

EMERGENCY AMBULANCE BOOING

A MINOR PROJECT REPORT

Submitted by

Y. VENKATA KASHYAPI [RA2112703010016]
S. SREENIVASULA REDDY [RA2112701010006]
Y.UMA VENKAT REDDY [RA2112701010019]

Under the Guidance of

Dr. R. Lakshminarayanan

Assistant Professor, Department of Networking and Communications

in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING

with specialization in CLOUD COMPUTING



DEPARTMENT OF NETWORKING AND

COMMUNICATIONS

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR- 603 203 NOVEMBER

2022



SRM INSTITUTE OF SCIENCE AND TECHNOLOGY KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that this B. Tech project report titled **“EMERGENCY AMBULANCE BOOKING ”** is the bonafide work of **Ms. Y. VENKATA KASHYAPI , Mr. S.SREENIVASULA REDDY and Y.UMA VENKAT REDDY**, who carried out the project work under our supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

SIGNATURE

SIGNATURE

**DR. R. LAKSHMINARAYANAN
SUPERVISOR**

Assistant Professor
Department of Networking &
Communications

**DR. ANNAPURANI PANAIYAPPAN. K
HEAD OF THE DEPARTMENT**

Professor
Department of Networking
& Communications

Signature of the Internal Examiner

Signature of the External Examiner

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr. C.Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr. T.V.Gopal**, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We are incredibly grateful to our Head of the Department, **Dr. K.Annapurani Panaiyappan** Professor, Department of Networking and Communications, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We register our immeasurable thanks to our Faculty Advisors, **Dr. Manikandan and Dr. Meenakshi**, Department of Network and Communication, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my guide, **Dr. R. Lakshminarayanan**, Assistant professor, Department of Networking and Communications, SRM Institute of Science and Technology, for providing me with an opportunity to pursue my project under her mentorship. She provided me with the freedom and support to explore the research topics of my interest. Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank the Networking and Communications Department staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, we would like to thank our parents, family members, and friends for their unconditional love, constant support, and encouragement.

Y.VENKATA KASHYAPI -RA2112703010016

S.SREENIVASULA REDDY -RA2112701010006

Y.UMA VENKAT REDDY -RA2112701010019

Department of Networking and Communications
SRM Institute of Science & Technology
Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Degree/ Course : M.Tech Integrated AI and Cyber Security.

Student Name : Y. KASHYAPI, S.SREENIVASULA REDDY and Y
.UMA VENKAT REDDY.

Registration Number : RA2112703010016, RA2112701010006, and 0019.

Title of Work : Emergency Ambulance Booking App.

We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present.
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my own work, except where indicated by referring, and that I have followed the good academic practices noted above.

If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

ABSTRACT

As in India, an individual passes on each tick of the clock along these lines, we have proposed an application that will give a crisis wellbeing reaction to the patient. The fundamental motivation behind this undertaking will fill the hole between the patient and emergency vehicle reaction time. Ambulances are a fundamental piece of crisis clinical benefits. For the most part, patients have a limited scope of rescue vehicle contacts, hence at whatever point in a crisis, they track down trouble. With this venture, it is recommended that the application would empower the patient to book a ride to the medical clinic. The patient can transfer their ongoing area as well as their objective area into this site.

The framework would then show the close-by accessible ambulances and the patient can pick its suitable rides by looking at the citations and distance of each ride over a district. Then again, the emergency vehicle driver would become a brief about the booking made by the patient. The driver gets an affirmation email from the client who gave the alarm. The rescue vehicle driver needs to affirm the booking or reject the solicitation made by once the emergency vehicle isn't accessible the driver can dismiss the booking. At the point when the driver acknowledges the solicitation the application will direct the driver towards the objective through the guide interface transferred by the client. The administrator would get all the focal data and get all the data of clients and rescue vehicle drivers day to day report, administrator would control the request and calling functionalities.

TABLES OF CONTENTS

ACKNOWLEDGEMENT	iii
ABSTRACT	v
CHAPTER 1	8
1. INTRODUCTION	8
1.1 OVERVIEW	8
1.2 MOTIVATION	8
1.3 PROBLEM STATEMENT	8
1.4 FUTURE SCOPE	9
CHAPTER 2	10
2. LITERATURE SURVEY	10
2.1 RESEARCH OBJECTIVES	10
2.2 REASEARCH FINDINGS	11
CHAPTER 3	12
3.EXISTING SYSTEMS	12
3.1 REQUIREMENT ANALYSIS	12
3.2 USE CASE DIAGRAMS	13
3.3 BLOCK DIAGRAM	14
3.4 SEQUENCE DIAGRAM	15
CHAPTER 4	16
4.MODULES AND ANALYSIS	16
4.1 CLIENT MODULE	16
4.2 DRIVER MODULE	16
4.3 ADMIN MODULE	17
CHAPTER 5	18
5.1 ARCHITECTURE	18
CHAPTER 6	19
6.1 ALGORITHM	19

CHAPTER 7	21
RESULTS AND DISCUSSIONS	21
7.1 PROGRAM CODE	21
7.2 SCREENSHOTS	34
 CHAPTER 8	 35
8.1 REFERENCES	35
8.2 RESULTS	36

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW.

Ambulances play a major role in our day to day life . We don't know what kind of health emergency may occur at any time. Ambulances play the major part in transporting the patient from that particular place to the hospitals. Emergency ambulance booking helps in last minute ambulance bookings. In times of emergency, the user will simply specify the pick-up location and destination, and the system will book the closest ambulance and hospitals on their behalf. A message will be sent to the ambulance driver after the booking has been confirmed. The User will be able to book an ambulance in advance according to the size of the ambulance and selected hospital, or the user can also book an ambulance for emergency regardless of its size and a random hospital will be allocated.

1.2 MOTIVATION

They can provide the medical aid that is the need of the hour. The backbone of any ambulance service is the medical professionals who treat the patient first. They ensure to keep the vitals stable until they reach the hospital, and the doctors proceed with the treatment.

1.3 PROBLEM STATEMENT.

Ambulance plays a very crucial role when an accident occurs on the road network or in case of any medical emergency and the need arises to save a human life. Manual booking of an ambulance at times of emergency can take away precious time as it is a time-consuming process. Furthermore, the delay caused due to the heavy traffic congestion in between the pickup spot and the hospital facility may increase the risk of death for the victim. The system proposed here will help the users book an ambulance easily in an instant. Once booked the ambulance operator will receive a notification for confirmation of the booking. The Ambulance driver can view the pick-up and drop location on Goggle Maps. The users will receive the contact details of the driver. The Hospitals

can also view the booking history. This is how this Ambulance Booking App will act as a life savior in times of medical emergency.

1.4 FUTURE SCOPE

An advantage of the application is that there is range for future advancements and modifications to improve and refine as and when required.

