



## Calhoun: The NPS Institutional Archive

---

Faculty and Researcher Publications

Faculty and Researcher Publications

---

2012-06

# SPARCCS Smartphone-assisted readiness, command and control system

Crewes, Monique

---

17th International Command and Control Research and Technology Symposium (ICCRTS),  
Fairfax, VA, June 2012.. Includes slide presentation.



Calhoun is a project of the Dudley Knox Library at NPS, furthering the precepts and goals of open government and government transparency. All information contained herein has been approved for release by the NPS Public Affairs Officer.

**Dudley Knox Library / Naval Postgraduate School**  
**411 Dyer Road / 1 University Circle**  
**Monterey, California USA 93943**

<http://www.nps.edu/library>

**17th ICCRTS**  
**“Operationalizing C2 Agility”**

**SPARCCS – Smartphone-Assisted Readiness, Command and Control System**

Topic 7: Architectures, Technologies, and Tools  
Topic 9: Military and Civil-Military Operations  
Topic 4: Collaboration, Shared Awareness, and Decision Making

*LT Monique Crewes (USN) (STUDENT), LT Michael Asche (USN) (STUDENT),  
Gurminder Singh, John H. Gibson*

**Point of Contact**

Gurminder Singh (gsingh@nps.edu)  
Department of Computer Science  
Naval Postgraduate School  
Monterey, CA 93943

831-656-3041

**Abstract**

Maintaining an accurate common operating picture (COP) is a strategic requirement for efficient and successful missions in both disaster response and battlefield scenarios. Past practices include utilizing cellular, radio, and computer based communication methods and updating individual maps. A drawback of these practices has been interoperability of these devices as well as inaccurate reporting and documentation among different entities of the effort.

Recent advances in technology have led to the utilization of collaborative maps for maintaining a COP amongst command centers. Despite the advantages this technique offers, it does not address the difficulties surrounding receiving reports from field entities as well as ensuring these entities also have good situational awareness. We are developing SPARCCS (Smartphone Assisted Readiness, Command, and Control System) to address these issues. We use smartphones in conjunction with cloud computing to extend the benefits of collaborative maps to mobile users while simultaneously ensuring that the command centers receive accurate, up-to-date reports from the field.

**Keywords:** handheld devices, smartphones, hastily-formed networks, first responders, humanitarian assistance, disaster response (HA/DR), distributed response.

**1. Introduction**

Maintaining situational awareness in HA/DR (Humanitarian Assistance/Disaster Response) and military missions is critical to decision makers as it allows them to understand the current operating environment and make critical, real-time decisions based on what is occurring at the present time and what could occur in the future. In addition it gives them the ability to get immediate feedback on past decisions enabling them to make better choices in the future.

Situational awareness (SA) is enhanced through the use of a common operating picture (COP). Usually when a disaster, emergency, or military response is needed, multiple agencies, often times both local and national, must get involved and cooperate. In order for the cooperating agencies to work together effectively they need a shared situational awareness, or what is known as a COP. With this COP it becomes easier for agencies to communicate and resolve the issues they are facing more efficiently and effectively, saving both lives and resources in the process.

However, producing an accurate COP can be very difficult. This is because with the introduction of multiple agencies into a crisis situation, interoperability becomes a significant issue preventing interagency cooperation. Often each agency has developed its own respective methodology and corresponding systems for achieving a COP. These individual systems and databases are often unable to communicate with each other. As a consequence, information dissemination across agencies becomes almost impossible in real time, leaving decision makers in a precarious position of having to make important decisions with stale and possibly inaccurate information.

In order to address these problems, we are developing SPARCCS (Smartphone Assisted Readiness, Command, and Control System). SPARCCS uses smartphones in conjunction with cloud computing to extend the benefits of collaborative maps to mobile users while simultaneously ensuring that the command centers receive accurate, up-to-date reports from the field.

In addition to the need for interoperability and real-time information dissemination, there are a number of important first responder requirements that need to be met (Singh and Ableiter, 2008). SPARCCS pays attention to these requirements as described below:

1. **Quick Set-Up.** A key requirement of first responders, especially immediately after a disaster has struck, is to get going with their mission at the fastest speed possible. This means little time to set-up. SPARCCS is designed to work with COTS smartphones which can be easily provisioned with the mission information. The process of creating missions to address has been simplified in SPARCCS so that the administrators can create missions and provision as many devices as are needed for the mission rapidly and easily. We plan to develop a process of automatically generating network configurations and the list of equipment needed to support a particular mission based on the mission requirements.
2. **Tight-loop, Frequent Communication.** An important task of first responders is to convey the ground reality to their co-workers and the control room. This needs to be done frequently and in real-time but without taking too much of their time and attention lest it start affecting their mission performance. To support this requirement, SPARCCS smartphones produce images, videos and text which is XML-tagged automatically on capture. As the content is produced, it can be automatically shared within the team, if needed through the team lead's system. A sync between the team lead's system and the cloud server makes this information available to remote observers and planners.
3. **Light-Weight Equipment.** Due to the nature of their work, first responders' equipment needs to be as light as possible. SPARCCS uses state-of-the-art smartphones which are light-weight and small but still provide the compute, networking, storage and content capture power that is so critical to the mission success.
4. **Scale-up as Team/Requirements Grow.** SPARCCS can easily and rapidly be configured to start supporting a mission as the need arises. As the mission evolves, additional gear can be added to the SPARCCS configuration to support the expanded team.
5. **Battery.** Due to their short charge life and weight, batteries that operate the smartphones are an important issue. Often first responders have to carry spare batteries, which increase the weight they have to carry. We plan to address this problem by implementing smart energy management techniques in SPARCCS. These techniques will look at the equipment usage pattern of the first

responders and make decisions in real-time on the most effective strategy for minimization of energy use.

In the following sections we describe our system architecture and the various important components of it. We start off with a detailed background section.

## **2. Background - Evolution of Command and Control Systems**

Ten years ago a mobile-enabled COP wouldn't have been possible. Five years ago, it was possible but not practical. The state-of-the-art smartphones have now advanced to a stage where it is possible, practical and economical to develop a mobile-enabled COP system. This section covers the evolution of Command and Control systems that were the basis for the foundation of our research.

### ***A. Worldwide Military Command and Control System***

Following the events of the Cuban Missile Crisis, the government saw a need for a system that would allow authorities to exercise command and control over U.S. forces dispersed throughout the world. The resulting system was the Worldwide Military Command and Control System (WMCCS). Its primary mission was to provide

“a means by which the President and the Secretary of Defense can: receive warning and intelligence upon which accurate and timely decisions can be made; apply the resources of the Military Departments; and assign military missions and provide direction to the Unified and Specified Commands” (DoD, 1971).

The program required that all Department of Defense systems be configured to support the umbrella system. The final system consisted of computers, software, and communications lines forming an international network. The use of the system, however, experienced several failures. One of the most notable occurred in June 1967 during the hostilities between Israel and Egypt. The Joint Chiefs of Staff utilized WMCCS to order the USS Liberty to move further away from Egyptian and Israeli coastline where it was performing reconnaissance. The five high-priority messages that were sent over the system didn't arrive for thirteen hours. As a result, 34 American sailors were killed by an Israeli attack on the ship.

