Emergency Alert and Response System

Pruthvi Elavarthi
Arizona State University
Pruthvi.Elavarthi@asu.edu

Sai Harshitha Rayidi
Arizona State University
srayidi@asu.edu

Teja Kunderu
Arizona State University
tkunderu@asu.edu

Praneeth Ganta
Arizona State University
pganta@asu.edu

ABSTRACT

Growing technologies leading to higher expectations and high scaling issues are one of the major reasons for the demand of mobile computing. In today's world because of increase in the crime rate there is lack of human safety which leads to many unwanted and untraceable situations. Being in such a situation, humans needs to look into their own safety, which now has become a primary concern. This application, Emergency Alert and Response System, plays a huge role in tracking the person's safety. Now, by integrating machine learning capabilities with mobile applications, we can trace a person's safety using this application.

Keywords

Android, Mobile Computing, Machine learning, crime rate.

1. INTRODUCTION

We planned on building an Emergency Alert and Response System on Android. This application will warn the user when he/she visits an unsafe location or locations which have a higher crime rate by monitoring his/her GPS location and predicting based on historical data. This application mainly helps the user guard himself/herself from security threats like kidnapping, shootouts etc., Whenever the user is travelling, the application predicts unsafe locations/places near him/her, and alerts the user regarding the same. User has the provision to open the application and view those locations to check if they fall in any of the locations he/she is visiting. Based on user settings, the application activates the emergency system, and sends text messages to the emergency contacts with the latest GPS locations based on two conditions. A) abnormal situation - phone switching off, network loss etc., B) user alert - user manually selects an alert situation. Application has a built-in fall detection algorithm that helps avoiding false alarms to the emergency contacts.

2. CHALLENGES AND CONSIDERATIONS

2.1 Fall Detection

The system uses a fall detection machine learning algorithm to avoid unnecessary activation of the alert system. Integrating this as a service to the application was a challenge.

2.2 Continuous Tracking

Even when the user closes the application, it keeps track of the user. Handling this with a service was a challenge.

2.3 Lost and regained connection

Detection of network loss and regained connection using the server capabilities and alerting the user based on the respective scenario was a challenge

2.4 API's Usage

Using Twilio API to send server messages for lost and regained connection and google API for maps and distance calculations was a challenge.

3. DESIGN & IMPLEMENTATION

This section consists of all the tasks which were developed for the successful execution of the project.

3.1 Login Screen – Task 3, 17, 22

The first stage when the user opens the application is the login screen. Along with the login screen. the user is prompted to give access to the location and SMS services built in the mobile for application usage.

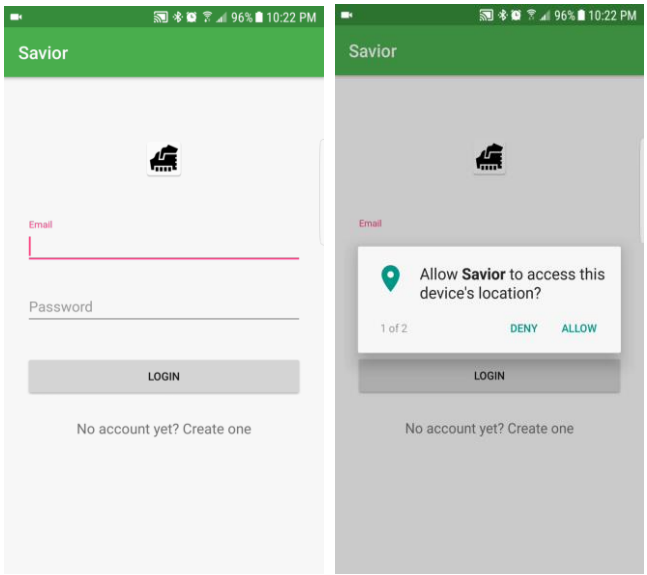

Figure 1.1 Login screen (left), Permissions (right).

3.2 Sign Up – Task 3, 4, 23

A first-time user of the application needs to sign up for the app by completing a form with his/personal details, username (email) and password.

