1. INTRODUCTION

1.1 OVERVIEW

In the twentieth country, the number of vehicles exponentially increase due to growth in the automobile industry, As the number of vehicles increases, the accident also increases. The reasons for most road accident are heterogeneous traffic and lack of traffic separation. In India, 13 million people were dead in road accident in the year of 2014-2015. The existing system mostly focused on the safety of the passenger but not on the immediate help after the accident.India has earned the dubious distinction of having a greater number of fatalities due to road accident in the world. Road safety is emerging as a major social concern around the world, especially in India.The system implemented by us aims at reducing life danger after road accidents occur. It is not only efficient but also worthy to be implemented. Accident detection and the messaging system can be fitted in the vehicle (Ambulance & Police) and they are informed about any such untoward incident at the go. GPS is used for taking the coordinate of the site of the accident while GSM is used for sending the message to a phone. To make this process all the control is made using Arduino.It sends the message through the GSM module for the numbers provided by the driver. The message received by mobile in which it displays a Google map showing the exact location of the accident location which helps to get medical help ASAP.

1.2 OBJECTIVES

The objective of this project is to create an IoT application “Emergency Alert notification system”, where any user (victim) is met with an accident and he/she/they are unable to contact their families, police station, hospital and there is no one to help them. In this situation, as soon as the accident has occurred, a message along with the location of the accident is sent to their family members, police station, hospital. By this, we can save valuable human life.

1.3 PROBLEM FORMULATION

Over speeding, drunken driving, a distraction to the driver, signal jumping,

overtaking in a wrong manner, and avoiding safety measures like seat belts and helmets are the basic reasons for a vehicle accident. When the accident takes place in a remote area or if the victim is unable to send any information to their family members/ to the hospital/ nearby police station, we can come up with the solution of sending the message to the registered phone numbers the location of the accident that took place through the GPS.

1.4 SCOPE

The scope of this project is that, the message along with the location of the accident to their family members, police station and hospital, when the major accident has occurred to any four-wheeler, three-wheeler, and two-wheeler (except cycles).In case of a minor accident, message along with the accident are sent to their family members, police station, hospital when the victim presses the button in our application.

1.5 FEASIBILITY:

It is to find out whether the current work practices and procedures support new system. Also, social factors i.e., how the organizational changes will affect the working lives of those affected by the system. This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology.

1.6 SYSTEM REQUIREMENTS

1.6.1 Software Requirements

Arduino IDE software
Windows 7 or Windows 8/8.1 or Windows 10
512 MB RAM(1GB RAM recommended)
600 MB of free disk space

1.6.2 Hardware Requirements

GSM Module
GPS Module
Arduino UNO
Bread Board

Accelerometer ADXL335

Push Button

Jumper Wires

Vibration Sensor

Node MCU

LCD Display

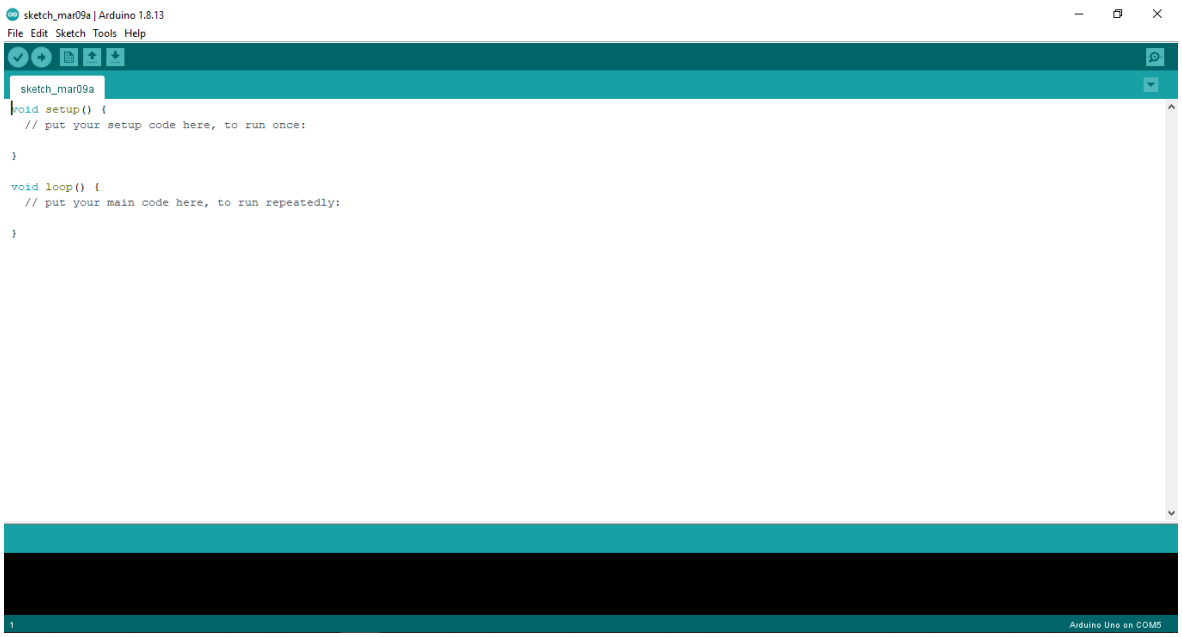
Shock Sensors

2. REQUIREMENTS

2.1 REQUIREMENT SPECIFICATION

1 Arduino IDE Software

- Arduino IDE is open-source software and a cross-platform application (for Windows, Mac OS, Linux) and is mainly used for writing and compiling/uploading the code into Arduino compatible boards. The code is mainly written in C++.

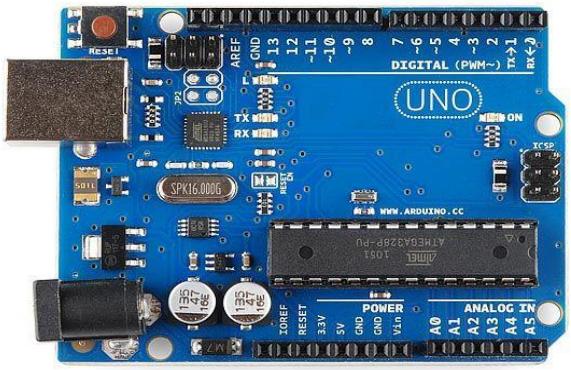


2 GSM Module (SIM 800A)

- SIM 800 is a quad-band GSM/GPRS module. It has one UART port, it also includes a microphone input and a receiver output. SIM 800 has one sim card interface.SIM 800 can be controlled/configured using simple AI commands.It operates from 3.4 to 4.4. It can be used for sending/receiving messages, making calls, sending/receiving data over the internet.

3 Arduino UNO

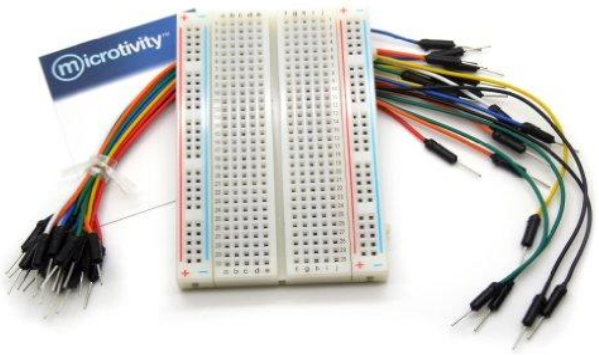
- Arduino UNO is a microcontroller board based on an 8-bit Amega 328p microcontroller. Arduino UNO has 14 digital input/output pins, 6 analogue input pins, a USB connection, A power barrel Jack and a rest button. When the code is compiled and uploaded into Arduino UO, it executes the code repeatedly because of the microcontroller.



4 BreadBoard and Jumper Wires

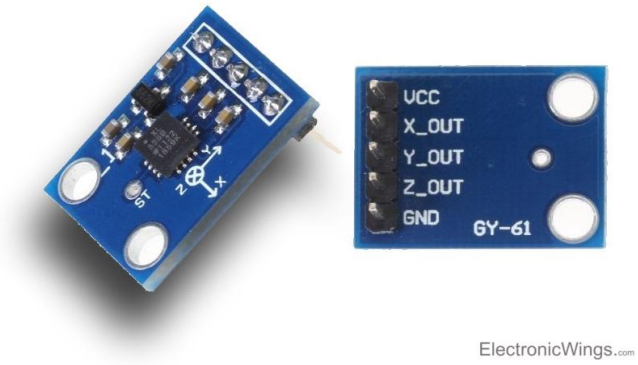
- A broadband is as less device for temporary prototype with electronics and test circuits designs the purpose of the breadboard is to make quick electrical connections between components like Icds, Gps sensor, Arduino UNO, Node

MCU, so that you can test your circuit before permanently soldering it together.Jumper Wires are simply wires that have connector pins at each end, allowing them to be used to connect two pints to each other without soldering. Jumper wires are typically used with breadbands and other prototype tools in order to make it easy to change a current as needed.



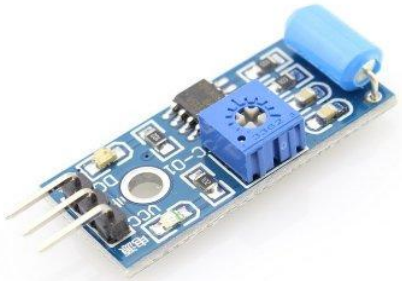
5 Accelerometer ADXL335

- The Adx1335 is a small, thin, low power, complete 3-axis accelerometer. The product measures acceleration with minimum fun-scale range of >=3g.By measuring the amount of acceleration due to gravity, an accelerometer can figure out the angle it is fitted at with respect to the earth.The module is made up of Adx1335 from Analog devices.



6 Vibration and Shock Sensor

- A vibration sensor is a device that measures the amount and frequency of vibration in a given system machine.A shock sensor detects an impact to the bod of the vehicle.To test a shock sensor user simply need to tap the glass/breadboard its affixed to shock sensors sensitivity can be increased or decreased.

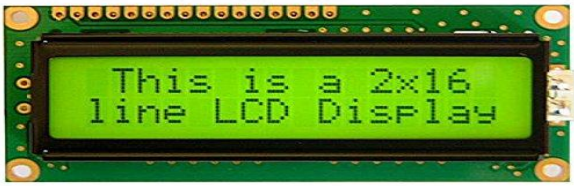


7 Node MCU

- Nodemcu is a low cost open source lot platform which runs on the Esp8266 wifi-soc from Espressit systems, and hardware which was based on the esp-12 module, Ithas 16 general purpose input-outputs, one analog pin, a micro-usb connection.It has 128KB RAM and 4MB of Flash memory to store data and programs.

8 LCD Display

- An LCD (Liquid Crystal Display) screen is an electronic display module. It can display 16 characters per line and there are 2 such lines.



2.2 USER INTERFACES

LOGIN SCREEN

A login screen is the screen where the user is asked to fill his credentials like phone number, name, etc.… to authenticate the user, Also, to validate whether the user already exist or not.

REGISTER SCREEN

A register screen is the screen where the user is asked to fil the form like username, password, phone number, emergency contact numbers etc. from a user to create a web-based account or profile. A user registration system generally asks a user to create a username and password.

SERIAL MONITORS

The Arduino IDE has a feature that can be a great help for controlling Arduino from your computer's keyboard. The serial monitor is a separate pop-up window that acts as a separate terminal that communicates by receiving and sending serial data.

MESSAGE SCREEN

Message screen is a screen where the accident occured information along with location of the accident is displayed.

MAPS SCREEN

Maps screen is the screen where it shows the extact location of the accident in the google maps.

3.3 CONSTRAINTS, PREREQUISITES

There are few constraints related to our project (Emergency Alert Notification System). The major aspect here is we are dependent on GSM Module (SIM 800A), Nodemcu for wifi-connectivity, GPS Sensor, Accelerometer Sensor (ADXL335) and Vibration Sensor.Certain situation is being considered for the project to be in working conditions.GPS sensor works well in outdoors but not

indoors (building). GSM Sensor will not work, if their is no network connection or having poor signal strength. NodeMCU will not execute the code untill it connected to any wifi with proper Internet connection.If the wires, sensor-pins are connected loosely or open-circuted will not work.If high voltage is provided to our application, it damages our application.

