ISSN No:-2456-2165

Salvator: An Automatic Accident Detection and Rescue System

Arathi A [1] , Chithra Gopi [1] , Sandra Chandran [1] , Teena George [2] , Sreeresmi T S [2]
[1]UG Student, [2]Assistant Professor,
Department of Computer Science and Engineering,
Adi Shankara Institute of Engineering and Technology,
Kalady, Ernakulam, Pin:683574

Abstract*:* - With the enhanced technology and increased population, the transportation system is becoming the most vital element of human civilization. In this hectic world, we frequently get to notice many accidents around us. The World Health Organization (WHO) articulates that in each year approximately 1.35 million people die and more people endures non-fatal injuries as a consequence of road crashes. There exist numerous reasons for increase in the death rate due to vehicle collisions. One of them is inadequate post-crash care. In many situations the family members or emergency services are not informed on time. This results in delayed emergency service response, which can lead to an individual's death or cause severe injury, often resulting in long-term disabilities. So it's an emergency to develop a method to provide adequate services for the victims. The purpose of this project is to design a system that helps to reach unnoticed accident cases by detecting the occurrence of an accident. This will help them provide immediate medical care.

Keywords-:- *Post-Crash Care, Delayed Emergency Services, Unnoticed Accident Cases.*

I. INTRODUCTION

Humans and their rush for meeting daily desires surge day by day. In such developed cities people need vehicles to run for their needs leading to busy traffic. This increase in traffic can ultimately accelerate the number of accidents. Although in many of the cases there may not be immediate death, a few minutes delay in providing medical assistance leading to loss of life are even more dreadful. These can be due to some situations of accidents being unseen consciously or unknowingly even if there might be people around ready to help.

In order to reduce the death rate due to unnoticed accident situations and lack of emergency responses, this paper proposes a prime solution. The accident detection can be more accurate when the vehicle's parameters are taken for its prediction. Once detection is completed, the next step should be to bring the accident location into notice of an available ambulance driver.

Our system introduces an automatic accident detection and rapid alert system using raspberry pi installed in the vehicle and an android interface respectively. Live tracking by GPS module helps in updating the location of victim to server. The server's request/response service helps the ambulance drivers to receive the request of a victim with no human intervention and best efficiency. Along with this the same information is passed to the nearest police station. Through SMS the hospital details are reported to the emergency contacts of the victim.

The rest of the paper is structured in the following manner. Section II discusses the Literature Review, Section III discusses Components and Specification, Section IV discusses Implementation. Finally, Future Scope, Conclusion and References were discussed.

II. LITERATURE REVIEW

A considerable amount of research is already done in the field of accident detection and rescue systems. Much of the researchers carried out their studies about accident detection systems based on data obtained from various detectors such as induction loops, infrared detectors, cameras etc. However, the performance of these detection and prediction systems are greatly restricted by the number of monitoring sensors, algorithms, weather, traffic flow etc. This paper aims to create a system which can detect accidents and provide rescue measures in time with sufficient preparation at the correct place.

MQTT based vehicle accident detection and alert system [1] is an IoT based platform having a combination of low-power sensors and a cost-effective microcontroller for creating an application that is fully automatic. HDy Copilot, is an android application for accident detection. It is integrated with multimodal alert dissemination, both via eCall and IEEE 802.11p [2]. GPS and Map Matching Based Vehicle Accident Detection System is to detect an accident from the map matched position of a vehicle by utilizing the GPS speed data and map matching algorithm [3]. Intelligent Accident Detection and Alert System by Nicky Kattukkaran [4] detects the accident, confirms the seriousness of the case and then alerts the nearest medical assist center to provide emergency medical aid to the accident victim.

III. COMPONENTS AND SPECIFICATION

A. Hardware

- *Raspberry pi 3 model b+* - 1.4ghz 64-bit quad-core processor, dual-band wireless lan, Bluetooth 4.2/ble, faster ethernet, and power-over-ethernet support.

- *Thonny IDE* - It is a free IDE which is meant especially for python beginners. It contains a built-in debugger, that detects the errors within and also performs step-by-step evaluation of expression and visualization of call stack.

