OmniView

Live navigation and communication tool for Google Glass

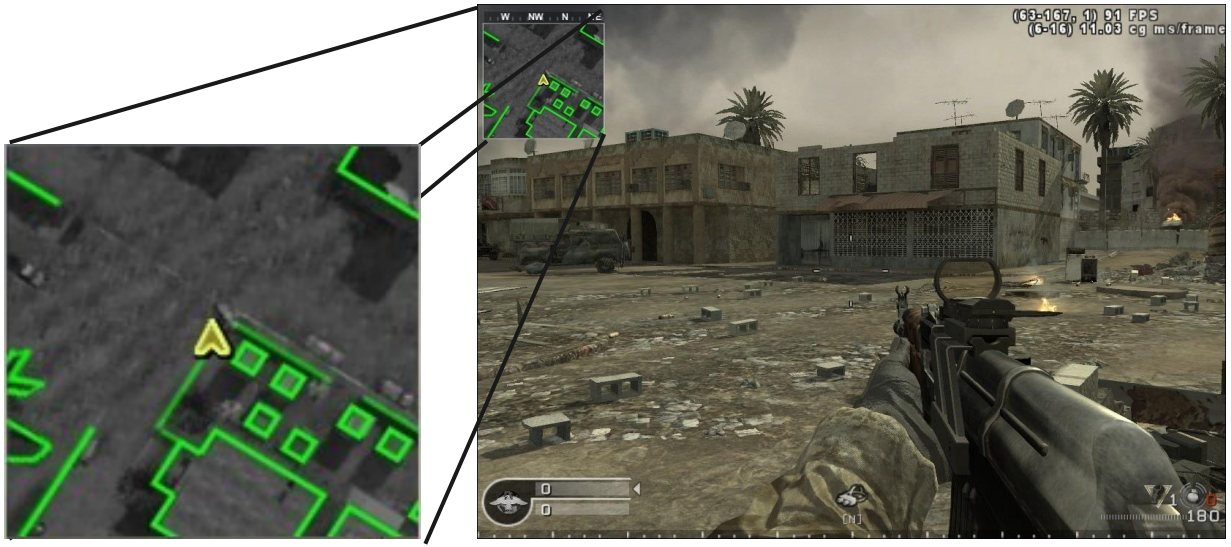
Senior Design Team 14-22

Senior Design Assignment

Client: Dr. Gilbert

Objective: Create a map and communication application on the Google Glass for US Army training missions.

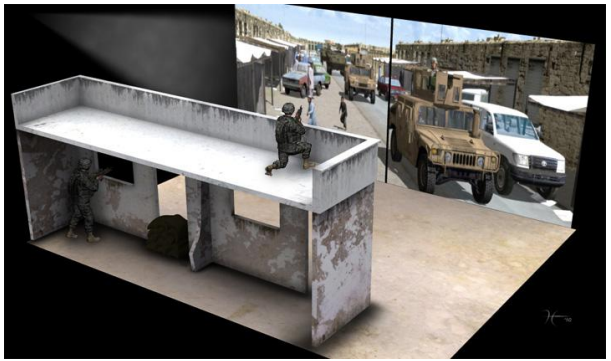
Project Concept



Call of Duty 4

Project Resources

Google Glass



MIRAGE

Project Resource: MIRAGE

Project Goals

- Real-time location data at a glance
  - Glass allows data and Images in peripheral vision
- Full communication with comrades (other users)
  - Bidirectional text messaging
- Scalable for multiple users supporting various devices
  - Google Glass
  - Android Phones
  - iOS Devices

Development Process

- Agile development process
- Meeting or emailing with Dr. Gilbert weekly to discuss our progress
- Two week ‘sprints’
- Each member of the team would work on specific goals for their specific modules.
- Weekly ‘stand-up emails’
  - Accomplishments
  - Goals
  - Challenges/Roadblocks

System Design

Our system can be best split into five applications:

- **MIRAGE**
- **Google Glass / Android Application**
- **iOS Application**
- **Server (Command Center)**
- **Web**

Incremental Test Plan

As we began developing the different modules for this project, we spent about 15% of our time testing each of them and ensuring the smoothest experience possible.