4. ANALYSIS

4.2 USE CASE MODEL

4.2.1 USE CASE MODEL

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

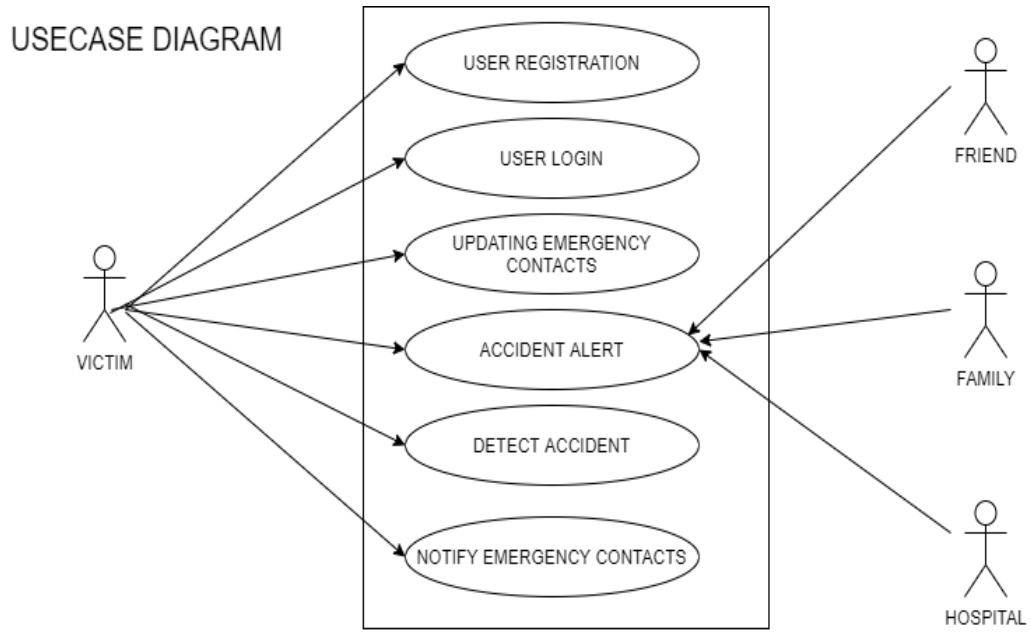


Figure 4.2.1 Use case Diagram

4.2.1.1 FLOW OF EVENTS

Our Application performs the functionalities like creating the login, registration, Profile settings, Setting the command Password for changing the device state, Sending the Location request, Finding the friends location, Update location and its frequency

4.2.2 ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe dynamic aspects of the system, Activity diagram is basically a flowchart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So, the control flow is drawn from one operation to another.

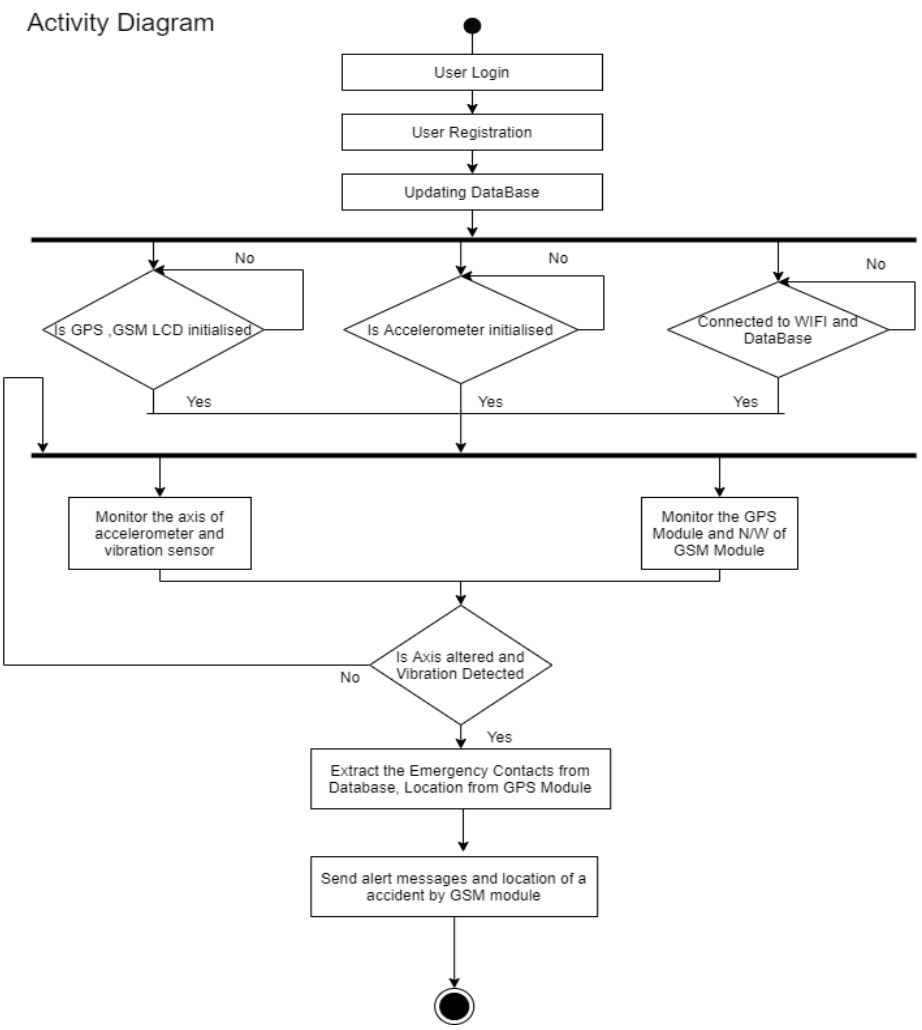


Figure 4.2.2 Activity Diagram

4.2.2.1 FLOW OF ACTIVITIES

Firstly, the Application checks whether the user is registered or not. If not registered asks to create a user else login. On Logging in the application you can three different activities like Requesting for the location, controlling the device state and locking the phone.

5. DESIGN

5.1 BLOCK DIAGRAM

A block diagram is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

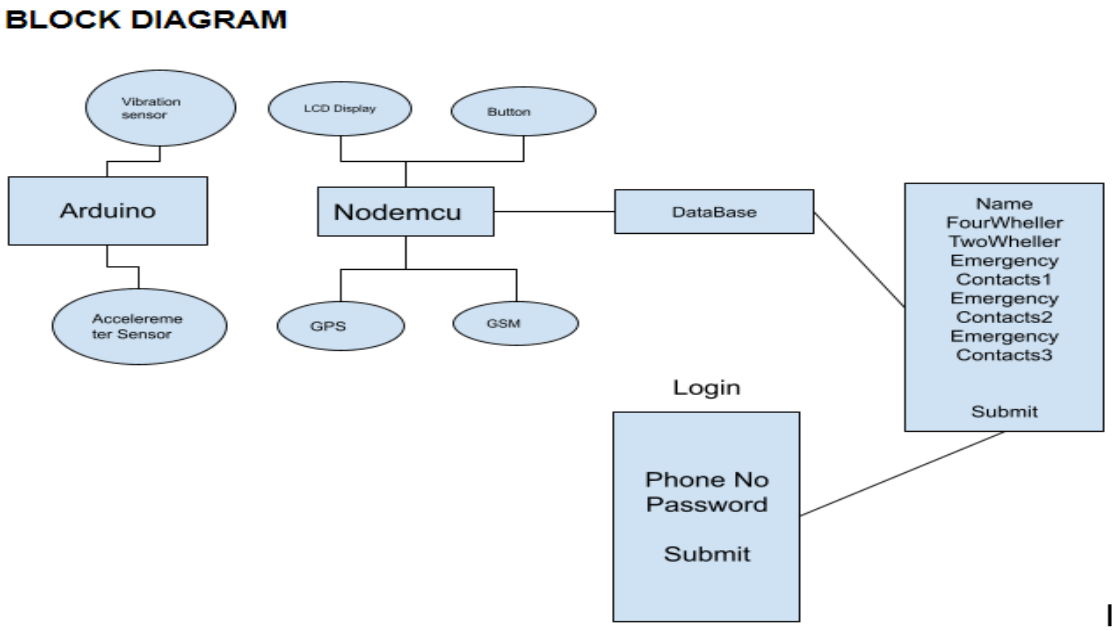


Figure 5.1 Block diagram

5.2 DATABASE DESIGN

ACCIDENT DETAIL TABLE:

ACCIDENT-DETECTION-SYSTEM-A1AD8-DEFAULT-RTBD			
FIELD TYPE	DATA TYPE	DEFAULT	KEY
PHONE	VARCHAR2(25)	NOTNULL	PRIMARY
NAME	VARCHAR2(25)	NOT NULL	
VTYPE	VARCHAR2(10)	NOT NULL	
EMERGENCY CONTACTNO-1	VARCHAR2(50)	NOT NULL	
EMERGENCY CONTACTNO-2	VARCHAR2(20)	NOT NULL	
EMERGENCY CONTACTNO-3	VARCHAR2(20)	NOT NULL	

5.4 SEQUENCE DIAGRAM

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence.

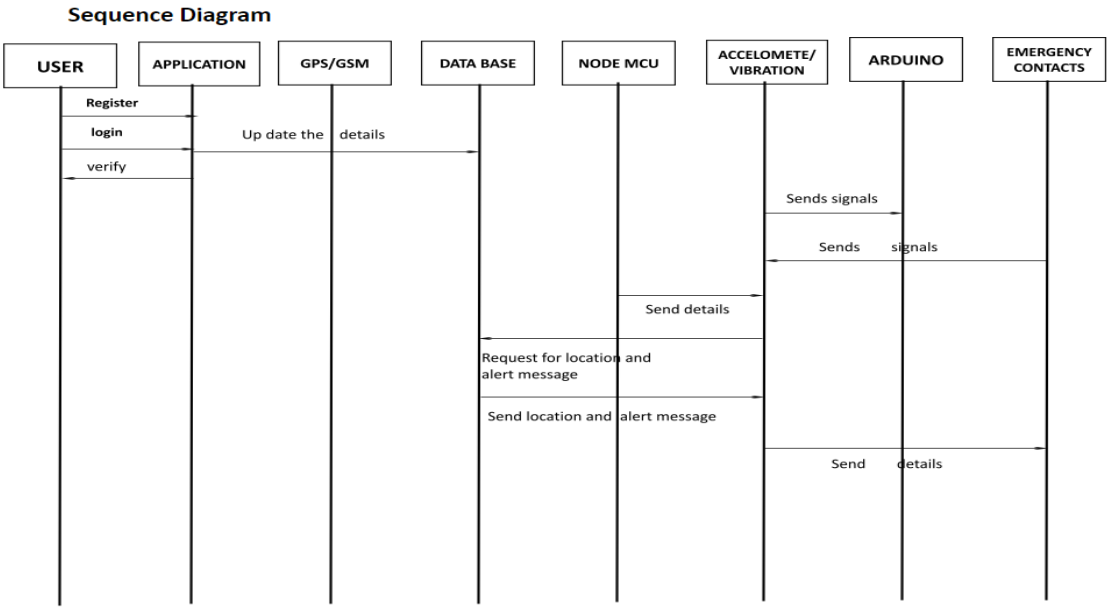


Figure 5.4 Sequence diagram

5.5 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system‘s classes, their attributes, operations (or methods), and the relationships among objects