- *Neo-6m GPS module* - High-performance GPS module with a ceramic patch antenna, an on-board chip and a backup battery which can be conveniently integrated with a wide range of microcontrollers. GPS module helps in live location tracking.

- *Push Button* - It is a normal button, in which the click is considered as the occurrence of an accident. Whenever the push button gets activated, a high value is sent to the raspberry pi.

B. Software

- *Xampp* - Xampp is an application which enables a web server and a database server instantly without any additional configuration. web server and a database server Platform which consists of apache-http server, MariaDB/MySQL- database and interpreters for php and Perl languages.

- *Android studio* - Android studio is an official IDE for Google's android os for android app development Application side of the project is developed through this IDE. Android studio's apply changes feature helps to push code and resource changes to running apps without restarting the app.

- *Dream weaver* -It is a platform which is used to edit html, php, JavaScript, CSS and related files and helps in uploading these files into a web server. It is mainly used for creating html websites and also as a WYSIWYG web development tool.

- *Android Google Map API* - It gives access to google maps servers, map display, response to map gestures and much more. API can make calls to add markers, polygons, and covers to a basic map, and to change the view of a user about a specific map area. These objects can provide extra information for map locations, and also allows the user for interacting with the map.

- *Google Play services* - Google Play services are an API package used to update Google apps and apps from Google Play. This component provides functionalities like authentication to Google services, synchronized contacts, access to all user privacy settings and with higher quality and lower-powered location-based services. Google Play services also enhance app experience.

- *Retrofit* - Retrofit is a helper library for Android and Java by Square for creating http requests and for processing http response. It makes it quite easy to retrieve and upload structured data through a REST based web service.

- *SMS gateway API* - SMS API is a software interface which enables code to send messages. SMS APIs are used to allow web applications to easily send and receive text messages through a SMS Gateway for standard web frameworks.

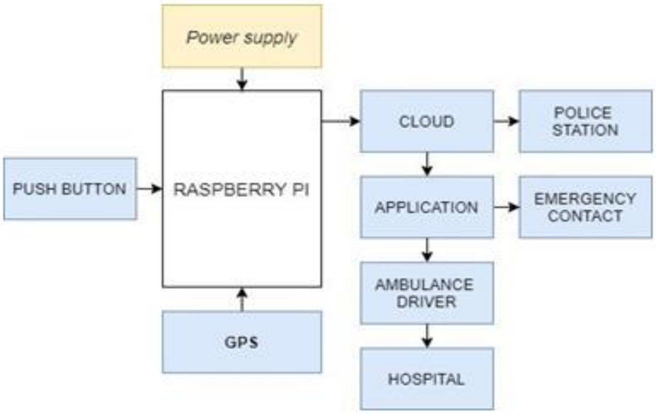

Fig 1:- Block Diagram

The above figure shows two phases of the system, the accident detection phase including the push button, raspberry pi and GPS and secondly the notification phase including the cloud and the application interface which are described in the following section.

IV. IMPLEMENTATION

A ACCIDENT DETECTION PHASE

The proposed project focuses on the reduction of time taken to detect an accident occurrence and reporting this to immediate medical care, thereby improving the survival rate. This automatic detection is achieved by integrating Global Positioning System, push button and cloud storage to the Raspberry Pi 3 Model B+ micro controller.

