ASHESI UNIVERSITY

LAMBER: AN EMERGECNY RESPONSE SYSTEM FOR AMBULANCE REQUESTS AND FIRST AID GUIDE

APPLIED PROJECT

BSc. Computer Science

Aileen Lisa Adze Akpalu

2022

ASHESI UNIVERSITY

LAMBER: EMERGECNY RESPONSE SYSTEM FOR AMBULANCE REQUESTS AND FIRST AID GUIDE

APPLIED PROJECT

Applied Project submitted to the Department of Computer Science, Ashesi University College in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Science.

Aileen Lisa Adzo Akpalu
2022

Declaration

I hereby declare that this Applied Project is the result of my own original work and that no partof it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

....

Candidate's Name:

.............Aileen Lisa Adzo Akpalu...

Date:

..............13th May 2022..

I hereby declare that preparation and presentation of this Applied Project were supervised in accordance with the guidelines on supervision of Applied Projects laid down by Ashesi University.

Supervisor's Signature:

....

Supervisor's Name:

............David Sampah..

Date: ....13th May 2022...

Acknowledgements

First and foremost, I want to thank God for how far He has brought me in my Ashesi journey. To my supervisor, David Sampah, whose encouragement, and academic advice helped me undertake this project. To my parents who have supported me and given me counsel on this project. A special thanks to Kwabena Obeng Koranteng for helping me in completing this project.

Abstract

Emergency Medical Services operations in Ghana are posing a challenge to the health sector in Ghana. Many people do not have access to emergency response services, let alone good amenities to secure their good health. Unfortunately, there is no digital application in Ghana that will allow people to request emergency response services to make life easier.

LAMBER is an emergency medical service software application that has three main applications for patients, clinics, and paramedics. The system offers two major services namely ambulance requests and First-Aid guides for potential patients. The system will allow patients to create requests and link them to the nearest ambulances based on their location. The patients are given the luxury to choose their ambulances and can send a general message to all available ambulances in case of any unforeseen emergencies.

Future work includes developing an AI-powered first aid guide.

TABLE OF CONTENTS

Chapter 1 Introduction

1.1 Background

Healthcare is an integral sector of any country's economy and each year a lot of money is budgeted for the sector, especially in Ghana. In the year 2018, 1.8% of Ghana's GDP was allocated to the health sector as compared to 1.1% in 2017[1]. One key component of the health sector is the emergency medical services which are made up of the National Ambulance Service and the National Fire Service [2]. The National Ambulance Service was set up in 2004 to establish and operate a nationwide comprehensive pre-hospital emergency care [2]. The Government of Ghana in 2020, commissioned the One Constituency-One Ambulance initiative to allow people to gain better access to emergency medical services [3]. Currently, there are three hundred ambulances in the country of which two hundred and seventy-five are situated in each constituency and thirty are stationed at the headquarters for extra support [3].

Generally, ambulance services were set up to offer 24/7 response to any medical emergencies, temporary treatment, and transport the affected persons to the hospital [4]. Despite the integral role that the ambulance service plays, many challenges undermine their value and thus contribute to the low patronage and utilisation [4]. According to Adamtey et, there are many factors that affect the ambulance services in Ghana, and they include but are not limited to the following:

- unavailability of ambulances
- poor road networks
- poor receptibility of ambulance services
- inadequate number of ambulances
- rising cost of fuel

- inexperienced trainees and staff [5].

This paper proposes a solution that will not only aid in the acceptance of the ambulance services but also connect citizens to emergency medical services efficiently, offer first aid information and reduce the delay time from the emergency medical service personnel.

1.2 Motivation

Due to the negative factors that affect the National Ambulance Services stated above, many people resort to using private cars and taxis to transport people in need of assistance to the hospitals. In 2012, the ratio of ambulances to citizens was 1:250,000. After the introduction of one constituency, one ambulance initiative [3], the ratio now stands at 1:84,000 as of 2020 [2]. Although there is a substantial decrease in the number of persons connected to one ambulance, it is still a high number and only having a stringent management system and plan can help make the problem better.

In 2014, the response time of the NAS was 0.3 per day which is relatively low as compared to other countries that remain within a range of 2 to 20 responses a day [2]. In a news article about an occurrence of a road accident on the Tema Motorway earlier this year, the road users blamed the slow response time and absence of the emergency response for the death of the affected persons [6]. For over two hours, the road users tried to attend to the people and treat them as if there was no sight of any emergency response [6]. In such a situation, if there was a faster and more effective way of reaching any ambulance, that death could have been avoided. The response time of the ambulance to the designated location resulted in preventable death and injuries [6]. Unfortunately, this is not the only case.

Aside from the internal problems facing the ambulance service, some external challenges also need special attention. Some of these challenges are

- the prank calls or false alarms
- lack of street names and house numbers [2].

These issues, especially the prank calls, cannot be solved immediately, but a system can be put in place to reduce the number drastically. Regarding the issue of street names, the government has implemented a digital address system where all structures are required to access their digital address [7]. There are still some citizens who do not have a digital address or an official street name, making it difficult to locate some people [2].

1.3 Related Work

There are various emergency medical services applications and similar projects relevant to this project. Although some of the projects may not be the same idea, the techniques employed by the authors have similar approaches to that of this project.

1.3.1 Verifly: A drone supported first aid system for Ashesi University

The capstone project: Verifly, was completed in the year 2019 by Verissa Owusu and it is a drone supported first aid system for Ashesi University. The main aim of Verifly was to utilize drones to deploy emergency healthcare and offer extra support to first responders to an incident or injured people. The project makes use of technologies like "GPS location, mobile applications, web services, drones, and ground control stations" [8]. Users of this application will be able to request for first aid kit and assistance on the mobile application, and the drones will deliver the requested items to the user [8].

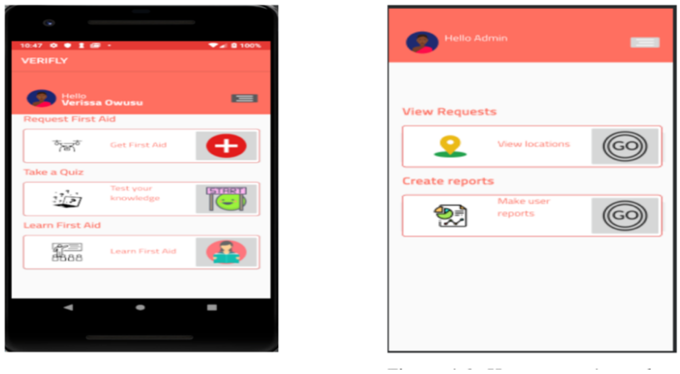


Figure 1.1 Home page for Verifly

1.3.2 Locating the nearest pharmacy with the desired medicine

The capstone project was completed in 2018 by Brenda Mboya. It is an application that allows users to request medicine and locate the nearest available pharmacy with the drug. Patients can also reserve medicinal products and gain access to directions to the place via google maps. The application enables pharmacists to view requests made by the client and gives them the ability to update the pharmacy's inventory [9].

Based on the requests made, the pharmacies can use the data to make informed decisions and know which products bring the most sales. With a simple click of a button, users can quickly locate the medications they want and reserve them [9].

1.3.3 A Location-Based Rescue Service App

The capstone project was completed in 2018 by Siyuan Lisa Shen of Pennsylvania University. The project aims at developing an efficient emergency response system for people over the age of 60. It consists of two main parts, one for the users and the other for the hospital administrators. The users have an Android application that will enable the seniors to send location signals those nearby hospitals will receive. The administrators can check for emergency updates on their interface and keep a constant check on the user regarding their current location. The application is designed to be a cohesive alternative to the traditional call and response system. The developer uses Android Development Tools, which is open source, to develop the application [10].

1.3.4 Automatic Ambulance Dispatch System via One-Click Smartphone Application

The research conducted was to propose a comprehensive system that tackles the problem of manual ambulance dispatching and replace it with the Automatic Ambulance Dispatch System (AADS) [11]. AADS consists of an android-based application where the user must press a "help" button on the application to signal nearby ambulances with the user's geographical location. The main objective of their work was to decrease the delay time of the ambulance to provide efficient service [11]. The application consists of an android application for both users and ambulance drivers and an online server.

1.4 Proposed Solution

The proposed solution for this problem is an application dubbed LAMBER, which will have a system for the users and a system for the private hospitals and clinics.

Lamber will seek to fill the gaps that the other related projects discussed in the previous section (1.3) do not have and incorporate the systematic procedure that the clinics follow with their ambulance services.

Lamber will be a three-part integrated system that will help the data flow between the potential patients, private hospitals, and the paramedics. The other related project, such as the AADS, made no mention of having an integrated system but instead focused more on the functionalities of the primary users. With Verifly, there was a system for their secondary users, but they could only view the requests made and get statistical analysis of the data.

There was no mention of whether this category of users accepts requests or has live tracking. Users can only see the locations of all the requests on a map.

Lamber will take a different approach as the patients, administrators, and the paramedics can live track ongoing requests. Private hospitals would also accept requests and dispatch the paramedics with the necessary information as it is done on the ground.

This integrated system will help digitise the collection of data pertaining to the users and their use of emergency services and thus make it easier to review, analyse and search for data.

Chapter 2 Requirement Analysis

2.1 Introduction

This section highlights the requirement analysis, identification of user categories and requirements specifications. Lamber consists of only a software component that all categories of users will use. The users of the application are:

1. Patients
2. Private Hospital Administrators
3. Paramedics in private clinics

2.1.1 Data Gathering and Analysis

The primary mode of gathering data was using questionnaires and interviews with key stakeholders (Appendix A) that encapsulated the whole idea of emergency medical services in Ghana and the introduction of Lamber to aid that process. The questions used for the interview were mainly directed toward key stakeholders such as doctors and EMS personnel. The questionnaires were primarily directed towards the users, who are potential patients who will use the system to make requests. Below are the questions used for the questionnaire and interview.

2.1.1.1 Questionnaire

Appendix A contains a list of relevant questions asked to potential users and patients for this project. There are two significant questions that address the problem this project seeks to address.

1. Have you ever called the National Ambulance Service or a private ambulance?
2. How effective do you think the NAS is?

3. …. Check appendix …

2.1.1.2 Interview

Appendix B contains a list of relevant questions asked to some doctors and private clinics that have private ambulances. There are two major questions that address the problem the project is seeking to address.

1. How many times do you get patients using the ambulance service?
2. Do you think having an application software will help facilitate EMS requests?

2.1.2 Analysis

Based on the conversations had with a private clinic and public health official, a summarised version of what functionalities Lamber will have and how the integrated system will operate are described below.

The system to be used by patients will have two main sections: the ambulance request services and first aid guide. For the request section, users can have the luxury of choosing their desired hospital with an ambulance or sending a general SOS message. Users are allowed to either type special notes or send a voice note when making specific requests.

The SOS message section will work by clicking a button, and then a request is sent to all nearby hospitals. Users can also make requests for others, especially in dire situations where affected persons cannot make the requests themselves. Users can choose between subscribing to a package or using a pay-as-you-go service to access the functionalities.

The system for the hospitals will have a section for the administrator and the paramedics. Hospital administrators will be able to monitor all requests coming into their hospital and assign paramedics to them. The paramedics will be able to receive the coordinates and information from their administrator for execution.

2. 2 Use Case Diagram

Figure 2.1 is a use case diagram that shows the relationship between the user(patient) and the hospital administrator. The diagram explains how each user uses the application.

