Bachelor of Science in Computer Science & Engineering

Smartphone based Automatic Traffic Signal Control System for Emergency Vehicles

by

Ziauddin Sameer Chowdhury

ID: 1504082

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Chattogram-4349, Bangladesh.

May, 2021

Smartphone based Automatic Traffic Signal Control System for Emergency Vehicles

Submitted in partial fulfilment of the requirements for
Degree of Bachelor of Science
in Computer Science & Engineering

by

Ziauddin Sameer Chowdhury

ID: 1504082

Supervised by

Md. Mynul Hasan

Lecturer

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Chattogram-4349, Bangladesh.

The thesis titled '**Smartphone based Automatic Traffic Signal Control System for Emergency Vehicles'** submitted by ID: 1504082, Session 2019-2020 has been accepted as satisfactory in fulfilment of the requirement for the degree of Bachelor of Science in Computer Science & Engineering to be awarded by the Chittagong University of Engineering & Technology (CUET).

Board of Examiners

Chairman

Md. Mynul Hasan
Lecturer
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Member (Ex-Officio)

Dr Md. Mokammel Haque
Professor & Head
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Member (External)

Dr. Md. Iqbal Hasan Sarkar
Assistant Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Declaration of Originality

This is to certify that I am the sole author of this thesis and that neither any part of this thesis nor the whole of the thesis has been submitted for a degree to any other institution.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. I am also aware that if any infringement of anyone's copyright is found, whether intentional or otherwise, I may be subject to legal and disciplinary action determined by Dept. of CSE, CUET.

I hereby assign every rights in the copyright of this thesis work to Dept. of CSE, CUET, who shall be the owner of the copyright of this work and any reproduction or use in any form or by any means whatsoever is prohibited without the consent of Dept. of CSE, CUET.

Signature of the candidate

Date:

Acknowledgements

The journey of writing this thesis towards its completion has only been possible though all the support and supervision from many individuals. This has been a fulfilling experience, both personally and professionally.

I would like to express my heartfelt gratitude towards my supervisor Mr. Md. Mynul Hasan by whose guidance this thesis becomes a reality. I am thankful for his continuous guidance and critical questions that has encouraged me to think out of my comfort zone and has levered my capabilities to take challenges that I never thought could be done by me.

I am thankful to the panel members for providing the approval of my work. I am grateful to my family members for their patience in bearing with me throughout this whole time. Their unconditional love and encouragement has provided support in every aspect of my life.

Abstract

Automatic Ambulance finding system, route management system and also traffic management system for ambulance has been a major challenge to overcome in modern era of techonology. No traffic signal management system particularly for emergency vehicles is yet fuctional. Moreover there is no system has yet been developed for finding nearby ambulance and medical for patient and to carry out those data to driver along with route automatically. To overcome this we developed a system with an android app. The app with a single tap can find nearest ambulance and medical automatically for patient. Patient also can keep track of ambulance. With the help of realtime database management system the app can let driver know the location of patient and nearest medical from the patient and show best route along with alternative routes between them. Also we placed nodemcu in traffic signal points. So whenever any ambulance reaches at about a certain distance from any signal point with the help of realtime database and nodemcu we can manage traffic signals along the way. This way our system helps both patient and driver with less user interaction and saves time.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

In todays world, traffic jams during rush hours is one of the major concerns. During rush hours, emergency vehicles like Ambulances, Police cars and Fire Brigade trucks get stuck in jams. Due to this, these emergency vehicles are not able to reach their destinations in time, resulting into a loss of human lives[1]. Also traffic congestion is another issue which leads to many traffic issues. One of the major effects of these traffic issues are faced by ambulances, fire-brigades and other emergency vehicles. Ambulance service is greatly affected because of traffic jams[2]. Also there is hardly any work has been done to find the ambulance automatically in case of any emergency and let it know the way to patient. For this thesis project we tried to make a system that can easily find the nearest ambulance and show it the route to the patient . The system also finds the nearest hospital automatically and controls traffic sytem automatically along the way. If applied properly and used correctly this may prove to be a very efficient update to existing systems. The overview of the proposed Smartphone based Automatic Traffic Signal Control System system and also the challenges encountered in the process, will be discussed in this chapter. The chapter also discusses the motivation and contribution of this specific thesis.

1.2 Automatic Ambulance finding with route management and Traffic Control

The part of finding nearby ambulance and drawing route is done basically through an android app. The app is very user friendly. It first takes a request from any user

and search for any ambulance nearby . It also searches for the nearest hospital from the patient. For this the app basically call the Google Places Api. Places Api returns many hospitals data.From those we pick the nearest hospital. The ambulance drivers must have the app installed and running in their smartphones. Once the system detects any available ambulance it sends a request to the driver's phone and shows him the location of Patient and also the nearest hospital from the patient. Then the app draw route between them. This calls the google maps Directions Api. The location of the driver is update in realtime and is shown in the patient's map.

Now for the part of controlling traffic signal we made a demo traffic signal system with a Nodemcu. We made the location of the signal point custom and put them in the firebase database server. The system is connected to the app through firebase. Whenever any ambulance reaches at about 100 meters from any signal point it sends the arduino a signal through firebase and makes the light of the ambulance's route to green and also the light of the opposite route to red. This way it controls traffic.

1.3 Difficulties

The major difficulties or challenges in finding the ambulance and managing traffic signal are enlisted as follows:

1. Finding the best routes in terms of short distance, less traffic,less travel time.
2. Designing the data flow
3. Finding nearest hospital and available closest driver.
4. Continuously calculating best routes among various alternate routes with the change of location.

1.4 Applications

Although the system we developed is not yet fully functional and perfect but with proper developement it can be used in many sectors. It can become an efficient tool in helping people in many ways. Some of them are:

1. It can be adopted by many hospitals to create a network and thus create an userbase.
2. It can adopted by any private ambulance service company .
3. By shifting base it can be adopted by many emergency service oranizations like Fire Brigade.
4. Traffic Signal Center with proper improvement can adopt this system.

1.5 Motivation

The motivation of working on this project originated from the urge to increase the safety of the citizens of this country by using available technology. Additionally, there is always room for improvement in this project. The primary motivation working behind this thesis can be enlisted as follows:

1. To find the available nearest ambulance automatically and quickly.
2. Automatic route selection which can save time.
3. Traffic signal management for emergency vehicles.
4. To develop an user friendly and efficient feedback system along with an integrated android app to provide realtime update to the users.

1.6 Contribution of the thesis

Thesis or Research work is performed, to achieve a specific set of goals, whether it is to define a new methodology or to improve the existing ones. In this thesis, the main focus was given to the process of finding nearby ambulance and hospital automatically, to help driver with showing routes to the destinations, to control

traffic signal and to develop an efficient and user friendly android app to be used as the feedback system. The primary contribution of this thesis is the following:

1. Developement of an app that can choose nearest available ambulance for any patient.
2. An app that can show driver the routes automatically.
3. An arduino based traffic control system.

1.7 Thesis Organization

The following is how the remainder of this study paper is organized:

1. Chapter 2 discusses about previous research works in the field of traffic management system and route management.
2. Chapter 3 depicts the proposed technique in full, including the creation of a model that demonstrates how our system discovers the nearest idle ambulance, calculates the optimal route, and manages traffic. It also depicts the Android mobile application's dataflow.
3. Chapter 4 provides the description of the working dataset and analysis of the performance measure for the proposed method.
4. The overall summary of this thesis work is presented in Chapter 5 along with some future recommendations.

1.8 Conclusion

An overview is presented in this chapter. This chapter has discussed the epitome of the entire system, as well as the challenges. The inspiration for this work, as well as the contributions made, are also detailed here. The background and current situation of the problem will be discussed in the following chapter. In this chapter, an overview of is provided. Along with the difficulties, the summary of the smartphone based automatic traffic signal control is described in this chapter.

The motivation behind this work and contributions are also stated here. In the next chapter, background and present state of the problem will be provided.

