DAYANANDA SAGAR UNIVERSITY

KUDLU GATE, BANGALORE – 560068

Bachelor of Technology
in
COMPUTER SCIENCE AND ENGINEERING

Major Project Phase-II Report

ACCURATE LOCATION-BASED ROUTING FOR EMERGENCY RESPONSE BY LEVERAGING APP DEVELOPMENT

By

Tejaal M -ENG19CS0334
Amitash M S- ENG19CT0004
Rahul K P- ENG19CT0028
S Adnan Mohamed- ENG19CT0030
Sumedh Hagaldivte- ENG19CT0042
Batch no – **78**

Under the supervision of
Mr. Baskar Venugopalan
Professor of Practice,
Department of Computer Science and Technology

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY,
BANGALORE

(2022-2023)

DAYANANDA SAGAR UNIVERSITY

School of Engineering

Department of Computer Science & Engineering

Kudlu Gate, Bangalore – 560068
Karnataka, India

CERTIFICATE

This is to certify that the Phase-II project work titled **"ACCURATE LOCATION-BASED ROUTING FOR EMERGENCY RESPONSE BY LEVERAGING APP DEVELOPMENT"** is carried out by **Tejaal M (ENG19CS0334), Amitash M S (ENG19CT0004), Rahul K P (ENG19CT0028), S Adnan Mohamed (ENG19CT0030), Sumedh Hagaldivte (ENG19CT0042),** bonafide students of Bachelor of Technology in Computer Science and Engineering at the School of Engineering, Dayananda Sagar University, Bangalore in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering, during the year **2022-2023**.

Prof. Baskar Venugopalan	**Dr. Girisha G S**	**Dr. Udaya Kumar Reddy K R**
Professor of Practice Dept. of CS&T, School of Engineering Dayananda Sagar University	Chairman CSE School of Engineering Dayananda Sagar University	Dean School of Engineering Dayananda Sagar University
Date:	Date:	Date:

Name of the Examiner	**Signature of Examiner**
1.	
2.	

DECLARATION

We, **Tejaal M (ENG19CS0334)**, **Amitash M S (ENG19CT0004), Rahul K P (ENG19CT0028), S Adnan Mohamed (ENG19CT0030), Sumedh Hagaldivte (ENG19CT0042),** are students of eighth semester B. Tech in **Computer Science and Engineering**, at School of Engineering, **Dayananda Sagar University**, hereby declare that the Major Project Stage-II titled "**ACCURATE LOCATION-BASED ROUTING FOR EMERGENCY RESPONSE BY LEVERAGING APP DEVELOPMENT"** has been carried out by us and submitted in partial fulfilment for the award of degree in **Bachelor of Technology in Computer Science and Engineering** during the academic year **2022-2023**.

Student	**Signature**
Name 1: Tejaal M **USN :- ENG19CS0334**	
Name 2: Amitash M S **USN : ENG19CT0004**	
Name 3: Rahul K P **USN : ENG19CT0028**	
Name 4: S Adnan Mohamed **USN : ENG19CT0030**	
Name 5: Sumedh Hagaldivte **USN : ENG19CT0042**	

Place : Bangalore
Date :

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our Bachelor's degree in this institution.

We would like to thank ***Dr. Udaya Kumar Reddy K R, Dean, School of Engineering & Technology, Dayananda Sagar University*** *for his constant encouragement and expert advice.*

It is a matter of immense pleasure to express our sincere thanks to ***Dr. Girisha G S, Department Chairman***, ***Computer Science and Engineering***, ***Dayananda Sagar University,*** *for providing right academic guidance that made our task possible.*

We would like to thank our guide ***Prof. Baskar Venugopalan, Professor of Practice, Dept. of Computer Science and Technology***, ***Dayananda Sagar University***, *for sparing his valuable time to extend help in every step of our project work, which paved the way for smooth progress and fruitful culmination of the project.*

We would like to thank our ***Project Coordinator Dr. Meenakshi Malhotra*** *and* ***Dr. Pramod Kumar Naik*** *as well as all the staff members of Computer Science and Engineering for their support.*

We are also grateful to our family and friends who provided us with every requirement throughout the course.

We would like to thank one and all who directly or indirectly helped us in the Project work.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

GPS	Global Positioning System
IOS	Apple’s Operating System for iPhones
GSM	Global System for Mobile Communications
GIS	Geographic Information System
HTTP	Hyper Text Transfer Protocol
API	Application Programming Interface
SDK	Software Development Kit
CLI	Command Line Interface
CI/CD	Continuous Integration and Continuous Deployment
JDK	Java Development Kit
REST	Representational State Transfer
PSO	Particle Swarm Optimization
UI	User Interface

LIST OF FIGURES

ABSTRACT

Transportation is an essential part of our daily life in the world today. With an increase in the number of commuters to and from work, traffic congestion has become a major worry, particularly in a densely populated country such as India. This is a problem that ambulance drivers have to deal with. Despite the advancements in technology, there is no way to guarantee a quicker response time for emergency services to reach the scene of an accident.

Our solution involves the use of real-time traffic data and advanced algorithms to calculate the fastest route to the accident site. This data is collected from various sources such as traffic APIs. The data is then analyzed to determine the best route for the ambulance driver to take. The route is then displayed on a map, allowing the driver to easily navigate to the destination. Additionally, the system can also provide alternate routes in case of traffic congestion or other obstacles.

Furthermore, the system can also be integrated with other emergency services such as police and fire departments. This would allow for a more efficient response to an emergency situation. The system can also be used to alert other drivers of the ambulance's presence, allowing them to move out of the way and make way for the ambulance.

CHAPTER 1

INTRODUCTION

CHAPTER 1 INTRODUCTION

1.1 BACKGROUND

In today's world, transportation facilities are a vital requirement in our daily lives. With an increase in the number of commuters to and from work, traffic congestion has become a prime matter of concern, especially in a highly populous country like India. Our focus lies on the fact that emergency services such as ambulances and fire services must rely on the same traffic-infested roads to reach their respective destinations, regardless of being on the clock, depending on the gravity of the situation.

Most governments have strict traffic rules to be followed to facilitate the smooth functioning of emergency services. However, the present methods require some sort of manual human intervention by traffic authorities to ensure a faster reach to the site of the accident. Hence, we aim to come up with a solution to automate the task of path clearance for emergency vehicles with little to no human intervention, preferably no human intervention at all.

In many situations, although the traffic signal is green, the ambulance is stuck in the traffic because the vehicles ahead aren’t aware of the ambulance’s arrival until they hear the ambulance siren and are confused in which direction to move causing a huge ruckus. In order to solve this issue, we came up with a solution where once the ambulance has picked up the patient, the ambulance’s location will be notified to all the users, who are within the particular threshold value. The users will be provided with an app where they can see the path that is chosen by the ambulance. This will help the users to make way for the ambulance prior to the ambulance's arrival. By providing the path chosen by the ambulance the user can either give way by moving to the left or can choose another path of commute. This will prevent the ambulance from getting stuck in the traffic although the traffic signal is green.

To ensure that the ambulance driver and the other users reach their desired destination as soon as possible and in a seamless manner, they will be provided with an option of layers where they can filter out the best route to commute to their required destination based on layers such as traffic congestion, road construction and the number of potholes present.

Our team came to a decision to develop a mobile app since:

- In today’s technology-dependent world, almost everyone owns a smartphone.
- It is quick, reliable, and easy to use. Also, since Android and IOS are the most used mobile operating systems with continuous support.

1.2 OBJECTIVE

- To develop an app that accurately plots a path from the ambulance driver to the patient's location.
- Provide different layers to emergency vehicle driver to choose a path based on various parameters such as traffic congestion, road conditions, distance.

1.3 SCOPE

The project's scope will be confined to displaying the ambulance's fastest route to ambulance drivers. The deliverables include an app that tracks ambulance locations and aids the ambulance by showing it the quickest route based on the customized algorithm and also ensures that the ambulance reaches the patients location coordinate accurately leveraging what3words grid.

In order to check the patient's location being selected accurately, the experimental analysis was done on a sample size that was small. A larger sample size would increase the generalizability of our findings. Secondly, our study was conducted in a controlled environment and did not account for factors such as deviation in route due to weather conditions.

CHAPTER 2

PROBLEM DEFINITION

CHAPTER 2 PROBLEM DEFINITION

In India, traffic management is something that is often looked over. Ambulances and other vehicles must use the same road to reach their respective destinations. Due to traffic, it might lead to circumstances when other vehicles cannot give way to an incoming ambulance, even after hearing the siren from a few hundred meters away. This results in the ambulance becoming a part of the congestion with no alternative route to take, which could result in a delay that might be fatal for the patient in the ambulance. Without manual intervention, traffic congestion cannot be cleared. If this problem were to be solved, it would become a boon to society.

CHAPTER 3

LITERATURE REVIEW

CHAPTER 3 LITERATURE REVIEW

This section discusses the existing implementations of concepts similar to our project. Dr. Shashikala et al **[1]** used ESP 8266 along with Neo 6M GPS to alert the ambulance of the traffic. They also used ultrasonic sensors to detect traffic in a lane. Traffic congestion within a radius of 2 kilometers of the ambulance is detected. Using a combination of all of these technologies, a significant amount of time is saved.

Shantanu Sarkar **[2]** has used various technologies such as Arduino, Maps, GPS, CoolTerm, and GSM. The location of the ambulance is shared on Google Maps based on which closest hospital redirects the ambulance to the nearest path. They try to minimize the casualty rate by making sure the ambulance reaches the hospital as soon as possible.

Ms. M Ramya et al **[3]** make use of a combination of GPS and GIS tracking to check the availability of ambulances. The patient can reach the hospital at the earliest with minimal rates of casualty by leveraging these technologies.

Ramasami et al **[4]** make use of an automated sensor that sends alerts to the closest hospital and the ambulance and uses GPS to keep track of the location, along with the use of VANET, and NS tools. By leveraging VANET, the traffic at a particular range is found.

Dr. S. Senthilkumar et al **[5]** combine the use of an Android user app, HTTP server, and GPS-based tracking. Along with determining the location of the user, the ambulance in closer proximity is identified.

K Agarwal, M K Nigam, S Bhattacharya, and Sumathi G **[6]** implement neural networks, image processing, vehicle detection, geocoding, and tracking. Using a combination of sensors and geocoding we can trace the shortest path for the ambulance to reach a nearby hospital.

Runmin Wang, Zhigang Xu, Xiangmo Zhao, and Jinchao Hu **[7]** implement vehicle-to-vehicle communication to detect traffic congestion. Comparisons are made to CoTEC and geomagnetic coil for the efficiency of this method.

G. Lakshminarasimhan et al **[8]** use infrared and sound sensors to keep track of the density of traffic in a particular area. If sound crosses a limit, emergency cases are detected and the signal is automatically altered.
Rajeshwari S, Santosh Hebbar, and Varaprasad Goalla **[9]** use radio frequency identification to identify the number of vehicles passing through a particular area. This in turn helps identify network congestion and involves communication with the police control room and traffic controller to turn the signal green.

Tejas Naik et al **[10]** use RFID to manage and regulate traffic signals when an emergency vehicle approaches. An Arduino and led displays are used for demonstrating real-time traffic.

Prof Deepali Ahir et al **[11]** use an android app that registers the ambulance on its network. When a halt happens, a signal is sent to the traffic signal server along with the direction and current position using GPS.

Néstor Cárdenas-Benítez et al **[12]** have a system where they identify traffic congestion and change routes accordingly. The system monitors a large amount of data based on a Big Data cluster in Cassandra used together with OMNeT++, SUMO, and Veins vehicular network.

Satbir Singh et al **[13]** have proposed a framework to continuously update vehicle count and generate an alarm when the number of vehicles is large. Image processing techniques are applied along with IoT.

Madhav Mishra et al **[14]** leverage IoT to in advance alert an aforementioned problem before the ambulance reaches the traffic signal. This will help patients reach on time and provide doctors with increased time to treat the patient.

S. Pradeep Kumar et al **[15]** use a Micro Electro Mechanical System (MEMS), heartbeat, and various other sensors connected wirelessly. These vital parameters of the patient are monitored by an app on an android/IOS phone connected through Bluetooth. In case of an emergency, Mobile GPS is automatically triggered and sent to the server.

[18] Accuracy has always been a top priority when sharing a location, even with basic grocery delivery apps that use mapping features. Traditionally, latitude and longitude coordinates are used to pinpoint a location. The What3word approach also intends to provide users with a more common vernacular to refer to spatial coordinates.

[23] The delivery person often fails to reach the precise location of the customer and instead ends up in the adjacent street or another nearby location close to the destination. In India, 80% of the addresses are written with respect to a landmark which typically lies between 50-1500 meters of the actual address; such addresses make geolocating very challenging. Accuracy in geolocation is critical for emergency services to navigate quickly to reach you

[24] What3words is an Alternative Addressing Scheme (AAS). The What3Words system follows a Grid referencing technique and divides the ground in a 3-meter by 3-meter grid and assigns it a three-dictionary word separated by a dot.

[22] It is based on an algorithm available across devices and smartphones to provide precise, individual identification of any location across the world with the three words that function as an address. The three word code makes it easy to remember a location (Wired, 2017).

[21] This paper focuses on the core concept of What3words to highlight the fact that accuracy is important in emergency situations (Fire incidents, Hospital, Police) and how exactly What3words can be beneficial for addressing emergency situations. Similar paths are taken by the services what3emojis and what3fucks which use emoji's and swearwords to encode a triangular geohash. In the lat-long coordinate method, an incorrectly entered digit could result in a drastically inaccurate and undesired location.

[18] To highlight the simplicity of the What3words approach, a meeting point at the Bethesda Fountain in Central Park could be referred to as either 40.774489, -73.970871 in lat-long co-ordinates or more simply as vocab.neck.stem

[20] To address a specific location, there are several alternative approaches that may be used. What3words has a very high degree of accuracy and a low error rate, according to the results of an experiment done to compare the error rates and accuracy of various

CHAPTER 4

PROJECT DESCRIPTION

CHAPTER 4 PROJECT DESCRIPTION

4.1 MOBILE APP:

We developed a mobile application that makes use of a map overlay provided by HERE platform. The application is equipped with a custom routing algorithm designed for emergency service vehicles. The app makes use of What3words API to show the what3words grid overlay over the map which will help us choose a marker that represents the patient's location. The app provides a set of layers that can be laid on the map such as a layer that displays the amount of traffic in a particular path or the delay that could be caused due to road conditions.