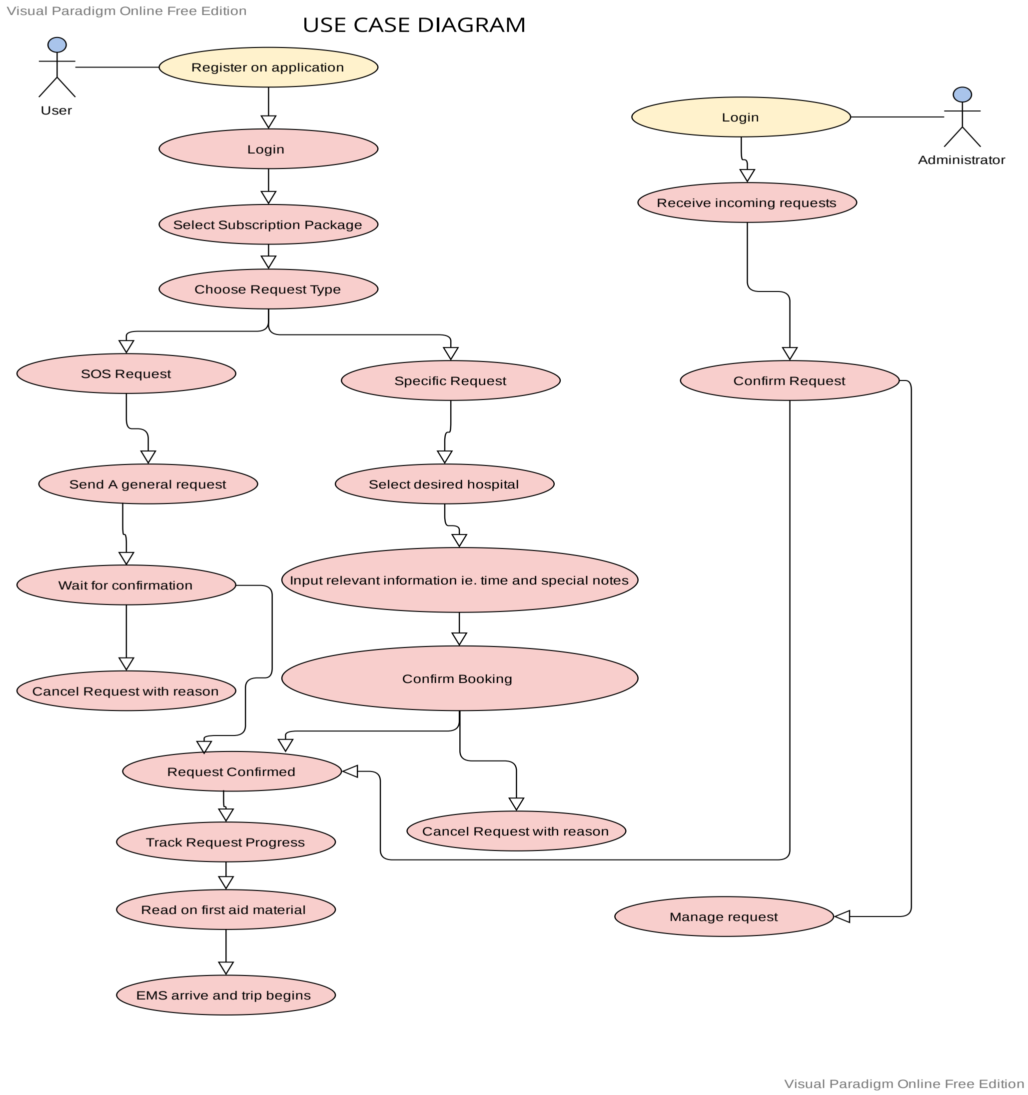


Figure 2.1 Use Case Diagram

2.2.1 Use Case Scenario

Ama is a 25-year-old woman who lives in her home with her 50-year-old mother. Ama has tried to call an ambulance to her home to transport her mother to the hospital on several occasions. She faces two problems; either the ambulance service is hard to reach or late to respond. She decides to sign up on Lamber and selects a subscription package. Her mother had just suffered from a mild concussion at home, so she decided to click on the SOS option on the application's home page. The application picked up her location, and she received a response in five minutes, and her designated ambulance was ten minutes away. Upon their arrival, her mother was attended to by the paramedics

In another scenario, User A referred to the concussion section on the first aid page of the application. She helped her mother regain some consciousness by following the instructions. However, further assistance was required, so User A scheduled an ambulance pick up for 11 AM with her preferred clinic. The clinic responded, got her location, and dispatched the ambulance to her.

2.3 User Categories

There are three major users of the system categorised into two main user categories for this project: primary users and secondary users. The primary users are the clients who will use the application to create requests. The secondary users are the administrators and EMS personnel, who will accept and manage those requests.

2.4 Functional Requirements

2.4.1 User Registration

<u>Specification</u>: The system should allow users to register and sign into the application.

Explanation: Both user categories will have different registration processes, especially regarding the information needed. The main goal is to allow users to register on the system easily.

2.4.2 Selective Requests

Specification: The system should allow primary users to select the hospital of their choice to make a request.

Explanation: The primary users have the option to select what hospital they would want to request from. The users can then add special notes for the personnel.

2.4.3 General Requests

Specification: The system should allow primary users to send a general request to all nearby hospitals in case of emergency

Explanation: The primary users have the option to send SOS messages to all nearby hospitals in a state of emergency. Users can use voice activation or click a button and add special requests.

2.4.4 Subscription Plan

Specification: The system should let primary users subscribe to a payment plan that will give them access to a few functions on the application.

Explanation: The primary users can choose which subscription plan they would like to be on. Users can upgrade to a higher plan or do a pay per request plan.

2.4.5 Cancellation Fee

Specification: The system should allow primary users to cancel their request after they have paid a fee

Explanation: The primary users have the option to cancel their request in progress after a minimum of 5 minutes. Users who are not subscription plans and are on a pay per request will have to pay to cancel.

2.4.6 Voice Requests

Specification: The system should allow primary users to make SOS requests using voice notes

Explanation: The primary users have the option to send voice notes for SOS requests. The system will then save and pick up the user's location and send the voice note across.

2.4.7 Voice Messages

Specification: The system should allow primary users to send voice messages for extra special notes for the personnel

Explanation: The primary users have the option to record a voice note for the additional notes they have on their request.

2.4.8 Location Tracker

Specification: The system should allow constantly pick up the location of the user and update when necessary.

Explanation: The system will run a GPS tracker in the background to monitor and pick up the user's current location.

2.4.9 Request Management

Specification: The system should allow secondary users to manage all incoming requests and track their finances.

Explanation: The secondary users have two levels: the administrator and the EMS personnel. Each level will have a request management system that will allow them to monitor all requests. The only difference is that the administrator can add new personnel to their system and track income.

2.5 Non-Functional Requirements

2.5.1 Security

Specification: The system should protect the user's data.

Explanation: The system should be designed such that no one has access to another user's data, especially with the payment process. User's data will be protected and confidential.

2.5.2 Reliability

Specification: The system should be reliable for both user categories.

Explanation: The system should not be malfunctioned and must be designed to be reliable.

2.5.3 Scalability

Specification: The system should be scalable to cater for concurrency

Explanation: The system should be able to handle multiple users and multiple requests simultaneously.

2.5.4 Efficiency

Specification: The system should have a highly efficient rate when being used.

Explanation: The system should be fast when used and should not have any delay.

Chapter 3 Approach, Design and Architecture

3.1 Approach

The approach adopted in the project is the agile model, which involves a software development approach using iterative development. There is constant interaction with the stakeholders whilst ongoing development to better understand what is essential. Tasks are broken down into smaller iterations, and this the project risk and reduce the time spent on development [12]. The stakeholders are:

1. Private clinics with ambulances
2. Citizens of Ghana
3. Paramedics in private clinics

3.2 Design

The system scope for this project is both a web application and a mobile application. The web application is for the administrators of the clinics, and the mobile application is for potential patients and paramedics.

The system's design includes a layered architecture, activity diagrams, use case diagrams and sequence diagrams. Each of these diagrams explains the different processes involved in the project and how these components work together.

3.2.1 System Overview

3.2.1.1 Activity Diagram

There are two main processes for the application, which will be centred around the user. The activity diagrams for the processes are shown below.

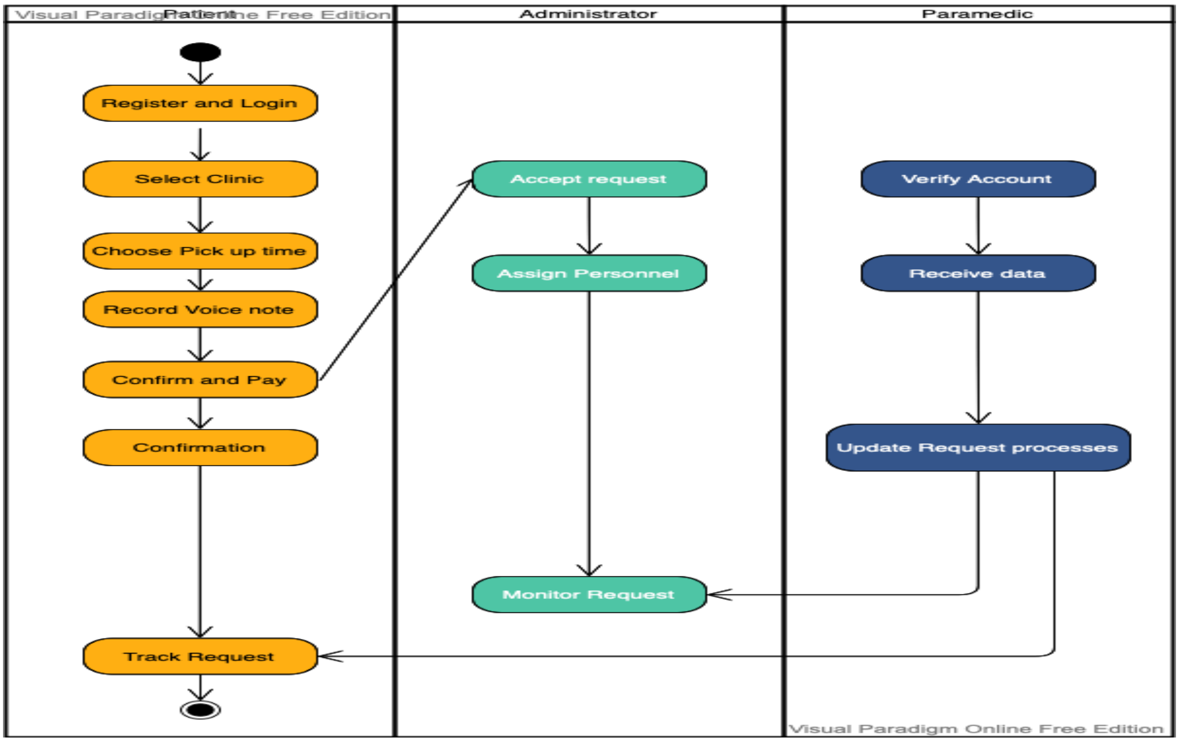


Figure 3.1: Activity Diagram for Specific Request

In Figure 1, the activity diagram shows the step-by-step process between the user, administrator, and paramedic when a user creates a specific request.

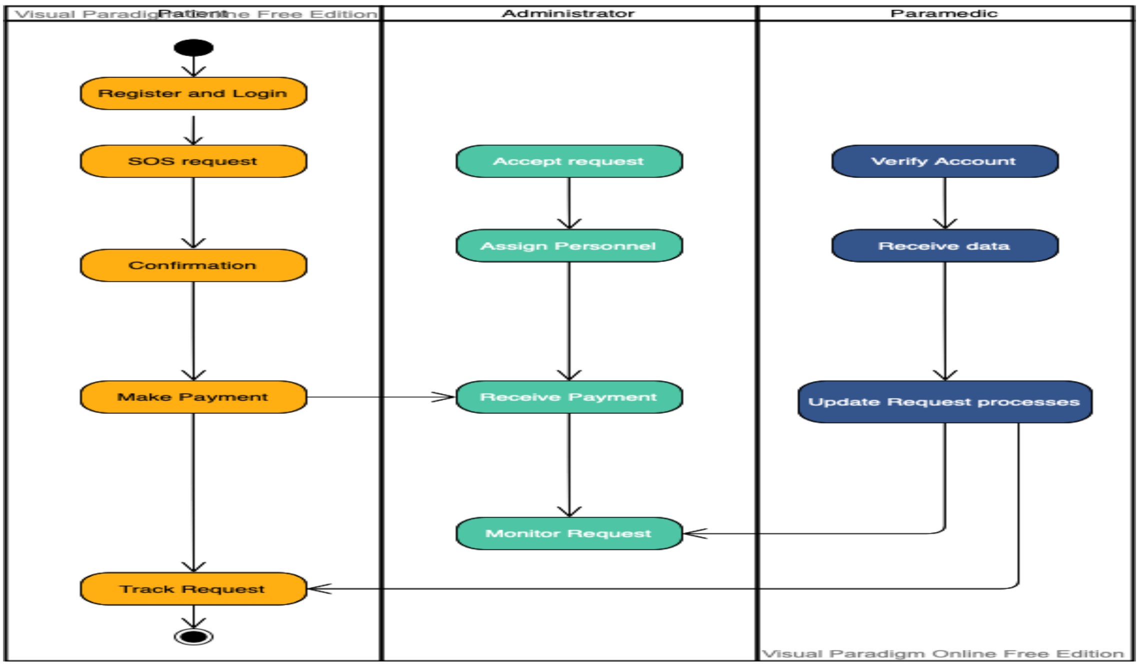


Figure 3.2 Activity Diagram for SOS request

In Figure 3.2, the activity diagram shows the step-by-step process between the user, administrator, and paramedic when a user sends an SOS request.

