EmerSave: A Novel Crowdsourcing Emergency Response Application

Vishnu S. Penubarthi

Massachusetts Academy of Math and Science at Worcester Polytechnic Institute

Acknowledgements

I thank Ms. Curran, my STEM teacher, Ms. Wildfong, my STEM advisor, and my parents for they help they provided in this project.

Table of Contents

Abstract

In America, only 30 percent of calls made to emergency medical services (EMS) require medical attention, and in cases where EMS is not required or not available, others can help. Furthermore, the 9-1-1 service has indicated that it is not always able to reach people in emergencies due to factors such as overloaded call-centers or geographic distance. Other applications attempt to address this issue but can be impractical because panicked users are often unable to spend time tapping buttons on their phone during an emergency. EmerSave is an Android application developed with Android Studio and Firebase, which sends hands-free distress messages to contacts within a specified radius. The application allows the user to call for help manually or with voice recognition and can even be accessed when the phone is locked. Contacts within a specified radius receive a notification that the user needs help. With a tap on the notification, a Google Maps page is opened providing directions to the user. One of the unique features of this app, the radius-based system, was tested to determine its accuracy and had a 100% percent success rate within the messaging-radius.

EmerSave: A Novel Crowdsourcing Emergency Response Application

Introduction

With the exponential growth of human population and technology, more individuals access emergency medical services (EMS) for a plethora of reasons. In the early 1970s, 70% of EMS callers needed paramedic support, while today, only 35% of EMS callers request paramedic support (Fire Chief Contributor, 2013). Despite the decrease in paramedic requests, EMS is often inundated with calls; this leads to blocked telephone lines which would prevent others from accessing EMS. In most paramedic situations, it is imperative that a first-responder gets to the distressed individual within five minutes, or the chance of fatality dramatically increases (Blackwell, K., 2002). The role of EmerSave is to act as a supplement for EMS to reduce stress on the response system. This application has a wide range of potential users who can benefit from the application's unique voice-activated call system. Although many groups could benefit from the development of this app, including pedestrians, and individuals caught in natural disasters; the targeted demographic is the elderly. By acting as an alternative source for quick, non-life-threatening aid, EmerSave can help EMS focus on life-threatening scenarios while being a way for other individuals to access help—saving more lives and preventing a deadly catastrophe.

Target Audience

Elderly

A quick and efficient method of calling for help can aid many senior citizens. In a survey recently conducted by the Center for Disease Control (CDC) of 835,200 individuals in 2014, 21% of people living in residential care had fallen in the previous ninety days. In scenarios such as falls, it is imperative that a first-responder gets to elderly individuals as soon as possible, because delayed response time can result in their condition worsening.

There is a significant demand for an effective personal emergency response system (PERS). Currently, the percentage of people between the ages of 60 and 64 with a PERS is 18%. The percentage of people with a PERS is 38% for people between the ages of 75 and 79, and 46% for individuals between the ages of 80 and 84 (Link-age, 2013). These values are expected to increase rapidly, as demand for telehealth services, which are classified as healthcare through the use of phones, has significantly increased in recent years. In 2013, 350,000 people in the United States had bought some variation of a telehealth service, and by 2017, 3.84 million people had invested in PERS. The percentage of people with PERS in the United States is projected to almost double to 7 million by 2018 (HIS, n.d.). The rapidly increasing demand for PERS illustrates the need for a comprehensive and easy to use application to help the elderly access immediate assistance.

Global Positioning Systems (GPS)

GPS technology has revolutionized the methods people employ to provide location-based services (LBS) to consumers. GPS technology is contingent upon a network of satellites identifying and communicating the location of an electronic device based on proximity to the satellites (National Park Service, n.d.). As the scope of such technology increases, individuals are looking to extend the use of GPS and mobile technology. H. H. Lee is the individual credited with developing a system in which mobile location is traced by GPS. He exploited the capabilities of GPS and mobile networks to create a system in which the location of electronic devices can be precisely tracked (Roy, M. & Dutta, S. R., 2016). Over time, mobile tracking technology, like the line of sight technique, were developed. The line of sight technique provides the position of a mobile device by using a direct signal from a satellite to the mobile device; this is the most commonly exercised means of providing location among GPS providers (Fattepur, M. B. et al., 2016). This simplicity of this technique differentiates it from its adversaries such as the multipath

technique, which utilizes multiple signals distributed in different paths to identify the location of a mobile device (Fattepur, M. B. et al., 2016). While GPS technology does have minor flaws in rapid location identification, it is superior to other common methods such as the use of Wi-Fi and cellular radio signals and is the best outdoor positioning technique to date (Garzon, S. R., 2016). Because GPS is more reliable and efficient than its competitors, it is the best technique for a mobile phone providing LBS.

Geofencing

Geofencing is a technique that utilizes GPS technology to track the movement of objects in and out of a set virtual fence (Fattepur, M. B., 2016). A geofence can be configured into any shape, but the most common configuration is a circular geofence around a specified center (Roy, M., 2016). After the geofence is created, objects such as smartphones can keep using GPS, and moving in or out of the geofence can trigger an action by an associated app (Garzon, S. R., 2016). For example, a Walmart store might be attempting to let people passing by their store know that there is a sale on an item. By creating a geofence around that store, they can push a notification about that sale to appear to all the phones of Walmart app holders within a mile of the store. Although geofencing is a relatively new field, many colleges and third-party companies attempting to create geofences with irregular shapes make the process of creating a geofence more user-friendly (Garzon, S. R., 2016).

Current Uses

Geofencing is commonly associated with providing LBS for commercial uses, yet many applications created in recent years demonstrate the potential of geofencing in emergency situations. One example, created in Japan, gets information about active natural disasters and creates geofences around dangerous areas. Once an individual enters this geofence, they are

warned of the danger and are urged to exit the danger zone. Once the natural disaster is over, the geofence around the previously designated danger zone is disassembled. Through thorough testing, researchers have determined that this app does correctly detect the entrance into a geofence, but it is not accurate at detecting when a user leaves the geofence (Suyama, A., 2016). While this app has its flaws, it is still a revolutionary example of the multitude of uses for geofencing. An example of geofencing in the medical field is an app called Alzimio, which uses activity recognition and geofencing to help people with Dementia, Autism, and Alzheimer's. This app utilizes geofencing to set up a designated safe zone such as the boundaries of a house. Once patients leave the safe zone, they, and other selected contacts, are alerted to move the patient back into the safe zone (Helmy, J., & Helmy, A., 2016). These applications show examples of how geofencing can be used to assist people with personalized safety.

Why Geofences are Not Efficient

Although geofencing is the optimal technology to use for certain products, it has many flaws. While geofencing technology has improved significantly, one flaw that remains is its impact on battery life. Not only is it required for smartphones to constantly monitor geofences, but they must also constantly send data to a server, causing excessive battery drain (Garzon, S. R., 2016). Another major issue with geofencing is low accuracy. Many experiments and studies have shown that geofencing applications must choose between optimizing accuracy and battery life, and even if accuracy is optimized, geofencing technology is not developed enough to predict when a smartphone enters or exits the geofence without error (Garzon, S. R., 2016). Moreover, the complexity and the cost required to maintain a geofence can exceed the financial or computational resources an organization can provide (Garzon, S.R., 2016). Finally, even if one were to overlook the aforementioned problems, Android Studio only allows an app to create 20 geofences, which

limits the number of users who can benefit from this app (Android Studio, n.d.). Due to the many pitfalls associated with geofencing, a similar, simpler technique of calculating distance by the means of using latitude and longitude coordinates would be a better method of distance computation.

Voice Recognition

Unlike many other technologies in development, voice recognition has quickly integrated into many different sectors due to its versatility. One of the leading companies in the field of voice recognition is Google. The principal purpose of voice recognition is to generate a world in which all technology is voice-activated, reducing the direct interaction between humans and technology (Wesley, J. A., 1991). Voice recognition software primarily utilizes artificial intelligence (AI) and analytics to ascertain speech (Walker, C., 2017). By 2016, Google's voice recognition software could analyze over 90% of over 5 million words: an amount which is still expected to increase dramatically (Walker, C., 2017). Furthermore, many leading companies such as Google have also recently developed context awareness for voice recognition software, granting the software the ability to determine the context of a command based on location or previous commands (Walker, C., 2017). Despite the benefits of voice recognition, it still presents one significant flaw: security (Walker, C., 2017). Once the security issue is nullified, hardware equipped with voice recognition software will be the best mode of interaction between humans and smart-technology.

Mobile Phones

One of the premier platforms for software development is mobile technology, more specifically, smartphone technology. The pretext for this statement is the growing popularity of the smartphone ever since its inception: between 2016 and 2020, the number of smartphone users is expected to increase from 2.1 billion to 2.87 billion (eMarketer, n.d.). The amount of time

millennials spend on smartphones is also significant, as the average millennial spends approximately three hours and five minutes on his or her phone every day (GlobalWebIndex, n.d.). With a vast pool of smartphone users, it is the methodical choice to focus on smartphone technology as a main platform for mobile development.

Play Store and App Store

Applications developed for smartphones are often purchased on application stores which house apps compatible with the operating system of the smartphone. The two preeminent application stores are the Google Play Store and Apple's App Store. In 2016, the Play Store offered 2.8 million applications to download, while the App Store offered 2.2 million applications to download (Android et. al., 2017). Furthermore, users have a greater confidence in applications developed on these platforms, as this indicates that an application is credible. This credibility is derived from quality tests performed by Google and Apple. These platforms also permit users to download free, useful applications, assisting them in completing tasks sans expensive hardware.

Programs

The purpose of this research is to develop a comprehensive application to aid people in emergencies, so it is essential to develop it using libraries and preexisting software which will give it the best opportunity to succeed. The software described below was chosen based on four main criteria: popularity, ease of use, reliability, and cost.

Android Studio

As this app is expected to be released on is the Google Play Store, the platform used to develop the application should be one which supports Android applications. In order to develop an application for the Google Play Store, an application development platform must support android; the two most premier development programs to do so are Android Studio and MIT's App

Inventor 2, which are both incredibly popular and free to use. App Inventor 2 is easier to use, as it utilizes drag and drop block programming. However, App Inventor, due to its intended simplicity, does not have many of the advanced features that Android Studio offers. Due to its use of block programming, App Inventor does not allow the developer to view source code and requires more processing time as the block code must be converted to byte code and compiled by a third-party compiler. Complications caused by the compiler include the application resulting in a much larger application size than an identical app written with the same number of lines of code in Android Studio. Another reason why applications developed in Android Studio are better than their counterparts developed in App Inventor 2 is that they offer better intra-app communication. One of the major setbacks of utilizing App Inventor 2 is that it only can communicate with other applications developed with App Inventor 2, whereas Android Studio allows communication with apps developed from various platforms. Moreover, the transmission time for a message between apps created with Android Studio was found to be much faster than the time for a message to travel between two apps created with App Inventor 2. It is due to Android Studio's many advanced features to aid in app development and its ability to develop efficient intraplatform and intra-operating-system applications that Android Studio is considered to be the best platform to develop an android application (Allison, L. & Fuad, M. M., 2016).

Firebase

Firebase is a Google mobile platform that of establishes an all-in-one platform for application developers to develop, test, analyze, and distribute their applications. Firebase also provides a real-time database for effortless integration with popular development platforms including iOS, Android, Unity, and the Web. Furthermore, the database supports development by multiple individuals with the aid of its valuable synchronization feature. Firebase also possesses

the ability to offer various external services on account of its development by Google; these services comprise of, but are not restricted to, cloud storage, ad-mob, analytics, and virtual devices. Security features such as user authentication are accorded as well.

Firebase functions rudimentarily to facilitate application developers of varying experience. The platform provides the intrinsic code required to integrate Firebase into a mobile application and allows the developer to install the files required to successfully run Firebase on mobile platforms. The data stored in the real-time database is stored in JavaScript Object Notation (JSON) format which allows maximized readability in a methodical format. It is due to its multitude of available tools and its effortless integration that Firebase is the most appropriate database for an emergency response mobile application (Irani, R., 2017).

Existing Systems

EMS

Existing products developed to fulfill a niche similar to those fulfilled by an application containing the aforementioned features have various issues. The purpose of this system is to function as a complement to emergency medical services (EMS). Since its introduction, emergency medical services have been an integral facet of public and private health services around the world. In the United States alone, approximately 240 million calls are made to 9-1-1 every year, with 80% of these calls made with the aid of a mobile device (NENA, 2017). Although 9-1-1 proves to be an effective method of delivering emergency services, there are still flaws which can hinder its effectiveness. One area where issues may arise are in emergency situations with the impaired or elderly. For instance, individuals who are deaf and hard of hearing require a teletypewriter to communicate with 9-1-1; these devices are arduous to operate and result in a prolonged communication time between emergency medical services and the endangered

individual. The area in which 9-1-1 is activated factors into the performance of the service as well. Large cities receive more than ten million emergency calls per year, and this volume often results in the overload of the call centers. Issues also persist with rural areas, as the extended distances between call centers results in a prolonged response time (911.gov, n.d.). In many of these scenarios, it would be beneficial to have a system which would take some of the stress off the 9-1-1 system; it could be utilized to assist individuals in non-life-threatening situations, provide access to quicker help in rural areas, and provide another source of aid in the events of a 9-1-1 system overload or a natural disaster.

Competitor Analysis

There are various existing products which are meant to fulfill similar goals that are desired from the mobile application, but these products have associated issues. One of the main issues for hardware technology targeted towards emergency aid is that it will be a manufactured product, ensuing in high prices. For example, many hardware devices solely created with the objective of GPS tracking cost, on average, between eight and forty-two dollars (Helmy, J., & Helmy, A., 2016). On the contrary, smartphone applications do not require hardware, resulting in minimal costs for consumers.

Current application options to fail satisfy certain criteria. Table 1 illustrates the capabilities and pitfalls of several competitors.

Table 1

Competitor Applications					
Criteria	SoS Emergency "GPS Bodyguard"	SoS Emergency App	5Star Urgent Response with GPS	Emergency Response App	mySos SA
Ability to send messages					
Sending contacts information					
GPS Directions					
Specific Radius based system					

Competitor Applications					
Criteria	SoS Emergency "GPS Bodyguard"	SoS Emergency App	5Star Urgent Response with GPS	Emergency Response App	mySos SA
Overall Review					
Server					
Access while locked					
Voice Activation					

Note: If a cell is filled in green, it means the application offers the corresponding criteria. If a cell is filled in red, it means that the corresponding criteria is not satisfied.

Project Charter

Engineering Problem

Many people in the world who cannot access help in emergency situations, even if they have access to a cell phone.

Engineering Goal

The goal of this project is to engineer an application which can deliver timely and low-cost help in emergency situations using location-based services.