- Web Module
  - We ran test calls to the web page that would populate the database. We then made sure that these changes propagated properly to the screen. These calls would simulate several users.

- MIRAGE Module
  - We created test data simulating data coming from the MIRAGE. We then made sure that this data would be converted properly into GPS coordinates and the correct POST call was created for the server.

- Android/Glass Module
  - Ensured that the web module would load on both phone devices and the Google Glass device. Also made sure that any requests the app would make were of the proper format for the web server.

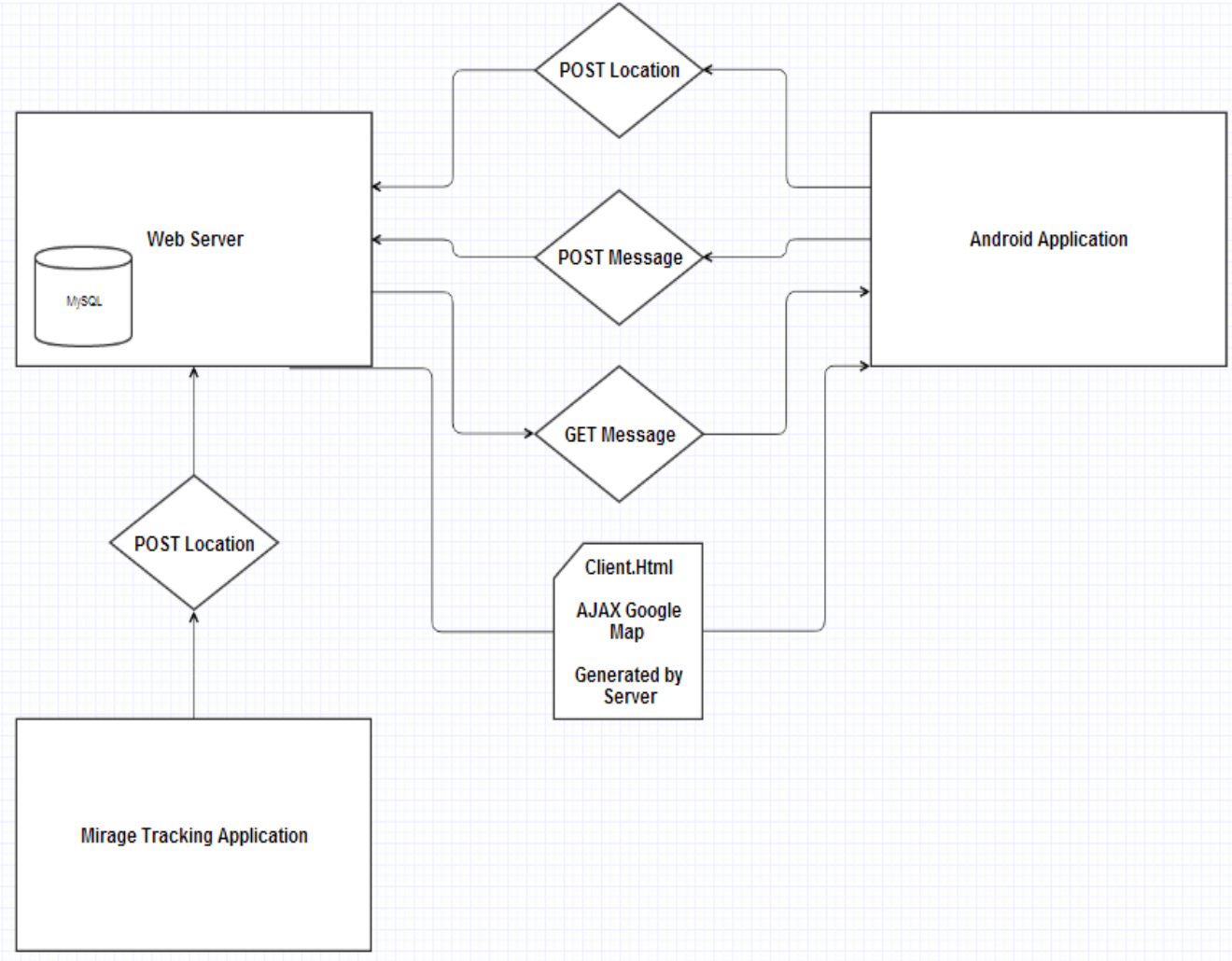
Technical Challenges

Challenge	Solution
Limited development environment for Google Glass. Google Maps not available due to no Google Play Services (rendering engine for Maps on Android)	Create Android WebView that renders the map from a website and uses AJAX for live updates to map
Using software in MIRAGE environment and mapping the internal system to the real world for integration with OmniView	Continual testing of MIRAGE software and manually calibrate our system so that it converts room coordinates to real world coordinates.

Technical Challenges

Challenge	Solution
Developing a system that can work across multiple platforms.	Write an API that could be called from virtually any platform and that works as a single point of truth.
Creating an intuitive user experience while maintaining functionality constrained by Google Glass' s limited interface.	Create simplistic, but powerful gesture-based interface using Glass's touchpad. Utilize Toast API to handle text message passing to users.

Final Design Block Diagram:



