. Once the database finishes the query it will send the list back to the HTML page to be displayed for the user.

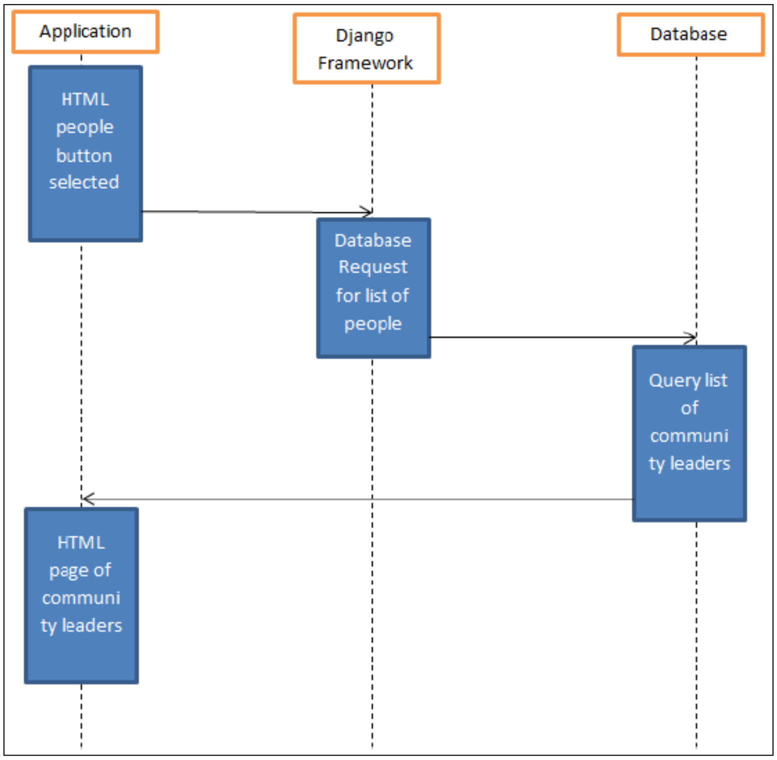

Application
Django Framework
Database
HTML people button selected
Database Request for list of people
Query list of communi ty leaders
HTML page of communi ty leaders

Fig. 2: Sequence diagram of how the application will interact with the database and display the people page.

2.3. Design Rationale

A look at how the various steps for loading the people page is necessary due to the fact that it is not obvious as to why the page is not loading properly. As there are several possible sources for the slowness, all of them must be examined to be able to determine the cause or causes. Once the problem has been pinpointed, the next step is to work on optimizing it, whether that involves changing how the data is stored, changing the query that fetches the data, or changing the algorithm that processes and presents the fetched data.

2.4. Improve searching algorithm of tool for more relevant events

The way the events are listed and how relevant they are to the user is important. With the prototype, events currently being extracted have the ability to not even be related to open source or programming itself. Implementing a new searching algorithm will be difficult and could cause a lot of troubleshooting listed below:

- Improving methods of recognizing relevant tags and keywords
- Removing any event that has no relation to open source projects
- Negative search words
- Analyse existing non-applicable groups

2.5. Design View

Our improvement with the searching algorithm viewpoint is shown below and mainly shows how the application will work with an efficient algorithm for searching the events. With the correct algorithm the Django framework will periodically query the meetups public posts searching for tags that have to do with the open source projects community. Once it returns those events they will be sent to the database and be stored for the HTML request will be sent to the Django framework where the query will be sent to the database to show the list of events on the application via HTML.

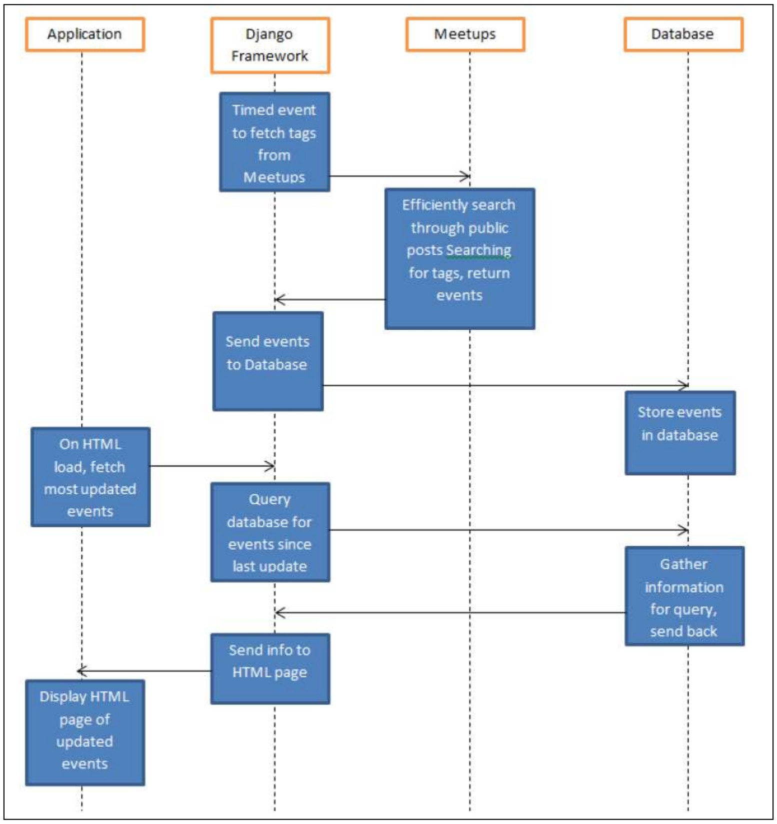

Application
Django Framework
Meetups
Database
Timed event to fetch tags from Meetups
Efficiently search through public posts Searching for tags, return events
Send events to Database
Store events in database
On HTML load, fetch most updated events
Query database for events since last update
Gather information for query, send back
Send info to HTML page
Display HTML page of updated events

Fig. 3: Sequence diagram showing the interactions between the application, django framework, meetups.com, and the database for improving the searching algorithm of events.

2.6. Design Rationale
There are two parts to this viewpoint: The fetching of events from Meetup, and showing the events to the user. Getting the events from Meetup will be a regular, scheduled event that will happen at appropriate intervals. Constant fetching from Meetup, or fetching every time the events page is loaded, is impractical and would slow the website down considerably. Once the most recent set of events has been stored in the database, it is a relatively simple matter to fetch them from the database when the events page is loaded.

3) Context Viewpoint

3.1. Tweet at a person listed in the database
Implementing another social media aspect other than just meetups.com is important for the project. This tool looks to connect every various source of data and also be able to give the user a tool to communicate with either a community leader, or a current user working on the project. Currently the requirement priority is listed as low, but possible issues for this requirement include:

- The time frame of learning how to implement different social media
- Implementing the feature itself integrating with already registered twitter accounts

3.2. Design Viewpoint
Our viewpoint for the ability to tweet at a community leader is shown below and has details regarding communication between the application, Django framework and the database. There will be an HTML button that can be pressed by the user and when this is pressed the Django framework will query the database for that community leaders twitter user name which will then be sent back to the Django framework. From here the Django framework will create the necessary query parameters that will be sent to the Twitter API. Once this is done the user should see a new page with the Twitter API already having the username ready and they can just enter the tweet they would like to send.

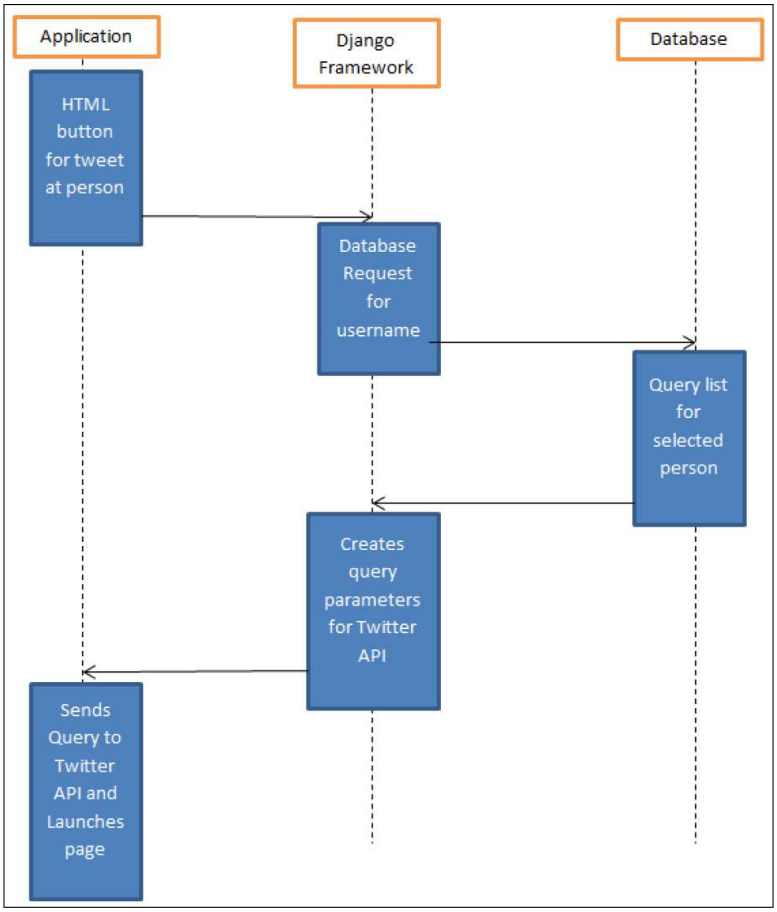

Application
Django Framework
Database
HTML button for tweet at person
Database Request for username
Query list for selected person
Creates query parameters for Twitter API
Sends Query to Twitter API and Launches page

Fig. 4: The sequence diagram for the application querying the database for the twitter handle to tweet at a single person.

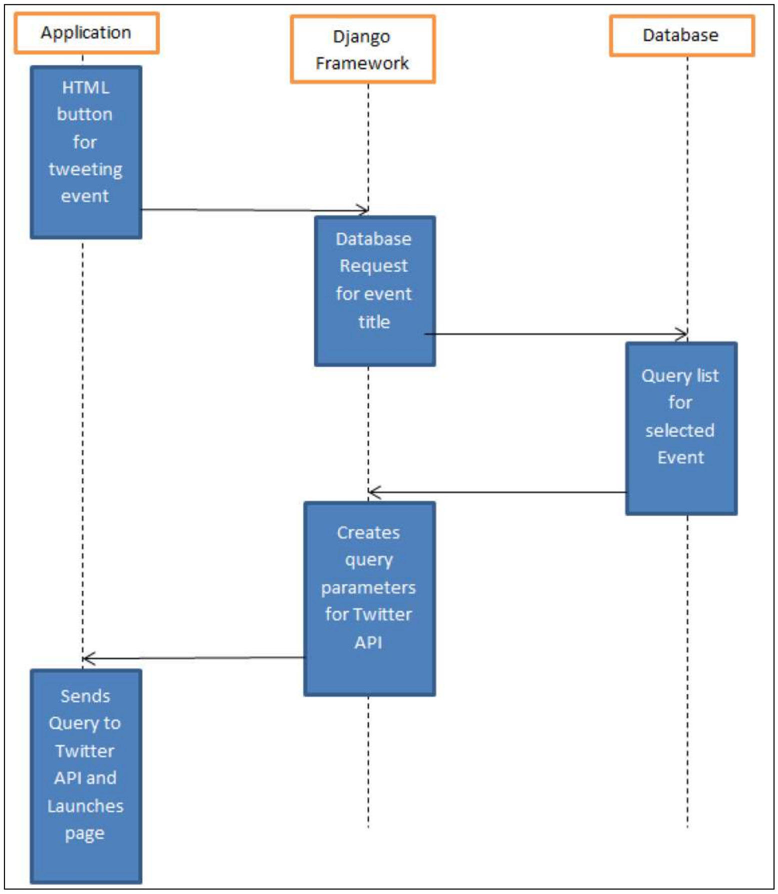

Application
Django Framework
Database
HTML button for tweeting event
Database Request for event title
Query list for selected Event
Creates query parameters for Twitter API
Sends Query to Twitter API and Launches page

Fig. 5: The above sequence diagram shows the same process but instead of tweeting directly at a person the user is tweeting to a group.

3.3. Design Rationale

Twitter has a way to create tweets for you with the information you wish to be contained in them. Using their API, a pre-constructed tweet can be constructed. Every person and group's Twitter handles will bestored in the database, so a query will need to be made to retrieve that so that it can be added to the tweet, along with whatever default message is decided upon.