Most of the problems WMCCS experienced were blamed on the system's attempt to force existing technology to meet its specifications. Additionally, a 1979 report stated that the

WWMCCS ADP program was not responsive to national or local level requirements, was not reliable, lacked economical and effective growth potential, could not transfer data and information efficiently, made it extremely difficult and costly to exploit ADP technology, impaired each command's operational backup capability, and encouraged independent and decentralized software development efforts. (World Wide Military, 1979)

The report demanded either a major overhaul of the system or a complete replacement. In the early 1980s, WWMCCS was modernized and, despite its track record, demonstrated excellent performance in Desert Storm. Shortly following the conflict the technology the system was founded on was deemed too outdated and the decision was made to replace the system rather than try to modernize.

### ***B. Global Command and Control System***

The follow on system to WWMCCS was the Global Command and Control System (GCCS). Upon the release of GCCS in 1996 the Department of Defense boasted the introduction of modern day client-server computer architecture to replace the mainframe computers WWMCCS had been using since the 1970s. They also stated that commanders could now effectively

coordinate widely dispersed units, receive accurate feedback, and execute more demanding, higher precision requirements in fast moving operations. (Global Command, 1996)

In addition, GCCS was built with the humanitarian as well as the military missions in mind. GCCS has evolved over the years and several versions now exist specialized for Army, Air, Naval and Joint forces. Parallel capabilities are operated on the Non-classified Internet Protocol Routing Network, the Secret Internet Protocol Routing Network, and the Joint Worldwide Intelligence Communications System enabling collaboration between U.S. forces and our NATO allies.

Looking at GCCS today, it does not seem like a very impressive system. The maps are of poor quality and, in our experience, the system often fails to update for several days. Despite its contribution to the U.S. Military, it is once again time to upgrade to new technology.

### ***C. Blue Force Tracking***

Blue Force Tracking (BFT) is a crucial element to situational awareness. It enables leaders to track where their forces are deployed. Furthermore, it enables field entities to know their current location and the location of friendly entities in their vicinity. Historically this has been done in the field with a compass and a map. The entities' location would then be relayed back to command centers where they would be tracked on a separate map. Unfortunately this system was prone to error. Not only was it difficult to determine one's exact location using only a compass and map, but there were often reporting errors as well as mistakes made when mapping reported locations in the command centers.

The advent of the Global Positioning System (GPS) removed a great deal of error from the equation. GPS enabled field entities to constantly know their exact location. GPS, however, did not remove the possibility of error in reporting or in command center mapping. In addition, even with GPS, command centers would only know the last reported location of field entities.

Wireless data transfer removed the remaining possibilities for error. The combination of GPS and wireless data transfer enabled units to retrieve their location and autonomously pass that location to command centers. Not only did this eliminate reporting and mapping errors, but also provided near real time tracking of entities.

Force XXI Battle Command Brigade and Below was one of the first systems to take advantage of these capabilities. The system, which initially surfaced in the late 1990s, was comprised of application software connected to GPS satellite Receivers. It utilized tactical internet to exchange positioning data between field units and command cells. (Ebbutt 2008)

Current technology has nullified the requirement to carry a device solely for BFT needs. Most modern smartphones are enabled with both GPS and wireless data transfer capabilities. These devices are smaller, easier to use, and provide more functionality

### ***D. Next Generation Incident Command System***

While a number of systems have utilized collaborative maps, one that has been getting attention lately is the Next Generation Incident Command System (NICS). Formally the Lincoln Distributed Disaster Response System, NICS is being developed by the Massachusetts Institute of Technology's Lincoln Laboratories in conjunction with California Department of Forestry and Fire Protection (CAL FIRE) and the Department of Homeland Security. NICS prides itself on being an integrated, sensing, command and control system. The system enables collaborative disaster response and interoperability to improve situational awareness. While the system is still in development, CAL FIRE began employing it in Southern California for the 2010 fire season and successfully used it in over 85% of Riverside and San Diego wildfires that season (Lincoln Laboratory, 2011).

NICS is a Google Map based system, which allows collaboration amongst users. In addition to user input, NICS receives sensor input from a number of deployed entities. GPS devices report location data for vehicles and personnel for display in NICS. Airborne mounted sensors relay Real-time video, images and weather to the system. The system can currently be accessed by mobile devices that employ the FireFox web-browser and is primarily accessed in the field by laptops, and tough-books with WiFi or SATCOM capability. Though it can be accessed from a smartphone with FireFox, the system was not formatted for a small screen size,

rendering it relatively impractical for these devices. Lincoln Laboratories has expressed interest in developing a mobile application, but as the system itself is still in development, the mobile application is not a current priority (Next Generation, 2011).

NICS provides a good example of a PC, server-based COP. The amount of investment received by NICS, coupled with the fact that it has been utilized in both testing and actual disaster relief scenarios, suggests that it is a good case study. The NICS functionality provides a solid benchmark of necessary functionality for a successful COP tool.

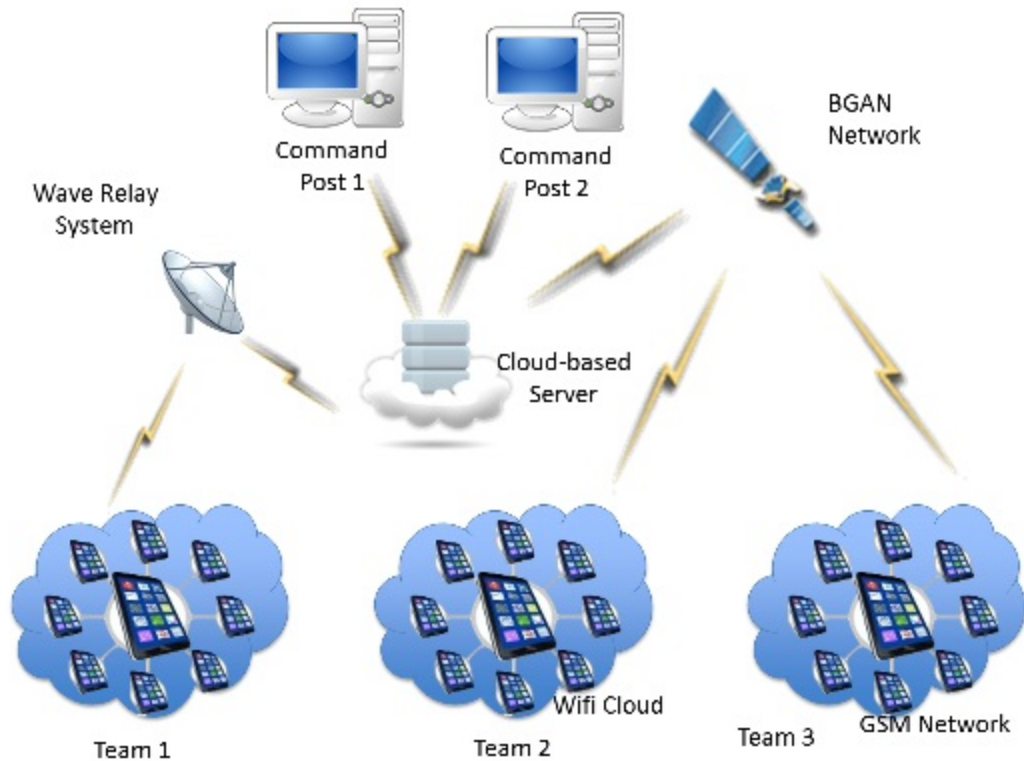
#### ***E. Advanced Ground Information Systems LifeRing***

Advanced Ground Information Systems (AGIS) currently offers a first responder mobile application for peer-to-peer communications in the field, called LifeRing (AGIS 2010). The application offers many useful capabilities for information distribution, communication, location and GeoIntel to provide a COP amongst users. LifeRing was originally only available on Windows Mobile-based devices but has recently been extended to include most mobile operating systems. LifeRing's key functionality includes the ability to exchange images and video, Blue Force Tracker, and collaboration between multiple devices utilizing different operating systems.

