Location-Based Mobile Application for Emergency Volunteering

David Faith Odesola

A submission presented in partial fulfilment of the requirements of the University of South Wales / Prifysgol De Cymru for the degree of Masters in Computer Science

September 2022

University of South Wales
Prifysgol De Cymru

Certificate of Research

This is to certify that, except where specific reference is made, the work described in this dissertation is the result of the candidate. Neither this dissertation, nor any part of it, has been presented, or is currently submitted, in candidature for any degree at any other University.

Student Name (Printed)	David Faith Odesola
Student Signature	o.F.Odesola O.D
Registered Course of Study	MSc Computer Science
Student Date of Signing	September 5, 2022
First Supervisor Name (Printed)	Dr. Shiny Verghese
Supervisor Signature	
Supervisor Date of Signing	September 5, 2022
Second Supervisor Name (Printed)	Dr. Janusz Kulon
Supervisor Signature	
Supervisor Date of Signing	

Acknowledgements

First and most importantly, I give praise to my Lord and Saviour Jesus Christ for giving me both the strength and wisdom to be able to conduct this research project. There is no doubt that I would not have been able to accomplish it without His assistance in my life. I would also like to thank my lovely parents Oladipo and Olubukola Odesola for their loving support and sacrifices towards me. For the rest of my life, I will be eternally thankful to them both for always being there for me, especially through difficult moments. Words are not enough to express my full appreciation.

Additionally, I would also want to express my sincere gratitude to my Uncle/Senior Daddy, Pastor Johnson Odesola, for his unceasing prayers and support throughout my life. He has always inspired me to be both excellent and ambitious in everything that I do. Also, I would like to specially acknowledge Dr Dan-Habu for his continual assistance and kindness towards me.

I would also like to thank both my supervisors Dr Shiny Verghese and Dr Janusz Kulon for their guidance and support throughout this research project.

To my lovely sisters Victoria, Esther, and Elizabeth, I would like to appreciate them for they have been a massive help throughout this research project. Lastly, big thanks to all my other family members and friends that took the time to support me during this research project.

Thank you and God Bless you!

Abstract

According to various organizations, volunteering initiatives have been said to have played a crucial role in the aspects of emergency response. Hence, with the current advancement in technology, there should be new ways of coordinating the use of volunteers in the provision of assistance to people within communities that require emergency aid. The primary idea is that trained volunteers that are located nearby could potentially minimize the time needed for aid to be rendered to the victim.

Furthermore, this research project was aimed at developing a mobile platform that would enable volunteers to assist with nearby emergency help requests. The entire project consisted of 3 main parts. (1) a mobile app for volunteers, (2) a mobile app for the user (help requester) and (3) a web interface as the administration panel to verify the volunteers registering within the platform.

In order to perform a proper assessment of the developed mobile application, 2 types of evaluations were carried out. The first was a heuristic evaluation of the user interface design and the second was a user evaluation that involved a random group of respondents who assessed the features of the mobile application in a mock environment. Based on responses gathered from the user evaluation phase, it was evident that the main objective of developing the volunteering mobile app with a real-time notification system was achieved. The registered volunteers who received notifications of emergency incidents reported by nearby users (help requesters) successfully used the GPS for location tracking and navigation available within the mobile app to reach the destination. The results of the heuristic evaluation carried out on the mobile application were also satisfactory.

Table of Contents

Table of Contents

Table of Contents

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

The ability of an individual or an organisation to reliably respond to emergencies such as fire, health-related incidents, and even natural disasters has grown increasingly crucial nowadays (Yang et al., 2009). According to (Larson et al., 2006; Fan et al., 2013), response time is most often the deciding factor in determining the efficiency of an emergency response. Furthermore, emergency services are always looking out to employ new methods or strategies specifically aimed at minimising response times. Some of the methods being employed include the use of volunteers, publicly available Automated External Defibrillators (AEDs), and even Unmanned Aerial Vehicles (UAVs) (Matinrad and Reuter-Oppermann, 2021).

Volunteering initiatives have been claimed to have a significant impact on emergency response services by minimising the time gap needed for aid to be rendered to a victim. Its relevance in crisis management also has had considerable influence and should not be neglected or ignored (Paylor, 2011). One of the key reasons why volunteers are deemed to be incredibly valuable is not just because of their capacity to improve the workforce, but also in their ability to collaborate with paid workers or staff members. This as a result could potentially ease the amount of workload within a given organization; further enhancing task efficiency. Upon further investigation, it was seen that volunteering efforts have played a critical role in some areas including emergency response, social/health services, and support works (Pardess, 2008; Paylor, 2011). In some countries, such as the US, Austria, and Israel, a significant amount of volunteers are currently being integrated into the Emergency Medical Service (EMS) workforce (Khalemsky et al., 2020). The (UK Cabinet, 2011) also stated that specialised volunteers have proven to have helped alleviate the pressure on the emergency services due to their ability to provide aid to a victim within a shorter time frame. This is primarily true for communities located in remote or rural areas which are often quite challenging for emergency response services to reach. To ameliorate the situation, local volunteers are being notified so that they can provide the necessary aid and support within a short period (Khalemsky and Schwart, 2017; Pilemalm, 2020).

1.2 Problem Statement

With the current advancement in technology, there should be newer and much more efficient ways of taking advantage of volunteering roles in the aspects of emergency response and citizen help requests. It was seen that existing emergency volunteering platforms were often quite restricted, mainly in terms of the kinds of emergency help assistance within which it functions. Hence, this research aims to tackle this issue by developing a mobile platform that would allow registered volunteers to assist with a variety of emergency aid requests which depends on their expertise or category level. Once an emergency is reported, the implemented system should be able to determine the most suited volunteer based on the nature of the help request, including other factors such as distance, category, or availability.

1.3 Research Questions

- What technical requirements are needed for a volunteering software platform?
- What evaluation would be carried out after the implementation of such a system?

1.4 Research Goals

1.4.1 Aim

The primary aim of this project is to develop a mobile application that would enable trained volunteers to respond to nearby emergency cases using a location-based service.

1.4.2 Objectives

- Investigate and report on existing emergency volunteering platforms.
- Research the technologies to be utilised for development.
- Develop a mobile app that would allow trained volunteers to respond to surrounding emergencies or help requests using a real-time location-based system.
- Design a real-time notification system that would notify a selected number of volunteers depending on a set of predetermined parameters.
- Incorporate a navigation system that would provide the ideal route to the emergency location.
- Perform a heuristic evaluation on the developed mobile application.
- Assess the functionality of the implemented system using an online survey.
- Undertake a qualitative evaluation on the use of the application.

2 Literature Review

This chapter aims to give a review of existing emergency response systems to address what they are and how they are being used. In addition, this chapter will also compare similar emergency volunteering platforms that have been previously implemented to provide further insight into its findings. Lastly, this chapter will also report on some of the relevant technologies to be utilised for development.

2.1 Emergency Response Systems

According to the (Cybersecurity and Infrastructure Security Agency, 2019), Medical, Law Enforcement, Fire and Rescue, Emergency Management, and Civil Works are the five main sectors that make up the Emergency Services. Furthermore, with the different variations of emergency types, there are systems in place that aims to assist in the allocation of resources depending on the given emergency. These are called Emergency Response Systems (ERS), which are systems that are being utilised by various organizations to aid in their response strategy during emergencies. The main goal of ERS is to identify the best course of action to be taken, based on the nature of any given emergency (Jennex, 2007). According to (Belardo et al., 1984), there is a standard model or key components that must be present with such systems which consist of a database, data analysis ability, normative models, and user interface. The database is meant to store the emergency information in a “RAW” format or in such a way that it could be reprocessed for further data analysis. The Normative model should be designed to provide the best course of action within any given emergency incident. Lastly, the user interface should be able to present all the data and suggestions in a format that is well understood by the end-user or the disaster management team. Subsequently, user usability is regarded as one of the most critical components since such systems would be rendered useless if the end users are unable to accurately interpret the data given by the system.

It was stated that there are still some underlying issues with current systems mainly in terms of its information infrastructure and how the data is being managed. To improve on this, (Abu-Elkheir et al., 2016) proposed the use of heterogeneous data analysis by crowdsourcing real-time data from mediums such as IoT and smart devices primarily distributed across a city to provide insight into the nature of the emergency. This would therefore provide the most relevant information needed for first responders to assist during emergencies.

2.2 Implemented Emergency Response Platforms

Currently, there are various emergency response platforms actively being employed to aid emergency services in responding to various types of emergencies. These systems have played a crucial role in disaster management and have helped revolutionize the principle of emergency response.

2.2.1 ECall Program

The ECall (short for "Emergency Call") is an emergency vehicle service that was initiated by the European Union (EU) back in 2002. The ECall system aims to provide real-time assistance for motor-based accidents in EU countries. This system used GPS technology for precise location tracking and an automated voice call provisioning system to establish a cellular network connection with the local emergency response centre, also called Public Safety Answering Point (PSAP). Additionally, ECall also supports the transfer of vehicular data known as ECall Minimum Set of Data (MSD) which consists of additional information such as the Vehicle's Identification Number (VIN), type, timestamp, passenger count, and much more. All this information is capable of being transferred through the in-band modem which is the same network being used for the voice communication link (Filjar et al., 2011). At the start of 2015, it has been made mandatory that all new cars to be sold within the EU must be equipped with ECall (Oorni and Goulart, 2017).

2.2.2 9-1-1

9-1-1 is an emergency telephone number that allows users to call and request help directly from the emergency services. The 911 system was initially developed by the United States back in the 1960s however, it was first actualised in Great Britain back in the 1930s with the "999" system. Before the development of the 911 emergency system, if a user wanted to report an emergency, they would have to dial up the appropriate services such as the police station, hospital, or fire station. Alternatively, one might need to dial up an operator, who would then contact the emergency services on their behalf. The countrywide 9-1-1 telephone system was then implemented with the primary purpose of streamlining the procedure of requesting assistance (New Hampshire Bureau of Emergency Communications, 2012).

2.2.3 E911

E911, also known as Enhanced 911 is the upgraded version of the 911 system. This enhanced 911 system enabled users to call the emergency services from their mobile phones which could not be done previously. This system is configured so that the caller's information including location data is automatically sent to a dedicated 911 call centre or PSAP, which is then relayed to the relevant emergency responders that would be dispatched to the caller’s location. When a call is placed to the 911 centre, the system automatically extracts the identification number of the phone being used and it is then queried against the Automatic Location Information (ALI) database to establish the caller's location.

The Federal Commission of Communication (FCC) also incorporated the use of interconnected Voice over Internet Protocols (VoIP) systems to further enhance the E911 system (Federal Communication Commission, 2017). According to reports, the E911 has significantly sped up emergency services' response times. Furthermore, it was seen that this platform has actively helped to increase patient survival rates and hospital discharge rates. (Athey and Stern, 2000).

2.2.4 Summary

The central purpose of emergency response systems indicates that they are meant to advise first responders on the appropriate course of action for every emergency incident. It must offer them the required information to help in their reaction during emergencies. A brief overview of existing emergency response systems reveals that transmitting key incident information including location data to first responders appears to be an important component that could significantly influence their response strategy.

2.3 Requirements for becoming a volunteer

According to (Stukas et al., 2014), a volunteer is defined as someone who works under non-profit means that is primarily aimed at benefiting certain groups or the lives of others. There are no significant requirements for becoming a volunteer, especially for non-critical tasks like food distribution, bloodletting, or other various charitable works. However, for medical volunteering organisations that involve real lives at stake, some conditions must be met before one could be identified as a registered volunteer. For instance, it is required that one should be at least 18 years of age and must have relevant certifications or referrals from approved health organizations. Even other volunteering roles require one to have at least a police clearance certificate to prove that one is a law-abiding citizen without a criminal record. This would necessitate obtaining approval paperwork from particular organizations (GoodSAM, 2022).

2.4 Impact of Volunteers in Emergency Response

It has been said previously that volunteering efforts have had an impact on emergency response and disaster management (Dostál and Balarinová, 2013; Stroop et al., 2020). One of the major ways that they have been deemed to be very useful is in their response time. This is because nearby volunteers are theoretically more likely to attend to the needs of a victim within a shorter time frame compared to the emergency services in certain scenarios. Additionally, according to (Dostál and Balarinová, 2013) there are economic benefits seen with the integration of volunteers within the emergency response services. The utilization of volunteers has been proven to reduce operational costs. This could be due to the notion that most volunteers are willing to offer their services without fees involved.

2.5 Emergency Volunteering Applications for Emergencies Response

With a key intention of assisting the Emergency Medical Services (EMS), few countries have implemented existing medical volunteering mobile platforms that focus on both alerting and dispatching nearby Volunteer First Responders (VFRs) to emergency locations. The main goal suggests that the nearby VFRs are meant to assist by closing the time gap needed for medical aid to be rendered to the victim. Both studies (Ogorevc and Lončarevič, 2014; Stroop et al., 2020), have employed the idea of dispatching nearby volunteers to assist OCHA cases (Out-Of-Hospital Cardiac Arrest). It was seen that it did not only just reduce the aid's response time but also improved the overall patient's hospital discharge rate. The section below would be going over some of the emergency volunteering platforms that have been both utilised and developed.

2.5.1 Existing Volunteering Platforms

2.5.1.1 GoodSAM

GoodSAM is a medical volunteering mobile platform that is currently being integrated into the EMS response for countries such as the UK, New Zealand, and Australia. This platform enables VFRs to assist with primarily Out-Of-Hospital Cardiac Arrest (OHCA) related cases. The way it works is that when a call is placed to the emergency services if the request has been identified to be an OHCA-related case, the GoodSAM system automatically gets triggered. This subsequently sends an alert to a VFR located within a certain radius from the specified emergency location to attend to the emergency incident. Some of the additional features of the GoodSAM app include a push-to-talk communication system and a map view which visualises all available electronic defibrillators located within a given area.

In order to become a registered volunteer within this platform, it is a requirement that one should possess a valid medical certification or qualification which must be properly verified. It is said that there are over 10,000 volunteers registered within the platform just in the UK alone (Smith et al., 2022). It was specified that this service was not aimed to replace the local emergency services but to supplement them by providing additional support.

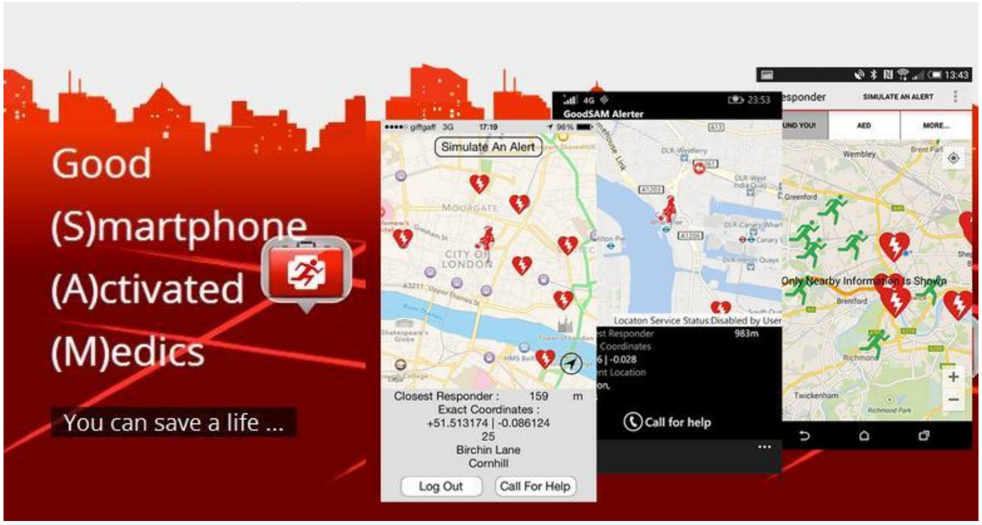


Figure 1 – GoodSAM Platform (GoodSAM, 2022)

2.5.1.2 iHelp

iHelp is an emergency volunteering platform that was started back in 2011. This platform was created to reduce the high mortality rate due to cardiac arrests by building a network of medically trained volunteers or VFRs who are responsible for both helping nearby victims and aiding the local population with defibrillators distributed within different locations. The iHelp Mobile application was designed to alert users such as family, friends, and even trained medical personnel that are within 300 meters of the victim’s location. Additionally, this mobile application also comes with some in-built useful features such as first-aid guide instructions which aim to provide useful training for various types of emergency scenarios. There is also a built-in map view which gives an overview of where to locate certain services such as hospitals, police stations, fire stations, and the nearest defibrillator.

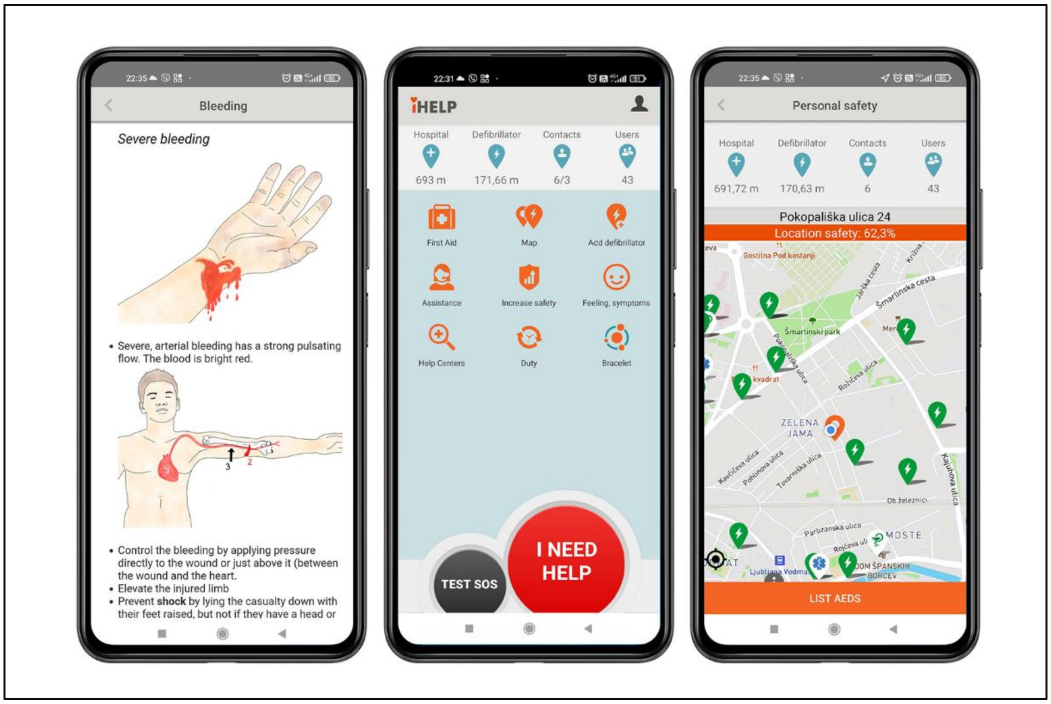


Figure 2 - iHelp Screen Overview (iHelp, 2022)

In situations wherein the user has no direct access to a mobile phone, the iHelp bracelet is quite useful or comes in handy. A long press of the given button triggers the SOS message through a Bluetooth connection established with a mobile phone placed within 20 meters which then subsequently sends an SOS message to the relevant individuals. With the current advancement in technology, there are newer versions of the iHelp bracelets that have been released which come with integrated sensors which are capable of both detecting health issues in the body and automatically sending SOS alerts when needed (Ogorevc and Lončarevič, 2014).