3.2.1.2 Entity Relationship Diagram

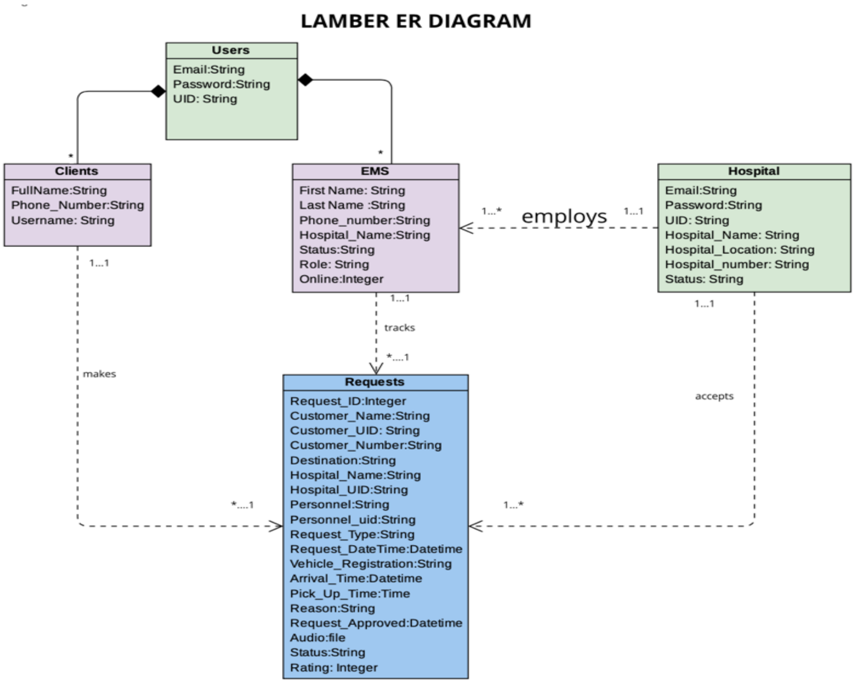


Figure 3.3 ER Diagram

3.2.2 System Architecture

3.2.2.2 Layered Architecture

A three-tier architecture will be adopted for this system. The database that will be adopted will either be Firebase. This is because it will help transfer and facilitate the sharing of data between mobile applications and web applications. The first layer is the presentation layer or the user interface, which would be on the mobile phone for the "patients" and the PC for the hospital personnel. Both categories of users will share the same database and server, the application and database layers. Below is the three-tier architecture.

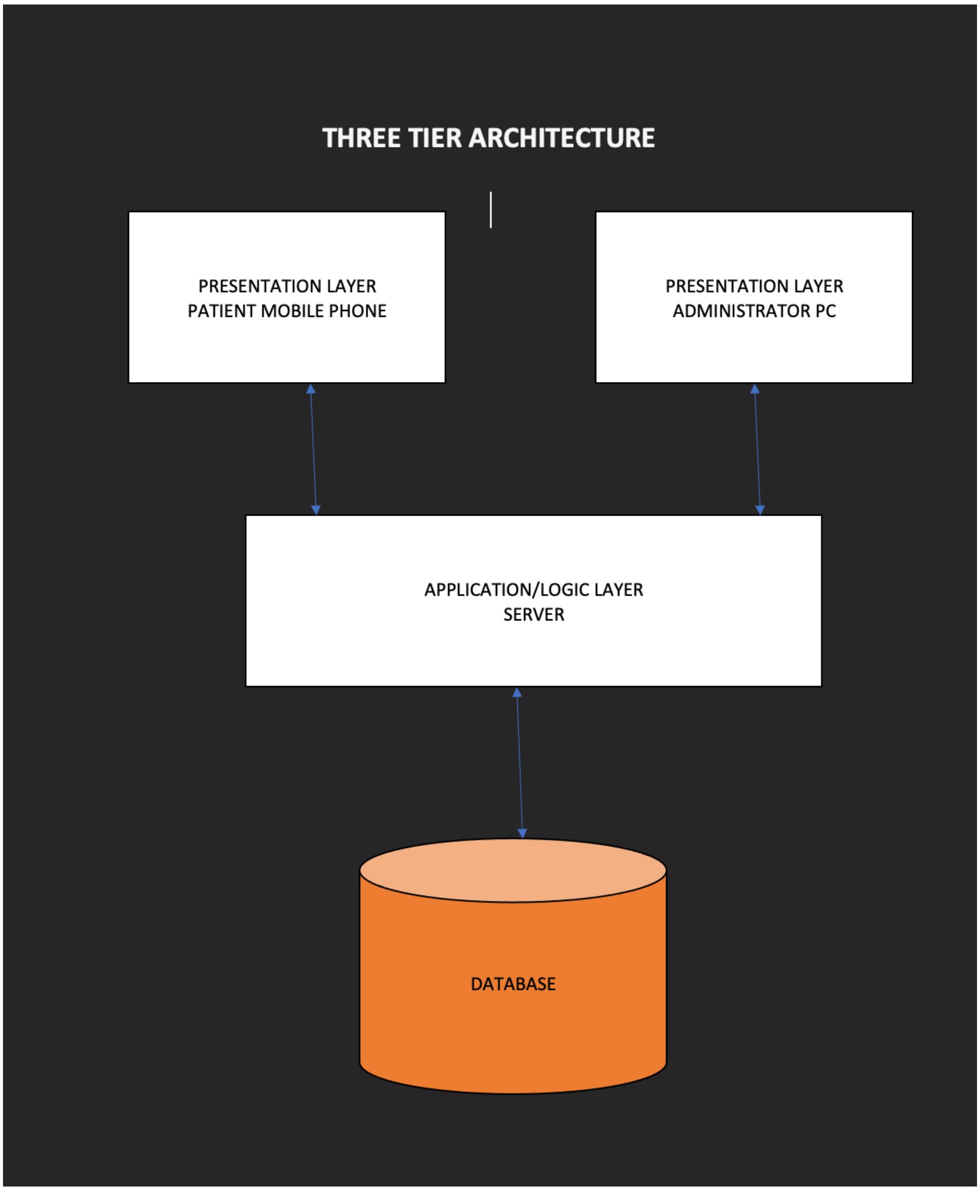


Figure 3.4 A three-tier architecture diagram

3.2.3 Software Design

There is an ideation and prototype phase under the design phase of the software development cycle. For the ideation, the application's design for both users and administrators will undergo various stages until a final prototype is completed. Two ideation phases and the final prototype have been completed.

Below are two sequence diagrams explaining how a user can make a request. The figure below (Figure 3.5) explains the process when users select their own hospital, fill the request form, pay, and send a request to them. The administrator then accepts and assigns the request to a paramedic. The paramedic then starts the request, and then live tracking of the request begins.

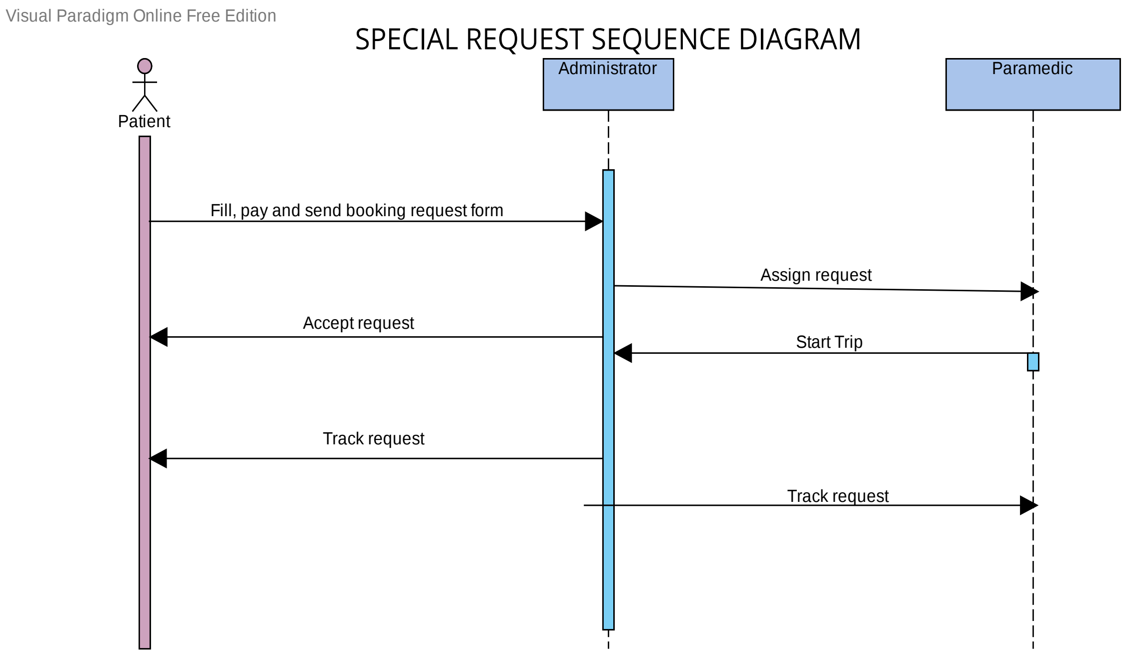


Figure 3.5 Sequence diagram for special request.

The figure below (Figure 3.6) explains the process when a user sends a general request to the hospitals in the user's location. An administrator from any of the hospitals then accepts and assigns the request to a paramedic. The paramedic then starts the request, and then live tracking of the request begins.

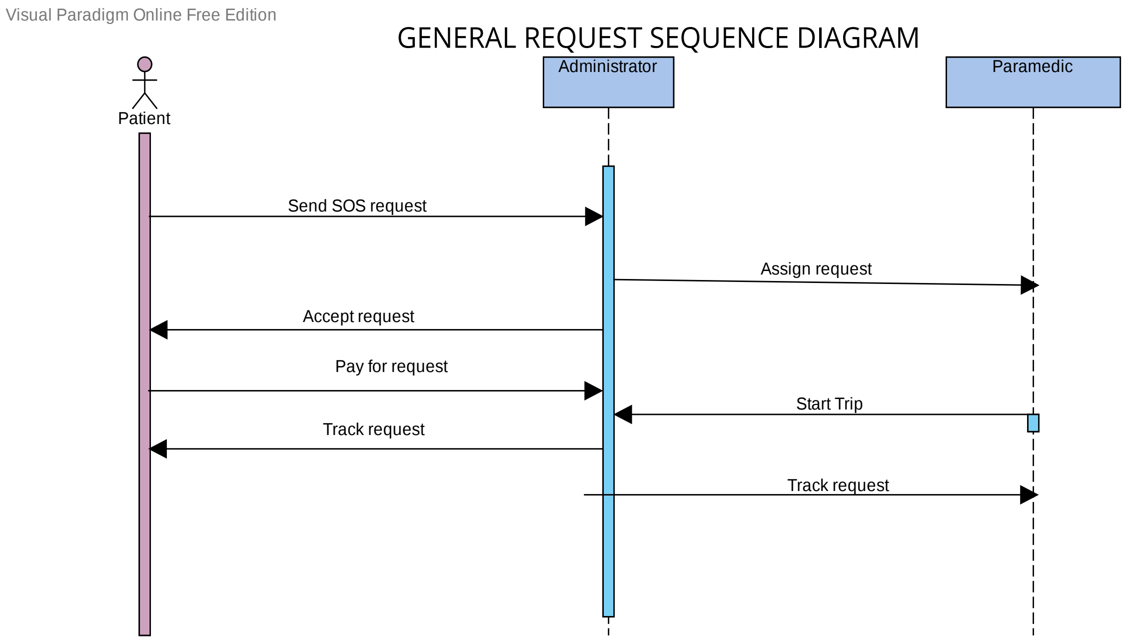


Figure 3.6 Sequence diagram for general request.

Below are screenshots of the final prototypes for the mobile application and web application.