Chapter 2

Literature Review

2.1 Introduction

There are a lot of contribution in traffic management system based on iot in recent years. But a few has been developed where there is involvement of smartphone with the help of iot device to do certain work and put them all together to built a system that works as intended. In short there are a few works where the smartphone based system works efficiently with an elaborate server system that also links up with iot device to make a perfect system. Now most of the IoT based applications have android app support. So it is also necessary to modernize the system by integrating a user friendly and efficient android app in the system. Whole process is divided into three major parts which are automatic ambulance finding , route creating and traffic signal management. A detailed description of these methods is provided in this chapter.

2.2 Related Literature Review

There are a lot of work done in this Iot based traffic management system. Each system has their own advantages and limitations. In existing solution there are many modules need like RFID tag, GSM module, Arduino Uno, raspberry pi etc. Also the process of user sending request is time consuming and no process shortest path finding is implemented.

Sneha Tammishetty et al [3] developed a system where request of an ambulance is done through an android app and the location of ambulance is processed through GSM module and Arduino to change the traffic signal. But this system is not

efficient when there is more than one ambulance in a certain time in two different joining route also it does not search for nearby ambulance.

S.N.Sivaraj et al [4] in his proposed system he designed the system where sensors automatically detects if a person is ill or needs medical emergency and by doing that it requests for an ambulance but in controlling traffic signal he didn't put any controller to automate the process of changing signal automatically.

Wani et al [5] developed a system where the computing unit search for the nearby ambulance via GSM module that is kept in the vehicle and after that it starts the timer and buzzer of the ambulance and by controller unit it controls the traffic signal based on location. This method is costly as it needs more components while the phone can be used instead of GSM module.

V.Bali et al [6] developed a system in which the RFID reader scans the RFID tag applied on the ambulance and updates the upcoming traffic light to switch to green and displays a message to vehicles ahead of ambulance to provide a "Green Corridor" by shifting other lanes. This system comprises whole traffic system and functions for emergency situations. This system only controls traffic no ambulance finding process is developed.

V.Khalique et al [7] developed a system where any user can request for an ambulance and the ambulance finding process is automatic is processed through AADS server. But in his design there is no method for finding the nearest hospital and send the information to the nearest driver. Also no system is designed for controlling traffic signal.

Shruthi U et al [8] in her system she used RFID tag and Arduino to control the traffic signal . The system uses Google maps to access the location of the user. But in the request portion of the system there is no process of finding the nearby ambulance. And also there is no indication of applying priority to the ambulance in the situation of two ambulances in two different root at the same time.

2.3 Conclusion

A detailed description of the literature review is provided in this chapter. The discussion was divided into three parts depending on the procedure of smartphone based automatic traffic control system for emergency vehicles. Different strategies and methods used by the researchers were described here. The next chapter contains an explanation of the methodology of the Smartphone based Automatic Traffic Control System for Emergency Vehicles in detail.

Chapter 3

Methodology

3.1 Introduction

This section contains all the detailed information about how we designed our system and how it works with appropriate figures. The first is mainly our proposed system which basically describes and shows the main parts that builds up the system. Later we divided it into four parts and and described each part in this section. Then we showed how we designed and created the android app and it's functionality. We have also added all the necessary figures and screenshots to understand better.

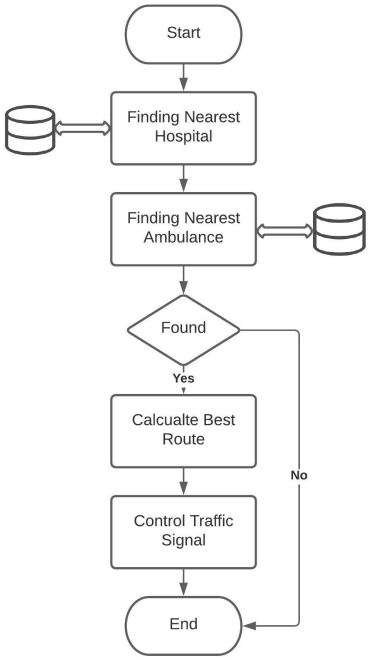

Figure 3.1: Methodology of the proposed system

3.2 Steps involved in Proposed System

Fig 3.1 shows the basic layout of our system design. It involves many part . First there is an android app for both user/patient and driver. Main activity for both patient/user and driver are different. The app is connected to the Firebase which is a realtime database server, cloud-hosted NoSQL database that lets one store and sync data between your users in realtime [9]. The location of both driver and user updates in realtime in Firebase. Then whenever any user sends request for an ambulance the nearest available driver accepts the request automatically . Then the driver can see the patient location in his map and the patient can see driver in his map. Now the app calculates the best route possible along with alternative routes from driver to patient and patient to nearest hospital and are shown in the drivers map. Now we placed our Nodemcu based Traffic Signal Control system in signal points. Now whenever any ambulance reaches within 100 meters from any signal point then the app will send a signal to the nodemcu through firebase. This signal will override the current processing of signal light of nodemcu and make the traffic light of the ambulance's route to green and other's to red. This is the way our system works.

3.3 Android App

We have integrated android app into the system. This app has basic login layout. It can create separate account for patient and driver and can store app data in the local storage of any android smartphone . It supports api level above 25. We have also integrated google maps api . Through which user or driver can easily navigate . We have connected the app with firebase realtime database system to update location of users and drivers in realtime and store information. The app can also store basic information and show them. Now in fig 3.2 we can see the basic parts of our android system. We will discuss about them in later topics .

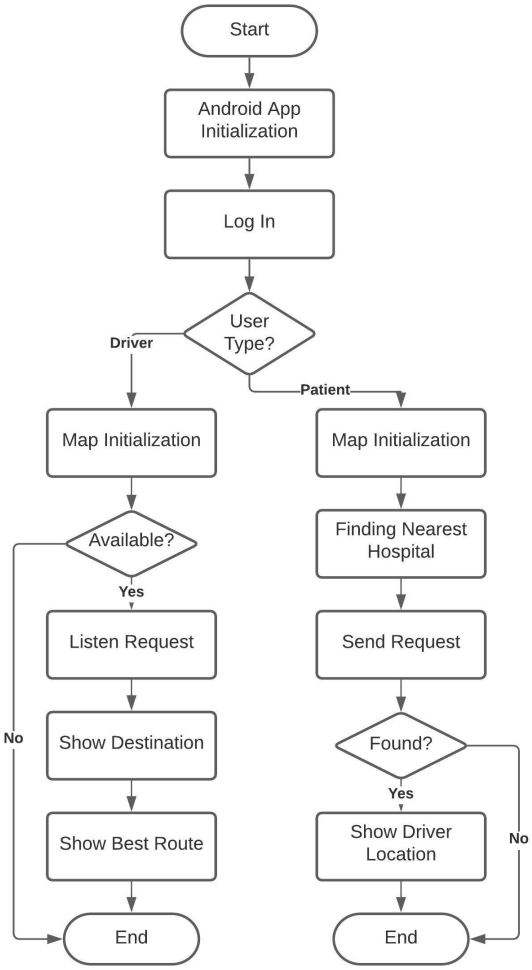

Figure 3.2: User and Driver Part of the App

3.3.1 Login System

We can see from fig 3.2 that whenever any user opens the app depending on the user type it initializes it's main activity. Now the data of the user is stored on the local storage . Whenever the app is opened first it fetches data from local storage or appdata. If any type of user was logged in previously then the app will initialize the maps activity otherwise it will ask user to login and will initialize the log in screen activity. Now if user haven't created any account yet there is a registration activity which user can open. Now depending on the user type they can crate account as a patient or a driver. All of the data like user email, phone number, name and password will be stored in firebase . After that user will be automatically moved to login page . User can login then and then the app will initialize map activity.