CHAPTER 2

LITERATURE SURVEY

2.1 RESEARCH OBJECTIVES

- You can locate the nearest available ambulance and request the same.
- Instantly get the information & contact details of the driver.
- Live track the ambulance with an estimated time to reach the hospital.
- Offer all sorts of first aid inside the ambulance so that the patient becomes stable till he reaches the hospital.
- The main objective of this project is to reduce the time of ambulance delay and to provide health services quickly.

2.2 RESEARCH FINDINGS

Author	Year	Approach	Description
Shivali Walvekar and Kinjal More	2016	GPS based tracking and health parameter detection.	It displays the current location of ambulances and patient's health parameter on the LCD display and sends that information to the hospital
B.Janani Saradha,G.Vijayshri T.Subha	2017	RFID & cloud approach	The RFID tag helps in controlling the flow of traffic by changing the traffic lights
Chennakesava Reddy Kamireddy, Bingisateesh, Keshavamurthy	2016	Density-based clustering	It groups the major accidental areas into clusters and assign an ambulance to each of them ,thereby reducing the average distance travelled by the ambulance
CS Vikas and Ashok Immanuel	2017	Restful API	It's a web based application that shows the nearest ambulances, clinic and pharmacies to the user.
Bhandari Prachi, Dalvi Kasturi and Chopade Priyanka	2014	Intelligent accident detection and RF communication	The sensors placed in the vehicles helps in automatic accident detection and inform it the ambulance

CHAPTER 3

EXISTING SYSTEMS

India's biggest app based ambulance network is built to deliver the best in emergency care to people across the nation through its ambulance services that is not only limited to emergencies, but can be used by a patient to schedule check-ups & appointments or simply to get home after being discharged from hospital. The mobile app connects users to emergency facilities by providing timely ambulance service when required. With just one click or a call, a patient can have our associated ambulances at the location of their choice.

Ambulance booking services is on a journey to change this paradigm where emergency ambulance services take a back seat.

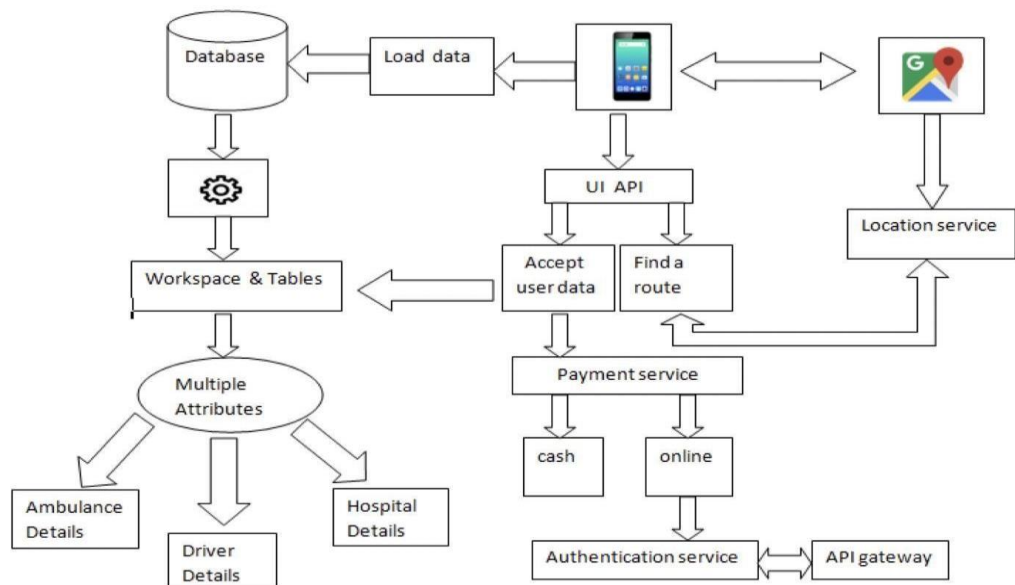
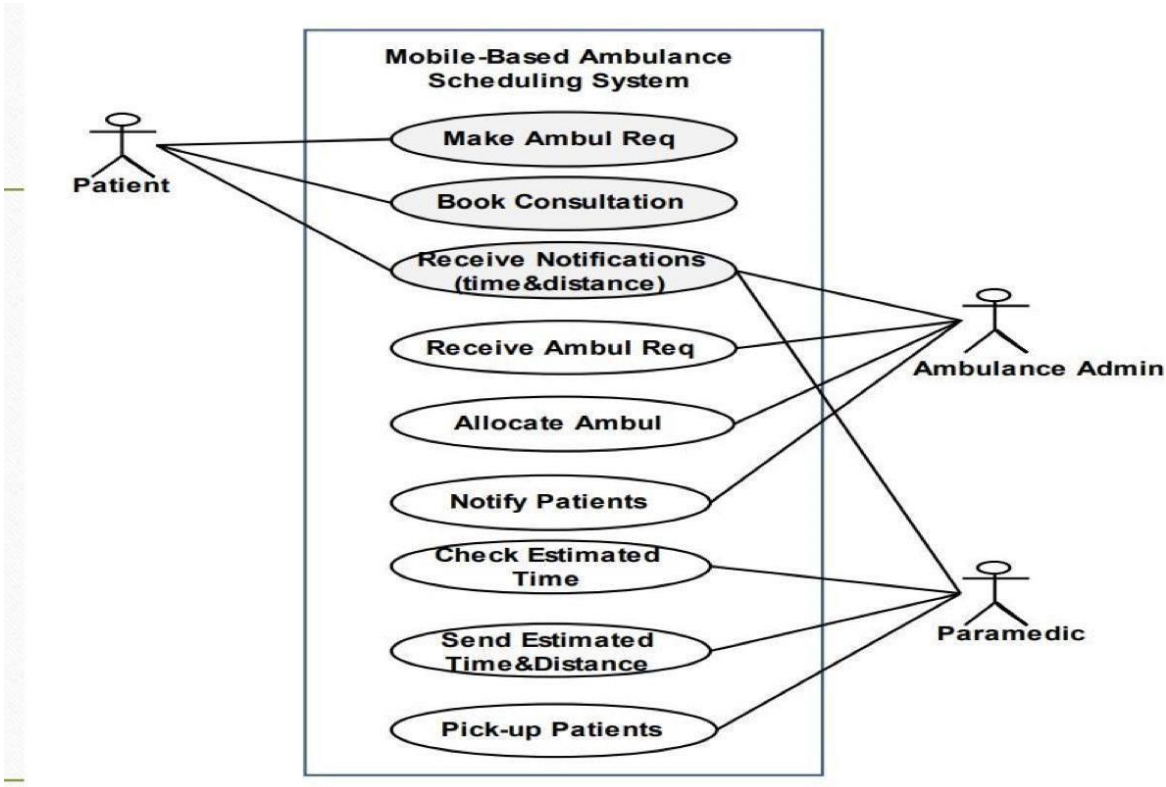
This medical emergency platform will help the citizens to book an ambulance using a mobile app and track the vehicle once it is assigned and predict the time when it will reach the desired venue. The fare paid for the ambulance will be predictable and the user will have a very seamless experience at the time when his or her mind is in turmoil due to the emergency happening in the household. Emergency ambulance booking is a start-up by 2 individuals, one from technology background and another one from the advertising/digital media background.