4.2 ROUTING ALGORITHM:

The routing algorithm that we have implemented is a custom algorithm that is developed by us. The generic name of the algorithm is A star algorithm. It is a modified variant of Dijkstra's algorithm. Various factors like traffic, shortest time of travel are considered while calculating the route. The algorithm uses open street maps to obtain graph data. The graph data provided by open street maps is a graph of an area where each node of the graph represents a junction between roads and every edge of the graph represents a road. This graph is ideal for a routing algorithm. We have implemented A star algorithm. This algorithm conduction of the experiment requires a routing algorithm to calculate the routes. The routing algorithm employed to conduct this experiment is a proprietary algorithm provided by Here Platform as an integration with Here Maps. This algorithm is considered to be fair and accurate as far as the results are concerned. The purpose of choosing this algorithm is to provide a consistent method of calculating the route between two given points.

The application is designed to calculate routes in two different ways. Both ways take the patient's/user's location into consideration. The first way will use the inherent current patient/user location to calculate a route. The second way will display the What3words grid around the patient's/user's current location. This will give the patient/user a choice to select the correct square, thereby making the location selected accurate.

4.3 FUNCTIONALITY:

The application is provided with a map overlay and a search bar to provide simple usability. The application provides two roles, an ambulance driver and a regular user. The regular user is provided with an option to book an ambulance. Upon booking the ambulance, the driver will get a request notification within the app. Upon accepting the request, the driver will be provided with a custom calculated route. This same route is shared to the user who booked the ambulance. As an additional aid for accuracy, the application makes use of What3words to provide a 3x3 grid on which the user can tap to select the destination.

4.4 SYSTEM DESIGN:

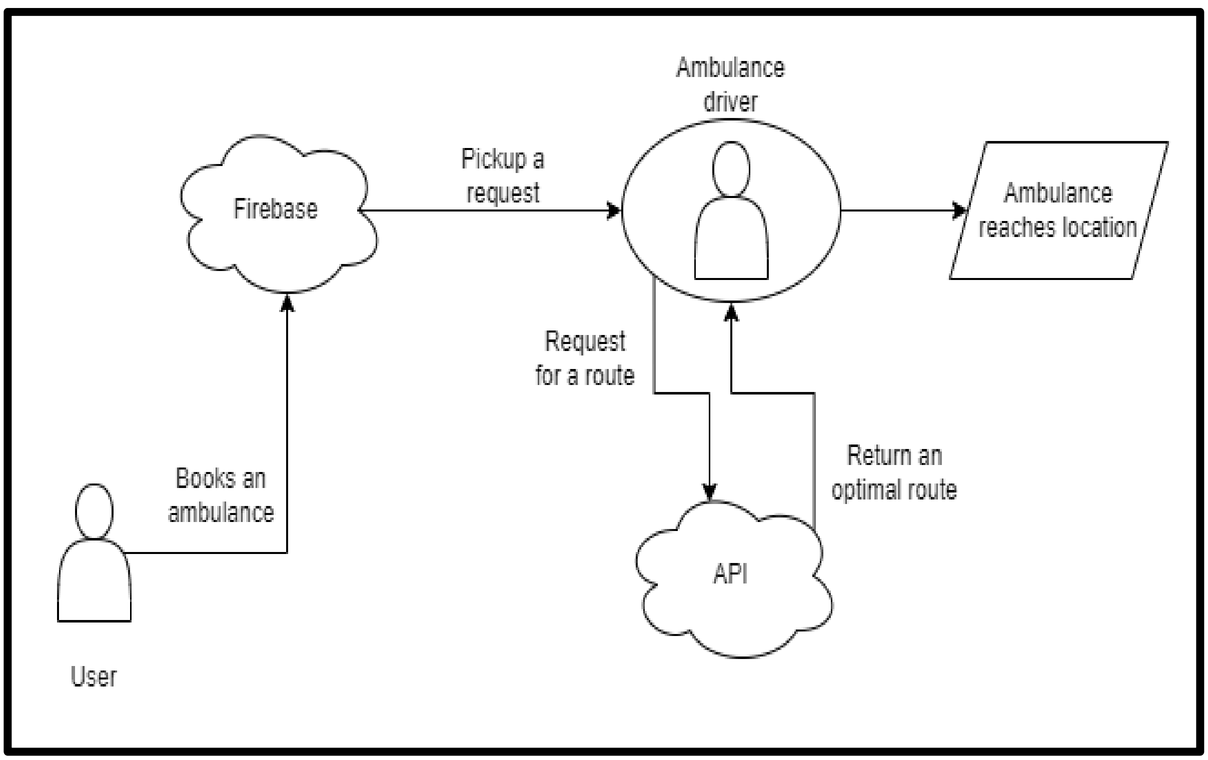


Figure 4.4(a): Basic app structure diagram

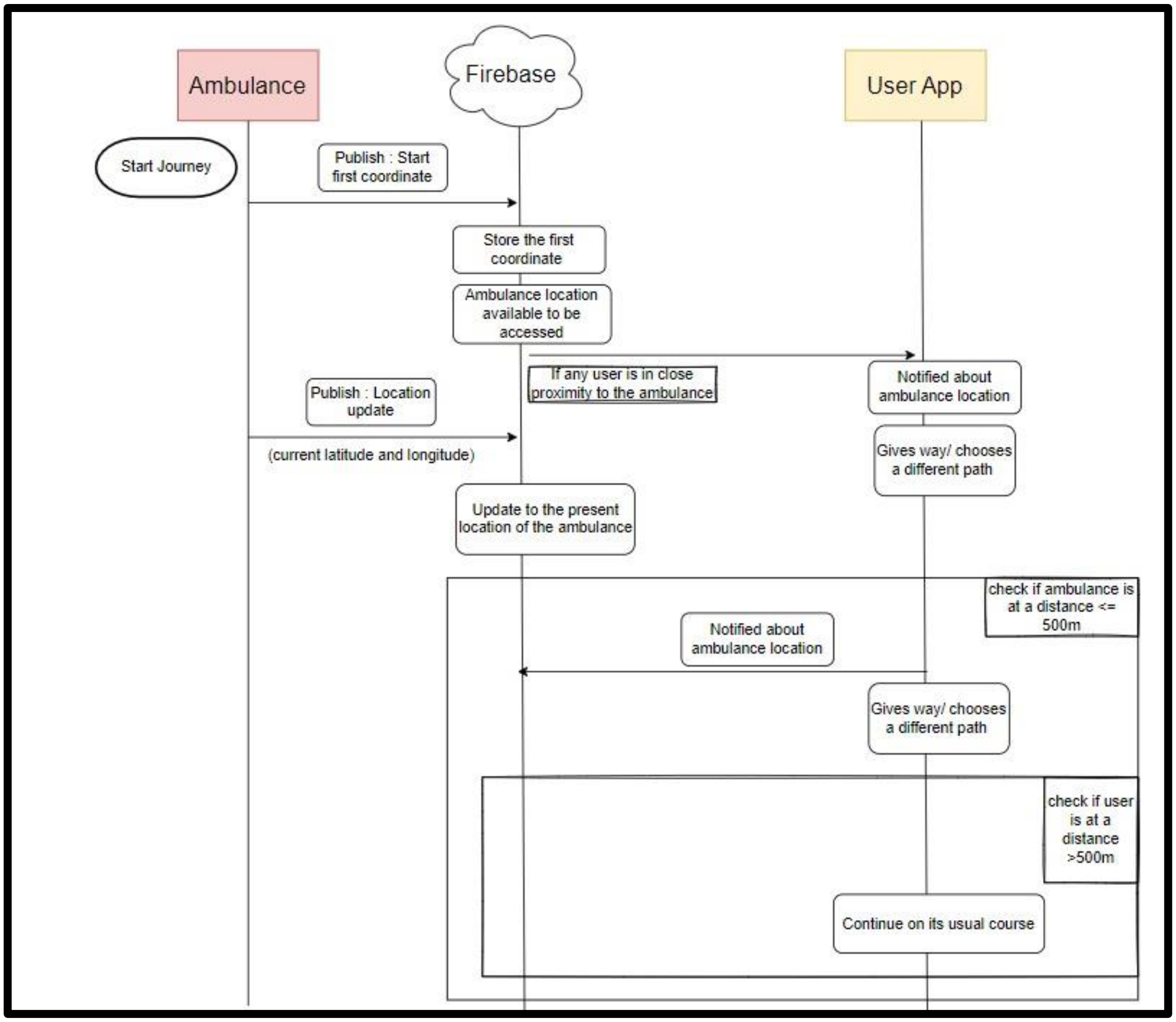


Figure 4.4(b): Product Architecture

4.5. ASSUMPTIONS AND DEPENDENCIES:

The section should discuss about the dependencies and the assumptions that the system is taking

Route availability – Even in a dire situation, we can plan the route instantaneously. In terms of development, the objective is to keep the time complexity of route calculation as low as possible.

Compatibility with different devices- Flutter is platform-independent and one of the key benefits of Flutter is its high performance. Flutter apps are compiled natively, which means they can achieve near-native performance on both Android and iOS devices.

User input –It is taken from the ambulance booking form, and details such as Patient Name, Age, Gender, and preferred hospital are collected.

Location is collected using what3words. The user taps on a square to select his location.

Dependencies:

Flutter dependencies-

- Here SDK - for maps overlay
- Shared preferences - for storing login credentials
- Dotenv - for accessing sensitive information in repo
- Firebase
- Flutter fire - to configure the project with firebase

Other dependencies-

- Firebase CLI - to connect project with firebase
- Firebase Realtime Database - to store information regarding route, location, etc.
- Microsoft Azure-Cloud resources - runs on cloud instance
- CICD pipeline - to build APKs for testing
- Node JS server - to store built APKs for later use
- API for routing algorithm
- Python Flask server to handle API requests
- Python script to calculate route

Python libraries-

- geopy
- numpy
- networkx
- osmnx
- scikit-learn

CHAPTER 5

REQUIREMENTS

CHAPTER 5 REQUIREMENTS

5.1 FUNCTIONAL REQUIREMENTS

The solution to the problem statement is a **Flutter Application**. Preliminary steps in building the application:

- A map Software layer must be decided in order to show maps in the application.
- There must be a method of transporting data between the instances of the application running on multiple devices. (server / serverless design).
- There are two types of users, a regular driver and an ambulance driver, this idea must be clear when building the application.
- A routing algorithm needs to be implemented to provide the fastest route for the emergency vehicles.

Regarding the above preliminary steps, several Map providers / SDKs have been tested, (Here Maps, Bing Maps, TomTom, OSMAND, Open Street Maps, and Mappls). The chosen SDK is **Here Maps Flutter SDK.** Data transport between instances is through **Firebase Realtime Database**.

Several routing algorithms have been tested, namely, Genetic, Particle swarm and A* algorithm. Out of these we chose A star algorithm. Details of the algorithm that we have implemented can be found at the end of this document.

5.2 NON-FUNCTIONAL REQUIREMENTS

- **Usability**: The application shall be user-friendly and doesn't require any guidance. In other words, the application will be easy to learn and use.
- **Reliability**: The application should not have any unexpected failure. In order to avoid any failure occurrence, the specifications have been respected and followed correctly. Any component can be modified to correct faults, improve performance or other attributes, or adapt to a changing environment, because of the development methodology followed.
- **Performance** - Monitor/Test location coordinate data transmission latency.
- **Correctness of Output** - Maintainability is ensured by timely monitoring of correct working of the system.

5.3 HARDWARE AND SOFTWARE REQUIREMENTS

5.3.1 Software Requirements:

- Operating System: Windows/Linux
- Flutter
- JDK
- Ci/Cd Tools - Microsoft Azure, GitHub Actions
- Text Editor-VS Code
- Python virtual environment (v3.8+)
- Docker

5.3.2 Hardware Requirements:

- A Desktop/Laptop with Stable Internet Connection
- Android /iOS mobile phone.

5.3.3 CI/CD Pipeline:

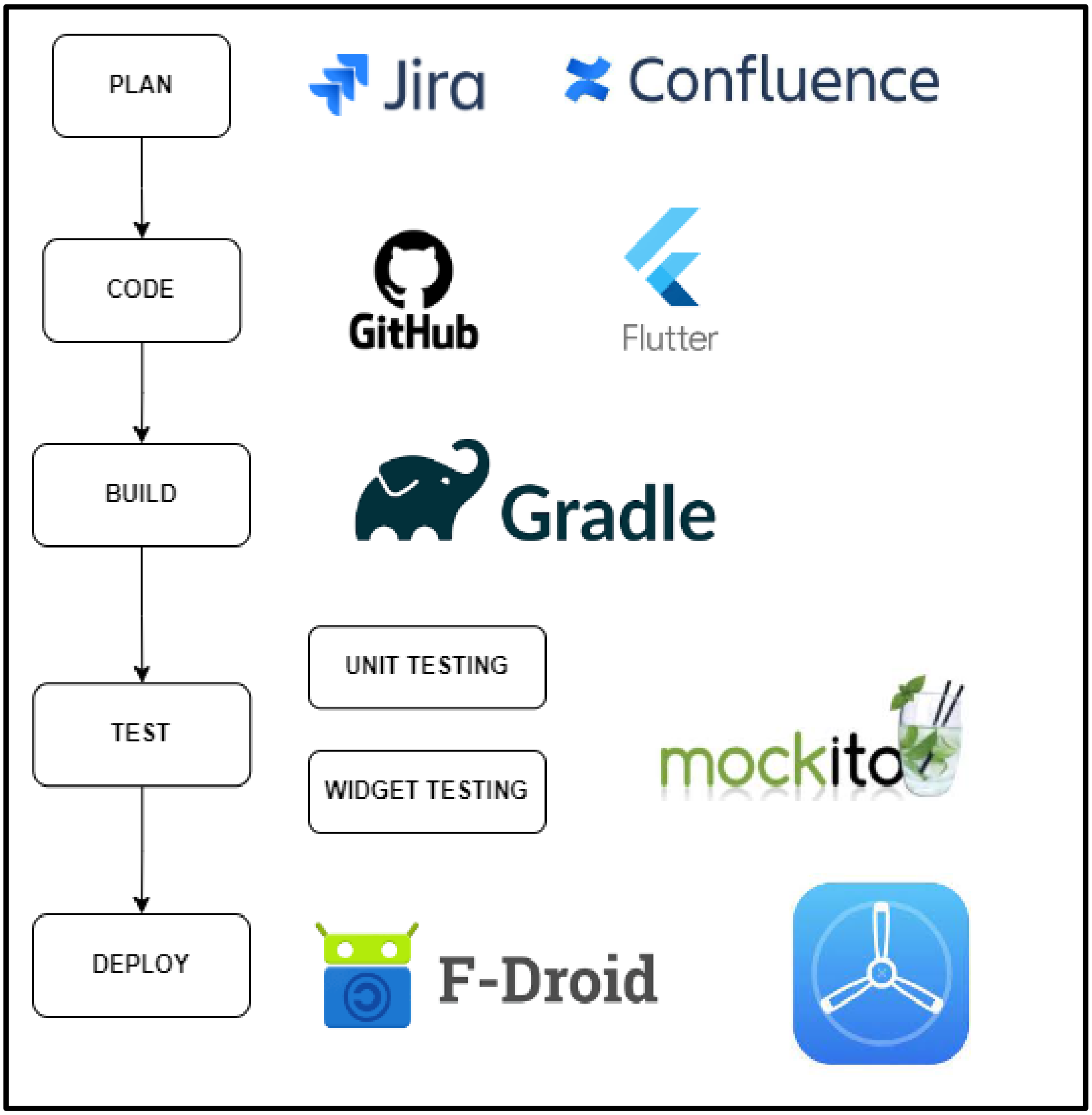


Figure 5.3.3: CI/CD Pipeline

CHAPTER 6

METHODOLOGY

CHAPTER 6 METHODOLOGY

Building the application requires an understanding of how typically a map application works and how data is transferred through the database.

