A Project Report on

IOT based Accident Alert System

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

in

Information Technology

by

Kunal Kale(16104027)
Viranchee Patil(16104044)
Chinmay Kubal(16104023)

Under the Guidance of

Prof. Kaushiki Upadhyaya
Prof. Geetanjali Kalme

Department of Information Technology
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615
UNIVERSITY OF MUMBAI

Academic Year 2019-2020

Approval Sheet

This Project Report entitled ***"IOT based Accident Alert System"*** Submitted by ***"Kunal Kale"(16104027),"Viranchee Patil"(16104044),"Chinmay Kubal"(16104023))***is approved for the partial fulfillment of the requirenment for the award of the degree of ***Bachelor of Engineering*** in ***Information Technology*** from ***University of Mumbai***.

(Prof. Geetanjali Kalme)
Co-Guide

(Prof. Kaushiki Upadhyaya)
Guide

Prof. Kiran Deshpande
Head Department of Information Technology

Place:A.P.Shah Institute of Technology, Thane
Date:

CERTIFICATE

This is to certify that the project entitled ***"IOT based Accident Alert System"*** submitted by ***"Kunal Kale" (16104044),"Viranchee Patil" (16104044),"Chinmay Kubal" (16104023)*** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Information Technology.***,to the University of Mumbai,is a bonafide work carried out during academic year 2019-2020.

(Prof. Geetanjali Kalme)
Co-Guide

(Prof. Kaushiki Upadhyaya)
Guide

Prof. Kiran Deshpande
Head Department of Information Technology

Dr. Uttam D.Kolekar
Principal

External Examiner(s)

1.

2.

Place:A.P.Shah Institute of Technology, Thane
Date:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

_____

(Kunal Kale 16104027)

_____

(Viranchee Patil 16104044)

_____

(Chinmay Kubal 16104023)

Date:

Abstract

Ever since India attained independence in 1947, the country has been getting diversified year after year. As a result, urbanisation has taken place in most part of the country. This has lead to an exponential increase in the number of vehicles on road as mapped against the limited number of roads present. As a matter of fact the number of accidents has also increased exponentially. The Indian Ministry of Road Transport and Highways reports that around 4,06,730 accidents take place each year and close to 86,000 are killed! In frequently crowded areas like cities or highways, more than 65 percent lives are lost due to delay in the arrival of emergency services. If we consider accidents taking place in secluded areas, then the percentage of lives lost is even more. Sadly, with the ever increasing population and lack of pothole – free roads these numbers will continue to increase even more. With an aim to reduce the number of car crash fatalities, we have implemented the concept of IoT, so that we can leverage the nearby sensors to help the vehicles which have collided or are in need of any help. In case of accident, there will be some collision in the vehicle which will be sensed by the sensors. The crash sensors will measure and report the intensity of collision based on certain parameters and operations related to the automotive design of the vehicle. This strength of collision mapped on a scale will then inform the respective nearby sensors in IoT which can come out to help the victims. In case of lighter collision, only the local car repairs and other nearby sensors forming an IoT network will be informed. In case of high intensity collision, the data relating to location of car and other things will be sent to cloud which will then inform the nearby car repairing showroom, hospital services and the repairing services.

Contents

List of Figures

Chapter 1

Introduction

The advent of technologies like Cloud computing, Internet of Things, Ubiquitous computing, Autonomous computing etc. have turned a number of unimaginable concepts and theories into reality. Earlier a major setback in the realization of any concept or a theory was the lack of availability of computing resources but now, with the introduction of these new technologies we are able to introduce our concepts to the future world in no time. One such example is the Automatic accident alert system. The concept of automatic accident detection is not relatively new, in fact, a lot of high end cars are already equipped with such systems. For example: OnStar from General Motors and ACN (Automatic Crash Notification System) by BMW. The way these systems work is via Onboard diagnostic system and a GPS sensor that is responsible for the collection of crash and location information respectively. They also use CDMA cellular technology for voice communications and data transmissions. The idea of having such a system installed in a car is not completely bulletproof, there are certain disadvantages as well, such as high costs, non-portability, etc. In order to tackle these drawbacks, we have designed a new system that works similar to these system but is more efficient and feasible. Our system not only detects an accident but also performs some follow up steps that may help the car crash victim drastically. Usually when an accident occurs, a lot of valuable time is wasted due to panic, followed up by a call to the emergency authorities and then applying first aid. Withe the help of our system, the victim can skip the part of calling/notifying emergency authorities as it is performed automatically by the accident alert system. The sytem gathers data from various sensors and checks whether the values cross a certain threshold. The data is then uploaded to the cloud and depending upon the intensity of the crash, nearby emergency services such as local clinics, hospitals, fire stations and police stations are alerted.

