INDOOR URBAN SEARCH AND RESCUE
GROUP 2

Ashraf Ali Shaik, Venkat Ramesh, VitastaWattal, Lakshmi Sahitya

Abstract

As the Google Maps or any other positioning system does not quite aid with the directions in an indoor system may be inside a building. So a navigating technique must be needed to help with the routes and directions inside a Building. Therefore, the main goal of our project is to design an android app that aids in Urban Search and Rescue operations. This application visualizes the hazards located within the indoor environment and shows entrapped victims that are discovered by the Urban Search and Rescue (USAR) responders.

Bluetooth low energy or iBeacon localization technique is used as the core technique in developing the ideology of the app. The range of the positioning system is estimated based on an approximation of the relation between the RSSI (Radio Signal Strength Indicator) and the associated distance between sender and receiver. The actual location estimation is carried out by using the Centroid method based on weighted distances. This design approach guarantees a cost-effective solution for the positioning service implementation.

Introduction to Beacons

Bluetooth low energy beacons are the trademark for an indoor positioning system that calls "a new class of low-powered, low-cost transmitters that can notify nearby Bluetooth devices of their presence." The technology enables a smart phone or other device to perform actions when in close proximity to Beacon. One application is to help smart phones determine their approximate location or context. With the help of anBeacon, a smartphone's software can approximately find its relative location to Beacon in a store. Beacons can help a phone show notifications of items nearby that are on sale, and it can enable payments at the point of sale (POS) where customers don’t need to remove their wallets or cards to make payments. Beacon technology works using the Bluetooth Low Energy (BLE) technology, also known as Bluetooth Smart.

Beacon uses Bluetooth low energy proximity sensing to transmit a universally unique identifier picked up by a compatible app or operating system. The identifier can then be looked up over the internet to determine the device's physical location or trigger an action on the device such as a check-in on social media or a push notification. Various vendors have made hardware iBeacons that come in a variety of form factors, including small coin cell devices, USB sticks, and generic Bluetooth 4.0 capable USB dongles.

Beacon is the main component of indoor positioning system in our project. These are low powered and low cost transmitters that can notify nearby Bluetooth enabled devices such as Tablet or any other PDAs. In an indoor positioning system, a Beacon deployment consists of one or more Beacon devices that transmit their own unique identification number (UUID) to the surrounding area. Software on a receiving device may then look up the Beacon and perform various functions based on the purpose of the applications; here in this project, the Android app we designed can receive RSSI of all beacons which are in the vicinity of the Tablet.

PHASE I:

Application Details:

The application includes the client android that can send the location of the ground zero to a rescuer (another android device). The client on android device calculates the exact position of the location using the Bluetooth low energy beacons and plot them in a google maps. This application takes the location of the ground zero and sends to the rescuer app using android sockets over a network channel.

Main features of the application:

- **Sorting (Bluetooth Low Energy) Beacons:**

Tags are the beacons that are deployed in the hallway of Marcus Basement. While the student tags are distributed among different student projects. Any Bluetooth enabled device receives continuous RSSI signals from all the beacons irrespective of distance tags or student tags. But, as the distance tags help in determining the position of the Tablet inside a building, we try to filter student tags in the proximity and consider RSSI signals only from the distance tags. This filtering or differentiating between two types of beacons is done using UUID associated with every beacon.

Every tag has a unique ID that is used to differentiate with the rest of the beacons. The beacons are sorted using the beacon intensities that are known from the Bluetooth driver installer in the device. The distances are sorted in ascending order and the first three are noted down using a Tree map data structure. The first three values of the intensities are stored in a Tree map and are analyzed.

The distances and beacon information are placed in the Tree Map. Tree map based on its property can sort the closest beacon and place the closest beacon on the top of the Tree Map. Then remove the first closest beacon from the Tree map and remember it as the first closest beacon. Now Tree map automatically sorts the remaining beacons and puts the next closest beacon on the top. Remove the next closest beacon and remember it. Repeat the same procedure for the third closest beacon.