Figure 3 - iHelp Bracelet

2.5.1.3 Assist-Me

(Mallat and Abdulrazak, 2015) developed Assist-Me which is an emergency volunteering mobile application that was specifically aimed at elderly people. This platform aimed to provide aid for four major categories: (1) Falls, (2) Wandering, (3) Health, and (4) Crime. The system uses a Fuzzy-Logic-based selection engine which creates an ordered list of ideal volunteers based on predefined parameters such as medical experience and the distance from the incident location. Once the selection phase is completed, each of the selected volunteers on the list is then alerted in a particular order regarding an emergency incident. If the volunteer is unable to connect to the internet, the system can send SMS messages.

2.5.1.4 A-SA SOS

Relatively like the Assist-Me platform, the A-SA SOS is an IoT-based volunteering help mobile application that was specifically aimed at the elderly living in urban environments located in Thailand. This project allowed the elderly to request emergency help by using a mobile phone or an optional smart device. The entire system works by having an elderly trigger an emergency alert which as a result sends a request to the cloud service, subsequently identifying the most appropriate volunteer profile located within a 1.5Km radius from the help request location. The optional smart device comes with a single button that may be triggered during an emergency for the elderly who do not have smartphones.

Overall, it was determined that this system considerably shortened the time required for assistance to reach elderly patients. According to the given estimates, it was said that the platform removed an average of 3 minutes from the standard emergency response time within the region of operation (Intawong et al., 2021).

2.5.2 Summary

It was seen from previous studies that there are currently already existing volunteering platforms, one of which is being integrated with the emergency response services. Among the existing platforms that were examined, it was seen that the majority are more inclined in providing medical-related assistance, particularly with OCHA-related incidents.

2.6 Overview of Relevant Technologies

This section seeks to provide an overview and description of some of the existing technologies that are relevant for the implementation phase.

2.6.1 Location Based Services

Location-based Services (LBS) are generally defined as any system, regardless of the technology which is meant to relay information to a user, or a device based on its geographical location (Aloudat and Michael, 2011). This primarily consists of technologies such as GPS, Wi-Fi, Bluetooth Low Energy (BLE), and even mobile cellular network systems (Gustafsson and Gunnarsson, 2005).

2.6.1.1 Global Positioning System (GPS)

The Global Positioning System (GPS) is a global navigation system that was developed back in the 1970s by the US Department of Defence (DOD). Initially, it was intended for military uses however it was then later made available to the public, freely accessible by anyone. There is no doubt that GPS systems have played an important role in our daily lives and its applications are countless. Currently, it is widely adopted in various areas ranging from transportation, agriculture, surveillance, and weather predictions to search and rescue (Abulude et al., 2015). The primary way GPS systems work is by having multiple (5+) satellites located approximately 4000 miles above the earth's surface broadcasting radio signals to GPS receivers. Through a procedure of pre-defined mathematical calculations, a precise geographical location could be triangulated into a series of geographic coordinating systems composed of longitude and latitude with accuracies ranging between 10 to 15 meters (El-Rabbany, 2002). This system enables any mobile device with a GPS receiver to be actively located or tracked irrespective of where it is located anywhere in the world. However, it cannot be used for indoor navigation because its radio signals are unable to penetrate through walls or obstructions (Jekabsons et al., 2011).

2.6.1.2 Wi-Fi Location Positioning

Mostly known as Wi-Fi fingerprinting, this is the process in which Wi-Fi systems are used for indoor location tracking. It functions by using multiple Wi-Fi access points and by having their signal strengths called the Received Signal Strength (RSS) measured, which is used to enable indoor location tracking. This solution requires no extra hardware or sensors to be installed into the mobile device (Jekabsons et al., 2011).

2.6.1.3 Cellular Location Positioning

Cellular-based location tracking is the process of locating a mobile device based on radio signals being received from various cellular towers. This is done by analysing the signal strength of neighbouring cell towers, by doing this both the phone's location and its velocity could be triangulated. This system is ideal in an urban or densely populated environment where there is wide coverage of mobile cellular towers (Trogh et al., 2019). When compared to GPS in terms of its accuracy, it was seen that GPS systems offer far better location accuracy (Hellebrandt and Mathar, 1999).

2.6.1.4 Bluetooth Location Positioning

Bluetooth is a short-range communication system that enables two or more devices to pair and share content. The range of its connection signal reach is approximately 10 meters. Bluetooth location tracking works by having multiple Bluetooth-enabled beacons placed in designated areas in an indoor environment. These devices are normally configured to a "scan mode" which constantly tries to scan for nearby Bluetooth devices. The beacon's Received Signal Strength Index (RSSI) is used to calculate the distances across the devices. Once a Bluetooth-enabled mobile device is detected within a given area, it responds to the beacon by identifying itself, which then signifies that the device is within the room associated with the beacon (Hays and Harle, 2009).

Table 1 - Comparison of Location-Based Services

Location-Based System	Indoor Navigation	Outdoor Navigation
GPS	Highly ineffective in an indoor environment due to lack of radio signal penetration caused by physical obstructions.	Most ideal for outdoor areas without any major obstructions. It has an accuracy ranging from 10-15 meters.
Wi-Fi	Best in an environment where there is wide Wi-Fi coverage. However, it is ideal for indoor environments.	Works in an environment where there is wide Wi-Fi coverage.

Cellular	Offer better indoor location accuracy when compared to GPS systems	Can be used for outdoor location tracking, however not as accurate as the GPS counterpart.
Bluetooth	Most recommended for indoor use.	Not ideal for outdoor use unless there are numerous Bluetooth-enabled devices located close by.

2.6.2 Internet of Things (IoT)

Internet of Things (IoT) is defined as the network between devices such as sensors, computing gadgets, and software applications. It is a networking paradigm that has gradually been gaining in popularity over the years. As of 2020, it has been estimated that there are currently over 10 billion devices adopting IoT; it is projected that this number would significantly increase as the years progresses (Oracle, 2022). The primary aim of IoT is to exchange information amongst different devices to achieve a common goal via the Internet. This technology has had a significant impact on our daily lives. It has enabled people to interact with devices and applications in a much smarter and more intelligent way. Currently, different sectors are utilizing IoT such as transportation systems, logistics, automation, and manufacturing (Kumar et al., 2019; Oracle, 2022).

2.6.3 IoT Architecture

According to (Sethi and Sarangi, 2017; Kumar et al., 2019), 5 layers must be present within current IoT systems which are the perception, network, middleware, application, and business layers.

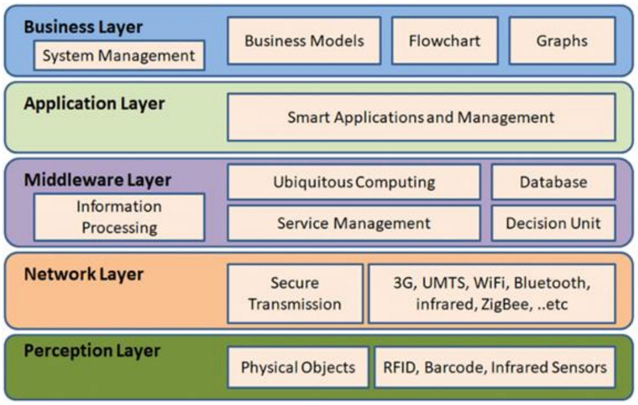


Figure 4 - IoT architecture Layers

Starting at the lowest layer level of the IoT architecture, the Perception Layer is normally defined as the physical layer. This layer is largely composed of sensors or devices such as RFID, scanners, and cameras which are meant to collect information regarding its surroundings. The Network Layer is the medium at which the information or sensor data gathered from the perception layer is then transmitted out for data processing. The data transmission channel could be in different mediums such as cellular, Wi-Fi, Bluetooth, Near-field Communication (NFC), Local-Area Networks (LAN), etc. The Middleware layer is also known as the processing layer. This layer is responsible for both analysing and processing all the information that is being received from the network layer. This layer is also responsible for making decisions which depend on the information being provided by sensor devices. The Application Layer is responsible for the network protocol being utilised to facilitate the transfer of information across different devices. Some of the commonly used communication protocols are MQ Telemetry Transport (MQTT), Constrained Application Protocol (CoAP), and the Advanced Message Queuing Protocol (AMQP) to name a few. Lastly, the Business Layer oversees the duty of managing the entire IoT system including the different layers. The business layer visualizes the data and statistics obtained from the application layer and uses this information for further plans and development.

2.6.4 Summary

An overview of related technologies to be used for implementation was evaluated. Different variations of existing location-based services were briefly touched upon such as the GPS, Wi-Fi, Bluetooth, and Cellular systems. Furthermore, it was concluded that the GPS is seen to be the most ideal location service to be used for the actual implementation due to its high accuracy in an outdoor environment. In terms of communication across different systems, the IoT infrastructure looks to fit well for this research project.

3 Methodology, Design, and Implementation

Based on the understanding derived from the literature review, this chapter discusses the methods used for the development and implementation of the given prototype. This project aims to both design and implement an emergency response system that would enable a registered volunteer to tend to nearby emergency incidents. To accomplish this primary goal, a series of guidelines or processes should be followed entirely towards its completion.

3.1 Methodology

3.1.1 Software Development Process

There are several commonly used software development approaches. The software development lifecycle that has been used for this research project is the Iterative Software Development Methodology. This software development framework is primarily focused on developing software systems over small iterations while adding new features after every cycle. In the initial stages, it isn't required to have all their requirements in place which makes it effortless to adapt or modify it throughout the project’s lifecycle (Okesola et al., 2020). Figure 5 as seen below demonstrates how the iterative software development model flows.

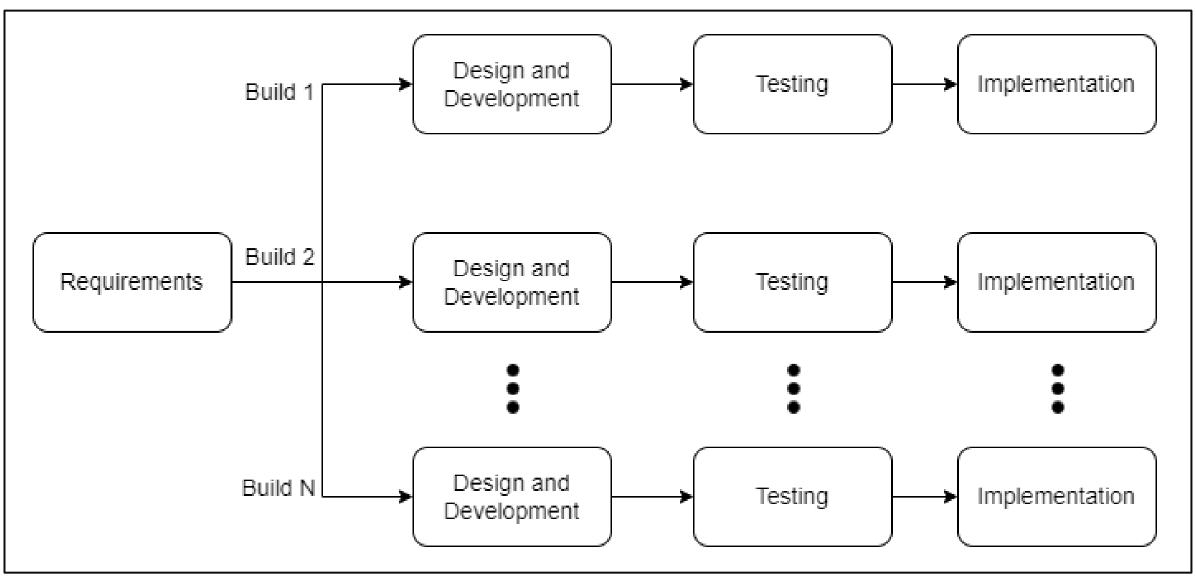


Figure 5 - Iterative Software Development Model

3.1.2 Software Application Requirements

For this project, there would be a need to develop two mobile applications and one web interface. The initial mobile application would be designed for the volunteers, allowing them to respond to nearby emergency requests. Meanwhile, the second mobile app will be developed so that the users or victims could send emergency help requests. For both software systems to be able to communicate efficiently with one another, there would be a need for a central backend server with a shared database environment for effective data communication. Lastly, a web-based admin panel would be needed to verify the credentials of each volunteer that registers through the mobile application. In terms of additional software requirements, a critical element that must be developed is a "volunteer matching" system that is meant to identify the best-suited volunteer for a given emergency incident based on predefined requirements.

3.1.3 Project Scope

This study primarily focuses on the prototype development of an emergency volunteering platform that would enable volunteers to assist with nearby emergency help requests. To avoid the issue of going out of project scope, a few limitations or constraints must be defined. The primary goal of this project would be to develop both mobile and web systems with a central database server. The other areas that could be further improved upon would be mentioned in the future recommendations section of the paper.

3.1.4 Evaluation of the Software

Software evaluation is important to determine both the usability and functionality of the mobile application that would be developed. Hence, two kinds of assessments will be conducted. The first will be a heuristics evaluation which aims to examine the user interface design on the mobile application according to defined user design principles. Afterwards, the second round of evaluation would be in a form of an online survey given to a limited, tightly controlled group of users to acquire their feedback and their overall evaluation while using the volunteering mobile app. The appendix section contains the list of questions given within the survey.

3.2 Design

3.2.1 Selection of Technology

3.2.1.1 Firebase

Firebase is an application development platform that is currently being maintained by Google. It is identified as a Backend-as-a-Service platform which enables developers to easily build highly functional mobile and web apps eliminating the need to develop complex server-side infrastructure for highly scalable web or mobile applications. One of the main advantages seen with using Firebase is that there are multiple services readily available within the platform that could be utilised with minimal difficulty. Listed below are some of the services that are provided within the Firebase platform. (Rawal, 2017; Khawas and Shah, 2018; Firebase, 2022a).

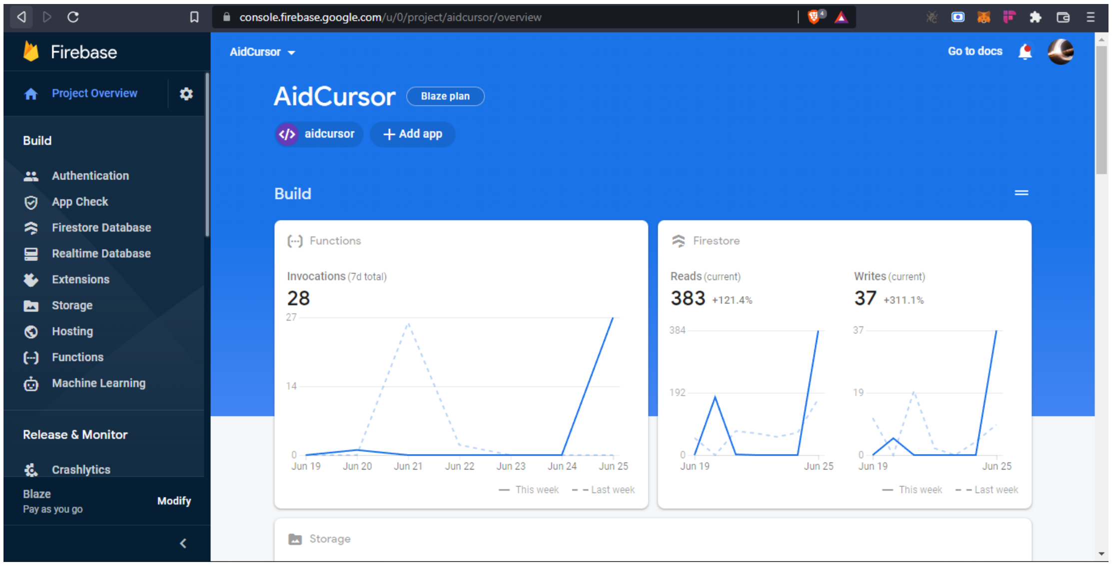


Figure 6 - Firebase Console

3.2.1.1.1 Database Module

Firebase offers two variants of (non-relational) NoSQL cloud databases: (1) Real-Time Database (RTD) and (2) Firestore. RTD is a database system that stores real-time data in a JavaScript Object Notation (JSON) format. As a result, any change made to the database is automatically reflected on the user’s end without additional configuration. On the other hand, Firestore is a much more scalable and structured NoSQL database compared to RTD, also capable of syncing real-time data across different devices. For the implementation of this project, the Firestore database was chosen as the preferred database due to its more enhanced features available.

3.2.1.1.2 Authentication Module

The Firebase Authentication service enables users to sign in and authenticate their accounts. In addition to the traditional email and password configuration, it also offers third-party authentication methods using Google, Facebook, Microsoft, or Twitter accounts.

3.2.1.1.3 Storage Module

The Firebase Cloud Storage is a storage service, primarily used to store different types of assets and files such as images, videos, audio, documents, and other media contents. Integrated into it, is a highly secure communication transfer link that is reliable irrespective of network connectivity speeds.

3.2.1.1.4 Analytics Module

Firebase Analytics is an assessment tool that offers reports on how people interact with a given software application. This analytics report is particularly aimed at providing the necessary data that would help businesses in their decision-making process.

3.2.1.2 Google Maps

Google Maps is a map service developed by Google. It is equipped with various features such as map navigation, satellite imagery, and street views. It is compatible with a variety of platforms, including Android, iOS, and web-based systems. Additionally, Google Maps also offers various API (Application Programming Interface) services that allow integration with other third-party software applications. Subsequently, this enables developers to be able to utilise its services in several ways across a range of software systems (Google, 2022a).

3.2.1.3 ReactJS

ReactJS is a JavaScript-based WEB framework developed by Facebook and is commonly used to build both fast and highly responsive web applications. The advantage of this web framework is that it allows reusable UI components to be employed within the application. The React framework also goes beyond the development of web applications. There is another variant that is aimed toward the production of mobile applications called React Native. React Native is a hybrid mobile app framework that supports the development of both android and iOS applications under the same codebase, but with a few platform-specific configurations (Altexsoft, 2020; React, 2022).

3.2.1.4 Expo Notifications

Expo Notifications, also called Expo's Push API is a service developed by the Expo team used to deliver push notification messages to mobile devices. The notification process is as follows: An Expo token that has already been generated and is unique for each device is then sent to the Expo Notification API in a POST request payload which as a result causes a push notification to be delivered to the corresponding device (Expo, 2022).