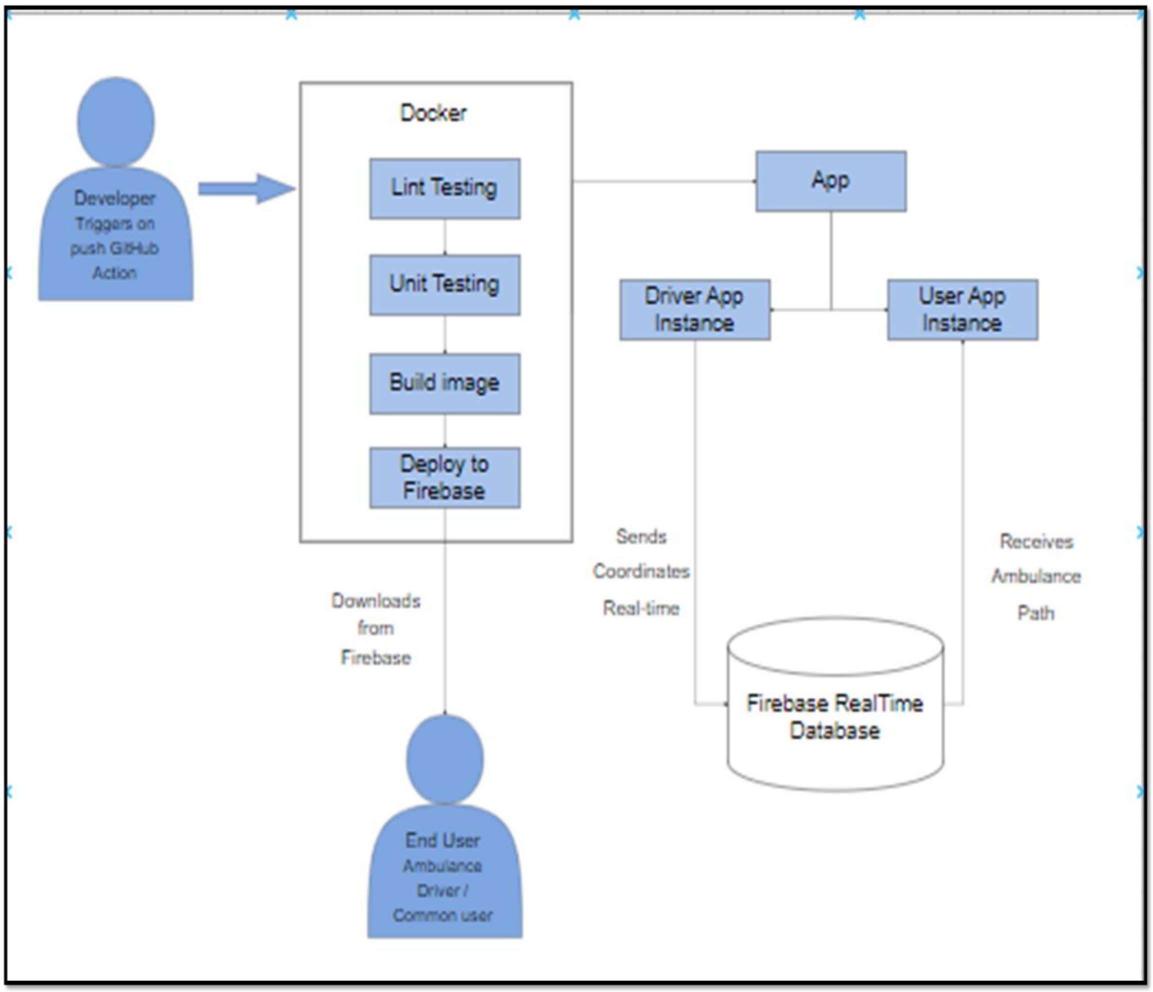


Figure 6.4: Component Diagram

6.1 BUILDING THE BASIC APP STRUCTURE

Development starts by building the basic skeleton app. The app will first request location permission from the user, After the said permission is granted by the user, the app will display an authentication page which will be used to authenticate the user and determine the type of user.

6.2 MAP UI

This part of the app is concerned with displaying the Map itself to the user along with other details. To access the map, the **Here Maps Explore Edition SDK** is used. The SDK requires REST API/ MAP API key, Client ID, and Client secret. All of these keys/secrets can be obtained from the **Here Maps** dashboard.

Once the user is authenticated, they will be shown the map centered on their current location. The map will be loaded using the SDK functions. The SDK is first initialized with the API keys.

Then the actual map view is rendered. The map view will have a transparent view layer on top of it which will contain buttons that will interact with the map. The user’s location button in this layer will center the map to the user’s current location. There is also a search bar to search for a particular location to place a marker on the map.

The application also has a feature to book an ambulance. There is a simple form provided. Upon submission of the form, the user will be prompted to choose their current location in terms of a What3words square.

6.3 WORKING SCENARIO

The app has two personas/perspectives, user and ambulance driver. For the patient, the app will show the ambulance's current location and the route that it is taking. This will give a choice to the user to choose an alternate route.

The app will calculate the fastest route for the ambulance driver to ensure speedy delivery of the patient to the hospital. The app will request for an optimal route through an API call to our server which will compute the optimal route possible in that situation.

6.4 CONCEPT OF LAYERS

The app makes use of What3words API to show the what3words grid overlay over the map which will help us choose a marker that represents the patient's location. The application makes use of RESTful API services provided by What3words [4] to display the grid overlay. The app also makes REST API calls to obtain 3-word-coordinate addresses from Geo Coordinates (Latitude and Longitude). The Latitude and Longitude are obtained when the user taps on the map and from the user's current location. Apart from the what3words grid layer, the app displays traffic congestion layer, road condition layer on the map.

CHAPTER 7

EXPERIMENTATION

CHAPTER 7 EXPERIMENTATION

7.1. WHAT3WORDS

We have experimented on the following algorithms for calculation of route. We have determined that A* algorithm is best suited for path calculation.

- For each route in the experiment, the source geo-coordinates are fixed at the beginning. Since it is hard-coded, there is no scope for fluctuation in location data, and hence, any irregularities that arise, are from the client side (the device used to book the emergency service).
- The experiment is done in two phases; one phase makes use of the client device's GPS as the destination which can be inconsistent.
- The other phase makes use of what3words which allows the user to select a square. This square will serve as the destination.

As a part of the procedure, a single route is plotted multiples times using GPS to detect the user's location. Here, difference refers to the difference in distance between a path computed using traditional GPS and What3Words. In the first phase of the experiment, the destination of each route is subject to fluctuation due to the inaccuracies of the GPS system, resulting in different distances being calculated for each iteration.

A negative value indicates that the calculated route is shorter compared to the actual route. On the other hand, a positive reading means that the calculated route is longer compared to the exact route. Both the above cases correspond to the scenario where the ambulance reaches an adjacent or nearby street in place of the exact location.

For the second phase, the source is manually chosen by the user, selecting a 3m x 3m square. This greatly improves the accuracy of the route, and the same route is calculated for each iteration.

7.2 ROUTING ALGORITHMS

Out of all routing algorithms available, we have shortlisted three of them, namely A*, Genetic and Particle Swarm. We will be evaluating their performance and select the most appropriate one for our specific use case. Additionally, we will be incorporating constraints such as traffic density, road condition, etc. into the chosen algorithm to find the most optimal path.

7.2.1. A* Algorithm

A* algorithm is a graph traversal and path searching algorithm. It is used in many fields of computer science due to its completeness, optimality and optimal efficiency. It finds the shortest path between a source and a destination. It is a handy algorithm that is often used for map traversal to find the shortest path to be taken. It searches for shorter paths first, thus making it an optimal and complete algorithm. An optimal algorithm will find the least cost outcome for a problem, while a complete algorithm finds all the possible outcomes of a problem. Another aspect that makes A* so powerful is the use of weighted graphs in its implementation. A weighted graph uses numbers to represent the cost of taking each path or course of action. This means that the algorithms can take the path with the least cost, and find the best route in terms of distance and time.

Working: At each node, it picks up a value ***f*** which is a parameter equal to the sum of two other parameters: ***g*** and ***h***. At each step it picks the node with the lowest value for ***f***.

g = movement cost to move from the source node to the given node following the path to get here

h = estimated movement cost to move from the given node to the destination

h is often referred to as heuristic, which is also called a smart guess.

Algorithm:

Create Start Node with Current Position

Add Start Node to Queue

    While Queue Not Empty

    Sort Node Queue by f(N) in Ascending

    Get First Node from Queue Call Node "N"

    if N is Goal Then Found and Exit Loop

    Else

Mark N Node as Visited

Expand each reachable Node from N Call Node "Next N"

f (Next N) = g (Next N) + h (Next N)

Loop

A* algorithm is the best path finding algorithm, but doesn't find the shortest path always, heavily relies on the heuristics/approximation to calculate ***h***. One major practical drawback of A* is its space complexity, as it stores all generated nodes in memory. Heuristic can differ depending on the use case, can be Euclidean, Manhattan, Haversian etc., and time complexity depends on the heuristic function. In our use case, the heuristic which we will be using is haversine distance.

7.2.2. Genetic Algorithm

Genetic Algorithm is a metaheuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms. It is commonly used to generate high quality solutions to optimization and search problems by relying on biologically inspired operators such as mutation, crossover and selection.

Working: Mutation is the process of introducing new genetic information into the population by randomly changing the value of a gene in an individual. Mutation can help prevent the algorithm from getting stuck in local optima and can introduce diversity into the population. Crossover is the process of creating new individuals by combining the genes of two parents. The goal of crossover is to create new individuals that inherit some of the traits of both parents. Selection is the process of choosing individuals from the population to be parents for the next generation. The goal of selection is to preferentially choose individuals with higher fitness scores.

Algorithm:

- Initialize a population of individuals randomly
- Evaluate the fitness of each individual using a fitness function
- For a fixed number of generations, repeat:
- Select the best individual from the previous generation and add it to the new population

- Create new individuals by combining the genes of two parents (crossover)
- Apply mutation to some individuals to introduce new genetic information into the population.
- Evaluate the fitness of each individual in the new population.
- Replace the old population with the new population.
- Return the best individual from the final population.

The algorithm stops when a certain condition is met, such as reaching a maximum number of generations, achieving a satisfactory fitness score, or reaching a computational time limit. There are a few limitations of the Genetic Algorithm compared to other path optimization algorithms which are as follows:

Repeated fitness function evaluation for complex problems is often the most prohibitive and limiting segment of artificial evolutionary algorithms.

Genetic algorithms do not scale well with complexity.

Operating on dynamic data sets is difficult, as genomes begin to converge early on towards solutions which may no longer be valid for later data.

Overall, the genetic algorithm is a flexible and powerful optimization method that can be used in a wide range of problems. The choice of representation, fitness function, selection, crossover, and mutation methods can have a significant impact on the performance of the algorithm.

7.2.3. Particle Swarm Optimization Algorithm

Particle swarm optimization (PSO) is one of the bio-inspired algorithms and it is a simple one to search for an optimal solution in the solution space. It is different from other optimization algorithms in such a way that only the objective function is needed and it is not dependent on the gradient or any differential form of the objective. It also has very few hyper parameters.

Working: Particles represent potential solutions to the problem being solved. Each particle is characterized by its position and velocity. The fitness function is used to evaluate the quality of a particle's position. The swarm is a collection of particles that move and interact with each other to find the optimal solution. The swarm size can affect the exploration and exploitation of the solution space. Each particle updates its position based on its current velocity and the best position it has experienced so far, as well as the best position the swarm has experienced so far. This update is used to move the particles towards better solutions. Each particle maintains its personal best position and the swarm maintains the global best position. These positions are used to guide the search towards better solutions.

Algorithm:

- Initialize a swarm of particles with random positions and velocities.
- Evaluate the fitness of each particle using the fitness function.
- For each particle, update its personal best position and fitness.
- Identify the particle with the best fitness as the global best position.
- For each particle, update its velocity and position:
    - $x^i(t+1) = x^i(t) + v_x^i(t+1)$
    - $y^i(t+1) = y^i(t) + v_y^i(t+1)$

    where $x^i(t)$ is the position of particle i at iteration t
- Repeat the above steps until a satisfactory fitness core is reached.

Overall, we can say that the Particle Swarm Optimization Algorithm is a good choice. A few advantages include insensitivity to scaling of design variables, can be easily parallelized for concurrent processing, and makes use of fewer parameters compared to other algorithms. The only drawback of PSO is the weak optimum local search-ability.