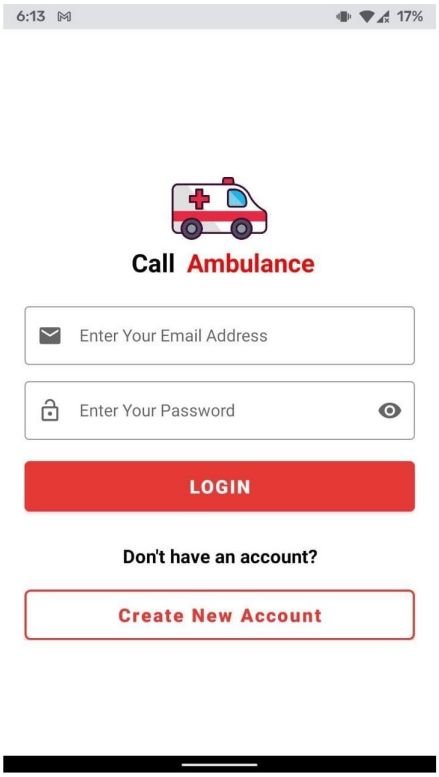

Figure 3.3: Login Activity

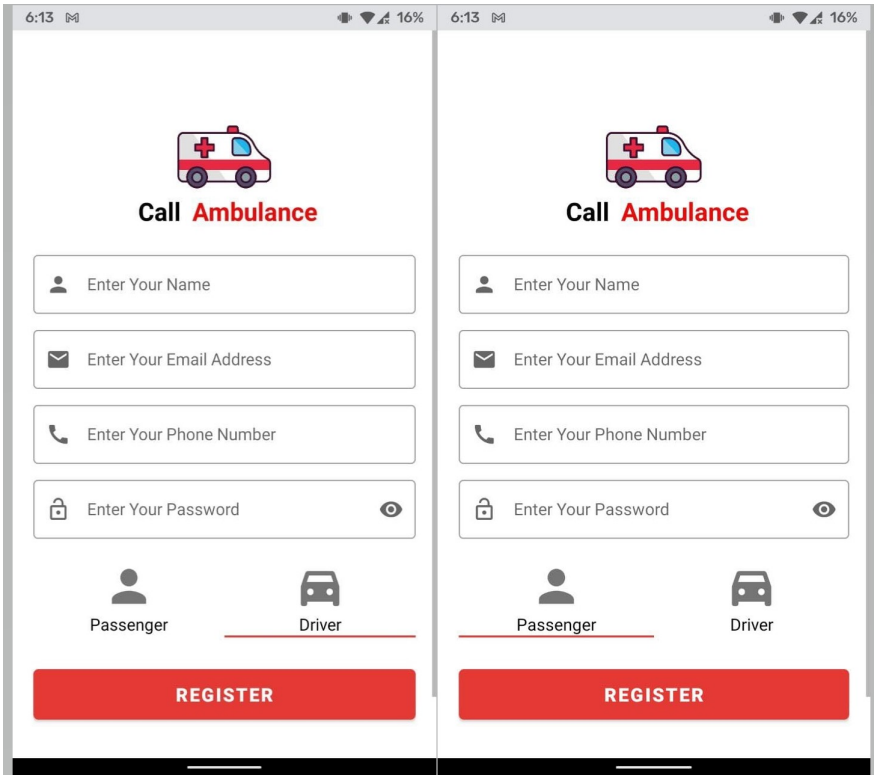

Figure 3.4: Driver and Passenger Registration Activity

3.3.2 User Part

Now From fig 3.5 we can see that After login the app will initialize map activity. As soon as patient's map activity initializes the app will call nearby places api. PLaces Api will return all the nearby hospital in a json object file. Now we will calculate the distance between all the hospitals and patient. By comparing we will select the hospital with less distance. Now with geofire the location of the patient and hospital along with certain information will be stored in the firebase realtime datatbase with an unique id. GeoFire is an open-source library for Android that allows you to store and query a set of keys based on their geographic location. GeoFire selectively loads only the data near certain locations, keeping applications light and responsive, even with extremely large datasets [10]. So the location data will be updated realtime Now user can add their username , profile photo in their profile and can logout anytime.

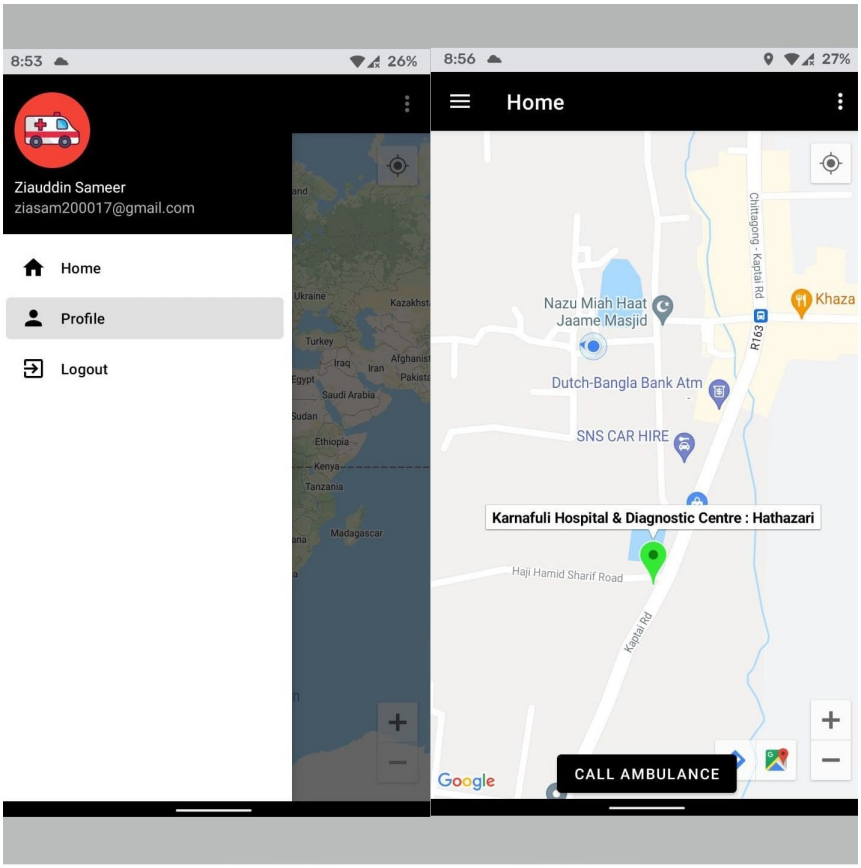

Figure 3.5: User Activity (Part-1 User Profile(left))/(Part-2 Nearest Hospital(Right))

3.3.2.1 Finding Nearest Hospital

Now when map activity initializes a function called nearestHospital() is called. When the function is called then with the help of Google Paces API, all the nearby hospitals data like (name, vicinity, location) within 2 kilometers are stored in a json file. Now from the location data (lattitude and longitude) we calculate the nearest hospital from the patient location . For this we measured the distance between all the hospitals and patient location . By this we selected the nearest hospital and showed on the map. Now data of the nearest hospital is stored on the firebase with the patients information and location.

3.3.2.2 Sending Request

From the patient's map activity we can see that there is a call ambulance button. Now whenever any user presses that button a user request is placed on the server with the user location and nearest hospital location . Now a function named closestDriver() is callled. Now the function retrieves realtime drivers data from the available-drivers child. Now it enters into a loop. First it searches for drivers in 1 kilometer range. if not found it increaments it's range. After a certain range it moves out of that loop and shows a message (left part of the Fig 3.6). If it detects any ambulance. It sends a pickup request to the ambulance and the ambulance position is showed in the map. The position of the drivers updates in realtime in the patient's map. We can see this in the right part of the patient's map.