Our strength lies in the fact that we put more care into all aspects related to health, first through getting the ambulance to the patient on time and for the right price and then ensuring that doctors, nurses, medical equipment will be made available to the people in a seamless manner. We have set ourselves on a mission to create healthy tomorrows by putting people first.

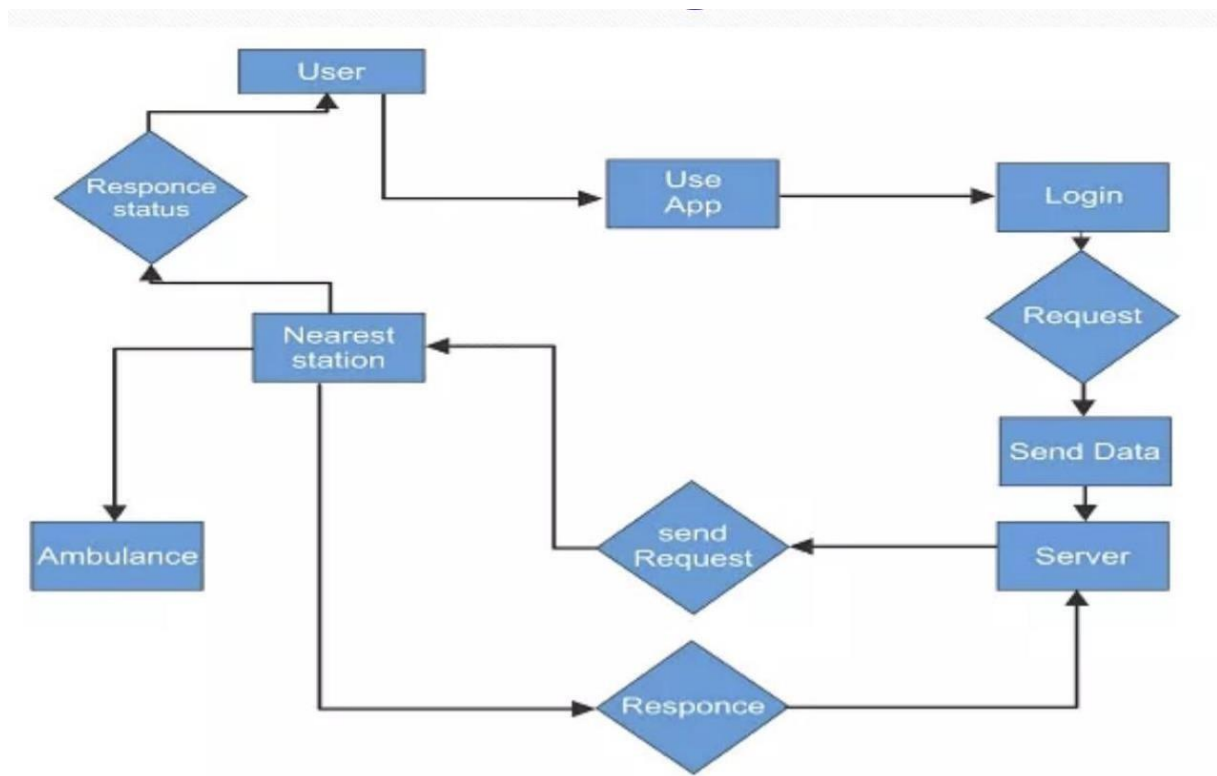
3.1 REQUIREMENT ANALYSIS

The project has a software section. The requirements for the software that we have covered in this project is java.

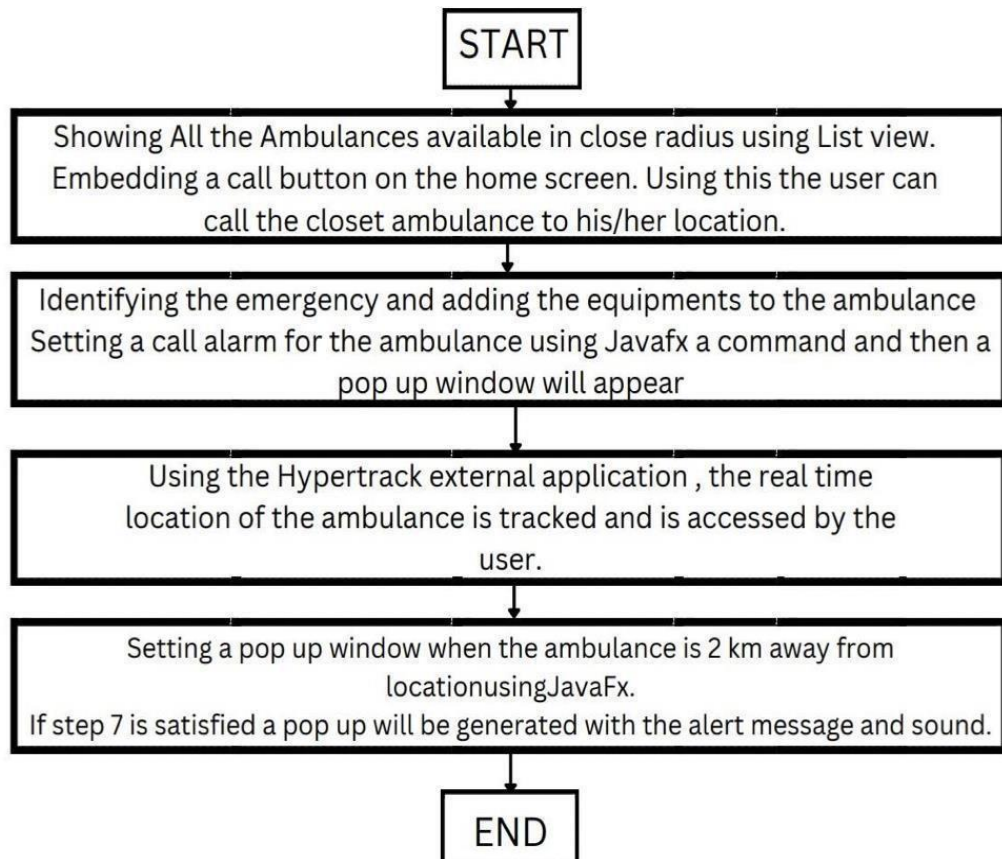
3.2 USE CASE DIAGRAM



3.3 BLOCK DIAGRAM



3.4 SEQUENCE DIAGRAM



CHAPTER 4

MODULES AND ANALYSIS

The proposed system consists of the following three main modules, namely: -

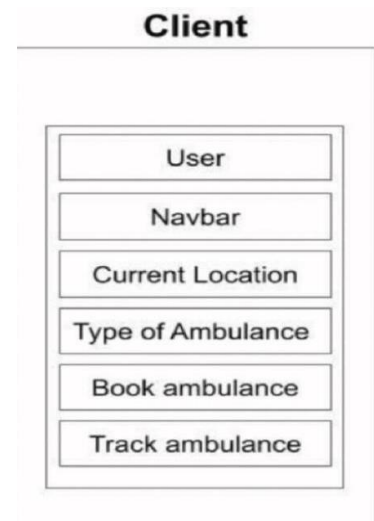
A. Client Module .