Design Criteria

1. App should have ability to send a message
2. App should have ability to send a message to specific groups
3. App should be able to offer responders directions to the individual in need of help
4. App should be able to send a message only to people within a specified radius
5. App should have a database that it utilizes to efficiently store information
6. App should be able to be utilized hands-free
7. App should be able to be accessed by users even when phone is locked

Methodology

Development

First, an empty application was created and uploaded to an android device to ensure that device could load applications made in Android Studio. Next, a button which displayed a popup when clicked was added. After that, code was added to request the user's location in the Manifest file and the main java file. A button was then added which displayed the user's location, in longitude and latitude when clicked. Next, a new java file and a corresponding XML file were created with the purpose of being an information posting page. A button was added to transition to this page from the main page. Then, code was added to the gradle file and the java file to connect the application to Firebase. To test connectivity, code was added in the java file which would put a message in the database when the app is launched. In addition, code to delete a message in the database was added and tested. After that, a listener was created in the code to detect changes in the database and display a popup when information was added. Code was then added to take new information added to the database and display it as a popup to test the applications ability to get information from the database. Then in the application, a button was added to send a distress request, which included the user's name, longitude and latitude, message, and message expiration time, which was set to two hours after the distress signal was called. In correspondence, a button to delete this request was created as well. These were both tested on the android device.

Next, in order to perform radius-based operations, the Haversine Formula was coded into the application so that distance between two sets of longitude and latitude points could be computed. This was then tested at different ranges to see if accurate accounts of distance were provided by the application. After that, a notification system was built. To accomplish this, Android Studio's notification builder class was utilized, and to test this, a simple notification was

displayed to the user in the notification bar when the app was launched. Then, a main listener which would look for help requests was set up in the application. This would check that it was not the same user who submitted the request, the person who sent the request was within 10 km, and that request was not an expired request. If all three of these requirements were met, the application would create a notification to the user with the name of the person who needs help and a message. These three requirements were tested in various configurations to ensure that all three were properly functioning. After that, code was added to the notification popup that would enable the user to open a Google Maps page displaying directions from their location to the location of the individual who needs help when the notification was clicked. Then, a button was added to the application which would enable the use of voice commands, and when the spoken command had the word "help," the help request would be triggered. If it contained "cancel," the help request would be canceled. A service was then created to enable the application to receive information in the background and display notifications to the user even if the application is not open.

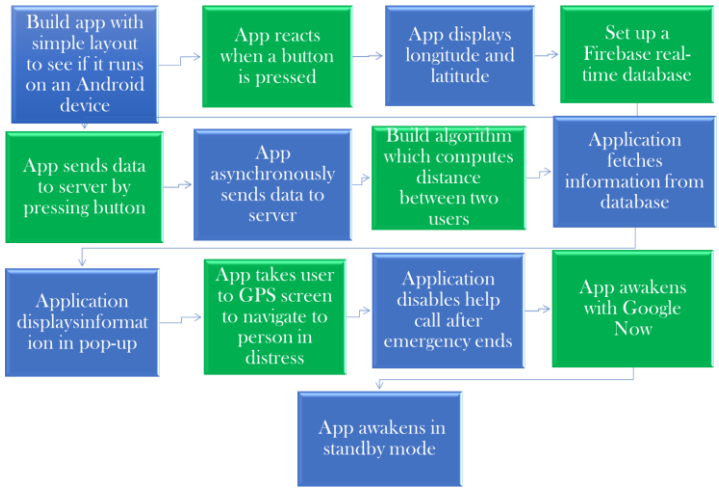

Figure 1: A flow chart summarizing the development of the application

Testing

First, the application was downloaded onto two android devices which were at least running Android Jelly Bean and had 100% charge. Next, both devices had internet and location settings enabled and had preferred radius set to 5 km. One phone, called Tester 1, was kept in a stationary position with one person. The other phone, labeled Tester 2, was held by another individual 4 km South of Tester 1. Then, the person with Tester 1 sent a distress signal. A check was recorded if the person with Tester 2 received the message and an "x" was recorded if the person with Tester 2 did not receive the message. This process was repeated four times and then that was repeated four times North, East, and West as well. Then all the previous steps were repeated for distances of 4.5 km, 5.5 km, and 6 km.

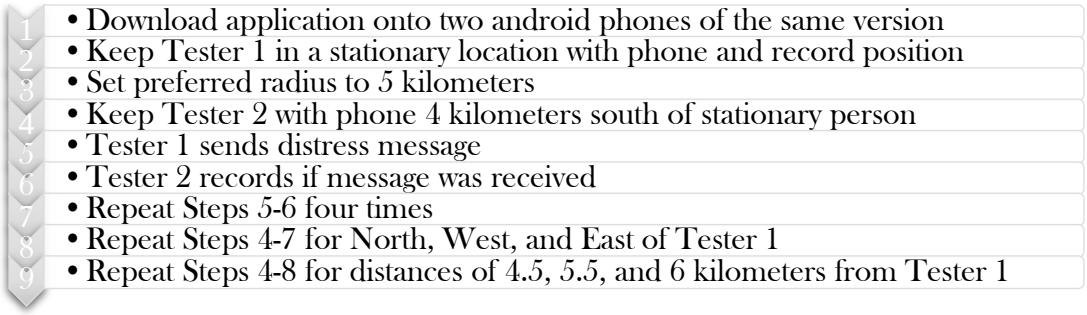

Figure 2: A flowchart detailing the testing procedure for the radius-based system

Results

After the completion of the final version of the application, several analyses were completed to determine the success of the application. This includes a qualitative assessment of the user interface, an assessment of the completion of the design criteria, and a quantitative assessment of the radius-based notification system.

User Interface

Throughout the development of this application, there were many changes made to the user interface in conjunction with the many changes made to the back-end code. Figure 3 displays the initial functional user interface created; this was developed in Version 1 of the application.

Figure 3: An image of the user interface of the original application which was named EmerGeo

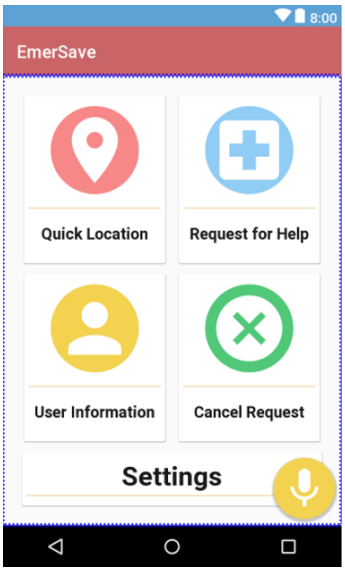

Figure 4: An image of latest user-interface of the application (Version 9)

After the development of the original application, several other additions to the user interface were made based on the needs of the application and feedback provided by peers and

mentors. After all these changes were considered and the interface was modified for aesthetic purposes, the final interface looked like Figure 4, which is the most up-to-date version of the user interface.

Feature Matrices

Table 2

Feature Matrix of First Application							
Features	Weight	SoS Emergency "GPS Bodyguard"	SoS Emergency App	5Star Urgent Response with GPS	mySOS SA	Emergency Response App	Version 4
Message Sending	5	Yes	Yes	Yes	Yes	Yes	Yes
Messaging to Contacts	3	Yes	Yes	No	Yes	No	No
GPS Directions	3	Yes	No	No	Yes	No	Yes
Specified Radius	5	No	No	No	No	No	No
Hands-Free	4	No	No	No	No	No	No
Database	3	Yes	Yes	No	Yes	Yes	Yes
Access while locked	3	No	No	No	No	No	No
	Totals	14	11	5	14	8	11

Note: *Version 4 was used to compare to the competitor applications as this was the newest version of the application at the time of the first comparison. The weighting scale is from 1-5, with 5 indicating the highest importance.*

Table 3

Feature Matrix of Newest Application							
Features	Weight	SoS Emergency "GPS Bodyguard"	SoS Emergency App	5Star Urgent Response with GPS	mySOS SA	Emergency Response App	Version 9
Message Sending	5	Yes	Yes	Yes	Yes	Yes	Yes
Messaging to Contacts	3	Yes	Yes	No	Yes	No	Yes
GPS Directions	3	Yes	No	No	Yes	No	Yes
Specified Radius	5	No	No	No	No	No	Yes
Hands-Free	4	No	No	No	No	No	Yes
Server	3	Yes	Yes	No	Yes	Yes	Yes
Access while locked	3	No	No	No	No	No	Yes
	Totals	14	11	5	14	8	26

Note: *Version 9 was used to compare to the competitor applications as this was the newest version of the application at the time of the comparison. The weighting scale is from 1-5, with 5 indicating the highest importance.*

As shown in Table 2, Version 4 of the application only fulfilled 3 out of the 7 design criteria which were target features of this applications, earning it a score of 11. Its best competitors, mySOS SA and SoS Emergency "GPS Bodyguard," fulfilled 4 out of the 7 design criteria, albeit the criteria fulfilled by these applications were more important to accomplish, as indicated by the weighting system, than those accomplished by Version 4 of the application, leading to a score of

14. As shown in Table 3, the newest version of the application (Version 9) fulfilled all 7 of the 7 design criteria and earned the maximum score of 26 in the feature matrix.

Notification System

The application was tested to determine the accuracy of the radius-based emergency notification system. 40 trials were performed within the set radius of 5 kilometers, and within this radius, the notification system had an accuracy of 100% at delivering notifications. 40 trials were also performed outside of the 5-kilometer radius. Outside of the set radius, the application had a 95% accuracy of not delivering the notification, and unexpectedly delivered notifications 2 times at a radius of 5.5 kilometers. Table 4 shows the different settings in which the notification system was tested and the results.

Table 4

Testing of Notification System with Radius of 5 Kilometers						
Distance from Tester 1 (km)	Direction Relative to Tester 1	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
4.0	South					
4.0	North					
4.0	West					
4.0	East					
4.5	South					
4.5	North					
4.5	West					
4.5	East					

5.5	South					
5.5	North					
5.5	West					
5.5	East					
6.0	South					
6.0	North					
6.0	West					
6.0	East					

Note: *The green boxes represent a trial which was a success at performing in the expected manner: radius<5 did send message, radius>5 did not send message. The red boxes represent trials which did not perform in the expected manner: radius<5 did not send message, radius>5 did send message.*

Conclusions

This application fulfilled the engineering goal of delivering timely and low-cost help in emergency situations and was also able to successfully meet all the design criteria. It was able to meet the goal of delivering timely help with its radius-based notification system and its easy-to-navigate user interface. The app was also able to meet the goal of providing a low-cost source of help by not using any proprietary software which costed money, which in turn allowed the app to not cost any money as well.

User Interface

After the creation of the initial user interface, it was concluded that it was functional but lacked a professional look and an intuitive layout which would make the app easily accessible to people of many different demographics. To address these issues, many professional apps were studied to determine common themes for professional apps such as icons and a uniform color palette, and these themes were applied to EmerSave. Another change made to create an intuitive layout was the removal of all the buttons from the home screen which were not integral to the

functionality of the application, as this allowed the app to not look too crowded and gave it a cleaner look. After these changes were made, a few more changes were made to the user interface to enable it to be uniform throughout a variety of android devices and provide optimal use to all android users.

Competitor Analysis

While evaluating the success of each version of the application, weights were given to the design criteria outlined in the project charter on a scale of 1-5, with five being the most important, based on their importance to the functionality of the application as well as the level of uniqueness it would provide to the application. Initially, the first comparable version of the application, Version 4, scored an 11 on the weighted comparison system, which tied it with the SoS Emergency App for the third highest score. Although this score itself was satisfactory, the main target of this application was to fulfill all the design criteria, so the rest of the design criteria were addressed, with the ones weighted heavier addressed first. After several more months of developing the application, the newest version, Version 9, earned the maximum score of 26 on the comparison system, which is almost twice the previous top score of 14 and is a 15-point improvement from Version 4. The application was able to get this score by successfully implementing the heavily weighted section of having a radius-based notification system as well as a hands-free method of access. A broken-down analysis of the different versions and the design criteria they fulfilled can be found in Appendix D.

Notification System

The notification system proved to be very successful with a 100% success rate within the specified radius of 5 kilometers and a 95% success rate outside of this radius. To be considered a success within the radius, the app had to successfully send a notification to the other phone, and to

be successful outside of the radius, the app had to not send a notification to the other phone. Overall, forty trials were conducted within the specified radius and forty were conducted outside of the radius, with the closest distance to the radius being 0.5 kilometers to ensure that the distance computing algorithm's rounding would not dramatically affect the accuracy of the application. A possible reason for the two non-successful trials outside of the radius could be that the application did not update its location properly after leaving the 5-kilometer radius, which caused the application to believe that it was within the radius when it really was not. Another possible source of error could be an error in the rounding of the distance. This is believed to be a factor because both errors occurred at a radius of 5.5 kilometers, which is close to the set radius of 5 kilometers, meaning that application could have accidently rounded incorrectly. While both these errors do show potential flaws in the application, they would both lead to the type one error of someone outside of the designated radius getting a distress call, which is preferred over the type two error where someone within the radius does not get a notification that he or she should get.

Future Extensions

Future extensions include adding more options for the user in the settings page to aid in making the application more personalized. Another major future extension is to release this application on the Google Play Store. To do this, this application must be modified to accommodate many users, which it did not experience during testing. Additionally, an efficient way to update the application must be developed to provide seamless updates to users. A potential obstacle in the release of this application could the cost to store all the distress messages in the Firebase Database. To address this cost, organizations such as colleges and elderly care centers could be approached to become sponsors to this application which would be beneficial to them.

References

911.gov.Ensuring access. Retrieved from https://www.911.gov/ensuringaccess.html

Suyama, A. & Inoue, U. (2016). Using geofencing for a disaster information system. Paper presented at the *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS),* 1-5. doi:10.1109/ICIS.2016.7550849

Android, & Apple, & Google, & Microsoft, & AppBrain, & BlackBerry, & Various sources (WindowsCentral.com, & International Games Week Berlin), & Amazon, & VentureBeat, & CNET. Number of apps available in leading app stores as of March 2017. Retrieved from https://www.statista.com.ezproxy.wpi.edu/statistics/276623/number-of-apps-available-in-leading-app-stores/

Blackwell, T. H., & Kaufman, J. S. (2002). Response time effectiveness: Comparison of response time and survival in an urban emergency medical services system. *Academic Emergency Medicine, 9*(4), 288-295. doi:10.1197/aemj.9.4.288

eMarketer. Mobile phone internet user penetration worldwide from 2014 to 2019 . Retrieved from https://www.statista.com.ezproxy.wpi.edu/statistics/284202/mobile-phone-internet-user-penetration-worldwide/

FIRE CHIEF Contributor. (2013). Matching EMS response times to patient needs. *Fire Chief (Online Exclusive),* Retrieved from https://search.proquest.com/docview/1412452486

GlobalWebIndex. Average daily time spent online via mobile by internet users worldwide as of 2nd quarter 2016, by generation (in minutes) . Retrieved from https://www.statista.com.ezproxy.wpi.edu/statistics/428425/daily-time-spent-online-mobile-age/

Irani, R. (2017, Developers - let's try: Power your mobile applications with firebase.