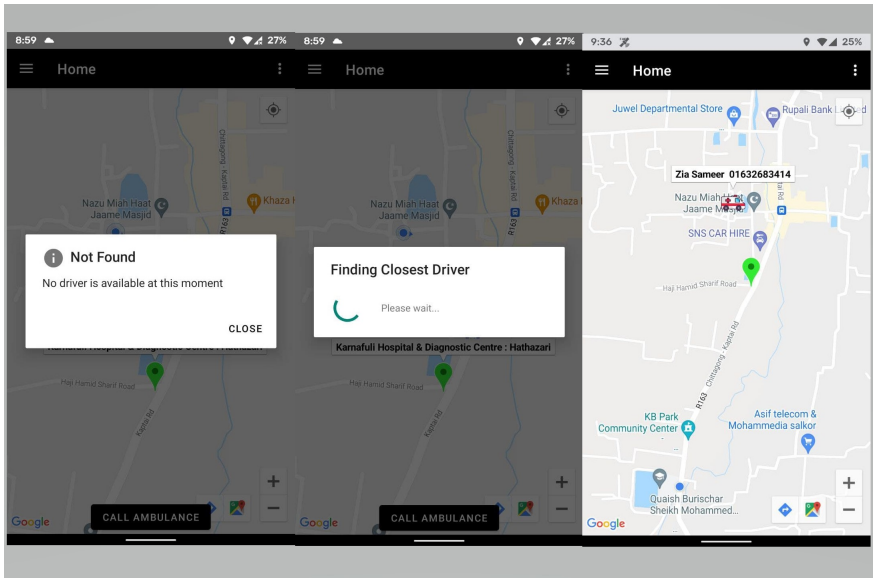

Figure 3.6: Sending Request and Finding Ambulance

3.3.2.3 Showing Driver Location

Now from the fig 3.6 we can see that after finding the closest driver the driver location is visible on the map. It is marked with an ambulance marker. Now the marker will update in realtime in patient's map whenever the driver moves from it's position. When the driver will reach within 10 meters from the patient's location then the patient will be greeted with a message saying "Ambulance Arrived".

3.3.3 Driver's Part

Now From fig 3.5 we can see that After login the app will initialize map activity. Now with geofire the location of the driver along with certain information will be stored in the firebase realtime datatbase with an unique id. Now with the help of geofire and firebase location data will be updated in realtime. Now if the driver doesn't have any request the driver data along with location data will be stored in avaialbleDrivers Child. Now user can add their username , profile photo in their profile and can logout anytime.

3.3.3.1 Listening Request

If any user is logged in as a driver. Then the driver is set as an available driver. His location data with name and number are stored in availableDrivers Child in firebase. When the driver is available he is always in listening request mode . We can see that from fig 3.7 left part . The assignedCustomer() function always check the passengerRequests child in server. Whenever any patient sends a request then the assigned customer with it's value even listener picks the customer information and assigned them in the nearest available driver's driver id as a request from customer. The location of the patient is shown in the drivers map Then the driver moves out from availableDriver child and the driver is no longer available until it picks up patient and drop him in the destination.

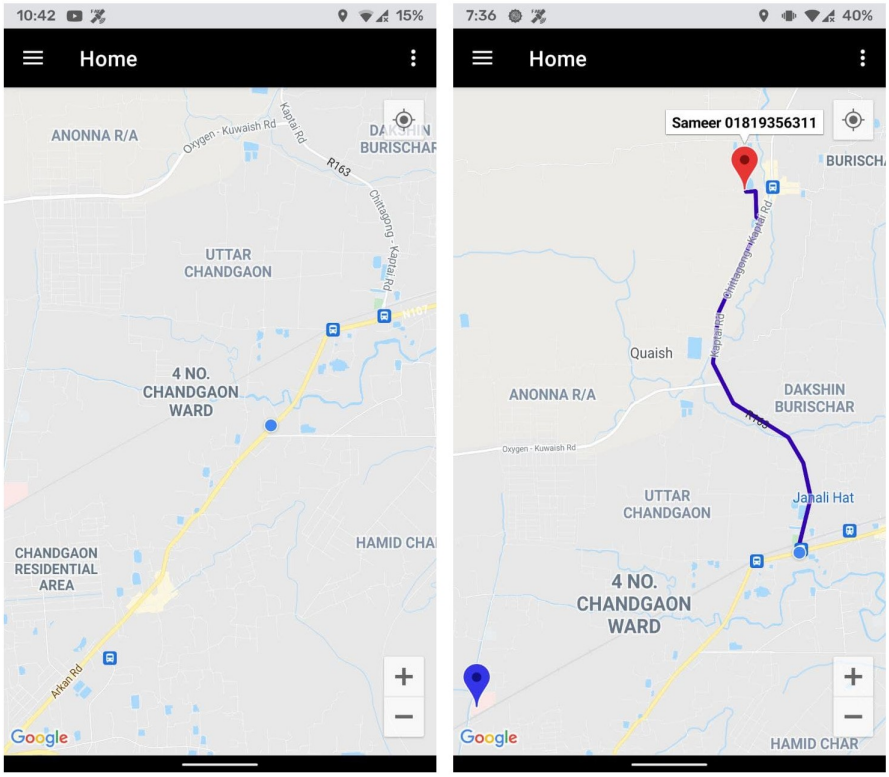

Figure 3.7: Driver's Part

3.3.3.2 Show Destination

Now assignedCustomer() when detects any request from patient it then calls assignedCustomerdestination() function. Now assignedCustomerDestination function in return picks the destination data (Nearest Hospital) data from the passengerRequests child. Now the location of that destination is then shown in the map as we can see from the right part of the fig 3.7.

3.3.3.3 Show Best Route

AsssignedCustomer() function will also call findBestRoute () function. Now the find best route function actually calls Google Directions Api and we define the transit mode when calling the Api. And when calling api we are seting alternative to true so that we can get all possible routes. Also we are not setting any way points cause if we set way points like both driver to patient and patient to hospital we won't get any alternative routes. So we think it would be the

best practise to split the route finding into two parts . It then stores the data and traffic layer of all possible routes like traffic condition, travel distance, travel time, mode of transport and etc in a json file. Now an array with the list of routes is created. Now the data of the routes are always updated realtime. and Google provides the data based on realtime traffic condition, shortest route and vehicle type and transportation mode. Now we could apply sorting algorithm taking some parameters like distance, duration and traffic condition but as it is an emergency vehicles sorting in terms of travel time would be the best choice as google already provides the travel time considering traffic condition. We are now choosing best route by considering duration and comparing the time with a pre-defined value. The route with minimum duration is considered as the best route and is plotted in the map using purple color. This condition is always checked with each location update and the best route may change based on realtime traffic even when travelling. We also print the all possible routes with different colors if any as you can see from Fig 3.8 .

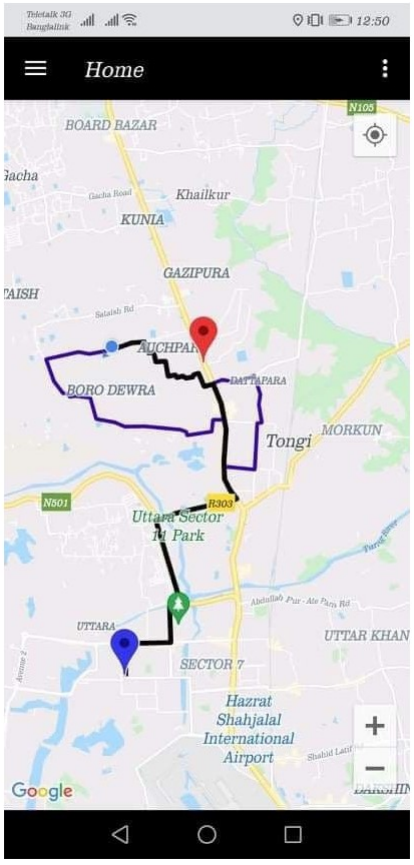

Figure 3.8: Find Best Route

3.3.4 Controlling Traffic Signal

We have designed a time based traffic signal system . A nodemcu is used for designing the traffic control signal. It has built in wifi. We have uploaded the code using Arduino IDE. For testing purposes we have picked some custom co ordinates as our signal points. Our system mimics the usual Traffic signal system. We have connected the system with firebase. Now our driver side of the app actually always checks the distance between custom co-ordinates and driver location with a function call named locatesignal(). Now if the driver reaches within 100 meters from any signal points it sends a "ON" signal to our traffic signal system through firebase. Upon receiving the signal our traffic signal system turns the light to green of Ambulance's route and turns others to red. and when the distance again increases and crosses 100 meter it again sends a "OFF" signal to the system . Again upon Receiving the signal the traffic signal system returns to it's normal state. Thus the Traffic signal works . The system is shown in Fig 3.9