4) Information Viewpoint

4.1. Add user accounts and track when a user has tweeted an event

In hopes to increase social media feed, the tool looks to integrate movement with community leaders and help users consistently track their activity. As a medium priority, the most concerning implementation is to correctly link the account with the correct user and to distinguish their own Twitter accounts with the correct user account on the website. (Hoot Suite).

4.2. Design View

Our viewpoint for creating user accounts is shown below with the communication between the application, Django framework, and the database. Here the user will select the create account HTML button which will begin the actions of the Django authentication system for creating accounts. This will bring a form to the application and present using HTML to the user who will enter in the required fields. Once submitted the Django authentication system will check the fields and then send it to the database for account creation. Once this is successful the application will display a message stating the account has been created successfully.

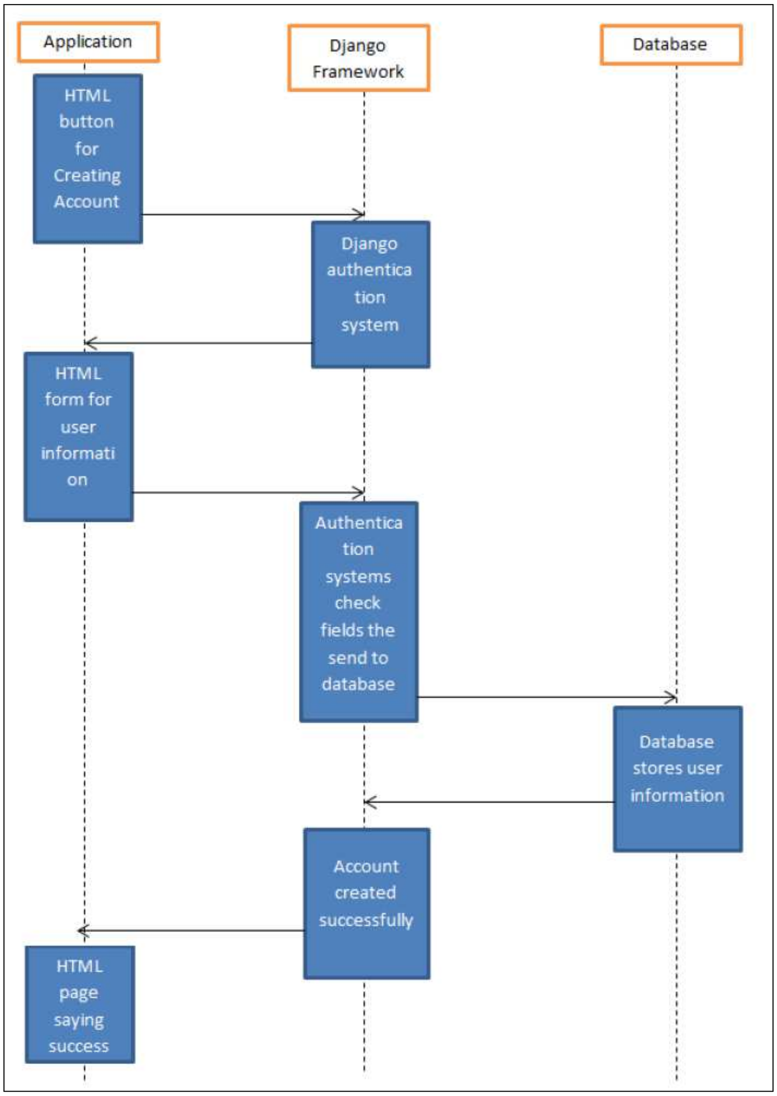

Application
Django Framework
Database
HTML button for Creating Account
Django authentica tion system
HTML form for user informati on
Authentica tion systems check fields the send to database
Database stores user information
Account created successfully
HTML page saying success

Fig. 6: The message sequence showing the application interacting with the database in order to create accounts and log users into the application.

4.3. Design Rationale

As the application is build in Django, we will be using Django's authentication system for user accounts. It is the best choice for this project because it is simple and already implemented within the Django platform.

5) Interface Viewpoint

5.1. Improve viewing method to examine the event itself

Improving the way users view the website is important to make it look more pristine and clean. All information should be clearly sectioned and any relevant information that the user would like to know should be put in a convenient viewing form.

- Categorizing certain data could turn to an issue in organizing the data
- Some data will be missing and handling that missing information in the event should still give the user enough information to look for it

5.2. Design View

Our viewpoint for the improve viewing method requirement that we will be implementing is shown below. This shows the layout of how the information should be spread out through the website. This is a basic look at how each location will store data for the user to easily navigate through in order to find what they are searching for. There are also a couple of links that are described for the tweet at someone or tweet at an event.

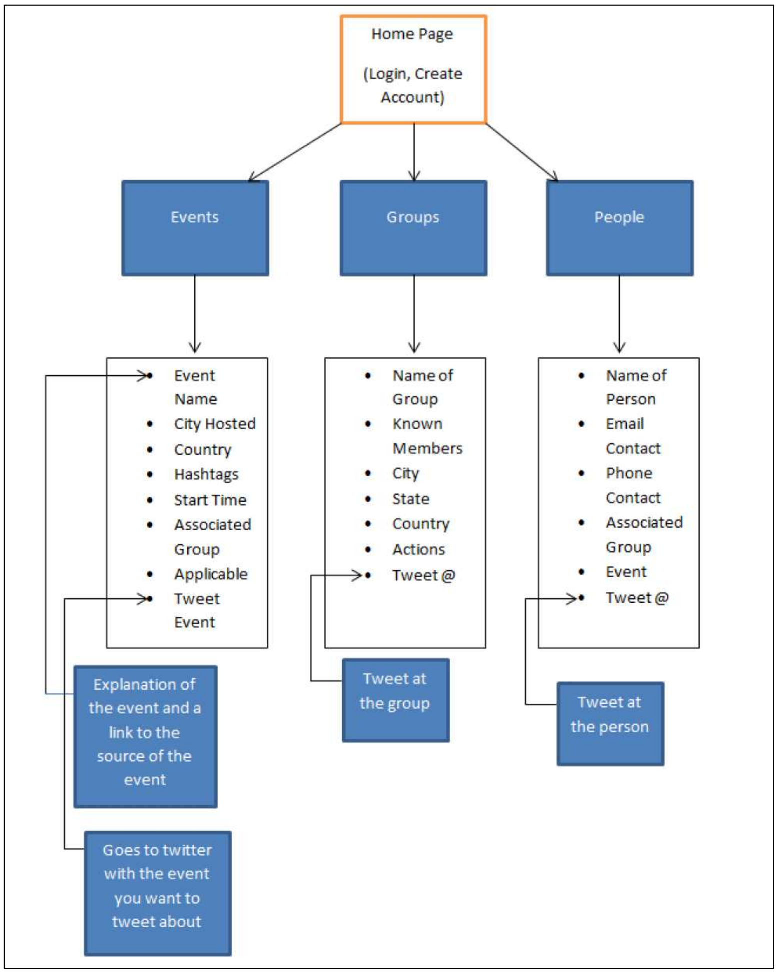


Fig. 7: This sequence diagram shows the interaction between the events, groups, and people pages with the database and URL's.

5.3. Design Rationale

The goal of this viewpoint is to make it as easy and intuitive as possible to view and interact with people and events. It is important that the relevant data be shown in such a way that makes it easy to find what the user needs, as well as easy access to the original information.

6) Resource Viewpoint

6.1. Implement system of improved sorting of finding events by nearby location

Listed as one the more difficult requirements to complete, the task is given a large amount of time to complete in order to get it fully functional. It is important and more relevant for the user to view events that can easily be travelled to by their location. Developing a way for the application to view and sort those locations could cause some concerns.

- Creating different approaches in sorting locations of the events currently listed and detecting the location of the user
- Creating a database of all locations and their immediate distance from each other

6.2. Design View

Our viewpoint for the location nearest the user functionality is shown below. We plan to make use of how most internet browsers have a location setting which can tell where a user is accessing the website from. From this information we will query the database and look for upcoming events that are close to the given location and return those events to the application to be displayed. This is our most difficult challenge and one that will take a lot of testing.

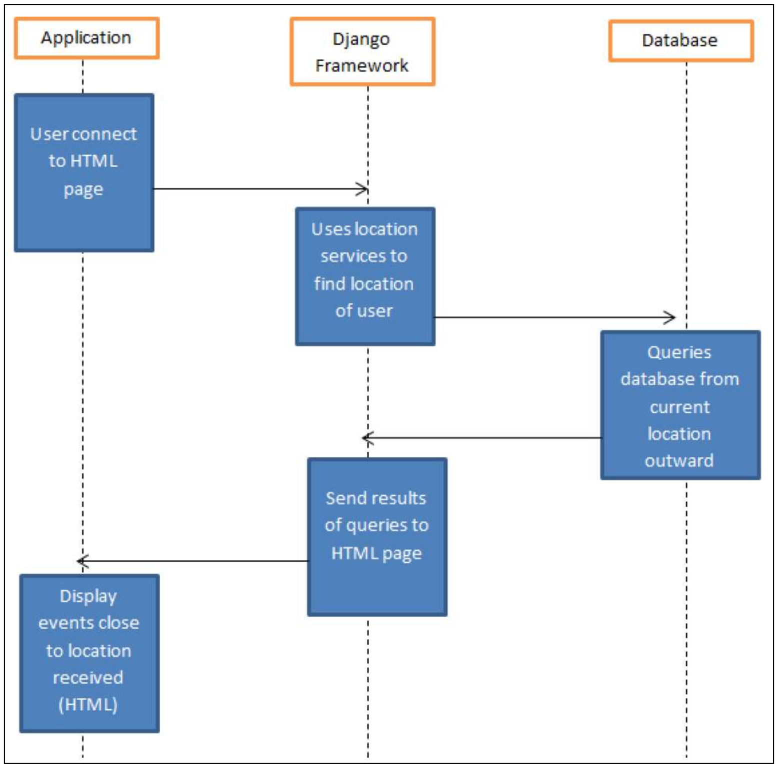

Application
Django Framework
Database
User connect to HTML page
Uses location services to find location of user
Queries database from current location outward
Send results of queries to HTML page
Display events close to location received (HTML)

Fig. 8: This sequence diagram shows the interaction between the Django framework and the database.

6.3. Design Rationale

This viewpoint will require manipulating location data. Once the user's location has been determined, either through location services or through manual entry, we will have to perform a search through the database to find all events within a certain distance of that location. This will be the most difficult part of the entire task. Once the events have been found it will be relatively simple to show them to the user, sorted by distance, so that the user can view those closest to them easily.

7) Composition Viewpoint

7.1. List tweets about events and/or people via the app and without

Currently the prototype lists only official events found on meetups.com. It is important to also view events not listed on that website and that are posted via Twitter. Design concerns include:

- Designing a way to distinguish between a regular tweet, and a tweet with an event
- Linking twitter accounts with the right user and the matching account
- People who tweet about events, but are not a part of the application
- Search via API all about the meetup page and linked towards events
- Create entry in database about a certain person tweeting at a certain time
- Want to be able to see who tweeted, and able to re-tweet that tweet

7.2. Design View

Our viewpoint for the tweeted events requirement is shown below with the communication between the application, Django framework, Twitter API, and the database. We plan to have the application communicate with the Twitter API such that the items that we find with the associated tags we will use will be displayed on the application and stored in the database.

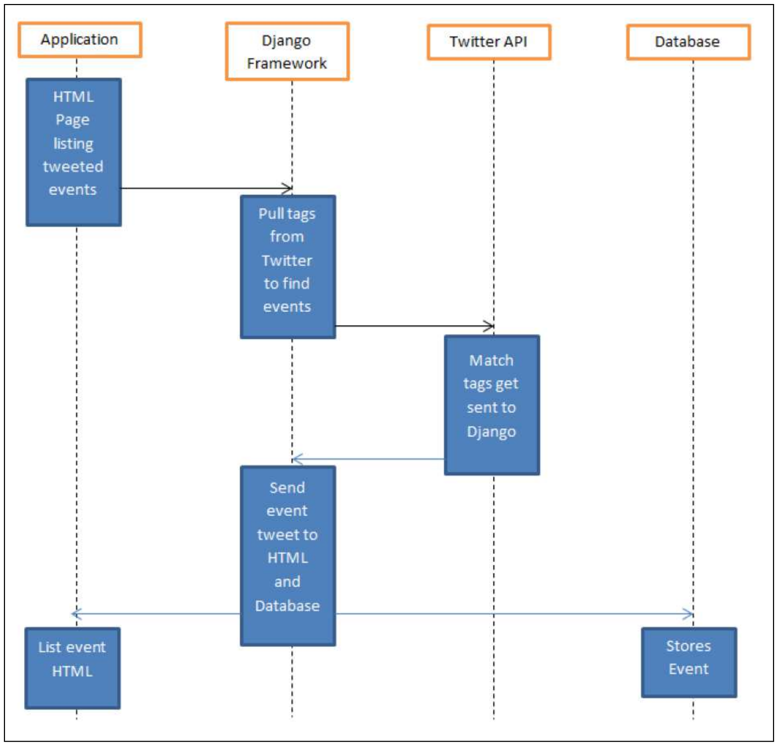

Application
Django Framework
Twitter API
Database
HTML Page listing tweeted events
Pull tags from Twitter to find events
Match tags get sent to Django
Send event tweet to HTML and Database
List event HTML
Stores Event

Fig. 9: This sequence diagram shows the interaction between the Django framework, twitter API, and database.

