A Crowdsourced Solution to Geographic Cataloging

In today’s digital age crowdsourcing is becoming a popular method for funding, project collaboration, and data collection. It uses the unique approach of enlisting services from a large number of users, typically unpaid, through a universal medium like the internet. Beyond the actual completion of a task, crowdsourcing also offers a unique chance for data analysis based on location, demographics, clusters, and patterns. A project based around this unique cultural element is interesting and has a lot of potential for growth.

Catalog (a temporary name before something more creative can be established) is a crowdsourced cataloging application spanning several disciplines of computer science. It catalogs geographic points, measured in latitude and longitude, using a mobile device. For the sake of this proposal we will take the example of a Yale School of Forestry student wanting to catalog all the Elm trees in New Haven. The final application will consist of three parts: a web interface, a mobile application, and a relational database.

The full application will flow as follows:

1. A user will enter a url and arrive at the web interface. He creates a profile with a name, email, password, and other personal contact information to be determined.
2. The user logs into the dashboard. From there he will have the option to view one of his catalogs, edit one of his catalogs, or create a new catalog. At this preliminary step, the dashboard will be empty so creation will be his only option. There will be three types of catalogs a user can create.
   1. Private - This catalog does not follow the crowdsource idea but is available for people trying to build their own data sets. Only the creator is able to contribute but they can decide whether the catalog is visible to the public or not.
   2. Contributor - For this catalog, the creator will invite other user’s by their email address to join the catalog. The invitees will not be able to edit properties about the catalog itself, but they can mark geographic locations. The creator can decide if the catalog is visible to the public or not.
   3. Public - In this catalog, any user can contribute to and view it. However, it has a unique feature in that every geographical point has a confirmation rating. Users can give a positive or negative rating to each point, removing some of the potential for incorrect points.
3. After a catalog is created the user will exit the web interface. Using an iOS device, they will then use their email and password to log into a native application. They will see a list of catalogs they’ve created or have been invited to contribute to. They will select the list they want to add a point to.
4. In the case of the forestry student, the user will walk up to an elm tree. The user will enter any information required by the catalog creator for a point (size, age, name, color, etc.) and then commit the information. Along with the populated fields, the application will also submit the device’s current longitude and latitude coordinates.
5. Once a point or collection of points has been logged, the user can return to the web interface to see a list of the points and the points laid out on a map interface.
The scope of this project in its entirety is large and beyond the possibility for completion during one semester worth of work in CPSC 490. However, the final product should be able to complete the necessary functions to maintain the spirit of the full application. It will also serve as a starting point for continued work and eventual completion.

The process will begin by building the database in MYSQL. The database is going to take the full application into account and will be created with everything in mind. Although not all of it will be used in this project, it will be able to handle any features added on post-project. It will need the ability to store:

1. User information including name, email, and password. Other fields are to be determined but will not be required.
2. Catalog information including which user created it, what type of catalog, which users have permission to view and/or contribute to the catalog, and the type of information required for each point (name, description, photo).
3. Point information including which catalog it belongs to, the information required by the creator, the geographical coordinates, and the confirmation rating (for points on public catalogs).

The web application will be a toned-down and unpolished version of the full application. The next step will be to build a piece of it using HTML, CSS, and PHP. This will include the user account creation form, the user login form, and the dashboard. The dashboard will give the ability to create and delete any of the three types of catalogs. This will include a way for the user to invite other users (by email) to contribute. The web portion will also include an API endpoint to communicate with the mobile application.

The next step will be to create a native iOS application to do the actual geographical cataloging. It will be programmed in Objective-C using Apple’s Xcode environment and tested on my personal iPhone. It will communicate with a web API and include a login form, the list of current catalogs, and the ability to add to a catalog. The adding will occur with the push of a button. It will upload the current latitude and longitude coordinates to the database.

The final piece will be to go back to the web interface and, using a set of submitted points, use a map interface to display them.

This application has a variety of uses, ranging from personal to business to academic. For example a college student could be backpacking through Europe and want to pinpoint a location and an associated picture for every stop along the way. Or a local township might want to know the location and condition of downed trees and power lines after a severe storm. Or a scientist may want to keep track of the exact location along a river bed where he took water samples for a study. Beyond the purposes for data collection, just having the data available could be useful as well. New parents could use a catalog of public schools in the area when deciding where to move. A restaurant franchiser could find the locations of his competition to directly combat them. Someone may want to live vicariously through someone as they road trip across the United States. There is definitely a use for it and the amount of potential data is endless.
The deliverables for this project will be presented as part of the entire application but also as source code. They are:

1. A MYSQL database built to support the entire application flow. The create statement will be submitted as source code.

2. A front facing web interface for the user. The code will be packaged (along with deliverable #3) and submitted.

3. A backend API to communicate with the iOS application and retrieve data for analysis. The code will be packaged (along with deliverable #2) and submitted.

4. A native iOS application that runs on an iPhone 5 using iOS7. The source code will be packaged and submitted.

5. A paper describing the process, showing a demo of the application, and talking about the potential add ons, data analysis, and direction for after this stage of the application.

If the project goes smoothly and I’m able to do a little extra work to further the end result, I’d like to implement the ability to specify fields on each point (this will be setup in the database, extra will be implementing in the web interface), introduce a photo element, or include a data-mining algorithm like clustering or nearest neighbors. However, those will be optional and determined near the end of the semester.

Overall, the project explores an interesting area. Crowdsourcing is becoming popular and more importantly easier now that everyone is connected with smartphones. Applications should start taking advantage of that. This project has a breadth aspect in that it will span several different digital mediums - web, mobile, databases - and require me to jump around different technologies and get them to work in sync. It also has a depth element, the database will have to be robust, efficient, and make good use of relationships. This will be particularly interesting considering the possible dynamic nature of data being collected.

Everything is to be submitted by noon on the last day of reading period.