While LifeRing is a useful system for mobile devices, it has some issues that SPARCCS addresses. The first issue is the maps on the mobile devices. While the PC version of LifeRing can access and utilize Google Maps, the mobile version cannot. The maps on the mobile devices are preloaded which use critically limited storage space. In addition, the maps on the mobile devices are poor quality and do not have satellite capability. SPARCCS uses Google Maps on both its mobile and headquarters version of the application. Another issue is the way devices are connected to each other. LifeRing is server-based while SPARCCS takes advantage of distributed computing combined with cloud computing for its server and database needs. By doing this SPARCCS takes advantage of all the capabilities cloud computing has to offer, especially that of disbursed data locations. This insures that an incident or disaster at a server location is unable to bring down the system. In addition, SPARCCS has the ability to turn a mobile device into a server for a unit or team. This should not only decrease bandwidth usage but also insure that should the server fail, the unit or team will be able to continue using the system.

### **3. SPARCCS Architecture**

SPARCCS is implemented in a highly distributed fashion where smartphones act as the primary device used by the first-responders and a cloud-based database system aggregates information provided by first-responders for the command posts. Figure 1 illustrates a high-level architecture of SPARCCS.



**Figure 1. SPARCCS Architecture**

A mission in SPARCCS is typically spread across multiple small teams of first-responders. Each first-responder is equipped with a smartphone which is provisioned with SPARCCS mobile application. A first-responder uses this device to capture and share his/her situational information as a content feed (made up of pictures, video, location, messages etc) with the team lead's device, which can be another smartphone or a heavier duty device depending on the need of the mission. This content feed is maintained in a database system running in the team lead's device. The team lead can share this information with the other first-responders who may be a part of his team as well as forward the aggregated feed from all team members to the SPARCCS server located in the cloud.

The SPARCCS server runs in the cloud and receives feeds from multiple teams participating in the mission. All of the content thus received is maintained in the SPARCCS server and made available to local and remote command posts. A command post can view all participants and teams on the map as well as aggregated and individual content from first-responders on the ground.

The number of teams and the number of members of a team is variable and can be dynamically configured based on the mission needs. It is possible for a single team as well as multiple, geographical dispersed teams to handle a mission.

The network connectivity among team members as well as the team lead and the SPARCCS servers can be realized in several ways. For example, it is possible to have a local WiFi cloud to support connectivity among the team members, and from the team leads server to the main SPARCCS server one may use a BGAN or a wave relay. There is considerable flexibility in the SPARCCS architecture enabling the team to work with the most convenient technologies available for the mission.

SPARCCS implements several key features making it a robust, practical system. These features are as follows:

- Login to prevent unauthorized access to the system.
- Create, join, edit, and view missions.
- Create, edit, view, and delete points of interest.

- Capture, edit, view, and delete images.
- View all missions, points of interest, and images on a map.
- View all mission, points of interest, and responder information in a list format.
- Retrieve smartphone GPS location for the placement of missions, points of interest, and images.
- Store all relevant information in the android's local SQLite database, or the cloud database.
- Syncing between the android and cloud databases.

The implementation of these functionalities is discussed in depth in the “SPARCCS Implementation” section below.

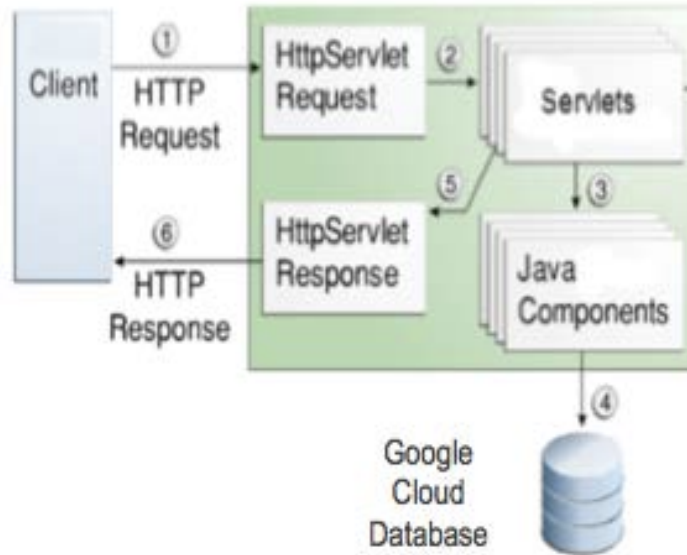
#### **4. SPARCCS Implementation**

SPARCCS implementation consists of over fifty classes and over 75,000 lines of code. Although it is a large program, it is a fairly simple architecture in implementation. The system centers around four main classes: Responders, Missions, Points of Interest and Images/Videos.

- The Responder class contains information about individuals reacting to operations around the world, as well as those operators working within the central command. The Responder class consists of the following notable data members: Id, First Name, Last Name, Middle Initial, Login, Password, Type (Military, Fire, Medical, Humanitarian Assistance or Law Enforcement), Unit, Phone Number, Email Address, IP Address, Associated Mission, Latitude and Longitude.
- The Mission class contains information about an operation responders are reacting to. The Mission class consists of the following notable data members: Id, Name, Description, Start Date, End Date, Latitude, Longitude, Mission Leader, Mission Creator, and Miscellaneous Information.
- The Point Of Interest class contains information about a particular place, person or item of significance that corresponds with an associated mission. The Point Of Interest class consists of the following notable data members: Id, Name, Description, Time, Creator, Correlating Mission, Location Notes, Latitude and Longitude.
- The Image/Video class contains a picture/video and corresponding information for that picture/video. The Image/Video class consists of the following notable data members: Id, Name, Description, Creator, Time, Correlating Mission, Correlating Point of Interest, Location Notes, Latitude and Longitude. Our current implementation focuses on images only.

These classes can be viewed individually in a list format or can be plotted on a Google Map. Instances of these classes are saved on a Google Cloud Database that is accessed from both the cloud application and the Android mobile application. Using CRUD (Create, Read, Update and Delete) operations through HTTP Servlets on the Android client and Remote Procedure Calls on the web client, instances of these classes can be manipulated.





**Figure 2. The Android System Architecture**

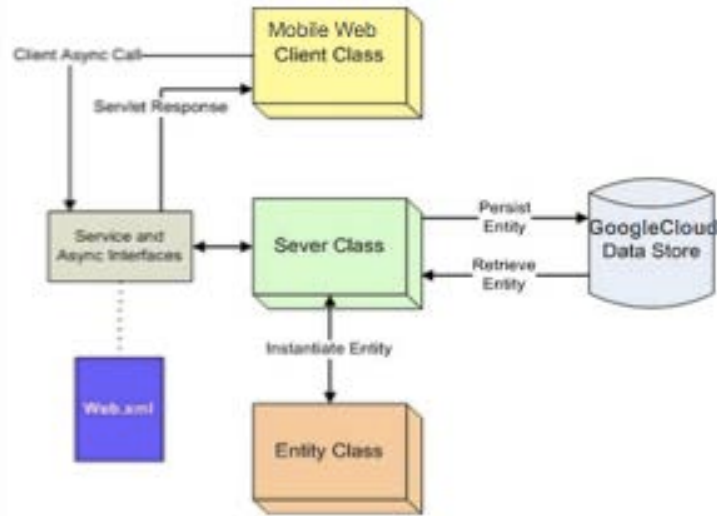
Figure 2 represents the architecture of our Android mobile client application running on smartphones. The client can request/send class information via an HTTP Request (1). The HTTP request interacts with the HTTP servlets using HttpServlet Requests (2). These servlets form Java components in our case Objectified Data objects (3) to perform CRUD operations with cloud database (4). The servlets then create the appropriate HTTP Servlet Response (5) back to the requesting client (6). The information is then displayed in the client.

