UNIVERSITY OF SOUTH FLORIDA
COLLEGE OF BUSINESS

10/19/2011

T. GRANDON GILL

PATHFINDERS EUREKA[1]

> *So far, we have funded development of the project from the money we earned for our work during the BP oil spill. We have nearly reached the end of that resource, however, and I have alerted the group that it is unlikely that there will be any more funding.*

Scott Lewis, the founder of the *Eagles' Wings Foundation (EWF)*, a not-for-profit, public foundation created to provide assistance to disaster survivors and Emergency Management personnel following any major disaster in the United States and Worldwide, pondered the current state of a technology venture that had led him far afield from his normal commercial and non-for-profit activities while he waited for his dinner. Less than a year before, the Pathfinders Task Force that he had organized had played an important role in identifying the extent of the worst oil spill in the history of the Gulf of Mexico, precipitated by an explosion on *British Petroleum's* Deepwater Horizon oil rig. Central to these efforts had been software—cobbled together from third party vendors—loaded on cellular telephones that allowed the task force to display spill data and many other types of information on headquarters maps in real time. Buoyed by numerous best practice commendations awarded to the task force for its innovative practices over the course of its 189 day deployment in the Gulf, members of the team began to envision many other non-disaster scenarios in which the same core functionality might be beneficial. Chief among these was the belief that a redesigned application could be for the basis of a new form of social media, one that was organized around maps, user-initiated surveys and flexible group formation. Lewis and his group of four key employees become so convinced of the potential that he decided to invest the net revenue gained from the Gulf operation into the development of completely new application build offering a clean interface to the functionalities that they found most useful.

The project had begun with a false start; a third party developer hired to build the application proved unable to deliver. Early in 2011, the company decided to manage development in-house, using contract programmers for most of the coding. By September 2011, it appeared that the project—referred to as *Pathfinders Eureka*—was nearing completion. Nevertheless, there were many concerns in Lewis' mind. The team envisioned a free version (supported by advertising) and a paid version of the application; a clear strategy for establishing pricing and channels for the latter had not been finalized. The software could be deployed in two different topologies (command-and-control and social); the team felt that there would be synergies between the two but had not provided a complete rationale for this belief. The team had little commercial software development experience; issues relating to patents, testing and support had yet to be resolved. Finally, the money from the Gulf was running out. If these issues were not resolved shortly, would it make sense to pull the plug on the project or should he use funds from his other businesses to keep them project going (in light of its tremendous potential)?

Scott Lewis, EWF and Pathfinders

Scott Lewis could be characterized as the prototypical entrepreneur (if such exists…). After graduating from Duke in the 1970s, he established a series of profitable businesses. A native Florida resident, he had been particularly successful in providing landscaping services. In the 1990s he sold one such business and, shortly after, re-started up another in a different location that proved equally successful.

Disaster Relief Activities

Strongly committed to the idea of volunteer service, Lewis shifted from the volunteer fire service to devote much of his energies to disaster relief in 1999. As described on the foundation's web site:

> The Eagles' Wings Foundation (EWF) was founded in 1999 during the Bahamian relief efforts following the catastrophic damage inflicted by Hurricane Floyd. Seeing that donations were not reaching survivors, that volunteer efforts were chaotic, and military relief was not coordinated with private relief, Scott Lewis founded EWF to fill the leadership gap, and provide a safe place for private donations. Scott was appointed by the Prime Minister of the Bahamas to the position of Incident Commander for the Abaco Relief Command and the first EWF leadership team was formed (http://www.theeagleswingsfoundation.org/index.php/history#2).

Because the activities of the EWF were typically initiated in response to crises, Lewis established an adjunct organization, *Pathfinders Task Force (PTF)*—named after the World War II, D Day, volunteer paratroop battalion that parachuted into enemy territory to guide in the initial invasion forces—whose function was to act as "the tip of the spear" for other incoming units. Subsequent to Hurricane Floyd, the PTF has participated in numerous disaster efforts, including Hurricanes Charley, Frances and Jeanne (all hitting Florida in rapid succession during 2004) and Hurricane Katrina (2005). In just the past year and a half, PTF units had deployed for the Haiti earthquake (2010), the BP oil spill (2010), the Japan earthquake and tsunami (2011), the Mississippi floods, and Georgia tornadoes. At the time of the case, Lewis has just returned from the Bahamas, where the PTF had been instrumental in assessing the damage from Hurricane Irene with an unusual commendation from the Bahamian Prime Minister on the success of the team and its software..

The principal activity performed by PTF units was cataloging the extent of each disaster and identifying individuals and situations requiring immediate action. Towards this end, volunteers—trained by Lewis' organization--frequently went door to door filling out damage reports and checking for survivors. Traditionally, these reports had been paper-based since the locations surveyed tended to lack both power and cellular service in the days immediately following the disaster. The inherent limitations of this process became most evident to Lewis during the aftermath of Hurricane Katrina when he handed a stack of nearly 7,500 completed forms to the incident commander and received a look of dismayed disbelief as his reward. At that very moment, Lewis became determined to find a better solution.

Initial Solution

The first solution employed by PTF involved adapting commercial, mobile software originally developed for the purpose of managing fleets of trucks. The application had two components, a server (deployed on a computer or laptop) and a client, deployed on an internet-enabled cellular phone. Lewis modified the

fleet application with a 2007 patent pending module which upgraded the mobile application to allow a second critical need of the PTF. The system now could:

1. Send out rudimentary forms that could be filled in on the cell phone
2. Store form information and GPS position even when no cellular network was present.

Together, these capabilities allowed volunteers to gather information on their client phones even when power and cellular service had been disrupted. Although the fleet application did not initially allow pictures to be recorded as well, this capability was later added, as a result of PTF's requests to the vendor.

The State of Louisiana heard of PTF's successes in tracking in the Haitian earthquake from an After Action Report published weeks earlier by FEMA. Seeing significant potential use for the Gulf oil crisis, in 2010, the State tasked in the team, with its cache of E bay purchased phones used in Port au Prince. After initial trials in the 100% disconnected environment of the Gulf of Mexico, the Incident Command teams spearheaded by US Coast Guard personnel had PTF purchase 400 brand new, military spec. cell phones with far wider hardware capabilities. The PTF phone application had evolved significantly (see Exhibit 1 for photos of the handset used). This event represented a major departure from the PTF's typical deployment in a number of ways:

- *It was much longer.* Whereas the typical PTF volunteer deployment was 10 days or less, the commanders in the Gulf incident required services for over 6 months. Since this was unrealistic for a volunteer deployment, Lewis used *Disaster Solutions, LLC,* the commercial company set up for non-volunteer activities, to bill the related agencies.
- *It involved many data gatherers outside of the PTF volunteers.* As part of BP's response to the spill, a special program was set up to provide alternative employment to individuals unemployed as a consequence of the spill. For example, one program called *Vessels of Opportunity* sent thousands of fishermen out to monitor for oil on the surface of the Gulf. These individuals were trained in the use of the PTF phones and made thousands of observations.
- *It was much more varied in the types of information gathered*. As opposed to damage and humanitarian data gathering, individuals with PTF phones were sent out to assess the status of many different things (e.g., oil in wetlands, wildlife), and the new hardware allowed for geotagging 2 MG photos, which were critical for documenting oil spill incidents.
- *It involved many overlapping organizational structures*. These included elements from FEMA, the Coast Guard, local parishes, BP and the state government. Each had both common and differing information needs.