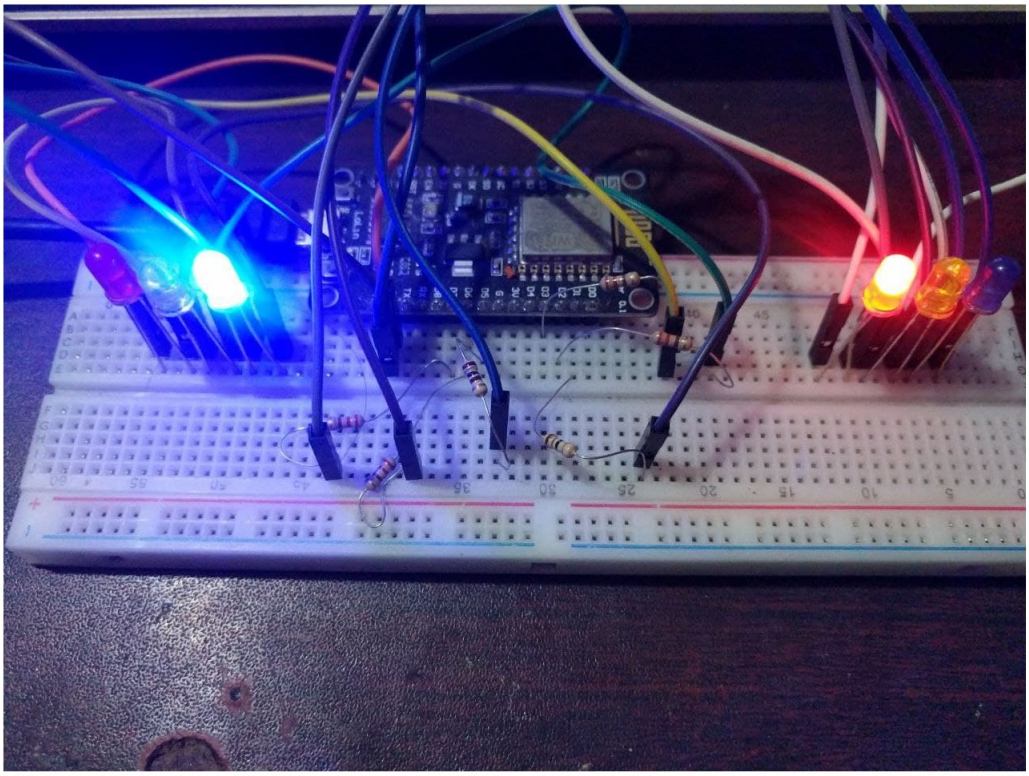

Figure 3.9: Traffic Signal System

3.3.5 App's Backend

Firebase Realtime Database is used as the backend of the app where all activity like passenger and drivers all the information, location customer request etc are stored. Only the system administrater can alter the data stored.

3.3.5.1 Firebase Realtime Database

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of the clients share one Realtime Database instance and automatically receive updates with the newest data. The Realtime Database provides a flexible, expression-based rules language, called Firebase Realtime Database Security Rules, to define how the data should be structured and when data can be read from or written to. When integrated with Firebase Authentication, developers can define who has access to what data, and how they can access it. The Realtime Database is a NoSQL database and as such has different optimizations and functionality compared to a relational database. The Realtime Database API is designed to only allow operations that can be executed quickly.

Figure 3.10: Firebase

3.3.6 Nodemcu

NodeMCU is open source platform, it's hardware design is open for edit/modify/build. NodeMCU Dev Kit/board consist of ESP8266 wifi enabled chip.The ESP8266 is a low-cost Wi-Fi chip developed by Espressif Systems with TCP/IP protocol. It can be used for a wide variety of IoT applications. It offers a complete and self contained Wi-Fi networking solution, allowing it to either host the application or to offload all Wi-Fi networking functions from another application processor. ESP8266 NodeMCU has powerful on-board processing and storage capabilities that allow it to be integrated with the sensors specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. [11]

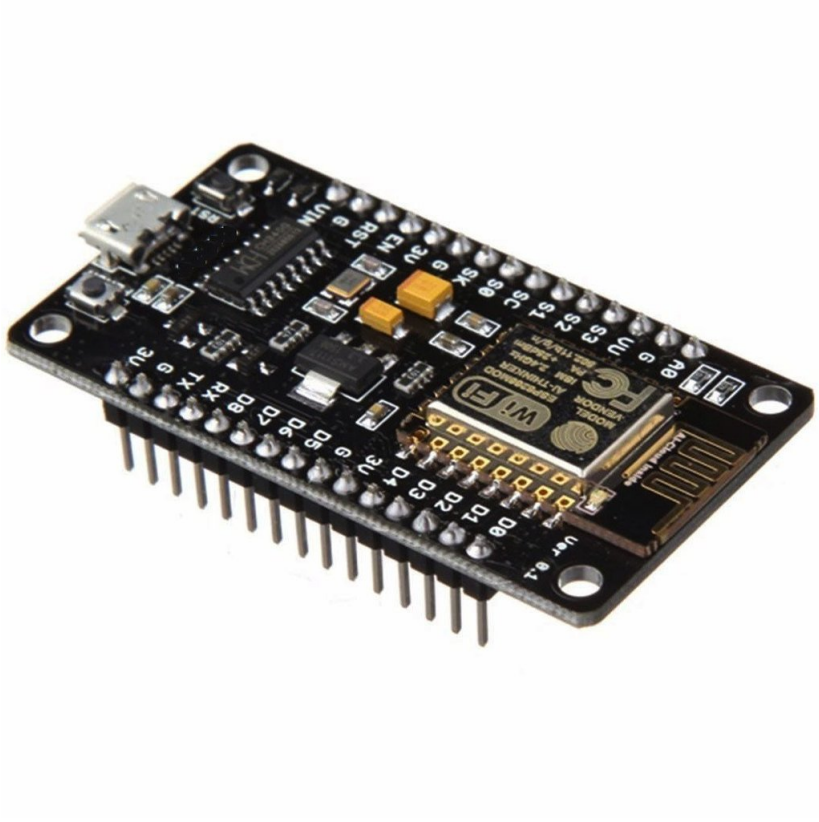

Figure 3.11: Nodemcu

3.3.7 Android Studio

Android Studio, which is based on IntelliJ IDEA, is the official Integrated Development Environment (IDE) for developing Android apps. Android Studio, in addition to IntelliJ's strong code editor and developer tools, includes additional

capabilities that improve your efficiency when developing Android apps, such as:

1. A flexible Gradle-based build system
2. A fast and feature-rich emulator
3. A unified environment where you can develop for all Android devices

Android Studio version 4.2.1 is used for development of the app.

3.4 Conclusion

We've given a quick summary of our smartphone-based emergency vehicle traffic signal control system. The Android application's working and development process has been explained. The next chapter will go over the entire analytical process.

Chapter 4

Results and Discussions

4.1 Introduction

We have given a brief description about how we designed the system and built it with figures. Now the performance of the system and the result based on data we gathered will be discussed in this chapter. Now the location data of the patient and the driver is mainly taken from smartphone's gps. And the firebase realtime database is used for storing and updating those data in realtime.

4.2 Nearby Ambulance and Hospital Selection(First Test)

From the fig 4.1 we can see that in first case there are three available drivers. Now in this case the passenger's UserID is "X3tZdwPYJBWIbywRmjcetm0HCIe2". Now the closest driver from this passenger in the first case is with UserID "hkRrL5jxWdWncD6s7RmpsWftwWy1". Now whenever the passenger will send request for an ambulance the the closest Driver with given ID will be assigned from these three drivers. We can see this in fig 4.2. The visual representation of the process is shown in fig 4.3.