7.3. Design Rationale

Once again the Twitter API will be useful in accomplishing a task. The API will be used to find tweets that have hashtags that are related to open source events. Once those have been acquired, they will be presented to users on the website as well as stored in the database for future reference.

F. Discussion on Design Document

Throughout the course of the year we have had to change many things for the design of our systems to due to either limitations we encountered or figuring out that our original design plan was not going to work. In this discussion we are going to look at each original design that we had to change and talk about what we had to change in order to be successful in this project.

When looking at the 'Fix the "People" page where profiles are not implemented' we learned that its not that the profiles are not implemented but there was actually a bug in the prototype we were given. The bug was found when looking how the table of information was displayed on the people page in which every single person was trying to be loaded at once to the same page which would eventually cause the browser to crash. Rather than worrying about how the data was stored that we talked about in our original design, we only had to deal with how the information was being displayed. The issue was then solved with a simpler design implementation of showing only a couple of tables at a time.

The 'tweet at a person listed in the database' design was fairly accurate with our actual implementation. The actual design has an HTML button with a Twitter symbol that is linked to the twitter handle that has been tracked down by the API calls the system has made already. The change that was made to this design was adding Hoot Suite which is a collaboration of social networks that are managed under one account. The user will need to login to their Hoot Suite account and then the twitter handle is put into a tweet for that user and ready to be sent via Twitter. This change was a better way to handle the Twitter handle and get the user to an appropriate action page.

For the 'Add user accounts and track when a user has tweeted an event' we successfully implemented our original design except for one aspect which is the tracking when a user tweets an event. Based on our design the Django framework was suppose to keep track of when the user is tweeting but instead Hoot Suite handled all of that. We decided that it would not be worth it to add the complication with the Hoot Suite to the design. The design is still accurate with the implementation of account creation and signing in to the application and the database interaction is the same using the models in the Django framework.

The listing tweets functionality is done by querying the Twitter with API calls searching for the appropriate hashtags associated with the tweets from our application and not from out application. The tweets are straight from Twitter and are parsed and displayed using HTML on the tweets section of the application. This way we are able to see tweets based on the hashtags that are selected and what have been tweeted from the application as well.

Looking through the design document we created at the beginning of this project the above discusses what changes we had to make in the design decisions for certain requirements. The other requirements kept the design we created although implementation struggles would cause us to change the way items were implemented mainly because we were learning techniques throughout this project. Ultimately, we feel that the design decisions made at the beginning of this project were sound.

V. TECHNOLOGY REVIEW

A. *Goal of Project*

The goal of this project is to build connections, collaborations, and identifications between community members and leaders in open source projects. This entails contributing to Apache Software Foundations existing software platform to create tools that identify and support community leaders and members. Better tools will help identify potential community leaders which will then allow community builders to engage with and support those leaders. Over the course of the project, consistent updates check what tools were used as intended and which ones could be improved and expanded. Evaluation of the tools implemented is done in order to measure its significance and effectiveness.

B. *Components and Technologies*

For this project we have broken down the technologies we could use into three different pieces which come together as part of the entire system: Web Application Framework, Backend Data Storage, User Interface, and User and Group Management System. Below you will find research done for the individual technologies available for these pieces along with a selection our team is using for the main system.

C. *Web Application Framework*

1) Introduction: As our project involves a website, we will need a web application framework to build it on. A web application framework is software that is designed to help with the development of websites, web applications, and more. A framework aims to help alleviate the overhead that comes with web development, with libraries that help with tasks such as database access and templating web pages. Possible frameworks are as follows:

- Django
- AngularJS
- Ruby on Rails

2) Django: Django is a free, open source, high-level Python web application framework which follows the model-view-controller (MVC) architectural pattern. Django aims to do most of the basic development for you. It emphasizes reusability and pluggability of components, rapid developments, and the dont repeat yourself principle. It provides an optional dynamic admin interface, which is controlled in a similar way as other portions of the framework, by models. Django also has a database migration system to make migrations easier, as well as a testing framework to make testing your web app easier. It can be used for both front-end and back-end application.

3) AngularJS: AngularJS is a free, open source, JavaScript web application framework designed to help with developing single-page applications. It aims to simplify both the development and the testing of such applications by providing a framework for client-side MVC and model-view-viewmodel (MVVM) architectures, along with components commonly used in rich Internet applications. The AngularJS library works by first reading the HTML page, which has embedded into it additional custom tag attributes. Angular interprets those attributes as directives to bind input or output parts of the page to a model that is represented by standard JavaScript variables. The values of those JavaScript variables can be manually set within the code, or retrieved from static or dynamic JSON resources.

4) Ruby On Rails: Ruby on Rails is a web application framework written in Ruby. Rails is an MVC framework, providing default structures for a database, a web service, and web pages. It encourages and facilitates the use of web standards such as JSON or XML for data transfer, and HTML, CSS and JavaScript for display and user interfacing. In addition to MVC, Rails emphasizes the use of other well-known software engineering patterns and paradigms, including convention over configuration (CoC), don't repeat yourself (DRY), and the active record pattern.

5) Selection: Our project already has a prototype, which uses Django. We will continue to use Django, because it is both already implemented, and a good, solid, easy to use framework that suits our needs. With the models, views, templates, administration, and database management, it will make our task much easier. Our other options require quite a bit more workto do the same things in another framework, and even more work for us re-implementing all the features that are currently there.

D. Back-end - Data Storage Piece

1) Introduction: For the back-end section of our design project, we will need to store large sets of data for things such as profiles, people, groups, community leaders, events, and information of those events. These items require a data storage component that is already implemented into the prototype of the website. These data need to be accessed and stored quickly in order for data to be periodically transferred fast enough to handle consistent updates of events. Depending on the priority of the objective, relatively pertaining to process speeds and data storage types, requires different types of possible technologies that could be used. For our priority, we assume that we will be concerned most about storage capacity in managing events and new community leaders that start a new project. Individual project groups are also projected to contain a large amount of space. Possible technologies for our data storage component are listed below.

- Apache Derby
- MySQL
- PostgreSQL

2) Apache Derby: Apache Derby is a database management system developed by Apache. The engine is a full-functioned relational embedded database-engine that supports JDBC and SQL for programming APIs, with IBM DB2 SQL syntax. Experience with Derby indicates small liabilities in areas other than test environment. One issue has to deal with interrupted I/Os cause Derby to fail outright on Solaris. Building a shim becomes necessary to protect it from those failures. Apache Derby tens to have low performance for complicated queries and low performance on large datasets. Apache Derby as a result becomes more of a testing tool for testing environments and performance testing.

3) MySQL: MySQL is the most popular and commonly used relational database management system. MySQL can handle a lot of data and can work efficiently and cuts corners for runtime efficiency. A majority of websites can work with MySQL. There are tools scalable that are easy of use and easy to manage Depending on the database-engine, MySQL can lack certain features such as the full-text search.

4) PostreSQL: PostgreSQL is the advanced, open-source object relational database management system that has become standard and is the DBMS that is used for our prototype. PostgreSQL is different from other RDBMSs because of its highly object-oriented functionality. It becomes very efficient in handling many tasks very efficiently. PostgreSQL is open-source and free and supported by a large and experienced community. Some possible disadvantages to PostgreSQL includes any simple use database to be overly complicated and would be better used with MySQL.

5) Selection: As mentioned, the prototype currently uses PostgreSQL and will continued for the entirety of the project. Viewing in the advantages and disadvantages of other technologies, PostgreSQL shows the most stability and provides the most efficiency that our project has as an objective. It provides substantial data integrity and may be useful for integration of some potential complex designs with the websites. Any type of custom feature designed would have the least amount of trouble being implemented with PostgreSQL.

E. Front-end - User Interface

1) Introduction: The majority of the website built will be dependent on its UI and how the user interacts and navigates through the site and its various options. This process requires and utilizes many different technologies including PHP, HTML, or JavaScript. The prototype utilizes the UI with the web framework of Django and its designing capabilities. With our front end piece, we will not have a specific technology that we will use entirely. Instead, we are utilizing part of each technologies to handle our data management, website navigation process, and the websites graphical layout. Possible technologies comprised for handling the front end user interface are listed below.

- PHP
- HTML
- JavaScript

2) PHP: PHP is a server-side language designed for general purpose as well as web development. With the ability to access and connect to external databases, it would be utilized for accessing web servers and connecting to our PostgreSQL database and access all of the users and groups in it. PHP can be mixed easily with HTML code and vice versa. Any PHP code is interpreted and executed by the web server that sends its output to the client. PHP is integrated and utilized into a large majority of the websites on the internet.

3) HTML: HTML is the most basic building block of all websites. The technology allows images and objects to be embedded and used to create interactive forms. Our web framework technology uses HTML to build its websites through its own interface and becomes the front end of our piece. HTML user interface is more secure than most sites as there is less of a change that you will get hacked. Another advantage is that you have a lot of control over the UI. Another advantage is that other coding languages can be easily integrated into the websites user interface. Some disadvantages of HTML include the length of time it takes to construct a UI with HTML. Another disadvantage of HTML is that if one character is out of place it can mean the entire UI doesn't load properly and it is a much more tedious process. The final disadvantage is that simple changes to the UI can take much longer to implement than you are willing to spend since you may have to make those changes one page at a time.

4) JavaScript: JavaScript is one of the most simple, versatile and effective languages used to extend functionality in websites. advantages of JavaScript include execution on the client side which means that the code is executed on the user's processor instead of the web server thus saving bandwidth and strain on the web server. JavaScript is also easy to learn and comprises syntax that is close to English. It uses the DOM model that provides plenty of prewritten functionality to the various objects on pages making it easier to develop a script to solve a custom purpose. Some disadvantages would include security issues. JavaScript snippets, once appended onto web pages execute on client servers immediately and therefore can also be used to exploit the user's system. While a certain restriction is set by modern web standards on browsers, malicious code can still be executed complying with the restrictions set. Another disadvantage would be JavaScript rendering varies for the user interface. Different layout engines may render JavaScript differently resulting in inconsistency in terms of the user interface.

5) Selection: Based on the prototype that has been implemented already we have decided to select HTML as our user interface technology as it works great with Django. With the advantage of easily integrating other coding languages into the website means that we can add other language features as well. HTML also has a lot of UI features working with Django and with the security of HTML being very good we think this is the best technology for our webpage user interface. JavaScript and PHP both have their benefits but we feel that HTML is the best fit for our project.

F. User and Group Management System

1) Introduction: The user and group management technology in relation to this project is very important as one of the requirements is to enable user and group management. As we continue building the web page we will be implementing a user and group management system that will allow users to create accounts. This is also key in the admin section as admin accounts will be needed for the admin users of the webpage. The technologies looked into are listed below.

- LDAP
- AD
- Django Authentication System

2) LDAP: LDAP is abbreviation for lightweight directory access protocol which is a software protocol for enabling anyone to locate organization, individuals, and other resources such as files and devices in a network, whether on the public internet or on a corporate intranet. The main benefit of using LDAP is the consolidation of certain types of information within a system. An example that supports this is all of the different lists of users within the system can be merged into one LDAP directory. Thisdirectory can be queried by any LDAP-enabled applications that need this information. The directory can also be sued by users who need directory information. Other LDAP benefits include its ease of implementation, and it well defined Application Programming Interface (API). On the con side of LDAP is that if you want to use this user and group management technology you will need LDAP enabled applications or you will need to use LDAP gateways. There currently arent a plethora of LDAP enabled applications available for Linux. While LDAP does support some access control, it does not support as many security features.