In each case, the data and photos acquired through the cell phones was relayed back to central stations (either through the cellular network, where available, or through a Bluetooth connection). There, it was overlaid on maps, becoming a key tool in managing the response to the crisis. For example, it proved instrumental in the placement and repositioning of booms to protect the local marshes.

What soon became very clear to Lewis and the other members of the task force was the effectiveness of the technology (see Exhibit 2 for a sample map and some statistics). Among the many different entities coordinating the response, the PTF data gathering activities were repeatedly singled out as best practices.

Pathfinder Eureka

As a consequence of their experiences in the Gulf, Lewis and his team began to consider creating their own application from scratch. The team shrunk in size as Lewis worked to streamline the evolving development, with initial false starts on several angles. While the application that they had used in the

Gulf had been widely acclaimed, its origins as a tool for trucking fleet management increasingly limited its adaptability. In particular, they envisioned a completely new application that would:

- Work on many devices, including smart phones, tablets and PCs
- Permit management from any device, not just from a server
- Not entail the exclusive use of proprietary (and costly) mapping technologies such as ESRI, whose applications dominated FEMA and other agencies
- Support overlapping organizational structures and social groups
- Facilitate the coordination of networks of individuals organized around a mapping theme
- Create matching and branching logic within the software to quickly merge personnel and assets in times of an emergency ramp up, while transposing this into day to day functionality

The potential offered by the last two of these capabilities were particularly exciting. Combined with the other features that they planned to implement, Lewis and the others recognized the possible emergence of a new map-based flavor of social media. Such a tool could exist either on its own or could be embedded within other social channels, such as Facebook. The opportunity so excited Lewis that he agreed to apply the extra funds accumulated from the oil spill—in excess of $500,000—to the development of the application.

Features

The group code referred to the new application as *Pathfinder Eureka*, code named after the Eureka beacons used by the Pathfinder battalion to guide gliders and other transport planes into Normandy on D-Day during World War II. The proposed feature set of the Eureka product is summarized (from a 36 page feature summary) in Exhibit 3. These included not only their "wish list" of major characteristics but also included many additional enhancements whose value became apparent as they begin to look at the detailed design.

In the broadest sense, the application was to be designed so that it could operate in two distinct modes: *command-and-control* and *social*. As illustrated in Exhibit 4, the command-and-control mode involved a star topology, with the system enabling 2-way communications between a hub and the members of the group. The hub would control access to the system, determine what forms would be distributed to participants, control all databases and would maintain control of the map. Participants would collect geo-coded data (forms, photos, messages) to be sent back to the hub. The command center at the hub could periodically choose to make maps and reports available to participants, but such information would not normally be available in real time.

The social mode of the application was built around a peer-to-peer topology. While the capabilities of the system would be similar, all participants in a particular social network would have access to the map, would be able to design forms and view reports, and would be allowed to opt-in or opt-out of access at any point in time. Participants could be members of many different social networks at any given time, and could control whether or not they could be viewed by others in each network. The team anticipated these features would not only support ad hoc networks, such as groups of friends or study groups, but would also support organizations that were not built around strict hierarchical structures, such as clubs and churches. This aspect could, in theory, link vast volunteer resources in times of disasters to command and control units.

Industry and Competition

Lewis and his team recognized that they were not alone in using GPS data as a way of bringing users together. In fact, the company envisioned three different groups of potential competitors:

1. Fleet and asset management
2. Volunteer coordination and control
3. Location-based social networking

Each of these groups had different characteristics, but all had potential disaster linkages.

Fleet and Asset Management: With GPS-enabled cell phones becoming nearly universal, a robust marketplace for using these devices in transportation and other industries had developed. Broadly speaking, competing products emphasized three different capabilities. Fleet management emphasized using GPS to locate and track vehicles, allowing a dispatcher to assist drivers as well as monitoring driver performance. Asset management features were particularly useful for geographically tagging inventory and high value equipment; bar code support was often included with this capability. Data gathering capabilities supported a wide range of possible needs—from clocking in employees to acquiring information from customers. Some of the key competitors in this industry are presented in Exhibit 5. Of particular note, an approach to pricing based on a monthly fee per mobile device is standard in the industry.

Volunteer coordination and control: Based upon Lewis' extensive experience managing rapidly assembled volunteer task forces, he recognized the potential value of the Eureka product for this activity. Participants in this industry included *Volunteer Integrated Management System* (VIMS) and *Volsoft*. VIMS pricing was based on a monthly subscription, with fees based on the number of registered volunteers. These ranged from $24/mo. For 25 volunteers or less to $80/mo. for up to 500 volunteers. Volsoft was priced at a flat $695. Both packages were oriented towards administration and did not use GPS data at all. Thus, they seemed more likely to complement Eureka than to act as direct competitors in this space.

Location-based social networking: The location-based social networking space was substantially larger. In a 2010 study, industry expert Claudio Schapsis identified roughly 150 competitors participating in the space, as well as over 20 competitors that had discontinued social networking operations[2]. As illustrated in Exhibit 6, the industry had a wide variety of participants. At one extreme, giants such as Facebook made some use of location-based information and provided portals/apps intended for mobile users. At the other extreme, more directly relevant to Eureka, were a number of participants focused specifically on geographic-related presentation and activities. These included participants that featured location-specific guides (e.g., Gowalla), participants that provided maps for locating friends (e.g., Mologogo), participants that combined friend location services with providing information about local businesses and services (e.g., Google Latitude, Loopt), participants providing similar services without using a map-based interface (e.g., Foursquare), and participants that featured map-based game activities such as scavenger hunts (e.g. SCVNGR). Many of these services originated in high-tech centers such as the Silicon Valley and Austin, Texas and had received substantial

[2] Source: http://bdnooz.com/lbsn-location-based-social-networking-links/#axzz1ZAggSAQv

venture funding. Many had also forged tight linkages with Facebook, the premier social networking site. A number of these services are summarized in Exhibit 7.

One of the key aspects of geo-location that attracted competitors was its potential for advertising revenue. For example, banner ads keyed to location were typically billed at three times the normal rate or more. In addition, local businesses would often pay to become part of a game or to become an attractive target for reviewers.

Marketing

The elements of the Eureka marketing plan, as of September 2011, are summarized in Exhibit 8. Karl Pfister, who had joined the company during its Gulf deployment shortly after receiving his MS in Entrepreneurship from the University of Florida, was jointly responsible for business development at the company with Cameron Kirkpatrick, who had majored in broadcasting as an undergraduate and had worked in sales. The two of them envisioned Eureka being marketed as two products:

1. *A free product*. This would be available for download as an App for various devices (e.g., PC, tablet, smart phones) and would provide a limited set of features. For example, it might not allow unlimited form creation, would not provide a report writer (for compiling and analyzing results) and would provide limited ability to control access.
2. *A subscription product*. This would available to users for a flat monthly fee, without the "per user" pricing typical of the industry. They originally anticipated charging somewhere around $50/mo. with microtransaction upgrades for enhanced reporting and additional form creations, although pricing policy was still under discussion. The subscription version would offer the product's full feature set and would provide organizations the ability to create a protected silo around their network when operating in command-and-control mode.