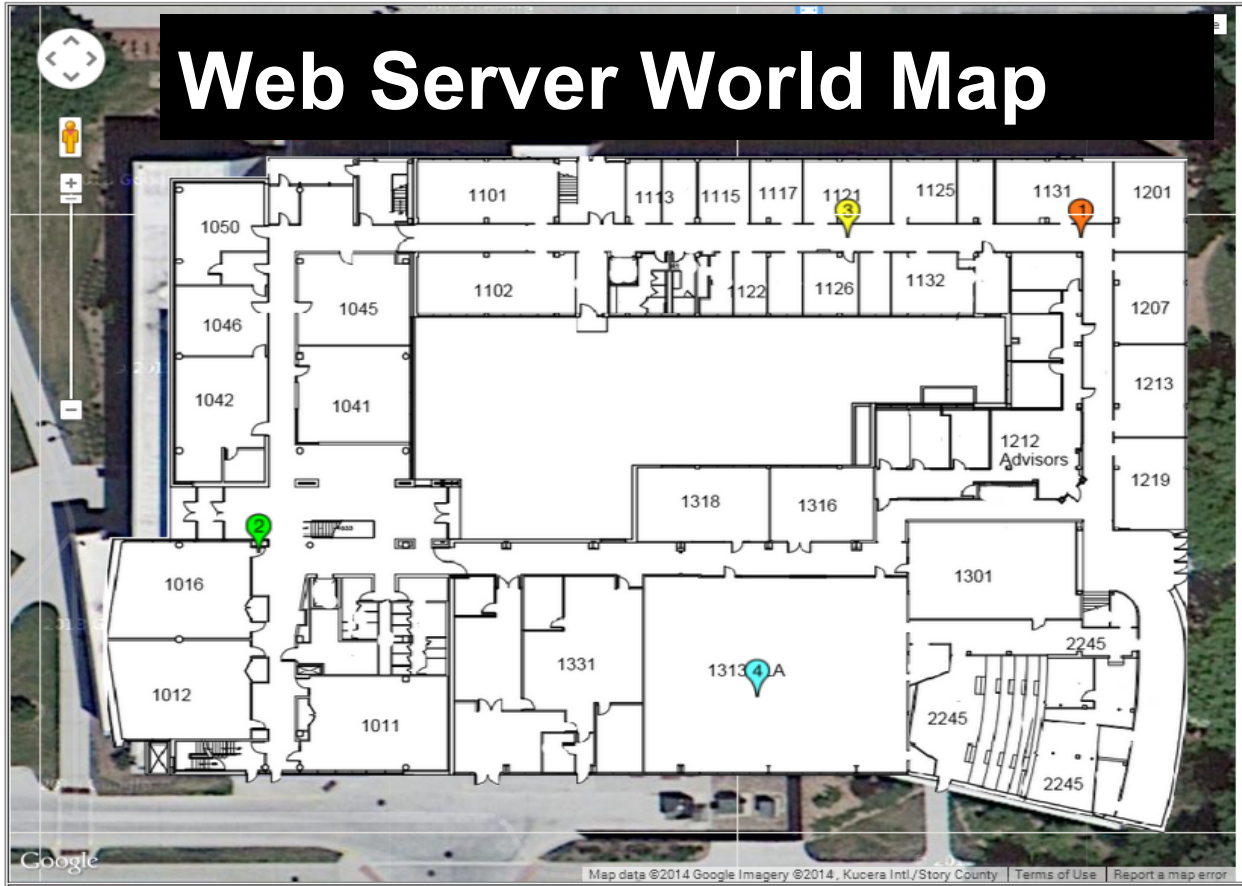
Final Design: Web Server

Front-End: JavaScript and HTML with Google Maps

Communication with Android devices and Mirage: Uses HTTP request methods GET and POST through PHP scripts

Storage and information: MySQL and XML files through PHP scripts

Web Server World Map



Connected Users:

1) Derek
2) Chris
3) Nick
4) Charles

Session Chat:

Commander: Stay safe.

Derek: Be on high alert!

Commander: Nick and Derek work together while moving through the hall