- ## Working of Beacon localization technique

Distance can be evaluated from the expression given in the 671 Course Beacon documentation, where the distance is calculated using the attenuation model for Tablets considering the damping effect, absorbing effect of the walls and all other surrounding obstacles.

$$\text{Distance} = -0.00903 * \text{RSSI}^2 - 2.171 * \text{RSSI} - 94$$

Sorting of the RSSI signals of all beacons is implemented using a Tree Map Data Structure. Let the keys of a Tree Map be the distances of beacons from the PDA and the values of Tree Map is key or the Unique Identification Number (UUID) of beacon. By the property of Tree Map, sorting is automatically performed by key and therefore the distances are sorted in ascending order. At the instance when a PDA receives RSSI from more than three beacons, the following approach is adopted to find the first three beacons in the proximity to the Bluetooth enabled device. Later the localization technique described in the above section is implemented to find the exact location of PDA. A Tree Map whose input is an ArrayList of distances of all the beacons close to the PDA. A Tree Map, which sorts the inputs, automatically makes distances between sender (Beacon) and receiver (Tablet) in ascending order. Let us consider the first element in the ascending order to be, say A. Then the first key value pair is assigned as the first closest beacon and is removed from Tree Map. Now the resultant Tree Map is sorted giving the next lowest valued distance to be the first element say B. Then the key value pair associated with the beacon B is assigned as the second closest beacon and is removed from the Tree Map. The same operation is performed to obtain the third closest beacon.

- ## Calculation of Latitude and longitudes

Localization can be done in numerous ways out of which largely used techniques includes three different methods. We assume that the latitude and longitudes of the given marcus basement Bluetooth low energy beacons locations as the Cartesian coordianates in a 2D xy-plane.Now these coordinates are used to deal with the following 2D localization techniques that will yield the required latitude and longitudes :

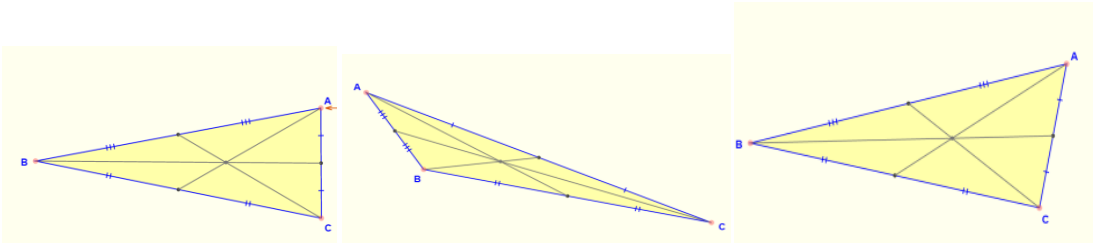
- **Trilateration**:
  This technique involves three circles that are formed from the three points of the known geometry and the required point is found from the intersection of the three circles. Technique involves a complex math and gives an accurate point to refer. This is not used for the complexity and the vulnerability for the closely placed points, As the distance between the points decreases the point of the intersection oscillates.
- **Bilateration:**
  This technique involves two circles that are drawn from any two points from the geometry and the intersection of the circles is noted. Two points are formed from the intersection and the two points are used to find the distance between them and

the third point. The closest point is considered to be the required point as the other point is out of the frame. This technique is not employed for its inaccuracy and the involvement of high complex math.

- **Weighted centroid:**
  This technique involves the weighted centroid calculation of the triangle formed from the three points of the known geometry with the help of the distances from the points to the required locations are known. This involves no complex math and also is more accurate compared to the bilateration technique. This is employed in the project for building a triangle from the closest beacons available and finding the point by calculating the weighted centroid of the triangle.

The localization technique is implemented in the project is by calculating the centroid of three points using the weighted average of distances. This approach gives almost the exact location of the Tablet which is receiving the Beacon signals, and the Beacon close to the PDA can be easily known, with error being up to a deviation of 1meter. In this project, we used weighted Centroid method against Trilateration or Bilateration to reduce the runtime complexity of the program, achieving precision up to 8 digits after decimal point for slight reduction in accuracy as trade off.

Now having the first three closest beacons A, B, C and their corresponding distances, and as we also know the latitude and longitude of these distance tags, the current location of a Tablet which is receiving Bluetooth signals can be determined using the weighted average of distances in centroid principle. Centroid of the triangle yields us a point that is inside the triangle as shown in the below figures. This could yield us the current location of Tablet with almost accuracy. The weighted distances give position relative to distance from Beacon.



Figure(1) weighted centroid estimation

The coordinates of the spacial current location of the tablet using localization technique is given by

X(latitude) = (X1*dis1+X2*dis2+X3*dis3)/(dis1+dis2+dis3)

Y(longitude) = (Y1*dis1+Y2*dis2+Y3*dis3)/(dis1+dis2+dis3)

This gives the current location of our Tablet with a deviation of up to1metre.