3) AD: For quite some time the standard in the user directory space has been Microsofts Active Directory (AD), which is embedded in organizations large and small. Some of the pros of using AD is active directory is generally considered to be a significant improvement over Windows NT Server 4.0 and AD provides a centralized administration mechanism over the entire network. It also provides for redundancy and fault tolerance when two or more domain controllers are deployed with a domain. Active directory automatically manages the communications between domain controllers to ensure the network remains viable. Users can access all resources on the network for which they are authorized through a single sign-on. All resources in the network are protected by a robust security mechanism that verifies the identity of users and the authorizations of resources on each access. The cons of AD are that it is difficult to integrate into pre-existing network systems. AD offers no means to manage non-Windows clients (such as Macintosh or UNIX) or servers and supports very little management control over pre-Windows 2000 systems. Another con is that AD relies upon DNS to function, but not all DNS servers are capable of supporting AD. Existing DNS systems may need to be upgraded or replaced before they can support AD.

4) Django Authentication System: When looking into the Django authentication system I have found a lot of documentation that helps in implementing the system so that a web template can be created where users and groups can be created. This is also easily managed as it uses a database to store the users information such as MySQL. Another pro to using the Django authentication system is that we are using Django as our framework for the web page that our tools will be utilized on. This will allow us to already have the tools necessary to implement the management of users and groups. Another pro to this system is that it is easy to set up the admin accounts so that accounts can be handled if needed. Another pro for using this system is that it lets you plug in other authentication sources as you can override Djangos default database-based scheme, or you can use the default system in tandem with other systems which makes it more portable. Looking through the documentation I was able to find some disadvantages of using Django authentication system as there are some documentation gaps that need to be addressed. Another disadvantage is that it can get difficult to implement as there are small but time consuming hurdles as we are on a tight time frame for the next couple of terms.

5) Selection: We have decided to select the Django authentication system as our technology for user and group management as the benefits from doing so give our team greater advantages then choosing the AD or LDAP methods. Since we are already using the Django web framework it will be much easier to implement the authentication system as the tools necessary are already at our disposal. AD and LDAP are good systems however the complexity of their implementations and how they seem to have limitations on interacting with other systems means it is less portable. Django as our framework is the main reason for selection but the system also offers portability, good documentation, and simplicity.

G. Discussion on Tech Review

Looking back at the technologies we have chosen for this project it is interesting to point out that we kept all of our selections for the areas of user interface, data storage, authentication, and web application framework. The Django web framework was heavily implemented into the prototype that we received at the beginning of the project which fit perfect for what this project is trying to accomplish thus we continued with the Django framework. The PostgreSQL database continued to full fill our needs of what we wanted the database to do thus we continued to use it. The authentication from Django was also perfect for the job.

What is great is that we ended up adding more technologies to the application as we continued developing. One important technology that was added is JavaScript. Throughout working with the Google Maps API and other HTML sections of web pages, JavaScript turned out to be very useful in implementing features such as the tweet at a person social share button. JavaScript also played a major role in the implementation of the Google Map used to display the locations of the events that have been imported from the Meetup website. We also utilized the API technologies for Google Maps, Meetup, and Twitter to accomplish the requirements at hand.

Other than the addition of JavaScript and utilizing API calls we stayed true the our technology decisions throughout the project to accomplish our goals.

VI. BLOG POSTS

Fall 2015 - Week 3 - Blog #0

Wednesday, October 14, 2015

PROGRESS: The rough draft of problem statement has been completed; we are waiting for Ross's input before submitting it to Kevin. UPDATE: We have successfully finished the problem statement document and were able to get it signed by our client before 12 PM on Friday 10/16. The final document has been uploaded to SharePoint and turned in to room 2098.

ISSUES: We've had issues involving getting a hold of our client as he has a very busy schedule. We expect to hold a meeting Thursday, October 15th at around 11:30 to discuss finalizing our problem statement and answering questions. UPDATE: We also had a problem getting the document signed as we needed to change the wording of our signing agreement but this was easily taken care of.

FUTURE PLANS: We plan to begin developing our requirements document. This includes setting up a meeting with our client Ross, discussing requirement details and implementation details.

Fall 2015 - Week 4 - Blog #1

Wednesday, October 21, 2015

PROGRESS: We gave our client Ross, expectations from him as well from the group. We've also started on our requirements document and got more familiar with the prototype and brainstormed ideas for our user stories. We created our meeting times for our TA.

ISSUES: We still have not received contact from our client this week, but we expect to hear back soon. There are still some confusions on what requirements should be on the project in the scope of what we think and what our client thinks.

FUTURE PLANS: We plan to meet up next Monday at 5:00 PM to finish the requirements document to send to our client for approval. Each individual group member is preparing their own user stories before our Monday meeting.

Fall 2015 - Week 5 - Blog #2

Wednesday, October 28, 2015

PROGRESS: We completed the rough draft of our Requirements Document. We received feedback from our client about correct priorities in the requirements tasks and we plan to email him our rough draft to look at the rest of it. We have met with our TA and discussed about communication issues. UPDATE: We were able to finish the requirements document and get it signed by Ross. There was thought of giving the assignment another week as Ross came up with some more requirements but we decided to just stick with the requirements we have so far. We did this because Ross felt we could just keep adding to it thus we will just stick with this baseline.

ISSUES: Communication exchanges have been occurring once a week lately. We expect responses more timely once the project and documents get going.

FUTURE PLANS: We plan to have another Skype conversation with our client discussing meeting times and clarification on Requirement Document and future documents due. UPDATE: We are now working on our technology review to find out if there is any other technology we will use other than what Ross has already provided.

Fall 2015 - Week 6 - Blog #3

Wednesday, November 4, 2015

PROGRESS: Currently, we have completed our Requirements document and will submit the draft to Kevin and Nels to review and see for suggestions on edits. We have discussed about how to tackle the technology review. Currently, for the pieces of our project we have come up with three pieces.
1.FRONT END: This is our website and what the user sees and the interface that they can interact with.
2.BACK END: This is our database center of where we keep track of all community groups and leader profiles.
3.NETWORKING: This includes the technologies required to interact with social media (currently includes meetups.com only).

ISSUES: We need to come up with one more piece to fill up all four pieces for the specified requirement that Kevin gave us. We are having some trouble figuring out which technologies are used, or currently are being used for some back-end and networking pieces on the prototype.

FUTURE PLANS: We plan to email Ross about specified technologies involving the back-end and networking side of the prototype. We also want to ask for another piece that the project uses so that we can specify it in our Technology Review. We plan to divide up the technology review to split the responsibility for each person to write one page per piece and collaborate on the final one. We plan to meet back on Monday to finalize our draft and submit it to Kevin and Nels to give them a day to take a look at it before submitting our final draft.

Fall 2015 - Week 7 - Blog #4

Tuesday, November 10, 2015

PROGRESS: We have completed the technology review. We've updated our Requirements Document is now to date. Input from both Kevin and Ross was helpful to organize our project and put into perspective of the technologies used. Currently we've added finishing touches to the technology review and discussed on methods to approach and design the Poster Board.

ISSUES: Since Wednesday is Veteran's Day, all of campus is closed so we are not able to meet up with Nels this week.

FUTURE PLANS: We set up meeting times to have a video conference with Ross next Monday at 4:30 PM. We will discuss and talk about any concerns coming up so far and questions that we have regarding upcoming deadlines approaching.

Fall 2015 - Week 8 - Blog #5

Wednesday, November 18, 2015

PROGRESS: We were able to have a conference with our client Ross. We caught him up with everything up to date and made sure that we told him new and current deadlines from class as well as personal dates to submit our drafts to Ross for review. We finished our draft of our Poster Board and submitted it on time. After meeting today, we got a good start on our Design Document and began discussing our elevator pitch that we are scheduled to present on Thursday of dead week.

ISSUES: We still aren't too clear on exactly what we are to discuss for our "Body" portion of the Design Document. Finding time to collaborate together on the document for a significant amount of time before the deadline will be difficult.

FUTURE PLANS: We plan to meet Monday of next to work and hopefully get a good part of the Design Document completed so that we may submit it to Ross for review. We plan to try to run the provided code in the prototype to learn more about its design that would help us to better understand the infrastructure of the project.

Fall 2015 - Week 9 - Blog #6

Wednesday, November 25, 2015

PROGRESS: Our group accomplished a lot during week 9. We were able to create our elevator pitch by giving each other parts to remember during the presentation. We have three sections which are a problem, a solution, and why it works. We also made a lot of progress for our design document as we are getting a rough draft ready to send to our client Ross on Monday November 30th which we hope to have a rough draft that is very close to being a final version. We also successfully practiced our elevator pitch and are ready to present when called on.

ISSUES: The issues we ran into during this week were when we worked on our design document as most of the designs a little difficult to construct as we needed to first figure out how the current implementation works. We will need to figure out exactly how the current prototype is working in order to make sure our designs will work correctly.

FUTURE PLANS: Our future plans will be to present our elevator pitch on Tuesday December 1st as we should be called on this day. We also will be sending off a rough draft of our design document to our client Ross on Monday November 30th. Next week we plan to have out design document completely finished by the end of the week and the progress report finished by Sunday December 6th.

Fall 2015 - Week 10 - Blog #7

Wednesday, December 2, 2015

PROGRESS: After two consistent days of meeting and planning our work, we have just about completed the design document and are currently finishing up the progress report as well. Monday consisted of working on the majority of the design document in terms of formatting and preparing to show Ross our rough draft. We had a Skype conversation with Ross scheduled at 4:30 on Monday and it went very well. Ross provided great input in terms of more clarifications of our listed requirements and issues as well as providing some ideas for the future on how to tackle the implementation of most of the requirements. Tuesday consisted of our presentation of our elevator pitch and a quick meeting of what to preparation of what to have done for our meeting on Wednesday. With most of our progress completed by Monday, we were able to send both Ross and Kevin early rough drafts of our design documents for them to take a look at before submitting it for grading on Friday.

ISSUES: Currently, we have not received any response by email from either Ross or Kevin, but Ross had already given some input from initially looking at our rough draft over Skype and Nels had given some of his thoughts at our TA meeting on earlier on Wednesday as well.

FUTURE PLANS: Progress looks great and we plan to complete both of our progress report and design document by the end of Wednesday for an early submission. If any of our members spot an issue before the deadline, we can easily fix that and resubmit it with time to spare. With our meeting with Ross, we were able to schedule regular weekly meetings with him every Wednesday at 10AM for progress reports and inquiries about the prototype.

Winter 2016 - Week 1 - Blog #8

Thursday, January 7, 2016

PROGRESS: After having our Winter Break, we met up and discussed plans for weekly meetings and deadlines after our 8AM lecture. We were able to set up the local host on two of our systems while still working on one more. We have a mock up the user interface for most of the buttons and links of the website and are beginning to develop our requirements. We set up a Git repository for our local system that develop with and eventually push onto the existing prototype.

ISSUES: We still need our third system working correctly with the local host. It is difficult to install on Windows, so we will use VMware to host it. We were unable to meet up with our client Ross for our weekly scheduled meeting. We have not been in contact our client since before the break.

FUTURE PLANS: We are waiting for a response from Ross by email and to hopefully meet with him next week. We plan to begin developing on the local host for the requirements. We decided our official meeting days for our group are Mondays, Tuesdays, and Thursdays.

Winter 2016 - Week 2 - Blog #9

Thursday, January 14, 2016

PROGRESS: As a group we were able to successfully load Docker onto all of our machines. Two of our group members had to load on to a virtual machine to run Linux to use Docker and to locally host the server. The week consisted of getting familiar with the project as well as beginning development of the website and its features. Plans to complete all of basic functions before Alpha stage is running smoothly. Currently a few more buttons are needed to complete all basic pages for the website.

ISSUES: Contact with Ross has been difficult. Exchanging emails have been working, but a Skype call is still needed to more clarification. A slight misunderstanding in scheduling has halted our regular meeting schedule but plans to have that solved by our next meeting date look good.

FUTURE PLANS: We are looking to complete all basic functions soon and to begin working on all of our requirements for Beta stage. We plan to have our meeting on Wednesday with Ross to clear most of our questions up.

Winter 2016 - Week 3 - Blog #10

Thursday, January 21, 2016

PROGRESS: This week was an interesting week for out group as we ran into some interesting problems. The progress that we did make was getting code ready for the nearby location requirement and it seems to be working well. Another good point from this week is that we were able to have our weekly meeting with our client Ross and we decided on a new branching system for our GitHub repository that helps a lot in development for our client and us. We also updated our requirements document with better language to make sure it is easier to read when looking at our requirements. We only have 3 to 4 more requirements that need to meet the alpha level as we have successfully completed the profiles for users that are hyperlinked with their names.

ISSUES: The issues we ran into this week were sickness and injuries. One of our team members got injured and could not attend our normal meeting time and another had to stay home sick thus we were not able to meet at least twice this week as a group which slowed down production. Another issue we ran into was the geolocation tool running into bugs and now showing up on the web page but we are looking into it.