Chris: Copy that sir.

Commander: Watch your left Chris

Charles: Moving in sir.

Commander: Meet at the marked location.

Add Custom Map

Map Image: Choose File | Map Applicat...Example.png

North-East Corner Latitude Coordinate: -92.32424 North-East Corner Longitude Coordinate: -92.32424

South-West Corner Latitude Coordinate: -92.32424 South-West Corner Longitude Coordinate: -92.32424 Add Map

Clear Inactive Users From Map

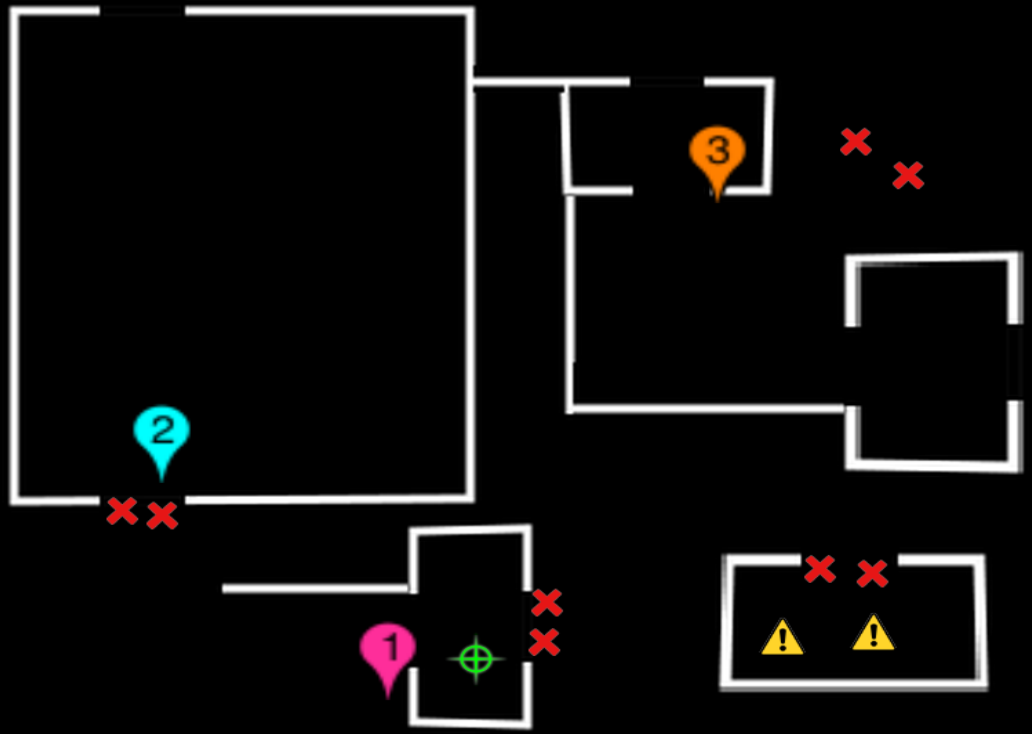
Clear Session Chat

Keep going!