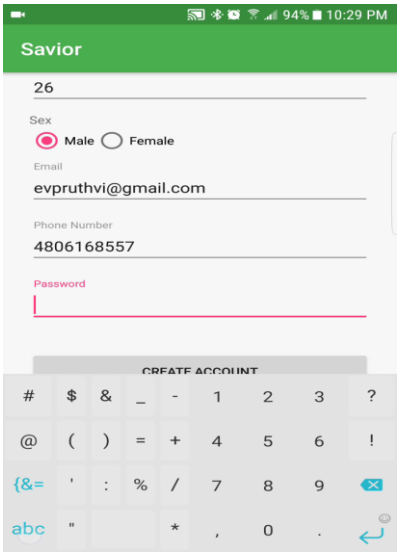

Figure 1.2 Signup Form.

3.3 Home Page – Task 2, 5, 8, 18, 19,

The home page consists of a map view which provides the user with all the crime locations pinned to the map based on the server crime data. These can be unsecured or threatening areas as recognized by the US Government.

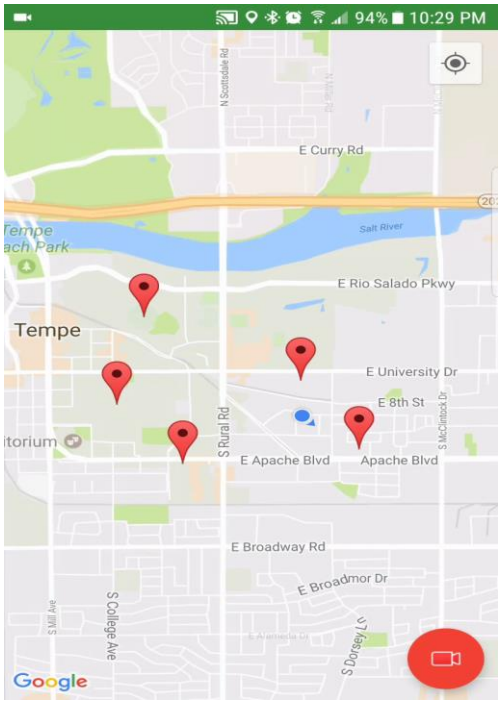

Figure 1.3 Map view home page.

This view also consists of an alert button at the bottom right which enables the user to activate the system manually when in emergency or in a threat situation. As mentioned previously, the emergency contacts will be alerted with a text message regarding the same. The map view is fully customizable and user interactive. It enables the user to use it as a google map and helps him/her to check different locations by zooming in and out. When the user selects one of the pinned locations, full address of that location is displayed.

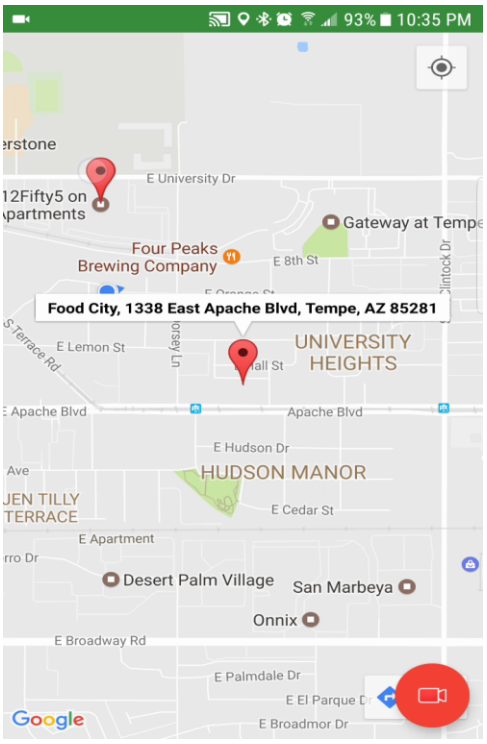

Figure 1.3 Selection view of pinned locations.

3.4 User Settings – Task 3, 6, 11

Once the user logs in, he/she should update the emergency contact details to let the application send alerts to the emergency contact in the situation of an emergency. The system provides the user with an additional feature to predict the threat based on a user defined distance threshold. The default value set by the application is 500m. This threshold is used internally by the application to calculate the nearest crime areas to the user within the threshold set by the user.

