R3 PROJECT REPORT
Crowd-Sourced Real-Time Map

CSC 555: Social Computing

NAME: Vijay Sankar Ganesh
UNITY ID: vgsankar
STUDENT ID: 200066345

1 Overview

This project creates a Mapping Application that is focused on Social Aspect. It does so by crowdsourcing many parts of the application.

The crowdsourcing aspect of the project comes from how the application gets the real-time updates. Users of the application will be able to tag various locations with various events. Users will also be able to view events that are shared by other users. By updating the map with various events in a timely fashion, this application can promise to provide near real-time information to its users.

Crowdsourcing is a way of obtaining information by enlisting the services of a number of people - either paid or unpaid. This application uses crowdsourcing to obtain two key pieces of information. First, the mapping information. That is, information about various events taking place in different locations are obtained. Secondly, using crowdsourcing to differentiate accurate sources of information from inaccurate ones. The second part is obtained through a system of votes that is common in many social networking platforms such as Reddit, YouTube, Imgur, etc.

This application tries to make the best use of the information it obtains to provide information to users in a very convenient manner. This is possible because the application is a Web-first application. That is, web frameworks are the foundation of this application, and it also uses various web technologies to provide interface to users of different screen sizes.

2 Description

This application is built on top of Google Maps using the Google Maps API for AngularJS, that is, ng-Maps. Twitter Bootstrap based Web interface is provided so that the application can dynamically adapt to different screen sizes. Therefore, this application is also mobile friendly.

By utilizing the Web Services provided by the API, the application is able to provide additional services than the ones that are already provided that are already provided by Google Maps.

While many applications exist today that crowdsource the mapping aspect, most of them are focused on specific areas. There is no general purpose mapping tool that can be built by the crowd for the crowd. This application aims to remedy this deficit.

By introducing the concept of social circles, the application also ensures that user's privacy is respected. For example, a user may wish to show his/her map only to a specific set of people. He can do so by sharing the map with one of his circles. The concept of circles is as follows. Depending on the relationship between two people, each person would categorize the other with a certain degree of closeness and would like to share his events only with similar people.

Another important crowdsourcing aspect of the application is its use of votes to distinguish reliable source of information from unreliable ones.

2.1 Behavior

As already mentioned, the application provides near real-time information about various events on the map.

A sample list of events is given below. The actual events, however, are not restricted to these types.

- *Traffic Information* — in case of high traffic on a particular highway, users will be able to log this information through the application, allowing other users to choose alternative routes.
- *Social Events* — tagging various geolocations on the map with social events, users will be able to invite others to attend the event. Users will also have option of sharing the event publicly or with a subset.
- *Alerts* — reporting security related events such as robberies, users will be able to avoid unsafe places and/or traffic associated with such events.

The application provides a social networking aspect, that is, focusing more on the user's friends list. This would allow users to share events with a particular set of users in their friends list.

2.2 Authenticity and Verification

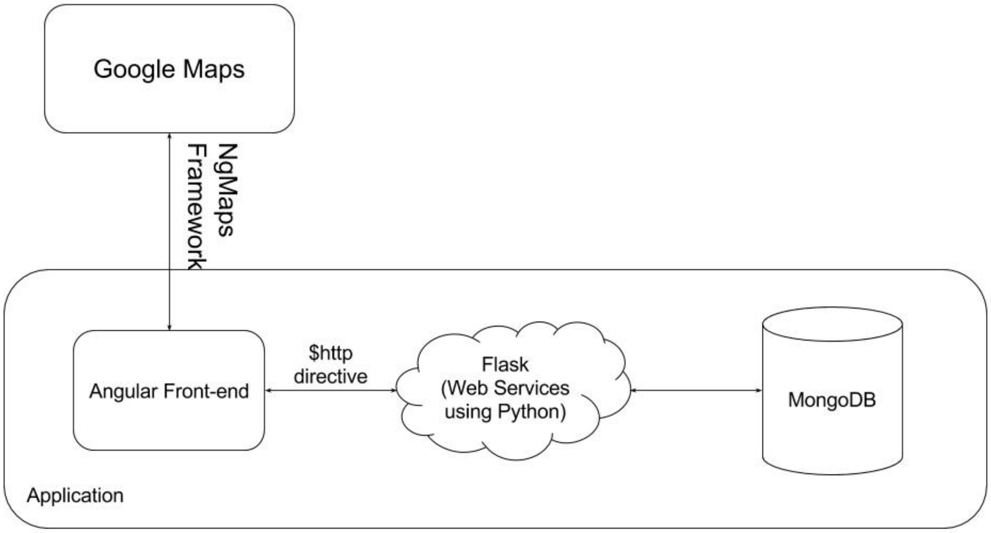
Since this is a crowd-sourced application, a mechanism must be added to protect the application from spam or incorrect information. The implementation of this *mechanism* will be as follows:

1. Each user's profile will start with a default rating – 100, the maximum attainable rating.
2. As each user posts *accurate* information, their profile rating will increase or remain the same, if it has already reached the maximum. This increase in rating is done through a *voting mechanism* in which other users will be able to *up-vote* accurate information.
3. Similarly, in case of *inaccurate* information, their profile ratings will be reduced through *down-votes*.