3.2.2 System Framework

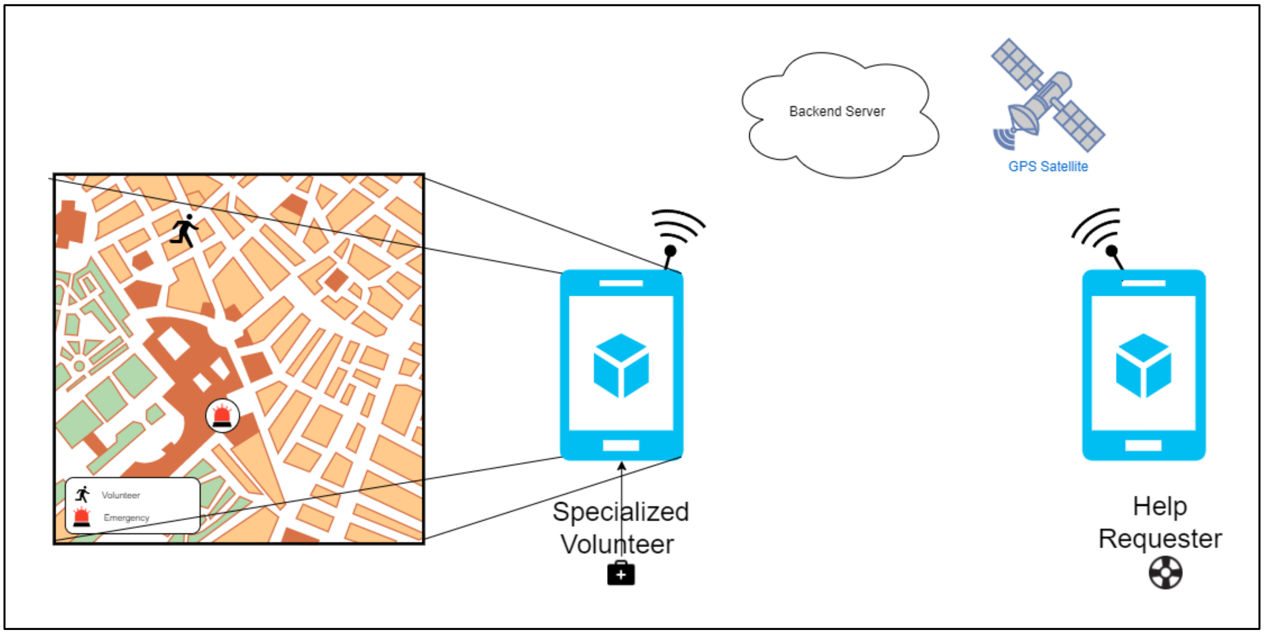


Figure 7 - System Framework Overview

Figure 7 above is a systematic framework that provides a high-level overview of how the entire system ought to function. The process starts when a user requires emergency assistance, he/she would then need to send out a help request from the mobile application containing key information such as a description of the emergency assistance needed and the current GPS location coordinates. This data is then automatically sent to the backend server which subsequently performs the necessary operations required to both identify and notify the ideal sets of volunteers that would be needed to assist the user. Furthermore, once alerted, the volunteer should be able to respond to the help request with the aid of the navigation system which is meant to provide the most ideal route required to reach the destination of the emergency incident.

3.2.3 Prototype Designs

Before proceeding to the implementation phase, some of the initial mock-up designs of the volunteer's mobile app were drawn out. The figure below showcases some of the prototype mock-up designs that were designed. Having the prototype designs drawn out first-hand helped sped up the time needed in developing the mobile app. However, it was expected that some of the designs that were initially drawn, would not make it to the final implemented prototype owing to continual revisions throughout the development lifecycle.

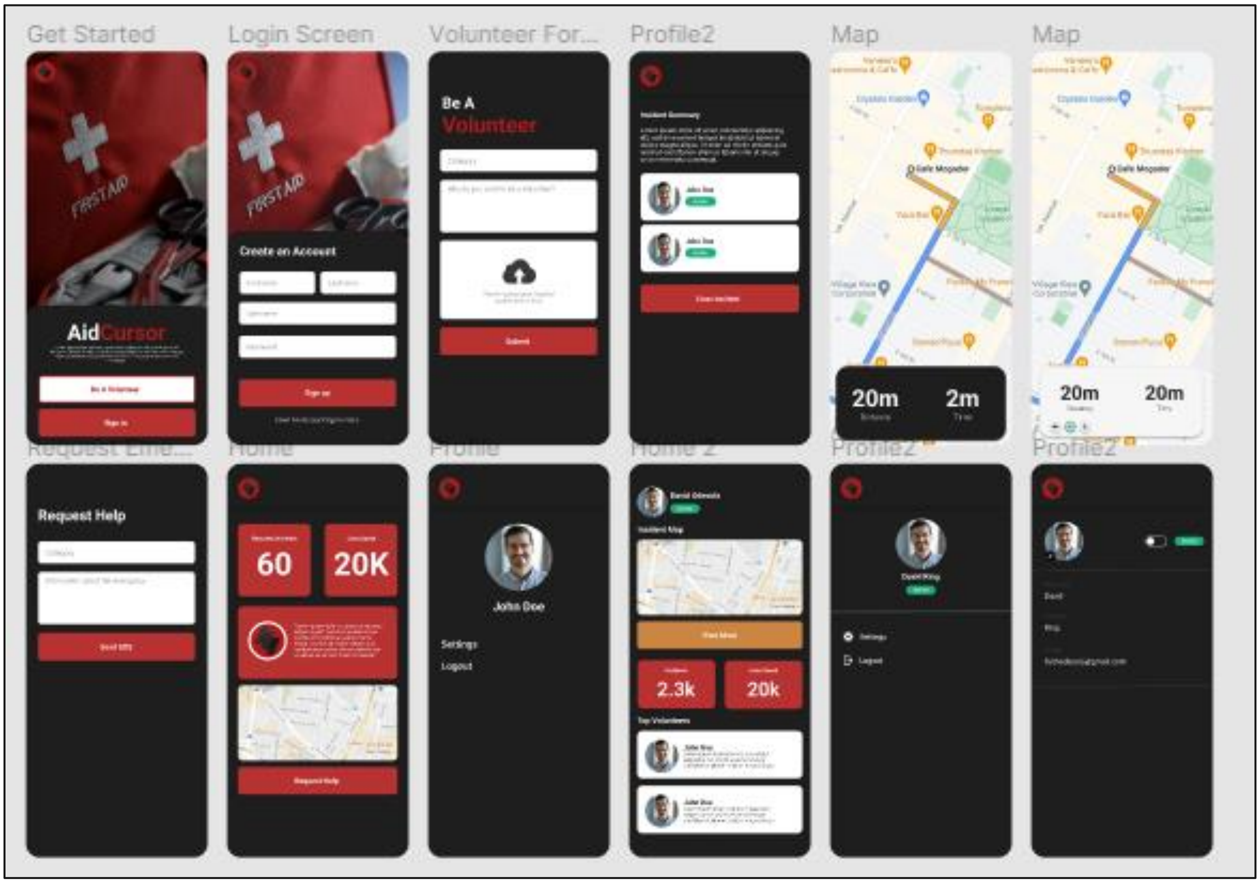


Figure 8 - Mock-up Designs

3.2.4 User Roles

Three distinctive software applications are being developed for complete user interactivity across the system. Furthermore, there would be three different categories of users involved which are composed of the volunteer, the user (help requester), and the administrator. The volunteer is the one who would respond to help requests, the user (help requester) is the one that would be generating the help requests and the administrator would primarily complete the assignment of either accepting or rejecting the volunteering requests being received. Across all applications, only the proper user must have access to the relevant user account. Figure 9 below gives an illustration of the three distinctive user roles involved within the entire system.

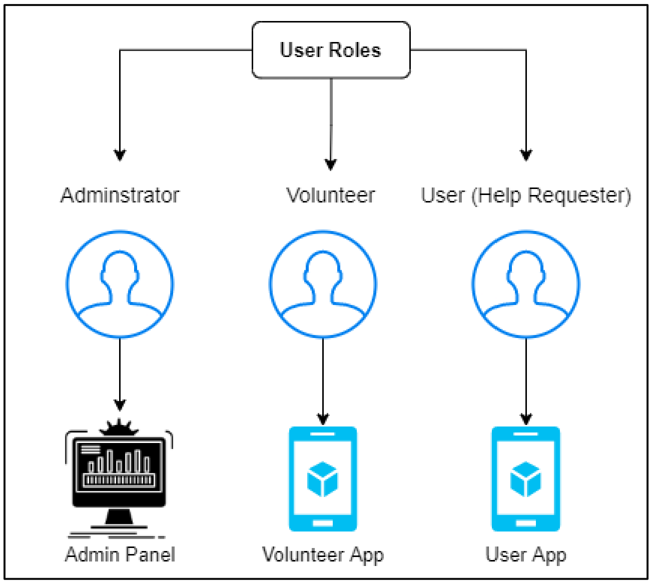


Figure 9 - User Roles

3.2.5 Registering as a Volunteer Process

Based on what was seen in the literature review it is important to have a verification procedure in place to guarantee that each volunteer on the platform has been thoroughly vetted and meets the requirements.

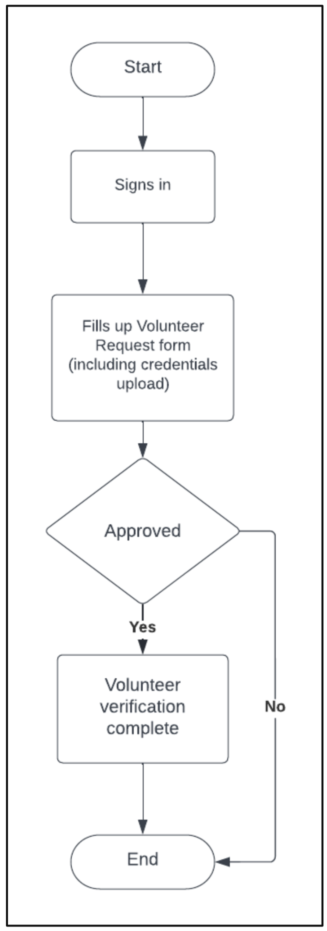


Figure 10 - Volunteer Verification Process Flow

Figure 10 above is a flow chart defining the process by which a volunteer gets registered. The process starts when the volunteer signs in, which then should redirect them to the “Volunteer Request” form. Subsequently, the volunteer will then fill up the form, requiring them to upload their credentials as part of the verification process. It is only after their request has successfully been approved that the volunteer could proceed out of the verification stage and be defined as a registered volunteer within the platform.

3.2.6 Database Design

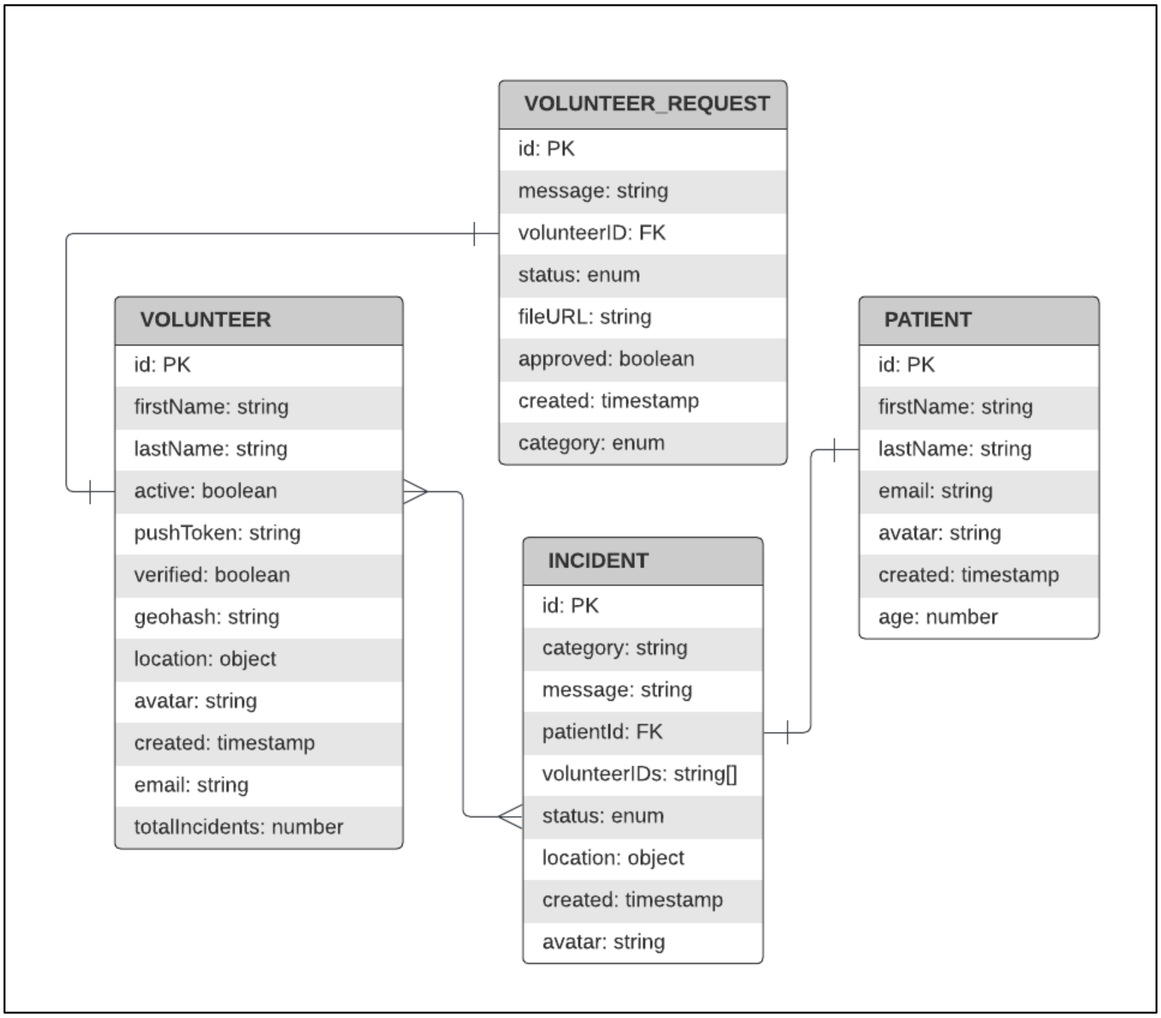


Figure 11 - Entity Relationship Diagram (ERD)

The diagram in Figure 11 is an ERD Diagram that shows the relationship across the different entities or tables within the NoSQL database. The **Volunteer** table contains all the necessary information associated with a volunteer. Linked to the table at the top is the **VolunteerRequest** table in which the volunteer's request information is stored. It is configured that a single volunteer can only have only one linked volunteering request which must get approval from the system administrator. The **Incident** table contains all the relevant information regarding a given emergency help request. This table is directly linked to the patient table, which represents the user seeking emergency assistance. To identify the volunteers that are associated with each of the incidents, there is the "volunteerIds" field within the incident table which is composed of an array of strings containing the specified lists of volunteer ids.

3.2.7 Volunteer Match Making

It is essential that during an emergency, the most appropriate volunteers are notified. Hence, there are a series of steps that should be followed in order to identify the best-suited volunteer. Here are some factors that should be considered during the identification process.

- Emergency category
- Volunteer's location
- Volunteer's availability status
- Volunteer's Ranking
- Volunteer allocation limit per help request

3.2.7.1 Location

Determining the location and distance of a volunteer from the emergency origin location is one of the most important components of the matchmaking process. It is of great importance that the closest qualified volunteer gets notified during an emergency. To accomplish this task, there were a series of techniques being employed in identifying nearby volunteers within a given emergency location. For example, this could be done by utilizing both geohash and the Google Direction Matrix API.

3.2.7.1.1 Geohash

Geohash is a string prefix that is used to identify certain regions around the earth. The main idea is that each of the areas around the earth is split into designated segments. These segments are then each given unique values with which they are identified. Essentially, geohashes allow for the standard latitude and longitude coordinating values to be translated into a simpler Base-32 string value. The length of a geohash usually determines the precision of the bounding region concerning the location to which it refers. So, the longer the geohash string, the more precise the location point (IBM, 2021; Firebase, 2022b). The main benefit seen in using geohashes in conjunction with latitude and longitude coordinate systems is that it makes it possible to efficiently query documents stored within the Firestore database based on their geographical location on the planet (Firebase, 2022b).

Figure 12 - Geohash Segmented Regions (Movable Type Scripts, 2022)

3.2.7.2 Identifying the appropriate volunteers given their locations

As stated previously, volunteers that are being alerted to respond to emergencies must be located within certain regions relative to the origin of the emergency location. To achieve this, a series of procedures were followed. The first step was to calculate the bounds of a circle with the emergency location at the centre point. The bounding circle should filter out the list of active volunteers that are situated nearby the emergency location. Hence, the radius of the circle shouldn’t be too small or great because this variable could significantly affect the number of volunteers that get notified. Furthermore, after some initial experimentation and configurations, it was seen that a radius of 2 Kilometres will be sufficient. The distance between the emergency site to the ideal volunteer must be within the predetermined region limits. The Haversine formula was initially considered as a solution to calculate the shortest distance between 2 geographical (latitude and longitude) coordinates (Azdy and Darnis, 2019). However, there is no doubt that a journey between 2 destinations is highly unlikely to be in a uniform arc line considering other variables such as road networks and traffic patterns which this formula doesn’t take into account (Nguyen and Szymanski, 2012). For instance, a volunteer may be geographically close to an incident's location however, when considering other variables like road patterns, this might not be the case. Furthermore, after some additional investigation into other alternatives, it was seen that the Google Distance Matrix API can address these issues. The illustration in Figure 13 below demonstrates how the volunteers are being selected with both navigation and road routes being considered.

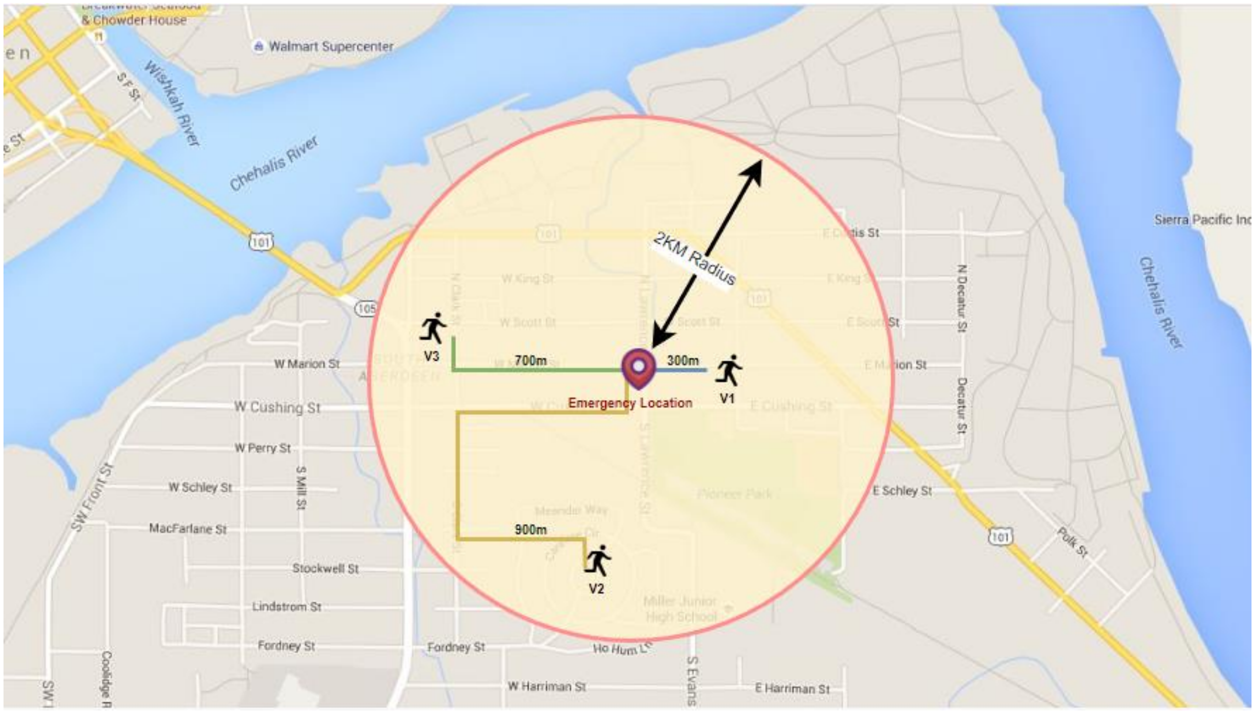


Figure 13 - Identification of 3 volunteers located within a 2KM radius

3.2.7.2.1 Google Distance Matrix API

Figure 14 - Distance Matrix API

The Distance Matrix API is a web-based service within the Google Maps platform which provides both distance and travel time estimation across origin and destination points. The major benefit is that it factors different modes of transportation, traffic patterns, and road networks into its predictions (Google, 2022c).