Send message to users

Click Coordinate Info: Latitude= 42.028742999569985 Longitude= -93.65087714510992

Web Server MIRAGE



Google

Map data ©2014 Google | Terms of Use | Report a map error

Session Chat:

Rachel: 10-4.

Commander: Rachel, meet up with Derek and Chris. Return to base.

Rachel: Objective reached.

Rachel: Entering the building now commander.

Chris: Copy that, holding steady.

Derek: Chris, let me catch up to your location.

Rachel: Entrance blocked commander, I will go around to the front of the building.

Derek: Exiting the building now.

Commander: Caution thermal reading received from shed. Map has been marked.

Commander: Copy that Derek, markers have been added to the map.

Derek: The exit is blocked.

Commander: Keep going!

Connected Users:

1) Rachel

2) Derek

3) Chris

Send message to users

Clear Inactive Users From Map

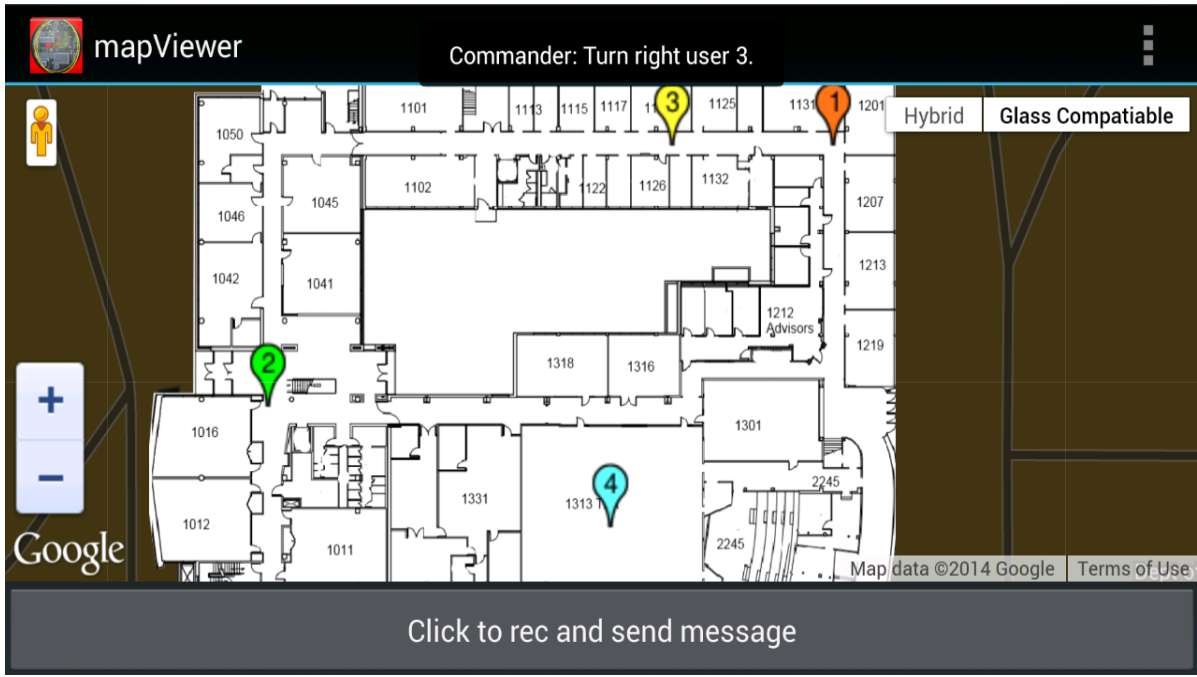
Clear Session Chat

Marker Style to add to map:

Final Design: Client App

-Map Generated from HTML page hosted on web server.