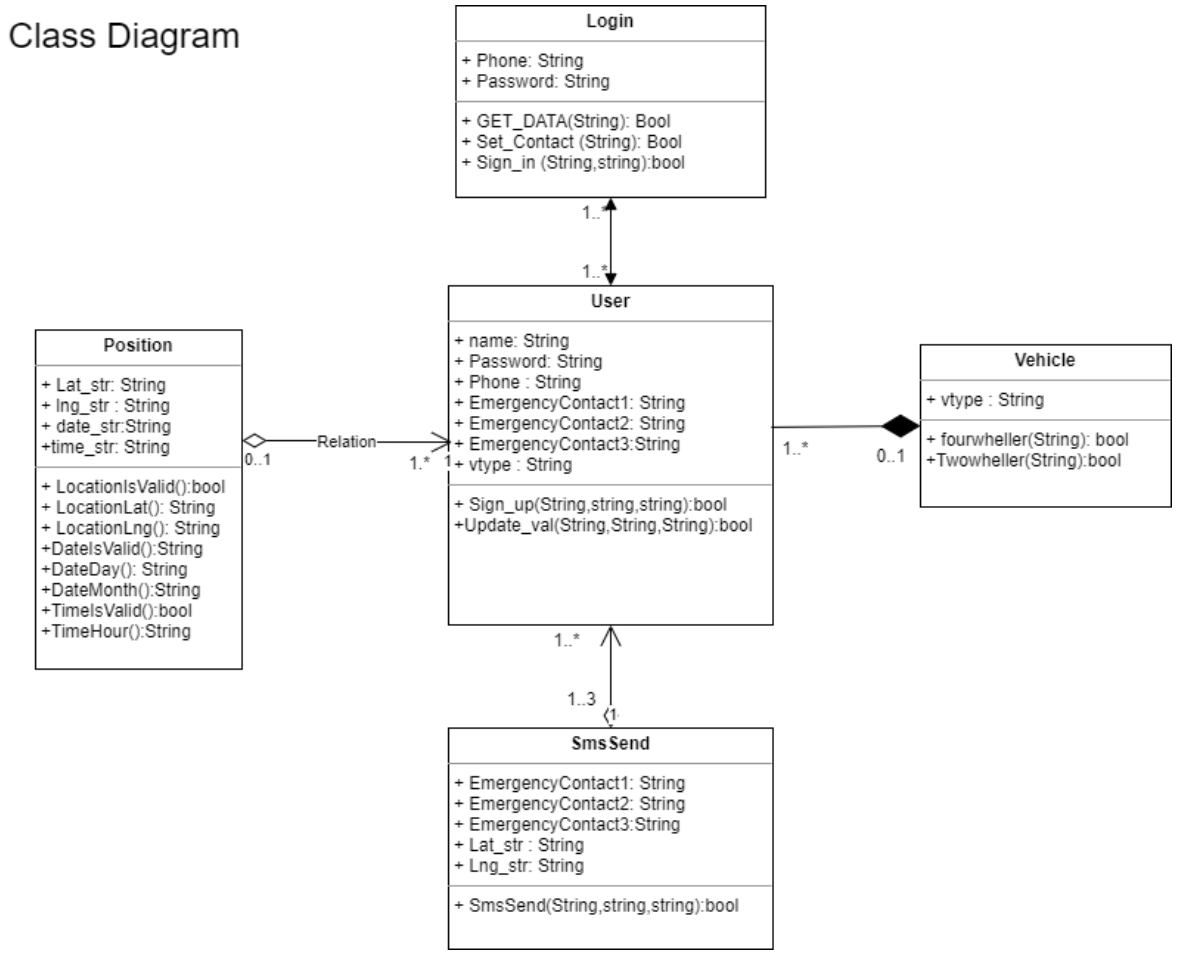


Figure 5.5 Class diagram

6. IMPLEMENTATION

Arduino is used for controlling the whole process with a GPS Receiver and GSM module. GPS Receiver is used for detecting coordinates of the vehicle, the GSM module is used for sending the alert SMS with the coordinates and the link to Google Map. Accelerometer namely ADXL335 is used for detecting accidents or sudden changes in any axis.We have used GPS Module NEO-6M and GSM Module SIM800A. When we are ready with our hardware-connections after uploading the program into Arduino, we can install it in our vehicle and

power it up. Now whenever there is an accident, the car gets tilt and accelerometer changes his axis values. These values read by Arduino and checks if any change occurs in any axis. If any change occurs then Arduino reads coordinates by extracting $GPGGA String from GPS module data and send SMS to the predefined number to the police or ambulance or family member with the location coordinates of accident place. The message also contains a Google Map link to the accident location, so that location can be easily tracked. When we receive the message then we only need to click the link and we will redirect to the Google map and then we can see the exact location of the vehicle.By this we will able to save many valuable human life.

6.3 CODE SNIPPETS

6.3.1 ACCELEROMETER AND VIBRATION SENSOR

```
#include <SoftwareSerial.h> //default library
#include <Wire.h> //default library
#define x A1
#define y A2
#define z A3
int vibration_Sensor = A5;
int present_condition = 0;
int previous_condition = 0;
int xsample = 0;
int ysample = 0;
int zsample = 0;

#define samples 10
//fourwheller
#define fourwheller-minVal 1100
#define fourwheller-MaxVal 2820
//twowheller
#define twowheller-minVal 700
#define twowheller-MaxVal 1000

void setup()
{
  Serial.begin(9600);
  Serial.println("Initializing....");
```

```
  pinMode(vibration_Sensor, INPUT);
   pinMode(2, OUTPUT);
   pinMode(3, OUTPUT);
   //pinMode(4, OUTPUT);
   digitalWrite(2, LOW);
   digitalWrite(3, LOW);
   //digitalWrite(4, HIGH);
  delay(1000);
  Accelerometerinit();
  delay(2000);
  Serial.println("Initialized Successfully");
  delay(2000);
  Serial.println("System Ready..");
}

void loop()
{

      int value1 = analogRead(x);
      int value2 = analogRead(y);
      int value3 = analogRead(z);
      int xValue = xsample - value1;
      if(xValue<0){
        xValue = -xValue;
      }
      Serial.print(" xValue = ");
      Serial.println(xValue);
      int yValue = ysample - value2;
       if(yValue<0){
        yValue = -yValue;
      }
      Serial.print(" yValue = ");
      Serial.println(yValue);
      int zValue = zsample - value3;
       if(zValue<0){
        zValue = -zValue;
      }
      Serial.print(" zValue = ");
      Serial.println(zValue);
      previous_condition = present_condition;
      Serial.println(previous_condition);
       present_condition = digitalRead(vibration_Sensor); // Reading digital data from the A5
Pin of the Arduino.
      Serial.println(present_condition);
      //210-250
```

```
      if (((xValue > 25 && xValue < 300) || (yValue > 25 && yValue < 300) || (zValue > 25
&& zValue < 300))&&(previous_condition != present_condition))
     {
       Serial.println("Checking the major-bike accident");
      digitalWrite(2,HIGH);
      digitalWrite(3, LOW);
      //digitalWrite(4,HIGH );
      delay(100);
      //130-160
     }else if(((xValue > 18 && xValue < 25) || (yValue > 18 && yValue < 25) || (zValue >
18 && zValue < 25))&&(previous_condition != present_condition)){
      Serial.println("Checking the minor-bike accident");
      digitalWrite(2,LOW);
      digitalWrite(3, LOW);
      //digitalWrite(4, LOW);
      delay(100);
    }
    //250-300
     if(((xValue > 25 && xValue < 300) || (yValue > 25 && yValue < 300) || (zValue > 25
&& zValue < 300))&&(previous_condition != present_condition)){
      Serial.println("Checking the major-car accident");
      digitalWrite(2, HIGH);
      digitalWrite(3, HIGH);
      //digitalWrite(4,HIGH );
      delay(100);
     }else if(((xValue > 18  && xValue < 25) || (yValue > 18 && yValue < 25) || (zValue >
18 && zValue < 25))&&(previous_condition != present_condition)){
      Serial.println("Checking the minor-car accident");
      digitalWrite(2, LOW);
      digitalWrite(3, HIGH);
      //digitalWrite(4,LOW);
      delay(100);
    }
     digitalWrite(2, LOW);
     digitalWrite(3, LOW);
     //digitalWrite(4, HIGH);
     delay(250);
}

void updateSerial()
{
 delay(100);
 while (Serial.available())
 {
   Serial.write(Serial.read());//Forward what Software Serial received to Serial Port
```

```
  }
}

void Accelerometerinit()
{
  for (int i = 0; i < samples; i++)
  {
    xsample += analogRead(x);
    ysample += analogRead(y);
    zsample += analogRead(z);
  }

  xsample /= samples;
  ysample /= samples;
  zsample /= samples;

  Serial.println(xsample);
  Serial.println(ysample);
  Serial.println(zsample);
}
```

6.3.2 LOGIN AND REGISTER ACTIVITY

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"

integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGg
FAW/dAiS6JXm" crossorigin="anonymous">
</head>
<style>
    nav {
        background-color: black;
        height: 64px;
        display: flex;
        align-items: center;
        justify-content: flex-end;
    }
</style>
```

```
<body>
  <nav class="w-100 ">
    <button class="btn btn-light mr-3 " onclick="RENDER('login')">Logout</button>
  </nav>

  <h1 class="text-center">Accident Detection</h1>

  <div class="container mt-5 w-50 signup d-none">
    <form class="form_info contact_info">
      <div class="form-group">
        <label for="name">Name</label>
        <input class="form-control" type="text" id="name">

      </div>
      <div class="form-group">
        <label for="password">Password</label>
        <input class="form-control" type="text" id="password">

      </div>
      <div class="form-group">
        <label for="phone">Phone</label>
        <input class="form-control" type="text" id="phone">
      </div>
      <div class="form-group">
        <label for="contact_name">Emergency Contact Name 1</label>
        <input class="form-control" type="text" id="contact_1"
name="contact_name_1">
      </div>
      <div class="form-group">
        <label for="emergency_contact">Emergency Contact Number </label>
        <input class="form-control" type="text" id="emergency_contact_1">
      </div>
      <div class="form-group">
        <label for="contact_name">Emergency Contact Name 2</label>
        <input class="form-control" type="text" id="contact_2"
name="contact_name_2">
      </div>
      <div class="form-group">
        <label for="emergency_contact">Emergency Contact Number</label>
        <input class="form-control" type="text" id="emergency_contact_2">
      </div>
      <div class="form-group">
        <label for="contact_name">Emergency Contact Name 3</label>
        <input class="form-control" type="text" id="contact_3"
name="contact_name_3">
```

```
        </div>
        <div class="form-group">
          <label for="emergency_contact">Emergency Contact Number</label>
          <input class="form-control" type="text" id="emergency_contact_3">
        </div>
        <div class="form-check">
          <input class="form-check-input" type="radio" name="vtype" id="2wheel"
value="2">
          <label class="form-check-label" for="2wheel">
            2 Wheeler
          </label>
        </div>
        <div class="form-check">
          <input class="form-check-input" type="radio" name="vtype" id="4wheel"
value="4">
          <label class="form-check-label" for="4Wheel">
            4 Wheeler
          </label>
        </div>

        <button type="submit" class="btn btn-primary">Sign Up</button>
        <p>have an account? <button class="btn btn btn-outline-success " type="button"
            onclick="RENDER('login')">Login</button></p>
      </form>
    </div>
    <div class="container mt-5 w-50 login">
      <form class="form_info sign_in">


        <div class="form-group">
          <label for="phone_num">Phone</label>
          <input class="form-control" type="text" id="phone_num" name="phone_num"
required>
        </div>
        <div class="form-group">
          <label for="password">Password</label>
          <input class="form-control" type="text" id="password" required>

        </div>

        <button type="submit" class="btn btn-primary">Login</button>
        <p>don't have an account? <button class="btn btn btn-outline-success "
type="button"
            onclick="RENDER('signup')">Sign
            UP</button></p>
```

```
        </form>
    </div>
    <div class="container mt-5 details d-none">
    </div>

    <script src="https://code.jquery.com/jquery-3.5.1.js"
        integrity="sha256-QWo7LDvxbWT2tbbQ97B53yJnYU3WhH/C8ycbRAkjPDc="
crossorigin="anonymous"></script>
    <!-- <button onclick="update_val('sathvik')">add</button> -->
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://www.gstatic.com/firebasejs/8.2.5/firebase-app.js"></script>

    <script src="https://www.gstatic.com/firebasejs/7.15.5/firebase-database.js"></script>
    <script src="https://www.gstatic.com/firebasejs/7.15.5/firebase-firestore.js"></script>

    <!-- TODO: Add SDKs for Firebase products that you want to use
     https://firebase.google.com/docs/web/setup#available-libraries -->

    <script>
        // Your web app's Firebase configuration
        var firebaseConfig = {
            apiKey: "AIzaSyABhq6khGslcqVV4HE4T9vyFXeuSFGxK7Y",
            authDomain: "accident-detection-syste-a1ad8.firebaseapp.com",
            databaseURL: "https://accident-detection-syste-a1ad8-default-rtdb.firebaseio.com",
            projectId: "accident-detection-syste-a1ad8",
            storageBucket: "accident-detection-syste-a1ad8.appspot.com",
            messagingSenderId: "890711908389",
            appId: "1:890711908389:web:19d4df6c4352b011b5dfcc",
            measurementId: "G-BD3CZ05WPK"

        };
        // Initialize Firebase
        firebase.initializeApp(firebaseConfig);
        const dbref = firebase.database().ref().child("phone");
        const db = firebase.firestore()
    </script>
    <script src="./main.js"></script>
</body>
</html>
```

6.3.3 LOGIN AND REGISTER ACTIVITY

```
dbref.on("value", (snapshot) => {
    let value = snapshot.val();
```

```
    console.log(value);

    set_contact(value)
});
const update_val = (name, contact_1, contact_2, contact_3, vtype) => {
    firebase
        .database()
        .ref()
        .set({
            name: name,
            contact_1: contact_1,
            contact_2: contact_2,
            contact_3: contact_3,
            vtype: vtype
        })
        .then(() => {
            console.log(name);
        });
}
$('.contact_info').on('submit', (e) => {
    e.preventDefault()
    const { name, password, phone, contact_name_1, contact_name_2, contact_name_3,
emergency_contact_1, emergency_contact_2, emergency_contact_3, vtype } = e.target
    SIGN_UP(name.value, password.value, phone.value, contact_name_1.value,
contact_name_2.value, contact_name_3.value, emergency_contact_1.value,
emergency_contact_2.value, emergency_contact_3.value, vtype.value)
    // SIGN_UP('croton', '007croton', "6096817869", "sathvik", "9502402016", '4')
})