3.2.7.3 Maximum Volunteers Per Help Request

There could be numerous available volunteers located within range and it would be impractical to have all of them within an area to respond to a singular emergency help request especially if they are in great numbers. Furthermore, to address this possible issue, there would be no more than three volunteers selected to attend to any given emergency.

3.2.7.4 Volunteer's Ranking

During the selection process of the volunteer, it was of great importance to rank the list of potential volunteers based on predefined parameters. For the sake of simplification, the short-listed sets of volunteers would be ranked according to the number of accepted help requests. However, an alternative and more-advanced approach would be to implement the use of a more complex ranking system that would rank the volunteers according to their skill level.

3.2.7.5 Emergency Type

Every emergency case is distinct in both its attribute and nature (Mladjan and Cvetković, 2013). For this project, two categories are identified (Medical and Others). The “Medical” category is primarily aimed at health-related help requests ranging from cardiac arrests to bruises. Meanwhile, the “Other” category is aimed at the different variety of help requests. However, for the volunteer to understand the main specifics of the help request, there would be an added description field that would enable the user (help requester) to give in more details for better clarification or context for the volunteer’s understanding. Table 2 details some examples of incidents that could fall under these two categories.

Table 2 - Emergency Categories Table

Medical	**Others**
Cardiac Arrests	In-Person Directions Assistance
Choking	Neighbourhood watch
Bruises	Lifting assistance
First Aids	

3.2.7.6 Volunteer Status

The volunteer's availability status is another factor that should be taken into consideration during the volunteer matchmaking process. The volunteer should be able to adjust his or her availability status to either ONLINE or OFFLINE. Furthermore, even though a volunteer may have fulfilled all the other requirements if their availability status has been set to OFFLINE, they should not get selected or notified to attend to any emergency request.

3.2.8 Map Navigation

For the assigned volunteer to effectively navigate to the location of the emergency incident, there must be an integrated map navigation system. The navigation system should provide the most optimal route to the destination depending on the type of transportation or mode of travel. There are certain regions where only specific types of transportation are permitted. Hence, the navigation system must be able to adapt to various modes of mobility being used to travel. To accomplish this objective, there is the Direction API service available within the Google Map Platform.

3.2.8.1 Google Direction API

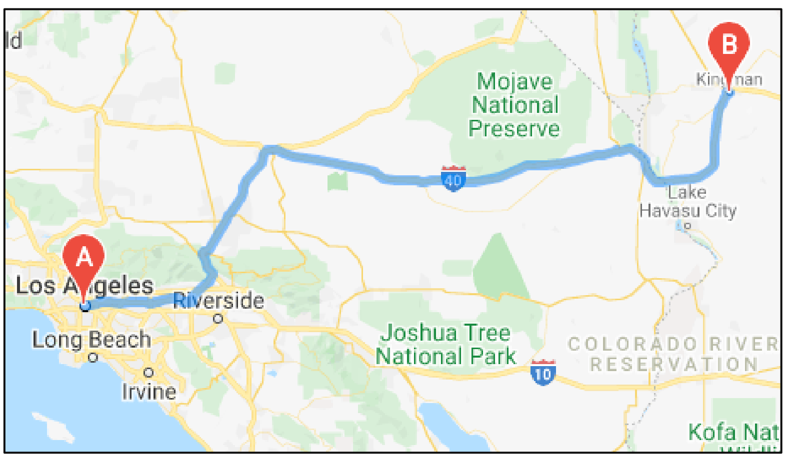


Figure 15 - Google Directions (Google, 2022)

The Directions API is a web-based service that is capable of providing directions across various destinations tailored to different transportation modes such as walking, driving, commuting, and biking (Google, 2022b). The directions derived from the API are outputted in a JSON (JavaScript Object Notation) or XML format. The usage of the Google Directions API would assist the volunteer in the navigation process by generating the most ideal route needed to reach the emergency location.

3.2.8.2 Map Navigation Process

The system process being carried out during the map navigation process is depicted in the flowchart in figure 16 below.

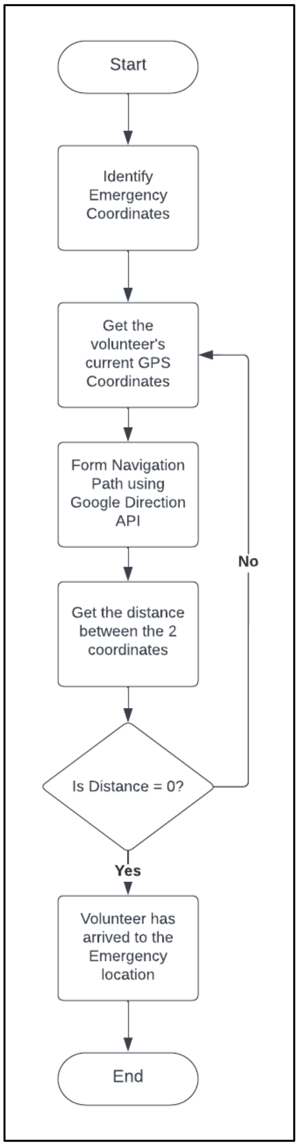


Figure 16 - Map Navigation Flow

The map navigation flow starts immediately after the GPS coordinates of the emergency location have successfully been identified. Subsequently, this is followed by fetching the current GPS location of the volunteer. Both these coordinates are then given to the Google Directions API, which as a result generates predefined route information between both locations. If the calculated distance between both coordinates is greater than zero, that means that the volunteer has not arrived at the destination. This process is constantly repeated, until the distance between both sets of coordinates equates to zero, indicating that the volunteer has successfully found the emergency location subsequently ending the navigation process.

3.2.9 Notification System Flow

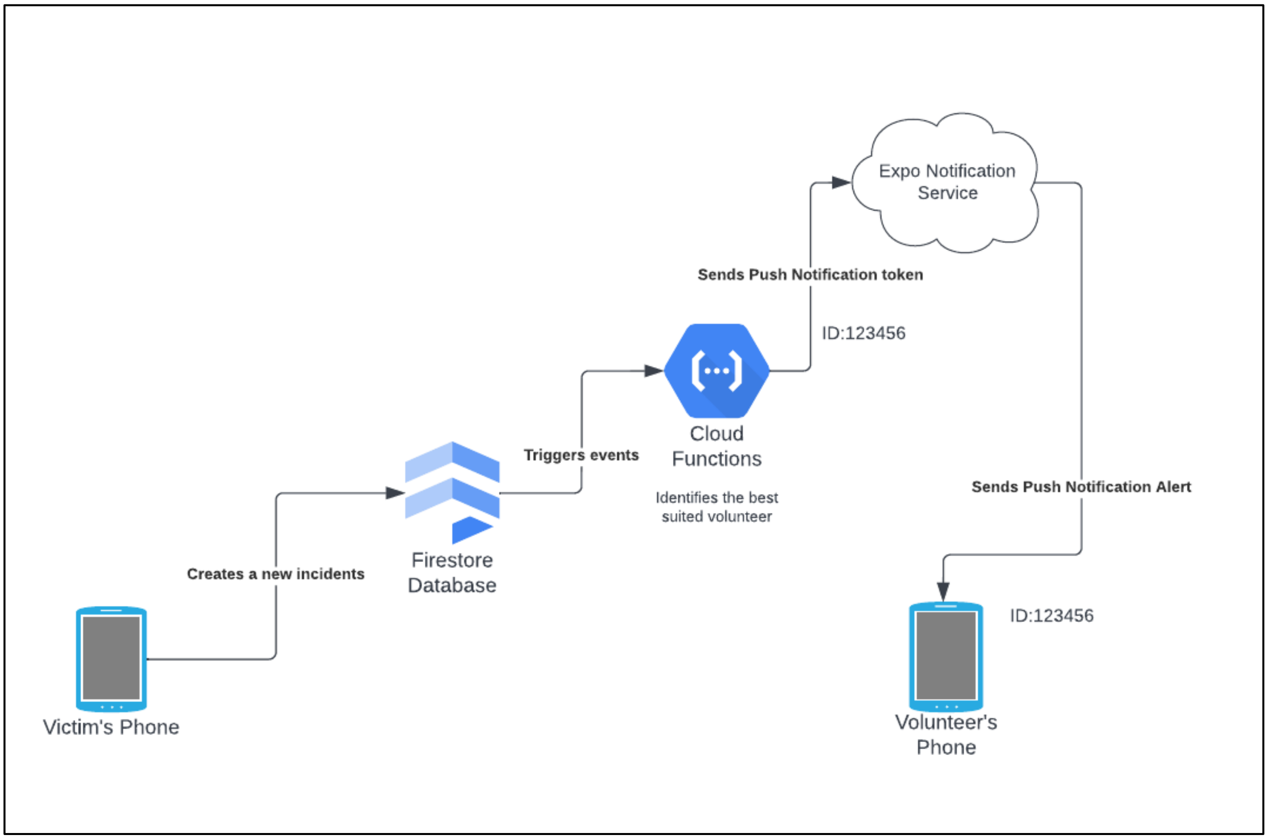


Figure 17 - Notification system design

Figure 17 shows the design of the notification system for implementation. The diagram describes the process by which real-time push notification messages are sent to volunteers. The process starts when a new incident is created and stored within the Firestore NoSQL database, automatically triggering the “onIncidentCreated” event within the Firebase cloud functions library. The “onIncidentCreated” function being executed within the cloud function contains the necessary logic needed to identify the most appropriate registered volunteer based on the predetermined factors. Once the relevant volunteers have been successfully shortlisted, the respective push Notification token ID which is unique to each volunteer’s mobile device is then extracted and sent to the Expo Notification API Service. This API service then sends out the push notification messages to the mobile device associated with the corresponding notification token ID.

3.2.10 Notification Interface

Once the request has been sent out by the Expo Notification API Service, a push notification prompt would be displayed on the device as seen in figure 18 below. When the notification popup card is pressed, there is an automatic redirection to the “Emergency Request” screen which allows the volunteer to see the help request in much greater detail.

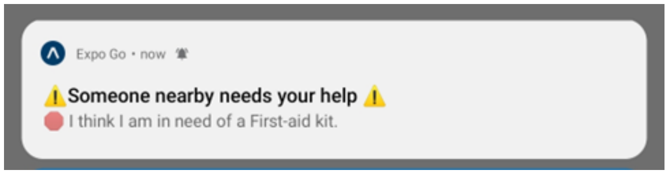


Figure 18 - Received Push Notification

3.3 Implementation

3.3.1 Volunteer’s App

3.3.1.1 Authentication Screens

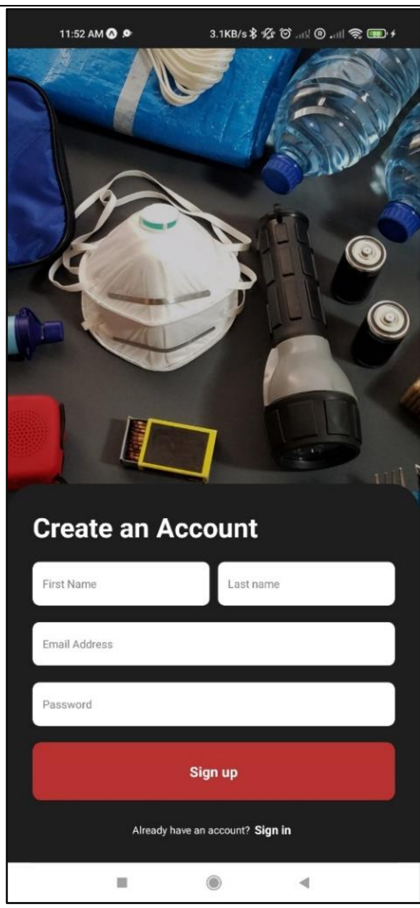
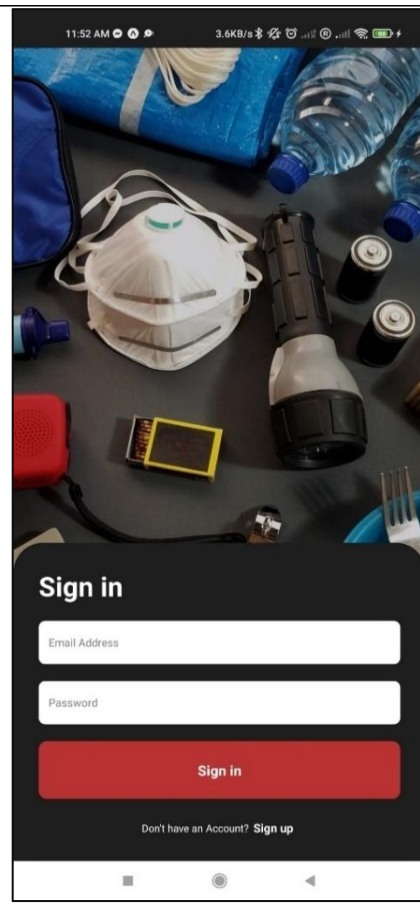
Create an Account	Sign in
	

Figure 19 - Authentication Screens

The figures above are the sign-in and sign-up screens that every volunteer would have to fill up to proceed further across the mobile app screens.

3.3.1.2 Volunteer Request Screen

Before one could officially become a registered volunteer, one would be required to be approved by the platform administrator. Hence, it would be necessary to fill up the Volunteer Request Form and provide some additional information as shown in Figure 20 below.

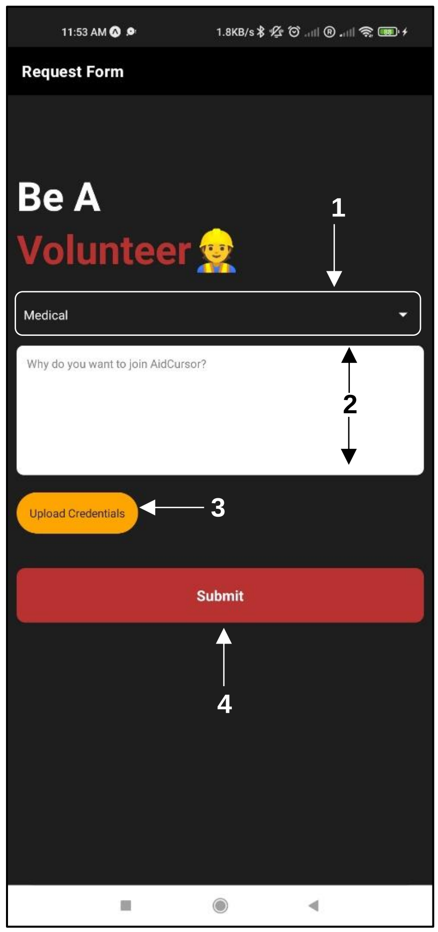


Figure 20 - Volunteer Request Form Screen

1. Emergency Category Field (Medical & Others)
2. Information Text Field
3. Upload Credentials Button
4. Submit Request Button

Upon registration, the emergency category dropdown field would allow the volunteer to decide on the category of emergency assistance that they would want to respond to. There are 2 categories within the dropdown list to choose from which are “Medical” and the “Other” categories. Further below is the information text field which is meant to allow the users to give additional information on why they would want to be a registered volunteer within the platform or anything that could provide further support for their volunteer request. The “Upload Credentials” button further below simply redirects to the File Manager. From there, the pending volunteer could select the necessary files that would be uploaded together with the request form. Once the given input fields are filled in, the request form could then be submitted, which is subsequently sent to Admin Panel for verification.

3.3.1.3 Home Screen

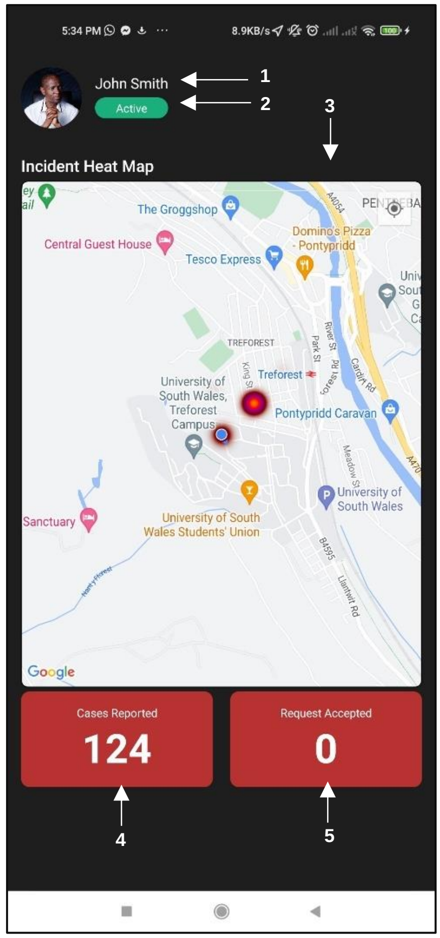


Figure 21 - Home Screen

1. Full Name
2. Availability Status
3. Incident Heat Map
4. The total number of incidents that were created.
5. The total number of Help Requests that were accepted

A volunteer may only view the home screen if the administrator has approved the volunteering request form via the admin panel. There are a few key components that reside on the home screen of the mobile app. At the top-left section of the screen is the volunteer’s profile information which includes the avatar photo, full name, and availability status. Placed over the profile image is an invisible clickable button which navigates to the volunteer’s profile screen.

Below the profile and the information section is the Heat Map[1] which gives a map visualization of past reported requests in conjunction with their corresponding location. The heat map is aimed to give a representation of the locations that are hotspots for emergency help requests. Furthermore, at the bottom left of the screen is a section that shows the total number of reported incidents within the platform. Next to it is a section that shows the total number of requests that the volunteer has accepted.

3.3.1.4 Profile Screen

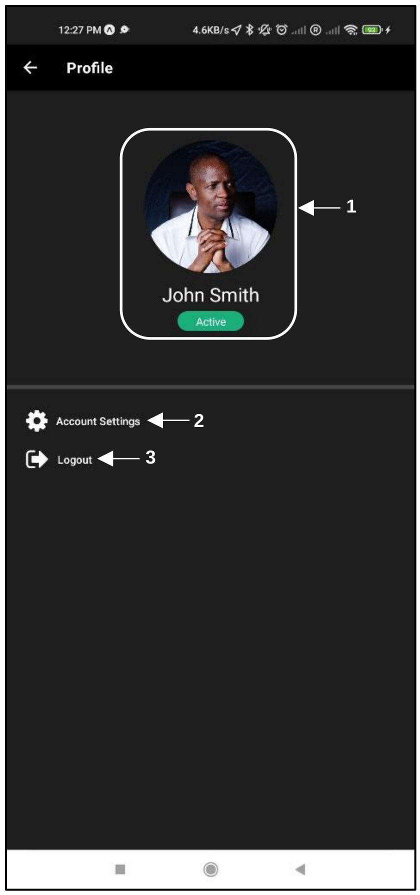


Figure 22 - Profile Screen

1. Profile information 2. Account Setting Button 3. Logout Button

[1] A heat map is a data representation technique used to present data in a form of colour.