Implementation:

Integration with Google Maps:

Beacon Tag Searching app is integrated to Google maps using the Google Maps V2 API. This enables the user to locate the current location of the ground zero area on a google map. Real time updating of the current location of the responder is obtained by using threads in the implementation. The User Interface thread handles all the operations that are needed.

The flow of operations in the User Interface thread:

- Beacons are sorted using the distances of the beacons from the current location of the tablet.
- The first three closest beacons positions are considered and Spatial coordinates or the LatLong of the current location of the tablet is calculated using the implementation of the localization technique.
- The current location of the tablet is marked using customized markers which contain images of the closest Beacon.The spacial coordinates of the tablet obtained from the above operation are assigned to the corresponding image of the nearest beacon and are marked on the maps.

Image markers:

Markers are needed for plotting the location on the google map. Image markers are used instead of standard markers to make things easier in locating the point. The images of all 17 beacons of the hallway are collected and are stored in a bit map. Whenever the device goes near a beacon, The bit map is called and the appropriate beacon image is displayed as a marker. This helps the user and rescuer to locate the points quickly and accurately. These images are again called in the server device that is used by the rescue team. The rescue team can identify the beacon image and can get to the location in a much easier way.

Working Android App:

Figure (2) is the demonstration of the real time updation of the responder's location, the customized marker image shows the beacon to which the responder is closest to (here in this case, the responder is closest to the beacon 11). Similarly Figure (3) shows the situation when the responder is closest to the beacon 13.

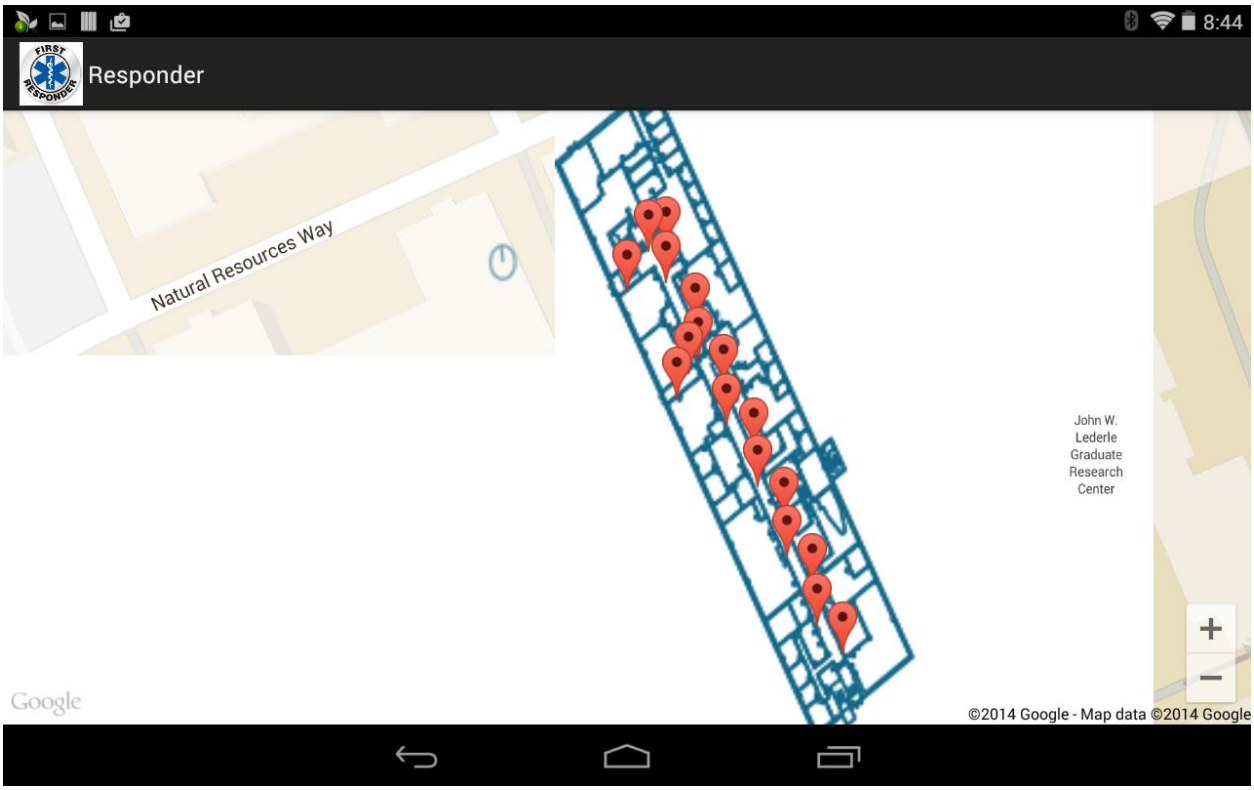


Figure (2)