CHAPTER 8

TESTING AND RESULTS

CHAPTER 8 TESTING AND RESULTS

8.1 RESULTS

8.1.1. Project screenshots

Figure 8.1(a): Firebase Realtime Database

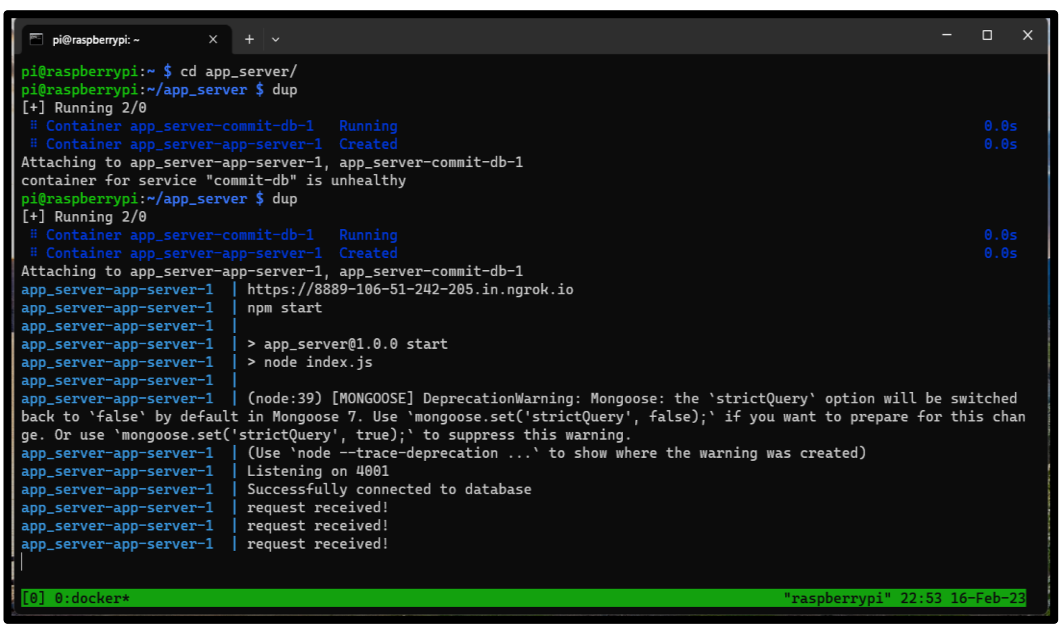


Figure 8.1(b):CI/CD server to maintain development releases

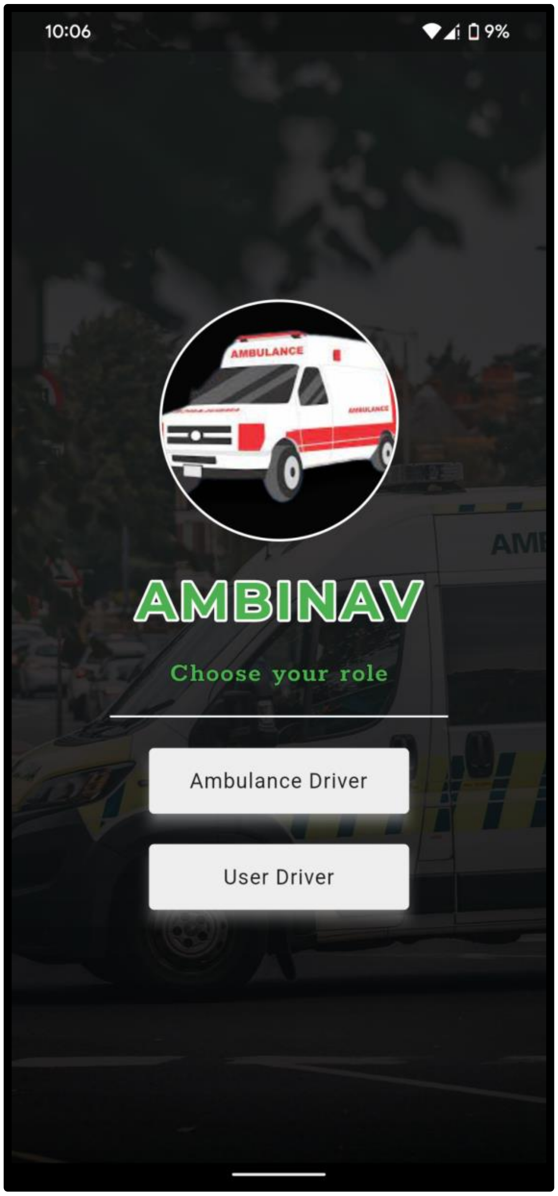


Figure 8.1(c): App screen before logging in

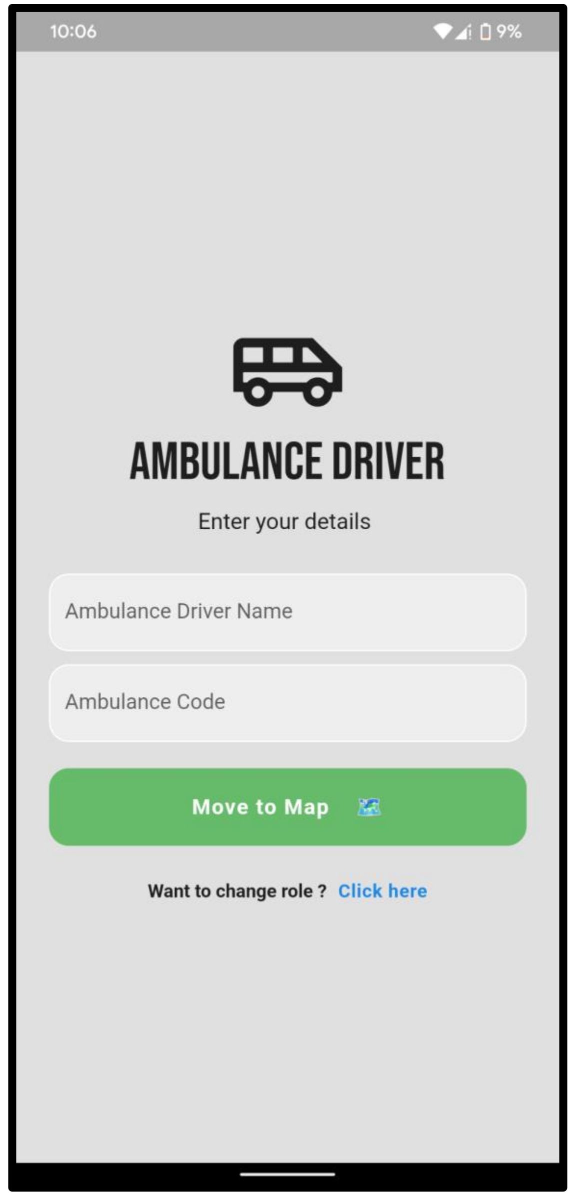


Figure 8.1(d): Ambulance driver login page

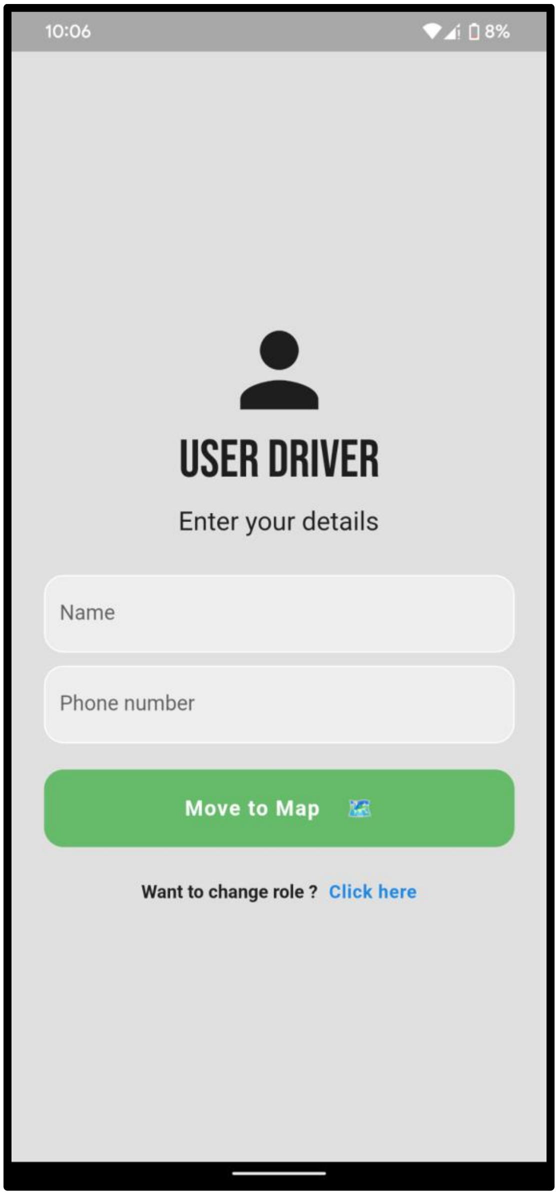


Figure 8.1(e): User driver login page

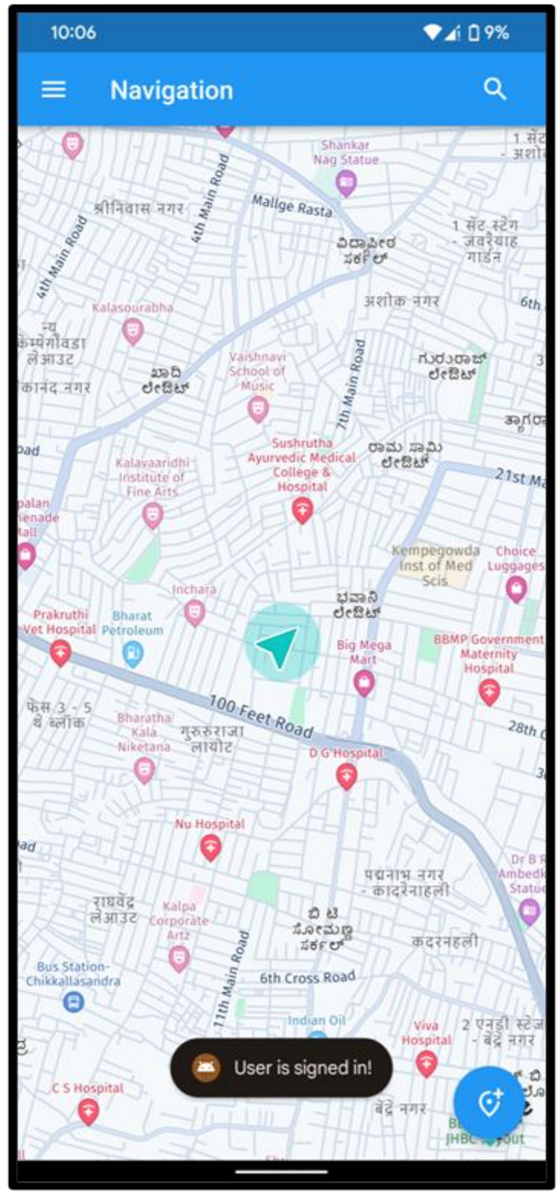
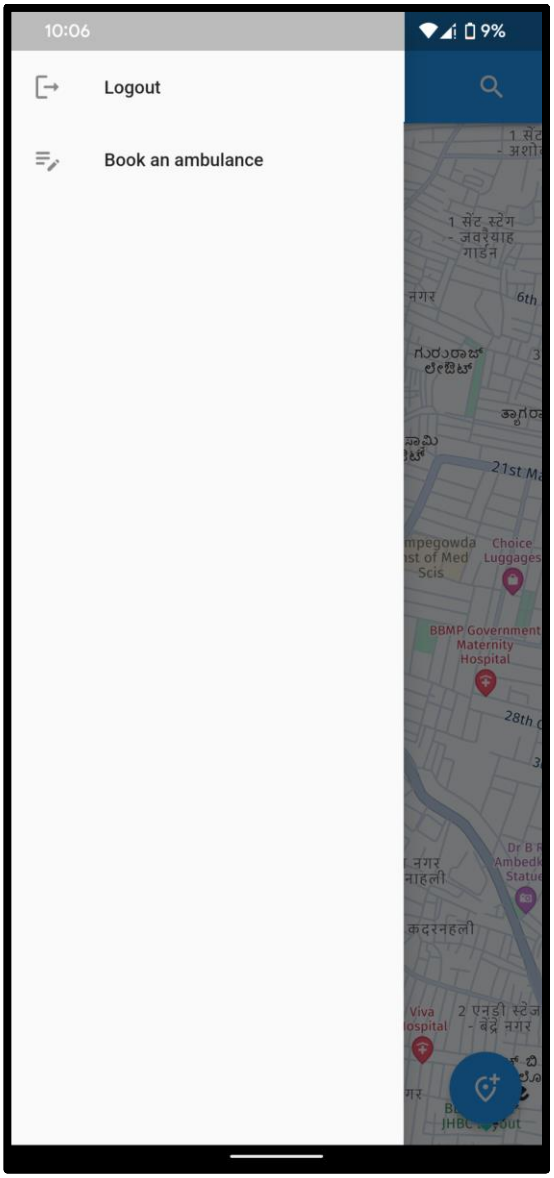


Figure 8.1(f): App screen after logging in



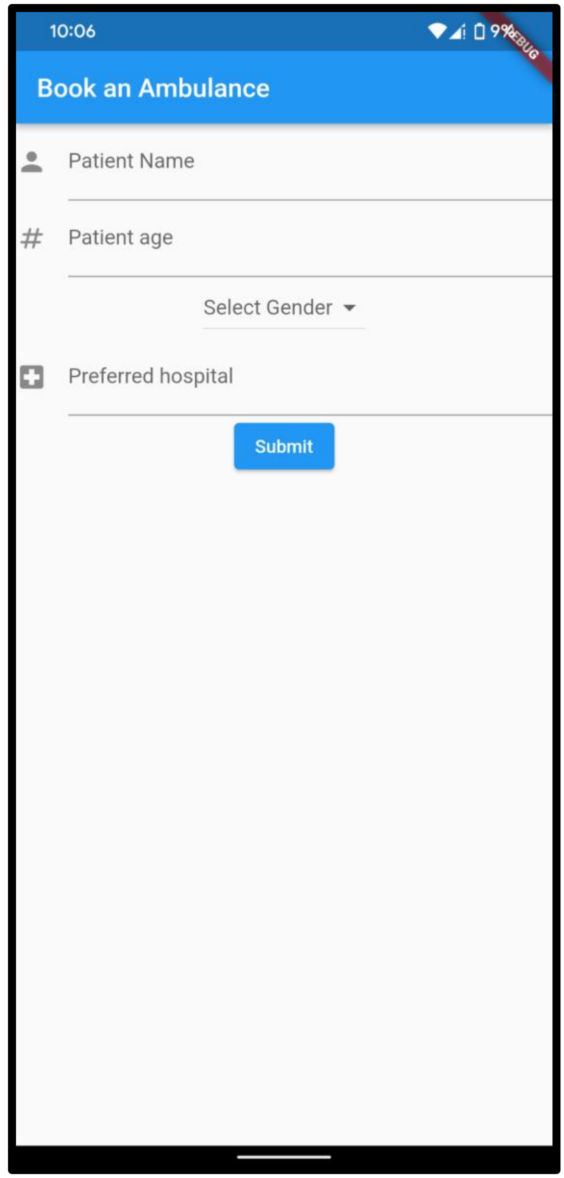
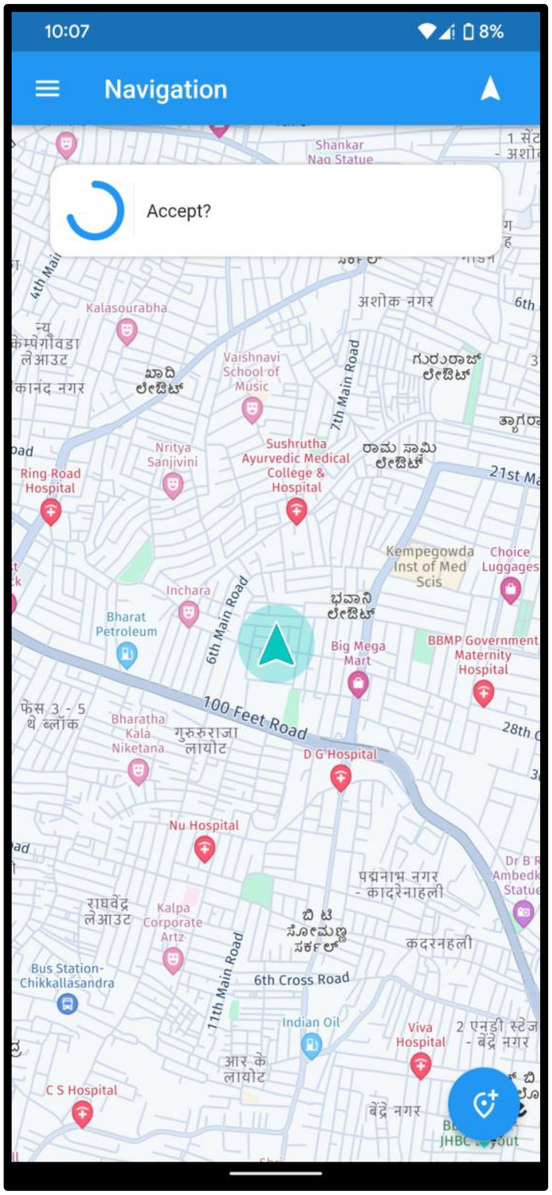