Figure 22 is the profile screen which shows the profile information with account settings and the logout buttons. The “account settings” button redirects to the Settings screen when clicked. Meanwhile, the logout button simply logs out the user and redirects back to the authentication screens.

3.3.1.5 Settings Screen

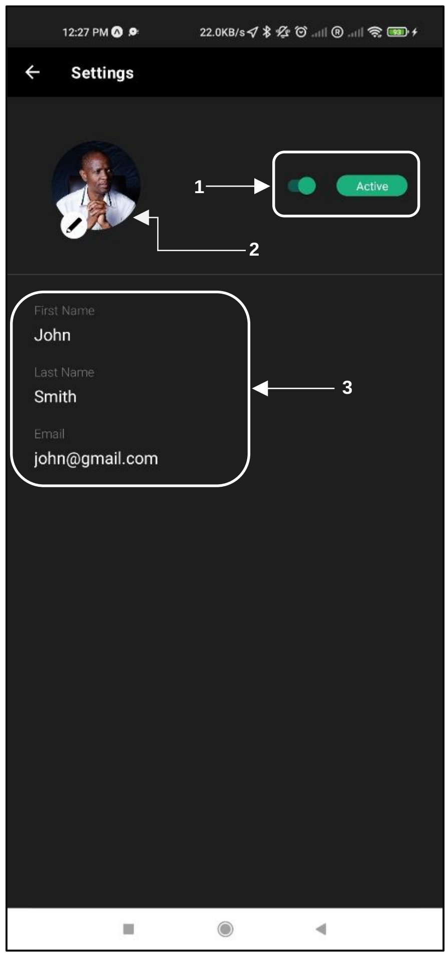


Figure 23 - Profile Settings Screen

1. Status Toggle 2. Editable Profile picture 3. Profile information

The Profile Settings screen is where the volunteer could change some of the profile settings. The toggle bar seen primarily allows the volunteer to toggle their availability status. When toggled to the right, it means that the volunteer is in an ONLINE state and is open to receiving nearby help request notifications. Meanwhile, when pressed again it is toggled to the left signalling that the volunteer is OFFLINE. When the profile is offline, both active database location updates and notifications for nearby emergency incidents are temporarily disabled.

Furthermore, the profile image can also be modified from this screen. This is done by clicking on the existing Circular image which then redirects to a file manager. The file manager enables the volunteer to select a profile picture from the local gallery of the mobile device.

3.3.1.6 Emergency Request Screen and Notification

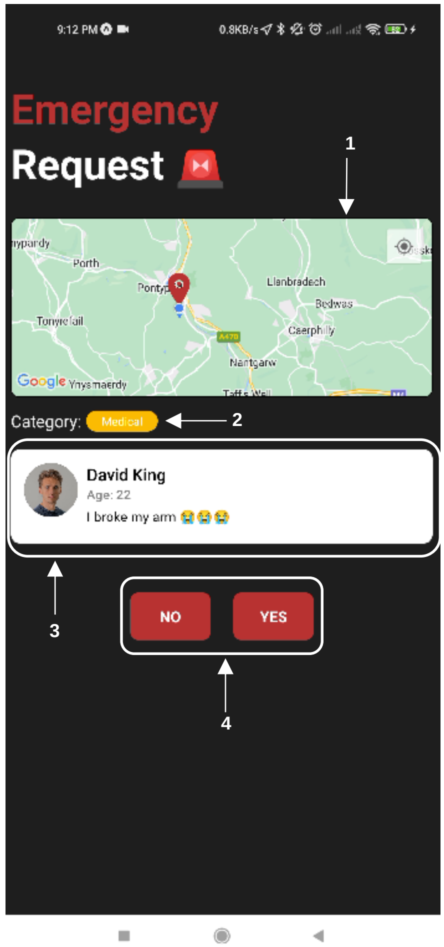


Figure 24 - Emergency Request Screen

1.	Mini Map
2.	Help request Category
3.	Card containing the help requester information including the description of the emergency
4.	(No & Yes) Decision buttons

When a push notification is triggered and clicked within the device, the volunteer is automatically redirected to the "emergency request" screen which is meant to contain more information about the emergency request being received. At the top section is a small-sized map which aims to show the distance between the emergency location and the volunteer's current location. The "Yes" and "No" buttons at the bottom of the screen allow the volunteer to decide on whether to accept or decline the help request. Pressing on the "No" Button just redirects back to the home screen. Meanwhile, a press on the “Yes” button redirects to the Map navigation screen.

3.3.1.7 Map Navigation Screen

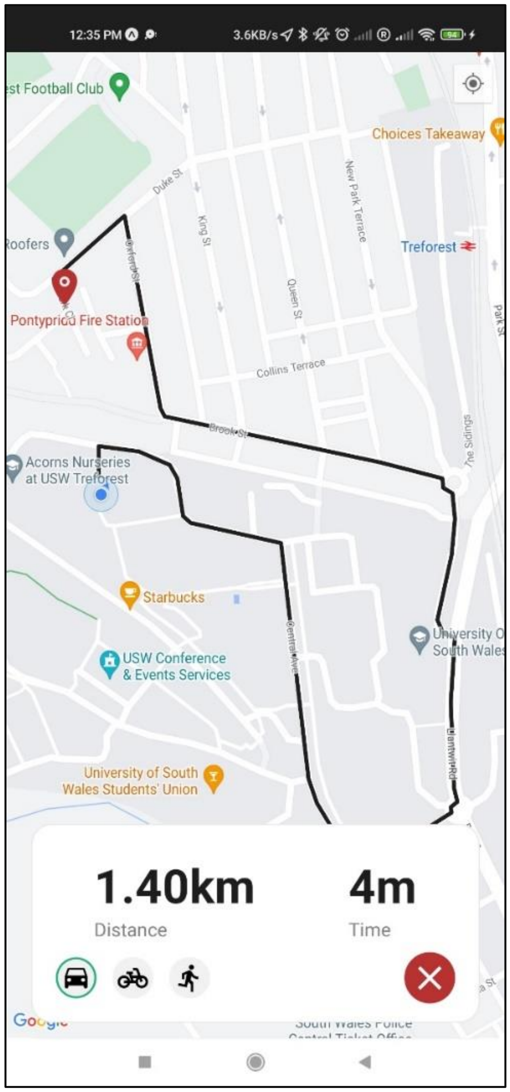


Figure 25 - Map Navigation Screen

The map navigation screen seen in Figure 25 above is aimed to assist the volunteer in accurately navigating to the location where the help request originated from. A predetermined path generated by the Google Directions API is integrated into the map view which provides directions to the destination based on the volunteer’s current location. The section below the map view shows some additional information regarding the entire planned journey. Both the total distance required to reach the destination and the total duration is shown. Additionally, three different travel modes could be chosen which are (1) Car, (2) Bicycle, and (3) Walking.

The path generated by the Google Directions API is primarily dependent on the selected method of travel. The generated route is automatically refreshed every five seconds. Lastly, the red button located at the bottom-right simply redirects back to the home screen, ending the navigation process.

3.3.2 User (Help Requester) App

3.3.2.1 Report Screen

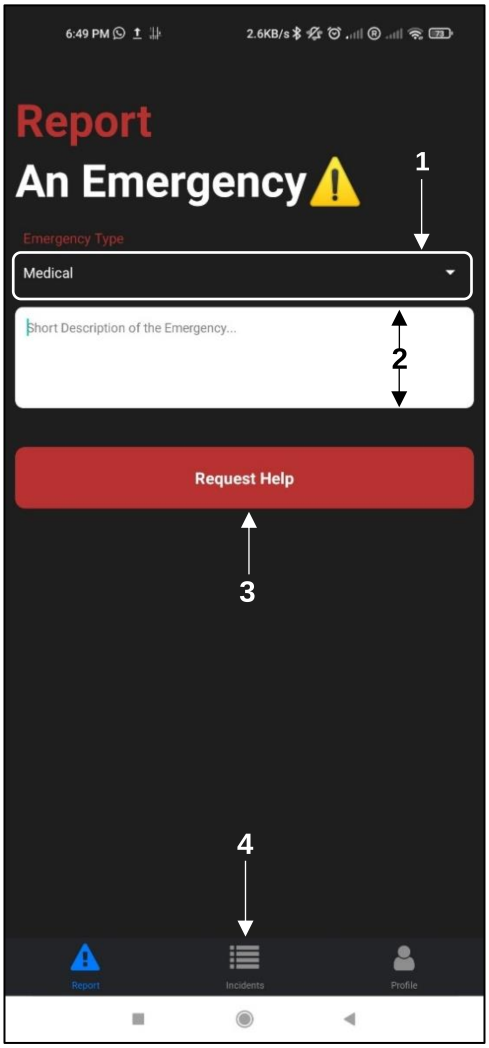


Figure 26 - Emergency Report Screen

1.	Emergency Category Dropdown Menu
2.	Emergency Description Text Field
3.	Submit Help Request Button
4.	Bottom Navigation Bar

The emergency report screen allows a user to generate a help request which is meant to be received by nearby volunteers. First, the emergency category must be chosen from the dropdown menu. Currently, 2 categories could be selected from the menu which are the “Medical” or the “Others” option. After the emergency category has been specified, a text input field is available for the user to be able to give some additional descriptive information regarding the nature of the emergency; after then the help request could then be submitted and relayed to the relevant volunteers available.

3.3.2.2 Incidents Screens

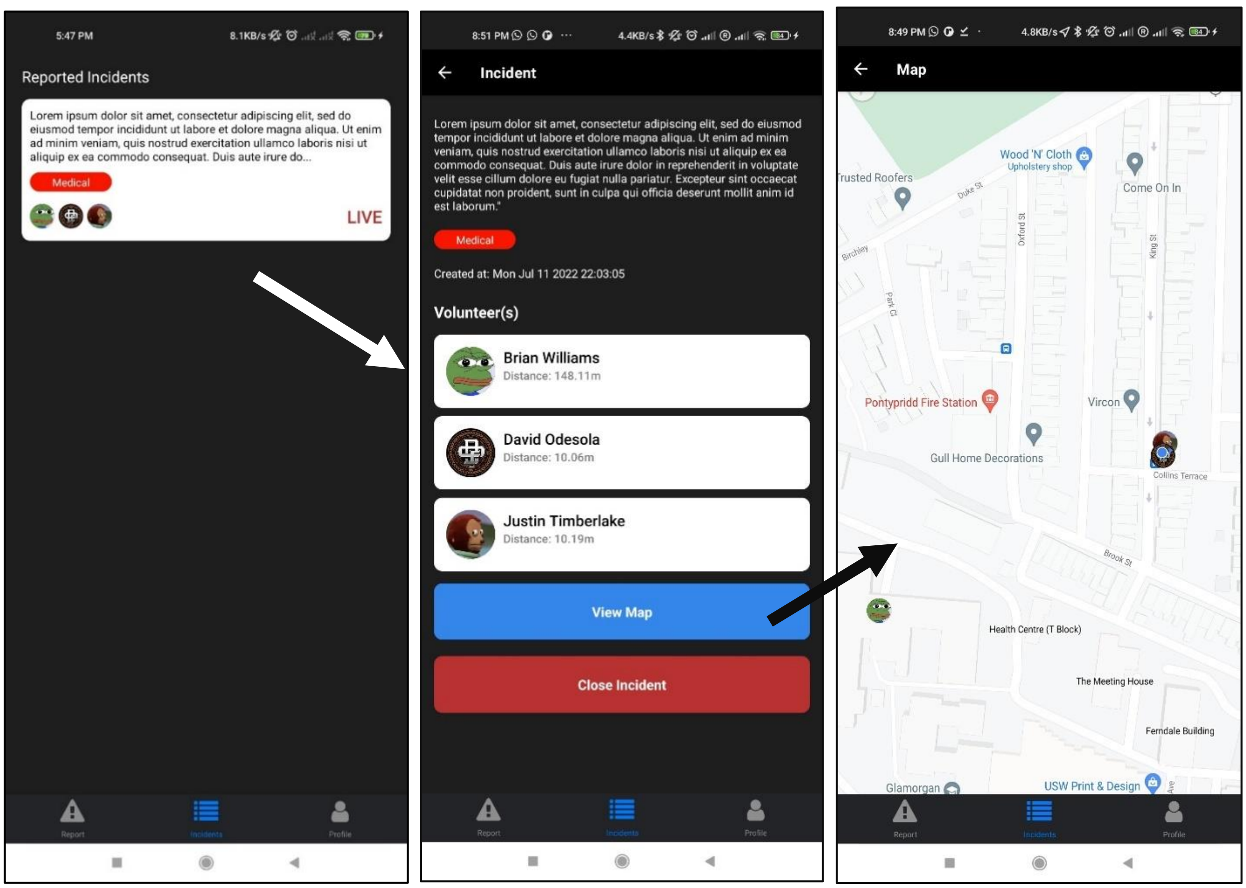


Figure 27 - Reported Incident Screens

Once an emergency help request has successfully been submitted, an incident card pops up within the “Reported Incident” Screen. The incident card simply contains the given information about the created help request that was submitted including the profile avatar of the volunteer that has accepted the request. When the incident card is pressed, there is an automatic redirection to a separate “Incident Info” screen which contains a more in-depth description of the request that was made. This screen would also contain the list of volunteers that have agreed to respond to the help request and includes their real-time distances from the emergency location in meters. Whenever a volunteer accepts the help request, the "View Map" button comes into view on this screen. This button redirects to a map view which shows all of the associated volunteers and their real-time GPS locations. The map view makes it easier for the person asking for assistance to see where the volunteer is currently located and how far away they are from their destination. Furthermore, after the necessary help has been tendered by the specified volunteer, there is an option to close the incident by clicking on the "Close Incident" button which simply just removes the incident card from the list screen.

3.3.2.3 Profile Screen

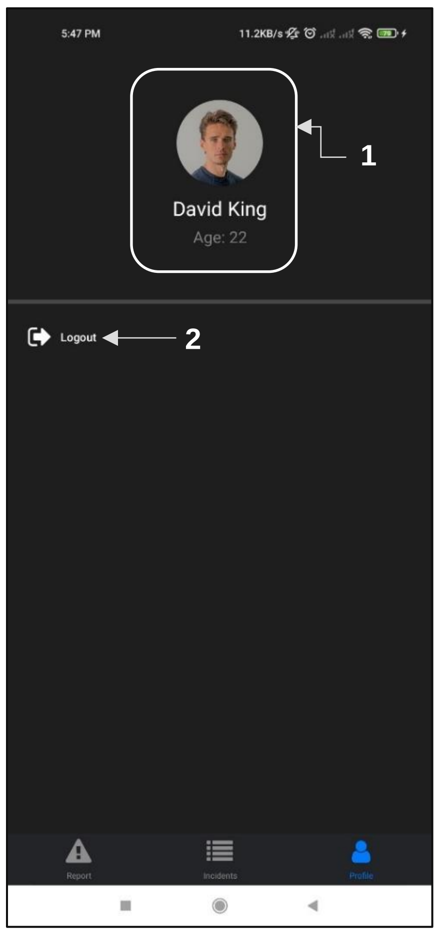


Figure 28 - Profile Screen

1. Profile Info 2. Logout Button

The Profile screen as shown in Figure 28 contains all the basic profile information about a user. The profile avatar could be updated by clicking on it, which then opens a file manager, enabling the user to select a preferred profile photo. Furthermore, at the bottom of the profile data section is the logout button which signs out the user and then redirects to the authentication screens.

3.3.3 Admin Panel

Once the volunteer request form has successfully been submitted by a volunteer, a request card is automatically displayed within the admin panel as shown below in Figure 29.

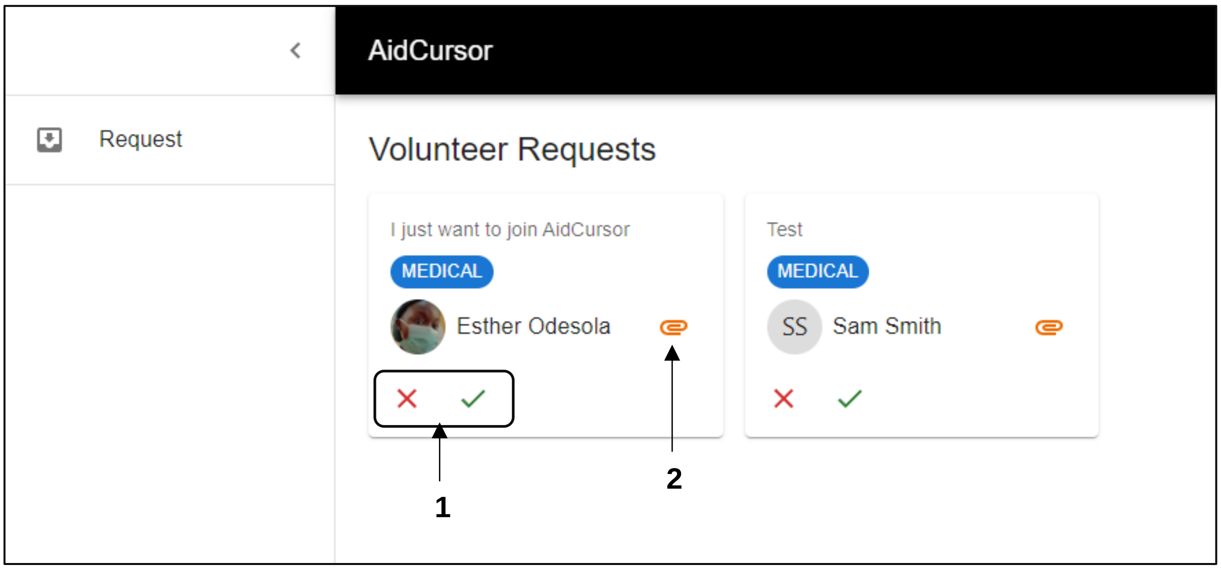


Figure 29 - Admin Panel Home Page

1. (Accept & Deny) Decision Buttons
2. View Attachment Button

The decision button at the bottom left section of the card enables the administrator to either approve or deny pending volunteering requests. After either of the buttons is clicked, the card then gets removed from the screen view. Meanwhile, the “View Attachment” button opens a new tab within the internet browser and displays the files that have been attached to the request form. Currently, it has been configured that the author’s account is the only one authorized to have access to the contents within the admin panel.

4 Evaluation

4.1 Heuristic Evaluation of User Interface

According to (Shneiderman et al., 2016), there are 8 standard principles used to examine the design of user interfaces, also known as “Shneiderman 8 Golden Rules”. These guidelines are aimed to highlight some of the essential elements that exhibit strong user usability including other aspects of a design that might be further enhanced.

4.1.1 Strive for consistency.

As seen previously in the implementation section, there is a clear and consistent theme that is being presented across all screens of the mobile app. There are 3 chosen colour themes which are “Black”, “Red”, and “White”. This theme can be seen throughout the entirety of the mobile app. The utilisation of the React framework for development made it simple to reuse both style and components (buttons, input fields, and menus) which as a result brought more consistency throughout the entire mobile application.

4.1.2 Seek universal usability.

This concept primarily focuses on ensuring that individuals of all backgrounds, age groups, and even professions would effortlessly grasp the user interface of the app that is being designed.