Figure 3 shows the overall system architecture from the web client perspective. The only difference being the remote procedure calls instead of the HTTP posts and requests. The following sections further detail the workings of each client application as it interacts within the overall SPARCCS architecture.

#### **A. SPARCCS Cloud Application**

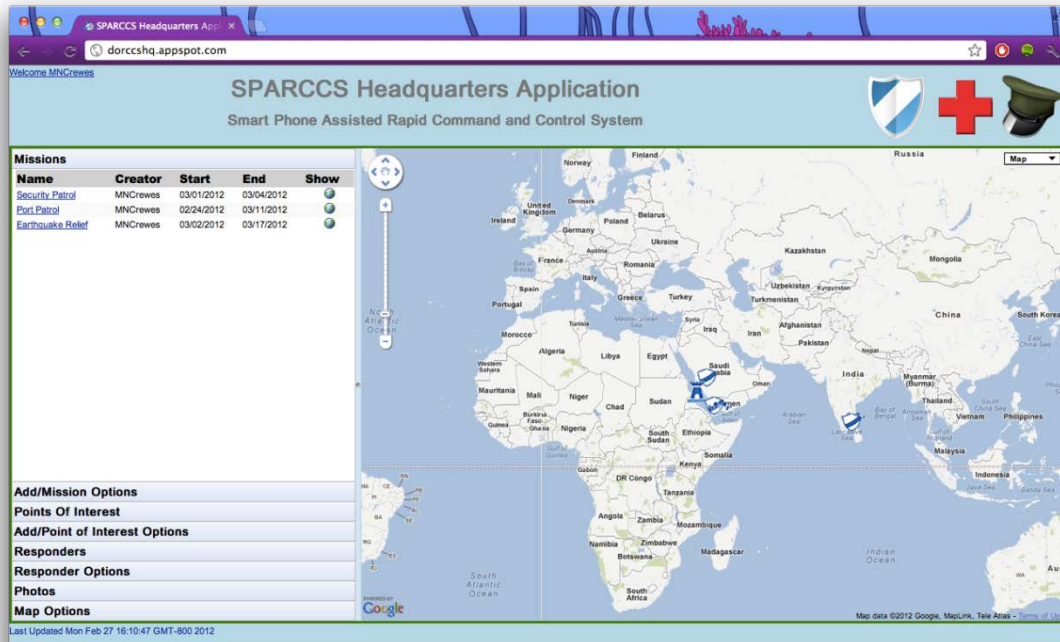
The cloud application begins with a login screen which enables existing users to login or new users to request user accounts. For new users, they must provide in all of their responder information and they will be given a user name consisting of their first initial, middle initial and last name. If there are matching names the login string is extended by a number based on when they registered.

Once the user has logged in, they are taken to the main interaction screen shown in Figure 4. On the large right hand panel of the screen is the user map. It is centered on the user's current location. Also displayed on the map are all the current missions, points of interest, responders, and images. On the left panel of the screen is a navigation panel that opens to a list of all of the missions. The navigation panel also contains other panels for further information that we will go into detail later in this section.



**Figure 3. Web Client System Architecture**

From the main screen the user can choose to interact with the map directly or interact with each of the navigation panes. Interacting with either the map or the navigation panes will update the opposite interaction tool so the user does not have to worry about inputting information twice. Next we will go through the key panels within the navigation pane and show how each pane interacts with the map and how the map interacts with the navigation panes.

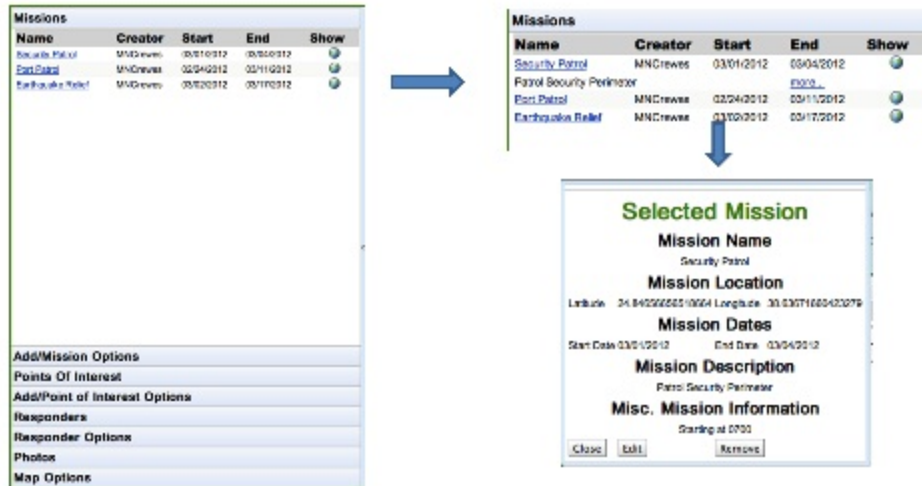


**Figure 4. SPARCCS Main Interaction Screen**

### *Missions Panel*

The Missions panel contains a list of all of the missions contained in the application. The initial information that can be seen is the name, creator, start date and end date. The creator of the mission is the responder's login to keep the names short for the display. However, if the creator's login is moused over then the full name of the mission creator will be displayed in a small pop-up panel. The mission name is an anchor. If this anchor is clicked then the mission's description will show up below the mission's information along with a second anchor that says "more..." If this anchor is clicked then all the mission information will be displayed along with an option to edit or remove the mission if the currently logged in user is the creator of the mission. This sequence is displayed in Figure 5. There is also a globe button associated with each mission in the list. If this button is pressed then the mission along with essential mission information will be displayed on the map. If the bubble on the map is expanded then all of the mission information is displayed.

There is an Add/Mission Options Panel which allows the user to manually add a mission to the list, provides mission list display options, show all missions on the map and shows mission statistics. From this panel the first option is a button for the user to add a mission. Clicking this button brings up a mission form for the user to input mission information displayed. The mission must have a name, valid location, and start date (the end date does not have to be known at the time of creating the mission).



**Figure 5. Missions Display and Flow**

***Points of Interest Panel***

The Points of interest panel is much like the Missions Panel. It displays a list of all the Points of Interests. The list contains an anchor for the Point of Interest name, the time when the point of interest was created, the Mission that the Point of Interest is associated with, the creator, and a button that will show the Point of Interest on the map. Since the creator name is actually the creator login for ease of identification the user name can be moused over to see the creator’s full name. Like the mission panel if the Point of Interest’s name anchor is clicked then the Point of Interest’s description is displayed in the list with a second anchor that says “more...” If this anchor is clicked then a panel is displayed with all the Point of Interest’s information. If the creator of the Point of Interest has clicked on the anchor then he/she has the option to edit or remove the Point of Interest. This flow is demonstrated in Figure 6.

Just like in the Missions panel, the globe button allows the user to see the Point of Interest on the map, as shown in Figure 7. If this button is pressed then the Point of Interest along with its essential information will be displayed on the map. If the bubble on the map is expanded then all of the Point of Interest’s information will be displayed. Within this expanded bubble links to see corresponding responders, missions, and pictures on the map at the same time. Once these associations are displayed on the map the user can click on them to get their corresponding information.