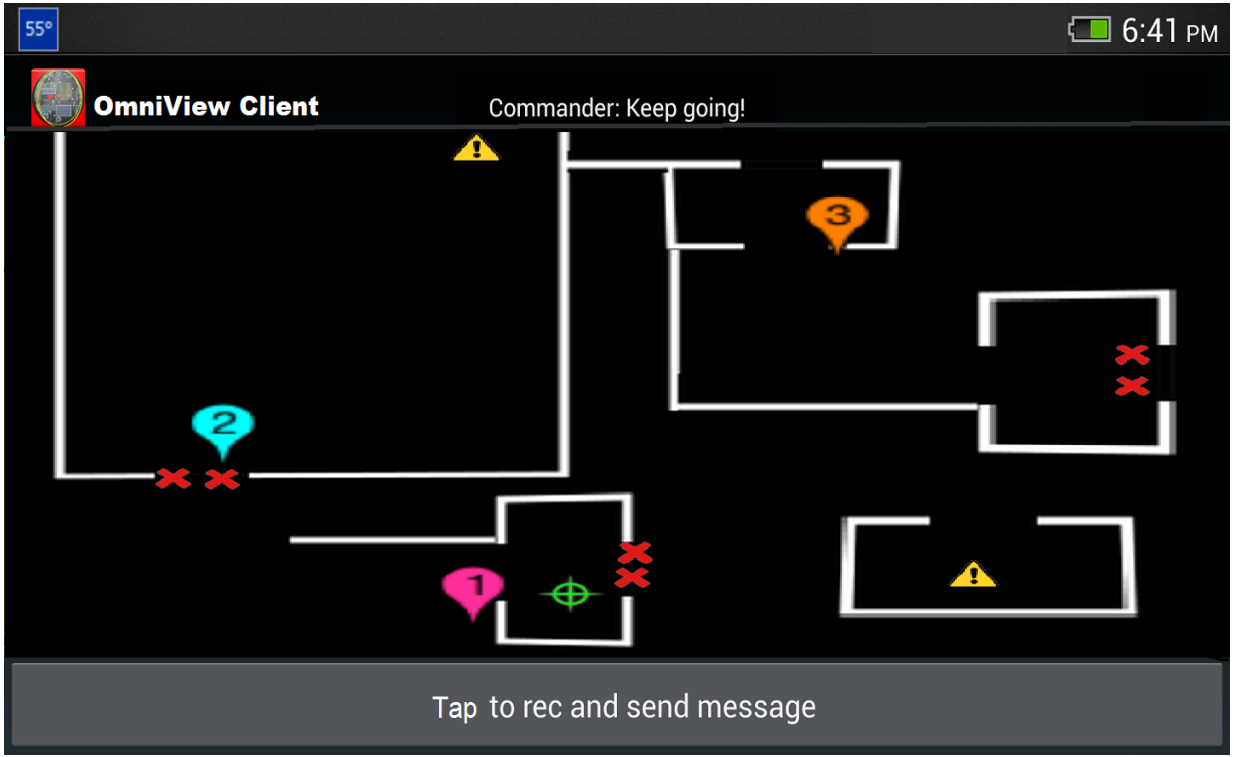
-GPS data is POSTed from Android to web server at each location update