The above screen shot shows the first screen after running the app in the hallway. The hallway is drawn in a java script and is called from an URL this shows the exact positions of the beacons in the hallway.

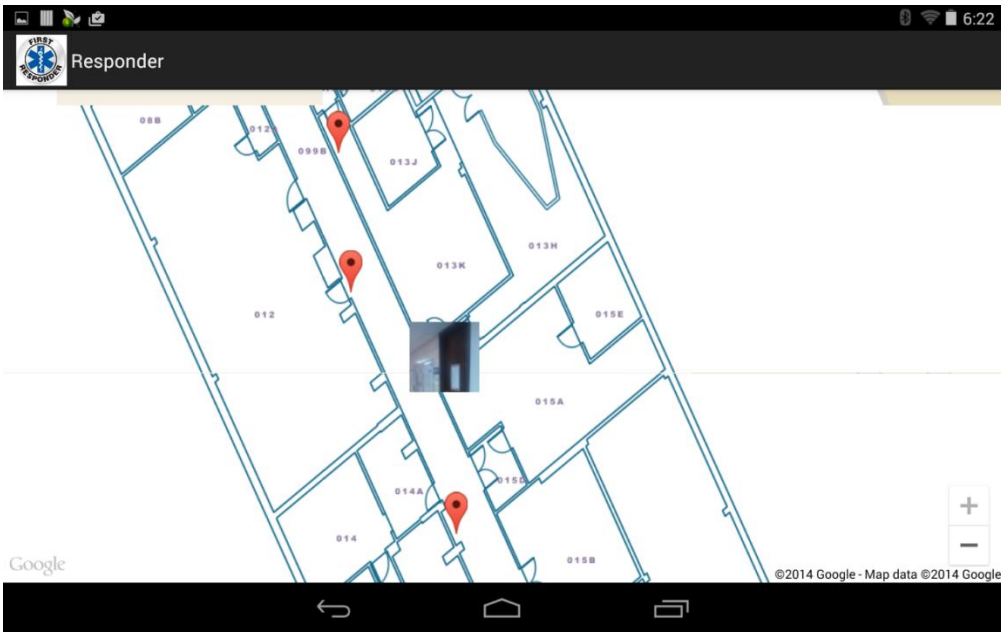


Figure (3)

The above screen shot shows the image marker of the closest beacon. The marker marks the current location of the device in the hallway. The closest beacon is calculated from the hash map created and analyzed. The device shows the closest beacon picture as the marker at the point. By which the user can know the nearest beacon location and it makes it easy for the rescuer team to respond quickly.

In the above figure it is shown that the device is nearer to beacon number 11 and the location is marked with the image marker.

PHASE II

Rescue Team Application! Why use Client – Server?

First responder responds immediately at the disaster affected region to initiate the rescue operation. In the process of search of victims, victims' information is transmittedconstantly to the medical center that is set up in near by location. The server resides at the medical center which has a set of teams which are to be deployed with all the medical equipment required based on the information of the responder(client). This approach of Client Server is a life saver and therefore helps immensely in evacuating and providing medical help to the victims very quickly.

Client –Server communication – Socket programming:

Socket programming is used for communication between Client and Server. Both client and server are connected to a network through where the information is sent in bytes. The entry activity details and the current location of the responder are a bit of information that are transmitted across the network. Use of sockets establishes a strong communication enabling the channel to transmit any number of bytes between client and the Server.

Client-Server Implementation:

Client-Server Architecture is implemented to demonstrate the real time working the urban search and rescue app.