Helmy, J. & Helmy, A. (2016). The alzimio app for dementia, autism & alzheimer's: Using novel activity recognition algorithms and geofencing. Paper presented at the *2016 IEEE International Conference on Smart Computing (SMARTCOMP),* 1-6. doi:10.1109/SMARTCOMP.2016.7501720

Allison, L. & Fuad, M. M. (2016). Inter-app communication between android apps developed in app-inventor and android studio. Paper presented at the *2016 IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILESoft),* 17-18. doi:10.1109/MobileSoft.2016.018

Fattepur, M. B., Sharvani, G. S., & Huttanagoudar, J. B. (2016). A solution to improve the performance of geofence enabled GNSS chipset. *2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS),* 112-116. doi:10.1109/CSITSS.2016.7779405

mySOS. (2017). mySOS SA (Version 3.1.6) [Mobile application software]. Retrieved from https://play.google.com/store/apps/details?id=za.co.widge.sos&hl=en

National Park Service. The history of GPS. Retrieved from https://www.nps.gov/gis/gps/history.html

NENA. (2017). 9-1-1 statistics
. Retrieved from https://www.nena.org/?page=911Statistics

Primer Analytics & Systems, Inc. (2016). Emergency Response App (Version 1.1) [Mobile application software]. Retrieved from https://play.google.com/store/apps/details?id=com.era.user&hl=en

Dutta, S. R. & Roy, M. (2016). Providing context-aware healthcare services using circular geofencing technique. Paper presented at the *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom),* 446-451.

Garzon, S. R., Elbehery, M., Deva, B., & Kpper, A. (2016). Reliable geofencing: Assisted configuration of proactive location-based services. Paper presented at the *2016 IEEE International Conference on Mobile Services (MS),* 204-207. doi:10.1109/MobServ.2016.42

Walker, C. (2017, Communications coming full circle--are we moving back into a voice-first world?*48*, 6+. Retrieved from http://libraries.state.ma.us/login?gwurl=http://go.galegroup.com/ps/i.do?p=ITOF&sw=w&u=mlin_c_worpoly&v=2.1&it=r&id=GALE%7CA509727960&sid=summon&asid=b962615170e84226e6a4f165e67d8feb

Wesley, J. A. (1991, The service threshold.*12*, 32+. Retrieved from http://libraries.state.ma.us/login?gwurl=http://go.galegroup.com/ps/i.do?p=ITOF&sw=w&u=mlin_c_worpoly&v=2.1&it=r&id=GALE%7CA10969922&asid=99e29851cde2edab627d82d7c08815bd

Appendix

Appendix A: Limitations and Assumptions

1.) There was a cost limitation as no software which costed money was utilized.

2.) There was a limitation on the development platform, as there was only access to a Windows computer and there was no access to a platform to efficiently develop iOS software.

3.) There was a limitation of knowledge as there was no experience with Object-C or Swift, so an iOS application could not be created.

4.) There was a limitation on time as there was a deadline on the development of the application.

5.) There was a limitation on software as some software was proprietary and was not available for public use.

6.) An assumption was made that the elderly would be willing to use my application to their advantage.

7.) An assumption was made that the elderly would have a basic knowledge of how to use mobile technology.

8.) An assumption was made that the Firebase Database will not crash at any point and jeopardize the data in the table.

9.) An assumption was made that the users will not try to compromise the functionality of the application with malicious intents.

10.) An assumption was made that the application would not crash unexpectedly.

11.) An assumption was made that the user has a recent version of android so that the features of the application will be useable.

12.) An assumption was made that the user will have a large enough scree to allow for the proper rendering of the user interface.

Appendix B: Overview of App's Functions and User Interface

Figure 5: An image of an infographic made in Piktochart which details the app's features

Appendix C: Overview of Application Use Process

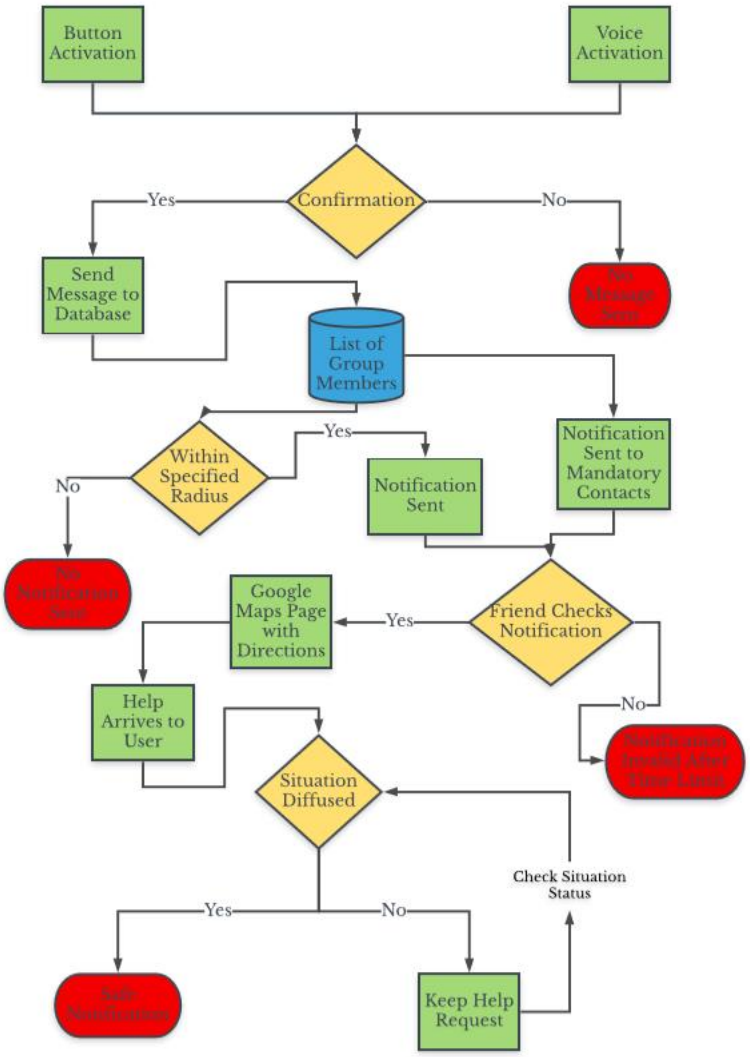

Figure 5: An image of a flowchart made in Ludichart which details the processes involved with the use of the application.

Appendix D: Versions of Application

Table 5

Features	Version 1	Version 2	Version 3	Version 4	Version 5	Version 6	Version 7	Version 8	Version 9
Message Sending	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Messaging to Contacts	No	No	No	No	No	Yes	Yes	Yes	Yes
GPS Directions	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Specified Radius	No	No	No	No	Yes	Yes	Yes	Yes	Yes
Hands-Free	No	No	No	No	No	No	Yes	Yes	Yes
Server	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Access while locked	No	No	No	No	No	No	No	No	Yes

Note: *A green box indicates that the application contains that feature while a red box indicates that the application did not contain that feature. Versions 7 and 8 did have minor changes in-between them but these changes were not part of the design criteria*

Appendix E: Code

This section contains all the pertinent code which was used in the application. Code which is commented out is code that was important for testing the features of the application but not to the actual functionality of the application.

Android Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.vishnu.emersave">

  <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
  <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
  <uses-permission android:name="android.permission.VIBRATE" />

  <!--
     The ACCESS_COARSE/FINE_LOCATION permissions are not required to use
     Google Maps Android API v2, but you must specify either coarse or fine
     location permissions for the 'MyLocation' functionality.
  -->

  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

  <application
    android:name=".MyApplication"
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity
      android:name=".MainActivity"
      android:label="@string/app_name"
      android:theme="@style/AppTheme.NoActionBar" />

    <!--
       The API key for Google Maps-based APIs is defined as a string resource.
       (See the file "res/values/google_maps_api.xml").
       Note that the API key is linked to the encryption key used to sign the APK.
       You need a different API key for each encryption key, including the release key that is
```

```
used to
        sign the APK for publishing.
        You can define the keys for the debug and release targets in src/debug/ and src/release/.
    -->
    <meta-data
      android:name="com.google.android.geo.API_KEY"
      android:value="@string/google_maps_key" />

    <activity
      android:name=".MapsActivity"
      android:label="@string/title_activity_maps" />
    <activity android:name=".PostingPage" />
    <activity
      android:name=".Home"
      android:screenOrientation="portrait" />
    <activity android:name=".JustTest" />

    <!--
      <receiver
      android:name=".MyAlarm"
      android:enabled="true"
      android:exported="true" />
    -->

    <service
      android:name=".MyService"
      android:enabled="true"
      android:exported="true" />

    <activity android:name=".SignIn">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />

        <!--
        <action android:name="com.google.android.gms.actions.SEARCH_ACTION" />
        <category android:name="android.intent.category.DEFAULT" />
        -->

      </intent-filter>
    </activity>
    <activity android:name=".SettingsPage" />
  </application>

</manifest>
```

Home.java File (Main Activity)

```
package com.vishnu.emersave;

import android.app.AlarmManager;
import android.location.LocationListener;
import android.os.Build;
import android.os.SystemClock;
import android.support.annotation.NonNull;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.Manifest;
import android.app.AlertDialog;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.app.SearchManager;
import android.content.ActivityNotFoundException;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationManager;
import android.net.Uri;
import android.os.Bundle;
import android.os.Handler;
import android.os.Vibrator;
import android.provider.Settings;
import android.speech.RecognizerIntent;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.NotificationCompat;
import android.view.View;
import android.support.design.widget.NavigationView;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
```

```
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.Query;
import com.google.firebase.database.Transaction;
import com.google.firebase.database.ValueEventListener;

import java.lang.reflect.Method;
import java.time.LocalDateTime;
import java.util.ArrayList;
import java.util.Date;
import java.util.Locale;

import com.google.android.gms.actions.SearchIntents;

import static com.vishnu.emersave.PostingPage.namef;

public class Home extends AppCompatActivity {

    private static final int REQUEST_LOCATION = 1;
    static Button button;
    static Button postToServerButton;
    //static LocationManager locationManager;
    static String Latitudes, Longitudes;
    //String Name;
    double latitude;
    double longitude;
    String id;
    String testyes;
    final Handler handler = new Handler();
    private TextView seeSpeak;
    Date date = new Date();
    FirebaseDatabase database = FirebaseDatabase.getInstance();
    static int moose = 2;
    long timey = 7000;
    ArrayList<String> ids = new ArrayList<String>();
    int mNotificationId = 1;
    int mNotificationSafe = 1;
    int mtNotificationSafe = 1;
    String previous = "";
    long[] vibr = {200, 200, 200};
    String message;
    String m_androidId;
```

```
    String lon;
    String lat;
    String latVishnu;
    String lonVishnu;
    String latFriend;
    String lonFriend;
    private FirebaseAuth mAuth;
    private FirebaseAuth.AuthStateListener mAuthListener;
    private LocationManager locationManager;
    private LocationListener locationListener;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);
        mAuth = FirebaseAuth.getInstance();

        mAuthListener = new FirebaseAuth.AuthStateListener() {
            @Override
            public void onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) {
                if (firebaseAuth.getCurrentUser() == null) {
                    startActivity(new Intent(Home.this, SignIn.class));
                }
            }
        };

        Intent intent = getIntent();
        DatabaseReference root = database.getReference("messages");
        if (SearchIntents.ACTION_SEARCH.equals(intent.getAction())) {
            String query = intent.getStringExtra(SearchManager.QUERY);
            doSearch(query);
        }


        if(Build.VERSION.SDK_INT>=23 &&
ContextCompat.checkSelfPermission(this,Manifest.permission.ACCESS_FINE_LOCATION)
                != PackageManager.PERMISSION_GRANTED &&
ContextCompat.checkSelfPermission(this,Manifest.permission.ACCESS_COARSE_LOCATIO
N)
                != PackageManager.PERMISSION_GRANTED){
            requestPermissions(new
String[]{Manifest.permission.ACCESS_FINE_LOCATION,Manifest.permission.ACCESS_CO
ARSE_LOCATION},100);

        }
```

```
        startService(new Intent(this, MyService.class));



        Thread.setDefaultUncaughtExceptionHandler(new MyExceptionHandler(this));
        if(getIntent().getBooleanExtra("crash",false))
        {

        }
        promptSpeechInput();
    }



    @Override
    public void onStart()
    {
        super.onStart();
        mAuth.addAuthStateListener(mAuthListener);
    }

    private void setAlarm() {
        AlarmManager alarmManager = (AlarmManager)
getSystemService(Context.ALARM_SERVICE);
        Intent intent = new Intent(this, MyAlarm.class);
        PendingIntent pendingIntent = PendingIntent.getBroadcast(this,0,intent,0);
      // alarmManager.setRepeating(AlarmManager.RTC_WAKEUP,
System.currentTimeMillis()+1000,3000,pendingIntent);
        alarmManager.setRepeating(AlarmManager.RTC_WAKEUP,System.currentTimeMillis(),
60000, pendingIntent);
       /* AlarmManager mgr=(AlarmManager)getSystemService(Context.ALARM_SERVICE);
        Intent i=new Intent(this, MyAlarm.class);
        PendingIntent pi=PendingIntent.getBroadcast(this, 0, i, 0);

        mgr.setRepeating(AlarmManager.ELAPSED_REALTIME_WAKEUP,
SystemClock.elapsedRealtime(), 6000,pi);*/
    }

    public void click(View v)
    {
        if(v.getId()==R.id.quick_location)
        {

            Vibrator z = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);
            z.vibrate(400);
            Toast.makeText(this, "Your Current Location:" + "\n" + "Latitude = " +
```

```
MyService.latitude
                + "\n" + "Longitude = " + MyService.longitude,
Toast.LENGTH_LONG).show();


    }
    if(v.getId()==R.id.help_request)
    {
       DatabaseReference  distress = database.getReference("messages").child(getId());

       distress.child("message").setValue("I need help");
       distress.child("latitude").setValue(MyService.latitude);
       distress.child("longitude").setValue(MyService.longitude);
      distress.child("Name").setValue(namef);
       Date date = new Date();
       distress.child("expires").setValue(date.getTime()+7200000);
       Toast.makeText(this,"Help Request has been Sent",Toast.LENGTH_LONG).show();
    }
    if(v.getId()==R.id.user_info)
    {
       Intent i = new Intent(
             Home.this,
             PostingPage.class);
       startActivity(i);
    }
    if(v.getId()==R.id.cancel)
    {
       DatabaseReference safe = database.getReference("messages").child(getId());
       safe.removeValue();
       Toast.makeText(this,"Request Cancelled",Toast.LENGTH_SHORT).show();
    }
    /*if(v.getId()==R.id.settings)
    {
       Toast.makeText(this,"Settings not available yet",Toast.LENGTH_SHORT).show();
    }*/
    if(v.getId()==R.id.floatingActionButton)
    {
       promptSpeechInput();
    }
    if(v.getId()==R.id.settings)
    {
       mAuth.signOut();
    }
 }

 public void notif(String Name, String latty, String longy) {
```

```
        Intent notificationIntent = new Intent(Intent.ACTION_VIEW);
        notificationIntent.setData(Uri.parse("http://maps.google.com/?q="+latty+","+longy));
        PendingIntent pi = PendingIntent.getActivity(this, 0, notificationIntent, 0);


        NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(this);

mBuilder.setSmallIcon(R.drawable.common_google_signin_btn_icon_dark).setContentTitle(Na
me + " needs help!").setContentText("Yeets").setContentIntent(pi);


        // Gets an instance of the NotificationManager service
        NotificationManager mNotifyMgr =
                (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
// Builds the notification and issues it.
        mNotifyMgr.notify(mNotificationId, mBuilder.build());
        mNotificationId++;
        //Can change value below to decide number of notifications on whim
        if(mNotificationId>20)
        {
            mNotificationId = 1;
        }

    }

    public void doSearch(String query)
    {
        if(query.contains("help"))
        {

            Toast.makeText(this,"This worked",Toast.LENGTH_LONG).show();
            Intent a = new Intent(
                    this,
                    Home.class);
            startActivity(a);
        }
    }

    public void safey(String Name)
    {
        NotificationCompat.Builder mBuildery = new NotificationCompat.Builder(this);

mBuildery.setSmallIcon(R.drawable.common_google_signin_btn_icon_dark).setContentTitle(
Name + " is Safe").setContentText("Your friend "+Name+" is out of danger. You should
```

```
still check on them to verify safety");


        // Gets an instance of the NotificationManager service
        NotificationManager mNotifyMgry =
                (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
// Builds the notification and issues it.
        mNotifyMgry.notify(mNotificationSafe, mBuildery.build());
        mNotificationSafe++;
        //Can change value below to decide number of notifications on whim
        if(mNotificationSafe>20)
        {
           mNotificationSafe = 1;
        }
    }
    AlarmManager alarmMgr;
    PendingIntent pendingIntent;

    public void startAlarmManager()
    {
        Intent dialogIntent = new Intent(getBaseContext(), MyAlarm.class);

        alarmMgr = (AlarmManager) this.getSystemService(Context.ALARM_SERVICE);
        pendingIntent = PendingIntent.getBroadcast(this, 0,
dialogIntent,PendingIntent.FLAG_CANCEL_CURRENT);


alarmMgr.setInexactRepeating(AlarmManager.RTC_WAKEUP,System.currentTimeMillis(),
10000, pendingIntent);

    }

    public String getId()
    {
        try {

            m_androidId = Settings.Secure.getString(getContentResolver(),
Settings.Secure.ANDROID_ID);

        }
        catch (Exception e)
        {
            System.out.println(e.getMessage());
        }
        return m_androidId;
    }
```

```
    public void notiftest(Context context)
    {
        NotificationCompat.Builder mtBuildery = new NotificationCompat.Builder(context);

mtBuildery.setSmallIcon(R.drawable.common_google_signin_btn_icon_dark).setContentTitle(
" is Safe").setContentText("Your friend is out of danger. You should still check on them to
verify safety");


        // Gets an instance of the NotificationManager service
        NotificationManager mtNotifyMgry =
                (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
// Builds the notification and issues it.
        mtNotifyMgry.notify(mtNotificationSafe, mtBuildery.build());
        mtNotificationSafe++;
        //Can change value below to decide number of notifications on whim
        if(mtNotificationSafe>20)
        {
            mtNotificationSafe = 1;
        }
    }
    public void promptSpeechInput()
    {
        Intent i = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
        i.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
        i.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault());
        i.putExtra(RecognizerIntent.EXTRA_PROMPT, "Hi "
+SignIn.account.getDisplayName());

        try {
            startActivityForResult(i, 100);
        }
        catch (ActivityNotFoundException a){
            Toast.makeText(this, "Sorry does not support your language",
Toast.LENGTH_SHORT).show();
        }
    }

    public void onActivityResult(int request_code, int result_code, Intent i)
    {
        super.onActivityResult(request_code,result_code,i);
        switch (request_code)
        {
            case 100: if(result_code== RESULT_OK && i!=null)
            {
```

```
            ArrayList<String> result =
i.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
            if(result.get(0).toLowerCase().contains("help"))
            {

               DatabaseReference  distress = database.getReference("messages").child(getId());

               distress.child("message").setValue("I need help");
               distress.child("latitude").setValue(latitude);
               distress.child("longitude").setValue(longitude);
               distress.child("Name").setValue(namef);
               Date date = new Date();
               distress.child("expires").setValue(date.getTime()+7200000);
               Toast.makeText(this,"Help Request has been
Sent",Toast.LENGTH_LONG).show();
            }
            else if(result.get(0).toLowerCase().contains("ok"))
            {
               DatabaseReference safe = database.getReference("messages").child(getId());
               safe.removeValue();
               Toast.makeText(this,"Request Cancelled",Toast.LENGTH_SHORT).show();
            }
            else
            {
               Toast.makeText(this,"Please say HELP if you need help, and OK if you wish to
cancel HELP request",Toast.LENGTH_LONG).show();
            }
         }
            break;
      }
   }

   public void getInfo()
   {
      DatabaseReference lats = database.getReferenceFromUrl("https://emersave-
2d464.firebaseio.com/Vishnu/Latitude");
      lats.addValueEventListener(new ValueEventListener() {
         @Override
         public void onDataChange(DataSnapshot dataSnapshot) {
            latVishnu = dataSnapshot.getValue(String.class);

         }

         @Override
         public void onCancelled(DatabaseError databaseError) {
```

```
        }
    });
    DatabaseReference lons = database.getReferenceFromUrl("https://emersave-
2d464.firebaseio.com/Vishnu/Longitude");
    lons.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            lonVishnu = dataSnapshot.getValue(String.class);

        }

        @Override
        public void onCancelled(DatabaseError databaseError) {

        }
    });
    DatabaseReference lonsF = database.getReferenceFromUrl("https://emersave-
2d464.firebaseio.com/Friend/Latitude");
    lonsF.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            latFriend = dataSnapshot.getValue(String.class);

        }

        @Override
        public void onCancelled(DatabaseError databaseError) {

        }
    });
    DatabaseReference latsF = database.getReferenceFromUrl("https://emersave-
2d464.firebaseio.com/Friend/Longitude");
    latsF.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            lonFriend = dataSnapshot.getValue(String.class);

        }

        @Override
        public void onCancelled(DatabaseError databaseError) {

        }
    });

Toast.makeText(this,String.valueOf(latVishnu+lonFriend),Toast.LENGTH_SHORT).show();
```

```
    }

    public void display()
    {
        DatabaseReference Longitude = database.getReference(getId()).child("Longitude");
        DatabaseReference Latitude = database.getReference(getId()).child("Latitude");
        Longitude.removeValue();
        Latitude.removeValue();
        Longitude.setValue(this.longitude);
        Latitude.setValue(this.latitude);
        DatabaseReference Message = database.getReference(getId()).child("Message");
        Message.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                message = dataSnapshot.getValue(String.class);
                if(!message.equals(previous)) {
                    NotificationCompat.Builder mBuilder =
                            new NotificationCompat.Builder(Home.this)
                                    .setSmallIcon(R.drawable.common_google_signin_btn_icon_dark)
                                    .setContentTitle("My notification")
                                    .setContentText(message);

                    int mNotificationId = 001;
// Gets an instance of the NotificationManager service
                    NotificationManager mNotifyMgr =
                            (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
// Builds the notification and issues it.
                    mNotifyMgr.notify(mNotificationId, mBuilder.build());
                }
                previous = message;
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        });
    }

    private void getLocation() {
        if (ActivityCompat.checkSelfPermission(Home.this,
Manifest.permission.ACCESS_FINE_LOCATION)
                != PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission
                (Home.this, Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
```

```
            ActivityCompat.requestPermissions(Home.this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION}, REQUEST_LOCATION);

        } else {
            Location location =
locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);

            if (location != null) {

                latitude = location.getLatitude();
                longitude = location.getLongitude();


            } else {
                Toast.makeText(this, "Unble to Determine your Location default",
Toast.LENGTH_SHORT).show();


            }
        }
    }

    protected void buildAlertMessageNoGps() {

        final AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setMessage("Please Turn ON your GPS Connection")
                .setCancelable(false)
                .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
                    public void onClick(final DialogInterface dialog, final int id) {
                        startActivity(new
Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS));
                    }
                })
                .setNegativeButton("No", new DialogInterface.OnClickListener() {
                    public void onClick(final DialogInterface dialog, final int id) {
                        dialog.cancel();
                    }
                });
        final AlertDialog alert = builder.create();
        alert.show();
    }

    public void showLocation(Location location) {
        String latitude = "Latitude: ";
        String longitude = "Longitude: ";
```

```
        if (location != null) {
            latitude += location.getLatitude();
            longitude += location.getLongitude();
            Toast.makeText(this, latitude + ", " + longitude, Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(this, "Unble to Determine your Location",
Toast.LENGTH_SHORT).show();
        }
    }

    static final double _eQuatorialEarthRadius = 6378.1370D;

    static final double _d2r = (Math.PI / 180D);

    public static int HaversineInM(double lat1, double long1, double lat2, double long2) {
        return (int) (1000D * HaversineInKM(lat1, long1, lat2, long2));
    }

    public static double HaversineInKM(double lat1, double long1, double lat2, double long2)
{
        double dlong = (long2 - long1) * _d2r;
        double dlat = (lat2 - lat1) * _d2r;
        double a = Math.pow(Math.sin(dlat / 2D), 2D) + Math.cos(lat1 * _d2r) * Math.cos(lat2 *
_d2r)
                * Math.pow(Math.sin(dlong / 2D), 2D);
        double c = 2D * Math.atan2(Math.sqrt(a), Math.sqrt(1D - a));
        double d = _eQuatorialEarthRadius * c;

        return d;
    }

    public static double HaversineInMi(double lat1, double long1, double lat2, double long2) {
        double dlong = (long2 - long1) * _d2r;
        double dlat = (lat2 - lat1) * _d2r;
        double a = Math.pow(Math.sin(dlat / 2D), 2D) + Math.cos(lat1 * _d2r) * Math.cos(lat2 *
_d2r)
                * Math.pow(Math.sin(dlong / 2D), 2D);
        double c = 2D * Math.atan2(Math.sqrt(a), Math.sqrt(1D - a));
        double d = _eQuatorialEarthRadius * c;
        d = d*0.62137119;
        return d;
    }

    public static double HaversineInFt(double lat1, double long1, double lat2, double long2) {
        double dlong = (long2 - long1) * _d2r;
```

```
        double dlat = (lat2 - lat1) * _d2r;
        double a = Math.pow(Math.sin(dlat / 2D), 2D) + Math.cos(lat1 * _d2r) * Math.cos(lat2 *
_d2r)
                * Math.pow(Math.sin(dlong / 2D), 2D);
        double c = 2D * Math.atan2(Math.sqrt(a), Math.sqrt(1D - a));
        double d = _eQuatorialEarthRadius * c;
        d = d*0.62137119*5280;
        return d;
    }

}
```

MyService.java (Background Service)

```
package com.vishnu.emersave;

import android.*;
import android.Manifest;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.app.Service;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.net.Uri;
import android.os.Bundle;
import android.os.Handler;
import android.os.IBinder;
import android.provider.Settings;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.NotificationCompat;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
```

```
import java.util.ArrayList;
import java.util.Date;

public class MyService extends Service {
    private static final int REQUEST_LOCATION = 1;
    static Button button;
    static Button postToServerButton;
   // static LocationManager locationManager;
    static String Latitudes, Longitudes;
    //String Name;
   static double latitude;
  static   double longitude;
    String id;
    String testyes;
    final Handler handler = new Handler();
    private TextView seeSpeak;
    Date date = new Date();
    FirebaseDatabase database = FirebaseDatabase.getInstance();
    static int moose = 2;
    long timey = 7000;
    ArrayList<String> ids = new ArrayList<String>();
    int mNotificationId = 1;
    int mNotificationSafe = 1;
    int mtNotificationSafe = 1;
    String previous = "";
    long[] vibr = {200,200,200};
    String message;
    String m_androidId;
    String lon;
    String lat;
    String latVishnu;
    String lonVishnu;
    String latFriend;
    String lonFriend;
    private LocationListener locationListener;
    private LocationManager locationManager;
   int nums = 1;
    public MyService() {

    }
    @SuppressLint("MissingPermission")
    @Override
    public void onCreate()
    {
        DatabaseReference root = database.getReference("messages");
```

```
        ChildEventListener listen = root.addChildEventListener(new ChildEventListener() {

            @Override
            public void onChildAdded(final DataSnapshot dataSnapshot, String s) {
             /*
                  For WAAAAAAYYYYY Later:
                  Not Severity, have it as a pre-set option in settings, check it here and display as well

              */
            }
            @Override
            public void onChildChanged(DataSnapshot dataSnapshot, String s) {

                boolean cont = dataSnapshot.hasChild("latitude") &&
dataSnapshot.hasChild("longitude") && dataSnapshot.hasChild("expires") &&
                        dataSnapshot.hasChild("Name") && dataSnapshot.hasChild("message");

                if(!cont)
                {

                    // String all = dataSnapshot.toString();
                    //Toast.makeText(MainActivity.this,all,Toast.LENGTH_LONG).show();
                    return;
                }

                //
Toast.makeText(MainActivity.this,dataSnapshot.toString(),Toast.LENGTH_SHORT).show();
                //
Toast.makeText(MainActivity.this,dataSnapshot.getKey(),Toast.LENGTH_SHORT).show();

//Toast.makeText(MainActivity.this,dataSnapshot.getKey(),Toast.LENGTH_SHORT).show();
                if(!getId().equals(dataSnapshot.getKey())) {

                    // testyes = String.valueOf(HaversineInKM(latitude, longitude,
dataSnapshot.child("latitude").getValue(double.class),
dataSnapshot.child("longitude").getValue(double.class)));
                    //Toast.makeText(MainActivity.this,testyes,Toast.LENGTH_SHORT).show();
//1 mile in km = 1.609344
                    if (HaversineInKM(latitude, longitude,
dataSnapshot.child("latitude").getValue(double.class),
dataSnapshot.child("longitude").getValue(double.class)) < 1.609344) {
                        //  Toast.makeText(MainActivity.this,"Within
10KM",Toast.LENGTH_SHORT).show();
                        if(date.getTime()<dataSnapshot.child("expires").getValue(double.class)) {

                            String Name = dataSnapshot.child("Name").getValue(String.class);
```

```
                    String latty =
String.valueOf(dataSnapshot.child("latitude").getValue(double.class));
                    String longy =
String.valueOf(dataSnapshot.child("longitude").getValue(double.class));
                    id = dataSnapshot.getKey();
                    if(!ids.contains(id)) {
                        notif(Name, latty, longy);
                        ids.add(id);
                        Toast.makeText(getApplicationContext(),"A Friend Needs Help. Check
Notifications",Toast.LENGTH_LONG).show();
                    }