$('.sign_in').on('submit', e => {
    e.preventDefault()
    const { phone_num, password } = e.target
    console.log(phone_num.value, password.value);
    SIGN_IN(phone_num.value, password.value)
    phone_num.value = ''
```

```
    password.value = ""
})
const set_contact = (phone) => {
    db.collection("userData")
        .get()
        .then((snapshot) => {
            snapshot.forEach((doc) => {
                data = doc.data()
                if (data.user_number == phone) {
                    console.log(data);
                    firebase
                        .database()
                        .ref()
                        .set({
                            phone: phone,
                            emergency_contact_1: data.emergency_contact_1,
                            emergency_contact_2: data.emergency_contact_2,
                            emergency_contact_3: data.emergency_contact_3,
                            name1: data.contact_name_1,
                            name2: data.contact_name_2,
                            name3: data.contact_name_3,
                            vtype: data.vtype

                        })
                        .then(() => {
                            console.log(name1, name2, name3);
                        });
                }
            })
        }).then(e => {
        })
}
const GET_DATA = (phone) => {
    db.collection("userData")
        .get()
```

```
        .then((snapshot) => {
            snapshot.forEach((doc) => {
                data = doc.data()
                console.log(phone);
                console.log(data.user_number);
                if (data.user_number == phone) {
                    console.log(data);
                    let html = `
                            <h1>Emergency Contact Details </h1>
                    <h4>Emergency Contact Name 1: ${data.contact_name_1}</h4>
                    <h4>Emergency Contact Number: ${data.emergency_contact_1}</h4>
                    <h4>Emergency Contact Name 2: ${data.contact_name_2}</h4>
                    <h4>Emergency Contact Number: ${data.emergency_contact_2}</h4>
                    <h4>Emergency Contact Name 3: ${data.contact_name_3}</h4>
                    <h4>Emergency Contact Number: ${data.emergency_contact_3}</h4>
                    <h4>Vehicle Type: ${data.vtype}</h4>
                    $('.details').html(html)
                    RENDER('details')
                }
            })
        }).then(e => {
        })

}
const SIGN_IN = (phone, password) => {
    db.collection("users")
        .get()
        .then((snapshot) => {
            snapshot.forEach((doc) => {
                let data = doc.data()
                console.log(phone, password);
                if (data.phone == phone && data.password == password) {
                    console.log("signed_in");
                    GET_DATA(phone);
                    set_contact(phone);
```

```
            }
        })
    }).then(e => {
    })
}
// GET_DATA()
// SIGN_IN('6096817869', '007croton')
const SIGN_UP = (name, password, phone, contact_name_1, contact_name_2,
contact_name_3, emergency_contact_1, emergency_contact_2, emergency_contact_3,
vtype) => {
    db.collection("users")
        .add({
            name,
            password,
            phone
        }).then(e => {
            db.collection('userData').add({ contact_name_1, contact_name_2, contact_name_3,
emergency_contact_1, emergency_contact_2, emergency_contact_3, vtype, user_number:
phone })
        }).then(() => {
            RENDER('login')
        })

}
// SIGN_UP('croton', '007croton', "6096817869", "sathvik", "9502402016", '4')
const RENDER = (page) => {
    switch (page) {
        case "login":
            $('.signup').addClass('d-none')
            $('.details').addClass('d-none')
            $('.login').removeClass('d-none')
            break;
        case "signup":
            $('.login').addClass('d-none')
            $('.details').addClass('d-none')
```

```
            $('.signup').removeClass('d-none')
            break;
        case "details":
            $('.login').addClass('d-none')
            $('.signup').addClass('d-none')
            $('.details').removeClass('d-none')
            break
        default:
            break;
    }
}
```

6.3.4 GPS SENSOR, GSM 800A, NODEMCU, LCD 16X2 DISPLAY ACTIVITY

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#define input1 13
#define input2 15
#define buttoninput 10

LiquidCrystal_I2C lcd(0x27, 16, 2);
TinyGPSPlus gps;  // The TinyGPS++ object
SoftwareSerial mySerial(0, 2); //SIM900 Tx & Rx is connected to Arduino #0 & #2
SoftwareSerial gpsSerial(14, 12);//tx and rx is 14 and 12
const char* ssid = "Galaxy";
const char* password = "geethasingh@10";
#define FIREBASE_HOST "accident-detection-syste-a1ad8-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "gSt3r7KMfvCeAggbSCCGO0CO0u9KJOPacwogUzrI"
String
Emergencycontact1,Emergencycontact2,Emergencycontact3,name1,name2,name3,phone,v
type;
float latitude , longitude;
```

```
int year , month , date, hour , minute , second;
String date_str , time_str , lat_str , lng_str;
int pm,count=0;
int intp1,intp2,intp3,intp4,buttoncount=0;
bool bikeminor = false;
bool carminor = false;
bool bikemajor = false;
bool carmajor = false;
String minorbikestr,minorcarstr,majorcarstr,majorbikestr;
void setup() {
  Serial.begin(115200);
  mySerial.begin(9600);
   Serial.println("Initializing the GSM init function.....");
  mySerial.println("AT"); //Handshaking with SIM900
  updateSerial();
  // Start the software serial port at the GPS's default baud
  gpsSerial.begin(9600);
  minorbikestr = "bike met with an minor-accident";
  minorcarstr = "car met with an minor-accident";
  majorbikestr = "bike met with an major-accident";
  majorcarstr = "car met with an major-accident";
  lcd.begin();
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("7901427367");
  pinMode(input1 , INPUT);
  pinMode(input2 , INPUT);
  pinMode(buttoninput , INPUT);
  Serial.println();
  wifiConnect();
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  delay(100);
}
```

```
void loop() {
 firebasedata();
 while (gpsSerial.available() > 0){
  if (gps.encode(gpsSerial.read()))
  {
   if(lat_str.length()<=1 && lng_str.length()<=1){
    displayInfo();
   }
  }
 }
 if (millis() > 28000 && gps.charsProcessed() < 10)
 {
  Serial.println("No GPS detected");
  //while(true);
 }
 //Serial.println("Read digital input: ");
 intp1 = digitalRead(input1);
 intp2 = digitalRead(input2);
 if(vtype == "4"){
  if(intp1==0 && intp2==1){
  Serial.println("car met with an minor-accident");
  carminor = true;
 }else if(intp1==1 && intp2==1){
   Serial.println("car met with an major-accident");
  carmajor = true;
  scrollText(1, majorcarstr, 320, 16);
   sendsms(Emergencycontact1);
    delay(20000);
    sendsms(Emergencycontact2);
    delay(20000);
    sendsms(Emergencycontact3);
    delay(20000);
 }
 }
 if(vtype == "2"){
```

```
 if(intp1==0 && intp2==1){
  Serial.println("bike met with an minor-accident");
  bikeminor = true;
 }else if(intp1==1 && intp2==1){
   Serial.println("Bike met with an major-accident");
  bikemajor = true;
  scrollText(1, majorbikestr, 320, 16);
   sendsms(Emergencycontact1);
    delay(20000);
    sendsms(Emergencycontact2);
    delay(20000);
    sendsms(Emergencycontact3);
    delay(20000);
 }
}
if(intp1==0 && intp2==0){
  Serial.println("NO Accident Occured!!!!");
}
if(bikeminor || carminor){
  intp4 = digitalRead(buttoninput);
  Serial.print(" button value :");
  Serial.println(intp4);
  if(intp4 == HIGH){
    Serial.println(" button pressed!!!!!!");
    if(bikeminor){
      scrollText(1, minorbikestr, 320, 16);
      sendsms(Emergencycontact1);
      delay(20000);
      sendsms(Emergencycontact2);
      delay(20000);
      sendsms(Emergencycontact3);
      delay(20000);
    }
    if(carminor){
      scrollText(1, minorcarstr, 250, 16);
```

```
        sendsms(Emergencycontact1);
        delay(20000);
        sendsms(Emergencycontact2);
        delay(20000);
        sendsms(Emergencycontact3);
        delay(20000);
      }
    }
  }
}

void updateSerial()
{
  delay(100);
  while (Serial.available())
  {
    mySerial.write(Serial.read());//Forward what Serial received to Software Serial Port
  }
  while(mySerial.available())
  {
    Serial.write(mySerial.read());//Forward what Software Serial received to Serial Port
  }
}

void displayInfo(){
    if (gps.location.isValid())
      {
        latitude = gps.location.lat();
        lat_str = String(latitude , 6);
        Serial.println(lat_str);
        longitude = gps.location.lng();
        lng_str = String(longitude , 6);
         Serial.println(lng_str);
      }else{
        Serial.println("Location: Not Available");
```

```
}
if (gps.date.isValid())
{
  date_str = "";
  date = gps.date.day();
  month = gps.date.month();
  year = gps.date.year();
  if (date < 10)
    date_str = '0';
  date_str += String(date);
  date_str += " / ";
  if (month < 10)
    date_str += '0';
  date_str += String(month);
  date_str += " / ";
  if (year < 10)
    date_str += '0';
  date_str += String(year);
  Serial.println(date_str);
}else{
  Serial.println("Date: Not Available");
}
if (gps.time.isValid())
{
  time_str = "";
  hour = gps.time.hour();
  minute = gps.time.minute();
  second = gps.time.second();
  minute = (minute + 30);
  if (minute > 59)
  {
    minute = minute - 60;
    hour = hour + 1;
  }
  hour = (hour + 5) ;
```

```
      if (hour > 23)
        hour = hour - 24;
      if (hour >= 12)
        pm = 1;
      else
        pm = 0;
      hour = hour % 12;
      if (hour < 10)
        time_str = '0';
      time_str += String(hour);
      time_str += " : ";
      if (minute < 10)
        time_str += '0';
      time_str += String(minute);
      time_str += " : ";
      if (second < 10)
        time_str += '0';
      time_str += String(second);
      if (pm == 1)
        time_str += " PM ";
      else
        time_str += " AM ";
     Serial.println(time_str);
     }else{
      Serial.println("Time: Not Available");
     }
      Serial.println();
      Serial.println();
      delay(100);
}

void sendsms(String num1)
{
  mySerial.println("AT+CMGF=1\r"); // Configuring TEXT mode
 updateSerial();
```

```
  String number =  "AT+CMGS=\"" + num1 + "\"";
   mySerial.println(number);//change ZZ with country code and xxxxxxxxxxx with phone number to sms
  updateSerial();
  if ((gps.location.isValid())&&(gps.time.isValid()))
  {
    mySerial.println("Accident occured Please help me!!!"); //text content
    updateSerial();
            mySerial.print("http://maps.google.com/maps?&z=15&mrt=yp&t=k&q=");    //text content
    updateSerial();
    mySerial.print(lat_str);
    updateSerial();
    mySerial.print("+");             //28.612953, 77.231545   //28.612953,77.2293563
    updateSerial();
    mySerial.println(lng_str);
    updateSerial();
//    mySerial.println("At time "); //text content
//    updateSerial();
//    mySerial.println(time_str); //text content
//    updateSerial();
    mySerial.write(26);
  }
  else
  {
    mySerial.print("Accident occured Please help me!!!"); //text content
    updateSerial();
    mySerial.write(26);
  }
  //satify = 1;
  Serial.println("SMS Sent");
  bikeminor = false;
  bikemajor = false;
  carminor = false;
  carmajor = false;
```

```
  lcd.clear();
  lcd.setCursor(0, 0);
 lcd.print("7901427367");
 //delay(1000);
}

void firebasedata(){
if((Emergencycontact1.length()<=1)&&(Emergencycontact2.length()<=1)&&(Emergency
contact3.length()<=1)&&(vtype.length()<=1)){
  Serial.print("Emergency-contact-1: ");
 Emergencycontact1 = Firebase.getString("emergency_contact_1");
 Serial.println(Emergencycontact1);
 delay(500);
 Serial.print("Emergency-contact-2: ");
 Emergencycontact2 = Firebase.getString("emergency_contact_2");
 Serial.println(Emergencycontact2);
 delay(500);
 Serial.print("Emergency-contact-3: ");
 Emergencycontact3 = Firebase.getString("emergency_contact_3");
 Serial.println(Emergencycontact3);
 delay(500);
 Serial.print("vtype: ");
 vtype = Firebase.getString("vtype");
 Serial.println(vtype);
 delay(500);
 if (Firebase.failed()) {
     Serial.print("setting /message failed:");
     Serial.println(Firebase.error());
     return;
 }
 }
if(WiFi.status() != WL_CONNECTED)
{
 wifiConnect();
}
```

```
}

void wifiConnect()
{
  WiFi.begin(ssid,password);             // Connect to the network
  Serial.print("Connecting to ");
  delay(200);
  Serial.print(ssid);
  Serial.println(" ...");
  int teller = 0;
  while (WiFi.status() != WL_CONNECTED)
  {                                      // Wait for the Wi-Fi to connect
    delay(200);
    Serial.print(++teller); Serial.print(' ');
  }
  Serial.println('\n');
  Serial.println("Connection established!");
  Serial.print("IP address:\t");
  Serial.println(WiFi.localIP());        // Send the IP address of the ESP8266 to the computer
}


void scrollText(int row, String message, int delayTime, int lcdColumns) {
  for (int i=0; i < lcdColumns; i++) {
    message = " " + message;
  }
  message = message + " ";
  for (int pos = 0; pos < message.length(); pos++) {
    lcd.setCursor(0, row);
    lcd.print(message.substring(pos, pos + lcdColumns));
    delay(delayTime);
  }
}
```