The following are the step-by-step process of search and rescue.
Client Side:

- Client starts the welcome activity and click on the search, and another activity is started.
- Google maps of Marcus Basement which markers of location tags will be displayed.
- Software in the client side application receives the beacon information.
- Sorting technique described in above sections is performed, and localization technique displays the location of the client with the image of that beacon to which the client is closest to.
- On Clicking the image Marker, Entry Activity is started where number of victims and their medical condition can be entered. We can also add Markers to mark a location

- On Submit, this information goes to the Server who takes the necessary action by deploying rescuers with first aid equipment.

Server Side:

- Server waits until it receives the list of victims from the client.
- Once received, server analyzes the criticality of the victims affected and deploys rescuer into the hazard affected building with necessary medical equipment.
- A line connecting rescuer and the victim is displayed on the server side device.

Graphical User Interface:

Entry activity is started on click of search button on Welcome Activity. Entry Activity: the responder uses the entry activity to enter the details of the victims. These details include number of the victims in a particular location, the medical condition of the victims and the level of priority of rescue. This level of priority includes three factors, critically injured, minimally injured and OK or survived with very minimum injuries. On Clicking Submit button the information is transferred to the Server.

On receiving the information from the client, server analyzes the information and immediately responds by deploying the rescue teams with all the primary medical equipment for evacuating and rescuing the victim. For the guidance of the victim location, rescue team will be provided with a line connecting their current location and the victim's location for them to estimate the distance. This gives a good understanding for the rescue team on how to reach to the victim's location.