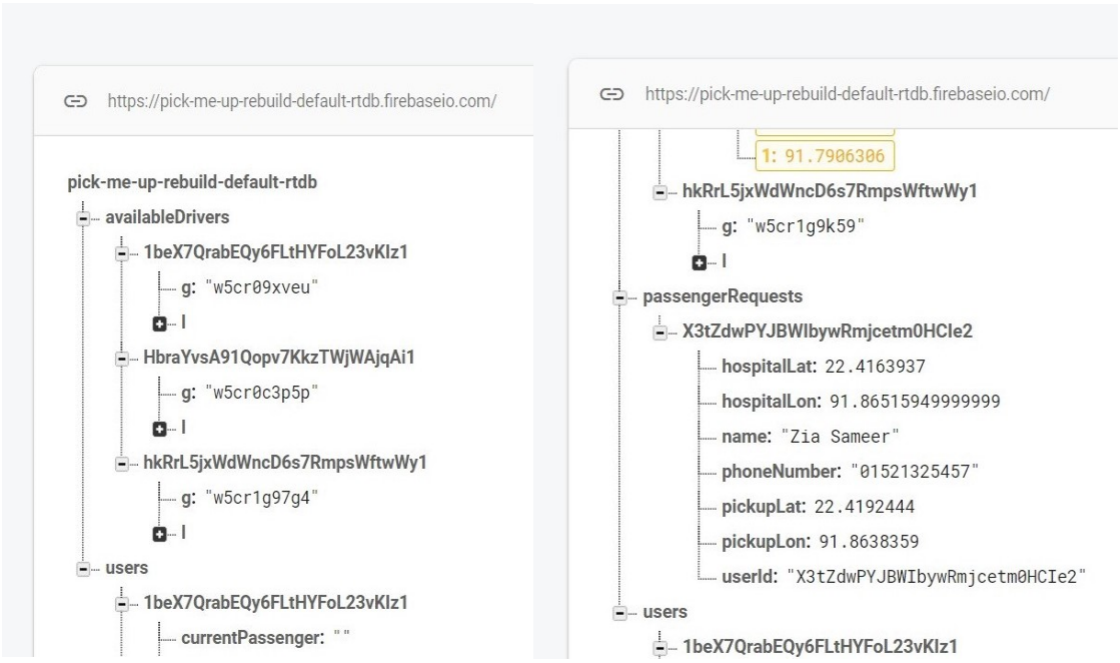

Figure 4.1: Closest Driver Selection(First Test)

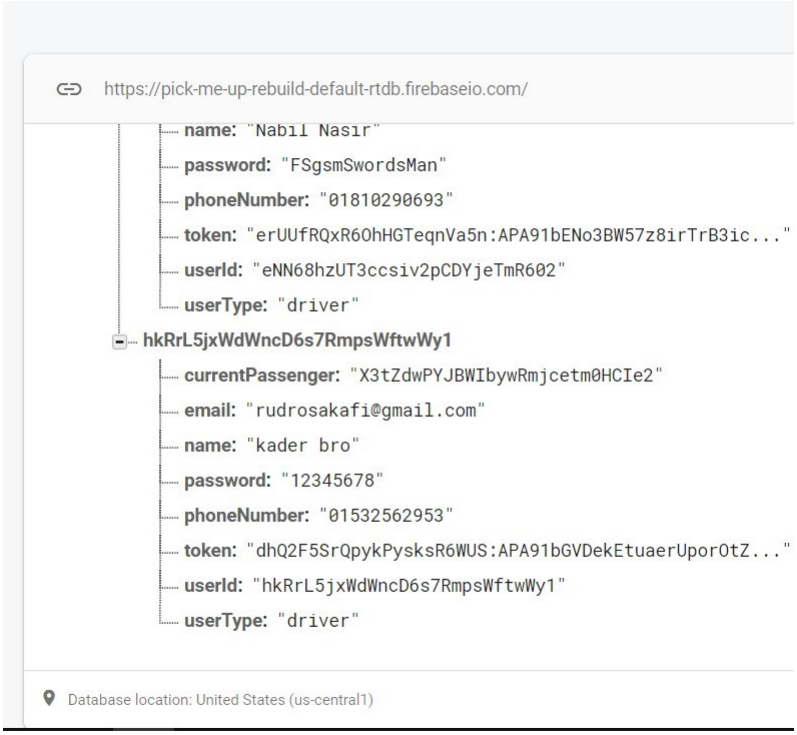

Figure 4.2: Closest Driver Selection(First Test)

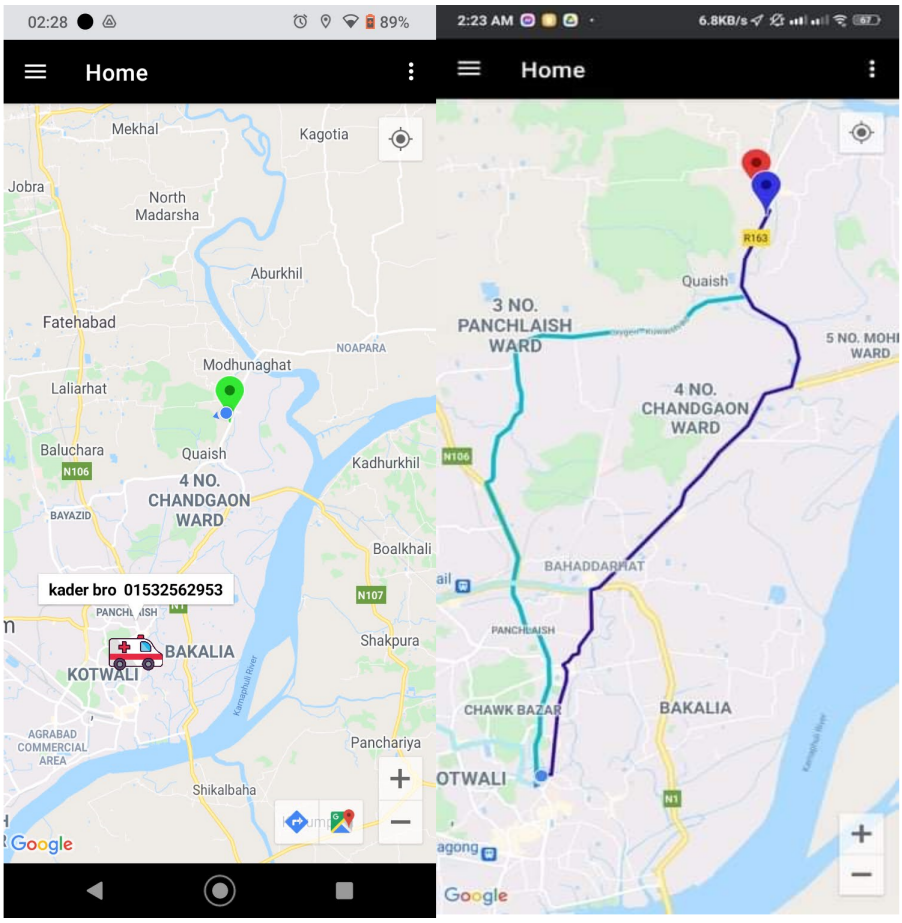

Figure 4.3: Visual representation(First Test)

4.3 Nearby Ambulance and Hospital Selection(Second Test)

Now in the second case fig 4.4 the closest available driver is changed. Now in this case also the passenger's userID is "X3tZdwPYJBWIbywRmjcetm0HCIe2".But now the closest driver from this passenger is with UserID "eNN68hzUT3ccsiv2pCDYjeTmR602". Now whenever the passenger will send request for an ambulance the closest Driver with given ID will be assigned from these three drivers. We can see this in fig 4.5. The visual representation of the process is shown in fig 4.6 .