Figure 8.1(g): Drawer Screen

Figure 8.1(h): To book an ambulance



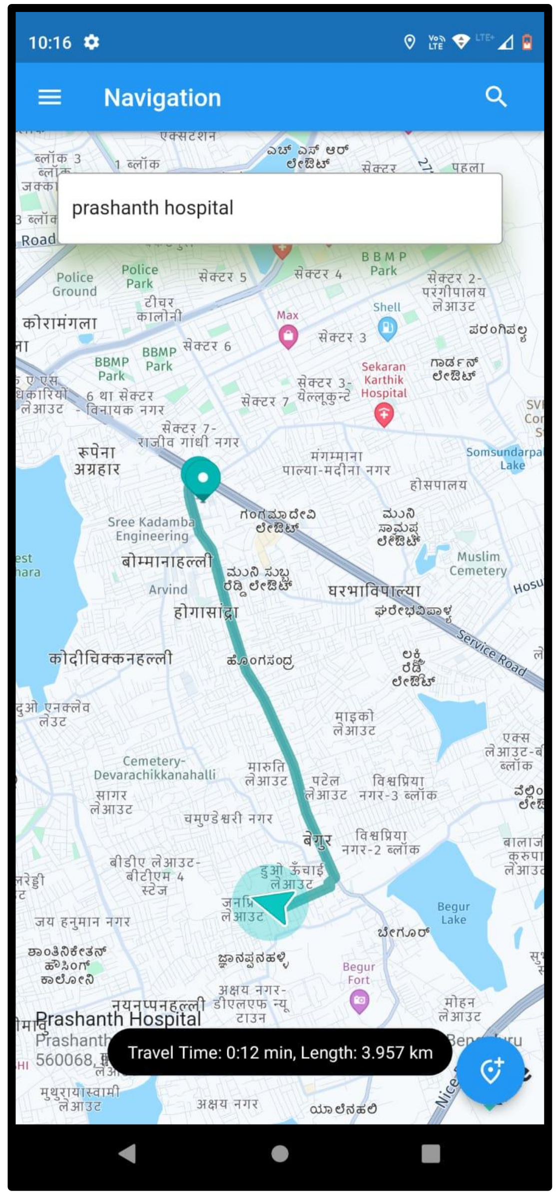


Figure 8.1(i): Display Route (User side)

Figure 8.1 (j): Request notification

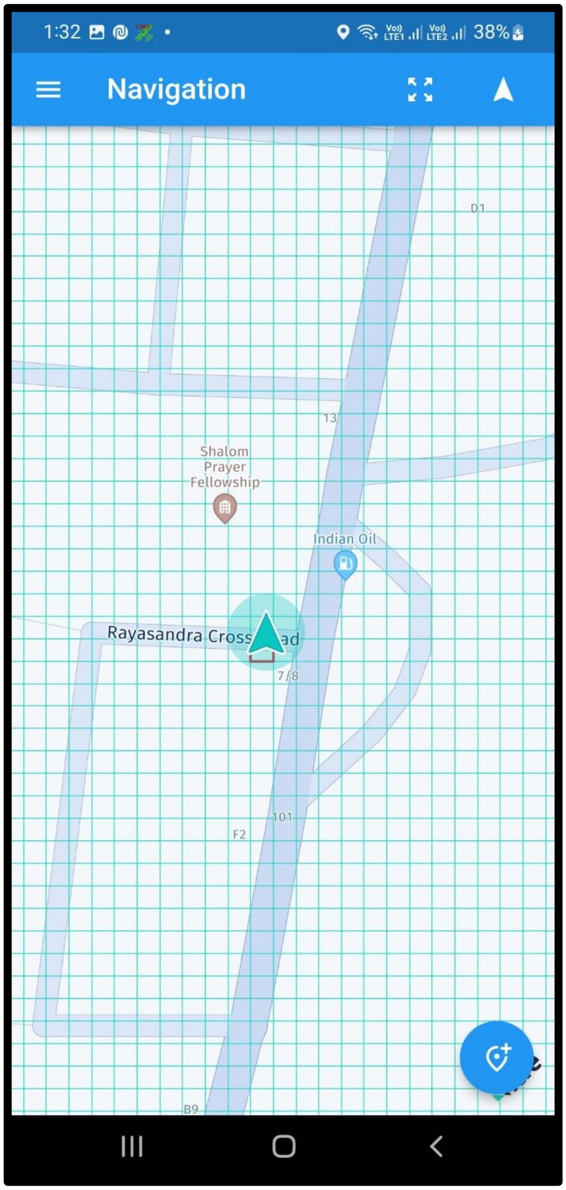


Figure 8.1(k): What3Words Grid Overlay

Figure 8.1(l): Route to hospital

8.2 DISCUSSION OF RESULTS

What3words:

A negative value indicates that the calculated route is shorter compared to the actual route. On the other hand, a positive entry means that the calculated route is longer compared to the exact route.

Both the above cases correspond to the scenario where the ambulance reaches an adjacent or nearby street in place of the exact location.

From the data analyzed (refer to section 7.1), In the first phase of the experiment, the destination of each route is subject to fluctuation due to the inaccuracies of the GPS system, resulting in different distances being calculated for each iteration. For the second phase, the source is manually chosen by the user, selecting a 3m x 3m square. This greatly improves the accuracy of the route, and the same route is calculated for each iteration.

Routing:

A combination of A star and Genetic algorithm for this project. The A star algorithm is used for fast route calculation. The genetic algorithm is used for tuning the A star algorithm the right way to obtain the optimal route.

CHAPTER 9

CONCLUSION AND FUTURE WORK

CHAPTER 9 CONCLUSION AND FUTURE WORK

9.1. CONCLUSION

The project delivers a mobile application that will aid emergency service vehicles to reach their destination / site of accident quickly. This is done by providing the ambulance driver with an optimal path to reach their destination (the patient). For future work, the route taken by the ambulance can be broadcasted to the traffic police to help control the signal in favor of the approaching ambulance. For further extensions, refer to annexure. The routing algorithm is designed to provide an optimal route based on several conditions like the current traffic conditions, distance, etc. The algorithm is designed to give priority to emergency vehicles. The what3words grid overlay will aid in choosing the correct source/destination when routing.

9.2. SCOPE FOR FUTURE-WORK

- Integrate it with smart traffic system
- Extend the service to fire emergencies
- Implement this concept using Machine to machine communication.
- Broadcast path to traffic authorities as an assistive measure in paving way for the ambulance / fire-truck.
- Providing users with a choice of routes based on constraints such as distance and traffic via a user-friendly UI (user interface)

CHAPTER 10

REFERENCES

CHAPTER 10 REFERENCES

[1] Roopashree V, Nikitha Bai E, Dr.Shashikala, Malavika D N, Suman A, "Traffic Congestion Detection and Alerting Ambulance using IoT" Vol.9 Issue 07, July 2020

[2] Shantanu Sarkar, "Ambulance Assistance for Emergency Services using GPS Navigation" IRJET Vol.3 Issue 09, Sep 2016 e-ISSN: 2395-0056J.

[3] Ms. M Ramya, Ms. Jesintha, Ms. R Janani, Ms. Kokila, Mr. R JayaSuriya, "Emergency Ambulance Tracking System using GPS Android Application", IJERT NCICCT 2022 ISSN: 2278-0181

[4] Ramasami, Gowri Shankar, Moulishankar, Sriramprasad, Sudharsan Narayanan,"Advanced Ambulance Emergency Services Using GPS Navigation" IJERT ETEDM 2018 Conference Proceedings ISSN: 2278-0181

[5] Dr.S. Senthilkumar, Dr.K. Sasikala, T. Geetha, V. Govindarajan, K. Jammiammal, "Ambulance Tracking System in Android Phone" Turkish Online Journal of Qualitative Inquiry (TOJQI), Volume 12, Issue 3, July 2021

[6] K Agarwal, M K Nigam, S Bhattacharya, Sumathi G, "Ambulance detection using image processing and neural networks", Journal of Physics: Conference Series

[7] Runmin Wang, Zhigang Xu, Xiangmo Zhao, Jinchao Hu: "V2V-based Method For The Detection Of Road Traffic Congestion", The Institution Of Engineering And Technology Journals, 2019, Vol 13 Issue 5

[8] G.Lakshminarasimhan,V. Parthipan, Mohammed Irfan Ahmed, Sri Harsha K Nvm, Dr.D. Dhanasekaran: "Traffic Density Detection And Signal Automation Using IoT", Saveetha School Of Engineering, Saveetha University, International Journal Of Pure And Applied Mathematics, 2017, Vol 116 No. 21

[9] Rajeshwari S., Santhoshs Hebbar, Varaprasad Golla: "Implementing Intelligent Traffic Control System for Congestion Control, Ambulance Clearance and Stolen Vehicle Detection", IEEE Sensors Journal, 2013, Vol 10

[10] Tejas Naik, Divya Ravi N., Pawdhan Jain: "RFID-based Smart Traffic Control Framework for Emergency Vehicles", Proceedings of The 2nd International Conference on Inventive Communication And Computational Technologies, 2018, Vol 12

[11] Prof. Deepali Ahir, Saurabh Bharade, Pradnya Botre, Sayali Nagano, Mihir Shah: “Intelligent Traffic Control System for Smart Ambulance”, International Research Journal of Engineering and Technology, June-2018, Vol 05, Issue 06

[12] Néstor Cárdenas-benítez, Raúl Aquino-santos, Pedro Magaña-espinoza, José Aguilarvelazco, Arthur Edwards-block, And Aldo Medina Cass: “Traffic Congestion Detection System Through Connected Vehicles And Big Data”, Sensors(basel, Switzerland), May 2016 , Vol 16(5)

[13] Satbir Singh, Baldev Singh, Ramandeep, Baljit Singh: “Automatic Vehicle Counting for IoT Based Smart Traffic Management System for Indian Urban Settings”, IEEE Sensors Journal, 2019, Vol 5

[14] Madhav Mishra, Seema Singh: “Advance Alert for Ambulance Pass by Using IoT for Smart City”, International Journal Of Engineering Science And Computing, June 2017, Volume 7 Issue No.6

[15] S Pradeep Kumar, D Akash, K Murali, R Shriram, “Call ambulance smart elderly monitoring system with nearest ambulance detection using Android and Bluetooth”, 2016 Second International Conference on Science Technology Engineering and Management (ICONSTEM)

[16] Pal, R., Ghosh, A., Kumar, R., Galwankar, S., Paul, S. K., Pal, S., Sinha, D.,Jaiswal,A. K.,MoscoteSalazar, L. R., Agrawal, A. (2019). Public health crisis of road traffic accidents in India: Risk factor assessment and recommendations on prevention on the behalf of the Academy of Family Physicians of India. Journal of family medicine and primary care, 8(3), 775–783.

[17] Jiang, W., Stefanakis, E., 2018. What3words Geocoding Extensions. Journal of Geovisualizationand Spatial Analysis, 2:7, 1-17.

[18] Wiebe, Kyle. ” Assessing the Utility & Feasibility of Geo-spatial Addressing in Mathare, Nairobi.” (2018).

[19] Jiang, W., and E. Stefanakis. ” A RESTFUL API FOR THE EXTENDED WHAT3WORDS ENCODING.” ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences 4, no. 4 (2018).

[20] Clemens, Konstantin. ” Comparative evaluation of alternative addressing schemes.” (2016): 118.