FUTURE PLANS: Our future plans are to continue developing and finish the requirements to reach an alpha level presentation which we hope to have accomplished by the end of next week. Then we will begin the beta level implementation of features. Next week we hope to only have 2 requirements that need to reach alpha level by the time we meet our TA on Wednesday. We also plan on meeting three times next week to get thing back on track due to the problems of this week.

Winter 2016 - Week 4 - Blog #11

Thursday, January 28, 2016

PROGRESS: This week was productive in terms of achieving a solid status on the Alpha phase of our project. We were successfully able to implement most if not all of the initial buttons on our site that directly links to either the intended requirement, or to an "Under Construction Page" for further development. That page was created to make the site more functional in the sense that it is still continually being worked on and that that part of the page is not yet complete. After taking a look at the overall requirement for the Alpha stage, we can be confident enough to have this presented by week 6 and we are beginning to work on the progress report document and the video that goes along with it. Our meeting with Ross was great in terms of clarification and what he expects of us at this point.

ISSUES: Currently our main issue is the implementation of the geo-location service of determining the location of the events and creating a specific mile radius to display to the user events nearby their location. We have code from a previous prototype by Apache that Ross provided to us, but trying to understand another coder's is really difficult so we as a group are figuring out different ways to approach this problem. This implementation is geared more towards the Beta stage, but the faster this problem is completed, the better situation we would be in for the future.

FUTURE PLANS: We plan to complete the required documents for the Alpha stage and to create our video to present as well. In parallel to that, we want to continue to develop our project and try to complete that location service as soon as possible, but at the same time completing the rest of our project.

Winter 2016 - Week 5 - Blog #12

Thursday, February 4, 2016

PROGRESS: For this week we have started to work on our midterm progress report document and are finishing up that portion of the assignment. We have organized the document into different sections beginning with the requirements, to issues while developing the application, to significant portions of code for the document. We also had our meeting with both Ross and Nels on Wednesday to both update our current progress on the assignment and clarify certain questions that we had. Communication with Ross has been solid and we are getting good feedback on certain issues that we have been facing on parts of features that we are trying to implement.

ISSUES: Unfortunately, we cannot have host our application on our ONID webpages because the Oregon State pages are not compatible with Django. We also unfortunately heard from Ross that he cannot provide us with a host for our application through Apache as well. This issue is not particularly too important because our source files are still on GitHub. It is just a little inconvenient for Ross to view our changes via pulling our local files.

FUTURE PLANS: As a group we plan to have finishing touches on the progress report soon and to begin recording for our finished presentation video portion. We plan to have that completed before Wednesday so we can both show to it Nels and Ross before the submission deadline.

Winter 2016 - Week 6 - Blog #13

Thursday, February 11, 2016

PROGRESS: Week 6 consisted of the completing documentation for the Midterm progress report and to begin and finish the video presentation that follows along with the report. As a group we spent about 2-3 work days' worth of work on the report and about 2 days of work on the video in total. Completing these two assignments within the deadline proved to be fine with estimated scheduling exactly the amount of time needed in order to show a rough draft to Nels for feedback. We were able to get good feedback from Nels for our format before the deadline and make appropriate changes before turning it into SharePoint and my own public html. We began recording the video presentation on Monday earlier this week and did not finish until the end of Tuesday. Editing was done overnight and we were able to complete the video project before Wednesday before our meeting with Ross.

ISSUES: There were not that much issues while working on these two assignments. The only issue was working on the video and getting all of the film done in the allotted time. We underestimated the amount of time it would take to get all of the recording done while renting out the audio recording studio in the library, and were lucky enough to find a room for the very next day to finish up recording.

FUTURE PLANS: After submitting our report and video presentation, we will begin to start development once again on the project and hope to complete Beta stage before the end of the term. Right now we are just on schedule with development, but the long stretch of requirements to fulfill before Beta deadline will be difficult to pull off.

Winter 2016 - Week 7 - Blog #14

Thursday, February 18, 2016

PROGRESS: With development in full progress, we've dedicated different requirements for the three of us to focus on getting as much implementation as possible. Justin's focus has been to work with the meetups API and accessing twitter handles that are within registered users for meetup.com. Megan has been handling the task of looking at the code that generates the People's page and improving the way that it generates its table and people and accessing why it is so slow on the actual prototype. Hai's task was focusing on the login and account creation page to access the backend of the user profile and to be able to successfully register users onto the database. As a group, we managed to complete the login and registering of user accounts of the webpage. Additionally, we were able to locate the jQuery data tables that contained the algorithm to generate the table of people and access where exactly the bug is happening. Finally, we were able to pull in the twitter handles of the user accounts from meetup but not yet into the backend database.

ISSUES: For the user and account creation, we have yet to implement the ability to extend the user accounts from the five basic attributes. Earlier methods to accomplishments are outdated compared to the current 1.9 version of Django. For the People's page generation, we have moved where the data is generated to after the table and everything is created. Unfortunately, there is no way to test this fix on the local prototype because the issue only occurs on the Apache hosted side. Finally, getting the twitter handle to show on the website is the current issue that is not yet to be determined for its reason.

FUTURE PLANS: Plans to complete these parts of the requirements are important, but they should not take our full attention compared to the overall project as a whole. Getting the basis of these requirements functional are just the main focus so we can begin to build the rest of the requirements because some of them depend on another requirement to be completed, such as implementation of user accounts to save location basis. These task are next on the priority list.

Winter 2016 - Week 8 - Blog #15

Saturday, March 12, 2016

PROGRESS: This week the team was able to complete different components of the requirements that everyone was assigned. Justin was able to successfully complete the tweet at a person requirement by figuring out how to use the Meetup API in order to pull the Twitter handle of the people that were being imported into the website. With that he was able to link the button on the people page to setting up Hootsuite which sends the tweet to the user. Megan was successful in having the people page fixed. By finding a bug in the prototype she was able to move some code around that led to the problem being solved, at least locally, for the people page. Hai successfully got the create accounts working along with login which makes it so that only if you are logged into the application you can use the actions buttons for the tools. This means if a registered user is not logged in when they go to perform the import meetups action the button will not be displayed but as soon as a registered user logs in the buttons will be active.

ISSUES: The issues we came across this week were mainly the standard issues a development team usually comes across. Justin was having an issue storing the twitter handle response he was getting from the Meetup API but eventually figured out where the problem lied. Hai was having trouble getting a model to work for the Django application but eventually was able to find out where the user authentication was happening in order to have the ability to create accounts and login/logout.

FUTURE PLANS: For the next week we plan on working with the search event by location requirement which will be handled by Justin. We also plan to get the exporting of information for people that have been imported completed as well which will be done by Hai. We also would like to take a look at the list tweets via/not via the application which will be handled by Megan.

Winter 2016 - Week 9 - Blog #17

Saturday, March 12, 2016

PROGRESS: This week we had a good amount of progress made. Hai successfully implemented the exporting of people information via an XLSX file which can be viewed on all operating systems as most can read a CSV file so that should work well. The exporting shows the file containing information such as twitter handle, name, bio, location, and URL associated with that person. This is implemented fully in the button that was placed for Alpha. Megan has been working hard to get the Twitter API to work with her in listing some tweets on our application. There has been some struggle but she has been able to display a list of tweets associated with the hashtags we are searching with but now we want to be able show tweets that are from people who tweeted about the events. Justin was able to get a map to show up on the application as he has been working with the Google Maps API and was able to get a search box associated with it for typing in different locations.

ISSUES: The main issues from this week were just more coding bumps as you would expect. Justin was running into issue with getting the correct locations of the events from the Meetup API as it was only pulling the groups origin location which is not good for displaying that on the events location page. Megan was struggling with the Twitter API as it does not have very good documentation and was not easy to understand right away.

FUTURE PLANS: For the next week we would like to focus our attention on completing the search event by location requirement and the list tweets about events on our application requirement. Hai will also be starting to look into improving the hashtag search as well to get even better results.

Winter 2016 - Week 10 - Blog #18

Saturday, March 12, 2016

PROGRESS: This week we were able to make a log of progress on our requirements. Justin has successfully implemented a map feature with search functionality that displays the events that have been imported as markers on the map. This will allow the user to search for a location to see what events are near that location. The application also uses geolocation to find out the position of the user from the browser they are using and displays their location on the map. Megan has also made a lot of progress on the list tweets requirements as she is successfully listing the tweets about the tweeted by the users that we have imported and related those with the hashtags we have set up. Hai also has improved the hashtag searching which will mark that requirement as complete as well. Justin was also able to add the Twitter handle to the people's profiles that are generated to have a way to make contact from developers to those people. This way that requirement is completed as well. This means for all of our requirements we have successfully completed Beta level functionality in all of our requirements.

ISSUES: The issues that we ran into this week were just struggle with the API calls but throughout the week we really buckled down and got a lot of our requirements done.

FUTURE PLANS: For the next week we plan to finish up our final report and presentation for winter term which will include everything we have done with our project so far and proves that we have beta level functionality. Our client Ross has been very happy with our progress and has been loving what we have accomplished thus far. He would like us to create patches for the project itself which we plan to do as well as we begin to make tweaks and finishes to our requirements to reach Version 1.0 level functionality.

Spring 2016 - Week 1 - Blog #19

Wednesday, March 30, 2016

PROGRESS: Heading into week 1 of spring term we had a great start in getting our version 1.0 release finished. Throughout the week we found time for all of us to meet for 3 hours on Monday's, Tuesday's, and Wednesday's along with setting up a time to meet with our TA every week. After we completed setting up schedules we immediately began working on code. We have already polished up some login features so now it lets you know when you have successfully logged out. We also completed the tweet at a person listed in the database requirement by getting rid of the tweet at person button when said person has no twitter handle to tweet to. Then we began working on the other requirements but already have two down.

ISSUES: The issues we ran into this week was having some group member feel under the weather and having to power through to get stuff done. We have not really met big issues for this week.

FUTURE PLANS: We are continuing polishing requirements as Megan is finishing up the list tweets via/not via the app, Hai is working on completing the login and logout so that the first and last names of the users are saved correctly, and Justin is working on displaying people that are hosts of events for more accurate community leaders. We plan to have these items completed by next week along with having our first TA and Client meetings of the term on Wednesday, April 6th.

Spring 2016 - Week 2 - Blog #20

Wednesday, April 6, 2016

PROGRESS: During our second week of spring term we have made some more progress. We have been able to make progress in polishing the tweets that are listed on our website that Megan has been working with. Tweets are now displaying with some organization. She has implemented a dictionary of tweets that have been imported and she is continuing those two requirements of listing tweets via and not via our application. Hai and Justin were able to get a list of event hosts to display on the application along with the events they are hosting. This list is located under the people page tab and the person profile generation requirement is nearly polished.

ISSUES: The issues we ran into this week were a lot of coding bumps. Hai and Justin were running into difficulties adding fields to the Django models which stopped them from getting a list of event hosts from being able to display on the application. After a couple of days of tampering with the code they were able to get the correct information stored with a correct API call. Megan has run into some issues with displaying the dictionary of tweets she has imported but is slowly making progress with it displaying at least text.

FUTURE PLANS: In the next week we would like to complete the profile generation for the event hosts for the people profile generation requirement to be completed. We would also like to continue with the listing tweets requirements as they are near completion. Also, we will be finishing up the event search by location requirement.

BOOTH PLANS: For our expo presentation booth the items that we would like to have is two large screens (monitors) and 2 power strips just in case. We plan on presenting our website using the laptops we own and the screens provided as displays. We also plan to use our own mouse so that will not be needed either. Since we are just displaying and walking through our website we don't need much for the booth except for being close to a power connection and 2 large monitors to display everything on.

Spring 2016 - Week 3 - Blog #21

Tuesday, April 19, 2016

PROGRESS: After our third week of developing for the version 1.0 release we actually accomplished a lot. Justin was about complete the event hosts profile generation that is accompanied by an import hosts button that will import the hosts of the events that have been imported. Hai has been getting further along with the event by map search implementation and plans to have that done by the end of next week. Megan has been finishing up the tweets requirements and plans to have those done next week as well.

ISSUES: The issues we ran into this week was scheduling confusion with our client. We originally had meetings starting the Wednesday of week 1 for this term and going till June 15th just in case. Our client accidently mixed up the dates and thought we were starting our meetings on June 15th which is not good. We will have to figure out another time for us to have our weekly meetings. We also ran into a coding bump of importing topics for event hosts.

FUTURE PLANS: For the next week we plan to finish the events by distance search and map implementation, the tweets application, and the exporting list functionality for the event hosts. We also plan to finish up our poster in week for and have that ready for Kevin by the 26th of April.