**Missions**  
Add/Mission Options

**Points Of Interest**

Name	Time	Mission	Creator	Show
<a href="#">Earthquake Relief Overflow</a>	00:09:32 02/28/2012	Earthquake Relief	MNCrews	
<a href="#">Weapons Cache Found</a>	00:11:20 02/28/2012	Port Patrol	MNCrews	
<a href="#">Hostiles Encountered</a>	01:12:27 02/28/2012	Security Patrol	MNCrews	

**Points Of Interest**

Name	Time	Mission	Creator	Show
<a href="#">Earthquake Relief Overflow</a>	00:09:32 02/28/2012	Earthquake Relief	MNCrews	
<a href="#">Weapons Cache Found</a>	00:11:20 02/28/2012	Port Patrol	MNCrews	
<a href="#">Hostiles Encountered</a>	01:12:27 02/28/2012	Security Patrol	MNCrews	

Tallest building on the street, with black flag on 4th story window. [more...](#)

**Selected Point of Interest**

**POI Name**  
Hostiles Encountered

**POI Date**  
Time Created 01:12:27 02/28/2012

**POI Location**  
Latitude 23.98625275300314 Longitude 39.5463984505201

**POI Location Notes**  
Hostiles found hiding in top floor of building

**POI Mission**  
Security Patrol

**POI Description**  
Tallest building on the street, with black flag on 4th story window.

Figure 6. Points of Interest Display and Flow

**POI Name:** Hostiles Encountered  
**POI Description:** Tallest building on the street, with black flag on 4th story window.  
**POI Time:** 01:12:27 02/28/2012  
**POI Creator:** MNCrews

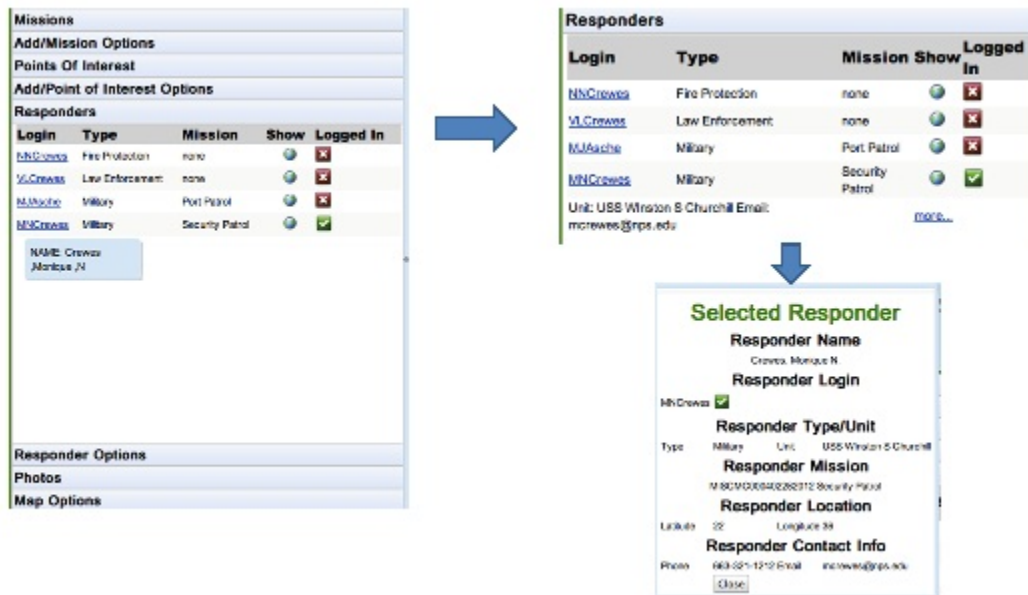
**POI Name:** Hostiles Encountered  
**POI ID:** POICM01122702282012  
**POI Description:** Tallest building on the street, with black flag on 4th story window.  
**POI Time Created:** 01:12:27 02/28/2012  
**POI Creator:** MNCrews Crews, Moushe N.  
**POI Mission:** MNCM000452282012 Security Patrol  
**POI Latitude:** 23.98625275300314  
**POI Longitude:** 39.5463984505201  
**POI Location Notes:** Hostiles found hiding in top floor of building  
**Associated Pictures:**  
**Show All Associated Pictures on Map:**  
**Show All Associations on Map:**

Figure 7. Point Of Interest and Expanded Point of Interest Information on Map

The Add/Point of Interest Options Panel allows the user to manually add a Point of Interest to the list, provides Point Of Interest list display options, show all Points Of Interests on the map, and shows Point Of Interest statistics. From this panel the first option is a button for the user to create a Point Of Interest.

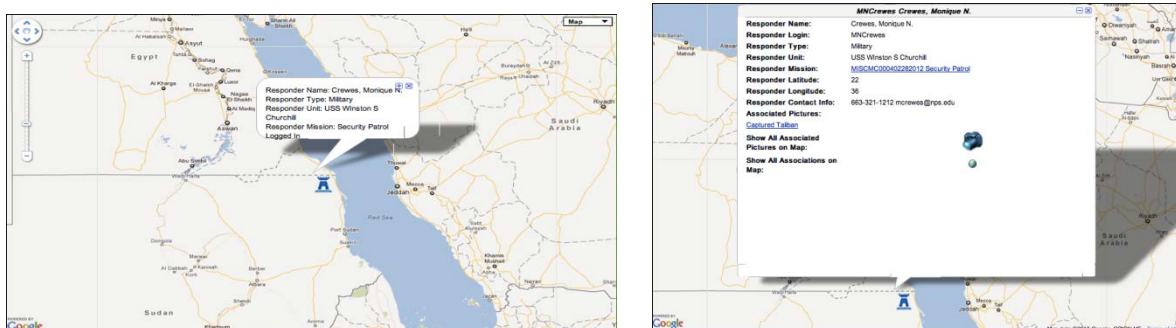
## Responders Panel

The Responders Panel contains a list of all the Responders in the SPARCCS system. The list contains the Responder's login anchor, type, unit, associated mission, and a graphic the responder is logged into the system. In this case since only the Responder's login is shown if the login is moused over then the user is able to see the Responder's full name. Like the Mission and Point Of Interest Panels, if the anchor is login anchor is clicked on then the Responder's unit and email is displayed along with an anchor that says "more..." If this anchor is clicked on then a panel containing all of the Responder's information is shown. This flow of information is illustrated in Figure 8.



**Figure 8. Responders Information Flow**

Likewise, there is a globe button associated with each Responder. If clicked, the Responder will display on the map with a bubble containing the Responder's essential information as shown in Figure 9. If the bubble on the map is expanded then all of the Responder's information will be displayed. Within this expanded bubble links to see corresponding missions and pictures on the map at the same time. Once these associations are displayed on the map the user can click on them to get their corresponding information.