                }
            }


        }
    }
    @Override
    public void onChildRemoved(DataSnapshot dataSnapshot) {
        String namer = dataSnapshot.child("Name").getValue(String.class);
        if(!getId().equals(dataSnapshot.getKey())) {
            safey(namer);
        }
        ids.remove(dataSnapshot.getKey());

    }

    @Override
    public void onChildMoved(DataSnapshot dataSnapshot, String s) {


    }

    @Override
    public void onCancelled(DatabaseError databaseError) {

Toast.makeText(getApplicationContext(),databaseError.toString(),Toast.LENGTH_LONG).sho
w();
    }

});

locationListener = new LocationListener() {
    @Override
    public void onLocationChanged(Location location) {
```

```
            latitude = location.getLatitude();
            longitude = location.getLongitude();
            //Toast.makeText(getApplicationContext(),"Lat "+latitude+"  Lon "
+longitude,Toast.LENGTH_SHORT).show();

        }

        @Override
        public void onStatusChanged(String s, int i, Bundle bundle) {

        }

        @Override
        public void onProviderEnabled(String s) {

        }

        @Override
        public void onProviderDisabled(String s) {
        Intent i = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
        i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(i);
        }
      };
      locationManager = (LocationManager)
getApplicationContext().getSystemService(Context.LOCATION_SERVICE);


locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,0,5,locationLis
tener);
   }




   @Override
   public void onDestroy()
   {
      super.onDestroy();
      if (locationManager!= null)
      {
         locationManager.removeUpdates(locationListener);
      }
```

```
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId)
    {
      // createnotification(getApplicationContext());
      // Toast.makeText(getApplicationContext(),"This works",Toast.LENGTH_SHORT).show();

        onTaskRemoved(intent);

        return START_STICKY;
    }
    @Override
    public void onTaskRemoved(Intent rootIntent)
    {
        Intent restartService = new Intent(getApplicationContext(),this.getClass());
        restartService.setPackage(getPackageName());
        startService(restartService);
        super.onTaskRemoved(rootIntent);
    }

    @Override
    public IBinder onBind(Intent intent) {
        // TODO: Return the communication channel to the service.
        throw new UnsupportedOperationException("Not yet implemented");
    }
    private void createnotification(Context context) {
        PendingIntent notificIntent = PendingIntent.getActivity(context,0,new
Intent(context,Home.class),0);
        NotificationCompat.Builder mbuilder = new NotificationCompat.Builder(context)
                .setSmallIcon(R.drawable.common_google_signin_btn_icon_dark)
                .setContentTitle(" is Safe")
                .setContentText("Your friend is out of danger. You should still check on them to
verify safety").setContentIntent(notificIntent).setAutoCancel(true);

        NotificationManager notificationManager = (NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);
        notificationManager.notify(nums,mbuilder.build());
        nums++;
        if(nums>10)
        {
            nums=1;
        }

    }
    public String getId()
```

```
    {
        try {

            m_androidId = Settings.Secure.getString(getContentResolver(),
Settings.Secure.ANDROID_ID);

        }
        catch (Exception e)
        {
            System.out.println(e.getMessage());
        }
        return m_androidId;
    }
    static final double _eQuatorialEarthRadius = 6378.1370D;

    static final double _d2r = (Math.PI / 180D);
    public static double HaversineInKM(double lat1, double long1, double lat2, double long2)
{
        double dlong = (long2 - long1) * _d2r;
        double dlat = (lat2 - lat1) * _d2r;
        double a = Math.pow(Math.sin(dlat / 2D), 2D) + Math.cos(lat1 * _d2r) * Math.cos(lat2 *
_d2r)
                * Math.pow(Math.sin(dlong / 2D), 2D);
        double c = 2D * Math.atan2(Math.sqrt(a), Math.sqrt(1D - a));
        double d = _eQuatorialEarthRadius * c;

        return d;
    }
    public static double HaversineInM(double lat1, double long1, double lat2, double long2) {
        double dlong = (long2 - long1) * _d2r;
        double dlat = (lat2 - lat1) * _d2r;
        double a = Math.pow(Math.sin(dlat / 2D), 2D) + Math.cos(lat1 * _d2r) * Math.cos(lat2 *
_d2r)
                * Math.pow(Math.sin(dlong / 2D), 2D);
        double c = 2D * Math.atan2(Math.sqrt(a), Math.sqrt(1D - a));
        double d = _eQuatorialEarthRadius * c/1000;

        return d;
    }
    public void notif(String Name, String latty, String longy) {


        Intent notificationIntent = new Intent(Intent.ACTION_VIEW);
        notificationIntent.setData(Uri.parse("http://maps.google.com/?q="+latty+","+longy));
        PendingIntent pi = PendingIntent.getActivity(this, 0, notificationIntent, 0);
```

```
        NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(this);

mBuilder.setSmallIcon(R.drawable.common_google_signin_btn_icon_dark).setContentTitle(Na
me + " needs help!").setContentText("Click here to get directions to their
location").setContentIntent(pi).setAutoCancel(true);


        // Gets an instance of the NotificationManager service
        NotificationManager mNotifyMgr =
                (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
// Builds the notification and issues it.
        mNotifyMgr.notify(mNotificationId, mBuilder.build());
        mNotificationId++;
        //Can change value below to decide number of notifications on whim
        if(mNotificationId>20)
        {
           mNotificationId = 1;
        }

    }
    public void safey(String Name)
    {
        if(mNotificationSafe<mNotificationId || mNotificationSafe==20) {


            NotificationCompat.Builder mBuildery = new NotificationCompat.Builder(this);

mBuildery.setSmallIcon(R.drawable.common_google_signin_btn_icon_dark).setContentTitle(
Name + " is Safe").setContentText("Your friend " + Name + " is out of danger. You should
still check on them to verify safety").setAutoCancel(true);


            // Gets an instance of the NotificationManager service
            NotificationManager mNotifyMgry =
                    (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
// Builds the notification and issues it.
            mNotifyMgry.notify(mNotificationSafe, mBuildery.build());
            mNotificationSafe++;
            //Can change value below to decide number of notifications on whim
            if (mNotificationSafe > 20) {
               mNotificationSafe = 1;
            }
        }
    }
    private void getLocation() {
```

```
        if (ActivityCompat.checkSelfPermission(getApplicationContext(),
android.Manifest.permission.ACCESS_FINE_LOCATION)
                != PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission
                (getApplicationContext(),
android.Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {

        //     ActivityCompat.requestPermissions(Home.this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION}, REQUEST_LOCATION);

        } else {
            Location location =
locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);

            if (location != null) {

                latitude = location.getLatitude();
                longitude = location.getLongitude();
                this.Latitudes = String.valueOf(latitude);
                this.Longitudes = String.valueOf(longitude);

            } else {
                Toast.makeText(this, "Unble to Determine your Location default",
Toast.LENGTH_SHORT).show();

                this.Latitudes = String.valueOf(latitude);
                this.Longitudes = String.valueOf(longitude);
            }
        }
    }

    protected void buildAlertMessageNoGps() {

        final AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setMessage("Please Turn ON your GPS Connection")
                .setCancelable(false)
                .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
                    public void onClick(final DialogInterface dialog, final int id) {
                        startActivity(new
Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS));
                    }
                })
                .setNegativeButton("No", new DialogInterface.OnClickListener() {
                    public void onClick(final DialogInterface dialog, final int id) {
                        dialog.cancel();
```

```
                }
            });
        final AlertDialog alert = builder.create();
        alert.show();
    }

}
```

MyExceptionHandler.java (To Handle App Crashing)

```
package com.vishnu.emersave;

import android.app.Activity;
import android.app.AlarmManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;

/**
 * Created by Vishn on 12/28/2017.
 */

public class MyExceptionHandler implements Thread.UncaughtExceptionHandler {
  private Activity activity;

  public MyExceptionHandler(Activity a)
  {
     activity=a;
  }

   @Override
   public void uncaughtException(Thread thread, Throwable ex)
   {
      Intent intent = new Intent(activity,Home.class);
      intent.putExtra("crash",true);intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP |
         Intent.FLAG_ACTIVITY_CLEAR_TASK | Intent.FLAG_ACTIVITY_NEW_TASK);

      PendingIntent pendingIntent =
PendingIntent.getActivity(MyApplication.getInstance().getBaseContext(),
            0,intent,PendingIntent.FLAG_ONE_SHOT);

      AlarmManager mgr = (AlarmManager) MyApplication.getInstance().getBaseContext()
            .getSystemService(Context.ALARM_SERVICE);
               mgr.set(AlarmManager.RTC,System.currentTimeMillis() + 1000, pendingIntent);

               activity.finish();
```

```
        System.exit(2);

    }
}
```

PostingPage.java (To Post User-Information)

```
package com.vishnu.emersave;

import android.*;
import android.Manifest;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationManager;
import android.os.Vibrator;
import android.provider.Settings;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import static com.vishnu.emersave.MainActivity.Latitudes;
import static com.vishnu.emersave.MainActivity.Longitudes;


public class PostingPage extends AppCompatActivity implements View.OnClickListener{
    FirebaseDatabase database = FirebaseDatabase.getInstance();
    static String namef;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_posting_page);
    }
    String m_androidId;
    public String getId()
    {
        try {

            m_androidId = Settings.Secure.getString(getContentResolver(),
Settings.Secure.ANDROID_ID);
```

```
        }
        catch (Exception e)
        {
            System.out.println(e.getMessage());
        }
        return m_androidId;
    }
    public void onClick(View v) {

        if (v.getId() == R.id.btnPost) {

            /* EditText name = (EditText) findViewById(R.id.Name);
            String namef = name.getText().toString();
            DatabaseReference names = database.getReference("Vishnu").child("Name");
            names.setValue(namef);

            EditText country = (EditText) findViewById(R.id.Country);
            String countryf = country.getText().toString();
            DatabaseReference countries = database.getReference("Vishnu").child("Country");
            countries.setValue(countryf);

            EditText age = (EditText) findViewById(R.id.Age);
            String agef = age.getText().toString();
            DatabaseReference ages = database.getReference("Vishnu").child("Age");
            ages.setValue(agef);

            DatabaseReference status = database.getReference("Vishnu").child("Status");
            status.setValue(0);

            DatabaseReference Longitude = database.getReference("Vishnu").child("Longitude");
            Longitude.setValue(longitude);
            DatabaseReference Latitude = database.getReference("Vishnu").child("Latitude");
            Latitude.setValue(latitude);*/
            namef="Vishnu Default";
            EditText name = (EditText) findViewById(R.id.Name);
             namef = name.getText().toString();
            //DatabaseReference names =
database.getReference("messages").child(getId()).child("Name");
            //names.setValue(namef);
            //getName(namef);
            EditText country = (EditText) findViewById(R.id.Country);
            String countryf = country.getText().toString();
            DatabaseReference countries = database.getReference(getId()).child("Country");
            countries.setValue(countryf);
```

```
        EditText age = (EditText) findViewById(R.id.Age);
        String agef = age.getText().toString();
        DatabaseReference ages = database.getReference(getId()).child("Age");
        ages.setValue(agef);

        DatabaseReference status = database.getReference(getId()).child("Status");
        status.setValue(0);

        DatabaseReference Longitude = database.getReference(getId()).child("Longitude");
        Longitude.setValue(Longitudes);
        DatabaseReference Latitude = database.getReference(getId()).child("Latitude");
        Latitude.setValue(Latitudes);

    }
  }
  public String getName(String namef)
  {

    return namef;
  }


}
```

SignIn.java (Handles User Sign-In)

```
package com.vishnu.emersave;

import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Toast;

import com.google.android.gms.auth.api.Auth;
import com.google.android.gms.auth.api.signin.GoogleSignInAccount;
import com.google.android.gms.auth.api.signin.GoogleSignInOptions;
import com.google.android.gms.auth.api.signin.GoogleSignInResult;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.SignInButton;
import com.google.android.gms.common.api.ApiException;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthCredential;
```

```
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.GoogleAuthProvider;

public class SignIn extends AppCompatActivity {
    private SignInButton mGoogleBtn;
    public static final int RC_SIGN_IN = 1;
    private GoogleApiClient mGoogleApiClient;
    private FirebaseAuth mAuth;
    private FirebaseAuth.AuthStateListener mAuthStateListener;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sign_in);
        mAuth = FirebaseAuth.getInstance();
        mGoogleBtn = (SignInButton) findViewById(R.id.googleBtn);
        mAuthStateListener = new FirebaseAuth.AuthStateListener() {
            @Override
            public void onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) {
                if(firebaseAuth.getCurrentUser()!=null)
                {
                    startActivity(new Intent(SignIn.this,Home.class));
                }
            }
        };

        GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
                .requestIdToken(getString(R.string.default_web_client_id))
                .requestEmail()
                .build();

        mGoogleApiClient = new GoogleApiClient.Builder(getApplicationContext())
                .enableAutoManage(this, new GoogleApiClient.OnConnectionFailedListener() {
                    @Override
                    public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {
                        Toast.makeText(SignIn.this,"You got an
Error",Toast.LENGTH_LONG).show();
                    }
                })
                .addApi(Auth.GOOGLE_SIGN_IN_API, gso)
                .build();
        mGoogleBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
```

```
            signIn();
        }
    });
}
@Override
protected void onStart()
{
    super.onStart();
    mAuth.addAuthStateListener(mAuthStateListener);
}
private void signIn()
{
    Intent signInIntent = Auth.GoogleSignInApi.getSignInIntent(mGoogleApiClient);
    startActivityForResult(signInIntent,RC_SIGN_IN);
}
static GoogleSignInAccount account;
@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if(requestCode==RC_SIGN_IN)
    {
        GoogleSignInResult result = Auth.GoogleSignInApi.getSignInResultFromIntent(data);
        if(result.isSuccess())
        {
            account = result.getSignInAccount();
            Toast.makeText(getApplicationContext(),"Successful Sign
In",Toast.LENGTH_SHORT).show();
            firebaseAuthWithGoogle(account);
        }
    }
}

private void firebaseAuthWithGoogle(GoogleSignInAccount account) {
    AuthCredential credential =
GoogleAuthProvider.getCredential(account.getIdToken(),null);

    mAuth.signInWithCredential(credential)
            .addOnCompleteListener(this, new OnCompleteListener<AuthResult>()
            {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if(!task.isSuccessful())
                    {
                        Toast.makeText(SignIn.this,"Authentication
Failed",Toast.LENGTH_LONG).show();
```

```
                }

            }

        });
    }

}
```

Settings.java (User Settings)

```
package com.vishnu.emersave;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class SettingsPage extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_settings_page);
    }
}
```

activity_home.xml

```
<?xml version="1.0" encoding="utf-8"?>

<FrameLayout xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"

    android:orientation="vertical"
    tools:context="com.vishnu.emersave.Home">

    <android.support.constraint.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="10dp">


        <android.support.v7.widget.CardView

            android:id="@+id/quick_location"
            android:layout_width="160dp"
```

```
        android:layout_height="190dp"
        android:clickable="true"
        android:foreground="?android:attr/selectableItemBackground"
        android:onClick="click"
        app:layout_constraintBottom_toTopOf="@+id/user_info"
        app:layout_constraintEnd_toStartOf="@+id/help_request"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:gravity="center"
            android:orientation="vertical">