Spring 2016 - Week 4 - Blog #22

Thursday, April 21, 2016

PROGRESS: This week was a very good week for our project. Justin was able to complete the profile generation requirement of event hosts including specified topics for those hosts. Megan and Hai were able to get sections of the HTML working for the listing of tweets and now they display much better than they were which completes the listing of the tweets requirement. We also were able to go through our poster one last time and make necessary tweaks so that it is ready for our expo presentation.

ISSUES: The biggest issue that we ran into this week was finding out that the best time for us to set up a new meeting time with our client was Monday's at 1:30 PM. This is not a bad time; the problem is that Ross will not be able to meet with us for the next 4 weeks which includes past expo. This puts us in an awkward situation of not discussing things with our client. However, we did come to an agreement to have an email sync at the same time we are supposed to have our meetings thus we should always be on the same page.

FUTURE PLANS: For the next week we are down to our last three requirements which are tracking if user tweets same person twice, improving the visuals, and adding a distance measuring by location search to the map for the events. We each have taken a task so we strongly believe that we will be able to get everything finished in a week or week and a half.

Spring 2016 - Week 5 - Blog #23

Saturday, April 30, 2016

PROGRESS: During the last week we have made a lot of progress with the visuals of the website. Hai was able to change the look of the events page, people page, and the profiles page of the event hosts and imported people. This has been a major improvement from the last version of the visuals. We have also made some progress with the tracking of a user tweeting by being able to tell when someone has clicked a button twice. This will help us from preventing a user from tweeting at a person or about an event twice on accident during that page interaction. The map feature seems to be our most difficult one at hand. We have added a radius to the map but we need to get it to work correctly.

ISSUES: The major issue we ran into this week was not being able to meet with our client at all. Not via Skype and he has not responded to our emails. We should still be able to make the deadline okay but it is a bump that we wish we could avoid. The other problem we ran into was the map feature not working as we would like for our distance implementation.

FUTURE PLANS: For the next week we will be submitting our poster before the March 2nd deadline. This will most likely be on Saturday April 30th. Then we will be finishing up the last couple of items for the version 1.0 release while finishing up our midterm report and presentation which is due on March 6th.

Spring 2016 - Week 6 - Blog #24

Wednesday, May 4, 2016

PROGRESS: In finishing up our 6th week of this spring term we have made it to the point where we have a version 1.0 ready project. The three main requirements we needed to finish were the visuals, map, and tracking tweet button actions. As we continued we found that tracking the tweet button was not going to accomplish the ultimate goal of tracking a user tweeting so we decided to not go with this idea with approval from our client. The next item was the map and events searching. Instead of having he table display the events closes to you we decided to put a circle around a certain radius to give the user a better idea of how close events were to them. In this we have completed the map requirement. The last requirement for improving the visuals we have decided to skip the user studies as this was not very important to our client. We did complete the task as we improved many pages to look cleaner and have the information spread out in a nicer manner. With this we have completed our version 1.0 by completing all requirements.

ISSUES: The main issue that we ran into this week was communication with our client. Our client is very busy with travel for the next couple of weeks and has had very slow responses to our questions. Thus some items we had to take into our own hands/interpretations in order to complete requirements. We have been emailing without client to get things cleared up but it still poses problems.

FUTURE PLANS: For the rest of this week we will be submitted our midterm progress report for our version 1.0 release with our slides, video presentation, and our paper as a latex document. We plan to submit all of this on Friday morning (May 6th). After that we will be creating small patches for the requirements we have accomplished to give to our client for final submission. After that we are just looking forward to Expo.

Spring 2016 - Week 7 - Blog #25

Thursday, May 12, 2016

PROGRESS: During this week we were able to find out where our table will be positioned during the engineering expo. We also were able to get a patch of all the features we have added to our project and will be sending that over to our client Ross. We are officially done with development and are in the final preparations for expo.

ISSUES: The only issue we came across in the last week was submitting the patch file to where our client would like it. The Apache website we are submitting it to is having some issues when we try to submit our patch. We plan to work through this with Ross even though he is very busy for the next couple of weeks.

FUTURE PLANS: This next week we plan on meeting to discuss what we will be saying at expo so that we are prepared for whatever questions we receive. We will also be finishing the submission of our patch once the upload issue is resolved. Heading to expo is the next big step.

Spring 2016 - Week 8 - Blog #26

Wednesday, May 18, 2016

PROGRESS: This week we have successfully submitted the final patch file to the Apache website our client asked us to submit it to. With this done we are waiting for the patch to be approved and applied to the live project. Our client informed us to keep hounding him about it so that the patch can be applied before the June 9th submission of our final report just so we have our official final product to show. We also picked up our poster and took it to Kevin's office. Now we are just waiting for 2 days to present at Expo.

ISSUES: The issues we ran into this week were just trying to figure out what questions we should practice for expo. Ultimately we continued to come up with questions to make sure we were best prepared to present at Expo in 2 days.

FUTURE PLANS: To continue this week we will be presenting at Expo on Friday, may 20th and then in the following we plan to start on our final report including taking all of our documents and converting them into a Latex document.

Spring 2016 - Week 9 - Blog #27

Tuesday, May 24, 2016

PROGRESS: This week we have successfully presented our project at the 2015 - 2016 Engineering Expo and are officially done with development and presentation preparation. We have also submitted our patch file with all of our new features to Apache thus we have completed our project. We are currently working on the final paper along with the final presentation.

ISSUES: We have not run into any issues during this week as we are just finishing up our final report and presentation.

FUTURE PLANS: This will officially be our last blog post due to the plan of having the paper and video presentation done by next week. We plan on turning in everything on Monday of finals week. We all have enjoyed this project and want to thank the university for getting us involved in such a great event.

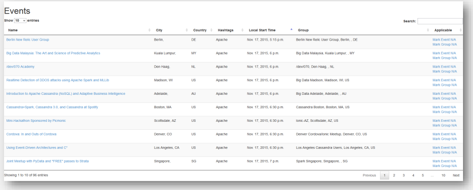
Introduction

The open source community has had a major impact on the computer science world as it is one of the biggest opportunities to learn about, create, and contribute to software creation. The biggest reason why we decided to undertake this project was to have the opportunity to create tools that will help grow the open source community as it is a great way to give back to a community that helps the field that we are obtaining our undergraduate degrees in.

These tools will be able to present eager developers with an opportunity to find exciting projects and see who are apart/creating these projects.

Background

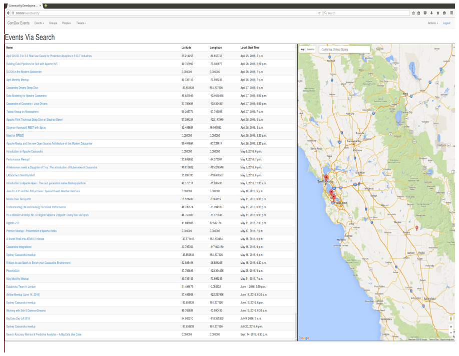
- A prototype had already been implemented prior to beginning the project.
- Many of the technologies were already decided and implemented in the prototype.
- Our goal of the project was to fix and implement new features to aid collaboration of new/current open source projects



Front page of project as of November 16, 2015

Community Driven Development

Tools Supporting Open Source Community Growth



Project Description

The website features many different tools that are used to find events relating to open source projects. These tools also give developers a chance to see leaders/members of events and gives them key information in order to get involved.

- Pulls data from Meetup.com, Twitter, and Google Maps
- Identifies community leaders to people who love to improve a specific project
- Connects various data publically available online to one source
- Provides users enough information to contact and engage with project leaders and assist them with their own unique skills
- Specific Technologies
  - ○ Django
  - ○ HTML
  - ○ JavaScript
  - ○ PostgreSQL

Features

- List events within a two week period related to Open Source Meetups and community groups
- Gathers information on members hosting events and members within a group
- Visual interface for viewing events by location via Google Maps
- Lists tweets with associated hashtags for Open Source
- Compose tweets directed at certain people or about events

REQ#	Requirement	Priority
1	Fix the "People" page where the list of community leaders are shown	High
2	Tweet at a person listed in the database	Low
3	Add user accounts to the application and have actions be hidden for accounts	Medium
4	List tweets about events and/or people via the app	Medium
5	List tweets about events and/or people not via the app	Medium
6	Export a list of people with information	Low
7	Improve hashtag searching of application for better results on relevant events	High
8	Improve the visuals of the tool looks as a whole	High
9	Implement a system of finding events nearby location entered or within a radius of the user	High
10	Add feature to generate a profile for community developers to have contacting information easy to view	High

Feature implementation priority list

Community Driven Dev

TEAM MEMBERS
Justin Bruntmyer
bruntmju@oregonstate.edu
Hai Nguyen
nguyehai@oregonstate.edu
Megan Goossens
goossenm@oregonstate.edu

PROJECT ADVISOR
Nels Oscar

PROJECT SPONSOR
Ross Gardler

Improvements and Results

- Added features to improve user interaction in finding and locating events.
- Improved interface where users can register and update important events.
- Gathers information from not only meetup.com, but now Twitter as well.

VII. PROJECT DOCUMENTATION

A. *How The Project Works*

Our project is a website built in Django. It uses Docker to maintain a consistent environment. The code is structured as a typical Django project. The most important parts are the views and the templates, and the models. Each view is either a page on the site, such as the events page, or an action, such as importing meetups. The models define the database. Each model has associated fields with it, e.g. the group model has name, city, state, and country fields.

1) Structure of Project & Theory of Operation: The structure and theory of operation go hand to hand with this project because the structure was created based on the theory of how this application operates. Below we will be talking about the structure of the project and how that structure is suppose to operate within the application. All the of the main structures of the application will be put together in a flow diagram to illustrate how the structure operates.

The structure for the project is simple. The Django framework creates a database using the PostgreSQL database model. From here we can create tables within the database which are also known as models. These models are used as sections of the database where we can store information in a way that makes sense. The Django framework also sets up an admin interface so that we have complete control over what happens to the framework. Once the appropriate models are created then there are the pages of where the events, people imported, event hosts, event search by area, view tweets, and logging in will be displayed. Once the pages are set with the HTML we then pull in the information needed to populate those pages.

In order to pull the appropriate information the project uses a hashtag search algorithm that takes a list of hashtags inserted into the database and queries meetup.com with an API call for meetup's that match the hashtags that we have stored. The API call returns the meetup information in the form of Json. The application then parses the Json information returned by the API call searching specifically for the information that we requested such as longitude, latitude of meetup, start time, who hosts the event, country, state, address, event name, group hosting, picture, and source page address for the meetup. All this information is stored in the database under the appropriate model and then pushed the HTML page where the HTML can access and display this information to the user.

Now that events have been pulled in we can pull the information for the people associated with a group and hosts of events. To import the people within a certain group the application takes the group ID and make an API call to meetup.com to give the information of who is in that group. The information is returned to the application in a Json format, the data is parsed and is stored in the database under the appropriate model with information such as twitter handle, address, longitude, latitude, name, picture, topics, meetup ID, and last activity. The information is then stored in the database and then can be accessed by the HTML pages to be displayed to the user. To import the event hosts the application take the events hosts ID's that have imported along with the events and then created an API call to meetup.com that asks for information for every event host ID that has been imported. The information is returned in Json format and parsed then stored in the database. The associated pages then have access to the database using HTML to be displayed to the user.

Once all of the information has been imported a user can export information about people or events using the export button at the top of the associated page for events and people. The exported lists includes biography information, twitter handles, start times, etc. that is information pertinent for the events or for the people. The login structure is also run by the Django framework with the Django authentication system which keeps track of the usernames and passwords created. The authentication checks the database with the credentials submitted at login and allows the user to login or be denied. Once a user has logged in they have full access to importing functions or marking groups as not applicable.

All of the technology that operates for this project is happening in a docker container. A docker container's wrap up a piece of software in a complete filesystem that contains everything it needs to run: code, runtime, system tools, system libraries anything you can install on a server. This guarantees that it will always run the same, regardless of the environment it is running in. The docker container wraps up our Django application so that it can be run on a local machine in order to be developed on. This is also how the application can be hosted on a web server. Our client and the team that worked on this before us found this to be the best way to run the application for development purposes. The docker container is running as a service on the local machine to open the local host to hosting the application.

B. *How to Install & Run Application*

The way our team developed on the project was using the docker container for Linux as we had countless troubles getting the container to run on the Windows platform. For the purposes of giving the best information on how to install and start this application on a local machine we are going to go over the installation process of what we know to work. Thus, below is a step by step installation process for getting the docker container to run correctly on a Linux operating system and starting the application.

1) Install and Run Docker for Linux
   Docker requires a 64-bit installation regardless of your Ubuntu version. Additionally, your kernel must be 3.10 at minimum. The latest 3.10 minor version or a newer maintained version are also acceptable. Kernels older than 3.10 lack some of the features required to run Docker containers. These older versions are known to have bugs which cause data loss and frequently panic under certain conditions.