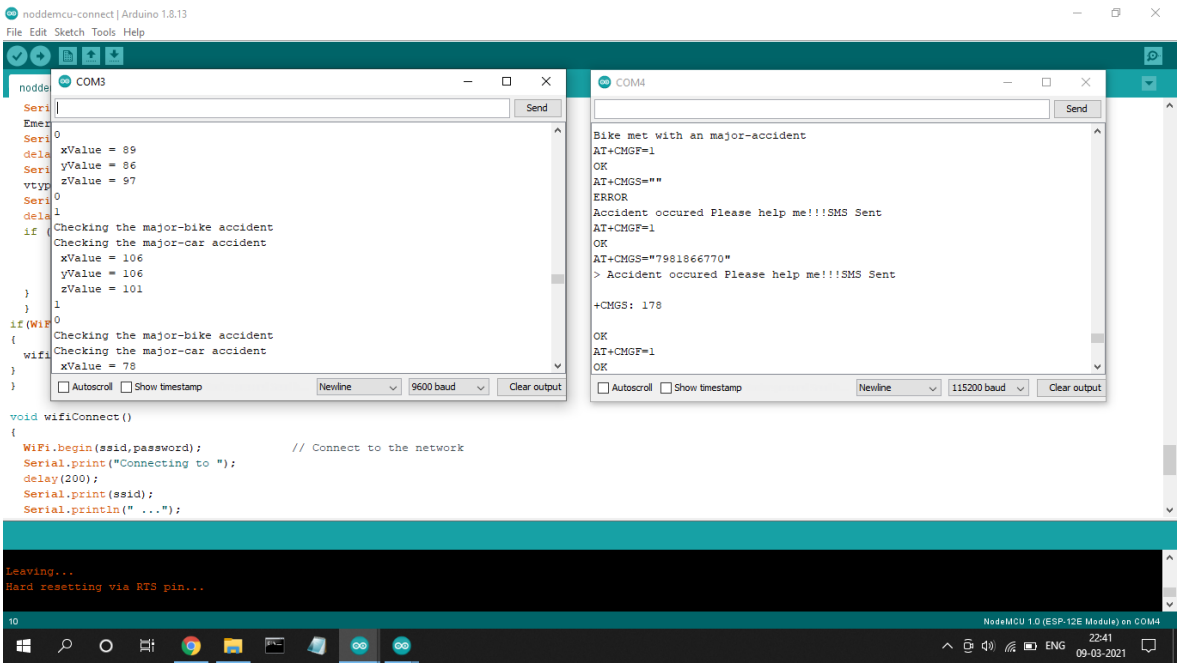
6.4 WORKING

6.4.1 ACCIDENT DETECTION BY ACCELEROMETER

When we are ready with our hardware-connections after uploading the program into Arduino, we can install it in our vehicle and power it up. Now whenever there is an accident, the car gets tilt and accelerometer changes his axis values. These values read by Arduino and checks if any change occurs in any axis.If any change occurs arduino sends the signal to the nodemcu. Nodemcu analysis what type of accident has occured. If the major accident accident is occured then nodemcu reads coordinates by extracting $GPGGA String from GPS module data and send SMS to the predefined number to the police or ambulance or family member with the location coordinates of accident place. The message also contains a Google Map link to the accident location, so that location can be easily tracked. When we receive the message then we only need to click the link and we will redirect to the Google map and then we can see the exact location of the vehicle.

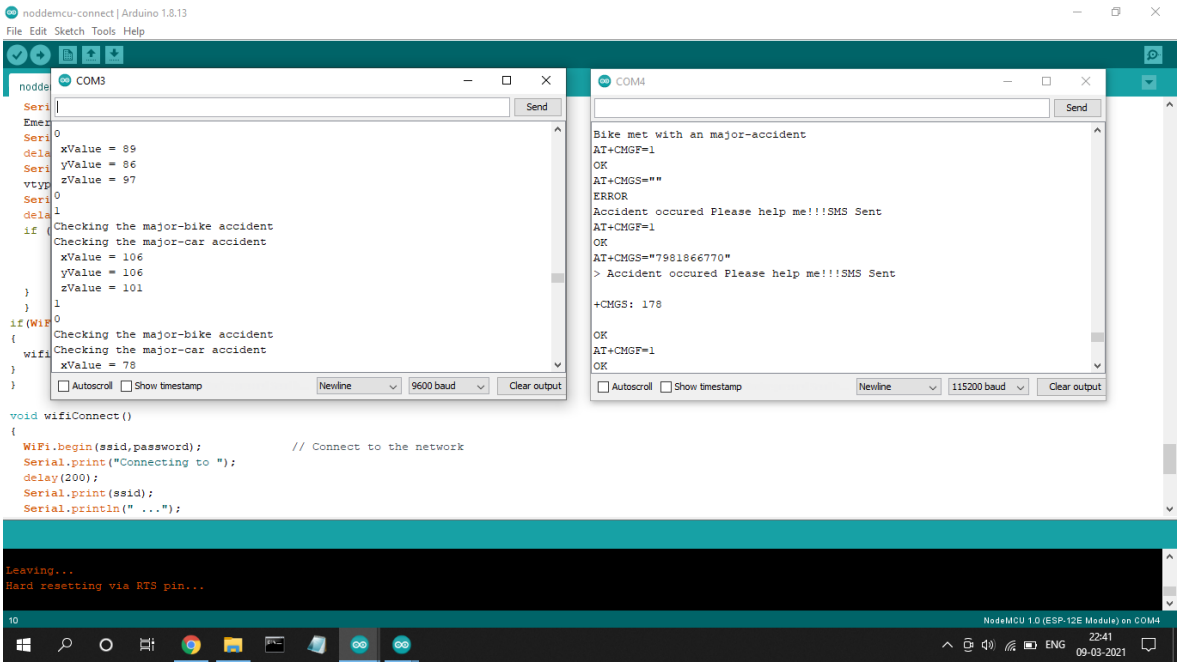
6.4.2 TWO WHELLER MINOR ACCIDENT

whenever there is an accident, the car gets tilt and accelerometer changes his axis values.These values read by Arduino and checks if any change occurs in any axis.If any change occurs arduino sends the signal to the nodemcu. Nodemcu analysis what type of accident has occured. If the minor accident accident is occured then nodemcu reads coordinates by extracting $GPGGA String from GPS module data and send SMS to the predefined number to the police or ambulance or family member with the location coordinates of accident place. The message also contains a Google Map link to the accident location, so that location can be easily tracked. When we receive the message then we only need to click the link and we will redirect to the Google map and then we can see the exact location of the vehicle.



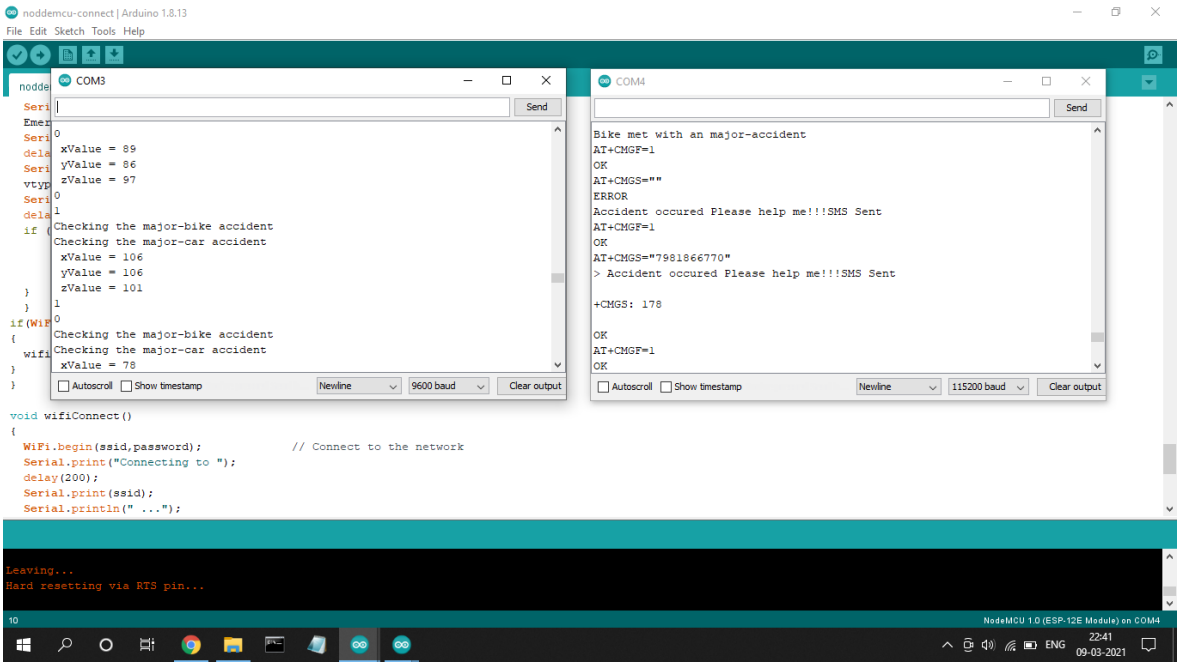
6.4.2 TWO WHELLER MAJOR ACCIDENT

whenever there is an accident, the car gets tilt and accelerometer changes his axis values.These values read by Arduino and checks if any change occurs in any axis.If any change occurs arduino sends the signal to the nodemcu. Nodemcu analysis what type of accident has occured. If the major accident accident is occured then nodemcu reads coordinates by extracting $GPGGA String from GPS module data and send SMS to the predefined number to the police or ambulance or family member with the location coordinates of accident place. The message also contains a Google Map link to the accident location, so that location can be easily tracked. When we receive the message then we only need to click the link and we will redirect to the Google map and then we can see the exact location of the vehicle.



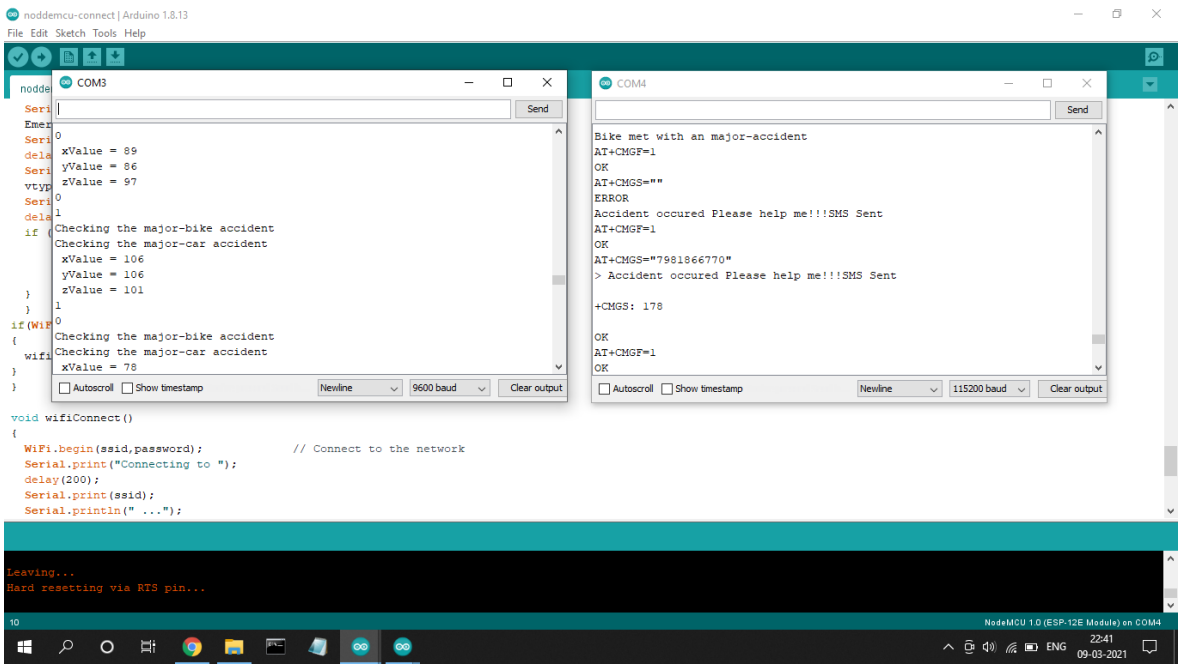
6.4.2 FOUR WHELLER MINOR ACCIDENT

whenever there is an accident, the car gets tilt and accelerometer changes his axis values.These values read by Arduino and checks if any change occurs in any axis.If any change occurs arduino sends the signal to the nodemcu. Nodemcu analysis what type of accident has occured. If the minor accident accident is occured then nodemcu reads coordinates by extracting $GPGGA String from GPS module data and send SMS to the predefined number to the police or ambulance or family member with the location coordinates of accident place. The message also contains a Google Map link to the accident location, so that location can be easily tracked. When we receive the message then we only need to click the link and we will redirect to the Google map and then we can see the exact location of the vehicle.