B. Driver Module.

C. Admin Module

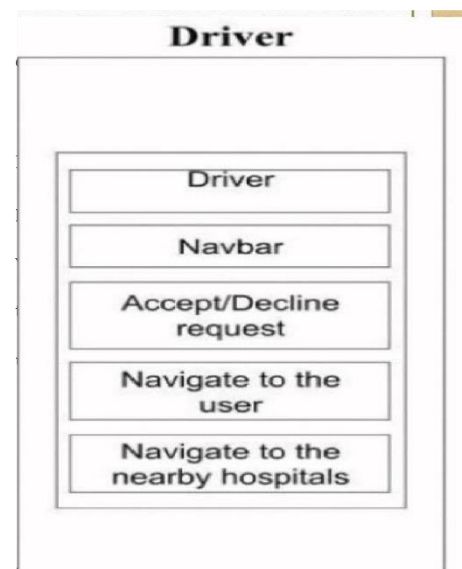
4.1 A. Client Module

The user will have features like booking an ambulance, cancel the booking, track the location of the ambulance. The user can choose the type of ambulance while booking for the ambulance services. The user can also manually choose his current location or automatically choose his current location. After successful booking of an ambulance, user can track the ambulance's live location for their convenience.



4.2 B. Driver Module

Same as the user module, here the driver can accept/decline requests, navigate to the user, navigate to a nearby hospital after picking up the patient. The location of the driver is automatically noted while entering the application. Here, the profile updating privileges are restricted because drivers do not have extensive usage like the Users and Citizens.



4.3 ADMIN module

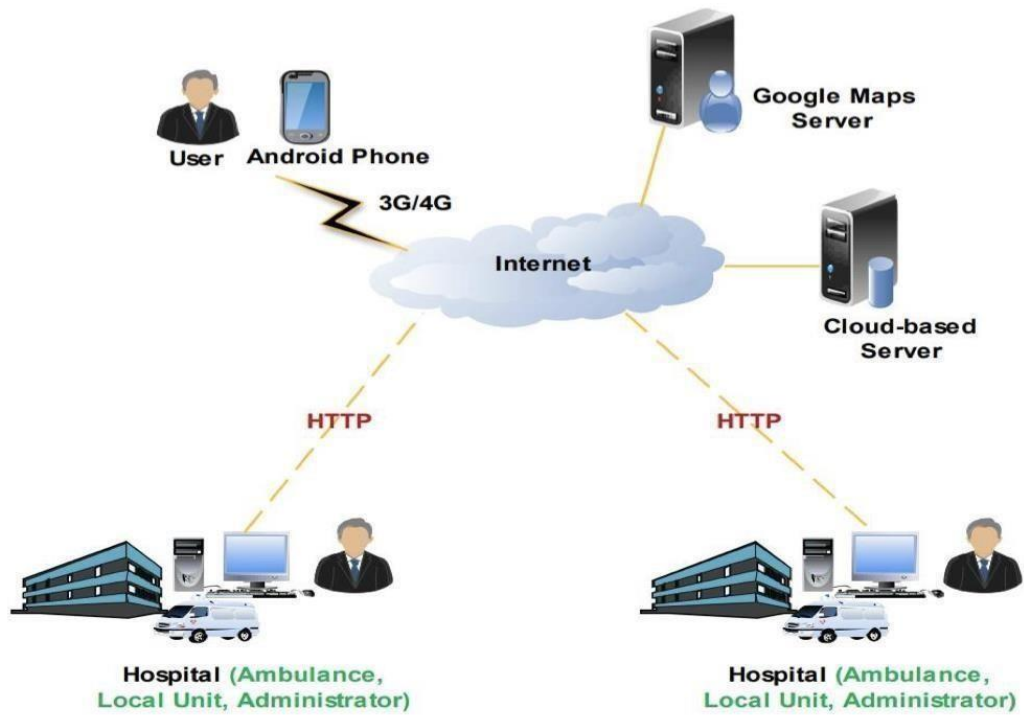
In the Admin module, the functionality is reduced but privileges are increased. The administrator application will have the functionalities like collecting details of the registered users, daily usage report, number of users registered into the application, trip details.