Paralleling the competitive analysis, Pfister felt that each of the three potential markets for the product involved different use cases and would likely require different channels. Use case for fleet and asset management would involve functions for managing remote workers (e.g., clock in, clock out, expense report forms, geocoded photos, messaging) and for tracking transportation. Managers also would benefit from custom reporting from form databases. These capabilities would mirror those of existing competing products but would be available at a much lower cost and would be more readily customizable by the user. Sales channels would include word-of-mouth from early adopters and beta testers, web search and, possibly, telemarketing. In addition, Pfister and Kirkpatrick expected that many individuals who would become familiar with the system as social users would develop an interest in using a paid version for work-related purposes.

Matt Campbell, who worked as a Pathfinder Task Force Leader, was particularly excited about the potential use of the Eureka product for volunteer management. He noted that while non-profit organizations spent millions of dollars on volunteer registry systems nationally, none of these systems were geocoded. In many situations, however, knowing where a volunteer is located is critical in using his or her talents effectively. By combining location with readily customizable user profiles that identified relevant skill sets, the efficiency of volunteer deployment could be

enhanced for nearly any organization. Special channels—mirroring those established for commercial customers—and pricing could be established for non-profit and public sector customers.

The use case for social users was primarily recreations. In social mode, friends could share their location, query them with forms, leave geocoded messages and photos and establish separate circles for different groups of friends and colleagues. The free social version would be marketed through various App stores (e.g., Google, iTunes), on the web and through viral channels, such as YouTube videos. The principle revenue stream would be advertising, which would be billed at high rates as a result of the position information made available. In addition, having a free version would permit rapid testing of new features as a prelude to introducing them to the commercial version.

Development of Eureka

Developing the Eureka application had proven to be challenging. Originally, Disaster Solutions had contracted the entire development effort to a 3rd party. By early 2011, however, it had become clear that this approach was not working. A substantial part of the problem stemmed from the fact that the vision of the final product was evolving even as development was occurring. This evolution required developers to adapt continually; such adaptation was inhibited by having developers acting independently of the PTF designers. In spring 2011, the relationship with the external developer was terminated. Instead, the PTF team chose to take a far more active role in managing the development process.

Eureka Architecture

In creating applications for mobile devices (e.g., cell phones, tablets, laptops), there are nearly always decisions to be made regarding what components of the system reside on the device itself, and what components are placed on external computers—known as servers—whose physical presence can be nearly anywhere. The key architectural decision made in designing Eureka was positioning itself on the client-centric to server-centric continuum.

Client-centric: Where an application demands high performance and limited (or zero) connectivity, most (or all) components are normally placed on the device. This typically requires a great deal of separate coding for each type of device, since development of device components for one device, such as an *iPhone/iPad* involves one set of development environment/operating system/programming language (e.g., Xcode, iOS, objective C/Cocoa), that are completely different from those used for a different device, such as an Android phone (Eclipse, Linux, Java).

Server-centric: Where an application demands access to a lot of shared information or data and needs to be deployed on many different types of devices, on the other hand, an alternative approach is to place most of the application on a server and to use the internal web browser build into the device to deliver the application's user interface. The "theoretical" advantage of this approach is that the underlying languages used to describe web pages and build interactivity (HTML and JavaScript) are supported by all web browsers. In practice, however, robust applications demand considerable customization, both as a result of the essential characteristics of the device—e.g., the display size of laptops and cell phones are so different that a display tailored to one is unlikely to be well suited to the other—and owing to quirks of

specific devices—e.g., the Apple iPhone and iPad do not support the Adobe Flash standard that is widely used to animate web content.

The nature of the Eureka application made a server-centric architecture a much better fit. For both economic reasons and in the hope of avoiding dependence on a single vendor (such as Microsoft), the team settled on two key technologies both of which had their roots in the open source community:

- *Ruby-on-Rails (RoR):* Server development used the open source Ruby programming language supported by the Rails framework for developing web applications. To maintain database information, they chose the open source PostgreSQL. (Another possible open source database, MySQL, had been rejected, in part, because it was sponsored by commercial database giant *Oracle*).
- *Sencha:* The Sencha toolset provided developers with the ability to define interface interactions for various client devices. The client-side program would then generate web pages and JavaScript code specifically tailored to the device characteristics. As a result, the Sencha developer could design interfaces suitable for different physical device categories (e.g., cell phone, laptop, touch-screen tablet) and the tool would generate a web page that would adapt to both the category of the device and the specific browser/vendor. The tool's roots were open source, but the vendor also offered commercial versions that did not require revealing source code.

The server side of the application was hosted on the *Amazon Elastic Cloud* (EC2), rather than on dedicated servers owned by the company. This architecture ensured that Eureka could be scaled up rapidly to accommodate growth in user demand. It also provided a reliable service at a relatively low-cost and did not require the team to get involved in server maintenance.

The fact that open source code was used in development did not mean that it was no-cost. Open source developers had a number of alternative approaches to acquire revenue streams. These included one or more of the following:

- Providing the software for free or for a nominal charge (typically associated with packaging the necessary components together) then charging for support
- Creating a free open source version of the software and a "premium" version that included separate components that were not open source
- Offering software-as-a-service (SaaS), where the company installed the necessary software components on its own server (or in "the cloud") then billing the customer for subscription to the service.

The team's extensive use of open source tools had, thus far, kept the cost of purchasing development tools low. It did mean, however, that careful attention needed to be given to the issue of licensing. Some "free" tools, such as those in the Sencha toolset, required that the developer include an application's source code with any product it shared if modifications had been made to the original code. As a consequence, the team sometimes chose to pay rather than use free software in order to avoid such requirements.

Eureka Developers

The team member responsible for managing the day-to-day development of the Eureka project was John Simion, IT Project Manager at *Disaster Solutions*. Although he coordinated the development activities, Simion did not characterize himself as a programmer. He had started an undergraduate degree at the *University of Florida* in engineering, but had quickly lost interest in that field and chose leave the school to pursue his other interests. Since that time, he had become fascinated by technology and, for the most part, had taught himself the technical skills he required.

Once *Disaster Solutions* had taken over development, Simion and other members of the team had assembled a far-flung group of contract programmers. At the time of the case, the group consisted of six individuals, three of whom were located in California, one in Canada, one in the Ukraine and one locally, in West Palm Beach, Florida. In addition, a group in Fort Lauderdale, Florida (who contracted two programmers in India) had already completed a version of the client application intended for regular cell phones, whose physical limitations made them suitable for data gathering but not for map display. The contract developers were typically paid on an hourly basis, at rates well below what they would demand for other commercial programming projects. This was possible because of the exciting nature of the project and the fact that developers were allowed considerable opportunity to apply their own creativity to their work, rather than being forced to code to strict design specifications.

The server-centric architecture allowed for the separation of development projects. Specifically, one developer was responsible for the RoR server code, another for the database design and the remaining developers focused on the browser-based front end. Simion spoke to the developers using Skype 2-3 times per week. In addition, every one to two weeks he would write a development plan that would be shared with the group. An extract from a plan is presented in Exhibit 9.

One particular challenge that the project had faced was limited availability of Sencha developers. As a relatively new entry to the marketplace, only a small group of individuals had extensive commercial experience. Experienced Sencha developers therefore tended to be quite expensive. Although this presented a bottleneck in the early stages of the project, by the time of the case it was not clear that bringing on new developers—who would then need to be trained on the Eureka application itself—would facilitate completing the project.