SAMPLE CODE

Main.dart

main.dart ×

lib > main.dart > ...

```dart
import 'package:AmbiNav/services.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'firebase_options.dart';
import 'package:flutter/material.dart';
import 'app_screen_ui.dart';
import 'package:here_sdk/core.engine.dart';
import 'package:here_sdk/core.dart';
import 'package:here_sdk/core.errors.dart'; //for handling InstantiationException while initializing sdk
import 'package:flutter_dotenv/flutter_dotenv.dart';
import 'starter.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'package:firebase_auth/firebase_auth.dart';

Future<void> _initializeHERESDK() async {
  // Needs to be called before accessing SDKOptions to load necessary libraries.
  SdkContext.init(IsolateOrigin.main);

  //loading the .env file
  await dotenv.load(fileName: "credentials.env");
  // Set your credentials for the HERE SDK.
  String accessKeyId = dotenv.env["here.access.key.id"]!;
  String accessKeySecret = dotenv.env["here.access.key.secret"]!;
  SDKOptions sdkOptions =
      SDKOptions.withAccessKeySecret(accessKeyId, accessKeySecret);

  try {
    await SDKNativeEngine.makeSharedInstance(sdkOptions);
  } on InstantiationException {
    throw Exception("Failed to initialize the HERE SDK.");
  }
}

void alreadyLoggedin() {
  SharedPreferences.getInstance().then((value) {
    bool newuser = (value.getBool('login') ?? true);

    if (newuser == false) {
      Services.usertype = value.getString('usertype')!;
      runApp(MaterialApp(
        debugShowCheckedModeBanner: false,
        home: AppScreen(),
      )); // MaterialApp
    } else {
      runApp(MaterialApp(
        debugShowCheckedModeBanner: false,
        home: loginpg(),
      )); // MaterialApp
    }
  });
}
```

```dart
void checkLoginStatus() {
  FirebaseAuth.instance.idTokenChanges().listen((User? user) {
    if (user == null) {
      Fluttertoast.showToast(msg: "User is currently signed out!");
    } else {
      Fluttertoast.showToast(msg: "User is signed in!");
    }
  });
}

Run | Debug | Profile
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Services.getPermissions(); // wait for permissions
  Services.setLoc(); // start streaming the location
  await _initializeHERESDK(); // initialise the HERE SDK
  alreadyLoggedin(); // check if user is already logged in
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );

  checkLoginStatus();
  try {
    final userCredential = await FirebaseAuth.instance.signInAnonymously();
    print("Signed in with temporary account.");
    Fluttertoast.showToast(msg: userCredential.user!.uid);
```

```dart
    Fluttertoast.showToast(msg: userCredential.user!.uid);
  } on FirebaseAuthException catch (e) {
    switch (e.code) {
      case "operation-not-allowed":
        print("Anonymous auth hasn't been enabled for this project.");
        break;
      default:
        print("Unknown error.");
    }
  }
  checkLoginStatus();
  _initializeHERESDK();
  alreadyLoggedin();
}
```

Ambulance form.dart

```dart
import 'dart:convert';
import 'package:AmbiNav/services.dart';
import 'package:firebase_database/firebase_database.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:crypto/crypto.dart';

// User defined ambulance form widget
class AmbulanceForm extends StatefulWidget {
  @override
  AmbulanceFormState createState() {
    return AmbulanceFormState();
  }

  String generateFormHash(String name, String age, String hospital) {
    var bytes = utf8.encode(name + age + hospital);
    var hash = sha256.convert(bytes);
    return hash.toString();
  }
}

// The form state class
class AmbulanceFormState extends State<AmbulanceForm> {
  // a global key to validate form and identify widget
  final _formKey = GlobalKey<FormState>();
  final appTitle = 'Book an Ambulance';
  TextEditingController patient_name = TextEditingController();
  TextEditingController age = TextEditingController();
  TextEditingController age = TextEditingController();
  TextEditingController preferred_hosp = TextEditingController();
  String? gender;
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
        title: appTitle,
        home: Scaffold(
            appBar: AppBar(
              title: Text(appTitle),
            ), // AppBar
            body: Form(
              key: _formKey,
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.center,
                children: <Widget>[
                  TextFormField(
                    controller: patient_name,
                    decoration: const InputDecoration(
                      icon: const Icon(Icons.person),
                      hintText: 'Enter patient name',
                      labelText: 'Patient Name',
                    ), // InputDecoration
                  ), // TextFormField
                  TextField(
                    controller: age,
                    decoration: new InputDecoration(
                        icon: const Icon(Icons.numbers_rounded),
```

```
                labelText: 'Patient Name',
              ), // InputDecoration
            ), // TextFormField
            TextField(
              controller: age,
              decoration: new InputDecoration(
                  icon: const Icon(Icons.numbers_rounded),
                  labelText: 'Patient age',
                  hintText: 'Enter patient age'), // InputDecoration
              keyboardType: TextInputType.number,
              inputFormatters: <TextInputFormatter>[
                FilteringTextInputFormatter.digitsOnly
              ], // <TextInputFormatter>[]
            ), // TextField
            DropdownButton<String>(
              hint: Text("Select Gender"),
              value: gender,
              onChanged: (String? value) {
                // This is called when the user selects an item.
                setState(() {
                  gender = value!;
                });
              },
              items: ["Male", "Female", "Other"]
                  .map<DropdownMenuItem<String>>((String value) {
                return DropdownMenuItem<String>(
                  value: value,
                  child: Text(value),
                ); // DropdownMenuItem
              }).toList(),
            ), // DropdownButton
            TextField(
              controller: preferred_hosp,
              decoration: const InputDecoration(
                icon: const Icon(Icons.local_hospital),
                hintText: 'Enter preferred hospital',
                labelText: 'Preferred hospital',
              ), // InputDecoration
            ), // TextField
            new Container(
                // padding: const EdgeInsets.only(left: 150.0, top: 40.0),
                child: new ElevatedButton(
              child: const Text("Submit"),
              onPressed: () async {
                Services.ref =
                    FirebaseDatabase.instance.ref("Bookings");
                //call to hashing function
                String hashvalue = AmbulanceForm().generateFormHash(
                    patient_name.text, age.text, preferred_hosp.text);
                Services.ref.update({
                  hashvalue: {
```

```
          patient_name.text, age.text, preferred_hosp.text);
        Services.ref.update({
          hashvalue: {
            "patient_name": patient_name.text,
            "age": age.text,
            "preferred_hospital": preferred_hosp.text,
            "gender": gender,
            "user_location": {
              "lat": Services.userLocation.latitude,
              "lon": Services.userLocation.longitude,
            }
          }
        });
        // ref.set({
        //   "patient_name": patient_name.text,
        //   "age": age.text,
        //   "preferred_hospital": preferred_hosp.text
        // });
      },
    )) // ElevatedButton // Container
  ], // <Widget>[]
), // Column
))); // Form // Scaffold // MaterialApp
  }
}

```

app_screen_res.dart

```
import 'package:AmbiNav/navig_notif_overlay_ui.dart';
import 'package:AmbiNav/search_overlay_ui.dart';
import 'package:AmbiNav/starter.dart';
import 'package:firebase_database/firebase_database.dart';
import 'package:flutter/material.dart';
import 'services.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'ambulance_form.dart';

class MapScreenRes {
  static void goToUserLoc() async {
    // Code to move the camera to user's current location
    // LocationData ld = await Services.locationData.first;
    Services.mapController.camera.lookAtPoint(Services.userLocation);
  }

  static List<Widget> getActionButtonList() {
    List<Widget> actionButtonList = [];
    if (Services.usertype == 'user') {
      actionButtonList.add(Padding(
          padding: const EdgeInsets.only(right: 15.0),
          child: IconButton(
              icon: Icon(Icons.search),
              onPressed: (() => Services.setStateOverlay(
                  () => SearchWidget.toggleVisibility()))))); // IconButton // Padding
    } else if (Services.usertype == 'driver') {
      actionButtonList.add(Padding(
          padding: const EdgeInsets.only(right: 15.0),
```

```
          child: IconButton(
            icon: Icon(Icons.navigation),
            onPressed: (() => Services.setStateOverlay(
                () => NavigationNotif.toggleVisibility())))); // IconButton // Padding
    }

    return actionButtonList;
  }

  static List<Widget> getDrawerOptions(BuildContext context) {
    List<Widget> drawerButtonList = [];
    drawerButtonList.add(GestureDetector(
      child: ListTile(
        title: const Text("Logout"),
        leading: Icon(Icons.logout_rounded),
      ), // ListTile
      onTap: () async {
        SharedPreferences logindata = await SharedPreferences.getInstance();
        logindata.setBool('login', true);
        logindata.setString('username', "");
        logindata.setString('usertype', "");
        Services.usertype = "";
        Navigator.pushReplacement(
            context, MaterialPageRoute(builder: ((context) => loginpg())));
      },
    )); // GestureDetector
    if (Services.usertype == 'user') {
      drawerButtonList.add(GestureDetector(
        child: ListTile(
          title: const Text('Book an ambulance'),
          leading: Icon(Icons.edit_note_rounded),
        ), // ListTile
        onTap: () => Navigator.push(
            context, MaterialPageRoute(builder: (context) => AmbulanceForm())),
      )); // GestureDetector
    }
    return drawerButtonList;
  }

  static void listenToBookings() {
    DatabaseReference ref = FirebaseDatabase.instance.ref("Bookings");
    ref.onChildAdded.listen((event) {
      Services.formDetails = event.snapshot;
      Services.setStateOverlay(() => NavigationNotif.toggleVisibility());
    });
  }

  static void search() async {
    // Code to implement search functionality
  }

  static Widget? chooseOverlayWidget() {
    if (Services.usertype == 'user') {
```

```
  }

  static void listenToBookings() {
    DatabaseReference ref = FirebaseDatabase.instance.ref("Bookings");
    ref.onChildAdded.listen((event) {
      Services.formDetails = event.snapshot;
      Services.setStateOverlay(() => NavigationNotif.toggleVisibility());
    });
  }

  static void search() async {
    // Code to implement search functionality
  }

  static Widget? chooseOverlayWidget() {
    if (Services.usertype == 'user') {
      return SearchWidget();
    } else if (Services.usertype == 'driver') {
      return NavigationNotif();
    }
    return null;
  }
}
```

app_screen_ui.dart

```
import 'package:AmbiNav/marker_details_ui.dart';
import 'package:AmbiNav/search_res.dart';
import 'package:AmbiNav/services.dart';
import 'package:flutter/material.dart';
import 'app_screen_res.dart';
import 'map.dart';

class AppScreen extends StatefulWidget {
  AppScreen({super.key});

  @override
  State<AppScreen> createState() => _AppScreenState();
}

class _AppScreenState extends State<AppScreen> {
  @override
  void initState() {
    super.initState();
    Services.mapContext = this.context;
    if(Services.usertype=="driver") {
      MapScreenRes.listenToBookings();
    }
  }

  //used to reference setState() for search widget (setState is copied to this variable in StatefulBuilder)
  var setStateOverlay;
  var setStateMarkerDetailsCard;
```

```dart
@override
Widget build(BuildContext context) {
  final scaffoldKey = GlobalKey<ScaffoldState>();
  return Scaffold(
    key: scaffoldKey,
    drawer: Drawer(
      child: SafeArea(
        child: Column(
          children: MapScreenRes.getDrawerOptions(context)
        ), // Column
      ), // SafeArea
    ), // Drawer
    appBar: AppBar(
        title: Text("Navigation"),
        leading: IconButton(
          //hamburger icon
          icon: Icon(Icons.menu),
          onPressed: () {
            if (scaffoldKey.currentState!.isDrawerOpen) {
              scaffoldKey.currentState!.closeDrawer();
            } else {
              scaffoldKey.currentState!.openDrawer();
            }
          },
        ), // IconButton
        actions: MapScreenRes.getActionButtonList()), // AppBar
    body: Stack(
      children: <Widget>[
        // MapWidget
        MapWidget(),
        //here the stateful builder is used to render search widget of search.dart (a card element to enter de
        //it renders without redrawing the entire screen
        //if the below lines are not included , map will be redrawn every time the search button is toggled
        StatefulBuilder(builder: ((context, setState) {
          Services.setStateOverlay = setState;
          return MapScreenRes.chooseOverlayWidget()!;
        })), // StatefulBuilder

        StatefulBuilder(builder: ((context, setState) {
          SearchRes.setStateMarkerDetailsCard = setState;
          return DisplayMarkerInfo();
        })), // StatefulBuilder
      ], // <Widget>[]
    ), // Stack
    floatingActionButton: FloatingActionButton(
        //this button moves the camera to user's current location - recenter button
        onPressed: (MapScreenRes.goToUserLoc),
        child: const Icon(
          Icons.add_location_alt_outlined,
          color: Colors.white,
        )), // Icon // FloatingActionButton
  ); // Scaffold
}
```

driver_details.dart

driver_details.dart ×

lib > driver_details.dart > ...

```dart
import 'package:AmbiNav/app_screen_ui.dart';
import 'package:AmbiNav/services.dart';
import 'package:flutter/material.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:google_fonts/google_fonts.dart';
import 'starter.dart';
import 'package:shared_preferences/shared_preferences.dart';

class AmbiDriverDetails extends StatefulWidget {
  AmbiDriverDetails({super.key});

  @override
  State<AmbiDriverDetails> createState() => _AmbiDriverDetailsState();
}

class _AmbiDriverDetailsState extends State<AmbiDriverDetails> {
  // final myController = TextEditingController();

  final nameController = TextEditingController();
  final codeController = TextEditingController();

  late SharedPreferences logindata;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.grey[300],
      body: SafeArea(
        child: Center(
          child: SingleChildScrollView(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                Icon(
                  Icons.airport_shuttle_outlined,
                  size: 90,
                ), // Icon
                Text(
                  'Ambulance Driver',
                  style: GoogleFonts.bebasNeue(
                    fontSize: 40,
                  ),
                ), // Text
                SizedBox(
                  height: 10,
                ), // SizedBox
                Text(
                  'Enter your details ',
                  style: TextStyle(
                    fontSize: 17,
                  ), // TextStyle
                ), // Text
                SizedBox(
                  height: 30,
```

```
              ), // SizedBox
              Padding(
                padding: const EdgeInsets.symmetric(horizontal: 25),
                child: TextField(
                  controller: nameController,
                  decoration: InputDecoration(
                    enabledBorder: OutlineInputBorder(
                      borderSide: BorderSide(color: Colors.white),
                      borderRadius: BorderRadius.circular(15),
                    ), // OutlineInputBorder
                    focusedBorder: OutlineInputBorder(
                      borderRadius: BorderRadius.circular(15),
                      borderSide: BorderSide(color: Colors.green.shade600),
                    ), // OutlineInputBorder
                    hintText: 'Ambulance Driver Name',
                    fillColor: Colors.grey[200],
                    filled: true,
                  ), // InputDecoration
                ), // TextField
              ), // Padding
              SizedBox(
                height: 10,
              ), // SizedBox
              Padding(
                padding: const EdgeInsets.symmetric(horizontal: 25),
                child: TextField(
                  controller: codeController,
                  decoration: InputDecoration(
                    enabledBorder: OutlineInputBorder(
                      borderSide: BorderSide(color: Colors.white),
                      borderRadius: BorderRadius.circular(15),
                    ), // OutlineInputBorder
                    focusedBorder: OutlineInputBorder(
                      borderRadius: BorderRadius.circular(15),
                      borderSide: BorderSide(color: Colors.green.shade600),
                    ), // OutlineInputBorder
                    hintText: 'Ambulance Code',
                    fillColor: Colors.grey[200],
                    filled: true,
                  ), // InputDecoration
                ), // TextField
              ), // Padding
              SizedBox(
                height: 20,
              ), // SizedBox
              Padding(
                padding: const EdgeInsets.symmetric(horizontal: 25),
                child: GestureDetector(
                  onTap: () async {
                    logindata = await SharedPreferences.getInstance();
                    String username = nameController.text;
                    String code = codeController.text;