Trip details
Driver details
User details
Ambulance details

CHAPTER 5

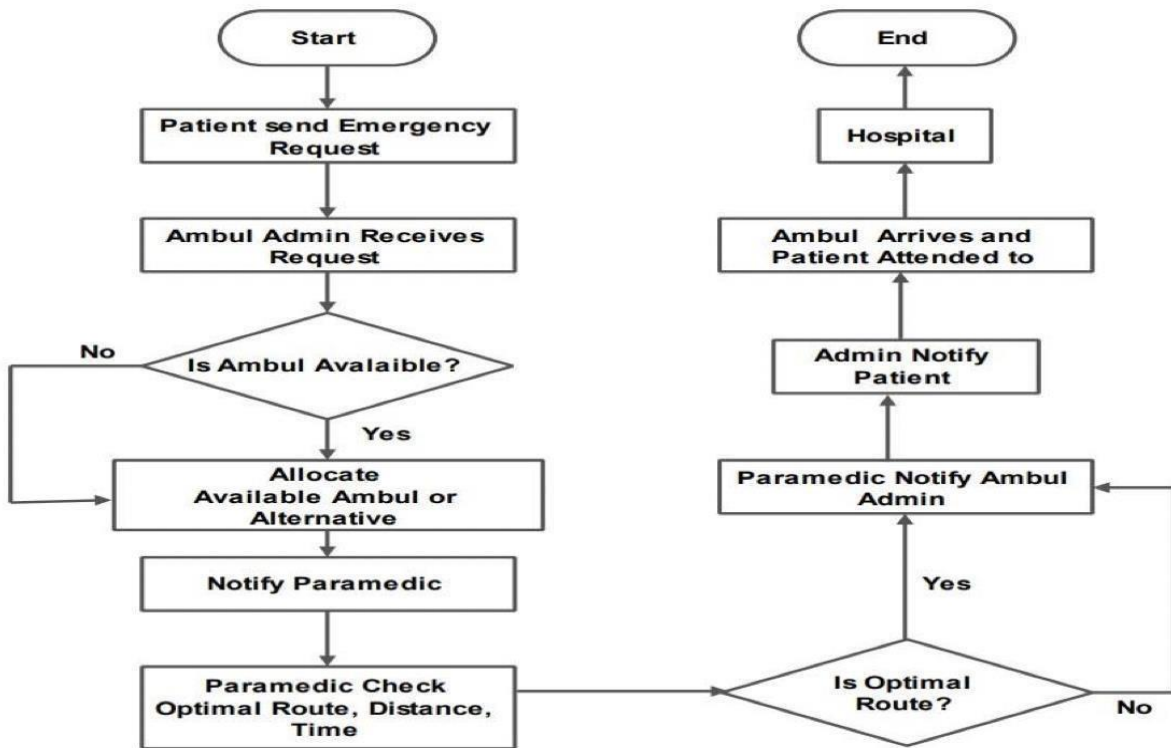
5.1 ARCHITECTURE

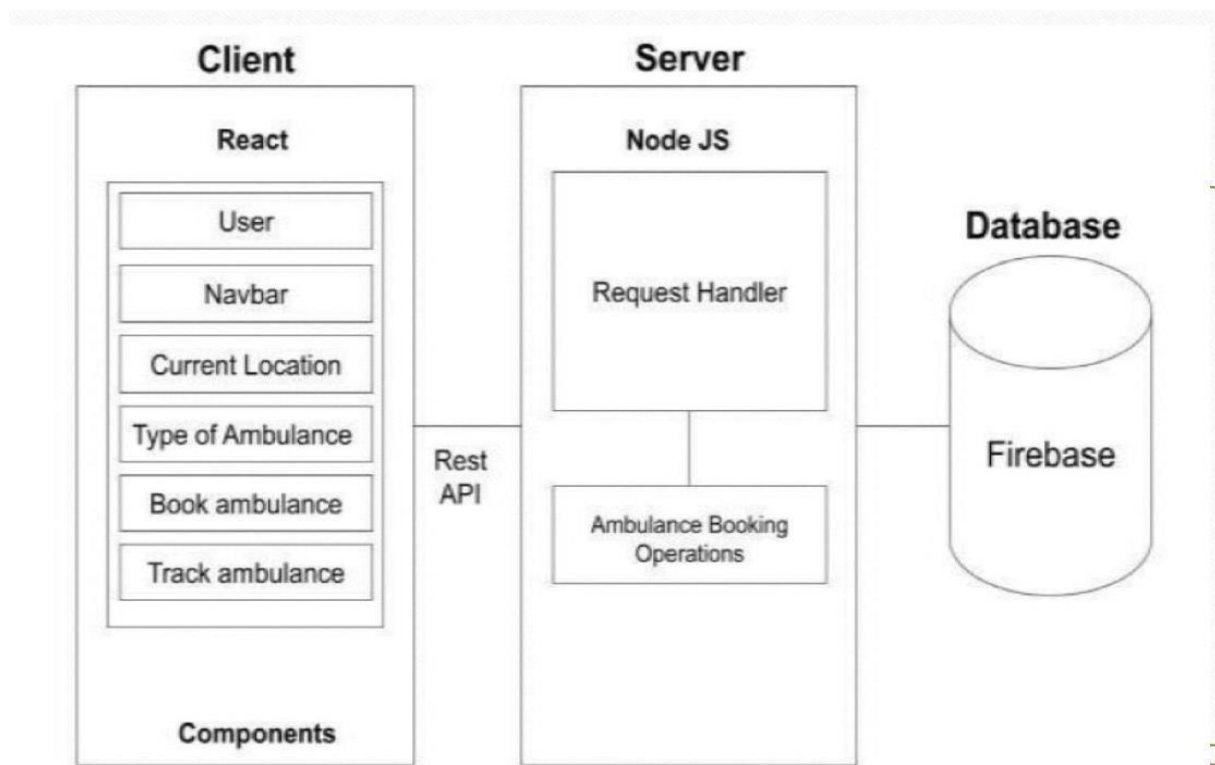
Figure 5.1 : General Architecture Diagram for emergency ambulance booking system



CHAPTER 6

6.1 ALGORITHM





CHAPTER 7

RESULTS AND DISCUSSIONS

This applications, all problems have a solution that is used to improve the ambulance service. This online way of booking ambulances can save lots of lives.

1. Our main objective is to develop an application that is used to Book / Track ambulances in emergency situations.
2. The outcome of this project helps to book the ambulance in an easy way by just a button click and also enables the driver to navigate to the user's location.
3. The main advantage of this project is reducing the time delay in ambulance services which nearly saves the death rate to half the amount in current civilization.
3. Our goal is basically focused on overcoming the existing system for ambulance booking and to introduce a new productive application which solves all the problems of the existing system.

7.1 PROGRAM CODE

MAIN ACTIVITY.

```
package com.android.debasrito.ambulanceapp;  
  
import android.Manifest;  
import android.annotation.SuppressLint;  
import android.annotation.TargetApi;  
import android.content.Context;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.location.Location;  
import android.location.LocationListener;  
import android.location.LocationManager;  
import android.net.ConnectivityManager;  
import android.net.NetworkInfo;  
import android.net.Uri;  
import android.os.Build;  
import android.os.Bundle;  
import android.os.Parcelable;  
import android.provider.Settings;  
import android.text.TextUtils;  
import android.view.View;
```

```

import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;

import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity implements LocationListener {
    TextView nameget, mobileget;
    String name, mobile, cartype="Select";
    Button book;
    Intent finderintent;
    LocationManager locationManager;
    Location loc;
    Spinner typeselect;
    @TargetApi(Build.VERSION_CODES.M)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        nameget = (TextView) findViewById(R.id.name);
        mobileget = (TextView) findViewById(R.id.mobile);
        book = (Button) findViewById(R.id.book);
        locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);
        while(Build.VERSION.SDK_INT >= Build.VERSION_CODES.M &&
            checkSelfPermission(Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED)
        {

            requestPermissions(new
String[] {Manifest.permission.ACCESS_FINE_LOCATION}, 0);
        }
        try {