            <ImageView
                android:layout_width="100dp"
                android:layout_height="100dp"
                android:background="@drawable/circlebackgroundred"
                android:padding="5dp"
                app:srcCompat="@drawable/ic_location_on_black_24dp" />

            <View
                android:layout_width="match_parent"
                android:layout_height="2dp"
                android:layout_marginBottom="5dp"
                android:layout_marginLeft="5dp"
                android:layout_marginRight="5dp"
                android:layout_marginTop="15dp"
                android:background="@color/colorborder" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_margin="10dp"
                android:text="Quick Location"
                android:textAlignment="center"
                android:textColor="@color/black"
                android:textSize="18sp"
                android:textStyle="bold" />

        </LinearLayout>

    </android.support.v7.widget.CardView>
```

```
<android.support.v7.widget.CardView
    android:id="@+id/user_info"
    android:layout_width="160dp"
    android:layout_height="190dp"
    android:clickable="true"
    android:foreground="?android:attr/selectableItemBackground"
    android:onClick="click"
    app:layout_constraintBottom_toTopOf="@+id/settings"
    app:layout_constraintEnd_toStartOf="@+id/cancel"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/quick_location">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:orientation="vertical">

        <ImageView
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:background="@drawable/circlebackgroundyellow"
            android:padding="5dp"
            app:srcCompat="@drawable/ic_person_black_24dp" />

        <View
            android:layout_width="match_parent"
            android:layout_height="2dp"
            android:layout_marginBottom="5dp"
            android:layout_marginLeft="5dp"
            android:layout_marginRight="5dp"
            android:layout_marginTop="15dp"
            android:background="@color/colorborder" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:text="User Information"
            android:textAlignment="center"
            android:textColor="@color/black"
            android:textSize="18sp"
            android:textStyle="bold" />
    </LinearLayout>
</android.support.v7.widget.CardView>
```

```
<android.support.v7.widget.CardView
    android:id="@+id/cancel"
    android:layout_width="160dp"
    android:layout_height="190dp"
    android:clickable="true"
    android:foreground="?android:attr/selectableItemBackground"
    android:onClick="click"
    app:layout_constraintBottom_toTopOf="@+id/settings"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toEndOf="@+id/user_info"
    app:layout_constraintTop_toBottomOf="@+id/help_request">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:orientation="vertical">

        <ImageView
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:background="@drawable/circlebackgroundemerald"
            android:padding="5dp"
            app:srcCompat="@drawable/ic_cancel_black_24dp" />

        <View
            android:layout_width="match_parent"
            android:layout_height="2dp"
            android:layout_marginBottom="5dp"
            android:layout_marginLeft="5dp"
            android:layout_marginRight="5dp"
            android:layout_marginTop="15dp"
            android:background="@color/colorborder" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:text="Cancel Request"
            android:textAlignment="center"
            android:textColor="@color/black"
            android:textSize="18sp"
            android:textStyle="bold" />
    </LinearLayout>
```

```
    </android.support.v7.widget.CardView>

    <android.support.v7.widget.CardView
      android:id="@+id/help_request"
      android:layout_width="160dp"
      android:layout_height="190dp"
      android:clickable="true"
      android:foreground="?android:attr/selectableItemBackground"
      android:onClick="click"
      app:layout_constraintBottom_toTopOf="@+id/cancel"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintHorizontal_bias="0.5"
      app:layout_constraintStart_toEndOf="@+id/quick_location"
      app:layout_constraintTop_toTopOf="parent">

      <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:orientation="vertical">

        <ImageView
          android:layout_width="100dp"
          android:layout_height="100dp"
          android:background="@drawable/circlebackgroundblue"
          android:padding="5dp"
          app:srcCompat="@drawable/ic_local_hospital_black_24dp" />

        <View
          android:layout_width="match_parent"
          android:layout_height="2dp"
          android:layout_marginBottom="5dp"
          android:layout_marginLeft="5dp"
          android:layout_marginRight="5dp"
          android:layout_marginTop="15dp"
          android:background="@color/colorborder" />

        <TextView
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:layout_margin="10dp"
          android:text="Request for Help"
          android:textAlignment="center"
          android:textColor="@color/black"
          android:textSize="18sp"
          android:textStyle="bold" />
```

```
        </LinearLayout>
    </android.support.v7.widget.CardView>

    <android.support.v7.widget.CardView
        android:id="@+id/settings"
        android:layout_width="340dp"
        android:layout_height="60dp"
        android:clickable="true"
        android:foreground="?android:attr/selectableItemBackground"
        android:onClick="click"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/user_info">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:gravity="center"
            android:orientation="vertical">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginBottom="1dp"
                android:text="Settings"
                android:textAlignment="center"
                android:textColor="@color/black"
                android:textSize="30sp"
                android:textStyle="bold" />

            <View
                android:layout_width="match_parent"
                android:layout_height="2dp"
                android:layout_marginBottom="5dp"
                android:layout_marginLeft="5dp"
                android:layout_marginRight="5dp"
                android:layout_marginTop="1dp"
                android:background="@color/colorborder" />

        </LinearLayout>
    </android.support.v7.widget.CardView>

</android.support.constraint.ConstraintLayout>
```

```
    <android.support.design.widget.FloatingActionButton
        android:id="@+id/floatingActionButton"
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:clickable="true"
        android:onClick="click"

        android:layout_gravity="bottom|end"
        android:layout_marginBottom="7dp"
        android:layout_marginRight="7dp"
        app:srcCompat="@drawable/ic_mic_black_24dp">
    </android.support.design.widget.FloatingActionButton>

</FrameLayout>
```

activity_posting_page.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.vishnu.emersave.PostingPage"><![CDATA[]]>
    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

    </RelativeLayout>

    <TextView
        android:id="@+id/tvIsConnected"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginBottom="5dp"
        android:background="@color/colorWhite"
        android:text="is connected?"
        android:textColor="#ff66c7"
        android:textSize="18dp" />

    <TextView
        android:id="@+id/Name1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/Name"
```

```
    android:text="Name" />

  <EditText
    android:id="@+id/Name"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_toRightOf="@+id/Name" />

  <TextView
    android:id="@+id/tvCountry"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/Country"
    android:layout_below="@+id/Name"
    android:text="Country" />

  <EditText
    android:id="@+id/Country"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/Name"
    android:layout_toRightOf="@+id/tvCountry" />

  <TextView
    android:id="@+id/tvTwitter"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/Age"
    android:layout_below="@+id/tvCountry"
    android:text="Age" />

  <EditText
    android:id="@+id/Age"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/Country"
    android:layout_toRightOf="@+id/tvTwitter" />

  <Button
    android:id="@+id/btnPost"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:text="POST"
    android:onClick="onClick"/>
</LinearLayout>
```

activity_settings_page.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context="com.vishnu.emersave.SettingsPage">

   <android.support.v4.widget.NestedScrollView
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:fillViewport="true">

   </android.support.v4.widget.NestedScrollView>
</android.support.constraint.ConstraintLayout>
```

activity_sign_in.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:background="@color/turq"
   tools:context="com.vishnu.emersave.SignIn">

   <com.google.android.gms.common.SignInButton
      android:id="@+id/googleBtn"
      android:layout_width="374dp"
      android:layout_height="58dp"


      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintHorizontal_bias="0.5"
      app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/Intro"></com.google.android.gms.common
.SignInButton>

   <TextView
      android:id="@+id/Intro"
```

```
        android:layout_width="382dp"
        android:layout_height="87dp"
        android:text="Welcome To EmerSave"
        android:textAlignment="center"
        android:textColor="@color/ORANGISH"
        android:textColorHighlight="@color/colorWhite"
        android:textSize="36sp"
        app:layout_constraintBottom_toTopOf="@+id/googleBtn"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
```

Appendix F: Images of User Interface

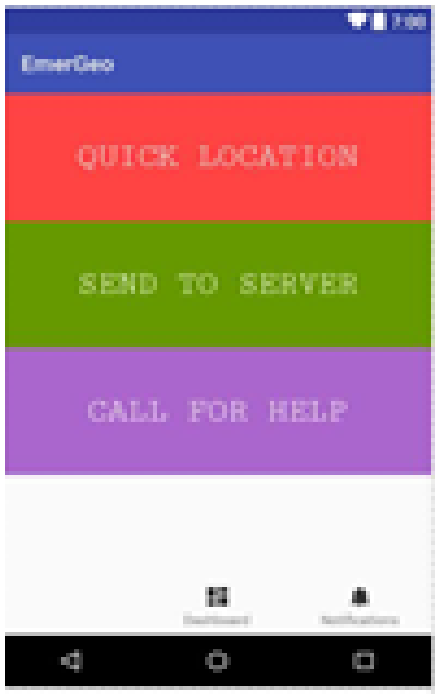

Figure 6: Image of Version 1 of application

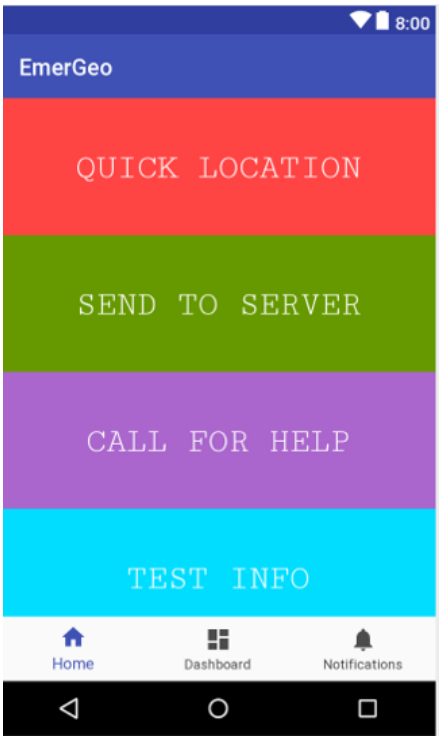

Figure 7: Image of Version 2 of application

Figure 8: Version 3, 4, 5 of application

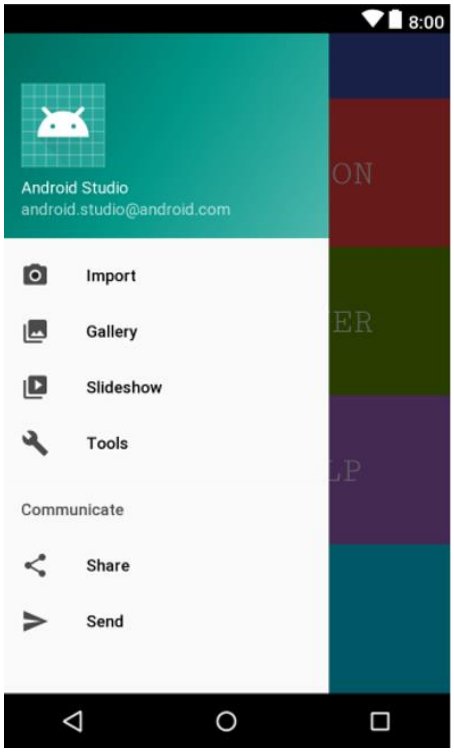

Figure 8: Image of navigation bar of Version 3,4,5 of application

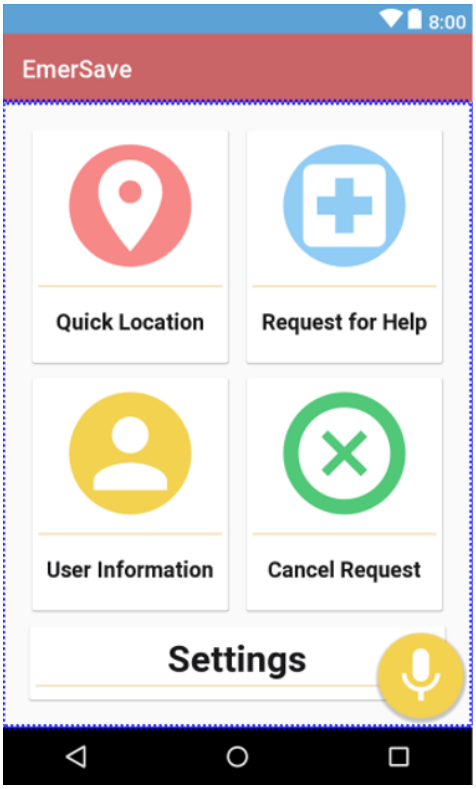

Figure 9: Image of user interface of Version 6-9 of application with minor and negligible changes made to sizes of buttons throughout versions 8-9.

Appendix G: Notes

Knowledge Gaps

KNOWLEDGE GAP	HOW TO RESOLVE IT	SOURCE/ WHERE INFO IS FOUND
What scenarios is geofencing used in?	Reading	(Suyama, 2016), Notes page 4, (Roy & Dutta, 2016) Notes page 5
What is geofencing ?	Reading	(Fattepur et al. 2016), Notes page 6
What similar products are already out there ?	Reading	Ryan, D. J. (2017)., Notes Page 8
How does the technology associate with GPS Tracking work?	Reading	(Suyama, 2016), Notes page 4, (Roy & Dutta, 2016) Notes page 5, (Garzon, 2016), Notes page 6
What are the advantages of using Android Studio for App Development?	Reading	

Literature Search Parameters

DATABASE/ SEARCH ENGINE	KEYWORDS
Scopus	Geofencing
Scopus	Geofencing + uses
IEEE Xplore	Geofencing
US Patent and Trademark Office	Geofencing + emergency response
Google Play Store	Emergency + Response + App
IEEE Xplore	GPS + tracking
IEEE Xplore	Geofencing + Location
IEEE Xplore	Geofencing + Smartphone
IEEE Xplore	Speech Recognition + Smartphone
IEEE Xplore	Android + Studio
Gale General OneFile	Voice + Activated
Statista	Emergency Response
Statista	Emergency Call
Google Scholar	Emergency Response Time
ProQuest	Emergency Response Time
Gale General OneFile	Firebase
Gale General OneFile	Voice Recognition

Notes on Existing Geofencing Systems

Source Title	***Using geofencing for a disaster information system***
Source Citation (APA)	Suyama, A. (2016, August 23). Using Geofencing for a Disaster Information System. Retrieved September 17, 2017, from https://www.computer.org/csdl/proceedings/icis/2016/0806/00/07550849.pdf
Source Found By	Searching Scopus for "geofencing"
Source Type	Published Conference Paper
Keywords	Geofencing, location based services
Summary	This is a paper which discusses a system which was created to inform individuals in Japan when they were entering an area of danger.
Reason for Interest	I might end up using a similar technology to call only people in a certain area to help the person in need.
Notes	• Uses client-server architecture. • Gets information from various outside sources-news, etc. • Using this information, creates a virtual fence • When person enters fence, they are informed that there is a dangerous situation • Or, after determining a range is dangerous, it sends out a message informing them they are in a danger zone. • This created product works greatly for people entering dangerous zone, but not for those exiting dangerous zone. • Japan suffers lots of natural disasters and there are high casualty rates due to the correct information not going out to those who need it. • According to provided tables, 68.9% of people stay in their houses after an evacuation directive, with 38.2% doing so because they did not know that this disaster affected them • A positive point is 1-1 directive, no need for news or tv or anything of that sort • Geofencing can dynamically change areas of interest. • They tested the accuracy of detecting an individual who entered the range
Questions	Are there further specifications of their testing method? What programming language and other programs did they implement in their app? What operating systems did it work with?