Figure 4.4: Closest Driver Selection(Second Test)

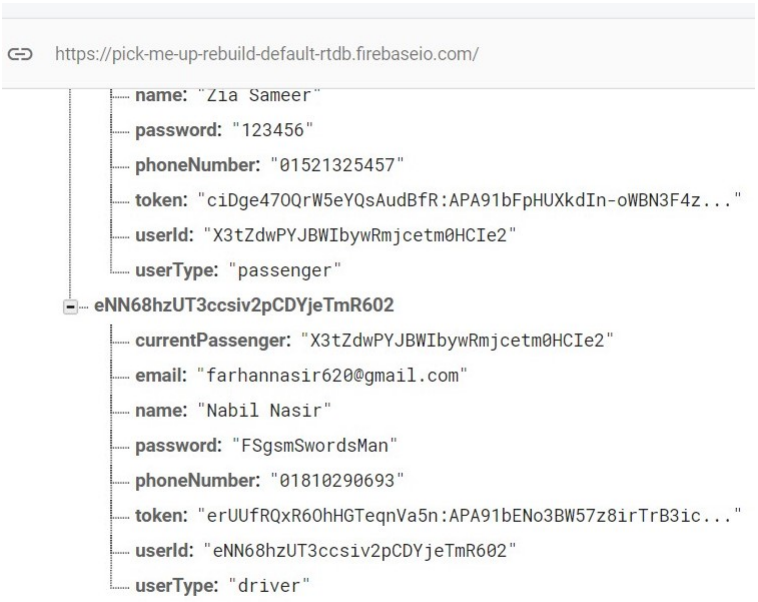

Figure 4.5: Closest Driver Selection(Second Test)

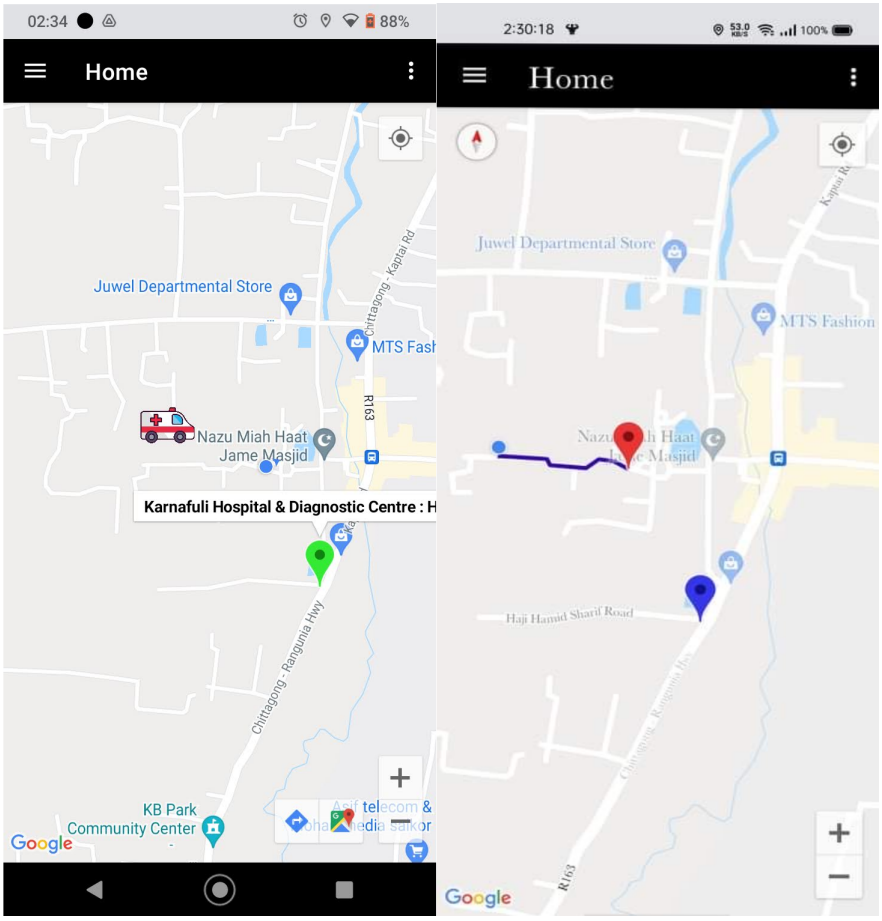

Figure 4.6: Visual representation(Second Test)

4.4 Collecting Route Data from Json File

From the fig 4.5 and fig 4.6 we can see that in the json file there is a lot of information regarding a route . Now these are all stored in an array. As we are considering routes for an emergency vehicle like ambulance we must have to find a route that has less travel time as compared to others. Here we could take many parameters for finding or calculating best route like travel distance , fuel consumption, traffic in those routes. Now the travel time google returns is already based on the traffic situation in those routes. Now we calculated the best route from the all possible route by comparing the travel time of each route ro another. And if best route found we print them in map with purple color. For printing routes we take the polylines object from json file. Here we also show all alternative routes with different colors. When travelling any alternative routes may become main route. This route finding methos is always in working mode

and is updated realtime with each location update of the driver. We are first printing routes from driver to patient and later after picking up patient then we print the routes from patient to hospital.

Figure 4.7: Route data from Json

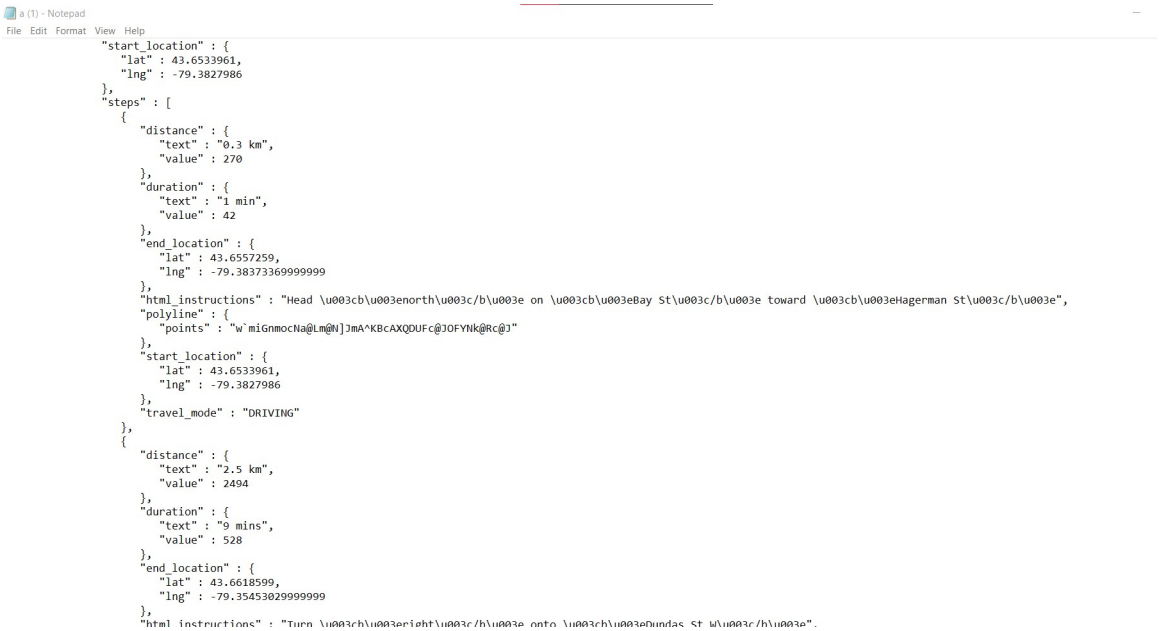

Figure 4.8: Polyline Data

4.5 User Request and Driver Availability Testing

Now whenever any user send a request for a driver the information of user like location data , nearest hospital location, userid are stored in a child called passengerRequests with passenger id . If a driver is available only the location data of that driver is stored in a child callled availableDrivers with that drivers uniques id. Now after making a request and if any driver is found nearby the passenger's data from that passengerRequests child are placed inside that available driver id. and That driver then moves from available to working mode. Fig 4.7 and fig 4.8 shows the database information.