Eureka Development Status

By mid-September 2011, Simion felt that development of the first Eureka version was nearing completion, with a full version expected by November. Most of the core server side functionality (database and RoR application) had already been completed, so much of the activity centered on building the user interface. This interface consisted of dozens of screens, used for viewing positions on the map, creating and filling out forms, defining and attaching tasks/events, specifying user profiles, creating groups and assigning membership, writing and editing memos, downloading data, and so forth. Samples of selected screens are presented in Exhibit 10.

Progress towards completion was tracked informally, based largely on identifying features that had been implemented versus those that had not. When new features were implemented, Simion

and other members of the team tried them out. He anticipated that a more systematic testing process would be implemented once initial development of the application was complete.

Current Situation

As the month of September 2011 neared its end, Lewis recognized that the time had come to finalize some decisions. A few months into the future, the reserve of oil spill cash that had been used to fund development would be exhausted. Without a clear plan on how to move forward, that could spell and end to the project.

Among the decisions that needed to be made:

1. *How should the markets for the Eureka application be prioritized?* The "command and control" market was obviously the one that was most familiar to the company. On the other hand, the "social" was much larger, more in tune with the buzz of the times, and more exciting to developers. Did the company need to choose, or could it pursue both?
2. *Should the company be trying to exploit the synergies between the commercial and social versions of the software?* To what extent would users of one product be attracted to the other? If the synergy was not great, would it make sense to separate the two products, perhaps even placing them in different organizations?
3. *Should the company be selling a product or a service?* Was Eureka to be sold as software, for a set price combined with free or paid support, or should be sold through the SaaS model, with Disaster solutions providing the servers and charging a subscription fee?
4. *How much money needed to be budgeted for marketing and sales?* Given the low-price the team envisioned for the Eureka product, it would be difficult to self-fund a rapid roll out of the product to paying customers. How much should be invested in seeding the process until it became self-supporting?
5. *At what point should the product's feature set be frozen?* The ability of developers and the Eureka team to adapt had made the development process a highly creative one. Such processes were not unusual in the agile development methods that are commonly used to create cutting edge software. But with the money running out, did it make sense to freeze all enhancements until a fully working product was delivered?
6. *How much time needed to be set aside for testing the product?* Even when the Eureka product was delivered in its final form, it would presumably require some testing before it was released to customers. How should that testing be accomplished and how long could it be expected to take?

In the back of his mind, Lewis further wondered what to do if the answers to these questions were such that exhausting the existing funds could not be avoided? Would it make business sense to allocate additional money from his existing companies, recognizing that such funds could otherwise support his non-profit rescue activities? Would it be possible to acquire funds from other sources, such as investors or even early customers of the product? Or would it simply make more sense to chalk up the Eureka project to "a learning experience" and move on to some other project?

He considered these questions as he waited for his dinner at the local pub. It would be hard to concentrate on the food, when it arrived.

Acknowledgements

This case study is based upon work supported by the National Science Foundation under Grant No. 1043919.

Biography

Grandon Gill is a Professor in the Information Systems and Decision Sciences department at the University of South Florida. He holds a doctorate in Management Information Systems from Harvard Business School, where he also received his M.B.A. His principal research areas are the impacts of complexity on decision-making and IS education, and he has published many articles describing how technologies and innovative pedagogies can be combined to increase the effectiveness of teaching across a broad range of IS topics. Currently, he is Editor-inChief of *Informing Science: The International Journal of an Emerging Transdiscipline* and an Editor of the *Journal of IT Education.*

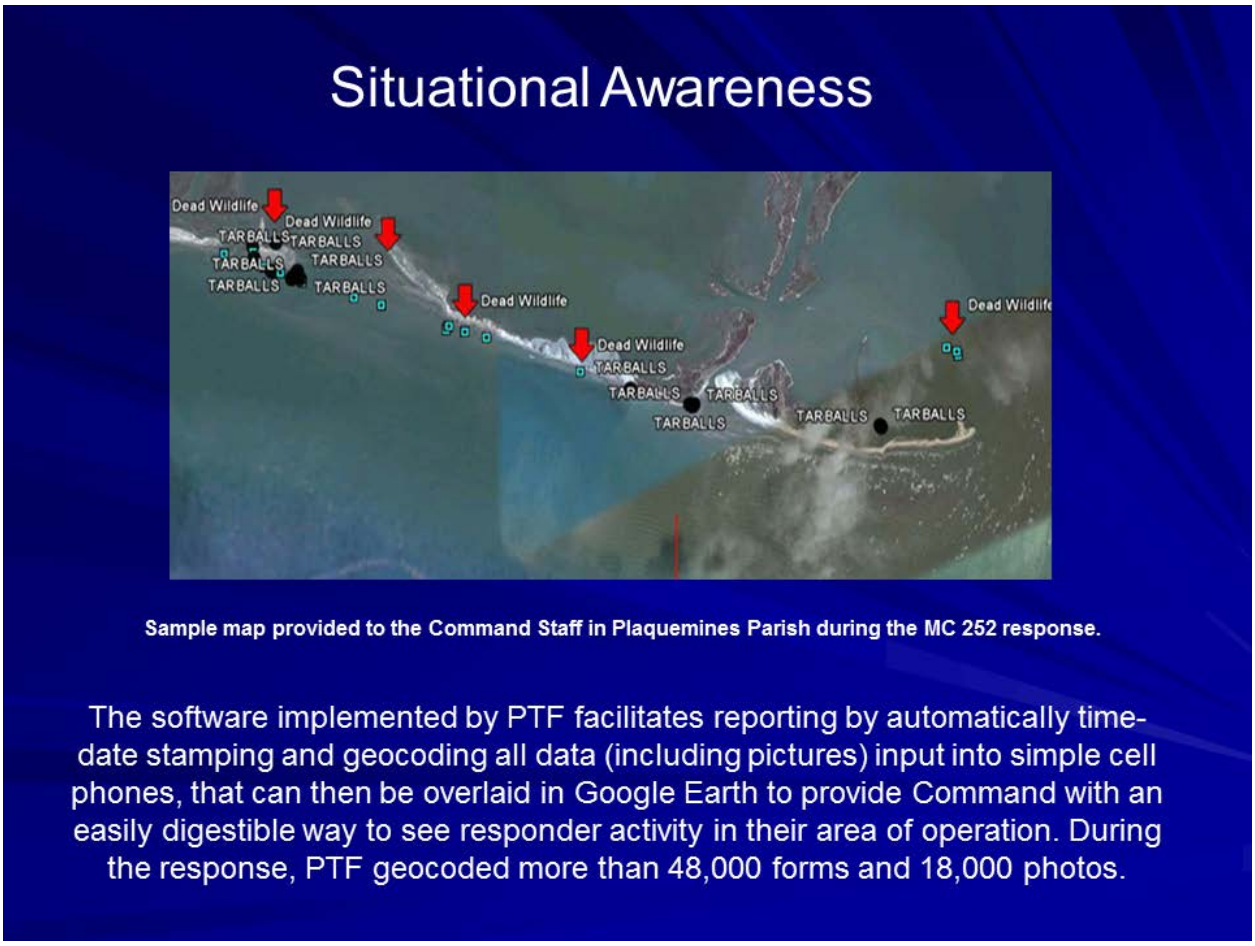
Exhibit 1: Pathfinder cellular handset

Pathfinders' Patent Pending Software was Pre-Loaded by PTF Technical Specialists

With one click, Field Users were presented with up to 40 different forms with drop downs and fields, customized for each Branch