Source Title	***Providing context-aware healthcare services using circular geofencing technique***

Source Citation (APA)	Roy, M., & Dutta, S. R. (2016, October 27). Providing context-aware healthcare services using circular geofencing technique. Retrieved September 17, 2017, from http://ieeexplore.ieee.org/document/7724304/
Source Found By	Searching Scopus for "geofencing" and "uses"
Source Type	Published Conference Paper
Keywords	GPS, Google-cloud messaging, Near-location algorithm
Summary	An application which discusses letting users find the nearest registered doctor to them nearby in the field that they need.
Reason for Interest	This also deals with emergency response, and it also tracks a person as the main geofence center and the doctors/responders as the people moving in and out of the range.
Notes	• App meant for disaster situation • Finds nearest doctor/ specialist if possible-seems to be a two tier search • Uses a circular-radius based geofence based around the patient • App patient registers with their symptoms, and those are sent out to nearby emergency personnel. • Name of main services called context-awareness services • A previous individual named H.H.Lee already has introduced a system which tracks position of mobile systems through GPS server mobile networks • Uses longitude and latitude of nearest doctor/hospital to find out their proximity • One plan is to construct a database of these locations and doctors and their specializations, and this data based can be pinged when the user makes a search • This database would be dynamic, and as the position of a doctor changes, it adjusts the position within the database as well • This app's goal is to also take into account that the nearest doctor might be busy, so the option to go to the next closest specialist in the smallest amount of time • The solution is called PCAS (Predictive Context Aware Services) • This database has to be updated by an administrator • There is also a web interface involved in this process • Main goals/ check criteria: FASTER emergency service, at any time any location; accurate prediction of the positioning of a doctor.
Questions	How was this app tested? Was it used/successful in the real-world? What if the disaster makes it so that internet connection is lost, is at least the non- updated database available?

Source Title	***Reliable Geofencing: Assisted Configuration of Proactive Location-based Services***
Source Citation (APA)	Garzon, S. R., Elbehery, M., Deva, B., & Küpper, A. (2016, June 27). Reliable Geofencing: Assisted Configuration of Proactive Location-based Services. Retrieved October 8, 2017, from http://ieeexplore.ieee.org.ezproxy.wpi.edu/stamp/stamp.jsp?arnumber=7787079
Source Found By	Searching IEEE Xplore for Geofencing and Location
Source Type	Published Conference Paper
Keywords	Geofencing, Location
Summary	This paper finds faults with the accuracy of geofences in everyday uses, and creates a web application to identify and rank the effectiveness of geofences and will help non-experts set up geofences for their location based system
Reason for Interest	I may want to set up geofences as well for my Location Based System

Notes	• Mobile devices can now track users position without considerable battery drain thanks to miniaturized and integrated sensor solutions
	• Reliability issues due to limited preciseness of techniques
	• Things which contribute to accuracy of an LBS
	  ○ Accuracy of the position technique: higher mean accuracy means greater chance of app predicting place correctly
	  ▪ More accurate will lead to more battery drain
	  ○ If communication has to go through a server, the message can be delayed getting through the air
	• Main focus of research is to optimize functionality and battery life
	• For this paper, to make their geofence more attractive to general public, they had to make technology easy to use
	• Takes background properties, transport infrastructure, and local environment into consideration
	• Many app developing software today offer APIs for creating geofences, but limit number of geofences which can be created, and also the geofences have to be circular
	• Many colleges and third-party developers are trying to break this by offering non-circular shapes and unlimited Geofences
	• Geofences usually have a decreased reliability for two reasons:
	  ○ The geofences are created or identified using incorrect/non-accurate techniques
	  ○ The complexity of the geofence or the amount of cost that it takes to maintain them can either exceed the computational or financial resources
	• A geofence's reliability/ impact on a LBS includes:
	  ○ Positioning technique used
	  ○ Network Used
	  ○ Local Radio Signal Characteristics

	• Terms to know
	  ○ Hover Time: Amount of time a device needs to be located within a Geofence
	  ○ If hover time is too long, then transitions into the Geofence and out of the Geofence can be missed
	  ○ If a higher sampling rate is used to try to address this issue, than it will cause a higher consumption of energy
	  ○ Much research in the geofence field has been on trying to find the balance between energy-consumption and accurate geo-fence reporting
	  ○ Hover time is unpredictable and is different for each geofence
	  ○ With new algorithms, emergency consumption rates have the ability to dramatically adjust to the user
	  ○ The new algorithms allow the operating systems to adjust the check rate based on all the other tasks that are occurring at the time
	• If one wishes to determine possible transit paths, the location and orientation of the geofence has to be taken into account, and then all possible paths have to be identified for each mode of transportation
	  ○ The above would have to assume speed to determine time
	  ○ If geofencing app has capability to take traffic into consideration, this would be even better
	• GPS is the best outdoor positioning technique there is today
	• In cities, GPS experiences less accuracy due to building everywhere
	• Wi-Fi and cellular radio signals are now being used for more accurate positioning
	  ○ Will either triangulate cell id position or will use radio signals being monitored to determine location
	  ○ It was determined that the above techniques are not as reliable as detections of the Wi-Fi can cause it false alarm, while it may not alarm when it should if it cannot detect Wi-Fi.
	• Used ordinary web service for implementation instead of the mobile device itself to save CPU power of the device and reduce battery drain
	• The web service is a good way to tackle the position tracking as the calculations are done independent of the app and are generally applicable
	• Traffic and population density should also be taken into consideration

Questions	Are there any free APIs that were found? Was the web-service developed independently?

Source Title	***The Alzimio App for Dementia, Autism & Alzheimer's: Using Novel Activity Recognition Algorithms and Geofencing***
Source Citation (APA)	Helmy, J., & Helmy, A. (2016, May 18). The Alzimio App for Dementia, Autism & Alzheimer's: Using Novel Activity Recognition Algorithms and Geofencing. Retrieved October 8, 2017, from http://ieeexplore.ieee.org.ezproxy.wpi.edu/stamp/stamp.jsp?arnumber=7501720
Source Found By	Searching IEEE Xplore for Geofencing and Smartphone
Source Type	Published Conference Paper
Keywords	Geofencing, Smartphone
Summary	This app looks to help people with the above-mentioned issues to remember to do important activities on time, and also to make sure that they do no leave safe boundaries
Reason for Interest	The geofencing aspect/ continuous detection aspect.

Notes	• Autism, Alzheimer's, and Dementia all have negative impacts on people's memory, leading to memory loss
	• Many solutions are there to help people who forget and to treat the disease, but nothing out there for unsafe activity detection and leaving safe zone detection
	• The goals
	  ○ Continuously without disrupting users lives
	  ○ Algorithm for getting information had to be timely and accurate
	  ○ Maximize detection while minimizing false alarms
	• App works in two ways: detecting unsafe zones, and unsafe activities
	• Users quadtrees structures for reducing power consumption in geofencing
	• A lot of today's technologies use customized technology, smartwatches, boxes, etc) limited reach, little scaling, and expensive
	• Smartphones have great potential for scaling, making them affordable alternatives for patients
	• Activity recognition feature is trained using previously available data
	• Not using Android API helps consume a lot less battery power
	• Does not trigger false alarms and notifications
	• App uses sensing, computation, and storage capabilities of smartphones, not other servers, to analyze real time data
	• Wireless carriers charge about $10 a month for location sharing

	• Many devices and watches which are solely for GPS tracking cost between $8-$42 • Tested for detection accuracy, reliability in working in background, and affordable and efficient • Has 4 main architectural points: sensing, detection, notification, and user interface • Detection uses the threshold-based algorithm and the max-in-window based algorithm • Notifies user for a disarm period in case of false alarms • Messaging system alerts contacts set by the user • Accuracy-delay trade-off function: look at paper again for complicated formula
Questions	How were you able to send the messages to the specified contacts?

Notes on Geofencing

Source Title	***A Solution to Improve the Performance of Geofence Enabled GNSS Chipset***
Source Citation (APA)	Fattepur, M. B., S, S. G., & Huttangoudar, J. B. (2016, June 16). A Solution to Improve the Performance of Geofence Enabled GNSS Chipset. Retrieved September 18, 2017, from http://ieeexplore.ieee.org.ezproxy.wpi.edu/stamp/stamp.jsp?arnumber=7779405
Source Found By	Searching IEEE Xplore for "geofencing"
Source Type	Published Conference Paper
Keywords	Circular Geofences, Geofence, GNSS Chipset
Summary	The enabling of the GNSS Chip is what allows geofencing apps to track other mobile devices, and this article discusses the problems and potential fixes for this issue.
Reason for Interest	This article hopefully gives me more of an understanding on how geofencing works.
Notes	• GNSS stands for global navigation satellite system • GNSSs are able to keep track of where devices are if the devices have GPS enabled, so that they are able to send signals to the satellites • Geofencing used to track the motion of objects in and out of a set virtual fence • There are several problems: one is the incorrect tracking of entry and exit of fence

	• Another big minus is a huge hit that it takes on battery life • The relative speed and technology of the system are both low • Geofences allow people to monitor movement of virtual devices in and out of a virtual fence. • Fences do not have to be in a circular motion, can be a variety of shapes • Different geofences with the same GNSS can have different events trigger • Signal to Noise Radio (SNR) check is used to test received signal strength of GNSSs • GNSS is not always a single line from satellite to device, it can be multipath and reflect and bounce off walls and other objects • The bouncing off terrains can mess up things such as correctly reporting when an individual enters or exits a fence, but reports it at the wrong time • Line-Of-Sight (LOS) path is a direct path between satellite and receiver • Terms that should be known when reporting geofencing o Dwell time- amount of time user is in or out of geofence before application gets a notice of this change o Notification Latency- The minimum amount of time that the geofence event has to be triggered to application o Unknown Timer- minimum time limit which after the UNCERTAIN event should be reported to the application o Fix Interval- reporting intervals between GNSS chipset and user application • With geofencing becoming more and more popular within applications, it is important that the technology is quick and accurate • Geofencing Current Status is a programming technique used to find the current status of an individual relative to the fence • Results o With the test that was performed, the GNSS chipset gave back a result in 95 seconds after it was reported which was expected, versus the old technology which buffered o The accuracy of the new technology was much better compared to the older ones used. o Overall, there was minor lag time in giving back a result for position entering and exiting the fence in newer technology compared to older technologies. • Conclusion o The purpose of this experiment was to test the and understand how a GNSS and geofence worked. o It was also concluded that not enough research and testing was done in this project to justify the claims completely. It is believed that more areas of different densities and more

	variations of experiments, as well as more data collection, should be completed.
Questions	What was the testing method used for this? What foreign programs were installed/downloaded for this? What types of data was collected?

Notes on Speech Recognition Technology

Source Title	***Multivariate Autoregressive Spectrogram Modeling for Noisy Speech Recognition***
Source Citation (APA)	Ganapathy S. (2017, July 11). Multivariate Autoregressive Spectrogram Modeling for Noisy Speech Recognition. Retrieved October 8, 2017, from http://ieeexplore.ieee.org.ezproxy.wpi.edu/stamp/stamp.jsp?arnumber=7973047
Source Found By	Searching IEEE Xplore for Speech Recognition + Smartphone
Source Type	Published Paper
Keywords	Speech Recognition, Smartphone
Summary	Paper introduces new training for speech recognition to improve recognition in non-ideal surroundings
Reason for Interest	My project has speech recognition in it
Notes	• Automatic speech recognition (ASR) is degraded when there is background noise and echo • Autoregressive (AR) method only uses preserved high volume regions to make sure that non-damaged sounds are being registered • Deep neural networks have been used to improve automatic speech recognition, as they could be trained in various surroundings to teach the program how to deal with them • Realistically, not always possible to receive training in every possible scenario • AR modeling has shown to be effective in multiple situations when modeling sound • Multivariable AR is better, as it predicts which sounds are actually correct based on previous sounds • MAR is widely used for forecasting in Economics
Questions	Is this the type of Speech recognition which would be used by Google?

Source Title	***Communications Coming Full Circle--Are We Moving Back into a Voice-first World?***
Source Citation (APA)	In RefWorks
Source Found By	Searching WPI Summon for Voice Activated Technology
Source Type	Published Paper
Keywords	Voice, AI, Smartphone
Summary	Paper discusses voice activated technology, its development

Reason for Interest	My project has speech recognition in it
Notes	• AI and analytics used for speech recognition • Goal of independence to make everything voice activated • Biggest thing is security: Do we want just anyone accessing commands? • Bankers are using voice authorization, where it learns the person's voice and make sure that the voices match each other • Potential People not liking stuff: For many of these to work, it has to be always accessible meaning that it always has to be listening and storing portions of audio to analyze it • Secure software has to be coupled with it to make sure that it can be used in the public for private info • By 2016, Google's Voice recognition could analyze over 90% of over 5 million words, but this still should be increased. • Voice Activation has to have context-awareness, meaning that it should be able to figure out context of your command based on your location • Even voice activation needs to have a security layer, needs to have something which will make sure that it they are not doing anything too dangerous/ aka call police or shut down nuclear reactor just because of accidently saying something • A lot of the development is holding itself back as people are afraid of investing • loTivity and other platforms are being used for improving platforms for speech recognition
Questions	Is this the type of Speech recognition which would be used by Google?

Source Title	***The service threshold***
Source Citation (APA)	In RefWorks
Source Found By	Searching General OneFile for Voice Activated Technology
Source Type	Published Paper
Keywords	Voice, AI, Smartphone
Summary	Paper discusses voice activated technology, its development, and its importance
Reason for Interest	My project has speech recognition in it
Notes	• The Informative Services field can be greatly improved through the use of voice recognition. • One major way in which the above is expected to make an immediate positive impact is in the patient service field. • Voice recognition, unlike other advanced methods, will be immediately accepted into society once it can be developed. • Most immediate use will be in medical records
Questions	Why is the patient care field emphasized?