```

```
                  if (username != '' && code != '') {
                    logindata.setBool('login', false);

                    logindata.setString('username', username);
                    logindata.setString('usertype', 'driver');
                    Services.usertype = 'driver';
                    Fluttertoast.showToast(msg: username);
                    Navigator.pop(context);
                    Navigator.pushReplacement(
                        context,
                        MaterialPageRoute(
                            builder: (context) => AppScreen())); // MaterialPageRoute
                  }
                },
                child: Container(
                  padding: EdgeInsets.all(20),
                  decoration: BoxDecoration(
                      color: Colors.green[400],
                      borderRadius: BorderRadius.circular(15)), // BoxDecoration
                  child: Center(
                    child: Text(
                      'Move to Map    🗺️',
                      style: TextStyle(
                        color: Colors.white,
                        fontWeight: FontWeight.bold,
                        letterSpacing: 1,
                        fontSize: 16,
                      ), // TextStyle
                    ), // Text
                  ), // Center
                ), // Container
              ), // GestureDetector
            ), // Padding
            SizedBox(
              height: 26,
            ), // SizedBox
            Row(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                Text(
                  'Want to change role ?',
                  style: TextStyle(
                    fontWeight: FontWeight.bold,
                  ), // TextStyle
                ), // Text
                Container(
                  child: GestureDetector(
                    onTap: () {
                      Navigator.of(context).pushReplacement(
                          MaterialPageRoute(
                              builder: ((context) => loginpg()))); // MaterialPageRoute
                    },
                    child: Text(
```

```
                        child: Text(
                          '  Click here',
                          style: TextStyle(
                            color: Colors.blue[600],
                            fontWeight: FontWeight.bold,
                            letterSpacing: 0.5), // TextStyle
                        ), // Text
                      ), // GestureDetector
                    ) // Container
                  ],
                ) // Row
              ],
            ), // Column
          ), // SingleChildScrollView
        ), // Center
      ), // SafeArea
    ); // Scaffold
  }

  @override
  void dispose() {
    nameController.dispose();
    codeController.dispose();
    super.dispose();
  }
}
```

FUNDING AND PUBLISHED PAPER DETAILS

Our article is being accepted by the **ICCES 2023 IEEE Conference**, which will be held from 1-3, June 2023 at PPG Institute of Technology, Coimbatore, India. We have been selected to deliver our oral presentation at the International Conference on Communication and Electronics Systems.

Accurate Location-Based Routing for Emergency Response: A Case Study of an App for First Responders using What3Words

Amitash M S
Department of Computer Science and Technology
Dayananda Sagar University
Bengaluru, India
amitu895@gmail.com

Rahul K P
Department of Computer Science and Technology
Dayananda Sagar University
Bengaluru, India
rahulkp01aug@gmail.com

Sumedh Hagaldivte
Department of Computer Science and Technology
Dayananda Sagar University
Bengaluru, India
sumedh.hagaldivte@gmail.com

Tejaal M
Department of Computer Science and Engineering
Dayananda Sagar University
Bengaluru, India
tejaalrani@gmail.com

Prof. Baskar Venugoplan
Department of Computer Science and Technology
Dayananda Sagar University
Bengaluru, India
bhaskarv-ct@dsu.edu.in

Dr. M Shahina Parveen
Department of Computer Science and Technology
Dayananda Sagar University
Bengaluru, India
chairman-ct@dsu.edu.in

***Abstract*—The timely delivery of emergency medical services is crucial in saving lives, particularly during the "Golden Hour" following a serious injury. Our study examines how different addressing schemes can impact the accuracy of reaching the exact emergency site on time. This paper discusses the comparison between the accuracy provided by GPS and an Alternate Addressing Scheme system (AAS). The potential benefits of using an AAS namely What3words are discussed in this paper.**

***Index Terms*—Global Positioning System (GPS), Alternate Addressing Schemes (AAS), What3words, Route accuracy, Representational State Transfer Application Programming Interface (REST API), Navigation, Mobile application**

I. Introduction

A. Background and Problem Statement

Road accidents are a major cause of fatalities and injuries worldwide. In 2017, officially reported road accidents were 464,910, claiming 147,913 deaths and 470,975 injured persons, that is, 405 deaths and 1,290 injuries each day from 1,274 accidents [1] [1].There are multiple documented incidents of positioning errors that have resulted in significant, often fatal, consequences [2]. One of the biggest challenges during this crucial period is the timely arrival of medical assistance to the crash site. Emergency medical services (EMS) face numerous obstacles such as traffic congestion, poorly marked roads, and the lack of real-time data on the quickest route to the accident location. In recent years, the widespread adoption of smartphones and the availability of navigation apps have presented an opportunity to address this challenge.

In this paper, we present a novel emergency navigation app that aims to assist users in availing ambulance services in emergency situations [2] and ensuring the timely arrival of the ambulance at the accident site and transportation of patients to hospitals. The app leverages a combination of real-time traffic and location data, as well as innovative features such as What3Words, to ensure error-free routing and improve the chances of saving lives during the "Golden hour".

Traditional route calculations make use of the user's location to calculate the route. This route is often inaccurate due to the reduced GPS accuracy. What3words has assigned 3-word addresses to 3m x 3m squares over the globe, making it highly accurate.

B. Research question

How can the use of what3words improve the accuracy and efficiency of emergency navigation systems?

II. Literature Survey

Accuracy has always been a top priority when sharing a location, even with basic grocery delivery apps that use mapping features. Traditionally, latitude and longitude coordinates are used to pinpoint a location. In this approach, the delivery person often fails to reach the precise location of the customer due to fluctuations in GPS and instead ends up in the adjacent street or another nearby location close to the destination. But, in case of an emergency, one cannot afford such mishaps like the emergency response vehicle reaching an adjacent street. The What3word approach intends to provide users with a more common vernacular to refer to spatial coordinates [4].

In India, 80% of the addresses are written with respect to a landmark which typically lies between 50-1500 meters of the actual address; such addresses make geolocating very challenging. Accuracy in geolocation is critical for emergency services to navigate quickly to reach you [9].

What3words is an Alternative Addressing Scheme (AAS). The What3Words system follows a Grid referencing technique and divides the ground in a 3-meter by 3-meter grid and assigns it a three-dictionary word separated by a dot [10]. It is designed to work on a variety of devices. The three-word code makes it easy to remember a location (Wired, 2017) [8]. This paper focuses on the core concept of What3words to highlight the fact that accuracy is important in emergency situations (Fire incidents, Hospital, Police) and how exactly What3words can be beneficial for addressing emergency situations. Similar paths are taken by the services what3emojis and what3fucks which use emojis and swearwords to encode a triangular geohash [7].

In the lat-long coordinate method, an incorrectly entered digit could result in a drastically inaccurate and undesired location. To highlight the simplicity of the What3words approach, a meeting point at the Bethesda Fountain in Central Park could be referred to as either 40.774489, -73.970871 in lat-long co-ordinates or more simply as vocab.neck.stem [4].

To address a specific location, there are several alternative approaches that may be used. What3words has a very high degree of accuracy and a low error rate, according to the results of an experiment done to compare the error rates and accuracy of various Alternative Addressing Schemes (AAS) [6].

A. Limitations of GPS

There are many causes of errors in the GPS data. A few of these are weather conditions, reflections in the atmosphere, and the reflection of the signals hitting buildings. This error rate may vary depending on the device used, the weather conditions at the location, and the density of the building [12]. According to a study conducted at the University of Georgia, the overall average horizontal position error of the iPhone 6 is in the 7–13 m range which is consistent with the general accuracy levels observed in recreation-grade GPS receivers [13].In an experiment undertaken at the Universiti Sains Malaysia in Malaysia, 78.3% of 92% of respondents reported GPS inaccuracy when using a GPS-based application. Furthermore, 78.3% of respondents think that GPS inaccuracy is mostly caused by a poor GPS satellite signal [14].

1) GPS signal strength: The location indicator in mapping apps relies on GPS technology to determine the user's location. If the device is in an area with poor GPS signal strength, such as indoors or in an area with tall buildings, the location indicator may fluctuate or even become inaccurate.

2) Interference: GPS signals can be affected by interference from other electronic devices, such as radios or microwaves. This interference can cause the location indicator to fluctuate.

3) Hardware issues: Sometimes the GPS hardware in a device can be faulty or malfunctioning, which can cause the location indicator to fluctuate or provide inaccurate readings.

4) App settings: Some mapping apps have settings that affect the accuracy of the location indicator. For example, if the app is set to low power mode, it may reduce the frequency of GPS updates, which can cause the location indicator to fluctuate.

5) Software issues: Occasionally, there may be software issues with the mapping app or the operating system of the device that can cause the location indicator to fluctuate.

But what3words overcomes all of the aforementioned drawbacks by using the position from the chosen 3word square rather than the mobile device's built-in GPS. However, What3words still has limitations, such as a fixed resolution and no consideration of elevation, which may impede efficient modeling in applications [5]. In reference to [3], Jiang and Stefanakis (2018) introduced a series of extensions of the w3w geocoding system to overcome these limitations.

III. Objectives

- To investigate the accuracy and reliability of traditional navigation systems in emergency situations.
- To evaluate the potential benefits of using what3words in emergency situations.
- To provide recommendations for the integration of what3words into emergency navigation systems.

IV. Methodology

A. Mobile App

We developed a mobile application, that includes HERE Maps to show the map on the screen. HERE Maps provides a generic routing algorithm that we have implemented to calculate routes for our experiment. The app makes use of What3words API to show the what3words grid overlay over the map which will help us choose a three-word square that represents the patient's location (refer to fig. 1).

B. Routing Algorithm

The experiment requires a routing algorithm to calculate the routes. The routing algorithm employed to conduct this experiment is a proprietary algorithm provided by HERE Platform as an integration with HERE Maps. The purpose of choosing this algorithm is to provide a consistent method of calculating the route between two given points.

The application is designed to calculate routes in two different ways. Both ways take the user's location as input. One method will use the patient's/user's current location (provided by GPS) to calculate a route. The second approach will display the What3words grid around the user's current location. This will give them a choice to select the correct square, thereby making the location selected accurate.

[1]Results are derived from a risk factor assessment that was conducted by the Academy of Family Physicians of India.

[2]Initially the focus was on ambulances but the service can be extended to other emergency services like fire engines.

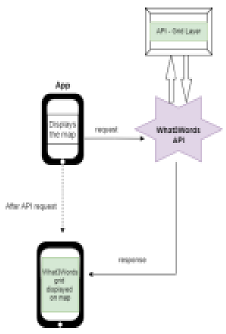


Fig. 1. What3words API call

C. Functionality

The application makes use of REST-ful API services provided by What3words to display the grid overlay [5]. The app makes REST API calls to obtain 3-word-coordinate addresses from Geo Coordinates (Latitude and Longitude). The Latitude and Longitude are obtained when the user taps on the map and from the user's current location [15].

V. Experimentation

A. Procedure

The first phase involves finding a route and the actual distance (between the hospital and the patient's address) using what3words, as illustrated below:

1) Select an initial point: For the purpose of the experiment, any hospital or fire station's geo-coordinates were picked and set as the origin at the start of each trial.
2) Select a three-word square from the grid: Each square represents a geo-coordinate. This square will serve as the destination when the ambulance is heading to a patient's location.
   On doing so, a function call is made to the HERE routing engine in the background with the geo-coordinates of the square as the destination.
3) Make note of the actual distance: After choosing a square, one can obtain a route between the initial point and the chosen square with an estimated distance. Record this value as the actual distance of the route.

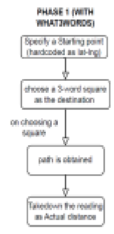


Fig. 2. Experiment procedure Phase- I

Now the what3words aspect is complete. The next phase is to map the same path multiple times, except this time with the GPS position as the destination rather than a 3-word square.

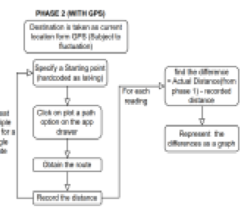


Fig. 3. Experiment procedure Phase- II

4) Choose a starting point just as before.
5) Now click on the "plot a path" button from the app drawer: which plots a path with current GPS coordinates (of the client device) as the destination.
   This step is synonymous with choosing a square in the first phase.
6) Note the path's distance: Here, It is observed that the distance acquired is not the same as previously, despite the fact that the path being plotted is between the same endpoints as before. This is due to the fluctuation in GPS (refer to section II-A).
7) Determine the difference: Actual distance obtained in step 3 subtracted from the distance observed in step 6.
   This distinction relates to the difference between a path estimated using standard GPS and a path computed using What3Words.
   Steps 4 to 7 should be repeated multiple times.

The graphs below show the irregularities in plotting a route several times. Here, difference refers to the difference in distance between a path computed using traditional GPS and What3Words. For this experiment, the following three different routes have been considered.

1) **Source:** Manipal Hospital, Rustam Bagh main road
   Destination: Reliable Dollar Colony, Huskur
   Distance: 20.1 km
   Average difference: 17.28 m

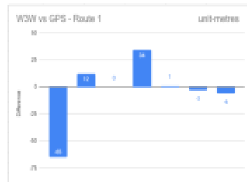


Fig. 4. Route 1

In fig. 4, from the first reading, we can infer that the destination being set by GPS falls short by 65 m in comparison with the actual destination point. Similarly, the second entry highlights the difference as 12m ahead of the actual destination point. Only in 1 out of the above 7 trials, the destination set by the GPS method coincides with the correct destination point which is represented as Zero meters. This means that the difference in the lengths of the paths calculated via What3Words and the GPS method is Zero. This corresponds to the ideal case.

2) **Source:** Sarjapur Road Fire Station
   Destination: Yellappa Garden, Banashankari 3rd stage
   Distance: 15.6 km
   Average difference: 152.75 m

In fig. 5, it is observed that there is a drastic difference in the "Average difference" parameter compared to the previous route. The maximum deviation seen is 194 m, which means the destination marker was 194 m ahead of the actual destination point which would result in an incorrect calculation of the distance. In this case, the distance (of the route) obtained using the GPS method is 15.79 Km. Similarly, the GPS marker is 141m behind the actual destination point in the first reading, resulting in an incorrect calculation of the distance as 15.45 Km.