Source: *Pathfinder Task Force* after action report for Gulf oil spill deployment

Exhibit 2: Pathfinder Task Force in Gulf



Source: *Pathfinder Task Force* orientation slide show

Exhibit 3: Eureka feature summary

1) **Private "Silo" Concept-** Every End-User organization could have their own Private sector of the database where they store all of their affiliated users and business / organization / company affiliations as well as all electronic data submissions. No other user may ever have access into the data contained in another user's Private "Silo" ~ All data transfer inside the network is done via an opt-in system, as in the only way data is shared is if one user chooses to "push" this data to another. Private Silos are not limited to just End-Users, any registered business / organization / company on the network also has their own private silo. Every registered Affiliation on the network shall have an administrator (generally the user who purchased the Affiliation)

2) **User/Account Profile-** Every End-User will create their own customized user-profile when they register for the Application and have logged in. This profile will display certain at-a-glance information about the user. If an end-user has created a business / organization / company, they have the option of creating a specific profile for this Affiliation and linking it to a location on the map. Users may not see personal data when they search the Application for other profiles to add as Friends.

3) **GPS Tracking & Social Networking-** Due to the nature of adding Friends and Affiliations, and the GPS Tracking component of the software, social networking shall be achieved via the usage of Electronic Submissions posted to other users. Users shall be able to post geo-located messages, notes, and photos to others and track them on a map, creating a social network through the Application.

4) **Organization Hierarchy Concept-** Certain "offerings" of the software (specifically for business / organization / company) shall contain the ability to list other affiliations as "Child" organizations under the "Parent" one. This will allow larger organizations the ability to manage other organizations underneath them via the Application. Child organizations (which have their own administrators) will be able to allow/disallow the Parent organization access to Personal Statistics of their users.

5) **Regional Commander Concept-** Approved Governmental Entities will have "Regional Commander" level access to their authorized users and resources, which is a special ability that allows the Regional Commander to issue alerts and offer requests to users in their jurisdiction to opt-in to the emergency response during a disaster.

6) **Enterprise Management Concept-** Similar to the Regional Commander, Enterprise Management refers to authorized Enterprise level access to authorized users inside a particular large business organization. Enterprise Management licenses have access to strong business analytics and the ability to create a large social network just for their Enterprise business.

7) **Organization Tools and Registry-** Contains all of the tools and structure to organize all user accounts inside "Groups" inside their respective "Company" (instance of the Application), as well as keep track of all devices assigned to users.

Users and Administrators will be able to quickly assign their users to Tasks, Needs, Events, Groups, or TBD. Pre-determined profile surveys & matching logic also allows users to be matched to other Companies as well as other users, allows Companies to be matched to users as well as other Companies, and allows users to be matched to Needs, Events, and Groups as well.

8) **Electronic Data Submission-** Allows users to submit geotagged and time/date stamped Form Data, Photos tied to forms, Work Orders, Events, and Places. Users may submit Customized Data Submissions are created via a drag & drop Form Creator or by completing Pre-Determined templates for Work Orders, Notes, Events, and Needs. All Electronically Submitted Data is saved "per-user" and per E-Submission name/type in the appropriate User or Company's database sector. Electronic Data submissions created via the Form Creator may be "deployed" to other users, meaning they will have access to this form and submissions are sent to the appropriate database sector.

9) **GPS Tracking/Map-** Contains all of the tools, architecture, and structure to plot Users, Devices, Form Submissions, Work Orders, Events, Places, and Custom Overlays. A historical track of all of this information may be obtained as well as an at-a-glance view of all activity on the user's network. A user will be able to "clock-in/out" to multiple businesses / organizations / companies at the same time, enabling simultaneous view of multiple organizations at the same time, as well as allowing the user to be viewed in several different organizations. User-Privacy is also controlled via the GPS Tracking/Map screen.

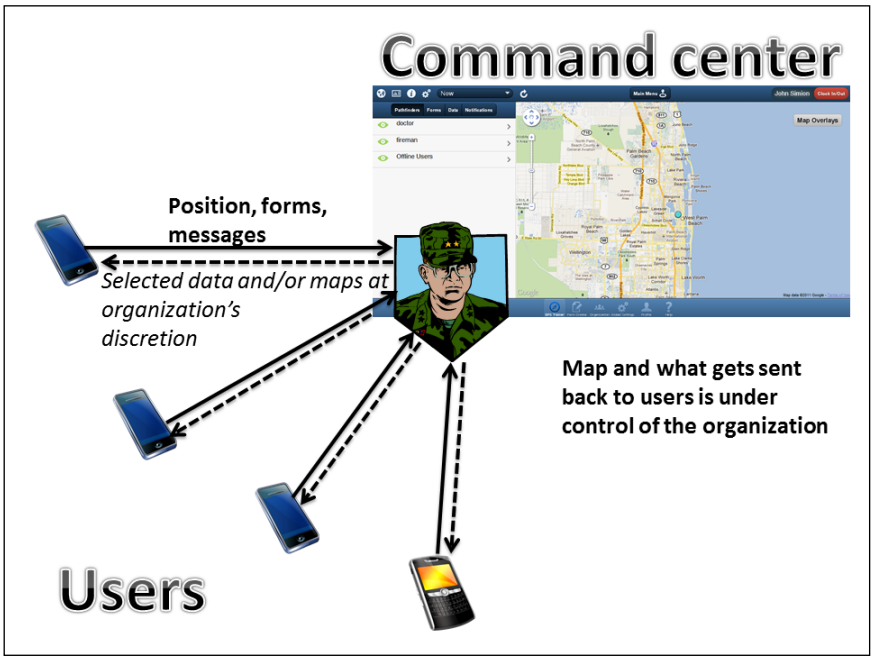
10) **Inventory Lists-** Users may choose to register assets into the software via a drag and drop interface. This will allow these users to map and track individual items, thus allowing donations management and critical-asset tracking. Inventory lists have pre-set classifications for the assets and may be created, exported, and deleted.

11) **Reporting/Data Export-** All user-submitted information that is logged into the database can potentially be analyzed and can be reported via Tableau Desktop (or other SQL Reporting Software) and allows for analytics to be run on any data-set existing in the software. The Application itself can only run pre-set SQL queries and manually export data to several formats, but the database may be exported to an external site in a format able to be integrated with SQL Reporting Software for analytics.

12) **Global Settings-** The Application shall have a dedicated screen for the purpose of changing global settings, including turning on/off location tracking & data transmission methods (Bluetooth, USB, WiFi), editing privacy settings, seeing all User Affiliations, and seeing information about data still on the handset (not sent to the server yet).