Notes on Existing Technologies Similar to the Project

Source Title	***Method and System for Geolocation and Coordinated Communication with Emergency Responders***
Source Citation (APA)	Ryan, D. J. (2017). *U.S. Patent No. 20170099579* . Washington, DC: U.S. Patent and Trademark Office
Source Found By	Searching US Patent & Trademark Office for "geofencing" + "emergency response"
Source Type	Patent
Keywords	Geofencing, Emergency Response
Summary	This patent is regarding a software which tracks emergency situations and reports it to responders who are within a designated area
Reason for Interest	This patent is similar to what I am doing but not an app or hands free or really people interactive

Notes	• Important Claims that are made
	  ○ A method to identify and designate responders to an emergency situation (Difference: Does not have people actually share their emergency, it finds it out on their own)
	  ○ Has its own server and use wireless telecommunications network (Just an app)

	○ A notification that sends a request of the locations of users in the area, determining which is the best, and sending them further information (Just first one to respond) ○ Having a chain of messages (Don't need it) ○ Tracking locations of responders until the emergency is no longer there (Again not really applicable) ○ Enlargening the size of the geofence to search if the number of active responders in the area is not sufficient to help the emergency (I will just call the cops instead of this) ○ Adding a second geofence around the first and sending messages to people inbetween the two fences (NOPE) ○ Third message tell responders of emergency resonders (Mine would be first/second) ○ Having either volunteer or professional responders (Mine would most likely be known friends or cops) ○ Providing location of responders to individual who needs help (Just the distance is what I'll do) ○ Notification of emergency is received by emergency server of an emergency center (Mine just goes out to all) ○ Essentially all of the above but with the server (Not really a problem) • These days, only verbal communication through phone is done, when all these smartphones have high-tech things that can be really useful in the future • Have to subscribe to this service and has to choose a range of services (medical, fire, etc.) (Mine is just an app) • Information is stored in a secured cryptographic way to ensure that the user does not have their location stolen • Basically, Emergency request is asked, and then dispatch center can choose to activate this if it deems that this is the quickest way to get responders to the area • **Users can have multiple or one emergency response groups for different things based on emergency that they report (This is somewhat…I have to be careful with how I handle this aspect)** • Can store multiple types of information {medical, family, etc} • People can register to be responders • A campus, neighborhood, venue security call • **Location determined by cellular network** • A message is sent out to current emergency responders when a situation is resolved • If person who needs emergency moves or the dangerous situation changes users are updated • **Dynamic Location elements** • **Users can report to responders ok or not ok and they will get this notification**

	• **In scenarios where person in distress cannot directly interact with phone, a normal 911 call or <u>simple actions on the phone</u> can be done THE DISPATCH CENTER CAN THEN ACTIVATE GEOLOCATION SERVICES AND INSTRUCT UE TO START VOICE OR VIDEO RECORDING. THE EMBODIMENT TECHNIQUE WORKS EVEN WHEN PHONE IS TURNED OFF BY ATTACKER OR CALL IS CUT** • **ANYONE CAN CALL TRUSTED PEOPLE AND THEN THEY CAN ON BEHALF OF THEM TRHOUGH THE NETWORK CALL THE COPS AND THE COPS WOULD GET THEIR LOCATION** • **PROGRAM CAN BE ATTACHED TO MEDICAL APPLICATION WHICH TRACKS MEDICAL STUFF AND INFORM EMERGENCY RESPONDER WHEN THERE IS A SUDDEN CHANGE** • **As the user calls 911, they can simultaneously have friends or others that they know get info about their status**
Questions	

Source Title	***Emergency Response App***
Source Citation (APA)	<u>Primer Analytics & Systems, Inc.</u> (2016). Emergency Response App (Version 1.1) [Mobile application software]. Retrieved from https://play.google.com/store/apps/details?id=com.era.user&hl=en
Source Found By	Searching the Google Play Store for “Emergency Response App”
Source Type	Mobile Application
Keywords	Emergency Response
Summary	This app is primarily meant to organize the many different types of emergency contact numbers so that calling emergency centers is quick and easy
Reason for Interest	To see what other types of emergency response applications are already available
Notes	• Sends location to command center and emergency response center • Sends details with pre- listed medical conditions • Able to send help to family within setup geofence • Panic button sends signal to general emergency response
Questions	

Source Title	***mySOS SA***

Source Citation (APA)	mySOS. (2017). mySOS SA (Version 3.1.6) [Mobile application software]. Retrieved from https://play.google.com/store/apps/details?id=za.co.widge.sos&hl=en
Source Found By	Searching the Google Play Store for "Emergency Response App"
Source Type	Mobile Application
Keywords	Emergency Response, Medical Emergency
Summary	This app is for South Africa and sends location of person in need of help to a response station along with selected contacts, and will even provide them GPS tracking to get back to person in need.
Reason for Interest	Apps that are already out there.
Notes	• Allows to activate for personal or friend emergency • Various types to choose from, including medical, fire, security, etc. • Sends name, phone number, type of emergency, exact location, view emergency from smartphone, navigate from current location to emergency location • Offers emergency info to closest appropriate responders based on emergency selected • Find near me feature finds emergency places near the person and displays them • There is a countdown timer which contacts emergency contacts after the timer reaches 0, and sends the above-mentioned information • It is an actual external hardware • Detects fall motion • Only have to press with finger to activate it • Uses Bluetooth technology to connect to phone • Can be used for running and other sports activities • Have to press it for more than 2 seconds
Questions	How did you do the feature to have the app give directions to the location? How were you able to locate the location of the user so accurately?

Notes on Android Studio

Source Title	***Inter-App Communication between Android Apps Developed in App-Inventor and Android Studio***
Source Citation (APA)	In RefWorks same with everything ABOVE THIS
Source Found By	Searching Android Studio in IEEE
Source Type	Published Journal Article
Keywords	Android, App Inventor, Android Studio, Mobile apps.
Summary	Discusses the techniques and improvements to sending information between Android Studio Apps and App Inventor Apps
Reason for Interest	I will be using Android Studio to develop my app
Notes	• Android was developed with inter-app communication in mind • Apps are developed using Android SDKs and Android IDEs like Android Studio • App Inventor originally made by Google, but has been recreated by MIT App Inventor 2 • Apps made in App Inventor can communicate with each other, same for Android Studio, but cannot communicate between those two platforms • AI does not have the advanced features which Android Studio has • App Inventor does not have standard naming conventions, and nobody can see the source code • AI uses visual blocks for programming • The experiment for this involved making 4 apps, two in each, and sending a string with upper and lower-case characters back and forth and testing run time • When AI apps were at receiving end, transfer time was much longer than when AS apps were on the receiving end • This is because AI applications require more time to process, as the block code has to be converted into byte code • AS does not need a third-party compiler to convert code, ending in a more standard application with faster response times • In Application size, AI was much bigger compared to AS apps even though amount of code was about the same • In AI you can only initialize 1 string value • AI apps can only receive strings to main activity • No intent filter to initiate specific activity
Questions	None

Notes on Firebase

Source Title	***Developers - Let's Try: Power Your Mobile Applications with Firebase***
Source Citation (APA)	In RefWorks same with everything ABOVE THIS
Source Found By	Searching Firebase in Gale OneFile
Source Type	Published Journal Article
Keywords	Firebase, Google
Summary	
Reason for Interest	I plan to use firebase for maintaining my server

Notes	• Google mobile platform
	• Used to develop high quality apps
	• Real time database
	  o Works on most popular operating systems
	  o Managed by Google
	  ▪ Allows easy integration with other Google features
	• By 2014, has become a full feature platform with development, testing, distribution, and analytics
	• "Develop and test application"
	  o The main portion of this is the Real-Time Database which Firebase provides, which serves as a Database-as-a-service model.
	  o Provides (Software Development Kit) SDK for all major development platforms such as iOS, Android, Unity, and Web
	  o Development by multiple users, synchronization across multiple users
	  o Crash report incase something breaks
	  o Authentication for users
	  o Optional cloud storage
	  o Mobile phone models for testing
	  o Performance monitoring for app
	  o Firebase SDK allows easy updates through all subscribed members for the apps
	  o Good at backend jobs, specifically cloud computing
	• "Grow and engage audience"
	• Analytics give stats to let you know things like demographics and regions and type of people using app, also allows checking of , notify of major events, allows admob, lets you view different app versions and app debugging

Questions	None

Notes on Emergency Response Times

Source Title	***Response Time Effectiveness: Comparison of Response Time and Survival in an Urban Emergency Medical Services System***
Source Citation (APA)	In Ref Works
Source Found By	Searching Google Scholar for emergency response time
Source Type	Published Journal Article
Keywords	Response time, EMS
Summary	Discusses a survey done throughout the country on emergency response times
Reason for Interest	I want to see if my argument can be supported
Notes	• 5424 cases reviewed • Stroke patients • With a p value of 0.002, there was significance that response time under 5 minutes had a much greater survival rate than response time under 5 minutes
Questions	None

Notes on Stats Related to Issue

Source Title	***Personal emergency response system device ownership rate among U.S. respondents in 2013, by age***
Source Citation (APA)	Link-age. (n.d.). Personal emergency response system device ownership rate among U.S. respondents in 2013, by age. In Statista - The Statistics Portal. Retrieved October 26, 2017, from https://www.statista.com.ezproxy.wpi.edu/statistics/385228/us-respondents-owning-a-personal-emergency-response-system-device/.
Source Found By	Searching Statistica for Emergency response
Source Type	Statistics
Keywords	PERS, Elderly
Summary	Discusses the statistics of elderly who have devices to call for emergency response
Reason for Interest	I want to see if there will be interest in this app/ market
Notes	• 8% of people between 55 and 59 have personal emergency response devices • 18% for 60-64 • 24% for 65-69 • 32% 70-74 • 38% for 75-79 • 46% for 80-84 • 57% for 85-9 • 55% for 90-100+
Questions	None

Source Title	***Projected number of telehealth* patients worldwide from 2013 to 2018 (in millions)***
Source Citation (APA)	IHS. (n.d.). Projected number of telehealth* patients worldwide from 2013 to 2018 (in millions). In Statista - The Statistics Portal. Retrieved October 26, 2017, from https://www.statista.com.ezproxy.wpi.edu/statistics/302641/global-telehealth-market-patients/.
Source Found By	Searching Statistica for emergency response
Source Type	Statistics
Keywords	Telehealth, patients
Summary	Discusses the statistics of number of people who will need telehealth sevices
Reason for Interest	I want to see if there will be interest in this app/ market

Notes	• .35mil 2013 • .64mil 2014 • .1.16 mil 2015 • 2.11 mil 2016 • 3.84 mil 2017 • 7 mil 2018
Questions	None

Source Title	***Emergency events among U.S. residential care residents in 2014, by community size***
Source Citation (APA)	CDC. (n.d.). Emergency events among U.S. residential care residents in 2014, by community size. In Statista - The Statistics Portal. Retrieved October 26, 2017, from https://www.statista.com.ezproxy.wpi.edu/statistics/492739/residential-care-residents-by-community-size-emergency-events/.
Source Found By	Searching Statistica for emergency calls
Source Type	Statistics
Keywords	Calls, emergency
Summary	Discusses the statistics of number of people in emergency situations
Reason for Interest	I want to see if there will be interest in this app/ market
Notes	• 835, 200 people surveyed • 21% had a fall last 90 days • Emergency visit in last 90 days 12% • 8% had overnight stay at hospital
Questions	None

Source Title	***9-1-1 Statistics***
Source Citation (APA)	In RefWorks
Source Found By	Recommended by Teacher
Source Type	Website
Keywords	9-1-1, PSAP
Summary	Discusses statistics of 9-1-1 calls made
Reason for Interest	I want to find some stats about 9-1-1 calls made in the country

Notes	• 240 million calls made to 9-1-1 every year • 80% are from wireless devices • Not all areas have the ability to have their local 9-1-1 get their location • Calling 9-1-1 also costs money, which can range from a few cents to $6.00 depending on the state
Questions	None

Source Title	***Ensuring Access***
Source Citation (APA)	In RefWorks
Source Found By	Recommended by Teacher
Source Type	Website
Keywords	9-1-1, NG911
Summary	Discusses statistics of 9-1-1 calls made and issues with it
Reason for Interest	I want to find some stats about 9-1-1 calls made in the country
Notes	• Individuals who are deaf, hard of hearing, old, or living in rural or large cities can have issues accessing 911 services • Deaf and hard of hearing must use a teletypewriter, which allows users to type messages to each other, but these are slow if they have to communicate with 911 and are hard to operate • 911 services are used for a lot more than just emergencies in rural areas • Takes first responders longer to reach scene because it is a bigger distance they must traverse • Call takers have to be on the phone longer with person in distress • Longer to get to hospital • Limited responders can be deadly in large-scale disaster • Large cities receive more than 10 million calls a year and they are often held up because of the volume. Some people with emergencies are not able to connect with responders due to lines being busy
Questions	None

Source Title	***Matching EMS response times to patient needs***

Source Citation (APA)	In RefWorks
Source Found By	Searching WPI Summons for Emergency Response Time
Source Type	Journal Article
Keywords	EMS,911
Summary	Study done on 9-1-1 calls
Reason for Interest	I want to find some stats about 9-1-1 calls made in the country
Notes	• In early 1970s, 70% of EMS callers needed paramedic support and 30% did not • Today, only 35% need paramedic support and the rest do not • Since 2008, the Los Angeles Fire Department has seen increasing EMS activation regardless of the fact that the population has not grown significantly • Many EMS systems set a goal of responding within 5 minutes of a call, which is a very expensive endeavor, while majority of these calls do not necessarily need a response in 5 minutes
Questions	None

Source Title	***Thousands die from ambulance delays***
Source Citation (APA)	In RefWorks
Source Found By	Searching Google for ambulance delays
Source Type	Newspaper Article
Keywords	Elderly, Heart-Attack
Summary	Article on reasons why the elderly are dying due to late EMS arrival
Reason for Interest	I want to find some stats about 9-1-1 calls made in the country
Notes	• Only 3 of England's 32 ambulance services actually meet the standard of reaching the person in under 8 minutes • About 3,000 people could be saved from a heart-attack if the calls were answered in that time
Questions	None

Source Title	***Number of apps available in leading app stores as of March 2017***
Source Citation (APA)	In RefWorks
Source Found By	Searching Google for applications
Source Type	Statisticas
Keywords	App Store, Play Store
Summary	Statistics on popularity of Google Play Store and App Store
Reason for Interest	I want to find stats on the popularity of the App Store and the Play Store
Notes	• Google play offers 2.8 million apps • App store offers 2.2 million apps
Questions	None

Source Title	***Mobile phone internet user penetration worldwide from 2014 to 2019***
Source Citation (APA)	In RefWorks
Source Found By	Related links to Statistics on Google Play Store and App Store
Source Type	Statistics
Keywords	Mobile technology
Summary	Statistics on use of internet on phones
Reason for Interest	I want to find stats on the number of people who access internet on mobile phones
Notes	• 58.9% 2017 • 61.2% 2018 • 63.4% 2019 • 2020, number of smartphone users 2.87 billion, up from 2.1 billion 2016
Questions	None

Source Title	***Average daily time spent online via mobile by internet users worldwide as of 2nd quarter 2016, by generation (in minutes)***
Source Citation (APA)	In RefWorks
Source Found By	Related links to Statistics on Google Play Store and App Store
Source Type	Statistics

Keywords	App Store, Play Store
Summary	Statistics on use of internet on phones
Reason for Interest	I want to find stats on the number of people who access internet on mobile phones
Notes	• 3hr 5 min average Millenials • 1hr 50 min average Generation X
Questions	None