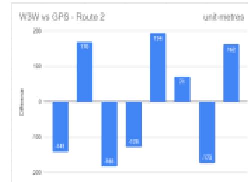


Fig. 5. Route 2

3) **Source:** Ramaiah Hospital, New BEL Road
   Destination: Classic Paradise Layout, Begur
   Distance: 14.7 km
   Average difference: 16.1 m

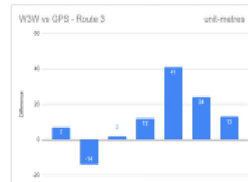


Fig. 6. Route 3

In fig. 6, the closest the GPS method could accurately set the marker was in the third trial, where the destination marker was just 2 m ahead of the actual destination point.

A negative value indicates that the calculated route is shorter compared to the actual route. On the other hand, a positive reading means that the calculated route is longer compared to the exact route.
Both the above cases correspond to the scenario where the ambulance reaches an adjacent or nearby street in place of the exact location.

In the first phase of the experiment, the destination of each route is subject to fluctuation due to the inaccuracies of the GPS system, resulting in different distances being calculated for each iteration. For the second phase, the destination is manually chosen by the user, selecting a 3m x 3m square. This greatly improves the accuracy and consistency (in terms of distance) of plotting the same route several times.

B. Results

The data gathered shows that there is a significant inconsistency in the average difference between the calculated paths for each route (with and without What3Words) since it is widely dispersed across a variety of values. The average difference ranges from 16.1m to 152m, depending on the route. But What3Words always yields an accuracy of up to 3m. Also, the probability of obtaining the correct and the most accurate route in the case of What3Words is 100% as opposed to the GPS method whose consistency is very low as proven by the above data.

VI. Discussion and Interpretation

The results of our study suggest that the use of what3words can significantly improve the response times of emergency navigation systems.

Our study found that traditional navigation systems were often inaccurate and unreliable in unforeseen situations. In contrast, the use of what3words allowed for precise navigation and improved response times.
These findings are consistent with previous research on the use of what3words in emergency response systems. According to the study conducted by the United Nations Development Programme, it was found that what3words improved the speed and accuracy of response times in emergency situations in Mongolia [11]. Our study builds on this research by developing an emergency navigation app that incorporates what3words and assessing its usability and effectiveness.
However, there are limitations to our study that should be acknowledged. Firstly, our sample size was small. A larger sample size would increase the generalizability of our findings. Secondly, our study was conducted in a controlled environment and did not account for deviation in route due to factors such as weather or road conditions. Further research is needed to assess the effectiveness of What3words in these situations.

VII. Conclusion

The what3words technique, as opposed to the conventional GPS method, consistently yields an accurate path at all times. Therefore, we can conclude that using the intrinsic location of a mobile device's GPS is not always the best way, especially in an emergency situation. Plotting the route using What3Words is always accurate. This is because it involves an additional input in the form of a 3-word square that corresponds to a Geo-coordinate. This location data is not subject to any fluctuations like those of the GPS. So essentially, the use of What3Words is synonymous with plotting a path with a source and destination that are both hard-coded. As a result, there is no room for fluctuations. This makes it more consistent and reliable.

In conclusion, the use of what3words has the potential to significantly improve the accuracy and efficiency of emergency navigation systems, leading to reduced response times and improved outcomes in emergency situations. Our study provides a framework for future research in this area and highlights the potential benefits of incorporating what3words into emergency response systems.

References

[1] Pal, Ranabir, Amrita Ghosh, Raman Kumar, Sagar Galwankar, Swapan Kumar Paul, Shrayan Pal, Debashis Sinha, A. K. Jaiswal, Luis Rafael Moscote-Salazar, and Amit Agrawal. "Public health crisis of road traffic accidents in India: Risk factor assessment and recommendations on prevention on the behalf of the Academy of Family Physicians of India." Journal of family medicine and primary care 8, no. 3 (2019): 775.
[2] Hadavi, Sheida, Sara Verlinde, Wouter Verbeke, Cathy Macharis, and Tias Guns. "Monitoring urban-freight transport based on GPS trajectories of heavy-goods vehicles." IEEE Transactions on Intelligent Transportation Systems 20, no. 10 (2018): 3747-3758.
[3] Jiang, Wen, and Emmanuel Stefanakis. "What3Words geocoding extensions." Journal of Geovisualization and Spatial Analysis 2 (2018): 1-18.
[4] Wiebe, Kyle. "Assessing the Utility Feasibility of Geo-spatial Addressing in Mathare, Nairobi." (2018).
[5] Jiang, W., and E. Stefanakis. "A RESTFUL API FOR THE EXTENDED WHAT3WORDS ENCODING." ISPRS Annals of Photogrammetry, Remote Sensing Spatial Information Sciences 4, no. 4 (2018).
[6] Clemens, Konstantin. "Comparative evaluation of alternative addressing schemes." (2016): 118.
[7] Clemens, Konstantin. "Geocoding User Queries." PhD diss., Technische Universität Berlin, 2020.
[8] Mokoloko, Boipelo, Thulisile Mphambukeli, and Verna Nel. "What's in a name? The dignity of an address in a smart city." Isocarp Review 13 (2017).
[9] Rastogi, Kabir, Samanta Bhattacharya, Margaret Church, and Ramesh Raskar. "What is the right addressing scheme for India?." arXiv preprint arXiv:1801.06540 (2018).
[10] Chander, Kailash, Harvir Singh, Sultan Singh, and Ajinkya Kulkarni. "Geolocation based Addressing system using GPS Coordinates for Smart Cities."
[11] United Nations Development Programme. (2019). Humanitarian Response with what3words: A Study in Mongolia. New York: United Nations Development Programme.
[12] Özdemir, Zeynep, and Bülent TUĞRUL. "Geofencing on the real-time GPS tracking system and improving GPS accuracy with moving average, Kalman filter and logistic regression analysis." In 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), pp. 1-6. IEEE, 2019.
[13] Merry, Krista, and Pete Bettinger. "Smartphone GPS accuracy study in an urban environment." PloS one 14, no. 7 (2019): e0219890.
[14] Thin, Li Nyen, Lau Ying Ting, Nor Adila Husna, and Mohd Heikal Husin. "GPS systems literature: inaccuracy factors and effective solutions." Int. J. Comput. Netw. Commun 8, no. 2 (2016): 123-131.
[15] Dreyer, Julian, and Ralf Tönjes. "What3Words Geo Encryption: An Alternative Approach to Location based Encryption." In IoTBDS, pp. 445-452. 2020.

8th International Conference on
Communication and Electronics Systems
ICCES-2023
1-3, June 2023 | Coimbatore, India
http://icoecs.org/2023/
icces2022@gmail.com

Letter of Acceptance

Details of Accepted Paper

Paper ID	Paper Title	Author(s)
ICCES395	Accurate Location-Based Routing for Emergency Response: A Case Study of an App for First Responders using What3Words	Dr.M Shahina Parveen, Prof.Baskar Venugoplan, Amitash M S, Rahul K P, Sumedh Hagaldivte, Tejaal M

On behalf of the Conference committee, I would like to congratulate you on having your article accepted by the ICCES 2023 IEEE Conference, which will be held from 1-3, June 2023 at PPG Institute of Technology, Coimbatore, India. You have been selected to deliver your oral presentation at the International Conference on Communication and Electronics Systems.

ICCES 2023 is an internationally-recognized IEEE conference, which dedicated solely for recommendation for inclusion in IEEE Xplore. Please visit the conference website for further updates [http://icoecs.org/2023/index.html].

As a result of the review and results, we are pleased inform that you can now submit the full-length paper for inclusion into the ICCES proceedings. We appreciate if you could send the final version of your research paper at your earliest convenience, in order to ensure the timely publication. When submitting your final paper, please highlight the changes made according to the review comments.

Thank you for your contribution to the ICCES 2023 conference.

Yours sincerely,

V. B

Dr. V. Bindhu

Conference Chair
ICCES- 2023.

International Conference on Communication and Electronics Systems ICCES

IEEE XPLORE COMPLIANT ISBN: 979-8-3503-9663-8
IEEE DVD ISBN: 979-8-3503-9662-1

Previous Publications
2022, 2021, 2020, 2019, 2018, 2017, 2016

IEEE COPYRIGHT AND CONSENT FORM

To ensure uniformity of treatment among all contributors, other forms may not be substituted for this form, nor may any wording of the form be changed. This form is intended for original material submitted to the IEEE and must accompany any such material in order to be published by the IEEE. Please read the form carefully and keep a copy for your files.

Accurate Location-Based Routing for Emergency Response: A Case Study of an App for First Responders using What3Words

Amitash M S, Rahul K P, Sumedh Hagaldivte, Tejaal M, Baskar Venugopalan, Dr. M Shahina Parveen

2023 8th International Conference on Communication and Electronics Systems (ICCES)

COPYRIGHT TRANSFER

The undersigned hereby assigns to The Institute of Electrical and Electronics Engineers, Incorporated (the "IEEE") all rights under copyright that may exist in and to: (a) the Work, including any revised or expanded derivative works submitted to the IEEE by the undersigned based on the Work; and (b) any associated written or multimedia components or other enhancements accompanying the Work.

Payment Receipt Transaction Reference: pay_LhlJS4peCR70pl

This is a payment receipt for your transaction on ICCES 2023 Non IEEE Members

AMOUNT PAID **₹ 7,000.00**

ISSUED TO
amitu895@gmail.com
+919972970828

PAID ON
23 Apr 2023

DESCRIPTION	UNIT PRICE	QTY	AMOUNT
UG PG Students Non IEEE Member	₹ 7,000.00	1	₹ 7,000.00

Total **₹ 7,000.00**

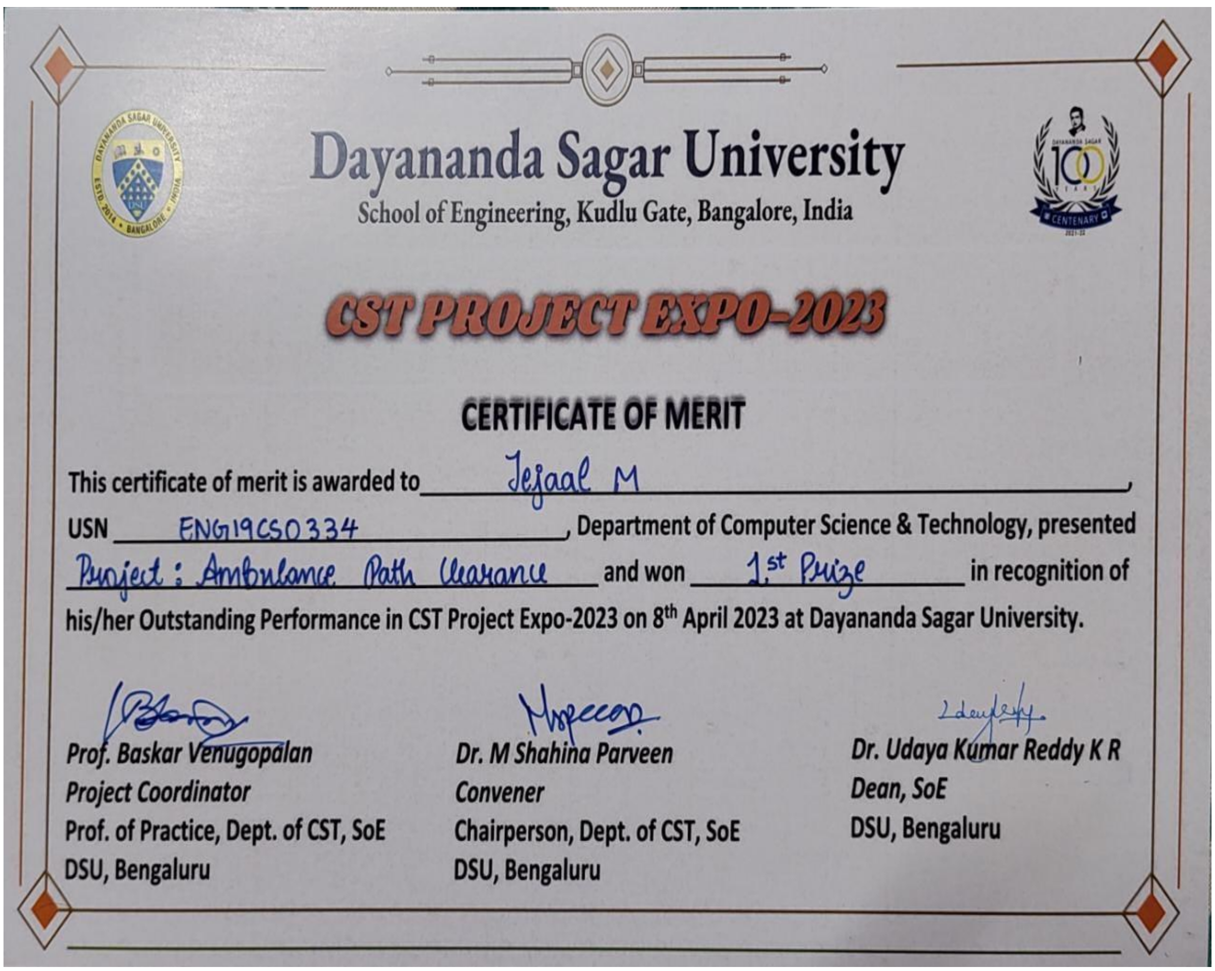
Amount Paid ₹ 7,000.00

PARTICIPATION IN PROJECT EXPO CONDUCTED BY CST & CSE

WINNING CERTIFICATE

CST Project Expo 2023

- Organized by Department of Computer Science and Technology.
- We participated in the CST Project Expo 2023 conducted on 8th of April,2023
- We are also happy to have **won FIRST prize** for it.



Dayananda Sagar University
School of Engineering, Kudlu Gate, Bangalore, India

CST PROJECT EXPO-2023

CERTIFICATE OF MERIT

This certificate of merit is awarded to Tejaal M, USN ENG19CS0334, Department of Computer Science & Technology, presented Project : Ambulance Path Clearance and won 1st Prize in recognition of his/her Outstanding Performance in CST Project Expo-2023 on 8th April 2023 at Dayananda Sagar University.

Prof. Baskar Venugopalan
Project Coordinator
Prof. of Practice, Dept. of CST, SoE
DSU, Bengaluru

Dr. M Shahina Parveen
Convener
Chairperson, Dept. of CST, SoE
DSU, Bengaluru

Dr. Udaya Kumar Reddy K R
Dean, SoE
DSU, Bengaluru

PARTICIPATION CERTIFICATE

ProjectExpo-2k23

- Organized by Department of Computer Science and Engineering
- We participated in the Project EXPO conducted on 13th of April,2023.
- We were selected as the **top 28 batches out of 100** of them.

GITHUB LINK

https://github.com/CSE-DSU/Team-78-AmbiNav