6.4.2 FOUR WHELLER MAJOR ACCIDENT

whenever there is an accident, the car gets tilt and accelerometer changes his axis values.These values read by Arduino and checks if any change occurs in any axis.If any change occurs arduino sends the signal to the nodemcu. Nodemcu analysis what type of accident has occured. If the major accident accident is occured then nodemcu reads coordinates by extracting $GPGGA String from GPS module data and send SMS to the predefined number to the police or ambulance or family member with the location coordinates of accident place. The message also contains a Google Map link to the accident location, so that location can be easily tracked. When we receive the message then we only need to click the link and we will redirect to the Google map and then we can see the exact location of the vehicle.



7. USER SCREENS

7.1 LOGIN SCREEN

A Login screen is the screen where the user is asked to fill his credential like phone number, name etc.. to authenticate the user, Also, to validate the user already exist or not.

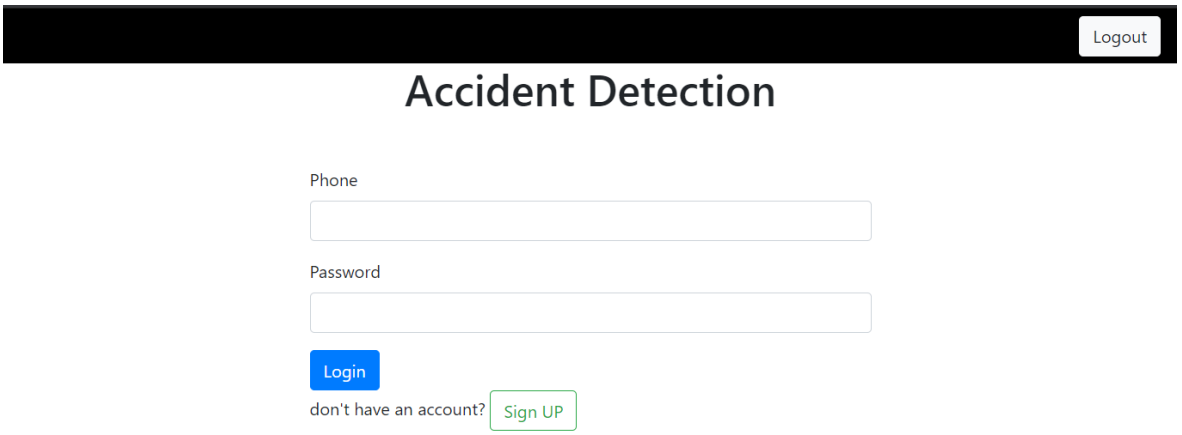
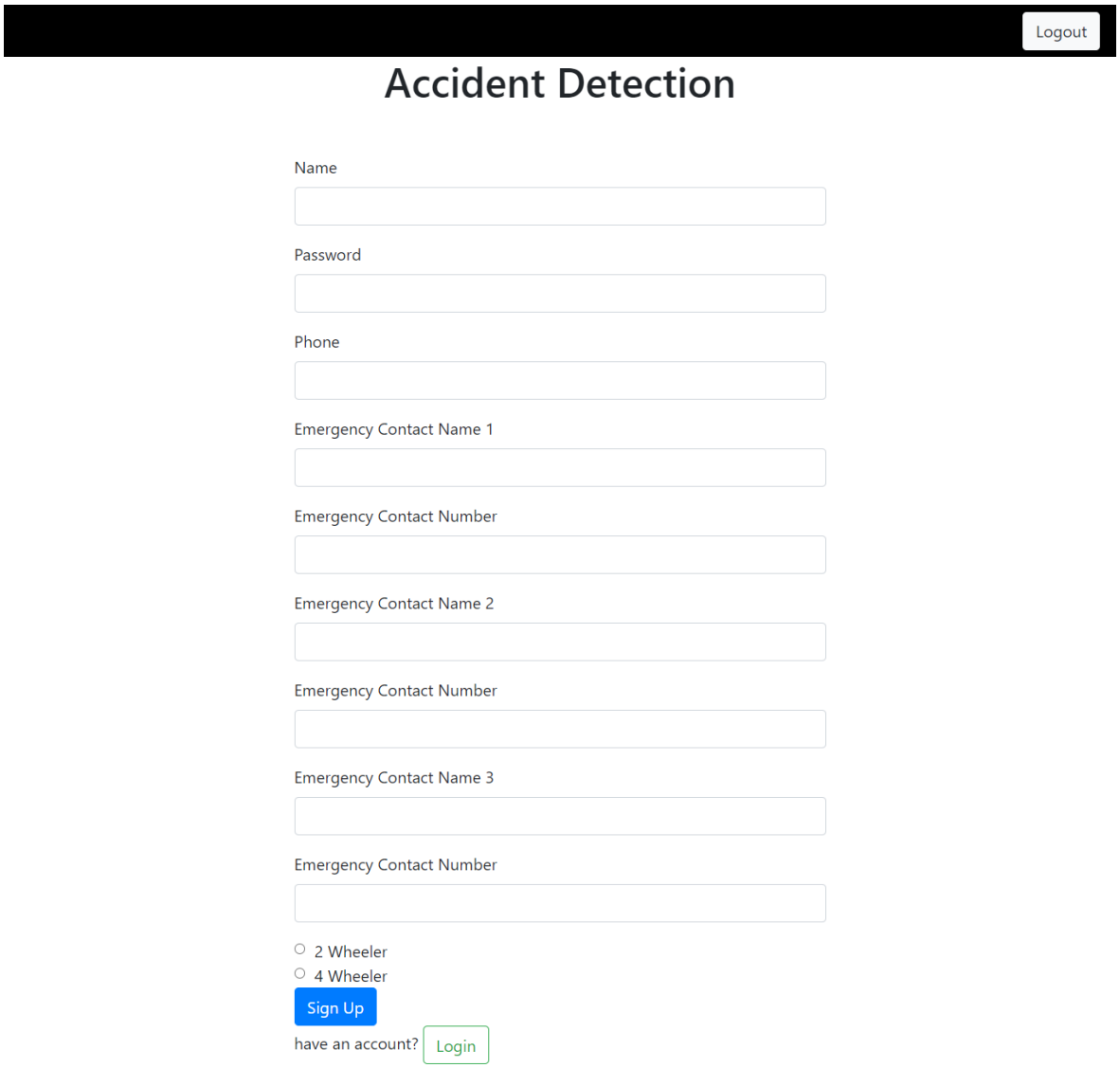


Figure 7.1 Login Screen

7.2 REGISTER SCREEN

A Register screen is the screen where the user is asked to fill the form like username,password,phone number,emergency contact numbers etc. from a user to create a web-based account or profile. A user registration system generally asks a user to create a username and password.



7.2 SERIAL MONITOR SCREEN

The Arduino IDE has a feature that can be a great help for controlling Arduino from your computer's keyboard. The serial monitor is a separate pop-up window that acts as a separate terminal that communicates by receiving and sending serial data.

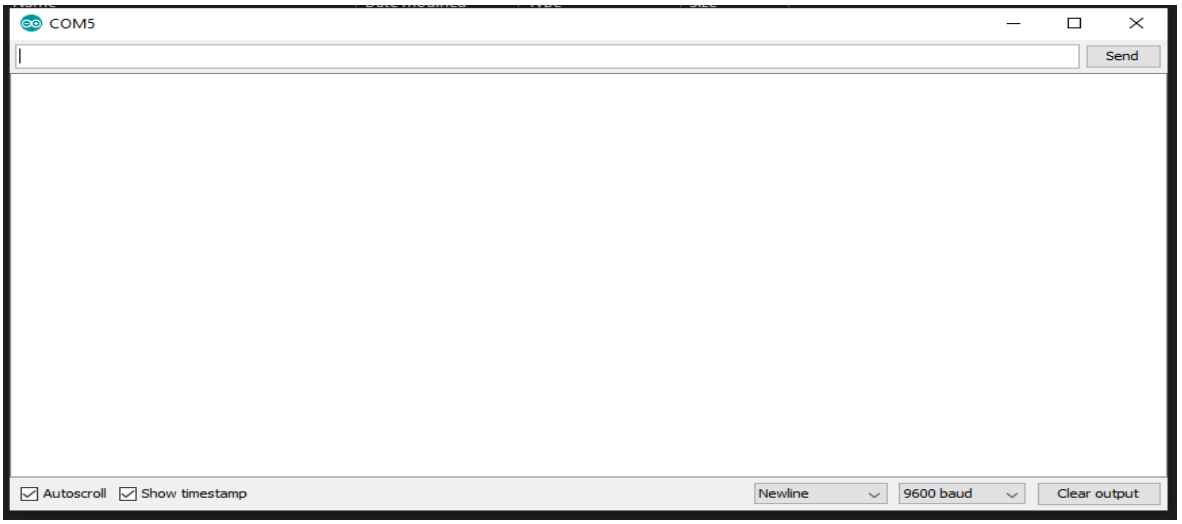


Figure 7.2 Serial monitor screen

7.3 MESSAGE SCREEN

Message screen is a screen where the accident occured information along with location of the accident is displayed

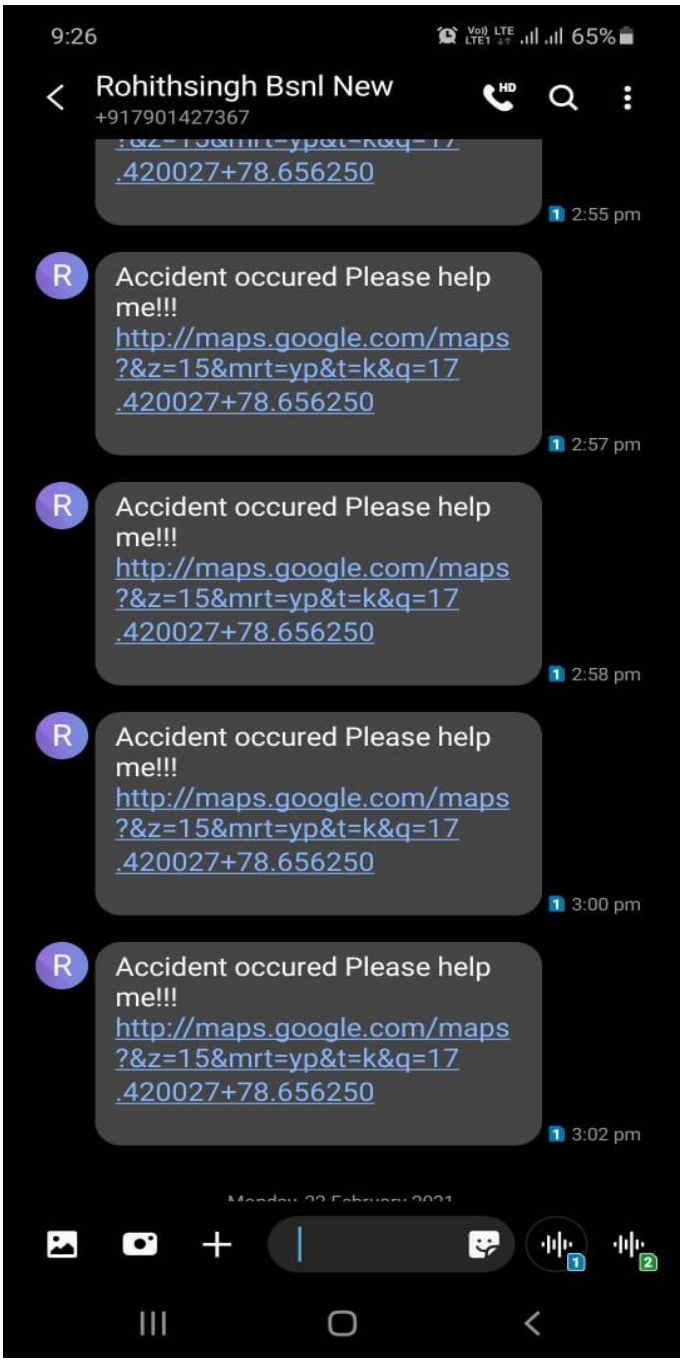


Figure 7.3 Message screen

7.4 MAPS SCREEN

Maps screen is the screen where it shows the extract location of the accident in the google maps.