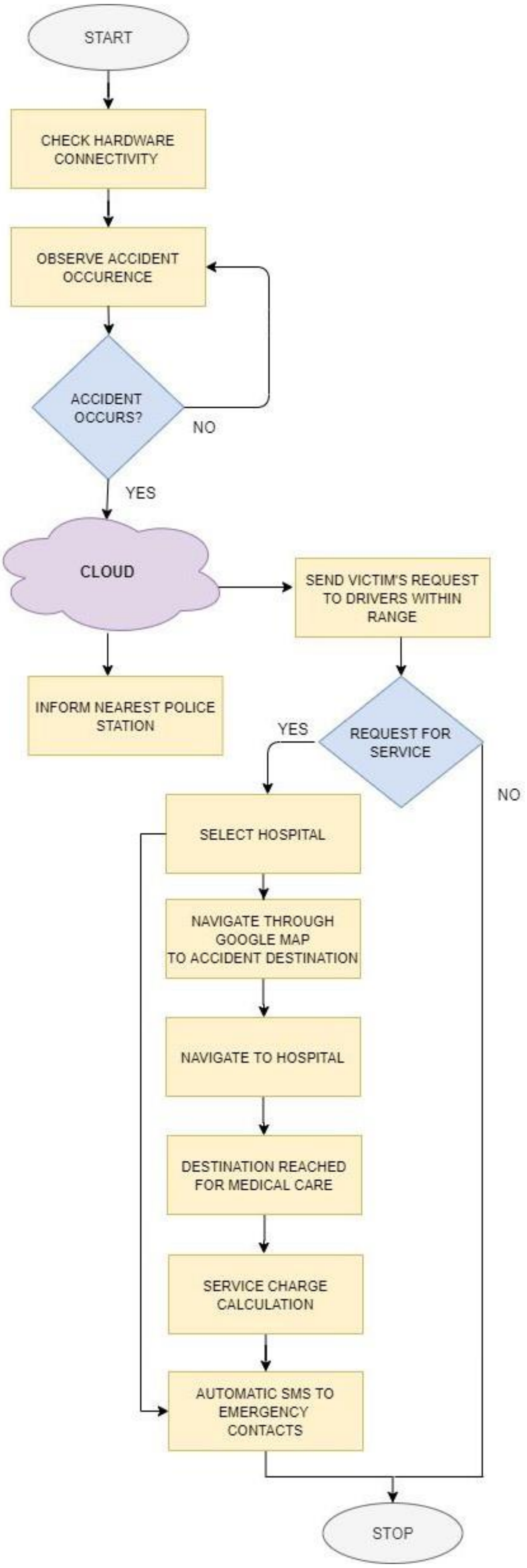
START
CHECK HARDWARE CONNECTIVITY
OBSERVE ACCIDENT OCCURENCE
ACCIDENT OCCURS?
NO
YES
CLOUD
SEND VICTIM'S REQUEST TO DRIVERS WITHIN RANGE
INFORM NEAREST POLICE STATION
YES
REQUEST FOR SERVICE
NO
SELECT HOSPITAL
NAVIGATE THROUGH GOOGLE MAP TO ACCIDENT DESTINATION
NAVIGATE TO HOSPITAL
DESTINATION REACHED FOR MEDICAL CARE
SERVICE CHARGE CALCULATION
AUTOMATIC SMS TO EMERGENCY CONTACTS
STOP

Fig 2:- Flowchart of the System

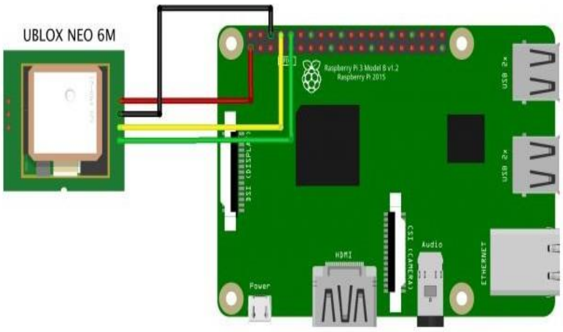

Fig 3:- Connection Diagram of Pi and GPS

In the proposed system, initially the Pi is installed with the Raspbian Operating System loaded in the micro SD card. This is then connected with the GPS module (Neo 6m gps) as in Fig.3. Unlike other GPS modules, it can capture up to 5 location updates per second with 2.5m Horizontal position accuracy. This keeps accurate live tracking of the user in the vehicle installed with the Pi. Python is the programming language used in Pi and is done on Thonny IDE. A push button is used to provide a high value into the Pi in case of an accident occurrence or a low value in other cases. So, the push button acts as an accident/crash detector [5]. The algorithm of accident detection is described below [6].

➢ *Algorithm 1: Algorithms for accident detection.*

```
Data: P, Value from push button
Result: Status of accident detection

if P = 1 then
        status = accident_occur;
        Get location from GPS;
        SET location (lat, lang) = current location (lat, lang);
        GET serial_no of the device;
        GET User_info from database using serial_no;
        MESSAGE = (location, serial_no);

        Send MESSAGE to server;
else
        status = no_accident;
End
```

In case of a high value, the GPS coordinates at that point and serial_no that uniquely identifies each user is sent to the cloud where the user details are fetched from the database using the unique serial_no [7].