The positive feedback given by the app testers proved that the mobile app was simple to use and easy to navigate without much effort. The integration of an in-app instruction guide that demonstrates how to utilise the app's main features could potentially help speed up the user's comprehension of the mobile interface.

4.1.3 Offer informative feedback.

According to this principle, whenever an action or task is performed within an application there should be appropriate feedback confirming the user’s action. While using the mobile app, when an action is done or a button is pressed, there is always appropriate feedback that is presented. For instance, when the volunteering request form is submitted by a volunteer, there is an alert prompt that is displayed which confirms that the request has successfully been submitted as shown in Figure 30. Additionally, within the map navigation screen, there is a short message that appears at the bottom of the screen which notifies the volunteer that he/she has arrived at the scene of the emergency.

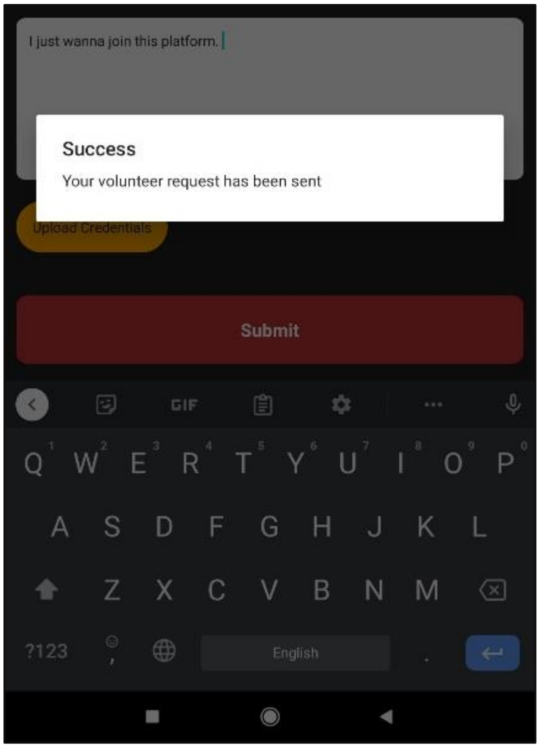


Figure 30 - Pop-up Alert

4.1.4 Design dialogues to yield closure.

This principle simply states that there should be confirmation feedback whenever a series of actions has been fully completed. Just as seen previously in Figure 30, after each field of the request form has been filled up and submitted, there is an alert that gets displayed confirming that the form has been completed, confirming the end of the process. Additionally, throughout the mobile app, whenever a user has reached the end of a process flow, there is always an alert which confirms its closure, which subsequently redirects the user back to the designated screen, completing the entire flow.

4.1.5 Prevent errors

This principle focuses on the idea that system error should be avoided by every means necessary. However, if something is wrong within the system there should be useful recovery steps given to the user to guide them in resolving the problem. One good example of this principle at work is within the authentication screens. Whenever an incorrect email or password is inputted when signing in, there is an alert popup that notifies the user that the credentials provided were invalid and must be retried.

4.1.6 Permit easy reversal of actions.

There are areas of the mobile application that allow the user to undo some of the actions done. For instance, within the map navigation screen whenever the user presses the back button, an alert is displayed to confirm that the user intends on exiting out of the map view. This logic in place prevents the user from accidentally exiting the navigation screen. However, the portion of the app which relates to form submissions is without the capability of undoing actions.

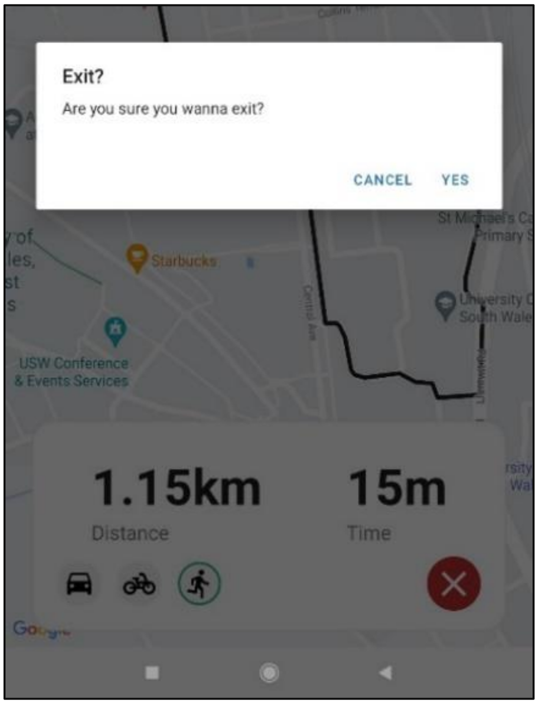


Figure 31 - Map exit Alert popup

4.1.7 Keep users in control.

It is essential that one should have a significant degree of control, particularly in terms of the overall mobile app's functionality. There are multiple areas within both the mobile app and admin panel that enables a user to control how the software should function. For instance, permission must be granted by a user before his/her current GPS location is sent to the backend server as shown in Figure 32 below. Other areas of the volunteering mobile app which allow people to control are their online/offline status and whether they would decide to respond to nearby help requests as previously seen within Section 3.3.1.2

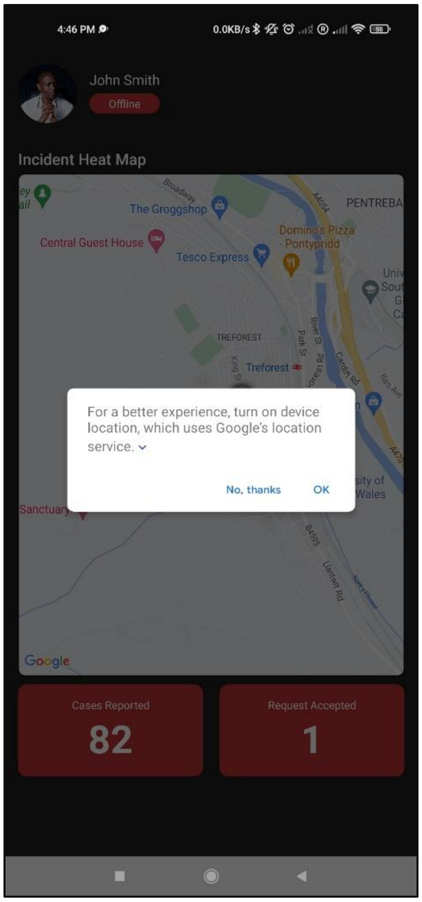


Figure 32 - Prompt to turn on GPS settings

4.1.8 Reduce short-term memory load.

This principle relies on the concept of limiting the amount of information being presented at a very given moment in order not to overwhelm the app user. Hence, every screen within a mobile app should be kept in a simple-to-understand manner and must only contain the relevant information for that given moment. As seen throughout the mobile interface, only the appropriate fields important to each screen are shown at any given time. The implementation of an in-app route navigation system makes it simple to toggle between displays while maintaining data synchronisation. This discouraged the need to place many fields on a single screen which might lead to the user being flooded with too much information at once. For instance, the profile screen seen in Section 3.3.1.4 is kept as a simple design with the account settings being on a separate screen in order not to overwhelm the end user and to improve the mobile app's ease of use.

4.2 User Evaluation

As stated in the previous chapter, an online survey was used as a medium to obtain feedback from the users evaluating the mobile application. A total of **eleven** respondents agreed to evaluate the volunteering mobile app that was developed. Once the mobile app had thoroughly been tested, the respondents were then prompted to fill up an online survey form. It took an average of about 10 minutes for each respondent to fully evaluate the mobile app. The section below is a full overview and summary of the responses that were received.

4.2.1 General Demographics

The section gives some additional information regarding the demographic of respondents that participated in the user evaluation. From the figure seen below, the males slightly outnumbered the females.

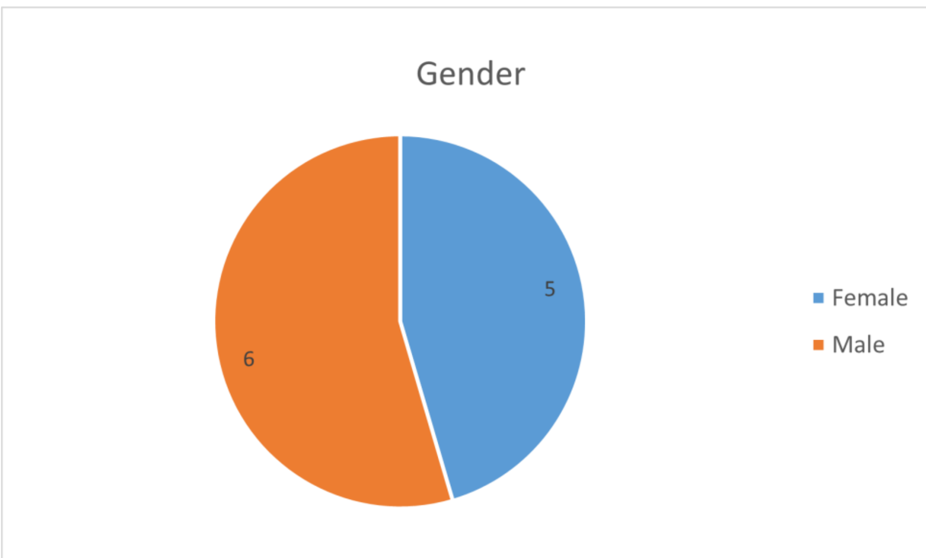


Figure 33 - Gender

In terms of the ethnic demographic, a significant percentage were “Black/African” at 72.73% followed by “Asian” and British with 18.18% and 9.09% respectively.

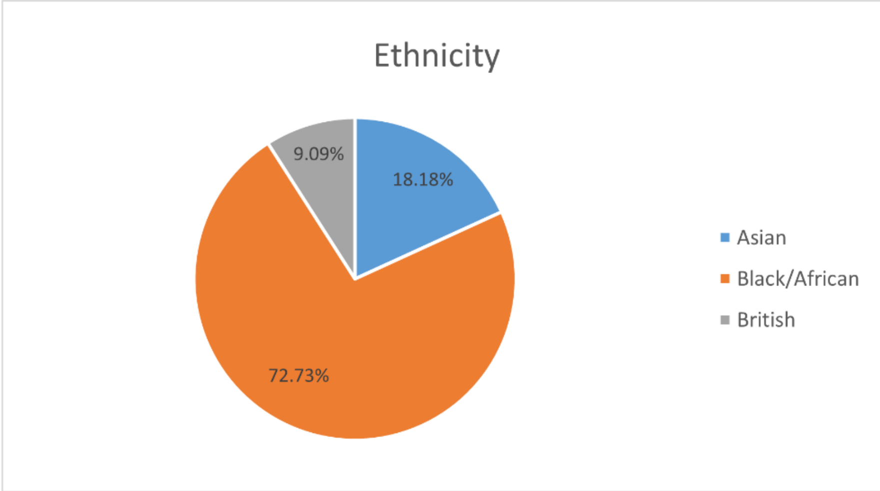


Figure 34 - Ethnicity

4.2.2 Navigation System

A question was asked about the navigation system that was implemented within the volunteer's mobile app. From the results shown below, it was seen that both 45.45% of the respondents responded with "very satisfied" and "somewhat satisfied" with the in-app navigation system. Meanwhile, the remaining 9.09% of the respondents were neither satisfied nor dissatisfied. Overall, none of the respondents had negative feedback and the responses received suggest that the navigation system was deemed useful in assisting the volunteer to locate the incident location.

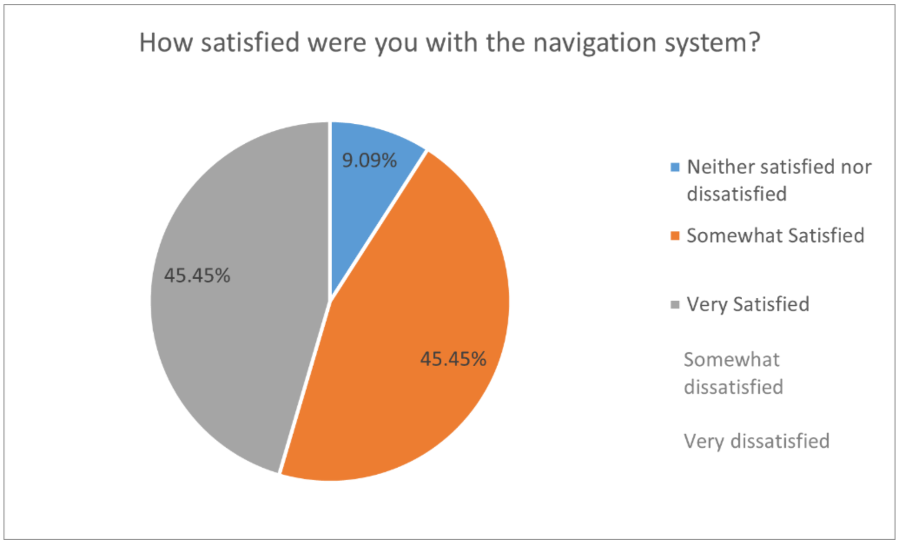


Figure 35 - Navigation System Satisfaction Summary

4.2.3 Notification System

Regarding the mobile app's notification system, 54.55% of the respondents responded as "Very Satisfied" and 36.36% chose "Somewhat Satisfied". Meanwhile, the remaining 9.09% had a neutral opinion on it. None of the respondents was dissatisfied with the app's notification system.

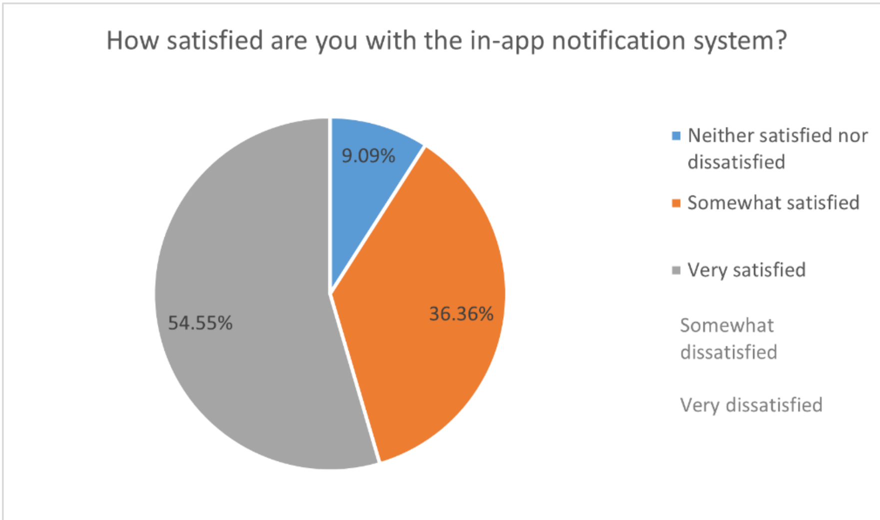


Figure 36 - Notification Satisfaction Summary

4.2.4 Ease of use

A question was asked to rate the ease of the mobile app. Based on the results seen, all the respondents had a positive response regarding its simplicity and ease of use. 45% of the respondents said that they had an excellent experience using the mobile app. Another 45% tagged it as "Very Good" and the remaining 9.09% rated it as "Good".

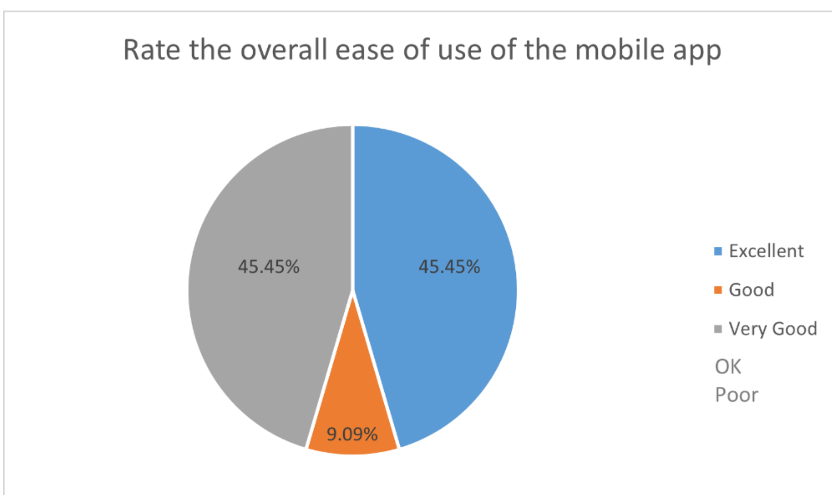


Figure 37 - Ease of Use Summary

4.2.5 Public use

A question was asked on whether they would ever consider using this mobile app if it were to be deployed to the public. Based on the response that was given, it was seen all the respondents said that they would consider using the app if it was publicly available. 81.82% of them rated the question with “Strongly Agreed” while the remaining 18.18% chose “Agree”. Overall, none of the respondents claimed that they would not use the app

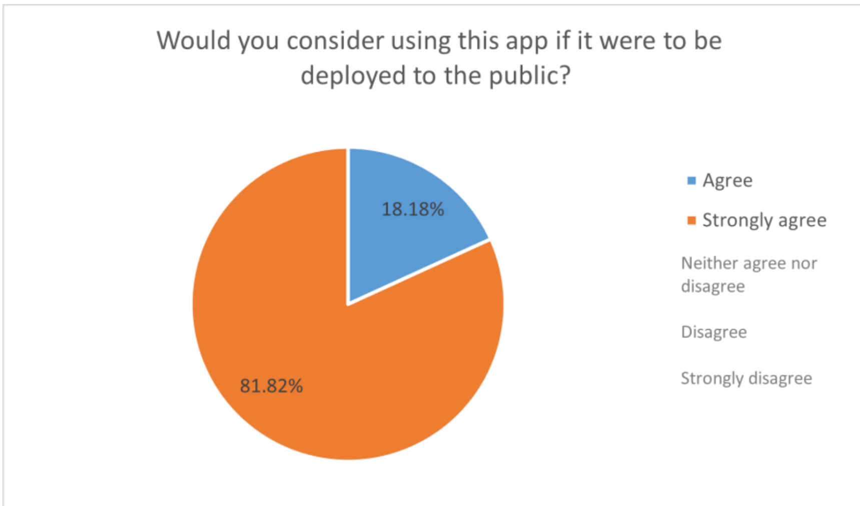


Figure 38 - If it were to be publicly available

4.2.6 Additional Feedback

There was an optional section within the survey form that allowed the respondents to give additional feedback regarding their overall experience while using the mobile app. The majority of the respondents provided positive appreciation concerning the performance of the mobile app’s features (GPS and notification). However, two issues were reported. One respondent reported that the volunteer's app did not operate correctly on their mobile device. The issue was a failure to receive relevant notifications on the volunteer app when a help request was sent. The other issue was the screen freeze which subsequently required a manual restart of the volunteer’s app. No obvious reasons could be identified for these technical failures.

As for future recommendations, one respondent suggested the use of a service rating system that would enable the user (help requester) to rate the service provided by the volunteer. There were other suggestions such as the incorporation of a *forget password flow* that is currently missing in the implementation and other minor fixes to help improve the overall user experience.

4.3 Final review of Research Objectives

4.3.1 Investigate and report on existing emergency volunteering platforms.

Within section 2.6 of Chapter 2 (Literature Review), four existing emergency volunteering platforms were examined and reported on. This includes GoodSAM, iHelp, Assist-Me, and A-SA SOS. Investigation of platforms showed some gaps in the range of emergency help requests it focuses on. Overall, it was seen that most of them are much focused on medical-related incidents, especially with OCHA or elderly cases.