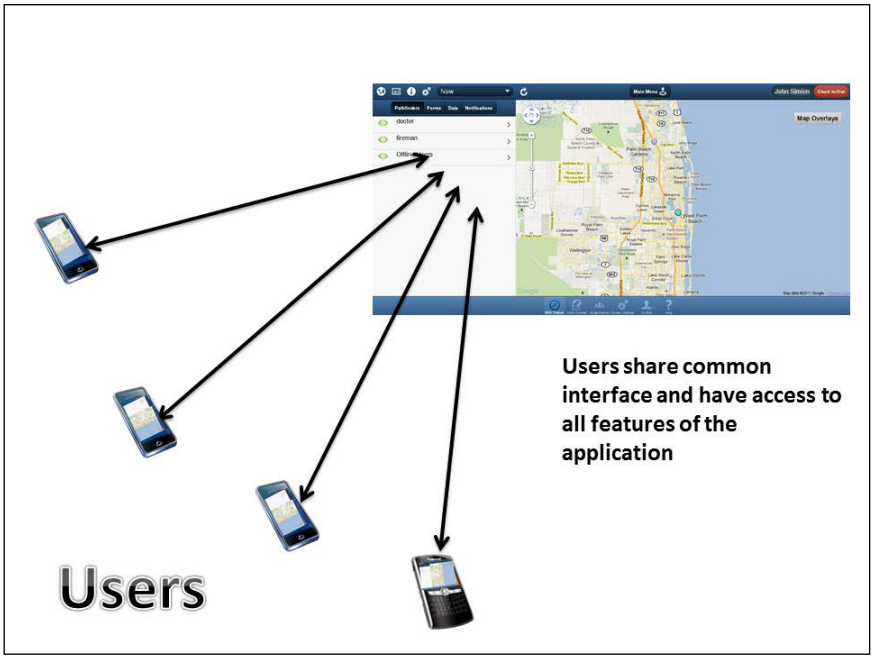
13) **User Help-** Due to the nature of the Application and the inherent complexities within, a comprehensive Help menu will allow users to easily understand how to use features and watch Tutorial videos pre-recorded using Sencha Touch.

Exhibit 4: Eureka modes



In *command and control mode*, users are principally a source of data. The map is viewed and forms are displayed in the command center. Information can be routed to user handsets although text-based phones may also be used. Information and instructions can be pushed out to users.

In *social mode*, the map interface is shared by all users. Generally, the system operates with participants treated as peers, although information is shared primarily using the map and messaging tools provided by the software, rather than through direct communications (which users may invoke using other cell phone capabilities, such as SMS). Users pull information from the map and application.



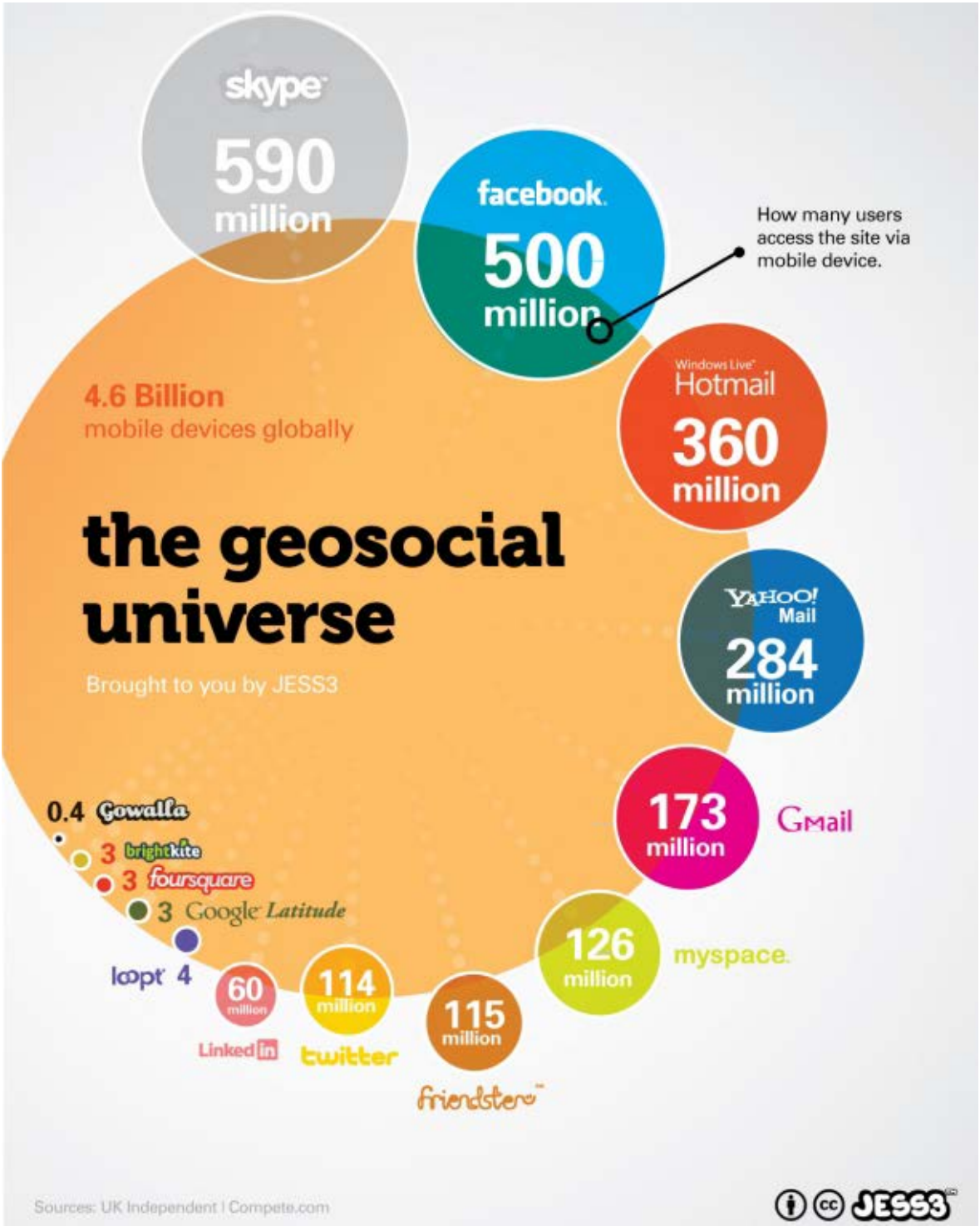
Source: Created by case writer

Exhibit 5: Competitive summary

Vendor	Description	Pricing
Xora	Fleet-management product supporting both cell phones and in-vehicle GPS devices. Allows forms to be sent to cell phones.	$30/mo. per device and up
Airclic	Software as a service (SaaS) mobile solutions hosted by the company. Provides GPS capability and data gathering (form) options. Offers industry-tailored solutions.	$40/mo. per device ~$7000 for setup and initial training $1500/day for customization
Datarrive	Geo-focused mobile solutions specifically tailored towards custom data gathering. Products include component modules and consultative services.	$60/mo. per device $15/mo. for the application $12/mo. per module
Telenav	Combines GPS with form-based capabilities, such as asset tracking. Also provides pure GPS apps.	$30/mo. per device
Actsoft	Provides a portfolio of GPS-enabled app products for fleet management, asset management and mobile worker management.	~$20/mo. per device