Once the fetching of details from the database is completed, the location of the accident along with details are forwarded to all the available ambulance drivers within the range.

- *Code for retrieving latitude and longitude from a GPS module*

```
import serial
import pynmea2
def parseGPS(str):

    if str.find('GGA') > 0:
           msg = pynmea2.parse(str)
           print "Lat: %s -- Lon: %s" % (msg.lat, msg.lon)

serialPort = serial.Serial("/dev/ttyAMA0", 9600, timeout = 0.5)
while True:
    str = serialPort.readline()
    parseGPS(str)
```

B. *NOTIFICATION PHASE*

An android application for the ambulance driver is the second phase of accident detection and alert system. The purpose of this application is to alert the occurrence of accidents.

The proposed system uses cloud for its storage features. The cloud storage stores all the information about nearby hospitals, police stations and victims, including emergency contacts of corresponding persons. Therefore, this storage plays an indispensable role in the emergency response to correlate the data collected from the GPS to reach the destination in a short time period. Each user is provided with a unique serial number tag. The victim’s personal data uploaded in cloud storage can be accessed using this unique serial number [7] of the victim that had been received from the pi.

The accident is informed to the nearby police station through an SMS, along with which the registered ambulance drivers who are within the fixed range of the accident location receive an alert message notification in the respective application of their Smartphone. The accident occurrence is intimated to the ambulance drivers from the cloud, where the accident location and the vehicle number of the victim are attached. This is found using the algorithm 2 described later [6].

The Haversine Formula is used for finding nearest police stations and ambulance drivers. This formula is essential for finding straight line distance between two coordinates on the earth using longitude and latitude parameters. The input of this method is the latitude and longitude which are the earth’s coordinates. The output of this formula is the distance between the two locations.

The Haversine formula works with a round object by giving three locations in determining its distance. This formula is closely related to the shape of a triangle drawn on the surface of the sphere.

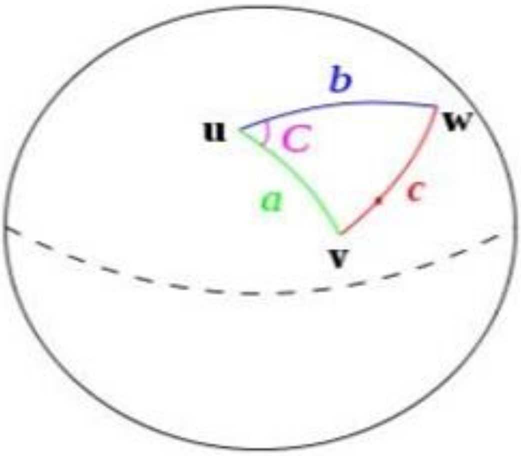

Fig 4: - Haversine Triangle

Fig.4 [8] is an explanation of the haversine method applied to the earth. There are three coordinates, "u", "v", and "w" and these three coordinates form the distance builder in the Haversine calculation. Haversine calculations can be formulated into the following equation:

$haversine(c) = haversine(a - b) + sin(a)sin(b)haversine(c)$
$a = sin^2(\Delta\phi/2)cos\phi1.cos\phi2$
$c = 2.atan2.(\sqrt{a}.\sqrt{(1 - a)})$

Where

ϕ = latitude
λ =longitude
R = earth radius (6371)

- *Algorithm 2: Algorithmfor finding ambulances in a range*

```
Data: MESSAGE = (location, Vehicle_no)
Result: Ambulance Dispatched;

Server decode the message;
R = radius of earth (mean radius = 6,371km)
lat1 = start.lat; //latitude of accident location
lon1 = start.lng; // longitude of accident location
lat2 = end.lat;
lon2 = end.lng;
lat_distance = toRad(lat2-lat1);
lon_distance = toRad(lon2-lon1);
a = Math.sin(lat_distance/2) * Math.sin(lat_distance/2) + Math.cos(toRad(lat1)) * Math.cos(toRad(lat2)) * Math.sin(l0on_distance/2) * Math.sin(lon_distance/ 2);
c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
distance = R * c;

Cloud finds the ambulances within a range using the HAVERSINE;
Server sends request to the application;
Ambulance dispatches;
```

Once the message is received, the ambulance driver selects the request. When requests are accepted by the responders, they will be able track the location of victims in real time on a Google map with route to location of emergency victims using Google Directions API. This ensures that drivers can continuously receive information about an accident. Once the driver reaches the location of the crash, the victim is taken to the nearest hospital as shown on the application. The driver needs to select the particular hospital from the listed nearby hospitals. The cloud has a database of hospitals and determines the nearest hospital using a mapping service (Google Maps API).