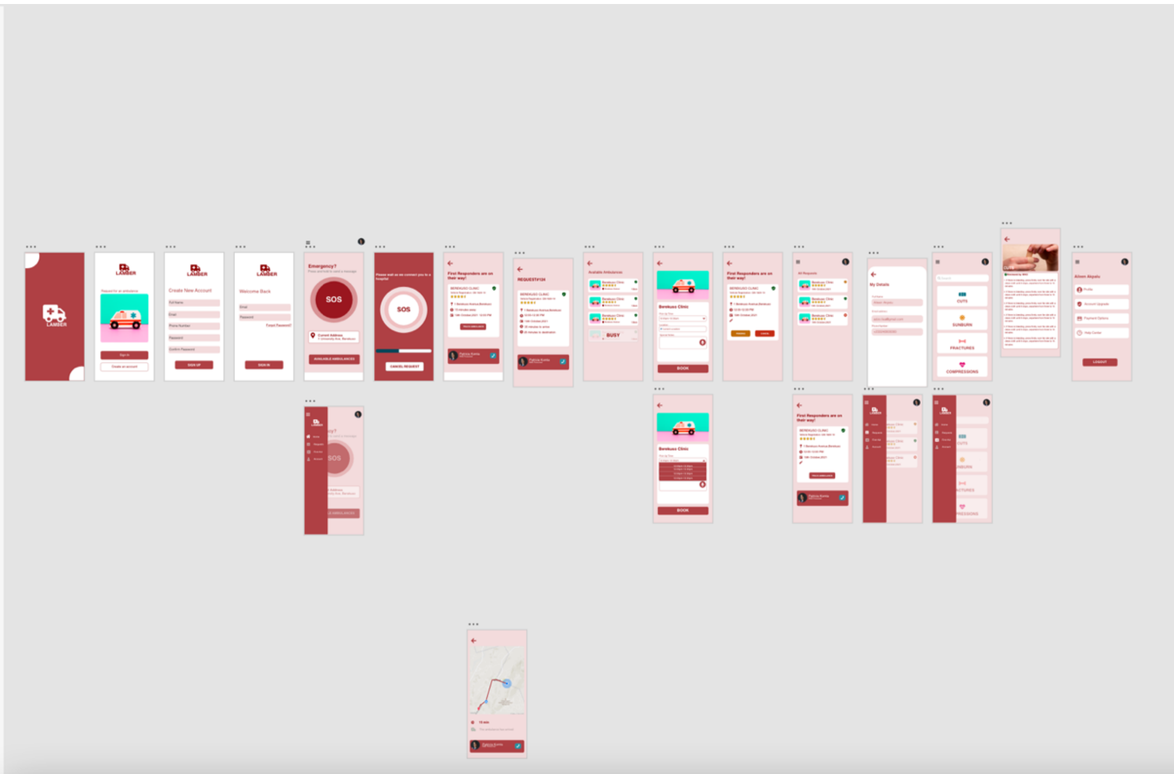


Figure 3.7 Prototype for primary users

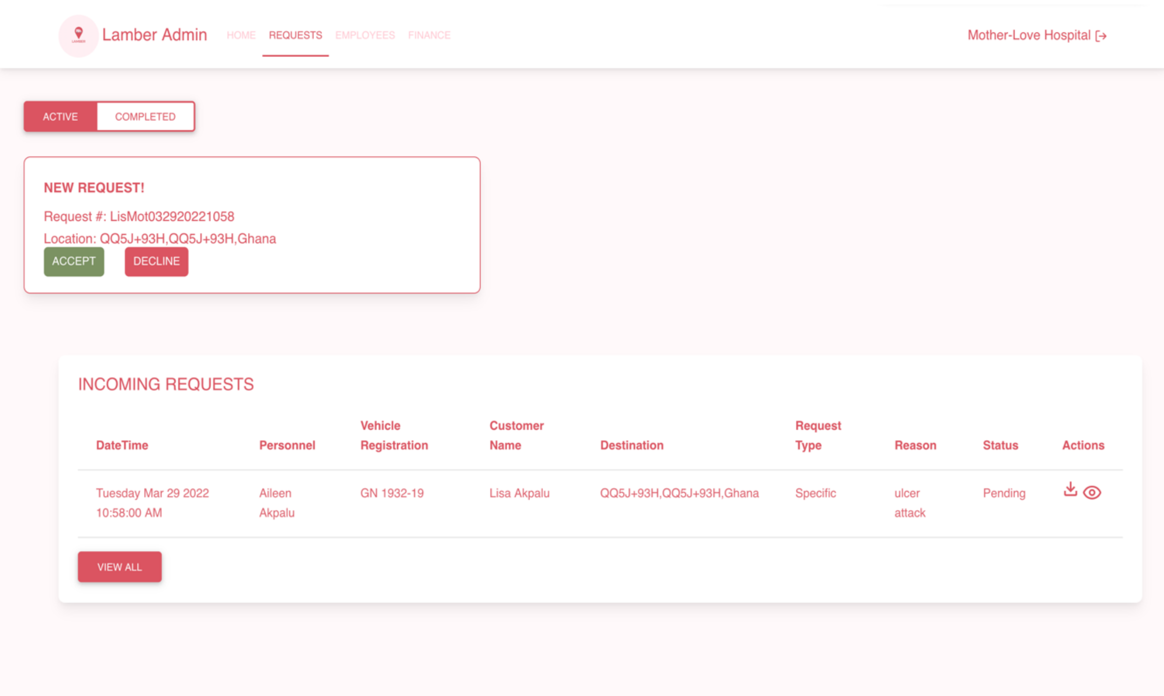


Figure 3.8 Request page for hospital administrators

3.3 Tools and Technologies

The implementation phases in the software development lifecycle involve what will be used to build the system and how it will be done. For this project, the tools that will be used to develop the systems are Github, Flutter, Visual Studio Code, Android Studio, Adobe XD, Figma and Firebase.

3.3.1 Cross-Mobile Platform

The technology used for building the mobile application for the patient and paramedic is flutter. Flutter was used because it is designed to create a cross-platform mobile application. It is also faster than many application frameworks as it has a reduced code development time [13]. The patients and the paramedics will use the mobile application as stated.

3.3.2 Web Application

The technology used for building the web application for the administrators is React JS Library. React JS is flexible and user friendly for testing [14]. The other technologies utilised are HTML, CSS, JavaScript, MDB UI and Node JS.

3.3.3 Google Maps API

The use of Google Maps API for this project is essential as it brings the maps directly to the mobile and web applications. The API is utilised on both web and mobile applications. The API allows for the addition of location features such as getting the user's current location to help connect them to the nearest hospitals [15].

3.3.4 Firebase

Firebase is a platform provided by Google that has functionalities that aid in the backend development of Android, IOS or web applications [16]. The tools for Lamber are Firebase Database, Firebase Cloud Messaging and Firebase Authentication.

The Firebase database is a NoSQL database which is a good choice, especially when dealing with massive datasets [17]. NoSql is relatively faster than MYSQL as it horizontally scales the data [4]. The database to be used will be the Realtime Database, a cloud-hosted database that can be used to build collaborative applications [18]. The data is stored in a JSON format and synchronizes with every connected client; thus, if there is any change, a connected device updates in milliseconds [18].

3.3.4.1 Firebase Authentication

Firebase authentication makes building secure authentication systems easy. It provides an end-to-end solution for identity using various forms such as email and password accounts, phone authentication, Google and many more [19].

For this project, the email and password accounts, as shown in Figure 3.9, will be used.

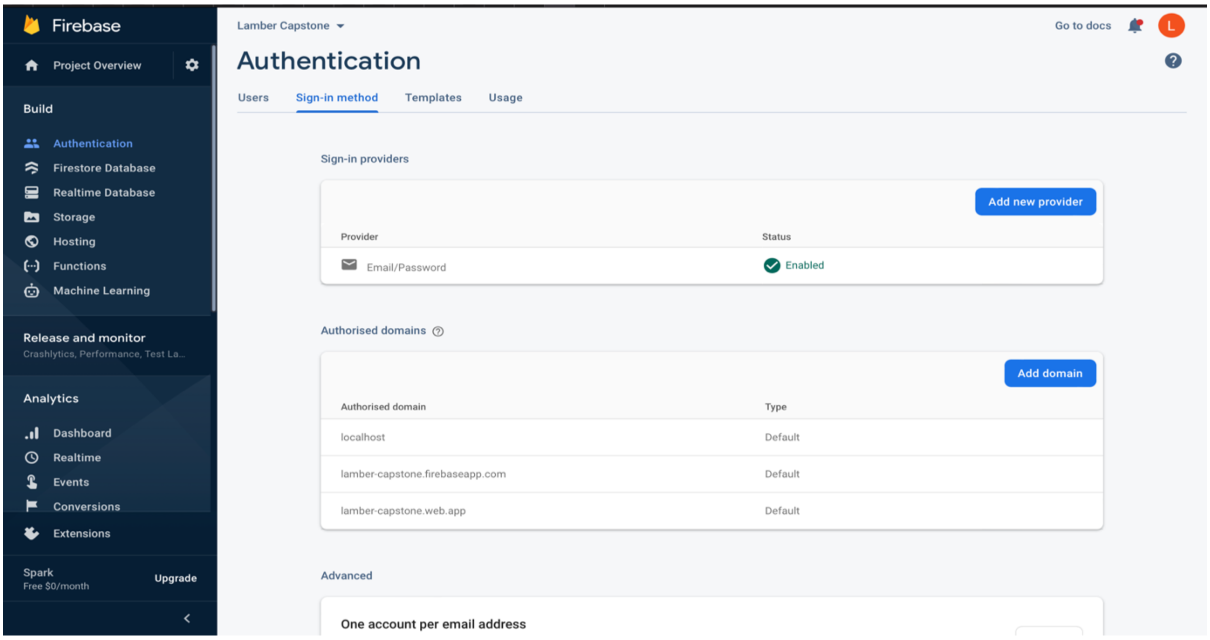


Figure 3.9 Firebase authentication page

3.3.4.2 Firebase Cloud Messaging

The cloud messaging component on Firebase is a cross-platform messaging solution that serves as a medium to send messages to all connected users [20]. Notifications to be sent on the application include updates on a request and cured first-aid content.

3.3.5 Paystack API

Paystack is an online payment platform that can be integrated into mobile and web applications. Integrating Paystack into an application is easy and straightforward. The API gives access to accepting payments, making transfers, and identifying customers [21].

CHAPTER 4: IMPLEMENTATION

4.1 Introduction

Lamber is an integrated software system with three major users, as stated in chapter 2. The system consists of three applications, two of which are mobile applications for the patients and the paramedics. The final application is a web application for the hospital administrators.

Patients make requests using either a general request, where the patient information is sent to all nearby hospitals or a specific request. A patient picks their hospital and sends a request to it with their data. The administrator then assigns a paramedic to the request, who then gets the necessary information and attends to the patient.

4.2 Key Functionalities

4.2.1 Patient Request Types

Patients using the mobile application can either make a general request or a specific request, as explained in the section above. The SOS request can be located on the homepage, as shown in Figure 4.1.

The process of the SOS request is described as follows. When a user clicks on the SOS button, the user's information and current location are sent as part of the request to all available ambulances within a radius of 10 kilometres (Figure 4.2). A loading page appears till a request is accepted (Figure 4.3). When an administrator agrees with a request, the page redirects to the main request page to indicate that the hospital has accepted its request (Figure 4.4).