**Figure 9. Responders on Map**

## Photos Panel

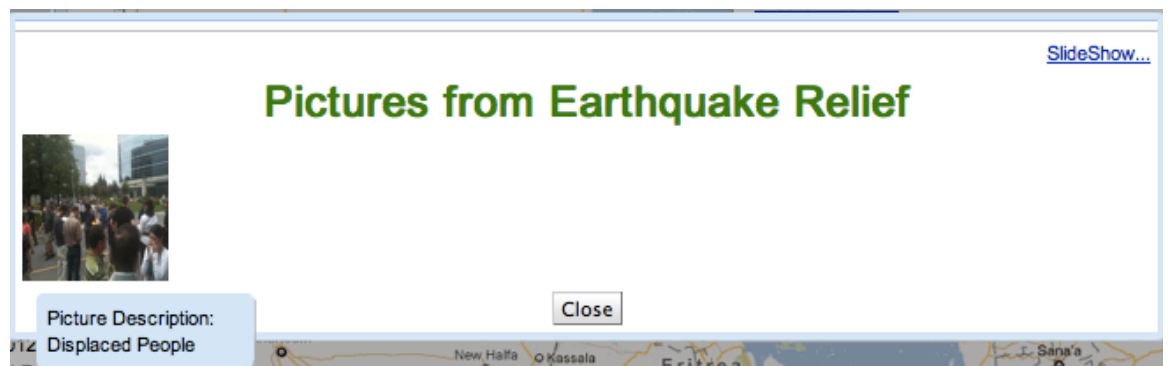
As shown in Figure 10, the Photos Panel consists of three main sections: the Missions section, the Point of Interest section, and the Responders section.

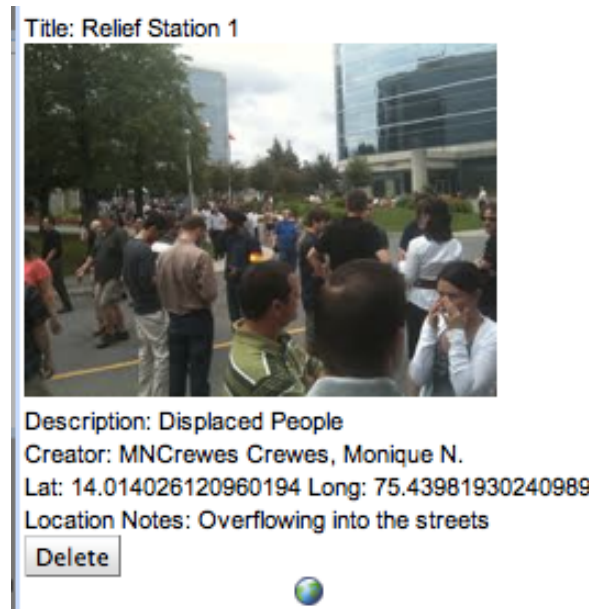
The screenshot shows a web interface for the SPARCCS Photos Panel. It is organized into several sections:

- Add/Mission Options**
- Points Of Interest**
- Add/Point of Interest Options**
- Responders**
- Responder Options**
- Photos**
- Missions**
  - Radio buttons:  Add Image to a Mission,  See Images From a Mission,  See All Mission Images
  - List box:
    - MISCMC000402282012 | Security Patrol
    - MISCMC000502282012 | Port Patrol
    - MISCMC000702282012 | Earthquake Relief
- Points of Interest (POI's)**
  - Radio buttons:  Add Image to a POI,  See Images From a POI,  See All POI Images
  - List box:
    - POICMC00093202282012 | Earthquake Relief Overflow | MISCMC000702282012 |
    - POICMC00112002282012 | Weapons Cache Found | MISCMC000502282012 | Po
    - POICMC01122702282012 | Hostiles Encounter | MISCMC000402282012 | Securi
- Responders**
  - Radio button:  See Images From a Responder
  - List box:
    - NNCrewes
    - VLCrewes
    - MJAsche
    - MNCrewes

Figure 10. SPARCCS Photos Panel

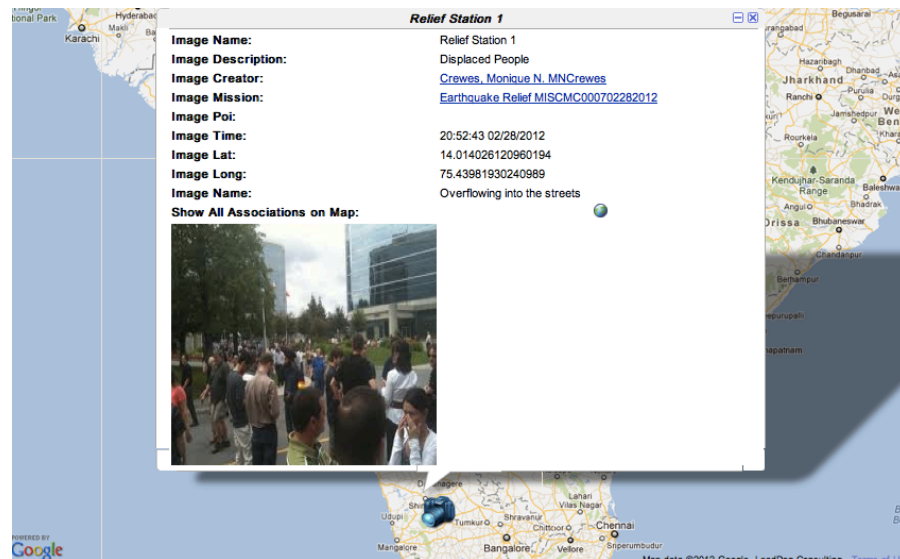
The Mission section contains three radio buttons and a list box. The radio buttons enable the user to add an Image to a selected Mission from the list box, see an Image from a selected Mission from the list box, or to see all Images from all Missions (see Figure 11).





**Figure 11. Individual Mission Picture**

There is also a globe button in this panel that will allow the user to show the Image's location on the map with a bubble containing all the Photo's essential information. If this bubble is expanded the user will again be able to see all the Image's information with anchors to see links to the Image's associations on the map as shown in Figure 12.



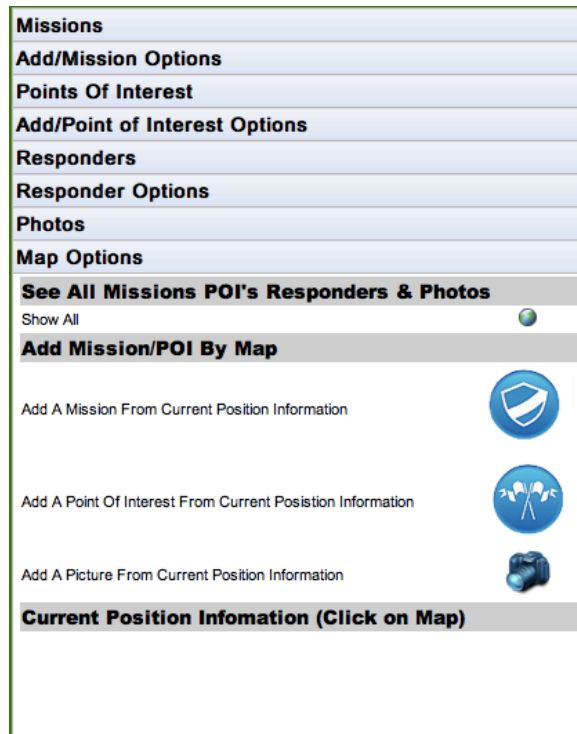
**Figure 12. Expanded Image Information**

### **Map Options Panel**

The Map Options Panel in Figure 13 is the final panel available for selection. This panel contains options for operating within the map portion of the application. The panel is split into three sections. The first Section allows the user to see all Missions, Points of Interest, Responders and Images on the map at one time. In the same manner as when the application opens. The second portion of the panel contains three buttons that allow the user to create a Mission, Point Of Interest or Image from a position the user has selected on the map. If any one of these buttons is clicked the corresponding creation form, as displayed



earlier, will be displayed with the latitude and longitude boxes already filled in with the coordinates from the position the user clicked on the map. The third portion of the screen displays a detailed geolocation as well as the latitude and longitude from a position selected on the map. Also if the user would like to clear the map of all overlays the user can click the clear button at the bottom portion of this section.