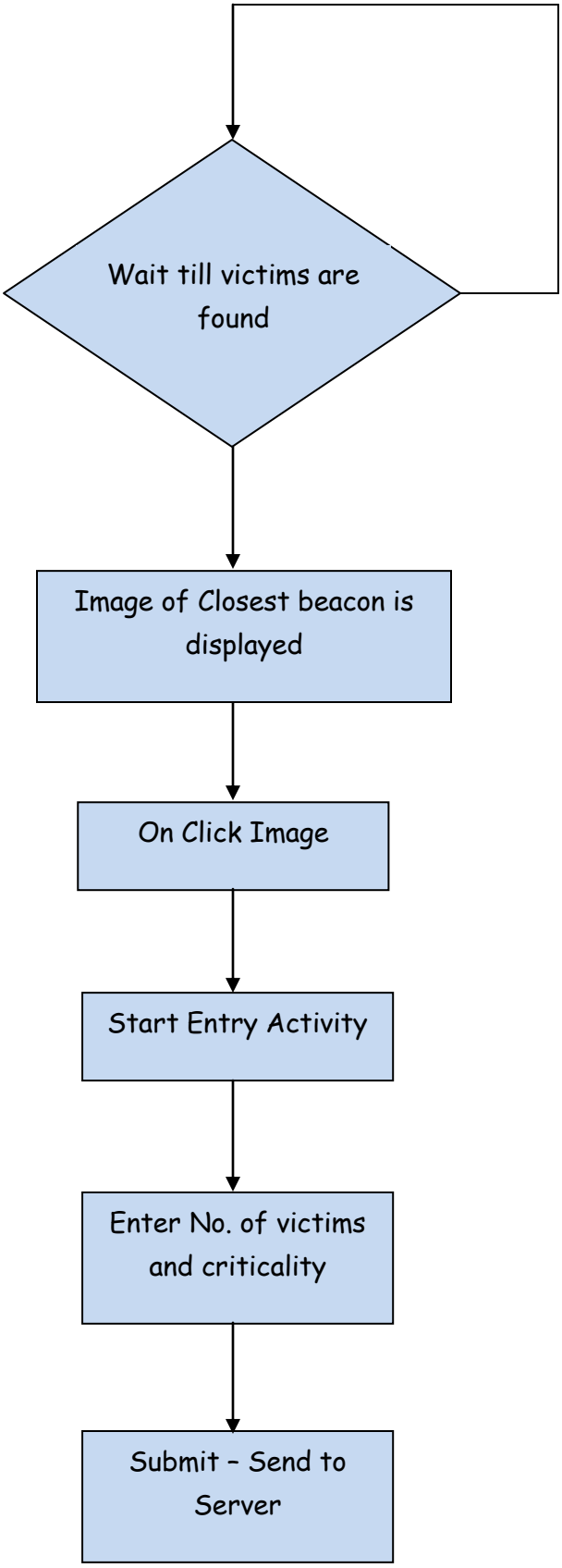


Figure (4) flowchart Client side

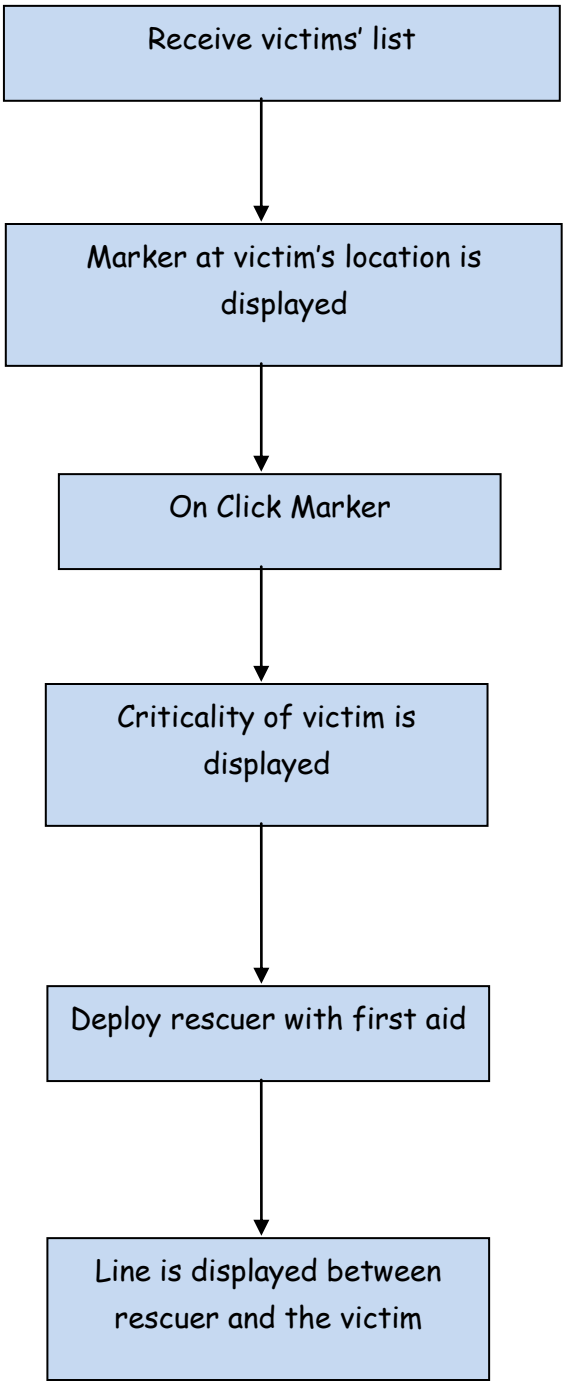


Figure (5) flowchart Server side

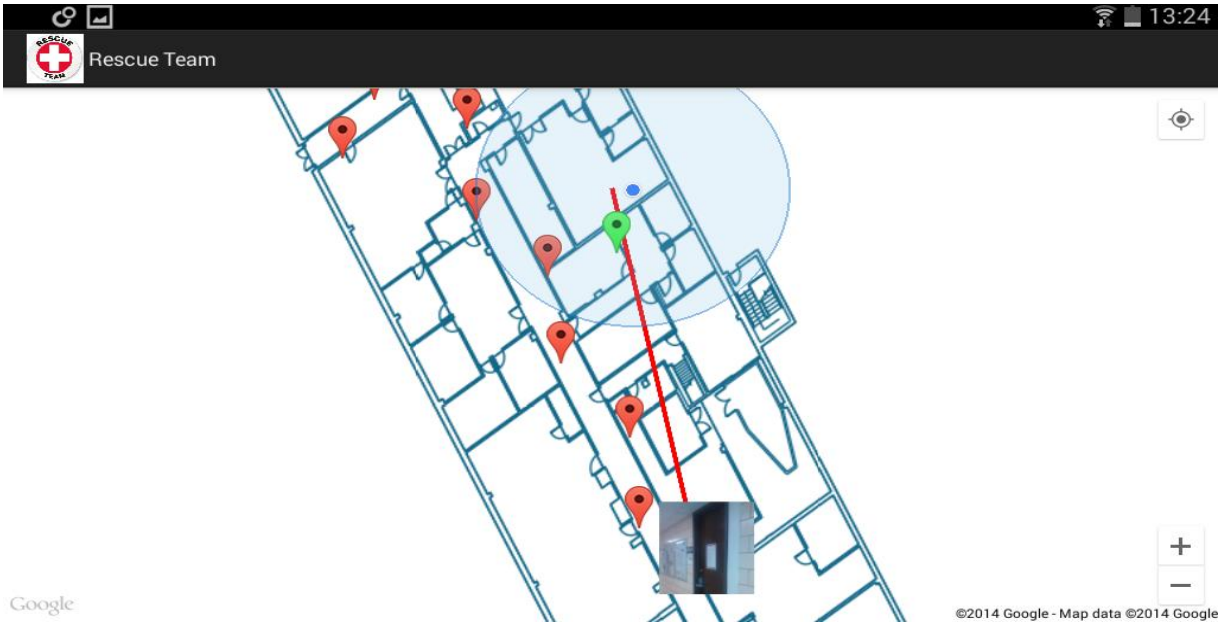


Figure (5)

This above picture, figure(5), depicts the rescuer application plotting the point that is sent through android socket on a network channel. In the picture we can see the marker is marked at the beacon on the map with a image marker. The green marker shows the rescuer team's location on the map. A route is drawn from the image marker i.e., the ground zero location to the green marker i.e., the rescuer location.

Figure (6) Entry Activity (Client)

The above figure(6) shows the responder application where the user gives the details of the incident after marking it on the map. We see the location of victim as LatLong position in the above figure, also date and time of the tracing the victim is observed in the above figure.

Figure (7)

This is another example where the responder gives the details of the emergency by sending the texts “minor injury” ,”low”,”2”. That helps the rescuing team to understand that there is a minor injury to 2 victims with a low severity. By this way the team could prioritize among the given data and make their way to the location.