Final Design: Client App

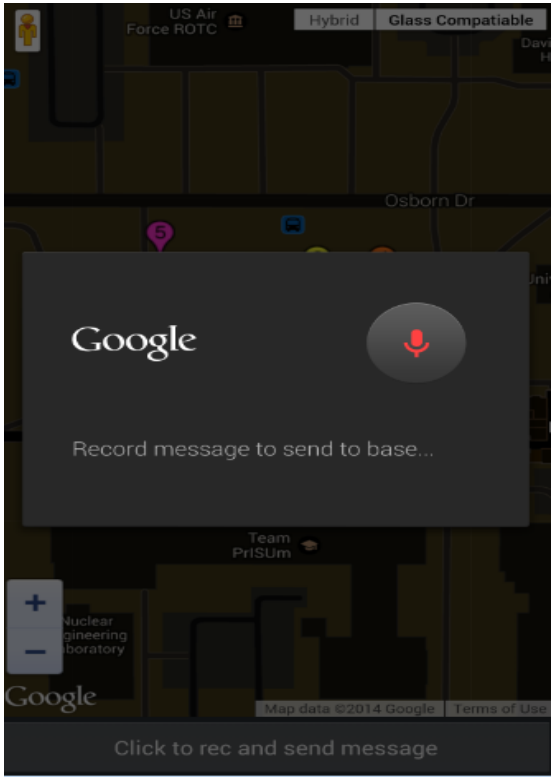
-Client App inside MIRAGE

-GPS data from MIRAGE system



Final Design: Client App

Chat through Voice-to-text which is then POSTed to web server database



Displays message with http response GET to web server database

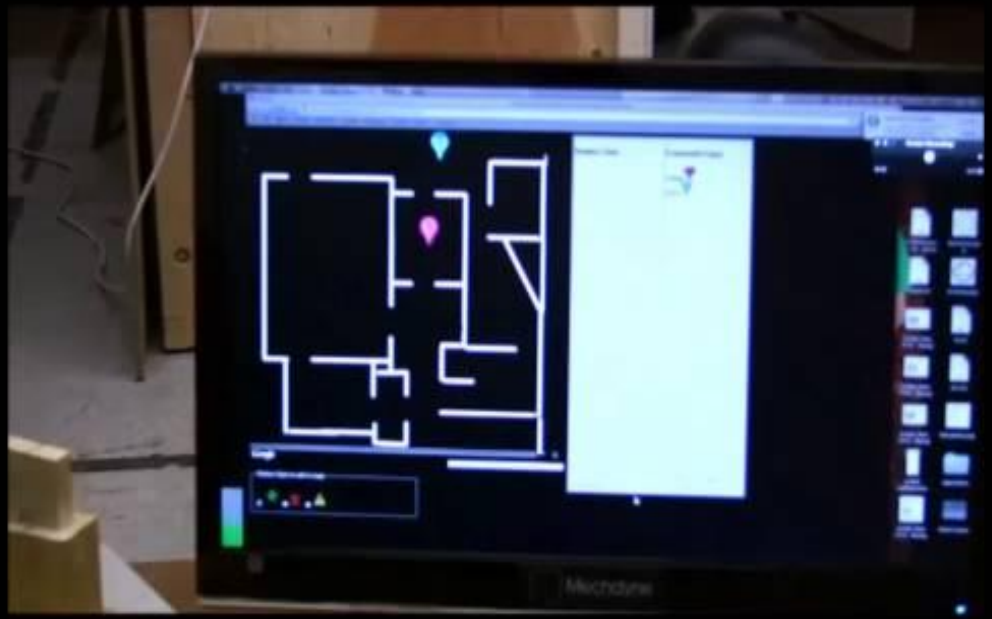
Final Design: MIRAGE

Uses data from the MIRAGE environment

Data converted from relative locations to GPS

Makes post requests to our application

Final Design:

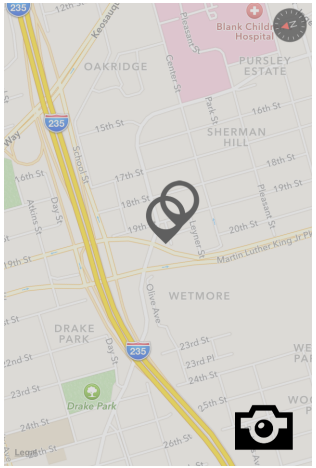


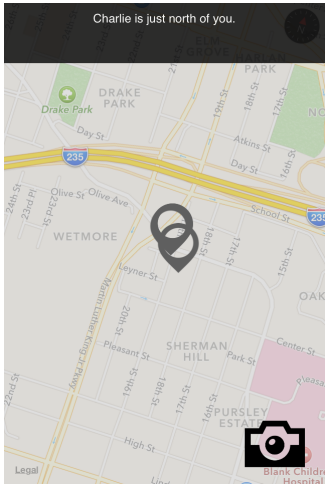
Questions?

Development Schedule

iPhone Application

Born out of the need for more devices using our APIs, we created an iPhone application that could receive messages, post photos, as well as track locations.





Integration Test Plan

We focused heavily on integrating the system and making sure that all of the components worked seamlessly as a unit.

- We will be testing the glass application extensively with members of our team and some volunteers to ensure that it has the most intuitive and smooth experience possible.
- We will be testing that multiple users are handled correctly.
- We will be testing that the glass application handles the different contexts (MIRAGE and GPS) properly.
- We will be testing that the Web Module has a responsive UI and updates correctly.