**Figure 13. Map Options Panel**

### ***B. SPARCCS Android Mobile Client***

The overall structure of the mobile client is explained in the beginning of section 4 (Figure 2). The mobile client stores all of the information the user has captured in a local database until it is uploaded to the server. This insures that the information is not lost even when the network connectivity is not available. All information is stored locally on the phone's native database. When the application is run for the first time on any device, the system creates a database with 15 tables. The tables are as follows:

- responder – stores the list of all responders in the system.
- pointOfInterest – Stores the list of all points of interest in the system.
- sendPointOfInterest – Stores the list of locally created points of interest that have yet to be uploaded to the Server.
- editPointOfInterest – Stores the list of locally edited points of interest that have yet to be modified in the server.
- mission – Stores the list of all missions in the system.
- sendMission – Stores the list of locally created missions that have yet to be uploaded to the server.
- editMission – Stores the list of locally edited missions that have yet to be modified in the server.
- Image – Stores the list of all images in the system.
- sendImage – Stores list of all locally created images that are not yet uploaded to the server.
- editImage – Stores list of all locally edited images that have yet to be modified in the server.
- myInfo – Same setup as the responder table, but only holds information for the current user.

- deletePointOfInterest – Stores the point of interest ids for locally deleted points of interest that have yet to be deleted from the server.
- deleteImage – Stores the image ids and associated blob keys (URLs) of locally deleted images that have yet to be deleted from the server.
- viewMission – Stores a list of mission ids whose entities user would like displayed on map.
- lastLogin – Stores the username of the last user logged in so they don't have to reenter their username each time they log in.

### ***Syncing***

Upon successful login to the system, a syncing thread is created. The constructor of the thread does an initial pull from the server retrieving all users, missions, images, and points of interest and inserting them into the database. Once this is done, a run function is called that starts a loop that continues until the program is exited. The first operation of this loop is to update the responder information.

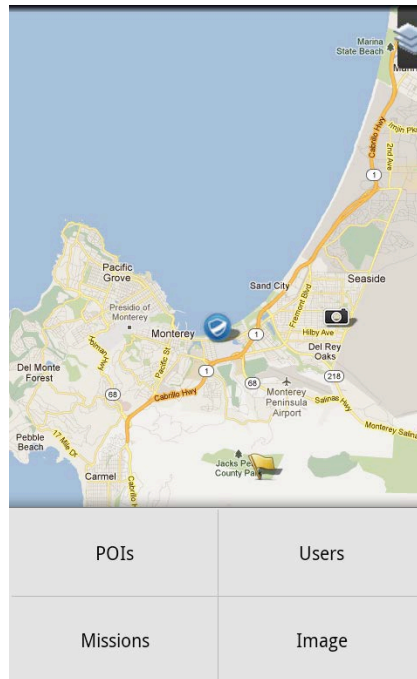
After updating the user information the loop pulls the list of missions to be sent from the database. Each time a mission is successfully uploaded to the server it is removed from the sendMission table of the database. If there are no new missions to be sent this section is skipped. This process is then repeated for points of interest to be sent, images to be sent, points of interest to be deleted, images to be deleted, points of interest to be edited, images to be edited, and missions to be edited.

Following the upload of all new information the system does another pull for all responders, missions, images, and points of interest. Their corresponding tables are cleared in the database and the new information is added. To prevent new entities from disappearing in the unlikely event that they were added between upload and download, the system checks the send tables and adds their information to the regular tables as well.

Once the sync is complete and the new information is entered into the database, the thread sends a message to the application, via a message handler. The message simply indicates that a sync has been completed, prompting the application to update the map overlays. Finally the thread enters a sleep period. For testing purposes this sleep has been set to thirty seconds. Through field test, an appropriate sleep period should be determined to maximize battery life while minimizing lag time.

### ***Android Client User Interface***

Like the cloud application, the android application enables an existing user to login or to request a new user account. Upon logging in, and receiving the user data back from the server, the application checks the user information to make sure the user is part of a mission and that the user's mission still exists. If so, the user is taken to the main screen as shown in Figure 14. Pressing any of these four buttons will open a sub menu.



**Figure 14. The main Android Client Screen**

### ***Points of Interest***

Points of interest are created by pressing the POIs menu button. The point of interest form has three fields: POI Title, POI Description, and Location Notes. On the bottom of the form, under the text “location,” there are three buttons: “Current,” “Choose,” and “Cancel.” Pressing “Cancel” discards the point of interest and returns to the main screen. Pressing “Current” retrieves the user’s location and returns to the main screen where the map is centered on a flag placed at the device’s current location. Pressing the “Choose” button also returns to the main screen, however, the next time the screen is tapped a flag will be dropped at the location of the tap.

### ***Images***

Images can be entered into the system in two ways: either by capturing an image with the device’s native camera, or by selecting a previously captured image from the photo gallery. This option is presented to users by pressing the menu button followed by the image menu item. When an image is captured it is saved as a .tmp file in the applications folder. The path to the image is saved in ImageClient class and is used to access the picture until it is uploaded to the server. Once the image is successfully uploaded it is assigned a URL for accessing the picture and the .tmp file is deleted.

Images can be viewed by either tapping on the image’s icon, or if associated with a point of interest, by opening the point of interest form and clicking on the image title under the label “Associated Images.” Viewing an image reopens the image form, shown in Figure 15, with additional fields for the image’s associated mission, point of interest, and the image creator. If the user viewing the image is the image creator they have the ability to edit the image description and location notes or to delete the image. Image’s are not stored locally on the device for memory purposes. Instead, each image is assigned a URL when it is uploaded to the server. When a user opens an image form for viewing an image the system downloads the image from the server then displays it on the form.

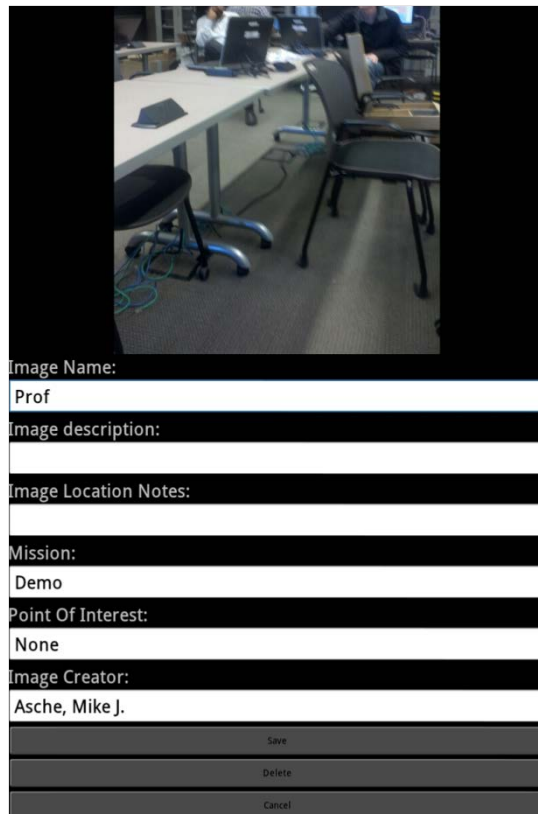


Image Name:	Prof
Image description:	
Image Location Notes:	
Mission:	Demo
Point Of Interest:	None
Image Creator:	Asche, Mike J.
	Save
	Delete
	Cancel

**Figure 15. View Image Form**

### *Users*

User information can be viewed by pressing the menu button, followed by the user menu item, followed by the list sub item then selecting a user. Once this is done, the user form is opened filled in with the user's information. The form has fields for first, middle, last, and user name, as well as unit, current mission, classification, email, and phone number. Clicking the done button at the bottom of the form returns the user to the list view.