4.3.2 Research the technologies to be utilised for development.

As seen in both Chapter 2 (Literature Review) and Chapter 3 (Design), the technologies that were chosen for development were thoroughly examined and documented to highlight some of their functionalities. It helped identify the best tools for the implementation phase of the research project. The technology stack that was chosen was ReactJS Framework (web and mobile development), Firebase (backend service), GPS (location-based system) and the Google Map Platform (navigation service).

4.3.3 Develop a mobile app that would allow trained volunteers to respond to surrounding emergencies or help requests using a real-time location-based system.

Once the right technologies had successfully been evaluated and chosen, what followed was the actual development and implementation of the volunteer mobile app as seen in Section 3.3.1. The volunteer app allowed people to register as volunteers and when approved within the system, it notified and enabled them to respond to nearby emergency requests. Along with the development of the volunteer app, 2 other software applications were also incorporated which were (1) the user (help requester) app which allowed a user to request help and (2) the admin panel which enabled the platform administrator to either approve or deny volunteer requests.

4.3.4 Design a real-time notification system that would notify a selected number of volunteers depending on a set of predetermined criteria.

As previously seen in Section 3.2.9, a notification system architecture was both designed and implemented to send out push notification messages. This system contained all the necessary operations needed to identify the best-suited volunteer based on key variables such as location, emergency type, and availability status. Once the push notification message had been received by the volunteer, he/she would be able to decide on whether to accept or deny the incoming help request.

4.3.5 Incorporate a navigation system that would provide the ideal route to the emergency location

The implementation of the navigation system was made quite straightforward using the Google Directions API which handled most of the necessary logic and route calculation considering different transportation modes. As a result, this made it simpler for the assigned volunteer to pinpoint the emergency incident location.

4.3.6 Perform a heuristic evaluation on the developed mobile application

As documented in Section 4.2 a heuristic evaluation was performed based on the 8 principles of User Interface Design according to (Shneiderman et al., 2016). Within this section, each of the principles was directly contrasted against the design of the mobile app that was developed. Overall, upon further evaluation, it was seen that most of the given rules were followed throughout the mobile app's interface.

4.3.7 Access the functionality of the implemented system using an online survey.

After the development phase was completed, the volunteering mobile app was then distributed to various respondents for both user testing and evaluation. Subsequently, once the user testing had been completed, an online survey was also sent to collect their feedback mainly regarding the app's overall usability or functionality. Overall, a sum of eleven respondents took part in the user evaluation.

4.3.8 Undertake a qualitative evaluation on the use of the application

The user feedback that was received from an online survey was subjected to a qualitative evaluation. Based on the responses that were received, it was seen that majority of the responses that were received had positive feedback regarding the functionality of the mobile app, especially with both the navigation and notification system. However, there was one participant that had difficulty using the mobile app, mentioning that the notification messages were at times not being received. It is unknown what caused this issue as it could be related to the model of the mobile phone used, and internet connectivity issues among other variable reasons. Overall, from the feedback received, all the respondents were likely to consider using the volunteering mobile app if it was publicly available.

5 Project Limitations & Issues

5.1 A limited number of respondents for user evaluation

Only a few respondents agreed to test and evaluate the features of the mobile app due to limited time constraints. It would have been beneficial to collect responses from a greater number of respondents. This subsequently would have added more weight and credibility to the conclusion (for example in terms of the diverse locations) that were drawn out from the feedback received.

5.2 Limited time available to further enhance the mobile application.

Given the short time frame available for development, some additional features such as a chatting system and speech-to-text conversion amongst other things could not be implemented into the mobile app. These features are addressed in greater detail within the future recommendations section (see Chapter 7).

5.3 There is no established mechanism for verifying volunteer credentials.

During the registration process as a volunteer, it is currently required that certificate(s) of training (for example First-aid training) or expertise history record is uploaded. However, there is no established way to verify the authenticity of the documents uploaded by each volunteer. Therefore, volunteers could upload their documents and there is no way of knowing if this is indeed legitimate or if it has been maliciously modified in one way or another. Hence, it is important and required to implement a validation system to ensure the credibility of every volunteer who registers within the platform.

5.4 Approval from regulatory bodies for public deployment.

According to (Public Health England, 2017), that are legal requirements must be satisfied before healthcare-based apps are to be utilised by the general public to a greater degree. In this case, there is a need to register the mobile app under the Care Quality Commission (CQC) since the app falls within its scope which focuses on the provision of health or social care under the Health and Social Care Act 2008 (Care Quality Commission, 2022). Additionally, it was also said that the regulatory organizations would need to assess certain areas of the software application such as its security, usability and accessibility, stability, and interoperability.

5.5 GDPR (General Data Protection Regulation) Compliance

GDPR compliance is very important for software applications that are primarily capable of collecting users' personal information (Castelluccia et al., 2017). The developed emergency volunteering mobile platform is heavily reliant on the use of GPS sensors for accurate location tracking. Hence, there could be privacy issues involved especially for the volunteers (while active) because their location must be actively monitored or captured for the system to be able to identify the nearest and the most ideal volunteer to respond to the emergency help requests. Additionally, with the user's personal information being stored on the cloud, there should be more thorough steps involved to safeguard their information.

6 Conclusion

This research project has successfully designed and developed an emergency volunteering system that enabled nearby registered volunteers within the area to assist with reported emergency incidents. As stated in Chapter 1, two main research questions were identified. The first on the software requirements needed for the development of the prototype and the second on the evaluation methods that would be employed to evaluate the developed system. It can be concluded that both these questions have been addressed as follows.

As part of the technical requirement, two mobile applications and a web-based interface had to be implemented. Part 1 was the mobile app for volunteers that was both designed and developed to allow them to assist with nearby emergency help requests. Part 2 was the mobile app for public users that allowed a person to send help requests when in need of assistance. Finally, the web interface was developed to function as an admin panel mainly aimed at verifying the registration request of each of the volunteers who signed up using the volunteer app.

To address the evaluation phase, two types of evaluations were conducted. The first phase of the evaluation that was conducted was a heuristic evaluation of the mobile app's user interface which was based on Shneiderman's 8 principles of User Design (Shneiderman et al., 2016). The second was a user evaluation in which a total of eleven respondents agreed to both evaluate the mobile app and give their initial feedback through an online survey. Based on the overall results that were gathered, it was seen that the app's functionality was well received by the respondents and the feedback was positively received to further improve the mobile application.

Overall, this project has been able to prove that an emergency volunteering mobile platform could be developed within a short time frame using currently available technologies.

7 Future Recommendations

To address the limitations of the project, the following future recommendations address the areas of the research project that could have been further improved upon.

7.1 Volunteer matchmaking system improvements

Further improvements could be made regarding the process by which volunteers are identified based on a given emergency request. An improvement to the volunteer matchmaking system would be to implement a matching technique such as Cosine Similarity, Pearson Correlation, or the Mean Squared Differences (MSD) (Agarwal and Chauhan, 2017). Alternatively, an Artificial Intelligence (AI) based recommendation model to predict and match the ideal volunteer based on their skill and/or experience level can be implemented.

7.2 Implementation within a real-life scenario.

It is within the future recommendation to have real certified volunteers utilize the mobile app in a real-life environment. Additionally, it is also recommended for the mobile app be evaluated and assessed by individuals working within emergency response organizations. Additionally, to further enhance the efficiency of the system, it is important to evaluate and receive feedback from volunteers using the mobile application within an actual emergency scenario.

7.3 In-built chatting system

The integration of a chatting system within the mobile app would is also a recommended addition to the platform. This will prove helpful in situations when the volunteer will be able to actively communicate with the user (help requester) in real-time while still on the journey towards the emergency location. This will also potentially help the volunteer to efficiently assist the user in a timely manner. Additionally, there could be a chat system implemented for the volunteers participating in a given emergency to improve their overall task coordination.

7.4 Integrate Speech-to-Text Conversion when generating a help request.

A speech-to-text system that transcribes the audio messages to a text format will allow a user (help requester) to send voice messages instead of typing out the description of the emergency incident. This is particularly useful when the help requester is limited by time or by other means that prevent them from typing text messages to request help. This will potentially improve the overall user experience when requesting help during an emergency. Google’s Speech-To-Text API service is widely accessible and is popular in mobile apps that have text messaging systems (Google, 2022d).

7.5 Accurate indoor tracking system

Currently, the implemented system lacks accurate indoor location tracking. An indoor tracking system is most useful within indoor environments that have multiple storey levels. At hand, the volunteer cannot reliably identify the floor within a multi-story structure from where the help request originated. Some research exists that evidence a potential solution to integrate the use of other location-based services such as Wi-Fi, Bluetooth, and Cellular systems which have been known to be more inclined toward indoor navigation which was previously discussed in the literature review.

8 References

Abu-Elkheir, M., Hassanein, H.S. and Oteafy, S.M.A. (2016) ‘Enhancing Emergency Response Systems through Leveraging Crowdsensing and Heterogeneous Data Analytics’., pp. 188–193.

Abulude, F.O., Akinnusotu, A. and Adeyemi, A. (2015) ‘Global Positioning System and it’s wide applications’. *Continental Journal of Information Technology* , pp. 22–32.

Agarwal, A. and Chauhan, M. (2017) ‘Similarity Measures used in Recommender Systems: A Study’. *International Journal of Engineering Technology Science and Research* 4(6), pp. 619–626.

Aloudat, A. and Michael, K. (2011) ‘The application of location based services in national emergency warning systems: SMS, cell broadcast services and beyond’. Available at: https://ro.uow.edu.au/infopapers/2031.

Altexsoft (2020) ‘ReactJS and React Native: Overview Pros and Cons’. *The Good and the Bad of ReactJS and React Native* 7 July. Available at: https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-reactjs-and-react-native/ (Accessed: 1 July 2022).

Athey, S. and Stern, S. (2000) ‘The Impact of Information technology Emergency Health Care Outcomes’. doi: 10.3386/w7887.

Azdy, R. and Darnis, F. (2019) ‘Use of Haversine Formula in Finding Distance Between Temporary Shelter and Waste End Processing Sites’. in Journal of Physics Conference Series.

Belardo, S., Karwan, K.R. and Wallace, W.A. (1984) ‘Managing the Response to Disasters Using Microcomputers’. *INFORMS* 14, pp. 29–39.

Care Quality Commission (2022) *The scope of registration*., p. 69. Available at: https://www.cqc.org.uk/sites/default/files/2022-05/20220517%20Scope%20of%20Registration%20Guidance%20May%202022.pdf.

Castelluccia, C., Guerses, S.F., Hansen, M.P., Hoepman, J.-H., Hoboken, J.V., and Bárbara Vieira (2017) ‘Privacy and data protection in mobile applications. A study on the app development ecosystem and the technical implementation of GDPR’. Available at: https://pure.uva.nl/ws/files/42887337/22302384.pdf.

Cybersecurity and Infrastructure Security Agency (2019) *Emergency Services Sector Landscape*. Available at: https://www.cisa.gov/sites/default/files/publications/Emergency%20Services%20Sector%20Landscape.pdf.

Dostál, J. and Balarinová, L. (2013) ‘Volunteers in emergency management: An investment in the future?’., pp. 381–391.

El-Rabbany, A. (2002) *Introduction to GPS The Global Positioning System*. Artech House.

Expo (2022) *Sending Notifications with Expo’s Push API*. Available at: https://docs.expo.dev/push-notifications/sending-notifications/ (Accessed: 16 July 2021).

References

Fan, Y., Dooley, M.F. and Stading, G.L. (2013) ‘Influencing factors for effective emergency response: Empirical evidence from tri-lakes monument fire protection District’. Baltimore

Federal Communication Commission (2017) *Inquiry Concerning 911 Access, Routing, and Location in Enterprise Communications Systems*.

Filjar, R., Vidovic, K., Rimac, M. and Britvic, P. (2011) ‘ECALL: AUTOMATIC NOTIFICATION OF A ROAD TRAFFIC ACCIDENT’. Croatia

Firebase (2022a) *Firebase Documentation*. Available at: https://firebase.google.com/docs.

Firebase (2022b) *Geo queries*. Available at: https://firebase.google.com/docs/firestore/solutions/geoqueries (Accessed: 7 July 2022).

GoodSAM (2022) ‘NHS Volunteer Responders’. *NHS Volunteer Responders* . Available at: https://www.goodsamapp.org/NHS (Accessed: 1 July 2022).

Google (2022a) ‘A look back at 15 years of mapping the world’. Available at:] https://blog.google/products/maps/look-back-15-years-mapping-world/ (Accessed: 1 July 2022).

Google (2022b) *Directions API*. Available at: https://developers.google.com/maps/documentation/directions (Accessed: 17 July 2022).

Google (2022c) *Distance Matrix API overview*. Available at: https://developers.google.com/maps/documentation/distance-matrix/overview (Accessed: 25 July 2022).

Google (2022d) *Speech-to-Text: Automatic Speech Recognition*. Available at: https://cloud.google.com/speech-to-text (Accessed: 18 August 2022).

Gustafsson, F. and Gunnarsson, F. (2005) ‘Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements’. *IEEE Signal Processing Magazine* 22(4), pp. 41–53.

Hays, S. and Harle, R. (2009) ‘Bluetooth Tracking without Discoverability’. in *Location and Context Awareness*., pp. 120–137.

Hellebrandt, M. and Mathar, R. (1999) ‘Location Tracking of Mobiles in Cellular Radio Networks’. *IEEE Transactions on Vehicular Technology* 48(5), pp. 1558–1562.

IBM (2021) *Geohashes*. Available at: https://www.ibm.com/docs/en/streams/4.3.0?topic=334-geohashes (Accessed: 5 July 2022).

iHelp (2022) *iHelp*. Available at: https://www.ihelp-world.com/ (Accessed: 17 August 2022).

Intawong, K., Boonchieng, W., Lerttrakarnnon, P., Boonchieng, E. and Puritat, K. (2021) ‘A-SA SOS: A Mobile- and IoT-based Pre-hospital Emergency Service for the Elderly and Village Health Volunteers’.

Jekabsons, G., Kairish, V. and Zuravlyov, V. (2011) ‘An Analysis of Wi-Fi Based Indoor Positioning Accuracy’. *Scientific Journal of Riga Technical University Computer Sciences*

References

Jennex, M.E. (2007) ‘Modeling Emergency Response Systems’.

Khalemsky, M. and Schwart, D.G. (2017) ‘Emergency Response Community Effectiveness: A simulation modeler for comparing Emergency Medical Services with smartphone-based Samaritan response’. Available at: https://reader.elsevier.com/reader/sd/pii/S0167923617301252?token=28C1A68C5FE79EFA0A2B628975A740BCDB22C51191944A03D24982AA92678E67F6F92E7A9D5F55B8C02CE1808F2A92F5&originRegion=eu-west-1&originCreation=20220701085545.

Khalemsky, M., Schwartz, D.G., Raphael Herbst and Jaffe, E. (2020) ‘Motivation of emergency medical services volunteers: a study of organized Good Samaritans’. Available at: https://ijhpr.biomedcentral.com/track/pdf/10.1186/s13584-020-00370-9.pdf.

Khawas, C. and Shah, P. (2018) ‘Application of Firebase in Android App Development-A Study’. *International Journal of Computer Applications* , pp. 49–53.

Kumar, S., Tiwari, P. and Zymbler, M. (2019) ‘Internet of Things is a revolutionary approach for future technology enhancement: a review’. *Journal of Big Data*

Larson, R.C., Metzger, M.D. and Cahn, M.F. (2006) ‘Responding to Emergencies: Lessons Learned and the Need for Analysis’. *Interfaces* , pp. 486–501. doi: 10.1287/inte.1060.0250.

Mallat, H.K. and Abdulrazak, B. (2015) ‘Assist-Me - A Volunteer Mobile Emergency System to Assist Elderly People’. *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing* , pp. 2373–2380.

Matinrad, N. and Reuter-Oppermann, M. (2021) ‘A review on initiatives for the management of daily medical emergencies prior to the arrival of emergency medical services’. *Central European Journal of Operations Research* . doi: 10.1007/s10100-021-00769-y.

Mladjan, D. and Cvetković, V.M. (2013) ‘Classification of emergency situations - Klasifikacija vanrednih situacija’., pp. 275–291.

Movable Type Scripts (2022) *Geohashes*. Available at: https://www.movable-type.co.uk/scripts/geohash.html (Accessed: 20 July 2022).

New Hampshire Bureau of Emergency Communications (2012) ‘History of 9-1-1 in the US’. *Information bulletin* 16 February. Available at: https://www.nh.gov/safety/divisions/emergservices/nh911/pubinfo/documents/history.pdf.

Nguyen, T.H. and Szymanski, B. (2012) ‘Using Location-Based Social Networks to Validate Human Mobility and Relationships Models’., pp. 1215–1221.

Ogorevc, A. and Lončarevič, B. (2014) ‘iHELP emergency care network’., pp. 252–255.

Okesola, O.J., Adebiyi, A.A., Owoade, A.A., Adeaga, O., Adeyemi, O. and Odun-Ayo, I. (2020) ‘Software Requirement in Iterative SDLC Model’. in Advances in Intelligent Systems and Computing. Springer, Cham, pp. 36–34. Available at: https://doi.org/10.1007/978-3-030-51965-0_2.

References

Oorni, R. and Goulart, A. (2017) ‘In-Vehicle Emergency Call Services: eCall and Beyond Risto Öörni and Ana Goulart’. *IEEE Communications Magazine* . Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7823355.

Oracle (2022) *What is IoT?* Available at: https://www.oracle.com/uk/internet-of-things/what-is-iot/ (Accessed: 1 July 2022).

Pardess, E. (2008) ‘Training and Mobilizing Volunteers for Emergency Response and Long-Term Support’.

Paylor, J. (2011) ‘Volunteering and health: evidence of impact and implications for policy and practice’.

Pilemalm, S. (2020) ‘Volunteer Co-production in Emergency Management in Excluded Areas: Using Civil Citizens and Semi-professionals as First Responders’. *eJournal of eDemocracy and Open Government (JeDEM)*

Public Health England (2017) *Criteria for health app assessment*. Available at: https://www.gov.uk/government/publications/health-app-assessment-criteria/criteria-for-health-app-assessment (Accessed: 17 August 2022).

Rawal, D. (2017) ‘Traditional Infrastructure Vs Firebase Infrastructure’. *IJSRD - International Journal for Scientific Research & Development* 5(4)

React (2022) *ReactJS*. Available at: https://reactjs.org/ (Accessed: 1 July 2022).

Sethi, P. and Sarangi, S.R. (2017) ‘Internet of Things: Architectures, Protocols, and Applications’. *Hindawi Journal of Electrical and Computer Engineering* . Available at: https://downloads.hindawi.com/journals/jece/2017/9324035.pdf.

Shneiderman, B., Jacobs, S.M., Cohen, M.S., Diakopoulos, N., Elmqvist, N. and Plaisant, C. (2016) ‘Designing the User Interface: Strategies for Effective Human-Computer Interaction’. *Pearson* . Available at: https://www.cs.umd.edu/users/ben/goldenrules.html (Accessed: 19 July 2022).