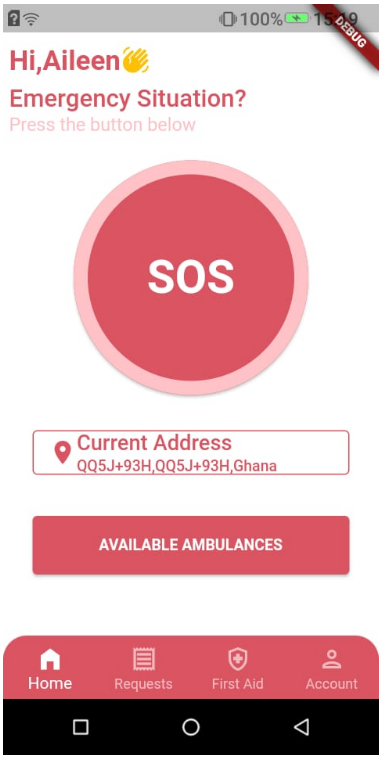


Figure 4.1 Home page of the mobile application.

```dart
Future<void> sosrequest(String street, String locat) async {
  final userid = FirebaseAuth.instance.currentUser?.uid;
  DatabaseReference ref = FirebaseDatabase.instance.ref("users/clients");
  Query query = ref.orderByKey().equalTo(userid);
  DataSnapshot event = await query.get();
  var username;
  var phone;
  Map<dynamic, dynamic> values = event.value as Map<dynamic, dynamic>;
  values.forEach((key, value) {
    username = value['FullName'].toString();
    phone = value['Phone'].toString();
  });
  final time = DateFormat("EEEEE MMM dd yyyy HH:mm:ss a").format(
      DateTime.now());
  final times = DateFormat("MMddyyyyHHmm").format(DateTime.now());
  final pick = DateFormat("HH:mm:ss a").format(DateTime.now());
  final requestid = username.toString().substring(0, 3).trim() + "Gen" + times.toString();
  DatabaseReference request = FirebaseDatabase.instance.ref(
      "requests/$requestid");
  try {
    await request.set({
      "Customer_Name": username,
      "Customer_Number": phone,
      "Customer_uid": userid,
      "Customer_location":locat,
      "Destination": street,
      "Pick_Up_Time": pick,
      "Request_DateTime": time,
      "Request_Type": "General",
      "Status": "Pending",
      "Request_id": requestid,
    });
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => LoaderPage(todo: requestid),
      ),  // MaterialPageRoute
```

Figure 4.2 Code snippet for sending SOS request data

The code snippet above shows the implementation for sending a patient's data using the general option. The main parameters required for the function are the location and the location coordinates of the patient. The user identification key is required to get important information such as the name and phone number of the patient to complete the request. The current date and time are retrieved to notify the hospitals when the request was made. Once the request has been processed, the user is taken to the loading page (Figure 4.3)

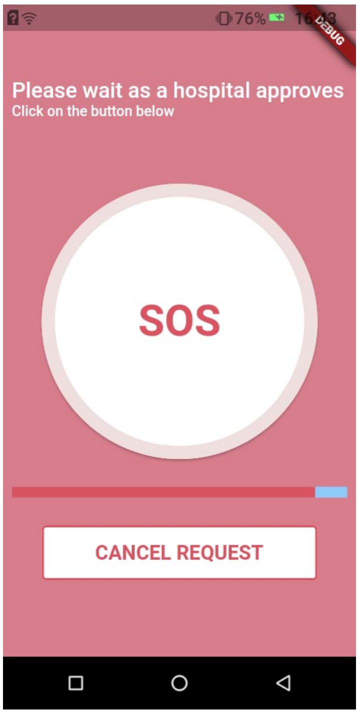


Figure 4.3 Loader page waiting for confirmation

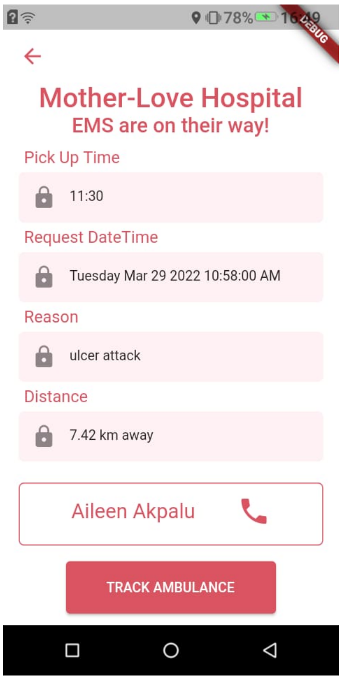


Figure 4.4 Confirmation page

The process for a specific request is described as follows. A user clicks on the available ambulances button on the home page and then it showed a list of ambulances within a 10-kilometre range (Figure 4.5).

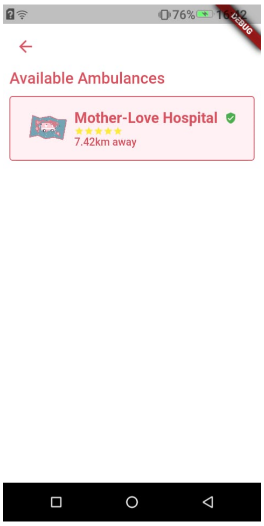


Figure 4.5 Page showing list of available ambulances

This range is a suggested response time of 8 minutes is required for all emergency response services [22]. A user can choose one of the available ambulances and then is taken to a booking page. For a specific request, the user is required to critical data such as the time they want the ambulance to arrive, a message to indicate what the issue is, and a voice note to explain the situation (Figure 4.6).

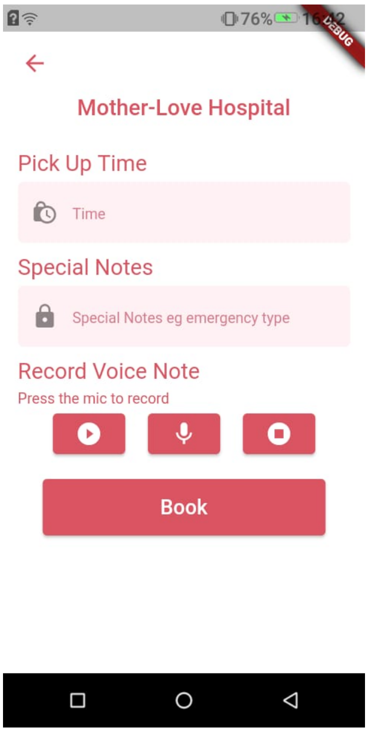
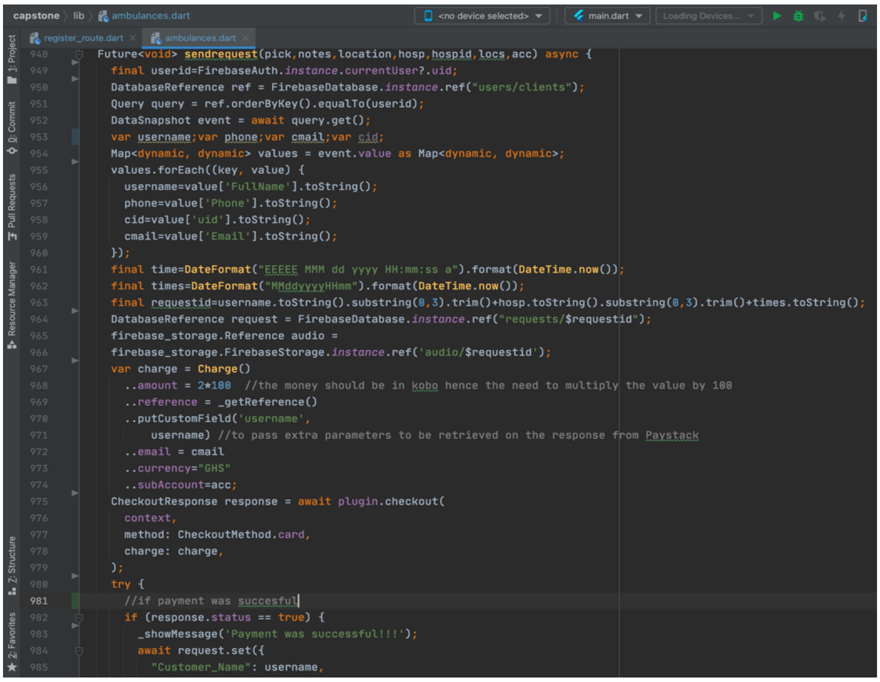


Figure 4.6 Booking page for selected hospital

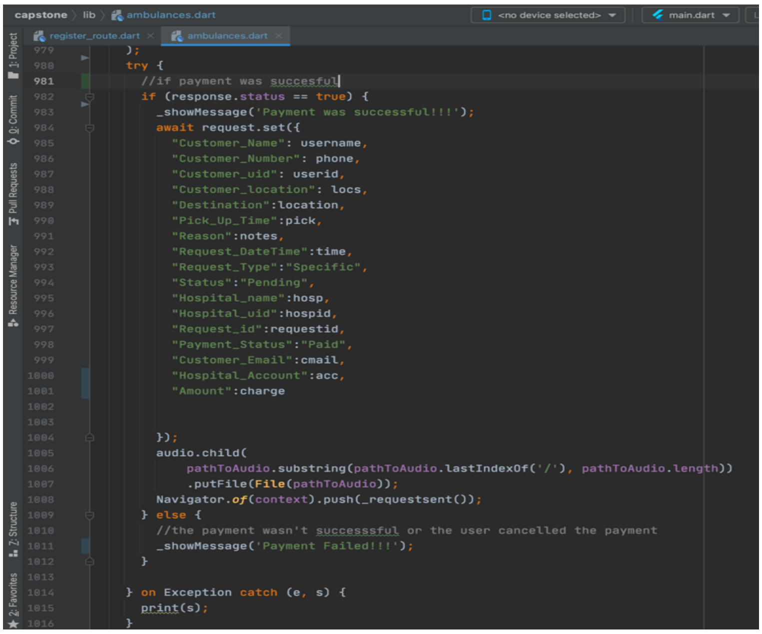
Then the hospital will proceed to accept the request and dispatch an ambulance. Figures 4.7 and 4.8 show the code implementation for processing a specific request.



```dart
Future<void> sendrequest(pick,notes,location,hosp,hospid,locs,acc) async {
  final userid=FirebaseAuth.instance.currentUser?.uid;
  DatabaseReference ref = FirebaseDatabase.instance.ref("users/clients");
  Query query = ref.orderByKey().equalTo(userid);
  DataSnapshot event = await query.get();
  var username;var phone;var cmail;var cid;
  Map<dynamic, dynamic> values = event.value as Map<dynamic, dynamic>;
  values.forEach((key, value) {
    username=value['FullName'].toString();
    phone=value['Phone'].toString();
    cid=value['uid'].toString();
    cmail=value['Email'].toString();
  });
  final time=DateFormat("EEEEE MMM dd yyyy HH:mm:ss a").format(DateTime.now());
  final times=DateFormat("MMddyyyyHHmm").format(DateTime.now());
  final requestid=username.toString().substring(0,3).trim()+hosp.toString().substring(0,3).trim()+times.toString();
  DatabaseReference request = FirebaseDatabase.instance.ref("requests/$requestid");
  firebase_storage.Reference audio =
  firebase_storage.FirebaseStorage.instance.ref('audio/$requestid');
  var charge = Charge()
    ..amount = 2*100  //the money should be in kobo hence the need to multiply the value by 100
    ..reference = _getReference()
    ..putCustomField('username',
        username) //to pass extra parameters to be retrieved on the response from Paystack
    ..email = cmail
    ..currency="GHS"
    ..subAccount=acc;
  CheckoutResponse response = await plugin.checkout(
    context,
    method: CheckoutMethod.card,
    charge: charge,
  );
  try {
    //if payment was succesful
    if (response.status == true) {
      _showMessage('Payment was successful!!!');
      await request.set({
        "Customer_Name": username,
```

Figure 4.7 Code snippet of getting the data for processing specific request

The code snippet above shows the information needed, such as the user's name, phone number, the current date and time and the amount to be charged before the request is processed.



```dart
  );
  try {
    //if payment was succesful
    if (response.status == true) {
      _showMessage('Payment was successful!!!');
      await request.set({
        "Customer_Name": username,
        "Customer_Number": phone,
        "Customer_uid": userid,
        "Customer_location": locs,
        "Destination":location,
        "Pick_Up_Time":pick,
        "Reason":notes,
        "Request_DateTime":time,
        "Request_Type":"Specific",
        "Status":"Pending",
        "Hospital_name":hosp,
        "Hospital_uid":hospid,
        "Request_id":requestid,
        "Payment_Status":"Paid",
        "Customer_Email":cmail,
        "Hospital_Account":acc,
        "Amount":charge


      });
      audio.child(
          pathToAudio.substring(pathToAudio.lastIndexOf('/'), pathToAudio.length))
          .putFile(File(pathToAudio));
      Navigator.of(context).push(_requestsent());
    } else {
      //the payment wasn't successsful or the user cancelled the payment
      _showMessage('Payment Failed!!!');
    }

  } on Exception catch (e, s) {
    print(s);
  }
```

Figure 4.8 Code snippet after payment is successful

The code snippet above shows the try-catch conditions based on the success of the payment. If payment is successful, the user's data will be set in the database and the audio recording will be saved in Firebase storage.

4.2.2 Request Tracking

Request tracking is done on both mobile and web applications. For the mobile application for patients, the user is shown the current location of the ambulance and is given regular updates based on the paramedic's actions (Figure 4.9). The code for implementing this functionality is shown below in Figure 4.10.