```

```

        locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
5000, 5, (LocationListener) this);
        loc =
locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
    }
    catch(SecurityException e) {
        e.printStackTrace();
    }
    final List<String> list=new ArrayList<>();
    list.add("Select");list.add("AC");list.add("Non-AC");
    ArrayAdapter<String> type=new
ArrayAdapter<String>(this,android.R.layout.simple_spinner_dropdown_item,list);
    type.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    typeselect=(Spinner) findViewById(R.id.typelist);
    typeselect.setAdapter(type);
    typeselect.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position, long
id) {
            cartype=list.get(position);
        }

        @Override
        public void onNothingSelected(AdapterView<?> parent) {

        }
    });
    book.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (!isNetworkAvailable())
            {
                Toast.makeText(MainActivity.this, "Please enable
internet.", Toast.LENGTH_SHORT).show();
            }
            else if(cartype.equals(list.get(0)))
            {
                Toast.makeText(MainActivity.this, "Please choose a car type.",
Toast.LENGTH_SHORT).show();
            }
            else if (TextUtils.isEmpty(nameget.getText())) {
                nameget.setError("Name Required");
            } else if (TextUtils.isEmpty(mobileget.getText())) {
                mobileget.setError("Mobile number required");
            } else {
                name = nameget.getText().toString();
                mobile = mobileget.getText().toString();
            }
        }
    });
}

```

```

        finderintent = new Intent(MainActivity.this, finder.class);
        finderintent.putExtra("NAME", name);
        finderintent.putExtra("MOBILE", mobile);
        finderintent.putExtra("LOCATION", (Parcelable) loc);
        finderintent.putExtra("CARTYPE", cartype);
        startActivity(finderintent);
    }
}
});
}

@Override
public void onLocationChanged(Location location) {
    loc=location;
}

@Override
public void onStatusChanged(String provider, int status, Bundle extras) {

}

@Override
public void onProviderEnabled(String provider) {

}

@Override
public void onProviderDisabled(String provider) {
    Toast.makeText(MainActivity.this, "Please Enable GPS and Internet",
        Toast.LENGTH_SHORT).show();
}

private boolean isNetworkAvailable() {
    ConnectivityManager connectivityManager
        = (ConnectivityManager)
        getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo activeNetworkInfo = connectivityManager.getActiveNetworkInfo();
    return activeNetworkInfo != null && activeNetworkInfo.isConnected();
}
}

Details.JAVA.
package com.android.debasrito.ambulanceapp;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;

```



```

import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.graphics.Color;
import android.location.Location;
import android.os.Bundle;
import android.os.Parcelable;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.MapView;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.maps.model.Polyline;
import com.google.android.gms.maps.model.PolylineOptions;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.google.maps.android.SphericalUtil;

public class details extends AppCompatActivity implements OnMapReadyCallback {
    TextView caseid, name, mobile, car;
    Intent i;
    String patient, pmobile;
    private MapView mapView;
    private GoogleMap gmap;
    private static final String MAP_VIEW_BUNDLE_KEY =
    "AlzaSyDhdIQRxEf0jaIepeZylPW64flu6zKgu1I";
    Location loc;
    Location dloc=new Location("");
    String dname = "PLEASE WAIT", dmobile = "PLEASE WAIT", did = "PLEASE WAIT",
    cardet = "PLEASE WAIT", cartype;
    double dist = 1000000000,distance;
    Marker dmark,dmark2,pmark,pmark2;
    Boolean status,active;
    boolean notavail=true;
    Double dlat,dlong;
    //long cost,totaltime,totaltimefinal;
    final DatabaseReference referencemap =
    FirebaseDatabase.getInstance().getReference("drivers");

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_details);
    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
    caseid = (TextView) findViewById(R.id.caseid);
    name = (TextView) findViewById(R.id.drivername);
    mobile = (TextView) findViewById(R.id.mobile);
    car = (TextView) findViewById(R.id.cardet);
    i = getIntent();
    patient = i.getStringExtra("NAME");
    pmobile = i.getStringExtra("MOBILE");
    loc = i.getParcelableExtra("LOCATION");
    status=i.getBooleanExtra("STATUS",false);
    active=i.getBooleanExtra("ACTIVE",false);
    dname=i.getStringExtra("DRIVER");
    dmobile= i.getStringExtra("DMOBILE");
    dloc=i.getParcelableExtra("DLOC");
    cardet=i.getStringExtra("CAR");
    did=i.getStringExtra("ID");
    notavail=i.getBooleanExtra("NOTAVAIL",true);
    cartype=i.getStringExtra("CARTYPE");
    final DatabaseReference reference =
    FirebaseDatabase.getInstance().getReference("drivers");
    if (notavail) {
        Toast.makeText(this, "No ambulance found", Toast.LENGTH_LONG).show();
        finish();
    } else {
        reference.child(did).child("driverstatus").setValue(false);
        reference.child(did).child("patientlatitude").setValue(loc.getLatitude());
        reference.child(did).child("patientlongitude").setValue(loc.getLongitude());
        reference.child(did).child("patientname").setValue(patient);
        reference.child(did).child("patientmobile").setValue(pmobile);
        caseid.setText(did);
        name.setText(dname);
        mobile.setText(dmobile);
        car.setText(cardet);
    }
    Bundle mapViewBundle = null;
    if (savedInstanceState != null) {
        mapViewBundle = savedInstanceState.getBundle(MAP_VIEW_BUNDLE_KEY);
    }

    mapView = findViewById(R.id.map);
    mapView.onCreate(mapViewBundle);
    mapView.getMapAsync(this);
}

```

```

}

@Override
public void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);

    Bundle mapViewBundle = outState.getBundle(MAP_VIEW_BUNDLE_KEY);
    if (mapViewBundle == null) {
        mapViewBundle = new Bundle();
        outState.putBundle(MAP_VIEW_BUNDLE_KEY, mapViewBundle);
    }

    mapView.onSaveInstanceState(mapViewBundle);
}

@Override
protected void onResume() {
    super.onResume();
    mapView.onResume();
}

@Override
protected void onStart() {
    super.onStart();
    mapView.onStart();
}

@Override
protected void onStop() {
    super.onStop();
    mapView.onStop();
}

@Override
protected void onPause() {
    mapView.onPause();
    super.onPause();
}

@Override
protected void onDestroy() {
    mapView.onDestroy();
    super.onDestroy();
}

@Override
public void onLowMemory() {