## **5. Current Status and Limitations**

SPARCCS is an ongoing project at the Naval Postgraduate School. We have implemented the first version of the system as described in this paper. The system is currently limited to image capture. We are in the process of developing video capability for this system. We expect this to be completed soon.

We plan to test this system with a number of typical first-responders at Camp Roberts in the coming months. We are developing a sample scenario where we will exercise the system, and based on the feedback from the test, revise the system before it is released for general use.

A limitation of the current system is that the client application has been developed as a native application for Android smartphones. While this practice is pervasive, it suffers from high development costs especially when the applications needs to ported and maintained on multiple platforms. We can avoid this issue by developing the application in HTML5. HTML5 provides access to the low-level device features that are needed to develop SPARCCS. However, the specification of HTML5 is still evolving. As soon as the specification is frozen, we would develop SPARCCS in HTML5.

Another limitation of SPARCCS is its lack of attention to security. While we fully appreciate the need for security, we have not had the time and resources to develop this aspect of the system. We plan to do so in our future versions.

SPARCCS is a general system for HA/DR applications and it can be used in many HA/DR tasks. It would however be best to specialize it for a certain target domain. We are considering specializing SPARCCS for medical triage. This investigation has started but is in its early stages.

### Acknowledgements

This work has benefited from the participation of several other students at NPS including CAPT Drew Abel (ARMY), CAPT Nusret Deminzci and Donna Dulo.

### References:

- Advanced Ground Information Systems (2010). "AGIS LifeRing Operators Manual." Retrieved from [http://www.agisinc.com/AGIS\\_Operators\\_manual.pdf](http://www.agisinc.com/AGIS_Operators_manual.pdf)
- Comptroller General. (1979 December 14). The World Wide Military Command And Control System—Major Changes Needed In Its Automated Data Processing Management And Direction (LCD-80-22). Washington, DC: Author. Retrieved from <http://www.gao.gov/assets/130/128411.pdf>.
- Databases. (n.d.). Retrieved September 16, 2011, from Wikipedia: <http://en.wikipedia.org/wiki/Database>
- Fay Chang, J. D. (2006). *Bigtable: A Distributed Storage System for Structured Data*. Proc. 7<sup>th</sup> USENIX Symp on Operating Systems Design and Implementation, Nov 6-8, 2006, Seattle, WA
- Department of Defense News Release (1996). Global Command and Control System Adopted. Washington, DC: Author. Retrieved from <http://www.defense.gov/releases/release.aspx?releaseid=1049>
- Deputy Secretary of Defense (AT&L) (1971). World-Wide Military Command and Control System (DoD Directive 5100.30). Washington, DC: Author. Retrieved from [http://www.fas.org/spp/military/docops/defense/d5100\\_30.htm](http://www.fas.org/spp/military/docops/defense/d5100_30.htm)
- Duling, J. M. (2009). *THE COMPONENTS NECESSARY FOR SUCCESSFUL INFORMATION SHARING*. Masters thesis, Naval Postgraduate School, Monterey, CA.
- Ebbutt, G. (2008). Blue-force tracking evolves for the modern battlefield. *HIS Jane's: Defense & Security Intelligence and Analysis*. Retrieved from <http://www.janes.com/products/janes/defence-security-report.aspx?id=1065926194>
- Endsley, M. R. (1995b). Toward a theory of situation awareness in dynamic systems. *Human Factors*, 37, 32–64.
- Howard, P. (2011, May 17). *When is a Database Not So Relational*. Retrieved September 23, 2011, from [http://www.theregister.co.uk/2011/05/17/cloud\\_databases1/](http://www.theregister.co.uk/2011/05/17/cloud_databases1/)
- Copeland, J. (2008). *Emergency Response: Unity of Effort Through A Common Operational Picture*. (Master's thesis, U.S. Army War College).
- Lincoln Laboratory. (2011). *Next Generation Incident Command System*. Retrieved September 29, 2011, from [http://www.ll.mit.edu/publications/technotes/TechNote\\_NICS.pdf](http://www.ll.mit.edu/publications/technotes/TechNote_NICS.pdf)
- Meyer, D. (2011, May 19). *Android shoots past iPhone OS in market share*. Retrieved January 23, 2012, from ZDNet UK / News and Analysis / Mobile IT / Mobile Devices: <http://www.zdnet.co.uk/news/mobile-devices/2011/05/19/android-shoots-past-iphone-os-in-market-share-40092829/>
- Microsoft. (2011). *Cloud Computing*. Retrieved September 24, 2011, [http://www.microsoft.com/industry/government/guides/cloud\\_computing/2-benefits.aspx](http://www.microsoft.com/industry/government/guides/cloud_computing/2-benefits.aspx)
- Milian, M. (2012, February 03). *CNN*. Retrieved February 03, 2012, from U.S. government, military to get secure Android phones: <http://edition.cnn.com/2012/02/03/tech/mobile/government-android-phones/index.html>

- National Commission on Terrorist Attacks Upon the United States (2004). *The 9/11 Commission Report*. New York: W.W. Norton & Company. Retrieved from <http://www.gpoaccess.gov/911/>
- Singh, G. and Ableiter, D. (2008) TwiddleNet: Smartphones as personal content servers. Proc. Mobile Response: 2<sup>nd</sup> International Workshop on Mobile Information Technology for Emergency Response, Bonn, Germany, May 2008, pp: 130-137
- Tourtlotte, D. R. (2010). *X3D-EARTH: FULL GLOBE COVERAGE UTILIZING MULTIPLE DATASETS*. Masters thesis, Naval Postgraduate School, Monterey, CA
- Vidan, Andy and Gregory Hogan. (2010). "Integrated Sensing and Command and Control System for Disaster Response." *IEEE*, December 3, 2010. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=05655066>

## **Author Profiles**

### **LT Monique Crewes (USN) (STUDENT)**

LT. Crewes is a full-time M.Sc. (Computer Science) student at the Naval Postgraduate School, Monterey, CA. She expects to graduate in March 2011.

### **LT Michael Asche (USN) (STUDENT)**

LT. Asche is a full-time M.Sc. (Computer Science) student at the Naval Postgraduate School, Monterey, CA. He expects to graduate in March 2011.

### **Gurminder Singh, Ph.D.**

Dr. Gurminder Singh is a Professor of Computer Science and Director of the Center for the Study of Mobile Devices and Communications at the Naval Postgraduate School (NPS), CA. His primary area of focus is wireless and handheld device technology. He is currently involved in research in Mobile Devices and Wireless Sensor Networks. Prior to NPS, he was the President and CEO of NewsTakes, Inc., a company specializing in repurposing of multimedia content for delivery to wireless networks and devices. He has been involved in this area for the last 10 years. He received his Ph.D. in Computing Science from the University of Alberta, Canada.

### **LT COL (Ret) John H. Gibson (USAF)**

John Gibson served as a communications officer in the USAF for over 20 years. He has taught computer science at the Naval Postgraduate School for 15 years, specializing in Networking and computer architecture. His research interests include cellular and wireless communications, disruption tolerant networking, and unmanned vehicles. He holds MS degrees in Computer Science and Systems Technology (JtC3) from NPS.