4. Each user's profile score (which is a simple aggregate of the votes) is linked to the probability that the user's post will be displayed on the map. That is, a lower profile rating increase the chances of the user's event being not displayed to the public. That is, for each event that the user posts, the application would randomly drop the event based on a probability score computed from the user's rating.
5. The rating will also have a maximum value - that is, the value beyond which the profile's rating will no longer increase. This ensures that users cannot build up a high *immunity* by temporarily posting valid events.

3 Implementation Details

A brief overview of the application is given through the following system diagram:



Like most applications, at its core, the application can be divided into two parts - the frontend and backend.

The frontend contains the actual map, and controls that allow the users to interact with the map - post events, rate events, etc. This part is implemented using AngularJS and ng-Maps for AngularJS. ng-Maps is Google Maps directive for AngularJS that provides many of Google Maps' services in an easy to use way for AngularJS Framework.

The second part, the backend part, handles all the data about the users, the events and provides these to the application in a real time fashion. The backend provides Web Services that the frontend application can make calls to and process the information returned.

The application will be designed as a Web-based one using JavaScript (AngularJS) and Python programming languages. Python will be used to provide a RESTful API through the Flask Web Framework. This will be used by the JavaScript clients.

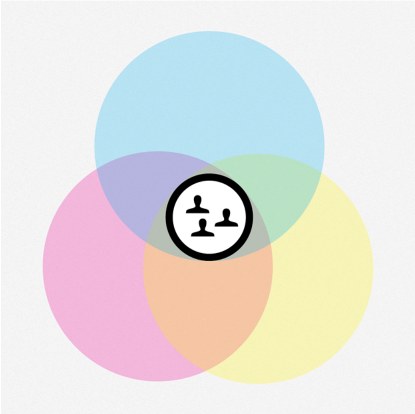
The application is developed using a REST Architectural style to make it more modular, scalable and modifiable. This allows the interface and the main application logic to be modified/upgraded independent of each other.

There are many reasons to decouple the data storage part of the application from the frontend - especially something that lives in the user's browser. But the most important ones are security and storage. It is highly insecure to bundle the business logic within the Angular application. It is also a very bad design choice to bundle the entire package together that has to travel through various networks.

In summary, the implementation of the backend is as follows: the Python backend starts a HTTP server using Flask and accepts HTTP requests and processes and retrieves information according to the request. The data is stored in a MongoDB database which allows unstructured documents to be stored and retrieved easily.

The application also uses the concept of **Social Circles** to the Social Platform, Google+ [5]. That is, a user may choose to share his/her events with a small set of people. This is defined through the "Circles" concept which circumvents the Share with All or None limitation of some other social media.

The basic concept here is that a single user may belong to multiple circles of another user and the other user may have a way of directing the content specific set of people. This is illustrated in the below diagram:



At this stage, the application contains the following circles built-in: Close Friends, Friends, Family, Others and Public. There is , however, no mechanism of creating new user defined circles in addition to the ones provided.

4 Hypothesis and Existing Results

- As the end-application would rely on human computation to post and rate events, there need to be an incentive for the users to do so accurately [2].

The incentive in this case would be to prevent self-damage. For example, as mentioned in subsection 2.2, by posting incorrect information the user risks getting voted down and eventually having his/her events dropped by the application. On the other hand, if the user incorrectly votes an accurate event, the user will also suffer the consequences of the action.

The above mechanism to distinguish high-quality content from others is similar to the one described for Yahoo! Answers in [4]. By aggregating vote counts, this application aims to generalize these results for other social media as well.

- By giving the user the option to share a post publicly or to a subset of his/her friends-list, the application also guarantees to protect the privacy of the user.

  This can further be improved by providing the user with the option to post events anonymously. However, the votes on an anonymous post would still be linked to his/her account ensuring that no free ticket is given to the user.

  Apart from ensuring privacy of the user, this concept also ensures that the user avoids the all-or-nothing dilemma in sharing events that exist in many other social networks [5].

5 Target Audience

This application is intended to be used through hand-held devices such as a Smartphone or Tablet PC so that the application can be updated in near real-time.

Being a web-based application with support for various screen sizes makes it more accessible. Any device with a web-browser would be able to access it. Since the RESTful services are provided as an independent layer, it is also possible to create native applications for various platforms that use the same API.

References

[1] Fernando Espando, Victor M. Guerrero *Survey with negative questions for sensitive items.*

[2] Galen Pickard, Wei Pan, et. al. *Time-Critical Social Mobilization.*

[3] Rachel Pain. *Social geography: Participatory Research.*

[4] Agichtein, Eugene, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. *"Finding high-quality content in social media."* In Proceedings of the 2008 International Conference on Web Search and Data Mining, pp. 183-194. ACM, 2008.

[5] Kairam, Sanjay, Mike Brzozowski, David Huffaker, and Ed Chi. *"Talking in circles: selective sharing in google+."* In Proceedings of the SIGCHI conference on human factors in computing systems, pp. 1065-1074. ACM, 2012.