   To check your current kernel version, open a terminal and use uname -r to display your kernel version:
   - uname -r

   Log into your machine as a user with sudo or root privileges and open a terminal. Install docker with the command shown below:
   - sudo apt-get install docker-engine

   Start the docker daemon with the below command:
   - sudo service docker start

   Verify docker is installed correctly with the command below, this command downloads a test image and runs it in a container. When the container runs, it prints an informational message. Then, it exits.:
   - sudo docker run hello-world

   You can find more specifics and deal with issues you may encounter here:
   https://docs.docker.com/engine/installation/linux/ubuntulinux/

2) Install Docker Machine for Linux
   To install docker machine for Linux, run the command below:
   - curl -L https://github.com/docker/machine/releases/download/v0.7.0/docker-machine -‘uname -s‘-‘uname -m‘ > /usr/local/bin/docker-machine && chmod +x /usr/local/bin/docker-machine

   To check if the installation was successful run this command:
   - docker-machine version

3) Installing Django (Python should already be in Linux)

   To install Django enter these commands into the terminal:
   - sudo apt-get update
   - sudo apt-get install python-django

4) Download Git Repository from GitHub

   Open a terminal in a location that you want to store these files. Then enter the commands below to download and switch to appropriate branch:
   - git clone https://github.com/MaraJade/seniorproject.git
   - git checkout alpha

5) Change Directory and Start Script

   First you will need to change to the right directory for starting the script. Enter the command below:
   - cd seniorproject/tools/

   From here you run the below command to start the application:
   - ./scripts/deploy.sh

Now you answer the next few questions with these responses in order:

- local
- Yes
- username: (anything you would like)
- password: (anything you would like)

Now you can go to a web browser and type in the address 0.0.0.0 for your local host and connect to the application. The account you can use right away is the one you just made above. Or you can create a new account.

VIII. How did you learn the new technology?

Throughout the project the main technologies that were new to us that we needed to learn were Twitter & Meetup & Google Maps API calls and how they worked, JavaScript, Django, and Docker Containers. With all of these technologies there was a learning curve for each of us throughout the project however the most difficult time would have to be during winter term of the project. Winter term was the time for us to began development for an Alpha and Beta release. It took some time but we will discuss how we learned about each of the technologies and learned how to use them effectively.

To learn the API technology for meetup.com, Twitter, and Google Maps there was a lot of documentation deciphering throughout the project. We quickly learned that some API's had better documentation than others. The meetup.com API documentation was the best of the three because all of the API calls were laid out with a description and examples of what you could do with them. There was also a console testing feature on their website that let you test out an API call to see what the result would be before you actually use your token to do so. The Google Maps API documentation was decent but still difficult to understand in some places as all the technical information of what is happening behind the scenes are not mentioned in the documentation. With patience and trial & error we were able to fully understand what the API was doing and how we could integrate that with our displaying of the events on the map plus the search functionality. The last API is also the most difficult to understand which is the Twitter API. The documentation did not seem to flow in a way that we could understand clearly. Megan began looking at the documentation first and asked for help on the explanation of what was happening and all of us were struggling. After taking each section and breaking down what was happening we began to understand the Twitter API more but we feel that this kind of documentation should be easier to understand.

The next technology that we learned was JavaScript. The main way we all went about learning how to code in JavaScript was by practice. The way we would begin to implement items for our requirements were to break down the requirement into sub-tasks and then write JavaScript to accomplish those tasks and put them together. The main source of documentation we would use was Stack Overflow examples that people have used JavaScript for along with tutorials at W3Schools which is a website with a lot of useful tutorials.

Django was new to the entire team thus we did a lot of research on the technology in the first couple of months while we were creating our documents in the fall. With Django and Python going hand and hand we had an advantage in understanding how to make changes in Django. The biggest item we needed to learn was how the framework is working and how the database interacts with the application itself. In order to do this we continued to read documentation on Django in order to progress in understanding it which let to us feeling fairly confident on using Django in the end.

The last technology that we had to learn was the docker container and how to run the container. This was a struggle at first as we wanted to use the Windows platform to run the application however there were many library complications that we could not resolve. Instead we began looking into the docker container and how it runs in a Linux environment. The documentation that we found led us to believe that running the container would be a simple process on Linux thus we tried just that. We followed the installation guide from the Docker website and had the container running our application fairly quickly. We learned about docker machine and running the docker service for Linux through the Docker documentation which was easy to understand.

We continued to learn as a team and were willing to help one another whenever we got stuck on a topic. The learning process took a huge team effort and contributed to us finishing this project.

IX. What did you learn from all of us? (Justin Bruntmyer)

A. What technical information did you learn?

Throughout this project I had the opportunity to learn interesting technical information. The technical information I learned a lot about was interacting with API's of several different websites. This not only taught me how to interact with the API calls and parce the information but it also taught me to read technical documentation for those API's. This was sometimes frustrating as some websites wrote documentation better than others. Another technical piece of information I learned would be how to functionally control Django and create an web application from it. I also learned how to manipulate database fields by parsing information and storing that information within models.

JavaScript was also a large portion of technical information I learned from this project. The Google Maps section of this project was a section I focused heavily on as it interested me. As I continued to learn not only about the Google Maps API but I also learned JavaScript coding as the map was basically JavaScript and forced me to understand technically how JavaScript works.

I not only had the chance to learn technical information in computer science topics but I also was able to learn a lot about the technical side of documentation. I learned that writing professional documents as good as you can the first time will bring better results when it comes to development. Though it is true that documents may need to change such as the Requirements Document or the Design Document it is important to create the document the first time to the best of your ability as this saves time and frustration.

B. What non-technical information did you learn?

When reflecting on the project the non-technical information that I learned was mainly how to interact with a client. I learned right away that communication was going to be the key to having a successful project. The reason I learned this early on is because I noticed the biggest problems we were having as a group was when the client and our team were not on the same page for a certain subject. This led to confusion epically when it came to trying to pick a time for having a weekly meeting. This also taught me that when you are working with people there are going to be different schedules that you are going to have to work with in order to make everyone happy on meeting times.

Another non-technical piece of information I learned from this project was valuing collaboration with your team. With group projects in the past, mainly in a class setting of a couple of weeks, I found that the collaboration wasn't all there between the members because most people just wanted to get it done with and move on which will get you the grade but doesn't necessarily give you the experience you could have obtained. With working in this team I found that as soon as we spent a lot of time with one another we began to gel in our coding methods and work with each other to get requirements done correctly, not just done. What this collaboration experience ultimately taught me is that working towards the benefit of the group as a whole will lift the group and bring the best out of one another. This was the best non-technical information I learned because it will help me realize the value of a team as I continue my computer science career.

C. What have you learned about project work?

What I have learned about project work throughout this experience is that client interaction and constant communication is very important to make sure that everyone is on the same page. This made weekly meetings a priority that had to be met to make sure that everyone was on the same page with the requirements, progress being made, and questions that needed to be answered. I also learned that honesty and communication within the development team is crucial as well. If someone is struggling on a certain requirement or on a part of a requirement than it is important to communicate those thoughts with the team as it can lead to pushing past the issue when everyone gives a hand. This will reduce the risk of requirements being stuck for many weeks at a time. This means that honesty about being stuck and not knowing a solution is important even if you feel that it is non-trivial and you should figure it out.

I also learned that it is important to

D. *What have you learned about project management?*

E. *What have you learned about working in teams?*

F. *If you could have done it all over, what would you do differently?*

X. WHAT DID YOU LEARN FROM ALL OF US? (MEGAN GOOSSENS)

A. *What technical information did you learn?*

I learned a lot about working with APIs during this project, which surprised me, as I didn't expect that to be one of the things we were going to work with. I learned that APIs can be easy to use, and they can be extremely difficult to use. Documentation, as with many things, can determine how easy the API is to work with. The Meetup.com API was the first one I worked with, and it was well documented. Information on how to use it was usually very clear and easy to find, and in some cases, they would even build your queries for you. I then worked with the Twitter API, and had almost the opposite experience. While the Twitter API might seem well documented at first glance, it is difficult to use. Their examples were almost useless, and I had a very difficult time finding what I needed to use. Fortunately, the information was in there somewhere, it just took me a long time to find it, and then another long time to figure out how to use it.

B. *What non-technical information did you learn?*

The non-technical information that I learned had to do with all the documenting and organizing that we had to do. Even though I learned about documentation in another class, this was the first time I did extensive documentation in a real project. I also learned a lot about estimating times for finishing requirements. It's one thing to know that the estimated times are going to be wrong, it's another thing to see exactly how wrong you were.

C. *What have you learned about project work?*

Firstly, I learned that it is extremely frustrating to have a client that is difficult to contact. It helps to set up a regular meeting with such a client, though it is a good idea to make sure that everyone is clear on the days and times of the meetings. Even if you have regular meetings, though, miscommunications still happen, and things may need to be clarified multiple times before everyone is on the same page.

D. *What have you learned about project management?*

I learned that it is possible to have a group that functions very well even if there are no assigned roles. Though there was nobody nominally in charge, we all held each other accountable.

I learned quite a bit from this project. On the technical side, I learned a lot about interfacing with APIs, and how documentation can either make them easy to use, or an absolute pain to use. On the non-technical side, I learned about writing even better reports, as well as how to explain our project in ways that make it easy for people outside the open source community to understand. In project management I learned things like how frustrating it can be when your client is hard to contact. If I had to do it all over again, I would work harder to understand exactly what this project is about. We spent a lot of time trying to understand what our client wanted from each requirement, time that would have been better spent fulfilling them.

XI. WHAT DID YOU LEARN FROM ALL OF US? (HAI NGUYEN)

A. *What technical information did you learn?*

While working on the project, there were many technical concepts that challenged me that I was mostly unfamiliar with. Since the beginning, I was only semi-familiar with the concepts of databases, while only working with Python and MySQL a limited amount of times. For the rest of the technologies from the project, I learned everything else from scratch. Those technologies include learning Django, Docker, and API calls.

With Django, the learning curve for me was gradual. Originally I attempted to hardcode everything in Python without using the useful tools that Django provides and adding objects and fields were very difficult. Our client was very helpful in providing useful documentation that aided us in making our changes and commits and eventually, Django became second-hand whenever we pushed changes.

Docker was mostly hard to handle in the beginning of the project when Justin and I had difficulty setting up the project on our Windows operating system. After several days of difficulty without successfully setting up the Docker, we eventually switched from our Windows system, to developing on a Virtual Machine with Linux. This made development overall a lot easier with the tools that Linux provided and all three of the group members on the same operating system.

Lastly, during the development process, the most prominent issue throughout the three terms were API calls and getting used to all different APIs that we handled from meetups.com, Twitter, and Google Maps. Since we were handling different APIs at the same time, keeping track of all and getting used to all three were particularly difficult when trying to pull the data into our database.

B. What non-technical information did you learn?

There are many non-technical aspects that fall under an engineering project. In terms of the developing process for the project, it was very clear of all of the documentation that we had to write, that we were accountable for all requirements that we proposed to do. The project as a whole was scaled to the perception that we would be working on it throughout the school year for a couple of hours a week, so we knew in terms of organization, that we needed to project the amount of work that was reachable for us.

Specifically for some requirements, we have a development and debug process where we would develop for a certain feature, but would have to review its functionality and importance in terms of how much priority that we had to spend in order to complete the feature and how much time and effort should be spent on a specific task compared to the entire project as a whole. In terms of a non-technical aspect, managing our priorities and the process of handling new feature requests were the most important aspects that we learned.

C. What have you learned about project work?

Client interaction was prominent where we would get feature requests after some completed requirements that would satisfy our sponsor. There would be a cycle in our development process where once weve completed a requirements, then our client would offer feedback and suggestions on the next step or how to take the feature even further. Most of the time during our development schedule, we did not account for any stretch goals, but we were able to complete some along the way.

I learned that preplanning for a project almost rarely goes according to plan or the schedule. This is especially relevant when we did not understand completely some requirements that we were given or suggested. It made things very hard to interpret when some of the requests were over email and we were not able to talk with our client on a weekly basis; therefore, there was a lot of extrapolation and interpretation for some of the requests which resulted in some features that were not in the same intention but ended up satisfying the same needs.

D. What have you learned about project management?

In terms of project management, while I did not take the main leadership role in this group project, I believe that I did contribute in a way that required leadership and management that was successful for our performance as a whole. Originally, my role in the group became the scribe and pinpointed issues in some particular efforts in the code. Eventually, as the term progressed, we diverged as a group into similar roles for everyone because each of the group members carried over the same amount of responsibility from each other.