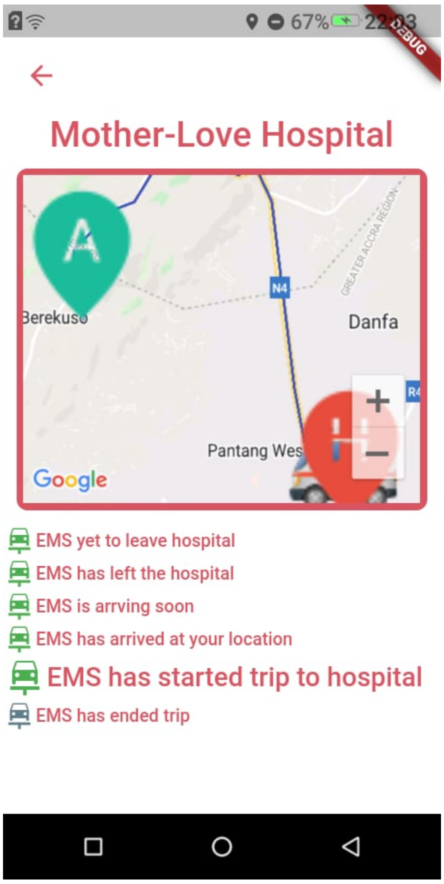


Figure 4.9 Tracking page for selected request

```
child: GoogleMapsWidget(
  apiKey: "AIzaSyCaG0u5EfOCiQTSWQyEia2HBbhl3wxcB-g",
  sourceLatLng: LatLng(hosplang, hosplong),
  destinationLatLng: LatLng(longf, langf),


  routeWidth: 2,
  sourceMarkerIconInfo: MarkerIconInfo(
  assetPath:"assets/images/hospital.png",
  ), // MarkerIconInfo
  destinationMarkerIconInfo: MarkerIconInfo(
  assetPath: "assets/images/home.png",
  ), // MarkerIconInfo
  driverMarkerIconInfo: MarkerIconInfo(
  assetPath: "assets/images/car.png",
  assetMarkerSize: Size.square(125),
  ), // MarkerIconInfo
  // mock stream
  driverCoordinatesStream: Stream.periodic(
  Duration(milliseconds: 500),
  (i) => LatLng(
    hosplang + i / 20000,
    hosplong - i / 20000,
  ), // LatLng
  ), // Stream.periodic
  sourceName: tod["Hospital_name"].toString(),
  driverName: tod["Personnel"].toString(),
  onTapDriverMarker: (currentLocation) {
  print("Driver is currently at $currentLocation");
  },
  totalTimeCallback: (time) => print(time),
  totalDistanceCallback: (distance) => print(distance),
  ), // GoogleMapsWidget
), // Center
))), // Padding, SizedBox, Container
```

Figure 4.10 Code implementation for the tracking in mobile application

For the web application, the administrator can track a request and get updates based on the actions of the paramedics (Figure 4.11).

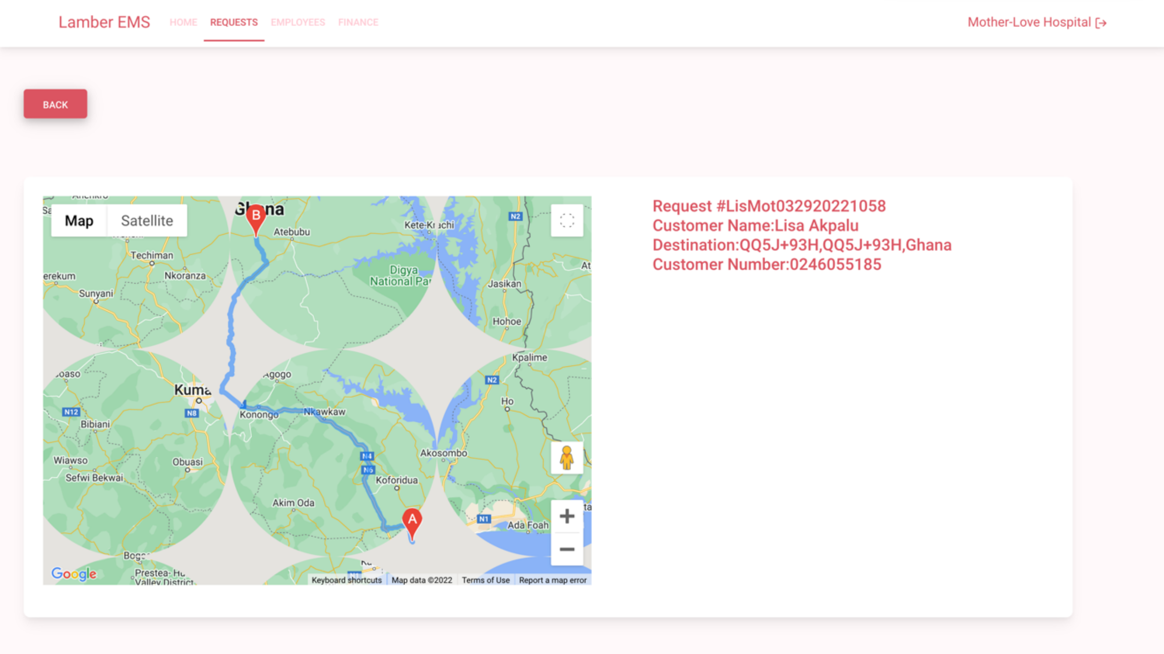


Figure 4.11 Tracking page for administrator

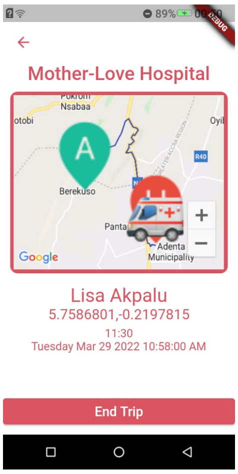


Figure 4.12 Request Processing page for paramedics

The state of a request is dependent on the paramedic as they are the ones handling the request. In figure 4.12 above, the paramedic updates the status of a trip when they click on the button at the end.

4.2.3 Administrator Request Process

The process for accepting or declining a request is described as follows. When a hospital receives a request, they can accept or decline it (figure 4.13).

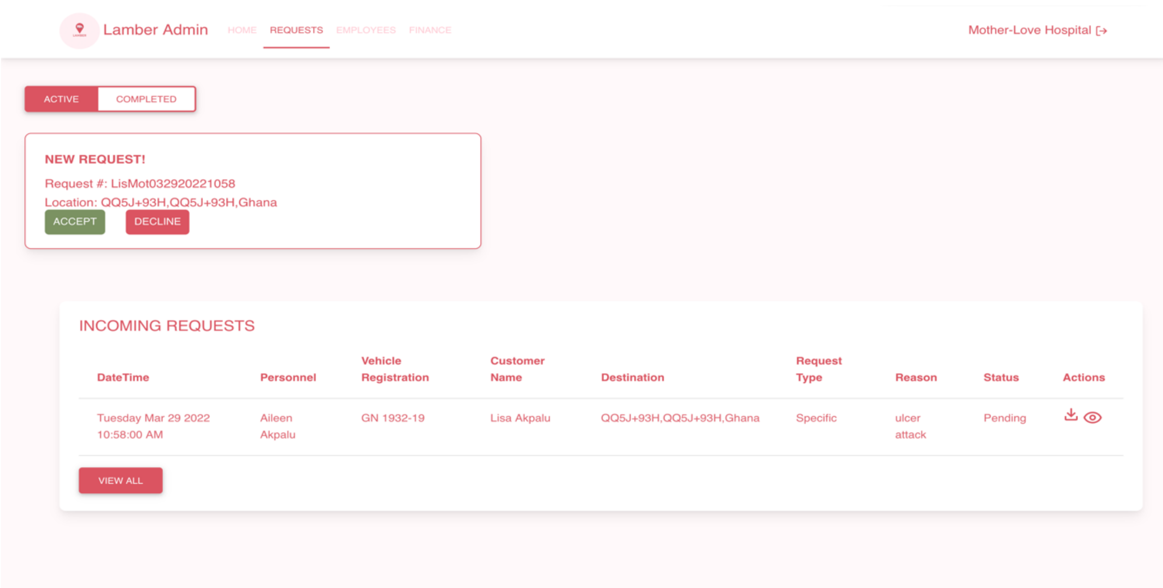


Figure 4.13 Administrator request page

A hospital can decline a request because the situation is not life-threatening to the affected person [23]. When the request is accepted, the administrator then selects an available paramedic and completes the process (Figure 4.14).

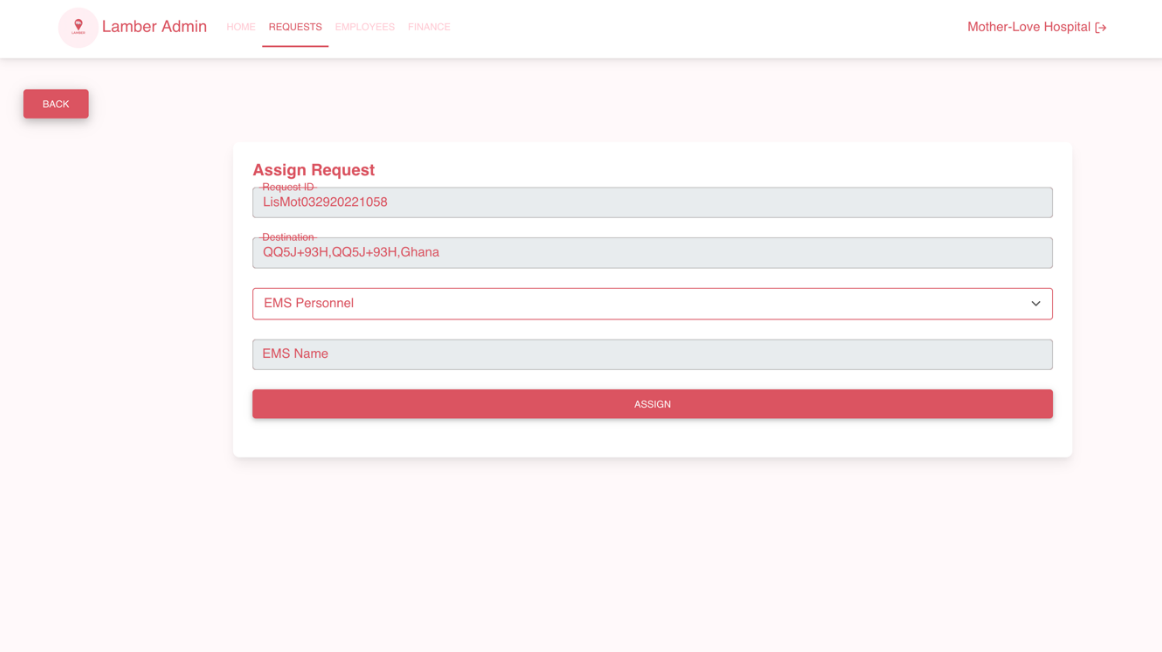


Figure 4.14 Page to assign paramedic to request

Figure 4.15 shows the processed data to be sent to the database. The information is then sent to the designated paramedic.

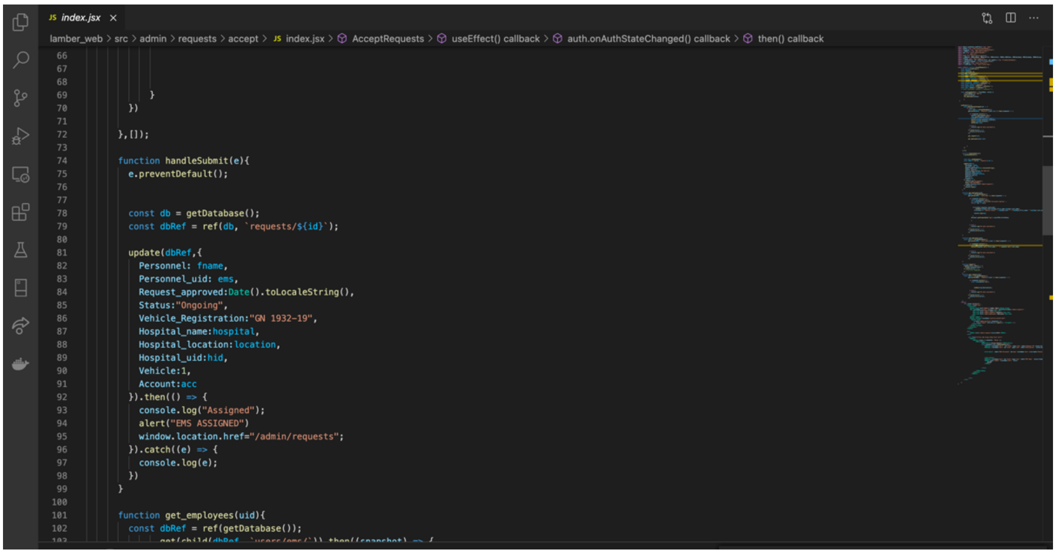


Figure 4.15 Code Snippet for assigning a paramedic to a request

4.2.4 Google Maps API

The implementation of the Google Maps API was used in both web and mobile applications.