```

```

        super.onLowMemory();
        mapView.onLowMemory();
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
        gmap = googleMap;
        gmap.setMinZoomPreference(12);
        LatLng ppos = new LatLng(loc.getLatitude(), loc.getLongitude());
        pmark = googleMap.addMarker(new MarkerOptions().position(ppos)
            .title("Your
position").icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_G
REEN)));
        if (dloc != null) {
            LatLng dpos = new LatLng(dloc.getLatitude(), dloc.getLongitude());
            dmark = gmap.addMarker(new MarkerOptions().position(dpos)
                .title("Ambulance
position").icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_C
YAN)));
            Polyline line = gmap.addPolyline(new PolylineOptions()
                .add(new LatLng(loc.getLatitude(), loc.getLongitude()), new
LatLng(dloc.getLatitude(), dloc.getLongitude()))
                .width(5)
                .color(Color.RED));
        }
        gmap.moveCamera(CameraUpdateFactory.newLatLng(ppos));
        //try new.....
        referencemap.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                if(!notavail) {

                    dloc.setLatitude(dataSnapshot.child(did).child("driverlatitude").getValue(Double.class));

                    dloc.setLongitude(dataSnapshot.child(did).child("driverlongitude").getValue(Double.class))
                    ;

                    gmap.clear();
                    LatLng ppos = new LatLng(loc.getLatitude(), loc.getLongitude());
                    pmark2 = gmap.addMarker(new MarkerOptions().position(ppos)
                        .title("Your
position").icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_G
REEN)));
                    LatLng dpos = new LatLng(dloc.getLatitude(), dloc.getLongitude());
                    dmark2 = gmap.addMarker(new MarkerOptions().position(dpos)
                        .title("Ambulance
position").icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_C
YAN)));
                }
            }
        });
    }
}

```

```

        Polyline line = gmap.addPolyline(new PolylineOptions()
            .add(new LatLng(loc.getLatitude(), loc.getLongitude()), new
LatLng(dloc.getLatitude(), dloc.getLongitude()))
            .width(10)
            .color(Color.BLACK));
    }
    /*else if(dataSnapshot.child(did).child("driverstatus").getValue(Boolean.class))
    {
        //Trip end calculations.
        //totaltime=dataSnapshot.child("endtime").getValue(Long.class)-
dataSnapshot.child("starttime").getValue(Long.class);
        totaltime=dataSnapshot.child("time").getValue(int.class);
        totaltimefinal=((totaltime/100)/60)/60);
        if(totaltimefinal<0.25)
            cost=25;
        else
            cost=totaltimefinal*100;
        System.out.println(cost);
        //Intent intent = new Intent();
        //startActivity(intent);
    }*/
}

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

}
});
//End try new.....
}
}

```

DRIVER MAIN ACTIVITY.

```

package com.android.debasrito.driver;

import android.Manifest;
import android.annotation.TargetApi;
import android.content.Context;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.net.ConnectivityManager;

```

```

import android.net.NetworkInfo;
import android.os.Build;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.os.Parcelable;
import android.text.TextUtils;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.maps.model.LatLng;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.sql.Driver;
import java.util.ArrayList;
import java.util.List;
import java.util.Objects;

public class MainActivity extends AppCompatActivity implements LocationListener {
    TextView nameget, mobileget, carget;
    String dname, dmobile, car, id, cartype="Select";
    Button check;
    Intent checkintent;
    LocationManager locationManager;
    Location loc;
    Boolean status=true,active=false,found=false;
    Spinner typeselect;
    @TargetApi(Build.VERSION_CODES.M)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        nameget=(TextView) findViewById(R.id.name);
        mobileget=(TextView) findViewById(R.id.mobile);
        carget=(TextView) findViewById(R.id.vehiclenumber);
        check=(Button) findViewById(R.id.checkstatus);
    }
}

```

```

        final List<String> list=new ArrayList<>();
        list.add("Select");list.add("AC");list.add("Non-AC");
        ArrayAdapter<String> type=new
ArrayAdapter<String>(this,android.R.layout.simple_spinner_dropdown_item,list);
        type.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        typeselect=(Spinner) findViewById(R.id.typelist);
        typeselect.setAdapter(type);
        typeselect.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long
id) {
                cartype=list.get(position);
            }

            @Override
            public void onNothingSelected(AdapterView<?> parent) {

            }
        });
        locationManager = (LocationManager)
getService(Context.LOCATION_SERVICE);
        while(Build.VERSION.SDK_INT >= Build.VERSION_CODES.M &&
            checkSelfPermission(Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED)
        {

            requestPermissions(new
String[] {Manifest.permission.ACCESS_FINE_LOCATION}, 0);
        }
        try {
            locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
5000, 5, (LocationListener) this);
            loc =
locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
        }
        catch(SecurityException e) {
            e.printStackTrace();
        }
        check.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (!isNetworkAvailable())
                {
                    Toast.makeText(MainActivity.this,"Please enable
internet.",Toast.LENGTH_SHORT).show();
                }
                else if(cartype.equals(list.get(0)))

```

```

        {
            Toast.makeText(MainActivity.this, "Please choose a car type.",
Toast.LENGTH_SHORT).show();
        }
        else if (TextUtils.isEmpty(nameget.getText())) {
            nameget.setError("Name Required");
        } else if (TextUtils.isEmpty(mobileget.getText())) {
            mobileget.setError("Mobile number required");
        } else if (TextUtils.isEmpty(carget.getText())) {
            carget.setError("Car number required");
        } else {
            dname = nameget.getText().toString();
            dmobile = mobileget.getText().toString();
            car = carget.getText().toString();
            final DatabaseReference reference =
FirebaseDatabase.getInstance().getReference("drivers");
            reference.addListenerForSingleValueEvent(new ValueEventListener() {
                @Override
                public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                    for (DataSnapshot datas : dataSnapshot.getChildren()) {
                        if (datas.child("drivername").getValue(String.class).equals(dname) &&
datas.child("cartype").getValue(String.class).equals(cartype) &&
datas.child("drivercar").getValue(String.class).equals(car) &&
datas.child("drivermobile").getValue(String.class).equals(dmobile)) {
                            status = datas.child("driverstatus").getValue(Boolean.class);
                            active = datas.child("driveractive").getValue(Boolean.class);
                            id= datas.getKey();
                            found=true;
                            if (!active) //not logged in. set logged in and go to next intent. pass
status as true to disable find route function
                        {
                            checkintent = new Intent(MainActivity.this, status.class);
                            checkintent.putExtra("NAME", dname);
                            checkintent.putExtra("MOBILE", dmobile);
                            checkintent.putExtra("CAR", car);
                            checkintent.putExtra("ID", id);
                            checkintent.putExtra("STATUS", true);
                            checkintent.putExtra("ACTIVE", true);
                            checkintent.putExtra("CARTYPE", cartype);
                            reference.child(id).child("driveractive").setValue(true);
                            reference.child(id).child("driverstatus").setValue(true);
                            //checkintent.putExtra("PATIENT", "Please wait");
                            //checkintent.putExtra("PMOB", "Please wait");
                            //checkintent.putExtra("PLAT", "Please wait");
                            //checkintent.putExtra("PLON", "Please wait");
                            startActivity(checkintent);
                            finish();
                        }
                    }
                }
            });
        }
    }
}