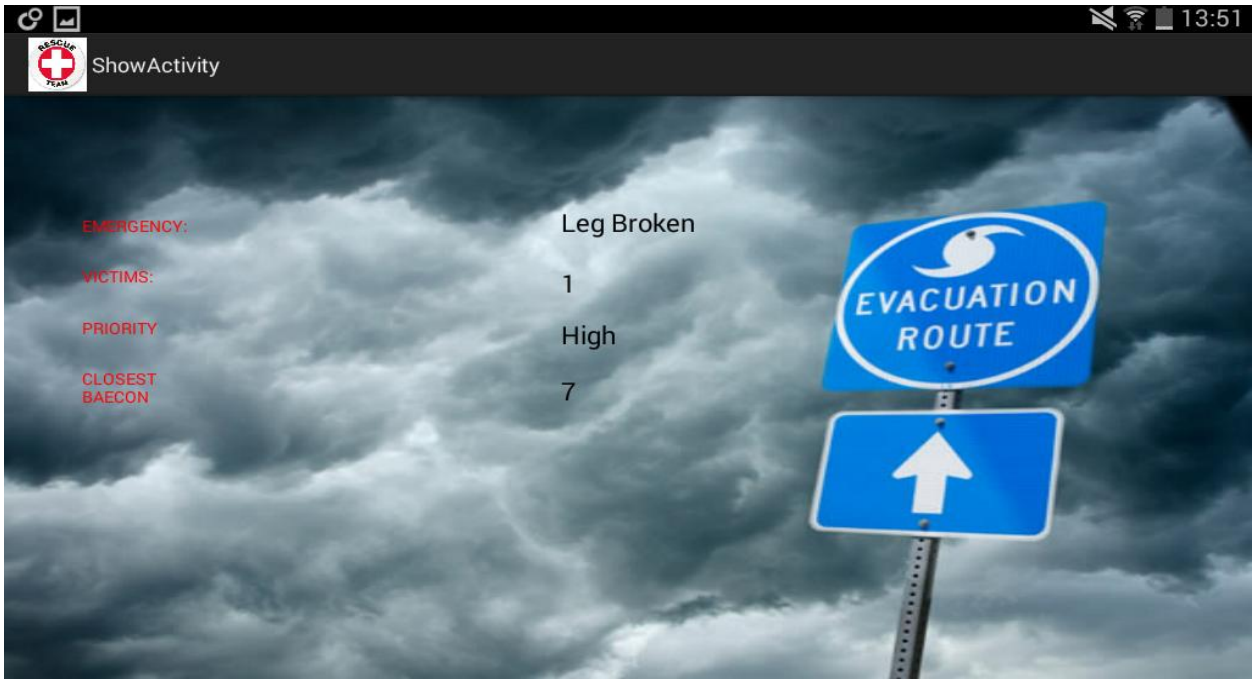


Figure (8) OnClick of image on Server side, above page is activity displayed

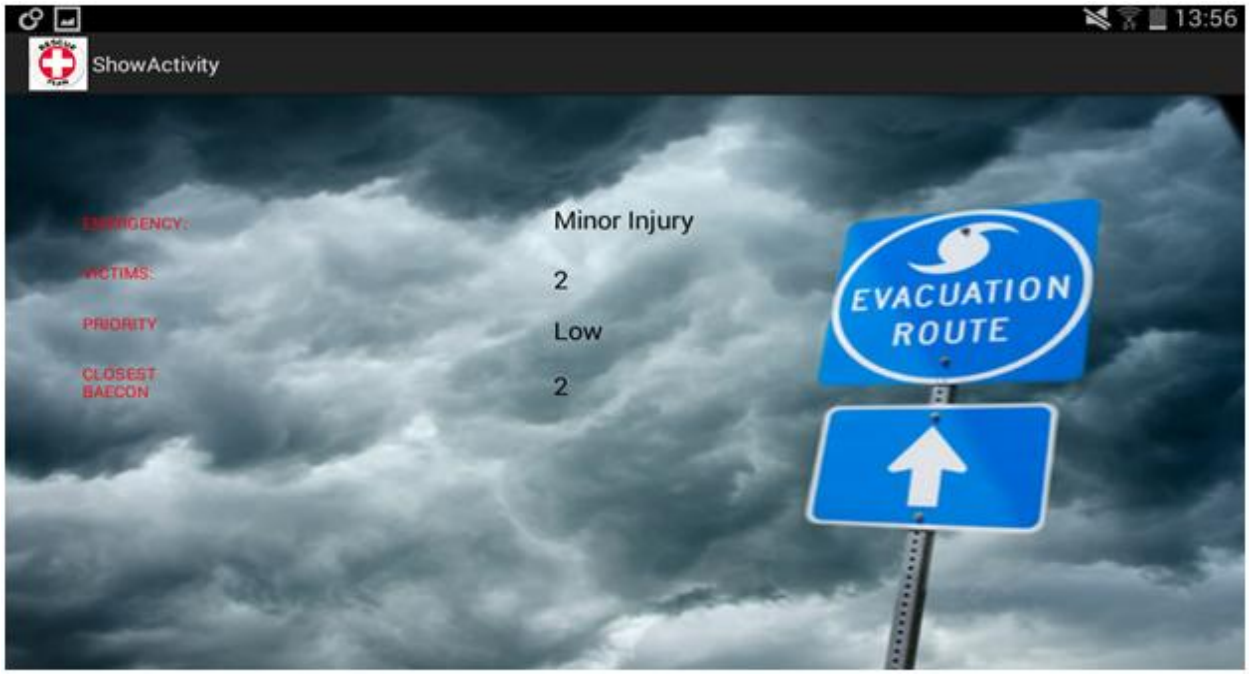


Figure (9) OnClick of image on Server side, above page is activity displayed

Figure (9) Server side trace of victim

APPENDIX

Ashraf Ali shaik	Image marker implementation on Server & Client; Sorting of Beacons; Debugging; Integration of code; Test cases
Venkat Ramesh Gali	Localization : Involves the algorithm and implementation along with test cases Android client and server: Socket programming, its implementation along with its test cases
VitastaWattal	Implementation of Sorting of beacons; Integration of code; User Interface both on Client & Server side; Test cases
Lakshmi Sahithya	Client – Server Implementation(Socket Programming); For Sorting, localization implementation(Code Snippets); Documentation; flowcharts; User Manual

REFERENCES

1.5g.ecs.umass.edu/671/

2.“An indoor Blurtooth-based positioning system: concept, Implementation and experimental evaluation” S Feldmann, K Kyamakya, Ana Zapter, ZiguoLue, Institute of Comuunications Engineering.

3. “An Inexpensive Bluetooth-Based Indoor Positioning Hack” Kenneth C. Cheung, Stephen S. Intille, Kent Larson

4. “A Low-Complexity Geometric Bilateration Method for Localization in Wireless Sensor Networks and Its Comparison with Least-Squares Methods” Juan C. Ruiz, Jose G. Rosiles, Ernesto Sifuentes, Pablo R. Perea