For the mobile application, the API was used to get the user's current location and get the nearest ambulances. The implementation of this functionality is shown in Figure 4.16.

```
Future<Position> _getGeoLocationPosition() async {
  bool serviceEnabled;
  LocationPermission permission;
  serviceEnabled = await Geolocator.isLocationServiceEnabled();
  if (!serviceEnabled) {
    await Geolocator.openLocationSettings();
    return Future.error('Location services are disabled.');
  }

  permission = await Geolocator.checkPermission();
  if (permission == LocationPermission.denied) {
    permission = await Geolocator.requestPermission();
    if (permission == LocationPermission.denied) {
      return Future.error('Location permissions are denied');
    }
  }

  if (permission == LocationPermission.deniedForever) {
    return Future.error(
        'Location permissions are permanently denied, we cannot request permissions.');
  }
  return await Geolocator.getCurrentPosition(
      desiredAccuracy: LocationAccuracy.high);
}
```

Figure 4.16 Code implementation to get the user's current location

The code snippet above shows how the API gets the user's current location on the mobile applications. The Google maps has a feature called Geolocation API that receives the device's location based on the cell towers and Wi-Fi nodes the device can detect [24].

For the application to gain access to the user's location, the user would have to enable location services for the application. After enabling the permission, the function returns the device's geographical location.

Chapter 5: Testing and Results

5.1 Introduction

Testing the system to validate if the essential functional requirements are functioning correctly. This section discusses how testing was conducted for these components. The testing has been sectioned into three major categories to address testing from different angles.

5.2 Component Testing

This section focuses on the testing of individual functionalities that constitute the system.

5.2.1 User Registration

5.2.1.1 Administrator

For the administrator system, the user first registers the clinic using the relevant details, such as the location on the system. After registration is successful, a verification mail is sent before they can progress to sign-in (Figure 5.1). Firebase has an in-built email verification feature that generates a link for each user. When verification is successful, they can now get access to requests and add employees to the system. They are to add bank details before officially processing any requests, as payment is a critical process for request processing. An authenticated Gmail account was used to register on the system, and a verification email, as shown in Figure 5.1, was sent.

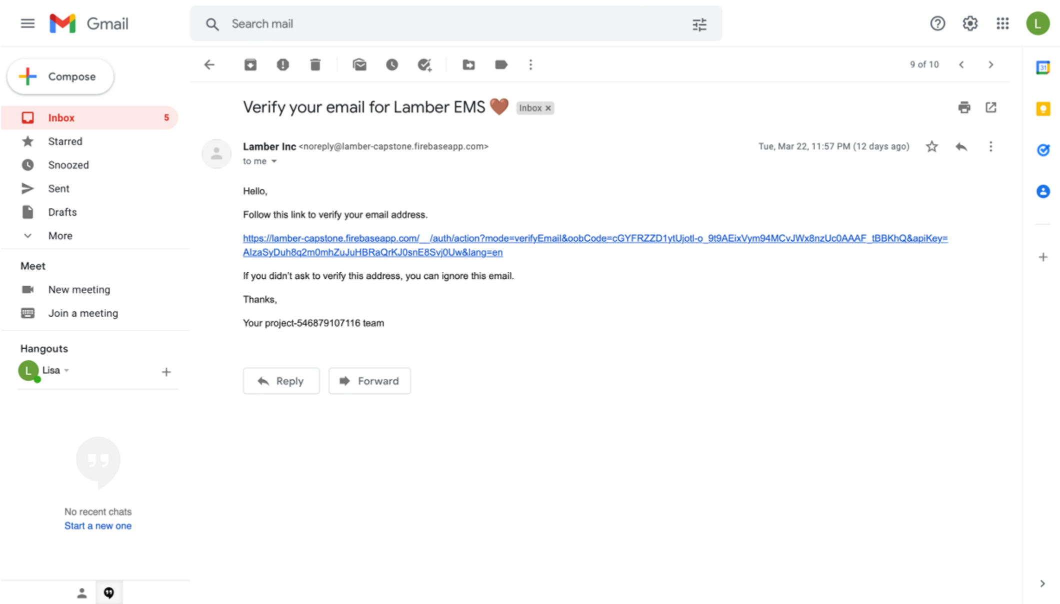


Figure 5.1 Verification mail sent to the user's email address

5.2.1.2 Patient

The user registers using relevant details for the patient system, as shown in Figure 5.2. If any field is empty, it will not process the data for registration, as shown in Figure 5.2. After successful registration, a verification mail is sent before the user can log in successfully. An authenticated Gmail account was used to register on the system, and a verification email was sent.

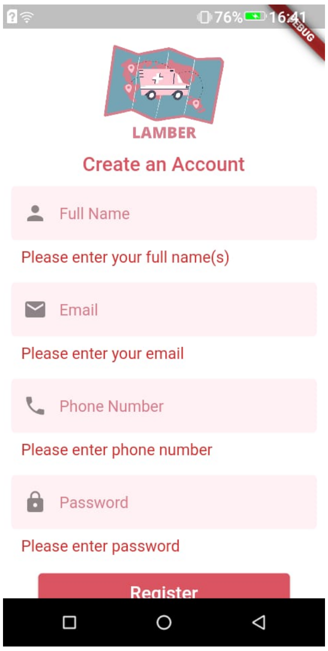


Figure 5.2 Screenshot of patient registration

5.2.1.3 Paramedic

For the paramedics, after the administrator has created their profile, they verify the mail with a new password to create an authenticated account (Figure 5.3). A verification mail is sent to the mail after completion.

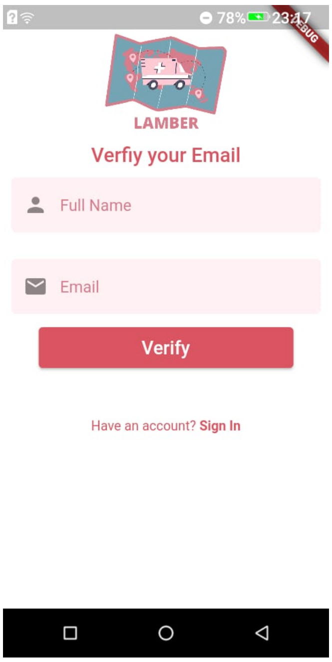


Figure 5.3 Screenshot of paramedic verification

5.2.2 Booking Requests

Booking request is the specific route that patients can opt for by choosing their ambulance. As shown in Figure 5.4, the request will not be completed if a field is empty. After successfully processing the data, a payment option will appear, and after payment confirmation, the request is fully processed and sent (Figure 5.5). If the card details are incorrect, it will prompt the user several times till the correct details are used.

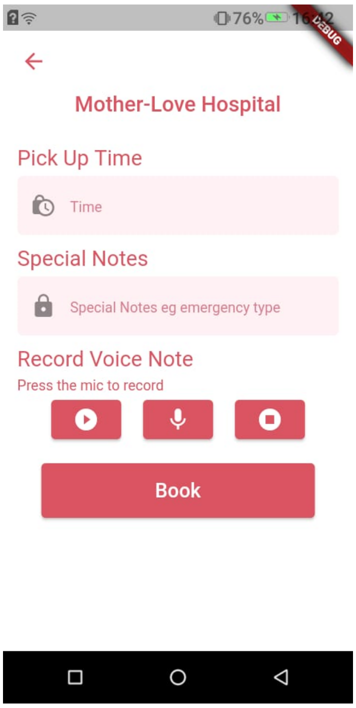


Figure 5.4 Screenshot of the Booking form.

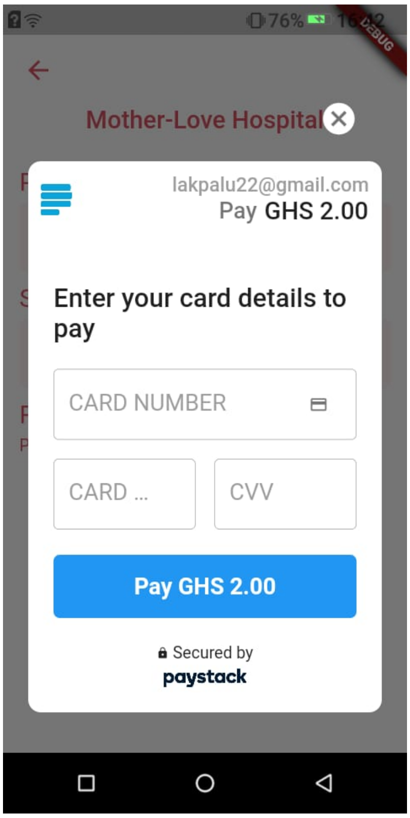


Figure 5.5 Screenshot of payment

5.2.3 SOS Requests

SOS requests are the general route that patients can opt for to send a request to all nearby clinics. When an administrator accepts the request, the user is taken to a confirmation page (Figure 5.6). Payment for this type of request is done after the request is processed.

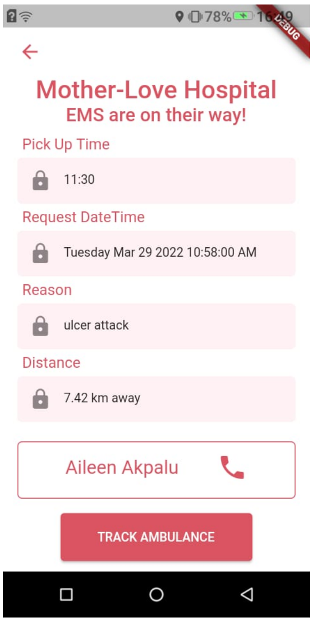


Figure 5.6 Screenshot of SOS request confirmation

5.3 System-Level Testing

System-level testing seeks to address the defects in both individual components and the system.

5.3.1 Request Processing

Request processing is a significant functionality of the system, and each sub-system goes through a process to authenticate a request. When a patient makes a request and payment are successful, the request is processed and sent to the designated hospital. The administrator is notified and then accepts the request and assigns an available paramedic to the request (Figure 5.7). The paramedic then sees the request and then starts the request. The user then can begin tracking the ambulance till it reaches their location.

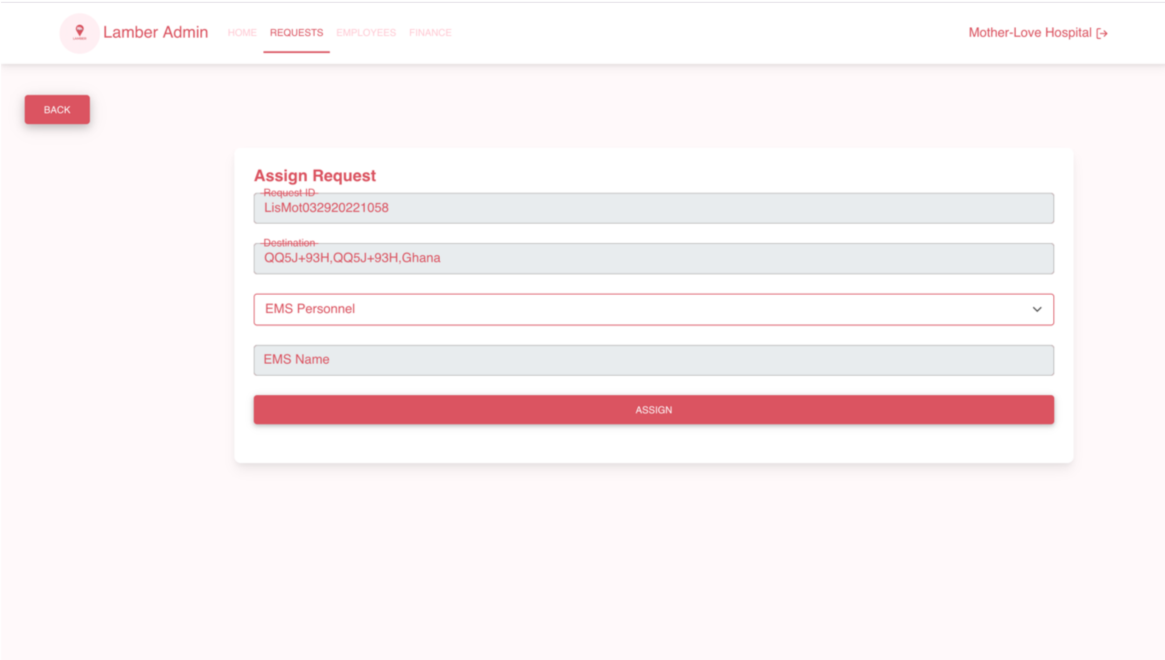


Figure 5.7 Screenshot showing the assignment of personnel

5.3.2 Request Tracking

The request tracking process is implemented in all three systems as such, each user connected to a request gets constant updates. There was one challenge, regarding the button state change. When a paramedic clicks a button (Figure 5.8) to update the administrator and the patient on their location, the button state does not change until you reload the page. The change has been reflected in the database, but the page must be reloaded before it shows. This has however been resolved.