Smith, C.M., Lall, R., Fothergill, R.T., Spaight, R. and Perkins, G.D. (2022) ‘The effect of the GoodSAM volunteer firstresponder app on survival to hospital discharge following out-of-hospital cardiac arrest’. *European Heart Journal: Acute Cardiovascular Care* , pp. 20–31.

Stroop, R., Kerner, T., Strickmann, B. and Hensel, M. (2020) ‘Mobile phone-based alerting of CPR-trained volunteers simultaneously with the ambulance can reduce the resuscitation-free interval and improve outcome after out-of-hospital cardiac arrest: A German, population-based cohort study’. *Resuscitation*

Stukas, A., Snyder, M. and Clary, E.G. (2014) ‘Volunteerism and community involvement: Antecedents, experiences, and consequences for the person and the situation.’ in *The Oxford handbook of prosocial behavior*. Oxford University Press

Trogh, J., Plets, D., Surewaard, E., Spiessens, M., Versichele, M., Martens, L. and Joseph, W. (2019) ‘Outdoor location tracking of mobile devices in cellular networks’. Available at: https://doi.org/10.1186/s13638-019-1459-4.

References

UK Cabinet (2011) ‘The Role of the Voluntary Sector) of Emergency Preparedness’. in *Revision to Emergency Preparedness*. Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/61037/Chapter-14-role-of-voluntary-sector-amends-10112011.pdf.

Yang, L., Prasanna, R. and King, M. (2009) ‘On-Site Information Systems Design for Emergency First Responders’. *Journal of Information Technology Theory and Application*

9 Appendices

9.1 Source Codes

The full source code is located within this GitHub repository account https://github.com/Olatokumbo. However, it is currently set to private; access will be granted upon request.

9.1.1 Notification.ts

```
import type { ExpoPushMessage } from "expo-server-sdk";
import { Expo } from "expo-server-sdk";

const expo = new Expo({ accessToken: process.env.EXPO_ACCESS_TOKEN });

type SendPushNotificationProps = {
  pushToken: string;
  message: string;
  data?: object;
};

export const sendPushNotification = async ({
  pushToken,
  message,
  data,
}: SendPushNotificationProps): Promise<void> => {
  const messages: ExpoPushMessage[] = [];

  if (!Expo.isExpoPushToken(pushToken)) {
    console.error(`Push token ${pushToken} is not a valid Expo push token`);
    return;
  }

  messages.push({
    to: pushToken,
    sound: undefined,
```

```
    title: "⚠Someone nearby needs your help ⚠",
    priority: "high",
    body: message,
    data,
  });

  try {
    await expo.sendPushNotificationsAsync(messages);
  } catch (error) {
    console.error(error);
  }
};

export default sendPushNotification;
```

9.1.2 usePushNotification.tsx

```
import useAsync from "react-use/lib/useAsync";
import { useRef, useState } from "react";
import {
  addNotificationReceivedListener,
  addNotificationResponseReceivedListener,
  AndroidImportance,
  removeNotificationSubscription,
  setNotificationChannelAsync,
  setNotificationHandler,
  useLastNotificationResponse,
} from "expo-notifications";
import type { Subscription } from "expo-modules-core";
import type { Notification, NotificationResponse } from "expo-notifications";
import { navigationRef } from "../navigation/testRoot";

setNotificationHandler({
  // eslint-disable-next-line @typescript-eslint/require-await
```

```
  handleNotification: async () => ({
    shouldShowAlert: true,
    shouldPlaySound: true,
    shouldSetBadge: false,
  }),
});

const usePushNotifications = (
  onTapNotification?: (response: NotificationResponse) => void
): {
  notification: Notification | null;
} => {
  const [notification, setNotification] = useState<Notification | null>(null);
  const notificationListener = useRef<Subscription>();
  const responseListener = useRef<Subscription>();

  const lastNotificationResponse = useLastNotificationResponse();

  useAsync(async () => {
    if (lastNotificationResponse) {

      //get the route
      const incidentId = (
        lastNotificationResponse.notification.request.content.data.incidentId
      );
      navigationRef.current?.navigate("RequestChoice", { incidentId });
    }
  }, [lastNotificationResponse]);

  useAsync(async () => {
    notificationListener.current =
      addNotificationReceivedListener(setNotification);

    responseListener.current = addNotificationResponseReceivedListener(
```

```
      (response) => onTapNotification?.(response)
      // () =>  Linking.openURL("exp://home")
    );

    await setNotificationChannelAsync("default", {
      name: "default",
      importance: AndroidImportance.MAX,
      vibrationPattern: [0, 250, 250, 250],
      lightColor: "#FF231F7C",
      enableVibrate: true,
      sound: "default",
    });

    return () => {
      if (notificationListener.current) {
        removeNotificationSubscription(notificationListener.current);
      }
      if (responseListener.current) {
        removeNotificationSubscription(responseListener.current);
      }
    };
  });

  return { notification };
};

export default usePushNotifications;
```

9.1.3 Incident.ts

```
import * as functions from "firebase-functions";
import { distanceBetweenCoords } from "../helper/distance";
import { sendPushNotification } from "../helper/notification";
import { queryVolunteerHashes } from "./volunteer";
export const radiusInM = 2 * 1000; // 2 KM Radius;
```

Appendices

```
interface ApiError {
  code: number;
  error: string;
}

function isApiError(x: any): x is ApiError {
  return typeof x.code === "number";
}
export const incidentCreated = functions.firestore
  .document("incidents/{incidentId}")
  .onCreate(async (snap, context) => {
    try {
      const matchingDocs: any[] = [];
      const incident = snap.data();
      const incidentId = context.params.incidentId;
      const { latitude, longitude } = incident.location;
      // PROCEDURE: Search volunteers collection and get list of pushToken

      let docs: any[] = await queryVolunteerHashes(
        latitude,
        longitude,
        incident.category
      );

       //sort volunteers by those who have accepted the most number incidents
      docs.sort((a, b)=> b.totalIncidents - a.totalIncidents)

     functions.logger.info("DOCS>>", docs);
      const pushTokens: any[] = await Promise.all(
        docs.map(async (doc) => {
          const distance = await distanceBetweenCoords(
            [latitude, longitude],
            [doc.location.latitude, doc.location.longitude]
          );
```

```
        functions.logger.info("DISTANCE (M)", distance);
        functions.logger.info("RADIUS (M)", radiusInM);
        //verify distance and get at most 3 volunteers
        if (distance <= radiusInM && matchingDocs.length < 3) {
          functions.logger.info("MATCH", doc);
          return doc.pushToken as string;
        }
      })
    );

    functions.logger.info("TOKENS", pushTokens);

    const promises: Promise<void>[] = [];

    if (pushTokens.length > 0) {
      pushTokens.forEach((pushToken) => {
        promises.push(
          sendPushNotification({
            pushToken,
            message: `🛑 ${incident.message}`,
            data: {
              incidentId,
            },
          })
        );
      });
    }

    return Promise.all(promises).then(() => {
      functions.logger.info("NOTIFICATIONS sent!!!!");
    });
  } catch (error) {
    if (isApiError(error)) {
```

```
        console.log(error);
      }
    }
  });
```

9.1.4 Distance.ts

```
import { Client, TravelMode } from "@googlemaps/google-maps-services-js";
import { GOOGLE_DIRECTIONS_API_KEY } from "./config";

export const distanceBetweenCoords = async (
  [lat1, lng1]: number[],
  [lat2, lng2]: number[]
) => {
  const client = new Client({});

  const response = await client.distancematrix({
    params: {
      origins: [{ lat: lat1, lng: lng1 }],
      destinations: [{ lat: lat2, lng: lng2 }],
      key: GOOGLE_DIRECTIONS_API_KEY,
      mode: TravelMode.bicycling,
    },
  });

  return response.data.rows[0].elements[0].distance.value;
};
```

9.1.5 Volunteer.ts

```
import * as admin from "firebase-admin";
import { geohashQueryBounds } from "geofire-common";
import { radiusInM } from "./incident";

export const queryVolunteerHashes = (
  c1: number,
  c2: number,
```

```
  category: string
) => {
  const bounds = geohashQueryBounds([c1, c2], radiusInM);
  const promises = [];
  for (const b of bounds) {
    const q = admin
      .firestore()
      .collection("volunteers")
      .where("active", "==", true) // Must be active
      .where("category", "==", category) // Must be under the specified the incident category
      .orderBy("geoHash")
      .startAt(b[0])
      .endAt(b[1]);

    promises.push(q.get());
  }

  // Collect all the query results together into a single list
  return Promise.all(promises)
    .then((snapshots) => {
      const docs: any[] = [];

      for (const snap of snapshots) {
        for (const doc of snap.docs) {
          docs.push(doc.data());
        }
      }
      return docs;
    })
};
```

9.1.6 Map.tsx

```
import { useNavigation } from "@react-navigation/native";
```

Appendices

```
import {
  NativeStackNavigationProp,
  NativeStackScreenProps,
} from "@react-navigation/native-stack";
import { Dimensions, StatusBar, StyleSheet } from "react-native";
import React, { useEffect, useRef, useState } from "react";
import { View, Text, Dimensions, BackHandler } from "react-native";
import MapView, { Marker } from "react-native-maps";
import useLocation from "../../hooks/useLocation";
import Ionicons from "@expo/vector-icons/Ionicons";
import { StackParams } from "../../navigation/root";
import { styles } from "./styles";
import MapViewDirections from "react-native-maps-directions";
import { Colors } from "../../constants/global";
import { GOOGLE_API_KEY } from "@env";
import { DirectionCard } from "../../components/DirectionCard";
import { travelMode } from "../../enum/travelMode.enum";

type MapScreenProps = NativeStackScreenProps<StackParams, "Map">;

const Map: React.FC<MapScreenProps> = ({ route, navigation }) => {
  const { latitude, longitude } = route.params;
  const { height, width } = Dimensions.get("window");
  const [distance, setDistance] = useState<number | null>(0);
  const [time, setTime] = useState<number | null>(0);
  const [mode, setMode] = useState<travelMode>(travelMode.DRIVING);
  const [arrived, setArrived] = useState<boolean>(false);
  const { location } = useLocation(true);
  const [loading, setLoading] = useState(true);
  const LATITUDE_DELTA = 0.28;
  const LONGITUDE_DELTA = LATITUDE_DELTA * (width / height);
  let mapView: MapView | null;
  useEffect(() => {
    setTimeout(() => {
```

```
      setLoading(false);
    }, 3000);

    const backHandler = BackHandler.addEventListener(
      "hardwareBackPress",
      () => true
    );
    return () => backHandler.remove();
  }, []);

  return (
    <View style={styles.container}>
      {location && (
        <>
          <MapView
            ref={(c) => (mapView = c)}
            mapType="standard"
            showsUserLocation
            showsMyLocationButton
            // customMapStyle={mapStyle}
            initialRegion={{
              latitude: location.latitude,
              longitude: location.longitude,
              latitudeDelta: LATITUDE_DELTA,
              longitudeDelta: LONGITUDE_DELTA,
            }}
            style={styles.map}
          >
            <MapViewDirections
              origin={{
                latitude: location.latitude,
                longitude: location.longitude,
              }}
              destination={{ latitude, longitude }}
```

Appendices

```
        apikey="insert your API Key here”
        strokeWidth={4}
        strokeColor={Colors.primary}
        // optimizeWaypoints={true}
        onStart={(params) => {}}
        onReady={(result) => {
          // console.log(result);
          setDistance(result.distance?.toFixed(2) ?? 0);
          setTime(result.duration?.toFixed(0) ?? 0);

          if (distance! <= 0.01 && !loading) {
            setArrived(true);
          } else {
            setArrived(false);
          }
          if (result.coordinates.length > 0) {
            mapView?.fitToCoordinates(result.coordinates, {
              edgePadding: {
                right: width / 20,
                bottom: height / 20,
                left: width / 20,
                top: height / 20,
              },
            });
          }
        }}
        mode={mode}
      />

      <Marker
        coordinate={{
          latitude,
          longitude,
        }}
```

```
          >
            <Ionicons name="location" size={40} color={Colors.secondary} />
          </Marker>
        </MapView>
        <DirectionCard
          navigation={navigation}
          time={time}
          distance={distance}
          mode={mode}
          arrived={arrived}
          setMode={setMode}
        />
      </>
    )}
  </View>
 );
};

export default Map;



const styles = StyleSheet.create({
 container: {
  flex: 1,
  backgroundColor: "#fff",
  alignItems: "center",
  justifyContent: "center",
  marginTop: StatusBar.currentHeight,
 },
 map: {
  width: Dimensions.get("window").width,
  height: Dimensions.get("window").height,
 },
});
```

9.1.7 useLocation.tsx

```
import React, { useState, useEffect } from "react";
import * as Location from "expo-location";

const useLocation = (active: boolean) => {
  const [location, setLocation] =
    useState<null | Location.LocationObjectCoords>(null);
  const [errorMsg, setErrorMsg] = useState<string | null>(null);

  useEffect(() => {
    getData();
    const interval = setInterval(() => {
      // active && getData(); //TODO: Fix this
      getData();
    }, 5000);
    return () => clearInterval(interval);
  }, []);

  const getData = async () => {
    try {
      let { status } = await Location.requestForegroundPermissionsAsync();
      if (status !== "granted") {
        setErrorMsg("Permission to access location was denied");
        return;
      }

      let location = await Location.getCurrentPositionAsync({});
      setLocation(location.coords);
    } catch (error) {
      alert(error);
    }
  };
  return {
    location,
```

```
    errorMsg,
  };
};

export default useLocation;
```

9.1.8 Roles.ts

```
import * as functions from "firebase-functions";
import * as admin from "firebase-admin";
enum Role {
  ADMIN = "ADMIN",
  VOLUNTEER = "VOLUNTEER",
  PATIENT = "PATIENT",
}

export const volunteerCreated = functions.firestore
  .document("volunteers/{volunteerId}")
  .onCreate(async (_snapshot, context) => {
    const volunteerId = context.params.volunteerId;
    return admin
      .auth()
      .setCustomUserClaims(volunteerId, { role: Role.VOLUNTEER });
  });

export const patientCreated = functions.firestore
  .document("patients/{patientId}")
  .onCreate(async (_snapshot, context) => {
    const patientId = context.params.patientId;
    return admin.auth().setCustomUserClaims(patientId, { role: Role.PATIENT });
  });

export const adminCreated = functions.firestore
  .document("admins/{adminId}")
```

```
.onCreate(async (_snapshot, context) => {
  const adminId = context.params.adminId;
  return admin.auth().setCustomUserClaims(adminId, { role: Role.ADMIN });
});
```

9.2 Online Survey Format

The format of the online survey that was used to evaluate the mobile application is as shown below.

1. Gender
    - Male
    - Female
    - Prefer not to say

2. Ethnicity
    - British
    - Asian
    - Caucasian
    - Hispanic
    - Black/African
    - Mixed Race
    - Others

3. Rate the overall ease of use of the mobile app.
    - Excellent
    - Very Good
    - Good
    - OK
    - Poor

4. How satisfied were you with the navigation system?
    - Very Satisfied
    - Somewhat Satisfied
    - Neither satisfied nor dissatisfied
    - Somewhat dissatisfied
    - Very dissatisfied

5. What aspect of the mobile app's design flow could be further improved?

6. How satisfied were you with the in-app notification system?
   - Very satisfied
   - Somewhat satisfied
   - Neither satisfied nor dissatisfied
   - Somewhat dissatisfied
   - Very dissatisfied

7. Would you consider using this app if it were to be deployed to the public?
   - Strongly agree
   - Agree
   - Neither agree nor disagree
   - Disagree
   - Strongly disagree

8. Any Additional Feedback.

9.3 Mobile UI Designs

The mobile mock-up designs can be viewed through this URL link:
https://www.figma.com/file/qa4CPWannSrfQmTNFIDjx8/AidCursor?node-id=0%3A1

9.4 Meeting Log

During the research project, there were regular weekly meetings conducted with Dr Shiny Verghese (the primary supervisor). Below is the summary of the list of meetings that took place.

Appendices

Date	Time	Agenda	Participants
June 9, 2022	12:00 pm – 12:30 pm	◦ Introduction ◦ Chapter 1 overview	David Odesola Dr. Shiny Verghese
June 15, 2022	12:00 pm – 12:30 pm	◦ App Notification Overview and Procedure ◦ Discussions on the newly added screens (Help Request Screen & Volunteer Form Screen) ◦ Discussion on Volunteering Requirements ◦ Overview of Relevant Technologies (Location Based Services, Evolution of Mobile Phone)	David Odesola Dr. Shiny Verghese
June 21, 2022	12:00 pm – 12:30 pm	◦ Web Admin Panel Overview ◦ New Emergency Request Screen Discussion ◦ Updates to the Literature Review	David Odesola Dr. Shiny Verghese
June 29, 2022	12:00 pm – 12:30 pm	◦ Development of Patient App ◦ Start of Chapter 3 (Selection of Technology/Diagrams) ◦ Correction most of the App Flow	David Odesola Dr. Shiny Verghese
July 6, 2022	2:00 pm – 2:30 pm	◦ Discussion on the development of Volunteer matchmaking system ◦ Adding Profile info to Help Request (Feature) ◦ Active Volunteer's location gets updates in db every 1 minute (Feature) ◦ Incident Cards (Feature) ◦ Add Incident Heatmap	David Odesola Dr. Shiny Verghese Simon Payne

		◦ Initial construction of Questionnaire ◦ Suggestion for better Referencing using the “Zotero”	
July 15, 2022	12:00 pm – 12:30 pm	◦ UI Overview Description ◦ Refining the Questionnaire Survey to include demographics ◦ Finishing up the mobile APP	David Odesola Dr. Shiny Verghese
July 20, 2022	12:00 pm – 12:30 pm	◦ Paper Restructuring ◦ Finish up UI Overview ◦ Created Questionnaire survey on MS Forms ◦ Started Heuristic Evaluation of User Interface ◦ Started the evaluation phase	David Odesola Dr. Shiny Verghese
July 25, 2022	5:00 pm – 5:15 pm	◦ Discussion on how the volunteer matchmaking system could be further improved. ◦ Decided on the integration of the Google Directions Matrix API	David Odesola Dr. Janusz Kulon
July 27, 2022	12:00 pm – 12:30 pm	◦ Finishing up the Heuristic Evaluation ◦ Add some complexity to the “Volunteer Match-Making” System ◦ Requested Changes to Paper	David Odesola Dr. Shiny Verghese
August 3, 2022	12:00 pm – 12:30 pm	◦ Requested Corrections and Writing improvements ◦ Evaluation of Research Objectives	David Odesola Dr. Shiny Verghese

Appendices

August 10, 2022	12:00 pm – 12:30 pm	◦ Discussion of User Evaluation ◦ Further improvements ideas to the paper	David Odesola Dr. Shiny Verghese
August 17, 2022	12:00 pm – 12:30 pm	◦ Review of the previous week's work. ◦ Reconstruction of the conclusion chapters ◦ Need to add more content to the Research Limitations sections	David Odesola Dr. Shiny Verghese
August 24, 2022	12:00pm – 12:30pm	◦ Review on Dissertation Document ◦ Discussion on further writing improvements	David Odesola Dr. Shiny Verghese
August 31, 2022	12:00pm – 12:15pm	◦ Discussion on the Report finalization	David Odesola Dr. Shiny Verghese
September 5, 2022	12:00pm – 12:45pm	◦ Final Discussion and corrections	David Odesola Dr. Shiny Verghese