Specifically, there are some aspects of project management that I learned that will carry over to my career and other projects. The most important aspect and knowing the responsibilities of each group member and their task to complete the issue. On plenty of occasions throughout the terms, our group as a whole had deadlines to complete with specific tasks designated for each group member. Although it was not anyone's job specifically to keep others in check, it still became important that as a group, we were consistently progressing to reach our goal to complete each requirement.

E. What have you learned about working in teams?

In the real world, especially in the Computer Science field, it will be very rare to start and complete projects that require individual work. On many if not all occasions, projects will be completed in groups ranging from 5 to thousands of different programmers at once. This is due to the aspect of so many features that go into a single program and all checks and debugging that needs to be done. This project is no exception. Even at our development stage, it is an open source project, which means that we as a group are not the only individuals working on the project, but rather, we are contributing to the Apache Software Foundation as a whole.

F. If you could have done it all over, what would you do differently?

If I had the option of doing everything over again, the most important thing that would change would be attention to detail and clarity. A lot of trial and error was a result from poor time management with all of the meetings and I believe tasks could be completed faster if everyone, including our client were on the same page. I

In regards to our client, his requirements that were listed were sometimes not clearly understood from what was written compared to the actually semantics of what wanted to be completed. As a result, the direction that our group initially headed towards development were misguided under the interpretation of some features that were requested. This was a direct result of communication and details not being specified on the group's end.

For our group, tasks were sometimes not detailed clearly enough and at times there were points where we were working to complete the same task at the same time where valuable time could be spent on working on different features.

XII. Problems Encountered

Throughout this term we have encountered many problems that have impeded our progress for this project. This was to be expected as with all projects. Events occur that can halt progress and bring forth challenges that, as a group, we needed to overcome. The problems that we have faced are listed below along with the methods we used to get through the situation. We also ran into the occasioin coding block where we would have days where not much progress was obtained. However, we feel this did not impeded our progress as it is just part of being a computer scientist.

A. Problem 1

Deciding which platform we were going to be developing on using Docker. This was a major issue as we originally planed to work with the Windows operating system however we could not get the application to run locally on a Windows platform. We continually ran into errors with creating a local database along with having the right Docker tools to support the application. We had available resources such as a README file that gave insight on the problem but whatever we tried did not seem to work. We eventually shifted gears and decided to try Docker and the application on Linux, specifically Ubuntu 14.04. Thanks to the Linux knowledge of Megan Goossens and plenty of online documentation we were able to get Docker installed successfully along with running the application on a local host. From this point we decided to continue developing in the Linux environment.

B. Problem 2

At the end of the Fall 2015 term we set up a weekly meeting with our client throughout Winter 2016 to discuss implementation details for the week and planned to utilize this time to make sure everyone is on the same page. Due to some miss-communication we were unable to meet with our client for the first two weeks. This halted our progress with because we had a lot of issues with getting Docker to work with Windows and we were counting on a meeting with our client to resolves those issues as soon as possible. We eventually got in contact with our client and figured out what was happening as the first week our client was on an unexpected trip to the UK and in the second week our client did not realize that these meetings occurred every week throughout the term. These things happen and once we all had a chance to get on the same page every weekly meeting is going smoothly. This problem also happened at the beginning of the spring term. A new weekly meeting time was set up with our client however our client believed that our first meeting was in the middle of June. This was a problem as we tried to set up a new meeting time but our client was very busy with travel and vacation that we were not able to meet with him three weeks prior to expo. Email communication was very difficult as well.

C. Problem 3

During week three of Winter 2016 we ran into the issue of meetings being cancelled due to illness and injury. One of our team members experience an injury that caused a full group meeting with a TA to occur which halted our progress in having to get everyone caught up on the same page. This same week another group member became sick and could not make it to two meetings for the week which meant that two people could not make it so two meetings were cancelled out of the weekly three meetings we have as a group. However, we were able to work individually at home but it was still a noticeable disruption from the normal work we produce in a week. It did not seem like it was going to effect the group at first however when we began to get back on track it took some adjustments to makes sure everyone was on the same page and try to make up for the week we missed.

D. Problem 4

During the first week of the term we began developing based on the requirements we had listed in our requirements document. The problem we ran into here was that we had issues understanding what are requirements were trying to say thus we went through the document and changed the language used for the requirements. This did not change the requirement but it made it easier to understand if someone is reading through it. This took a days worth of progress which was frustrating due to the fact that we believed to have this done last term. We currently are happy with the updated requirements document as it has been approved by our client, professor, and TA. This halted our progress by being an unnecessary step in the implementation process as it should have been completed last term.

XIII. Appendix

A. Essential Code Listings

```
    def _twitterAuth():
        # Encode the keys
        key = base64.b64encode(TWITTER_API_KEY)

        # Set needed values
        authURL = "https://api.twitter.com/oauth2/token"
        content_type = "application/x-www-form-urlencoded;charset=UTF-8"
        body = "grant_type=client_credentials"

        # Create the header
        authHeaders = {'Content-Type': content_type, 'Authorization': "Basic " +
key}
        # Get auth
        auth = requests.post(authURL, headers=authHeaders, data=body)
        # Get the response in a useable format
        authJSON = auth.json()

        return authJSON['access_token']

    # Gets html from Twitter to embed the tweets nicely
    def _oembedTweets(tweets):
        # Get the hashtags
        hashtags = Hashtag.objects.all().exclude(name = "Meetup")

        oembed = dict()
        for hashtag in hashtags:
            oembed[hashtag.name] = []
            for i in range(0, len(tweets[hashtag.name]) - 1):
                url = "https://api.twitter.com/1/statuses/oembed.json?id=" + str(
tweets[hashtag.name][i])
                embededResponse = requests.get(url)
                # Get the json so that the data can actually be accessed
                embeded = embededResponse.json()
                # Only need the html for the template
                oembed[hashtag.name].append(embeded['html'])

        return oembed

    def tweetsNotApp(request):
        # Auth with twitter
        accessToken = _twitterAuth()

        # Get the hashtags
        hashtags = Hashtag.objects.all().exclude(name = "Meetup")

        # Gets all the tweets containing the hashtags
        # Keeps them seperated by hashtag for viewing
        allTweets = dict()
        for hashtag in hashtags:
            allTweets[hashtag.name] = []
            url = "https://api.twitter.com/1.1/search/tweets.json?q=%23" + hashtag
.name + "&src=typd"
            headers = {'Authorization': "Bearer " + accessToken}
            response = requests.get(url, headers=headers)
            # Gets the json version so that the data can actually be accessed
            tweetsJSON = response.json()
```

```
           for tweet in tweetsJSON['statuses']:
               # Only need the id to get the oembed data
               allTweets[hashtag.name].append(tweet['id'])

       oembed = _oembedTweets(allTweets)

       return render(request, 'tweets/notApp.html', {'tweets': oembed})
```

Code Snippet: 1: Views.py showing the Twitter authorization and search for tweets from the application.

```
   def importHosts(hostID):
       log = Log()
       log.description = "Hosts imported"
       log.action_type = Log.EVENT_IMPORT
       log.save()

       # New API call to get a specific persons information based on the Host ID
       url = "https://api.meetup.com/2/member/"+ str(hostID) +"?offset=0&format=
json&photo-host=public&page=500&sig_id=148657742&key=" + MEETUP_API_KEY
       response = urllib2.urlopen(url)
       result = response.read()

       data = json.loads(result)
       hostsInfo = data

       #for hostsInfo in data:
       try:
           host = Host.objects.get(meetupID = hostsInfo['id'])
       except Host.DoesNotExist:
           host = Host()
       try:
           host.country = hostsInfo['country']
           host.fullName = hostsInfo['name']
           host.city = hostsInfo['city']
           host.state = hostsInfo['state']
           visited = float(str(hostsInfo['visited'])[0:-3])
           host.lastVisit = datetime.utcfromtimestamp(visited)
           if 'lon' in hostsInfo.keys():
               host.longitude = hostsInfo['lon']
           if 'lat' in hostsInfo.keys():
               host.latitude = hostsInfo['lat']
           host.url = hostsInfo['link']
           #Gathering the other service information for the hosts, looking for
Twitter
           if 'other_services' in hostsInfo.keys():
               if 'twitter' in hostsInfo['other_services'].keys():
                   if 'identifier' in hostsInfo['other_services']['twitter'].keys
():
                       host.service = hostsInfo['other_services']['twitter']['
identifier']
           host.save()

           if 'topics' in hostsInfo.keys():
               for topic in hostsInfo['topics']:
                   try:
                       record = HostTopic.objects.get(meetupID = topic['id'])
                   except HostTopic.DoesNotExist:
                       record = HostTopic()
                   record.urlkey = topic['urlkey']
                   record.name = topic['name']
                   record.meetupID = topic['id']
                   record.save()
                   host.topics.add(record)


           person.save()
       except:
           print('Unable to save Host object: '), sys.exc_info()[0], sys.exc_info
()[1]
```

Code Snippet: 2: Views.py showing the importing of hosts of the events

```
<script>
  // Initiating the map here.
  function initMap() {
    var map = new google.maps.Map(document.getElementById('map'), {
      center: {lat: 44.5645659, lng: -123.2620435},
      zoom: 13,
      mapTypeId: google.maps.MapTypeId.ROADMAP
    });
    var infoWindow = new google.maps.InfoWindow({map: map});

    if (navigator.geolocation) {
       navigator.geolocation.getCurrentPosition(function(position) {
          var pos = {
             lat: position.coords.latitude,
             lng: position.coords.longitude
          };

          infoWindow.setPosition(pos);
          infoWindow.setContent('You are here!');
          map.setCenter(pos);
          }, function() {
            handleLocationError(true, infoWindow, map.getCenter());
          });
     } else {
       //if stuff doesnt work
       handleLocationError(false, infoWindow, map.getCenter());
     }
    // Create the search box and link it to the UI element.
    var input = document.getElementById('pac-input');
    var searchBox = new google.maps.places.SearchBox(input);
    map.controls[google.maps.ControlPosition.TOP_LEFT].push(input);
    // Bias the SearchBox results towards current map's viewport.
    map.addListener('bounds_changed', function() {
      searchBox.setBounds(map.getBounds());
    });
    var markers = [];
    var cirlces = [];
```

```
     // Listen for the event fired when the user selects a prediction and
retrieve
     // more details for that place.
     // Loop through the upcoming_events_lists and gather each events lat and lon
 to create
     // the markers associated with it.
     {% for even in upcoming_events_list %}
           var myLatLng = {lat: {{even.latitude}}, lng: {{even.longitude}}};
           var marker = new google.maps.Marker({
               position: myLatLng,
               map: map,
               title: '{{ even.name }}'
           });
     {% endfor %}
     searchBox.addListener('places_changed', function() {
     var places = searchBox.getPlaces();
     if (places.length == 0) {
       return;
     }

     //Clear out the old markers.
     markers.forEach(function(marker) {
       circles.forEach(function(circle) {
         circle.setMap(null);
       });
       marker.setMap(null);
     });
     markers = [];
     circles = [];
     // For each place, get the icon, name and location.
     var bounds = new google.maps.LatLngBounds();
     places.forEach(function(place) {
     var icon = {
       url: place.icon,
       size: new google.maps.Size(71, 71),
       origin: new google.maps.Point(0, 0),
       anchor: new google.maps.Point(17, 34),
       scaledSize: new google.maps.Size(25, 25)
     };
     // Create a marker for each place.
     markers.push(new google.maps.Marker({
       map: map,
       icon: icon,
       title: place.name,
       position: place.geometry.location
     }));
     // Create a circle for each place searched.
     circles.push(new google.maps.Circle({
       map: map,
       radius: 16093, //10 mi radius
       strokeWeight: 1,
       fillColor: '#AA0000',
       fillOpacity: 0.20,
       center: place.geometry.location
     }));
     if (place.geometry.viewport) {
       // Only geocodes have viewport.
       bounds.union(place.geometry.viewport);
       } else {
         bounds.extend(place.geometry.location);
       }
     });
     map.fitBounds(bounds);
   });
  }
  function handleLocationError(browserHasGeolocation, infoWindow, pos) {
     infoWindow.setPosition(pos);
     infoWindow.setContent(browserHasGeolocation ?
                           'Error: The Geolocation service is unavailable.' :
                           'Error: Your browser doesn\'t support geolocation.');
  }
 </script>
 <script src="https://maps.googleapis.com/maps/api/js?key=
AIzaSyA46B8dAoEQyc1tdHK89306TdQD_Ox6iw0&libraries=places&sensor=true&callback=
initMap"
      async defer></script>
```

Code Snippet: 3: eventSearch.html showing JavaScript that creates the map for the events search