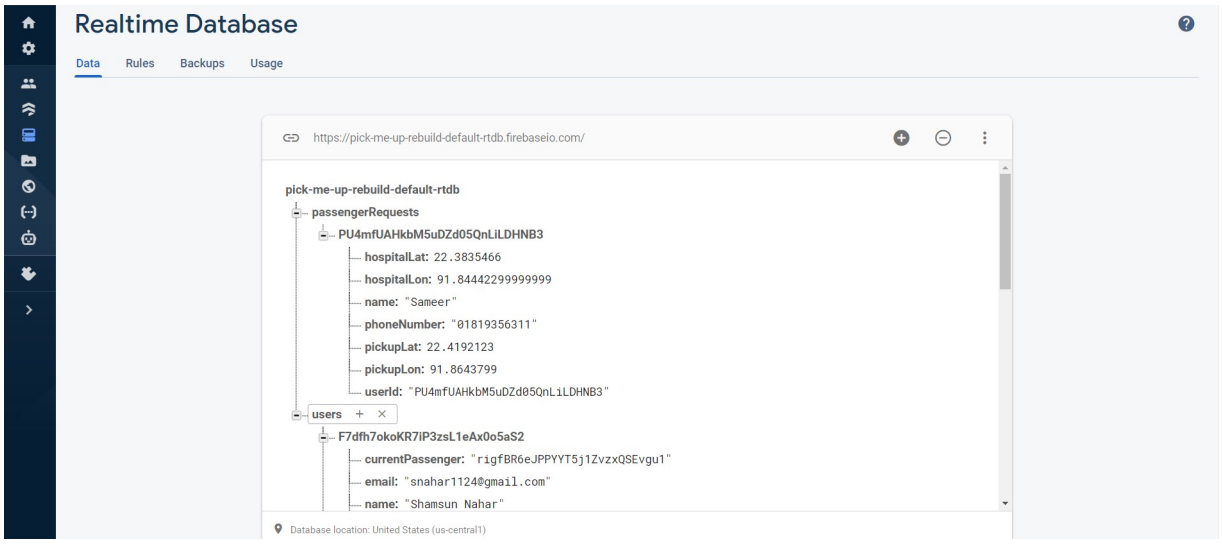

Figure 4.9: Passenger Request Child

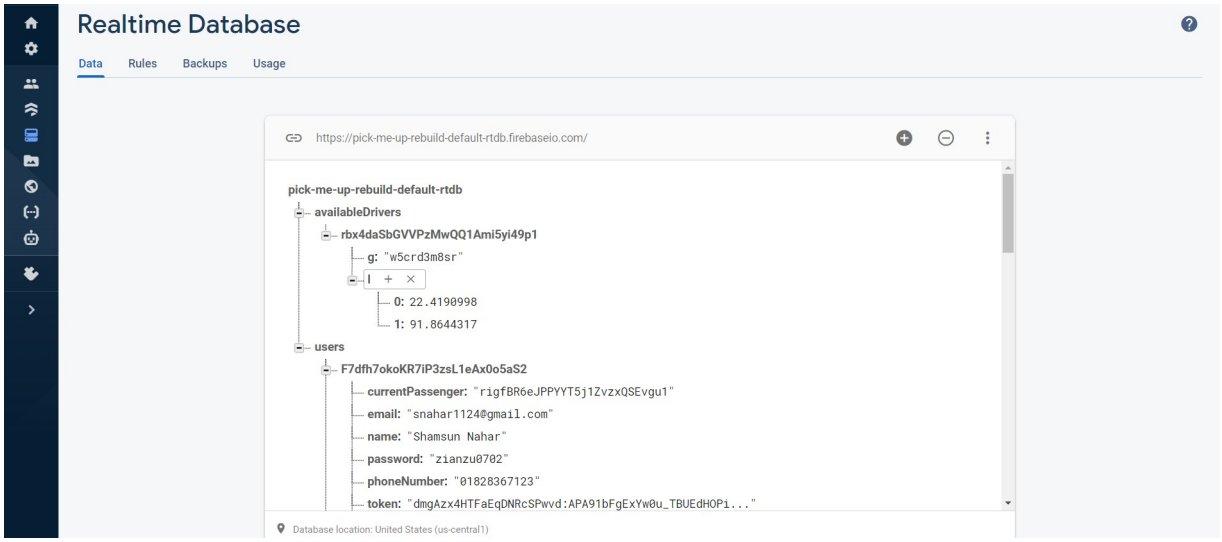

Figure 4.10: Available Driver child

4.5.1 Impact on Our Life

If we consider normal way to do this we have to go through many steps. We first have to search for any medical nearby from google map. Then we have collect number of that hospital and have to call to check if any ambulance is available. Those may waste many time . also waiting on traffic signal points and if stuck searching for alternative routes manually may also kill valuable time. Moreover we can't track the ambulance if any found. So by our project the time of manually finding nearby hospital and driver will be saved which may become life saving for many patient.

4.6 Conclusion

In this chapter we have provided the data we have used to find best route . We have also provided a short summary about how we stored our dat in our server and impact of this thesis project in our life . The thesis work is brought to a close in the next chapter.

Chapter 5

Conclusion

5.1 Conclusion

The original idea of this project was conceived from the facts and figures from the daily newspaper and media depicting the number of deaths due to not getting treatment timely. Using this application users can find the nearest hospital and the closest ambulance driver with just one click. Then the request will be sent to the closest active ambulance driver and among various routes the most efficient route will be highlighted on the drivers end. This system is effective and user friendly and thus its usage is not restricted or limited to any class of users. This app can also be used for other emergency services. Good health care facility is a necessity for every individual. Hence, this helps to provide proper assistance even in the rural areas.

5.2 Future Work

There is plenty of room for developement in ths system or model. One can include many useful features like Some renowned Doctors Profile and field of work with contact information. This can be location specific. Also the efficient route can be further optimized by taking consideration of more parameters. The app can be further optimized to check patient's health data based on smartphone's sensor data.

References

[1] G. K. Pal and N. Salman, 'An advanced ambulance rescue system using prioritized traffic switching,' *International Journal of Research*, vol. 4, no. 17, pp. 2776–2782, 2017 (cit. on p. 1).

[2] O. Udawant, N. Thombare, D. Chauhan, A. Hadke and D. Waghole, 'Smart ambulance system using iot,' in *2017 International Conference on Big Data, IoT and Data Science (BID)*, IEEE, 2017, pp. 171–176 (cit. on p. 1).

[3] S. Tammishetty, T. Ragunathan, S. K. Battula, B. V. Rani, P. RaviBabu, R. Nagireddy, V. Jorika and V. M. Reddy, 'Iot-based traffic signal control technique for helping emergency vehicles,' in *Proceedings of the First International Conference on Computational Intelligence and Informatics*, Springer, 2017, pp. 433–440 (cit. on p. 6).

[4] P. Manikanta, S. S. Hussian and R. T. Kodi, 'Iot ambulance with automatic traffic light control,' in *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, IEEE, 2019, pp. 1–3 (cit. on p. 7).

[5] M. M. Wani, S. Khan and M. Alam, 'Iot based traffic management system for ambulances,' *arXiv preprint arXiv:2005.07596*, 2020 (cit. on p. 7).

[6] V. Bali, S. Mathur, V. Sharma and D. Gaur, 'Smart traffic management system using iot enabled technology,' in *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, 2020, pp. 565–568. DOI: 10.1109/ICACCCN51052.2020.9362753 (cit. on p. 7).

[7] V. Khalique, S. Shaikh, M. Dass, S. M. S. Shah and M. Zaib, 'Automatic ambulance dispatch system via one-click smartphone application,' *Indian Journal of Science and Technology*, vol. 10, no. 36, pp. 1–9, 2017 (cit. on p. 7).

[8] *Iot based smart ambulance system - google search*, https://www.google.com/search?q=IOT+BASED+SMART+AMBULANCE+SYSTEM&oq=IOT+BASED+SMART+AMBULANCE+SYSTEM&aqs=chrome..69i57j69i59j69i60.197j0j4&sourceid=chrome&ie=UTF-8, (Accessed on 05/23/2021) (cit. on p. 7).

[9] *Firebase realtime database*, https://firebase.google.com/docs/database, (Accessed on 05/23/2021) (cit. on p. 10).

[10] *Firebase/geofire-android: Geofire for android apps*, https://github.com/firebase/geofire-android, (Accessed on 05/23/2021) (cit. on p. 13).

[11] *Internet of things and nodemcu a review of use of nodemcu esp8266 in iot products - google search*, https://www.google.com/search?q=Internet+of+Things+and+Nodemcu+A+review+of+use+of+Nodemcu+ESP8266+in+

IoT+products&oq=Internet+of+Things+and+Nodemcu+A+review+of+use+of+Nodemcu+ESP8266+in+IoT+products&aqs=chrome..69i57j69i60.388j0j9&sourceid=chrome&ie=UTF-8, (Accessed on 05/23/2021) (cit. on p. 20).