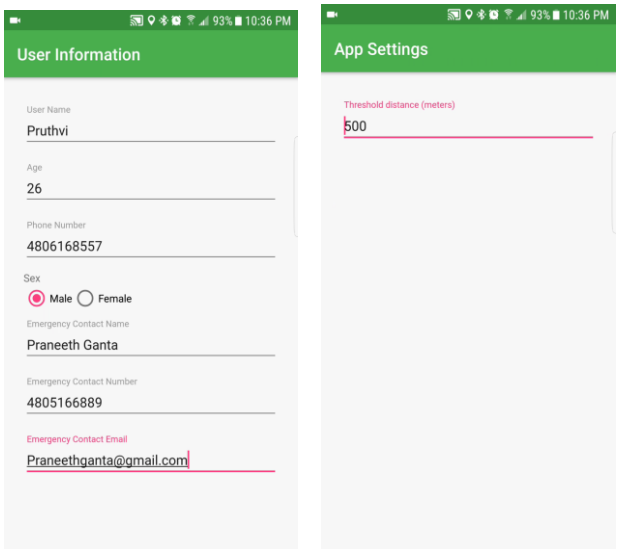

Figure 1.4 User Settings.

3.5 Manual Alert System – Task 2, 7, 13, 14, 16

The basic and primary feature for the user is to be able to activate this system manually. This can be achieved by clicking the push button present at the bottom right of the home screen. Which automatically starts the alert system and sends the alert messages to the emergency contacts along with the GPS location.

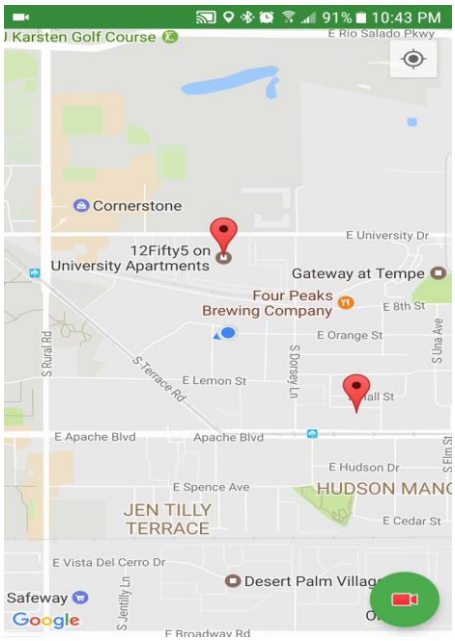

Figure 1.5 Activated alert System home screen.

Once the alert system is activated the push button changes from red to green color indicating the alert system is active and alert messages are automatically being sent to the contacts registered as emergency contacts.

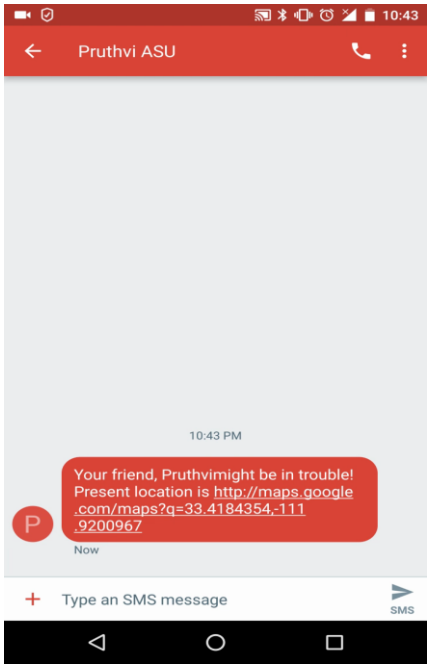

Figure 1.6 Message sent by the user.

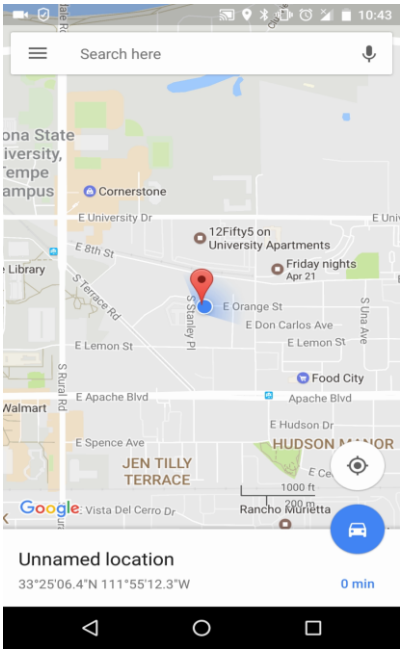

Figure 1.7 Message redirection to Google Maps once clicked on message content.