```



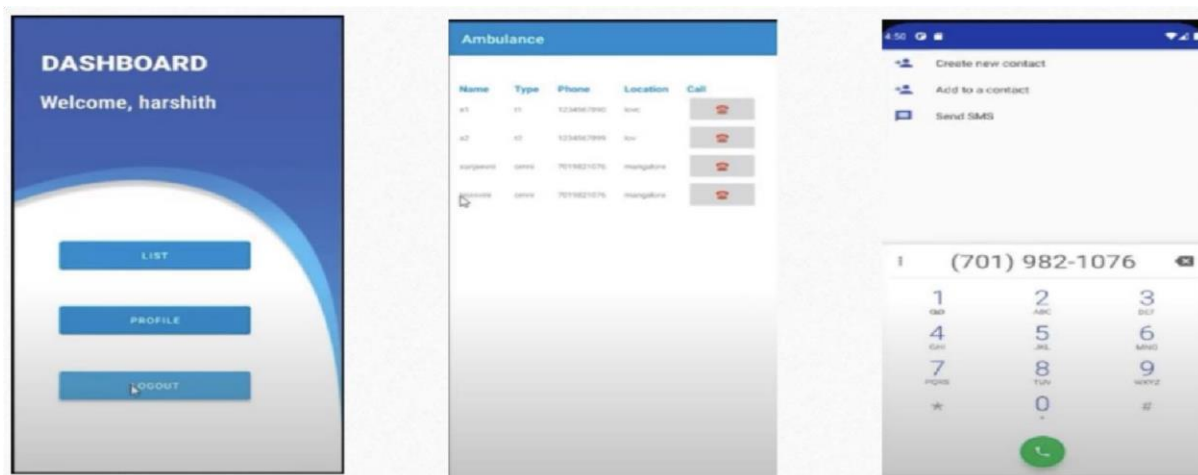
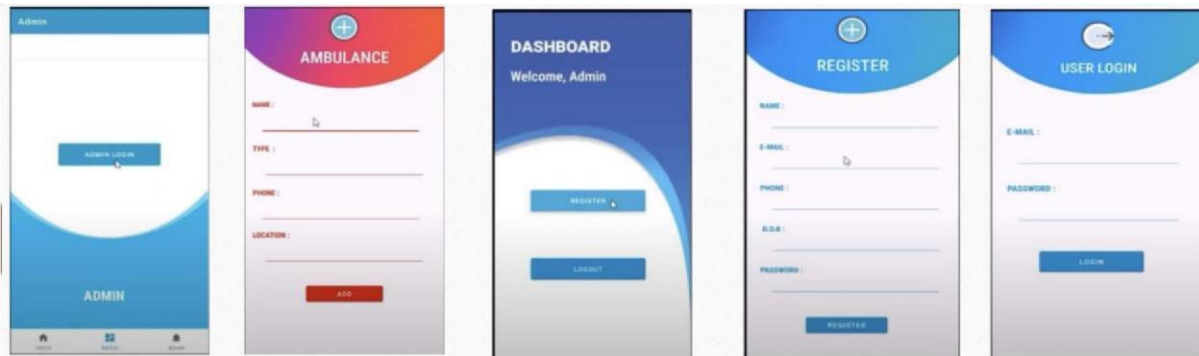
```

    }
    else if(active && !status)//logged in and patient taken
    {
        //get details of patient and pass that along with status to set up find
route buttons

        checkintent = new Intent(MainActivity.this, status.class);
        checkintent.putExtra("NAME", dname);
        checkintent.putExtra("MOBILE", dmobile);
        checkintent.putExtra("CAR", car);
        checkintent.putExtra("ID",id);
        checkintent.putExtra("STATUS",false);
        checkintent.putExtra("ACTIVE",true);
        checkintent.putExtra("CARTYPE",cartype);
        //checkintent.putExtra("PATIENT", "Please wait");
        //checkintent.putExtra("PMOB", "Please wait");
        //checkintent.putExtra("PLAT", "Please wait");
        //checkintent.putExtra("PLON", "Please wait");
        startActivity(checkintent);
        finish();
    }
    if(!found)
    {
        //else account not found. create it using push() method
        id=reference.push().getKey();
        assert id != null;
        reference.child(id).setValue(new details(dname,dmobile,car,"Please
wait", "Please wait",cartype,true,true,loc.getLatitude(),loc.getLongitude(),0.0,0.0));
        checkintent = new Intent(MainActivity.this, status.class);
        checkintent.putExtra("NAME", dname);
        checkintent.putExtra("MOBILE", dmobile);
        checkintent.putExtra("CAR", car);
        checkintent.putExtra("ID",id);
        checkintent.putExtra("STATUS",true);
        checkintent.putExtra("ACTIVE",true);
        checkintent.putExtra("CARTYPE",cartype);
        startActivity(checkintent);
        finish();
    }
}
}
@Override
public void onCancelled(@NonNull DatabaseError databaseError) {
}
});
}

```

7.2 OUTPUT SCREENSHOT



CHAPTER 8

REFERENCES

1. Irtam Ghazi, Muhammad Rashid Maqbool, Ihtisham ul Haq, Sanaan Saud, “GPS Based Autonomous Vehicle Navigation and Control System” , Proceedings of 2016 13th International Bhurban Conference on Applied Sciences & Technology (IBCAST),January 2016
2. Samir El-Masri & Basema Saddik, ‘An Emergency System to Improve Ambulance Dispatching, Ambulance Diversion and Clinical Handover Communication—A Proposed Model’ in J Med Syst (2012) 36:3917–3923.
3. Almadani, Basem., Bin-Yahya, Manaf., Shakshuki, Elhadi M. 2015. „E- ambulance: RealTime Integration Platform for Heterogeneous Medical Telemetry System“. Procedia Computer Science. 63, 400 – 407.
4. Anantharaman, V., Han Lim, Swee., 2001. „Hospital and emergency ambulance link: using IT to enhance emergency pre-hospital care“.International Journal of Medical Informatics. 61, 147–161.
5. Aboueljinane, L., Jemai E., Sahin, Z., 2013. „A review on simulation models applied to emergency medical service operations“. International Journal of Computers and Industrial Engineering. 66, 734– 750
6. Almehdawe, Eman., Jewkes, Beth., He, Qi-Ming., 2013. „A Markovian queueing model for ambulance offload delays“. European Journal of Operational Research. 226, 602–614.
7. Almehdawe, Eman., Jewkes, Beth., He, Qi-Ming., 2013. „A Markovian queueing model for ambulance offload delays“.European Journal of Operational Research. 226, 602–614
8. Muhd Zafeeruddin Bin Mohd Sakriya, Joshua Samuel,AMBULANCE EMERGENCY RESPONSE APPLICATION’ in International Journal of Information System

8.2 RESULTS

In this system, the User will be able to book an ambulance in advance according to the size of the ambulance and selected hospital, or the user can also book an ambulance for emergency regardless of its size and a random hospital will be allocated to the user. Then the ambulance driver will accept or reject the booking from the user, after accepting or rejecting the status will be updated for the same to the user. Hospital can view the bookings history of the user for that particular hospital.