Source: Compiled from Disaster Solutions research by case writer

Exhibit 6: Location-based social networking



Source: Wikimedia commons

Exhibit 7: Selected participants in geo-social space

Service	Description	Reported Outside Funding
Loopt	Service founded in 2005 that uses a map-based display to identify position of friends and to, push special offers to the user. Closely integrated with Facebook and other services.	$17 million[3]
Google Latitude	Google's location service, integrated with Google Maps, that allows individuals to locate each other's cell phones as well as accessing other information available on Google maps.	N/A – Self-funded by Google
Mologogo	One of the earliest free map-based services for cell phones, founded in 2005. Allows individuals to link with each other. Appears to be declining in popularity.	Undisclosed
Foursquare	Location-based service designed to allow users to communicate with each other and to submit reviews of restaurants, merchants, etc. Closely tied to Facebook. Does not use a map-based interface.	$70 million[4]
Brightkite	Location-based service that supports group text messaging. Recently chose to de-emphasize location based aspect to emphasize the text messaging features. Does not use a map-based interface.	$9 million[5]
Gowalla	Location-based city and attraction guides. Does not use a map-based interface.	$8.4 million[6]
SCVNGR	Location-based scavenger hunt game with a map-based interface, designed to encourage users to engage in specific activities. Funded by Google.	$19 million[7]

Source: Case writer

[3] Source: http://venturebeat.com/2011/06/22/loopt-lets-local-business-customers- request-daily-deals/
[4] Source: http://techcrunch.com/2011/06/24/foursquare-closes-50m-at-a-600m-valuation/
[5] Source: http://techcrunch.com/2009/04/07/mobile-socializing-limbo-merges-with-brightkite-and-announces-9-million-funding-round/
[6] Source: http://www.statesman.com/business/content/business/stories/technology/2009/12/10/1210gowalla.html
[7] Source: http://techcrunch.com/2009/12/24/scvngr-google/

Exhibit 8: Summary table adapted from Eureka marketing plan

	Internal Marketing Plan
Category	**Strategy**
Positioning Statement	The Pathfinders Eureka Software is an application that has revolutionized the world of mobile data collection. Combining cutting-edge GPS, tracking, and reporting technologies, the Pathfinders Eureka Software is designed from the ground up to improve efficiency, accountability, and communication between all levels of business.
Offering to Customers	Free Software, Subscription Software, Customized Industry Specific Software, Support, Ease of Use, Form Templates, Exporting/Reporting, Profile, Tags, Matching Logic, Tracking, Accountability, Photo, Time/Date Stamp, Form Creation, Create Notes, Unlimited Users, Administrative Abilities, Permission Sets, PRIVACY and PRIVACY controls, Historical Data, Mass Notifications, Records
Target Market	General Population, NGO, Real Estate, Heating and Cooling, Church, Hotel, Security, Gov't, Emergency Management, Public Works, Contractors, Auto, Media, Warehousing, Insurance, Oil and Chemicals, Tobacco, Polling/Survey Co., Marketing Firms, Door to Door Salespeople
Marketing Research	Researching Industry Specific Forms (Flavors), Competition Pricing Model, Possible Ad revenue for Free Version, Specific Cell Phone Market Share (current and projected), Tracking Apps, Focus Groups, User testing, Communications and Software annual budgets based on Industry (percentage), Workforce management Software Viability analysis, Distribution Costs, Hosting Costs, Operating Costs based on projections, Conceptual Surveys to Business Owners, "Day in the Life"

Promotion Strategy	Using "Flavors" of Application to draw "industry specific" clientele and showcase Eureka capabilities. Paid banner ads, review sites, trade magazines, and trade shows. We will also pursue free avenues such as news articles about our compelling team and development (Associated Press, USA Today, etc.), Application Review Websites, Business Journals, Trade Magazines, Technology Magazines, Blogosphere, etc. Viral campaigning, Constant Contact, Discounted Rates for advanced subscription payment, App Markets, Social Networks, Reddit, Coming Soon Email Flyers, It's Here Flyer, Word of mouth
Sales Strategy	Cold calling target market in order to ascertain which yield the best results (reception). Reward Achievement and Incentivize Sales Force by offering 10% Commission. Utilize Cell Phone Sales People, Capitalize on existing relationships and contacts.
Service Strategy	One dedicated Salesperson and one IT Person for each client. Business Hours Support, Subscribers move to top of Support Queue, Feedback Forum on Website, Training Videos and Help Menus, FAQ, Service Blocks
Distribution	Method for downloading through Eureka website, Android Market, iPhone Market, Able to push to "dumb" phones
Pricing Strategy	Models based on organizational need. 1) Free with Advertisements 2) 501c3 Free Application 3) Subscriber 4) Customized Organization 6)Micro-Transactions 7) Set-Up Fee
Future Iteration Offerings	Mass Emails, Calendar, Org to Org Communication, Additional Templates

Source: Disaster Solutions

Exhibit 9: Extracts from programming assignment

DATE PREPARED: 09/06/2011 **PREPARED BY:** John Simion
OPERATIONAL PERIOD: 09/06/11 – 09/20/11

GENERAL OBJECTIVES: **Create PHONEGAP files for Front-End Devs; Create AFFILIATIONS in the Back-End; Create FORM SUBMISSIONS in front-end**

Notes:

We are getting close to Iteration 1 being complete. Major goals/obstacles:

1) Users do not have distinct permission-sets / friends / affiliations from one-another
2) Forms may be created but not submitted / edited
3) Events and Tasks are still not completed
4) The files required for PHONEGAP must be provided
5) No Design work has started for the Android / iPhone App

Administrative:
- Connect Tableau to PostgreSQL database
- Create sample Jasper Reports // Template ?
- Begin design of smartphone versions of the App
- Plan user flow for Search/Add Friends & Affiliations
- Plan user flow for creating a new Affiliation (in software??)
- Plan user flow for managing Affiliation users
- Plan user flow for managing Personal Affiliation (friends & groups)

Programming:
-***Sencha Touch-*** Implement Form Editor into App
-Sencha Touch- Implement Form Submissions into App
-***Sencha Touch-*** Finalize Groups (Org Screen)
-Sencha Touch- Finalize Events (Org Screen)
-***Ruby on Rails-*** Agilis→PTF Application Integration – FINISH JSON integration & Parsing
-***Ruby on Rails-*** Implement Groups/Teams into DB
-***Ruby on Rails-*** Implement Events into DB
-***Ruby on Rails-*** Implement Profile Surveys into DB
-***Ruby on Rails-*** Finish Events Back-End
-***Ruby on Rails-*** Finish Work Orders Back-End
- ***Java (Featurephones) –*** Code Featurephone Application / Meet with DS
- ***Website (Store / Software):*** Begin work on planning & designing this (think upsell / service charges)
- ***Website (User Accounts):*** Begin work on creating shared user accounts between Website, Store, and Application

INDIVIDUAL OBJECTIVES: *[Case writer note: Names Removed]*

JOHN SIMION- Design the first version of Android App. Create sample Jasper Report; Design Affiliation Flow

DEVELOPER 1- Implement Affiliations & Individual User Accounts

D**EVELOPER 2-** Integrate Form Creator into Application – Create Interpreted "Forms" for view & submission via the Map.