3.6 Password Authentication for deactivation – Task 4, 9

When a user unknowingly activates the alert system or feels safe after a certain amount of time, he/she can deactivate the system but, to avoid cases where the user's mobile phone gets into the hands of an unauthorized user who is a threat and wants to deactivate the system, there should be some kind of authentication mechanism. For the same reason, a password enabled deactivation system is used, and the user is prompted to enter a password in order to deactivate the system.

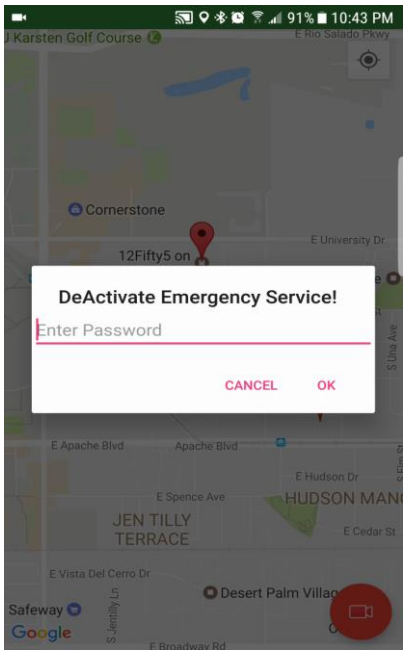

Figure 1.8 Password Prompt Screen.

Once the user provides the correct password, the system goes back into the original state.

3.7 User notifications – Task 1, 2, 12, 13, 14, 17, 20

User is notified with an alert whenever he/she is approaching a crime location. At this point, the prediction system comes into play, and detects the nearest crime locations to the user. It the starts alerting him with a beep notification. Once this is initiated, the user can manually start the alert system, if he/she feels unsafe.

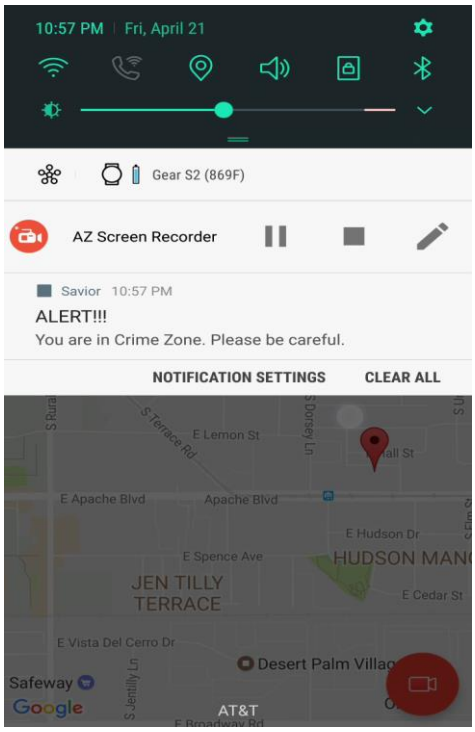

Figure 1.9 User notification regarding the alert.

3.8 Automatic Alert System – Task 7, 10, 24

3.8.1 Method 1

In this method, whenever the user enters a crime location, the system immediately alerts him with a notification. While the system is in this situation, we used a fall detection algorithm, to recognize whether the user is in a normal state or has faced a sudden fall. This fall can either be the user falling while holding his/her phone or someone throwing away the user's phone. When a fall is detected, the system automatically activates the alert system, and starts sending messages to the emergency contact (These are the similar to the messages shown in Figure 1.6).

3.8.2 Method 2

In this method, we have a central server which keeps track of each user, and activates the system based on specific conditions. One major scenario is network loss. The server has a buffer of 30 seconds. If it doesn't receive any location update from the mobile application, it checks whether the last recorded location of the user in with the threshold range of a crime area. If it is, then the emergency contacts are sent an alert message saying the connection with the user is lost. This scenario is implemented using a python server and Twilio messaging API to send alerts to the emergency contact.