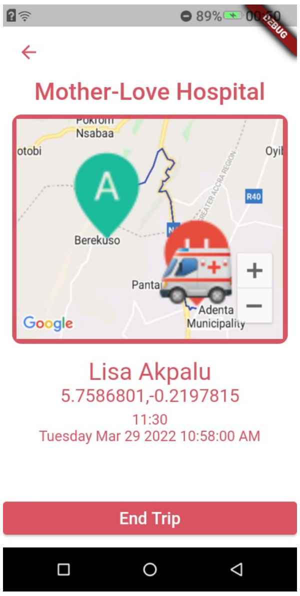


Figure 5.8 Screenshot of paramedic tracking page

5.4 User Testing

User testing to be conducted with real-life users to get relevant feedback and validate how effective the system is.

Two individuals tested each of the three applications of Lamber, and this was done to validate how integrated the system was and if it addresses the problem it is trying to solve.

User 1, Eben, is a 21-year-old male student who does not utilise the ambulance service but transports his family to the hospital when there is an emergency. He tested out the patient and paramedic mobile applications.

User 2, Yaa, is a 50-year-old female doctor who runs a clinic and has a private ambulance. She tested out the administrator system and patient system.

5.4.1 User Testing on Patient system

Eben tested the patient system and had relevant feedback. At the time of testing with Eben, version 1.0 of the systems was done, and version 1.2 was already in the works. Eben asked if the voice note component had a delete feature should a user make a mistake. The voice note was built such that if a user wants to re-record a voice note, the user can tap on the microphone button again and record. Eben had no problems when signing in and making requests using both types.

5.4.2 User Testing on Paramedic system

Eben tested the paramedic system and had a few suggestions regarding the tracking of requests. The defect with that page, as stated, was the button state-changing whenever the paramedic makes a change. Another relevant feedback was to explain to paramedics why they must verify their mail before proceeding.

5.4.3 User Testing on Administrator system

Yaa tested the administrator system and had relevant feedback. She suggested a page that shows the financial transactions for the administrator to track. Aside from that, all functionalities implemented were usable by her.

5.5 Analysis of Results

The system is a location-based system, and thus, if location services are not enabled on the patient and paramedic applications, there will not be any functioning system. Each section of the testing shows that the system is functioning as it should. The feedback gotten from the user testing was taken into consideration, and relevant changes were made.

Chapter 6 Conclusions and Recommendations

6.1 Conclusion

The project sought to provide a better and more efficient alternative to the traditional calls for making requests for emergency response in Ghana. The scope was narrowed down to private clinics with ambulances. The problems that the system aimed to solve were:

1. slow response time
2. inaccurate street names
3. prank calls

The issue of slow response time was tackled by having an integrated system for easy data flow between all applications and notifying the administrator anytime there is a new request.

The issue of inaccurate street names was resolved using Google Maps API to obtain the geolocation and coordinates of the patient and the paramedic. The API was used to aid with map navigation for the paramedic.

The issue of prank calls was resolved by implementing a payment action before a patient completes a booking for an ambulance. If a user decides to cancel a request, the money will not be refunded. In doing this, a user is losing their money, and thus, users are deterred from making prank requests.

The project aims to improve the response time of the emergency response whilst getting accurate street coordinates for easy navigation. Users can resort to making requests on the mobile application for patients instead of using traditional calls where navigating to the user can be relatively difficult. Private clinics can also monitor and manage their requests efficiently as the paramedics have their applications solely for tracking requests and navigation, and the administrators can make manage and assign requests when they receive them.

The project is focused on having a centralised system where all the three sub-systems work in tandem.

6.2 Challenges

During this project, some limitations arose as well as limitations with the system. The general limitation is the time constraint as it affects the implementation of certain functionalities and the testing process. The implementation aspect of the project was completed over four months, and this was not enough time, especially as three different applications were built. There was not enough time to get interviews with more private clinics to get diversified opinions and feedback on the system regarding the testing process.

Another challenge is regarding internet connectivity. The system relies on the internet to function correctly and thus, if there is no internet connectivity, the users cannot proceed with certain functionalities such as posting requests. The application, as stated earlier, relies on location and thus, if the location is not enabled, processes such as posting requests and tracking cannot function.

Another challenge is that although the aim is to improve response time, human beings will still receive and accept requests. Thus, if the administrator is not focused on the application and accepts the request after a long period, then the whole purpose is defeated. Measures such as notifying the administrator when there is a new request were put in place to avoid such human errors.

6.3 Future Works

The section discusses additional features that can be added to the system to make it more efficient.

6.3.1 Recommendation system for first aid content

The system for the patient has a first aid material section that a user can utilise to gain more knowledge. However, to make the application more interactive, a recommendation system using data from the patient's requests can help inform the system to suggest first aid content for the user's benefit. For example, the system can pick up data that a user is consistently making requests for a fracture, and then the system can recommend the first aid content for fracture so that the user can learn.

6.3.2 Recommendation system for ambulances

A machine learning model that will use relevant data such as the ratings of clinics and the user's location to suggest the highest-rated ambulance specific to the user's location. This will make the use of the application much more accessible.

6.3.3 In-built wallet

An in-built wallet for the patient to track all transactions and load money onto the system will make processing requests easier. Instead of constantly inputting your details when the need is, having an in-built wallet will make transaction processing easy.

6.4.4 IOS Version

A complete IOS version for the mobile applications will be developed.

6.4.5 Insurance Scheme

An inclusion of the National Health Insurance Schemes into the payment method for people unable to afford the service.

REFERENCES

[1] Sasu, D. 2021. Ghana: governmental health spending as a share of GDP | Statista. *Statista*. https://www.statista.com/statistics/1124307/ghana-domestic-general-government-health-expenditure-as-share-of-gdp/.

[2] Zakariah, A., Stewart, B., Boateng, E., Achena, C., Tansley, G. and Mock, C. 2016. The Birth and Growth of the National Ambulance Service in Ghana. *Prehospital and Disaster Medicine 32*, 1, 83-93.

[3] Ministry of Health. 2021. President Akufo-Addo Commissions 307 Ambulances - Ministry of Health. *Ministry Of Health*. https://www.moh.gov.gh/president-akufo-addo-commissions-307-ambulances/.

[4] American Medical Response. 2015. *Amr.net*. https://www.amr.net/resources/knowledge-center/201506-3836-hbems_white-paper_4pgfinal.pdf.

[5] Adamtey, R., Frimpong, J. and Dinye, R. 2015. An Analysis of Emergency Healthcare Delivery in Ghana: Lessons from Ambulance and Emergency Services in Bibiani Anhwiaso Bekwai District. *Ghana Journal of Development Studies 12*, 1-2, 71.

[6] Koranteng, M. 2021. No-show of ambulance to blame for death in motorway accident – Eyewitnesses - MyJoyOnline.com. *MyJoyOnline.com*. https://www.myjoyonline.com/no-show-of-ambulance-to-blame-for-death-in-motorway-accident-eyewitnesses/.

[7] Ansah, M. and Longdon, B. 2021. Every house to get digital address by 2022 – Bawumia. *Citinewsroom - Comprehensive News in Ghana*. https://citinewsroom.com/2021/07/every-house-to-get-digital-address-by-2022-bawumia/.

[8] Owusu, V. 2019. Verifly: A Drone Supported First-Aid System for Ashesi University.

[9] Mboya, M. 2018. Locating the Nearest Pharmacy with the Desired Medicine.

[10] Shen, S. 2019. Emergency System for Seniors: A Location-based Rescue Service App.

[11] Khalique, V., Shaikh, S., Dass, M., Shah, S. and Zaib, M. 2017. Automatic Ambulance Dispatch System via One-Click Smartphone Application. *Indian Journal of Science and Technology 10*, 36, 1-9.

[12] Agile Model (Software Engineering) - javatpoint. *www.javatpoint.com*. https://www.javatpoint.com/software-engineering-agile-model.

[13] Dziuba, A. Top 8 Flutter Advantages. Relevant Software. https://relevant.software/blog/top-8-flutter-advantages-and-why-you-should-try-flutter-on-your-next-project/.

[14] Surve, S. 2021. Why You Should Use React.js For Web Development. freeCodeCamp.org. https://www.freecodecamp.org/news/why-use-react-for-web-development/.

[15] Google Maps API. 2019. Static.googleusercontent.com. https://static.googleusercontent.com/media/maps.google.com/en//help/maps/casestudies/maps-api-web.pdf.

[16] What is Firebase? Secrets unlocked (pros, cons, pricing, etc). Back4App Blog. https://blog.back4app.com/firebase/.

[17] NoSQL vs SQL — Which Database Type is Better for Big Data Applications. Analytics India Magazine. https://analyticsindiamag.com/nosql-vs-sql-database-type-better-big-data-applications/.

[18] Firebase Realtime Database | Firebase Documentation. Firebase. https://firebase.google.com/docs/database.

[19] Firebase Authentication | Simple, no-cost multi-platform sign-in. Firebase. https://firebase.google.com/products/auth?gclid=EAIaIQobChMI6ciT8JyA9gIVgqZ3Ch0qHQEPEAAYASAAEgIhbPD_BwE&gclsrc=aw.ds.

[20] Firebase Cloud Messaging | Firebase Documentation. Firebase. https://firebase.google.com/docs/cloud-messaging.

[21] Home. *Paystack.com*. https://paystack.com/docs/.

[22] The National Highway Traffic Safety Adminstration, National EMS Scope Of Practice Model. (2007). Washington Available: https://www.ems.gov/education/EMSScope.pdf.

[23] When-and when not-to, call an ambulance. *Emergencyphysicians.org*. https://www.emergencyphysicians.org/article/er101/when---and-when-not---to-call-an-ambulance.

[24] Overview | Geolocation API | Google Developers. *Google Developers*. https://developers.google.com/maps/documentation/geolocation/overview.

APPENDICES

A-Questionnaire

1. Have you ever called the National Ambulance Service or a private ambulance?
2. If yes, how long did it take to get to your location?
3. If yes, what was your immediate reaction to the emergency?
4. If no, the Mode of transportation to the hospital in case of emergency
5. Without any pressing emergency, how did you transport to the hospital
6. How effective do you think the NAS is?
7. Have you had or heard about any bad experiences with an ambulance service?
8. Do you think having an application that allows multiple ambulances to be connected to a patient when they request for one?
9. Should there be an application that will help the NAS and the private hospitals with one manage their ambulance service?
10. Should there be an application that will allow users to get hands-on information in case of an emergency and a waiting period for the ambulance?
11. Do you think that an application still allows users to get help even when their designated ambulance is busy?

B-Interviews

1. How many times do you get patients using the ambulance service?
2. Have you treated a patient who was in critical condition and used the ambulance service?
3. Have you treated a patient in critical condition and did not use the ambulance service?
4. How many times do you get a request for the ambulance in a day/month?
5. How many times do you get a prank call in a day/month?
6. Has the system ever been overwhelmed with requests?
7. How many EMS personnel do you have?
8. Do you think having an application software will help facilitate EMS requests?

C-Project Management Gantt Chart

I. Table with data

Task Name	Start (Da	End (Date	Duration (Days)
Gather Data	20/08/2021	25/08/2021	5
Interview with Medical Pra	22/08/2021	22/08/2021	0
Functional and Non-functi	04/09/2021	06/09/2021	2
Ideation Phase I and II	14/09/2021	23/10/2021	39
Prototyping Phase I for mob	02/11/2021	05/11/2021	3
Prototyping Phase II for mo	12/11/2021	18/11/2021	6
Test Protoypes	14/12/2021	17/12/2021	3
Development of frontend a	12/01/2022	20/01/2022	8
Database system design	18/01/2022	20/01/2022	2
Phase I Mobile application(	14/01/2022	28/01/2022	14
Phase II Mobile application(	30/01/2022	10/02/2022	11
Phase III Mobile application	14/02/2022	15/03/2022	29
Phase IV Mobile applicatior	20/03/2022	27/03/2022	7
Phase V Mobile application	04/04/2022	30/04/2022	26
Chapter Writing	15/09/2021	04/04/2022	201

II. Chart