When the selection is performed, it automatically sends an SMS about the accident occurrence to the victim's emergency contacts taken from the database using SMS gateway API. The SMS contains information about the location of the hospital.

In the payment side, when destined at the hospital, service charges would be calculated. This application provides an option to append the distance travelled in kilometers. The amount per kilometer is already fixed. When the distance is entered by the driver, the amount to be paid is autogenerated. This total amount will also be sent to the emergency contacts of the victim as SMS.

Thus, the system guides the ambulance service in reaching the accident site and shares data between the cloud and emergency service providers in real time. This data is shared in order to emphasize the rescue aspects of the system.

V. FUTURE SCOPE

In future, more research is needed to make the accident detection part more reliable and accurate. By adding components like gyroscope, camera (to automatically take pictures of the accident at that time), a voice recognition module (to detect noises during a vehicle crash like noise when airbags are deployed along with additional sensors) etc. can improve the accuracy of an accident detection. Reduction in rescue time can be obtained by controlling traffic flow in respective areas on ambulance dispatch, thus the driver can select the simplest path with less traffic very easily. Also automatic payment options can be added in future by integrating an additional payment gateway to the system. The system can be expanded to have a high population density warning system for areas such as schools, colleges etc. The system can be modified as a Stolen Vehicle tracking system by expanding its software capabilities, without needing any new hardware.

VI. CONCLUSION

The system for automatic accident detection is designed in order to reduce the time lag in taking an accident victim to the hospital. Thus helps the community reduce the death rates due to vehicle accidents. So this prototype accelerates the action taken by the responsible team in reaching the accident spot. A main advantage of the prototype is that it handles the accident detection and location tracking without human intervention. Even if the device gets destroyed during an accident, the continuous updating of location to cloud helps in getting the latest updated location. Also until a request gets accepted by one of the drivers, the request remains visible to all nearby drivers. Thus, the system clearly helps an accident victim at the earliest even if they are unconscious.

REFERENCES

[1]. Ravi K Kodali and Shubhi Sahu (2017, December). MQTT based vehicle accident detection and alert system. In *2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)* (pp. 186-189). IEEE.

[2]. Fernandes, B., Gomes, V., Ferreira, J., & Oliveira, A. (2015, May). Mobile application for automatic accident detection and multimodal alert. In *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)* (pp. 1-5). IEEE.

[3]. Amin, M. S., Bhuiyan, M. A. S., Reaz, M. B. I., & Nasir, S. S. (2013,December). GPS and Map matching based vehicle accident detection system. In *2013 IEEE Student Conference on Research and Development* (pp. 520-523). IEEE.

[4]. Kattukaran, N., George, A., & Haridas, T. M. (2017, January). Intelligent accident detection and alert system for emergency medical assistance. In *2017 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-6). IEEE.

[5]. Desima, M. A., Ramli, P., Ramdani, D. F., & Rahman, S. (2017, November). Alarm system to detect the location of IOT-based public vehicle accidents. In *2017 International Conference on Computing, Engineering, and Design (ICCED)* (pp. 1-5). IEEE.

[6]. Bhatti, F., Shah, M. A., Maple, C., & Islam, S. U. (2019). A novel internet of things-enabled accident detection and reporting system for smart city environments. *Sensors*, *19*(9), 2071.

[7]. Badi Alekhya, S.Aarthi, V.Glory(2017,March). Accident Detection with Location and Victim Information Transmission Using Raspberry Pi.

[8]. Prasetya, D. A., Nguyen, P. T., Faizullin, R., Iswanto, I., & Armay, E. F. (2020). Resolving the Shortest Path Problem using the Haversine Algorithm. *Journal of Critical Reviews*, *7*(1), 62-64.