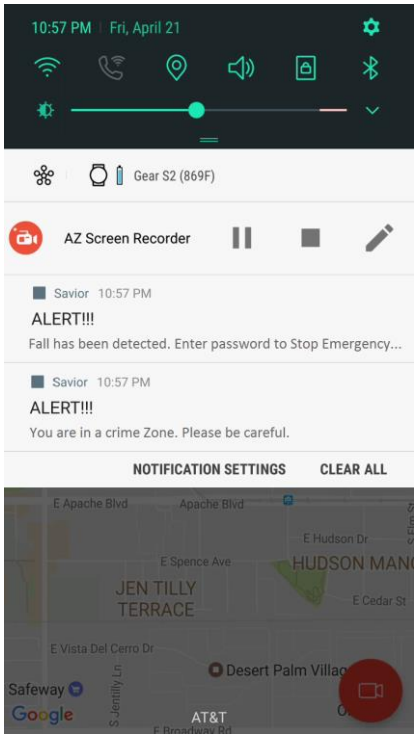

Figure 1.10 Fall detection alert system activation.

Once the user regains his/her network, the system recognizes the recovery, and alerts the emergency contact regarding the safety.

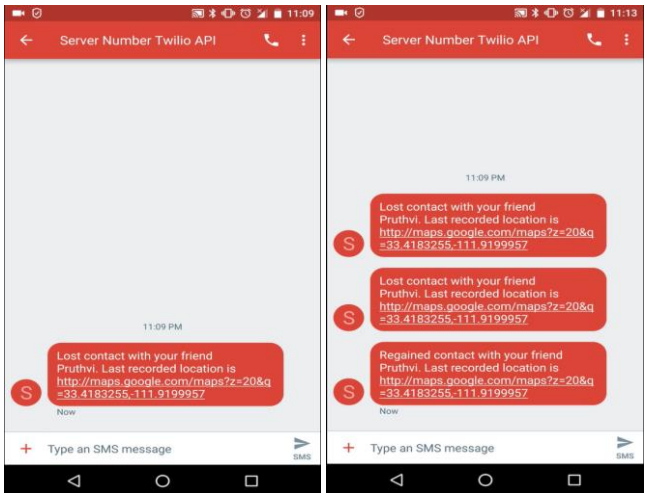

Figure 1.11 Server message for contact loss (left), Server message for contact regain (right).

4. TASKS STATUS

Task No	Task	Assignee
1	Location Using GPS	Praneeth Ganta
2	Refreshing local database	Praneeth Ganta
3	Material Design concepts for UI	Praneeth Ganta
4	Password Input	Praneeth Ganta
5	Maintaining crime data	Praneeth Ganta
6	Emergency contacts input	Pruthvi Elavarthi
7	Sending alert messages	Pruthvi Elavarthi
8	Plotting crime data on map	Pruthvi Elavarthi
9	Password Authentication	Pruthvi Elavarthi
10	Server polling for network status	Pruthvi Elavarthi
11	User Settings	Sai Harshitha Rayidi
12	Network connectivity status	Sai Harshitha Rayidi
13	Local Database updation	Sai Harshitha Rayidi
14	Server Database updation of user location data	Sai Harshitha Rayidi
15	Background working using services	Sai Harshitha Rayidi
16	Manual alert system activation	Teja Kunderu
17	Notifications	Teja Kunderu
18	Getting nearest locations	Teja Kunderu
19	Address view on map	Teja Kunderu
20	Prediction of threat locations	Teja Kunderu
21	Fall Detection	Pruthvi Elavarthi
22	Login	Sai Harshitha Rayidi
23	Sign Up	Praneeth Ganta
24	Twilio and Google distance matrix API for alerts on server	Teja Kunderu

5. ADDITIONAL TASKS

A fall detection algorithm is implemented as an additional task to make the application more robust and less error prone. We have even implemented login management as another additional task which will keep un authorized users away from the application.

6. ACKNOWLEDGMENTS

Our sincere thanks to Dr. Ayan Banerjee for giving us relevant knowledge about real world scenarios of mobile computing and guiding us in the right direction in implementing a successful application.

7. REFERENCES

[1] Google. 2008. Android Developers API Guide. (2008).https://developer.android.com.

[2] StackOverflow. 2006. StackOverflow developers forum. (2006). https://stackoverflow.com.

[3] Twilio REST API - https://www.twilio.com/docs/api/rest

[4] Google Maps API – https://developers.google.com/maps/

[5] Google Maps Distance API - https://developers.google.com/maps/documentation/distance-matrix/

[6] Login Management, Material Design Concepts - https://sourcey.com/beautiful-android-login-and-signup-screens-with-material-design/