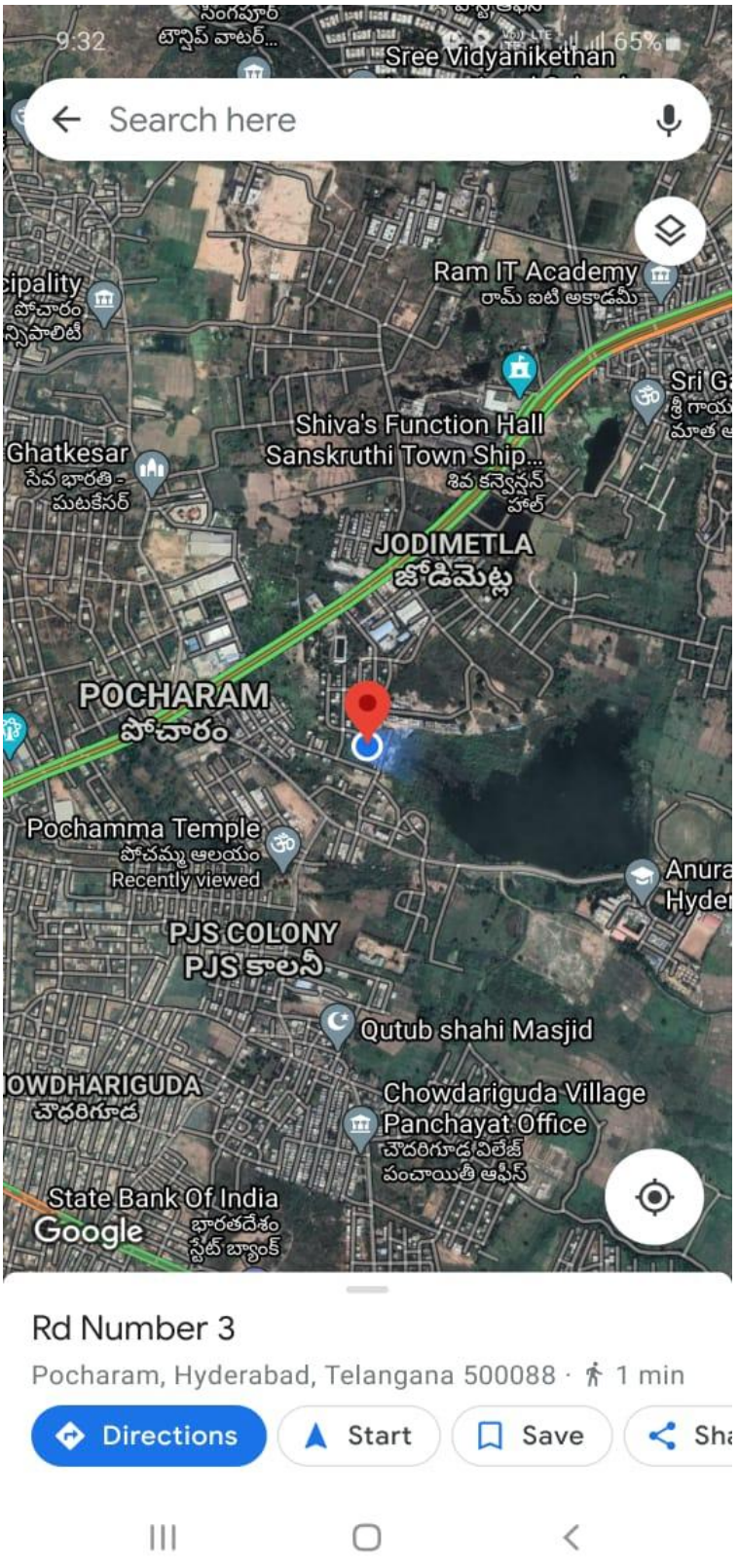


Figure 7.4 Maps Screen

8. TESTING

8.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision b$_r$anches and internal code flow should be validated. It is the testing of individual o$_w$= units of the application . It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

8.1.1.1 GPS SENSOR

1. Connect the antenna to the gps sensor.
2. Solder the header pins as needed to the GPS Module
3. Connect the Tx pin,Rx pin, Gnd pin and Vin pin to the Arduino uno Digital Pins, Gnd pins and 5V pin by using Jumper wires.
4. Connect the Usb-cable to your computer and arduino uno microcontroller.
5. Upload the code to arduino uno microcontroller.
6. To make sure your’re receiving some sort of signal, open with the Arduino IDE and set it to 9600 baudrate. If the GPS is receiving or outputting a serial signal. You will see some sort of “ proof of life” indication.
7. When the GPS Sensor has Established the connection with the satellites, The led will blink Every Sec.

8.1.1.2 GPS SENSOR CODE

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include <ESP8266WiFi.h>
TinyGPSPlus gps;  // The TinyGPS++ object
SoftwareSerial ss(14, 12); // The serial connection to the GPS device

const char* ssid = "iotproject";
const char* password = "123456789";
```

```
float latitude , longitude;
int year , month , date, hour , minute , second;
String date_str , time_str , lat_str , lng_str;
int pm;

WiFiServer server(80);
void setup()
{
  Serial.begin(115200);
  ss.begin(9600);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  delay(2000);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  delay(5000);
  server.begin();
  Serial.println("Server started");
 delay(2000);
  // Print the IP address
  Serial.println(WiFi.localIP());
  delay(10000);
}

void loop()
{
```

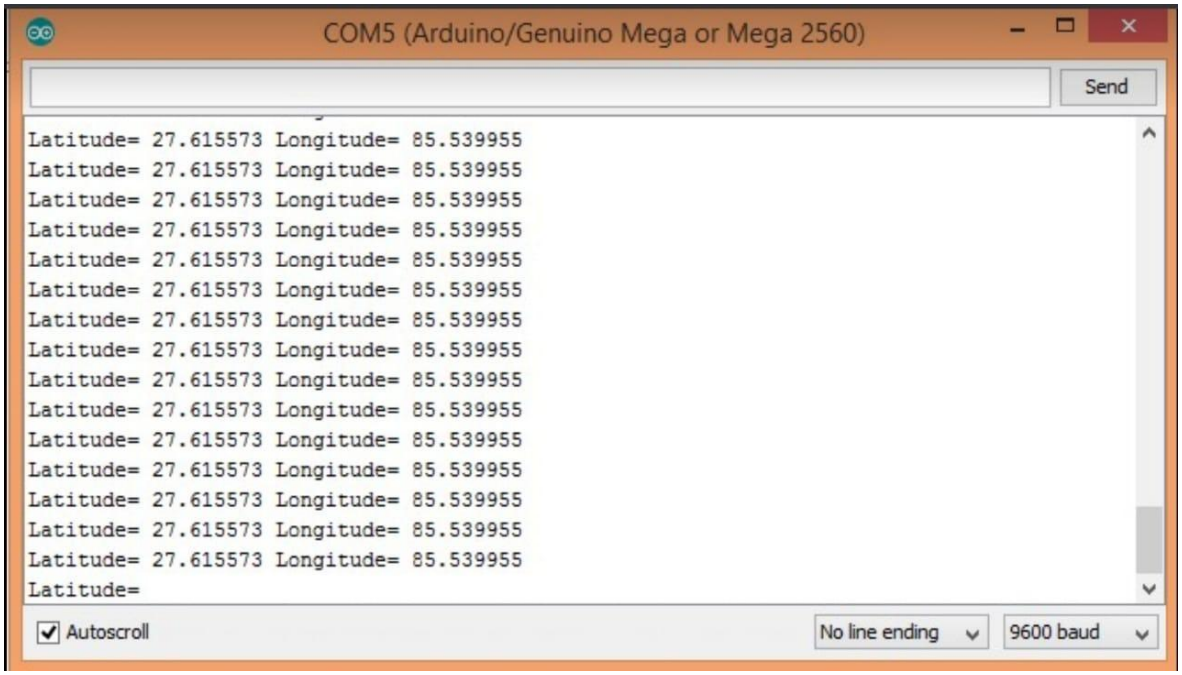
```
  while (ss.available() > 0)
    if (gps.encode(ss.read()))
    {
      displayInfo();
    }
     // If 5000 milliseconds pass and there are no characters coming in
  // over the software serial port, show a "No GPS detected" error
  if (millis() > 25000 && gps.charsProcessed() < 10)
  {
    Serial.println("No GPS detected");
    //while(true);
  }
  delay(100);

}

void displayInfo(){
  Serial.print("lat_str = ");
  Serial.println(lat_str.length());
  Serial.print("lng_str = ");
  Serial.println(lng_str.length());
  if (gps.location.isValid())
      {
        latitude = gps.location.lat();
        lat_str = String(latitude , 6);
        Serial.println(lat_str);
        Serial.print("lat_str = ");
        Serial.println(lat_str.length());
        longitude = gps.location.lng();
        lng_str = String(longitude , 6);
         Serial.println(lng_str);
         Serial.print("lng_str = ");
        Serial.println(lng_str.length());
      }else{
        Serial.println("Location: Not Available");
```

```
}

if (gps.date.isValid())
{
 date_str = "";
 date = gps.date.day();
 month = gps.date.month();
 year = gps.date.year();

 if (date < 10)
  date_str = '0';
 date_str += String(date);

 date_str += " / ";

 if (month < 10)
  date_str += '0';
 date_str += String(month);

 date_str += " / ";

 if (year < 10)
  date_str += '0';
 date_str += String(year);
 Serial.println(date_str);
}else{
 Serial.println("Date: Not Available");
}

if (gps.time.isValid())
{
 time_str = "";
 hour = gps.time.hour();
 minute = gps.time.minute();
 second = gps.time.second();
```

```
minute = (minute + 30);
if (minute > 59)
{
  minute = minute - 60;
  hour = hour + 1;
}
hour = (hour + 5) ;
if (hour > 23)
  hour = hour - 24;

if (hour >= 12)
  pm = 1;
else
  pm = 0;

hour = hour % 12;

if (hour < 10)
  time_str = '0';
time_str += String(hour);

time_str += " : ";

if (minute < 10)
  time_str += '0';
time_str += String(minute);

time_str += " : ";

if (second < 10)
  time_str += '0';
time_str += String(second);

if (pm == 1)
```

```
    time_str += " PM ";
   else
    time_str += " AM ";
  Serial.println(time_str);
 }else{
  Serial.println("Time: Not Available");
 }
 Serial.println();
 Serial.println();
 delay(100);
}
```

8.1.1.3 GPS SENSOR CODE OUTPUT



8.1.2.1 GSM SENSOR

1. Connect the antenna to the GSM Sensor.
2. Solder header pins as needed to the GSM Module.
3. Insert any 2G/3G simcard into the slot provided in the GSM Module.
4. Connect to the 5V adopter.
5. Wait untill it establishes the connection with the nearby network.Once it is established the connection , the led will blink every 3 secs.
6. Connect Tx pin, Rx pin, Gnd pin to the Arduino uno Digital pins, Gnd pins by using Jumper wires.

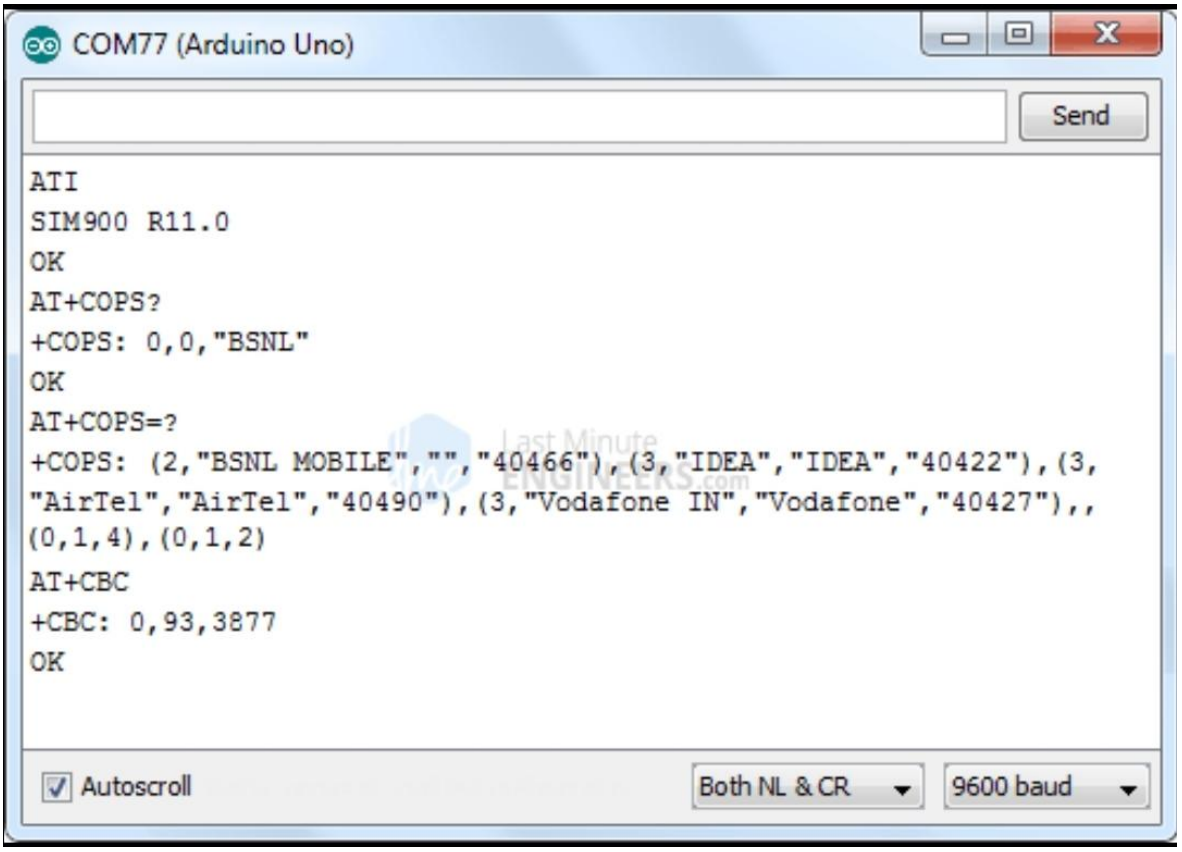
7. Connect the usb-cable to your computer and arduino uno microcontroller.
8. Upload the code to Arduino uno microcontroller.
9. To make sure your're receiving some sort of signal, open with the Arduino IDE and set it to 9600 baudrate. If the GPS is receiving or outputting a serial signal. You will see some sort of " proof of life" indication.

8.1.2.2 GSM MODULE CODE