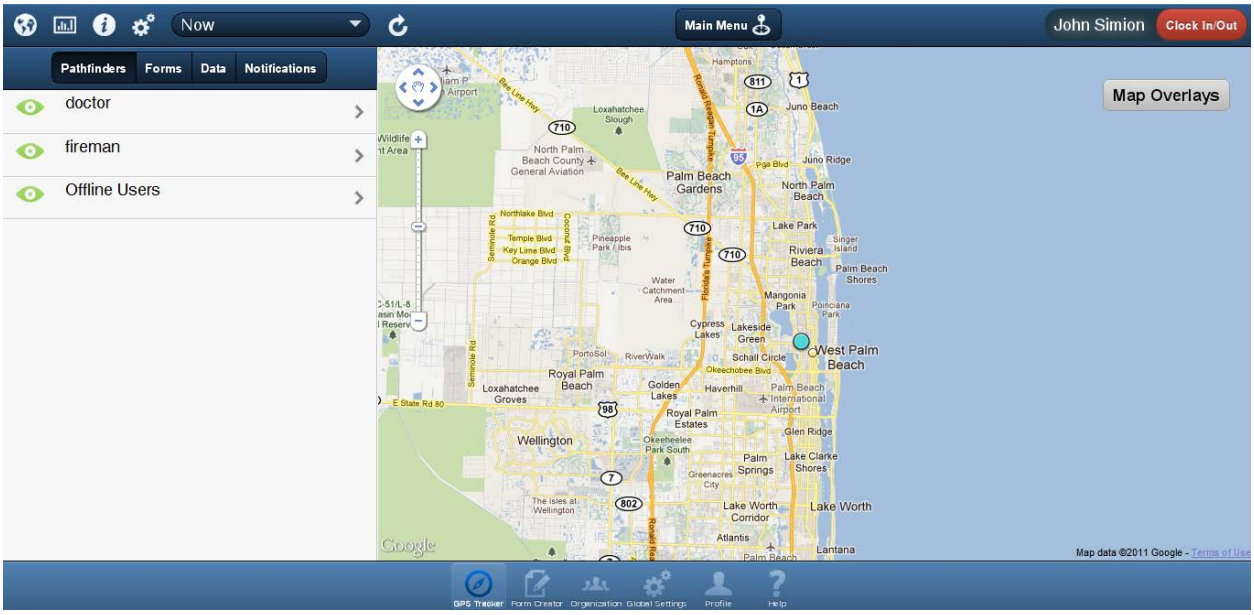
DEVELOPER 3- Finish Organization Screen Implementations – Implement Changes to Map & Fix Github Bugs

DEVELOPER 4- Design of Store

DEVELOPER 5- Begin implementing ruby user-registration / login into the website

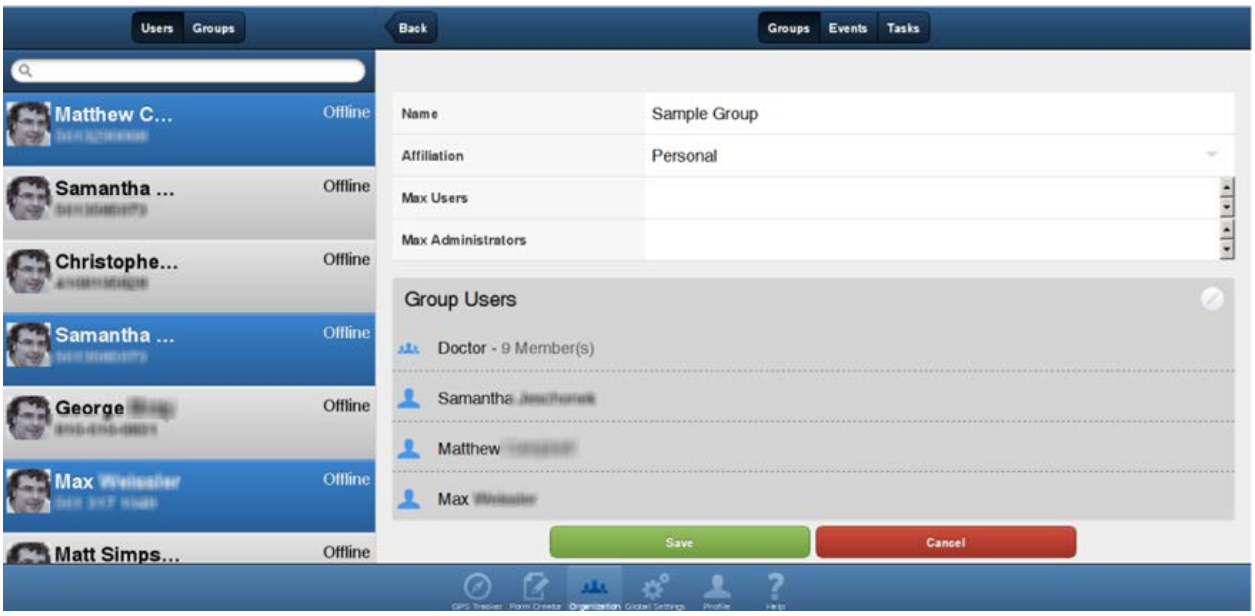
DEVELOPER 6- Finish module 1; begin module 2. Meet with DS about the re-write to make the app work disconnected

Exhibit 10: Sample screen shots from user interface

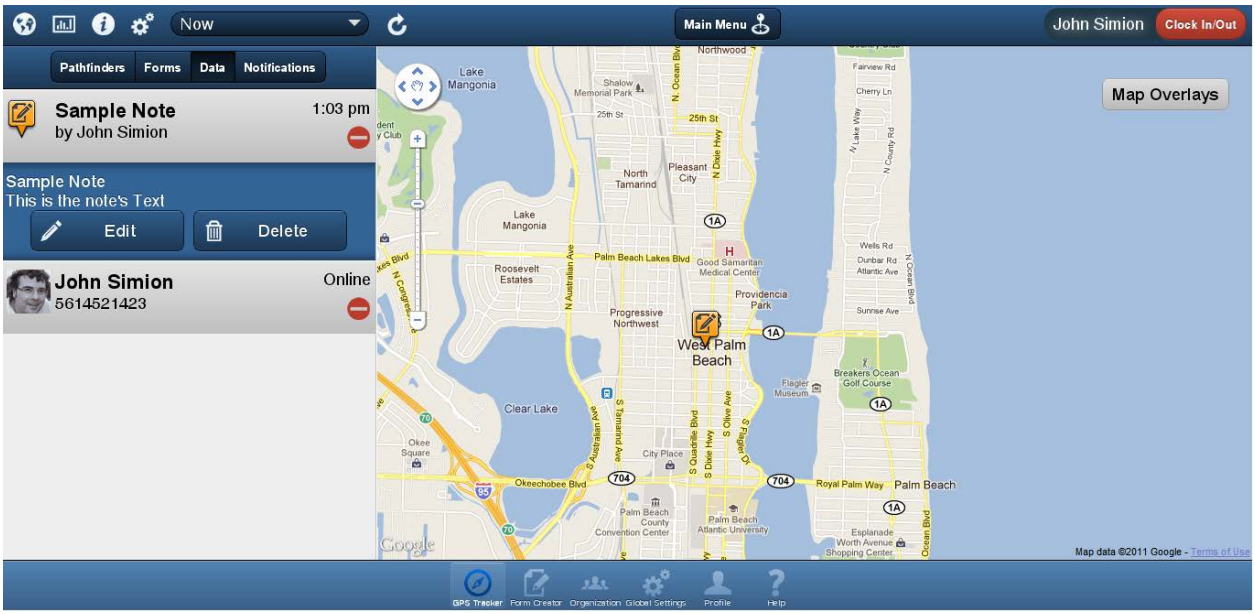


1. Main screen

2. Filling out a form



3. Defining a group (names and phone numbers blurred out by the case writer)



4. Attaching a note to a map