```
#include <SoftwareSerial.h>

//Create software serial object to communicate with SIM900
SoftwareSerial mySerial(0, 2); //SIM900 Tx & Rx is connected to Arduino #0 & #2

void setup()
{
  //Begin serial communication with Arduino and Arduino IDE (Serial Monitor)
  Serial.begin(115200);

  //Begin serial communication with Arduino and SIM900
  mySerial.begin(9600);

  Serial.println("Initializing...");
  delay(1000);

  mySerial.println("AT"); //Handshaking with SIM900
  updateSerial();
  mySerial.println("AT+CSQ"); //Signal quality test, value range is 0-31 , 31 is the best
  updateSerial();
   mySerial.println("AT+CCID"); //Read SIM information to confirm whether the SIM is plugged
  updateSerial();
  mySerial.println("AT+CREG?"); //Check whether it has registered in the network
  updateSerial();
  mySerial.println("ATI"); //Get the shield name and revision
  updateSerial();
  mySerial.println("AT+COPS=?"); //Return the list of operators present in the network
  updateSerial();
```

```
  mySerial.println("AT+CBC"); //will return the lipo battery state. The second number is the % full (in this case its 93%) and the third number is the actual voltage in mV (in this case, 3.877 V)
  updateSerial();
}

void loop()
{
  updateSerial();
}

void updateSerial()
{
  delay(500);
  while (Serial.available())
  {
    mySerial.write(Serial.read());//Forward what Serial received to Software Serial Port
  }
  while(mySerial.available())
  {
    Serial.write(mySerial.read());//Forward what Software Serial received to Serial Port
  }
}
```

8.1.2.3 GSM MODULE OUTPUT



8.1.3.1 ACCELEROMETER (ADXL335)

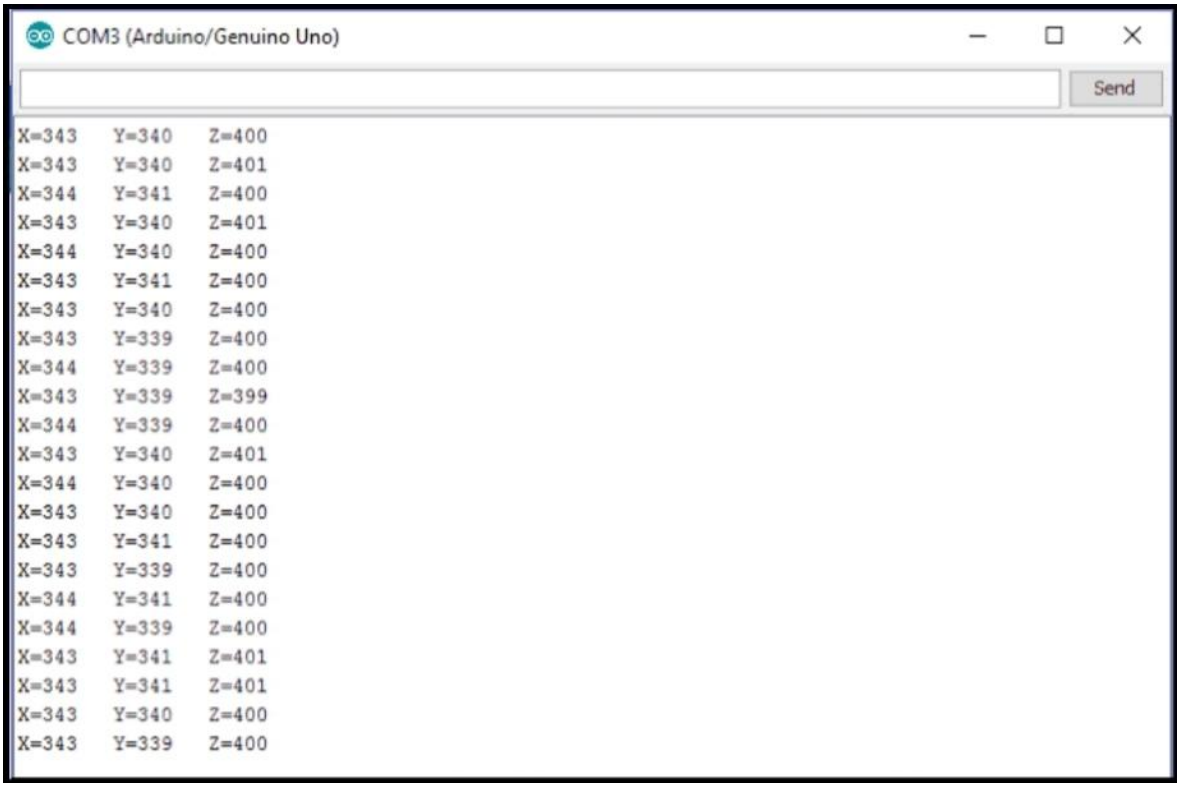
1. Solder header pins as needed to the Accelerometer sensor.
2. Connect x-axis,y-axis,z-axis Gnd pin and Vin of accelerometer to Arduino uno Analog pins, GND pins and 5V pins by Jumper wires.
3. Connect the USB-cable to your computer and arduino uno microcontroller.
4. Upload the code to arduino uno microcontroller.
5. To make sure your're receiving some sort of signal,open the port and set it to 9600 baud rate.If the accelerometer is outputting a serial signal, you will see sort of "proof of life" indication.

8.1.3.2 ACCELEROMETER (ADXL335) CODE

```
int xpin = A0;
int ypin = A1;
int zpin = A2;
int xvalue;
int yvalue;
int zvalue;
```

```
void setup()
{
  Serial.begin(9600);          // initialize the serial communications:
}
void loop()
{
 xvalue = analogRead(xpin);
 int x = map(xvalue, 275, 1000, -100, 100);
 float xg = (float)x/(-100.00);
 Serial.print(xg);
 Serial.print("g   ");
 yvalue = analogRead(ypin);
 int y = map(yvalue, 281, 1000, -100, 100);
 float yg = (float)y/(-100.00);
 Serial.print("\t");
 Serial.print(yg);
 Serial.print("g   ");
 zvalue = analogRead(zpin);
 int z = map(zvalue, 228, 1000, -100, 100);
 float zg = (float)z/(100.00);
 Serial.print("\t");
 Serial.print(zg);
 Serial.println("g   ");
 delay(100);
}
```

8.1.3.2 ACCELEROMETER (ADXL335) OUTPUT



8.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfied, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

8.3 TEST CASE SCENARIOS

Testcase ID	Testcase Name	Input	Expected	Actual Output
1	Existing user logging in	Fill Phone Number	Login successful	Login successful
2	Existing user logging in	Fill Wrongly Formatted Phone Number	Login Failed	Login Failed
3	New/Unregistered user logging in	Fill Phone Number	Login Failed	Login Failed
4	Login as New User	Fill Phone number and Password	Login successful	Login successful
5	Login as New User	Fill wrongly Formatted Email ID	Login Failed	Login Failed
6	GPS testing indoors	GPS Sensor connected to nodemcu	No location found	No location found
7	GPS testing outdoors	GPS Sensor connected to nodemcu	Location found	Location found
8	GSM testing indoors	GSM Sensor connected to nodemcu	Connected to network	Connected to network
9	GSM testing outdoors	GSM Sensor connected to nodemcu	Connected to network	Connected to network
10	Nodemcu wifi testing	Wrong wifi name and password	Not connected to wifi	Not connected to wifi
11	Nodemcu wifi testing	Correct wifi and password	connected to wifi	connected to wifi
12	LCD Display testing	3.3V Provided	No Output	No Output

13	LCD Display testing	5.5V Provided	Output is Displayed	Output is Displayed
14	Minor three wheeler accident occured	Accelerometer, vibration sensor and push button input	Message is send	Message is send
15	Minor two wheeler accident occured	Accelerometer, vibration sensor and push button input	Message is send	Message is send
16	Major three wheeler accident occured	Accelerometer, vibration sensor input	Message is send	Message is send
17	Major two wheeler accident occured	Accelerometer, vibration sensor input	Message is send	Message is send
18	No accident occured	Accelerometer, vibration sensor input	No message is send	No message is send

9. FUTURE ENHANCEMENTS

Future scope of this system is to control the accidents and providing useful details about the accidental vehicle, thereby reducing the rate of accidents taking place due to drunken driving.It is very difficult to find that an accident has occurred and to find the position where it the accident occurred. (Global System for Mobile Communication)GSM and Global positioning systems are used. The GPS based accident identification module contains a Micro Electro Mechanical System(MEMS),vibrating sensor, firesensor,infrared sensor and a GPS module connected to the microcontroller.

10. CONCLUSION

The main idea of this system is used to detect any hazardous accidents and to automate the emergency services in a faster manner. As we know that there is an exponential growth of usage of number of vehicles which can increase the fatality rate due to accidents. Hence there must be a device which could be installed in a vehicle and sends the message to assistance people in a short time when an accident happens. This system is a device which can save valuable lives with at most fast. A Pushbutton is also provided which can hasthe capability to terminate the sending of message to rescue teams such as police, ambulance and predefined personal's mobile numbers to save their time The proposed system deals with the detection of the accidents. But this can be extended by providing medication to the victims at the accident spot. By increasing the technology we can also avoid accidents by providing alerts systems that can stop the vehicle to overcome the accidents

11. REFERENCES

1. https://circuitdigest.com/microcontroller-projects/interfacing-gps-with-nodemcu-esp12
2. https://lastminuteengineers.com/sim900-gsm-shield-arduino-tutorial/
3. https://lastminuteengineers.com/adxl335-accelerometer-arduino-tutorial/
4. https://circuitdigest.com/microcontroller-projects/arduino-sw-420-vibration-sensor-module-interfacing
5. https://randomnerdtutorials.com/esp32-esp8266-i2c-lcd-arduino-ide/
6. https://www.youtube.com/watch?v=nLzC0-VaqDs
7. https://www.youtube.com/watch?v=ZZ5JvSA-Ed8&t=460s
8. https://create.arduino.cc/projecthub/albertoz/vibration-sensor-module-c88067
9. https://create.arduino.cc/projecthub/akshayjoseph666/interface-i2c-16x2-lcd-with-arduino-uno-just-4-wires-273b24
10. https://www.instructables.com/Esp8266-Firebase-Connection/
11. https://circuitdigest.com/microcontroller-projects/arduino-based-accident-alert-system-using-gps-gsm-accelerometer
12. https://lastminuteengineers.com/neo6m-gps-arduino-tutorial/
13. https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet
14. https://console.firebase.google.com/u/0/project/accident-detection-syste-a1ad8/database/accident-detection-syste-a1ad8-default-rtdb/data