Chapter 2

Literature Review

[1] "Intelligent Accident Management system using IoT and Cloud Computing"
Authors: Akriti Singhal , Sarishma , Ravi Tomar

The model presented here involves a collective integration of different types of sensors as well as a microcontroller unit which acknowledge emergency calling system. This technology includes the benefits of GSM modem used as automatic emergency calling system and GPS sensor for location calling. GSM modem requires a SIM card and works with a GSM wireless network. Accelerometer sensors and vibration sensors are collaborated. The accelerometer is used to measure the acceleration and the vibration sensor is used to measure various physical parameters like change in acceleration, temperature, etc. During an accident abnormal changes in the physical parameters mentioned above are detected and thus, the accident is confirmed. The severity of the accident is mapped on an intensity scale. Further, the display is provided in LCD interfaced with the microcontroller and LED indicates the occurrence of an accident. The GSM module is activated at once by the microcontroller, which has a saved number of a nearby emergency calling location. This emergency location is decided depending upon the impact of collision. The main advantage of this system is that it is cheap and uploads data onto the cloud. The disadvantage is that the processing speed and internet speed is compromised due to microcontroller and GSM module respectively.

[2] "S-Car Crash: Real-time crash detection analysis and emergency alert using smartphone"
Authors: Harit Sharma , Ravi Kanth Reddy , Archana Karthik

In this paper, they developed a system which integrates the functonalities of sensors present inside a mobile phone such as GPS receiver, accelerometer, gyroscope, etc. In this model, using smartphone sensors as the source for input stream, makes it a cheaper alternative to expensive in-vehicle accident detection systems. Using smartphone for developing such solution gives an easy access to the already established wireless network infrastructure, which can be used to notify emergency responders in case an accident is detected. Since the primary source of accident detection in this system is the actual smartphone itself, many false positives in the acceleration can be detected which may trigger the emergency mechanism. The proposed model in this paper eliminates cases of false positives by calculating the collision index and then depending upon it's value, emergency authorities are informed. The main advantage of this system is that all the sensors required are present inside the mobile

itself that's why no additional components/modules are required. The biggest disadvantage of this system is that the mobile device that will detect the car crash can itself get destroyed at the time of an accident. In such a scenario, the system fails immediately.

[3] "A Smart Accident Detection and Control System in Vehicular Network"
Authors: Md. Saeef Abdul Hadi, Abhijit Saha, Faysal Ahmad, Mohammad Shahriyar Hasan, Mehebub Hasan Milon

In this paper, they developed a smart accident detection system which is used to detect accidents that happen at a junction. The smart accident detection system integrates the working of an Access Point infrastructure , a Control Box infrastructure and a Control Room. In case of an accident, a message regarding the accident is broadcasted to all the vehicles that are present in the same lane wherein the accdient has occured. For this the API uses a microcontroller which uses 'push method' for broadcasting messages and the receiver present inside vehicles uses 'pull method' to receive the incoming message. On receiving the message, drivers are alerted about the accident's lane number. Due to this they can divert their path and avoid further damages. The system is also capable of alerting emergency authorities in case of an accident. A major disadvantage of this system is that in case of heavy taffic routes, a slight delay in broadcasting the message reagrding the accident can lead to more vehicles getting involved in accidents as well.

[4] "A Comprehensive Solution to Road Traffic Detection and Ambulance Management"
Authors: Hari Sankar S, Jayadev K, Suraj B, Aparna P

In this paper, they developed a system to address the problem of delay minimization right from when an accident occurs till the time when the patient is handed to the casualty. A dedicated in-vehicle accident module which consists of a microcontroller, GPS receiver, accelerometer and Raspberry Pi B+ is used. A model view controller based Django framework is used to collect data about the accident and nearby ambulances using POST method. When an accident occurs, the server utilizes the Distance matrix service from Google Maps API web services to find the real map distances and transit times of nearby ambulances to the accident location. Only ambulances belonging to the same jurisdiction as that of the accident are considered for sorting. The "nearest" ambulance is found by using a simple insertion sort on the transit times. The term "nearest" is defined in terms of time required to reach the accident spot. It is to be noted that the transit time for each ambulance suggested by Google Maps API web servicesdependson theprevalent trafficconditions. Accident coordinates are then relayed to this "nearest" ambulance. Utilizing other functionalities of Google Maps API web services like transit time, it is possible to track and keep a check on the performance of the ambulance driver from a single terminal preferably at a control room.

graphicx

Chapter 3

Proposed System

Given below is the flow of modules in our project:

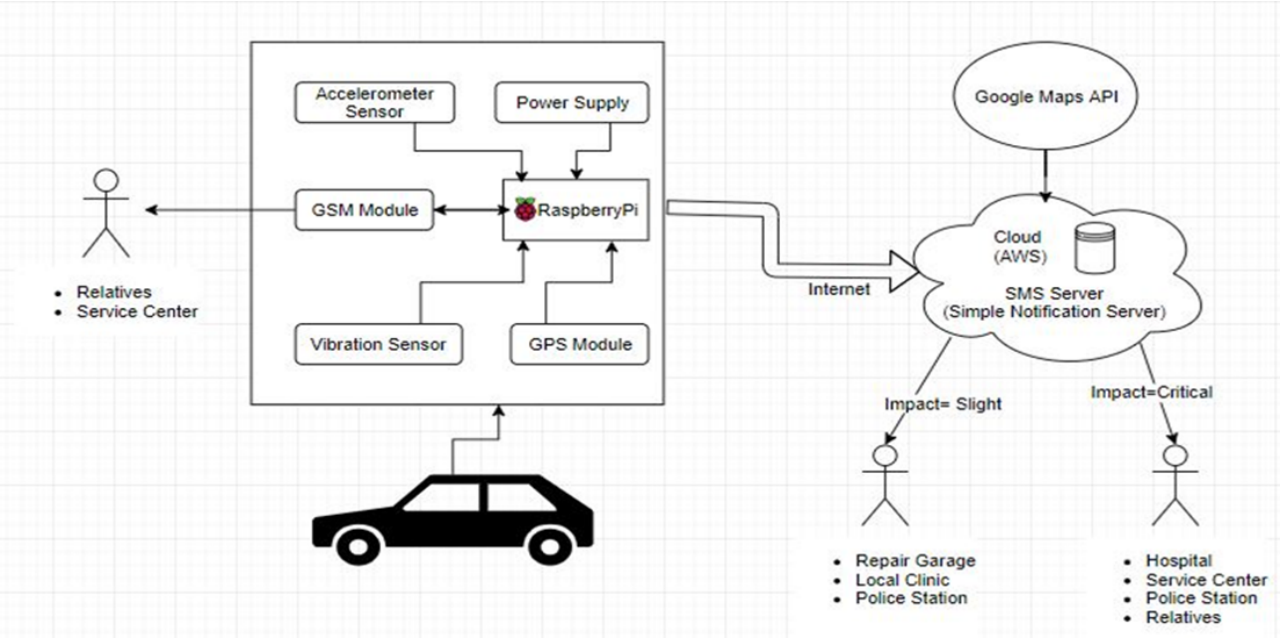


Figure 3.1: Proposed System

In our system,a Raspberry Pi acts as the main microcomputer, to which various sensors are connected. Out of these sensors the accelerometer and the vibration sensor play the most important role in detecting the accident. As we know, the value of gravitational constant g= 9.8 on Earth's surface. A sudden abnormal change in its value may indicate a crash or a false alarm. This is confirmed by the vibration sensor which constantly monitors the changes in temperature,pressure,etc. If at all a crash takes place,the location of the crash is detected via the GPS module and sent to the Cloud with the help of the wifi connectivity offered by the Raspberry Pi. From the cloud,a notification is sent to the emergency authorities depending upon the impact of the collision via AWS Simple Notification Service (SNS). In case the crash takes place in a secluded area where internet connectivity is poor,the system will make use of a GSM module which will send SMS text messages to all the numbers stored in it regarding the accident.

For determining the intensity of the car crash,we made use of the following table:

Accident severity	Value of z	Approximate value of G range
Safe level	1	0-4 g
Slight level	2	4-20 g
Moderate level	3	20-40 g
Critical level	4	40+ g

Figure 3.2: Intensity corresponding to G-forces

All of the above mentioned values are according to a survey conducted by National Highway Traffic Safety Administration (NHTSA),India in the year 2017.

Working of the project at the backend (Cloud part):
We will be using Amazon Web Services for our backend.
The following services will be used:

1. S3 Bucket: Will be used as the main storage space for storing the processed as well the generated data.

2. EC2 Instance: Will be used as the main server for running various AWS services.

3. Simple Notification Service: Will be used for sending notification alerts to the emergency services and/or the relatives of the car crash victim. Subscribing to this service is mandatory in order to send or recieve notification alerts.

4. Reverse Geocoding API: To convert the data generated by the GPS module,which is in NMEA format,into human readable format.

Chapter 4

Project Design

4.1 Basic Diagrams:

Any real-world system is used by different users.The users can be developers, testers, business people, analysts, and many more.Hence, before designing a system,the architecture is made with different perspectives in mind.The most important part is to visualize the system from the perspective of different viewers. The better we understand the better we can build the system. Hence UML diagrams are used to represent the idea of users with the help of various diagrams. UML diagrams is a standard language for specifying, visualizing,constructing, and documenting the artifacts of software systems.

4.1.1 Usecase Diagram:

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 system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

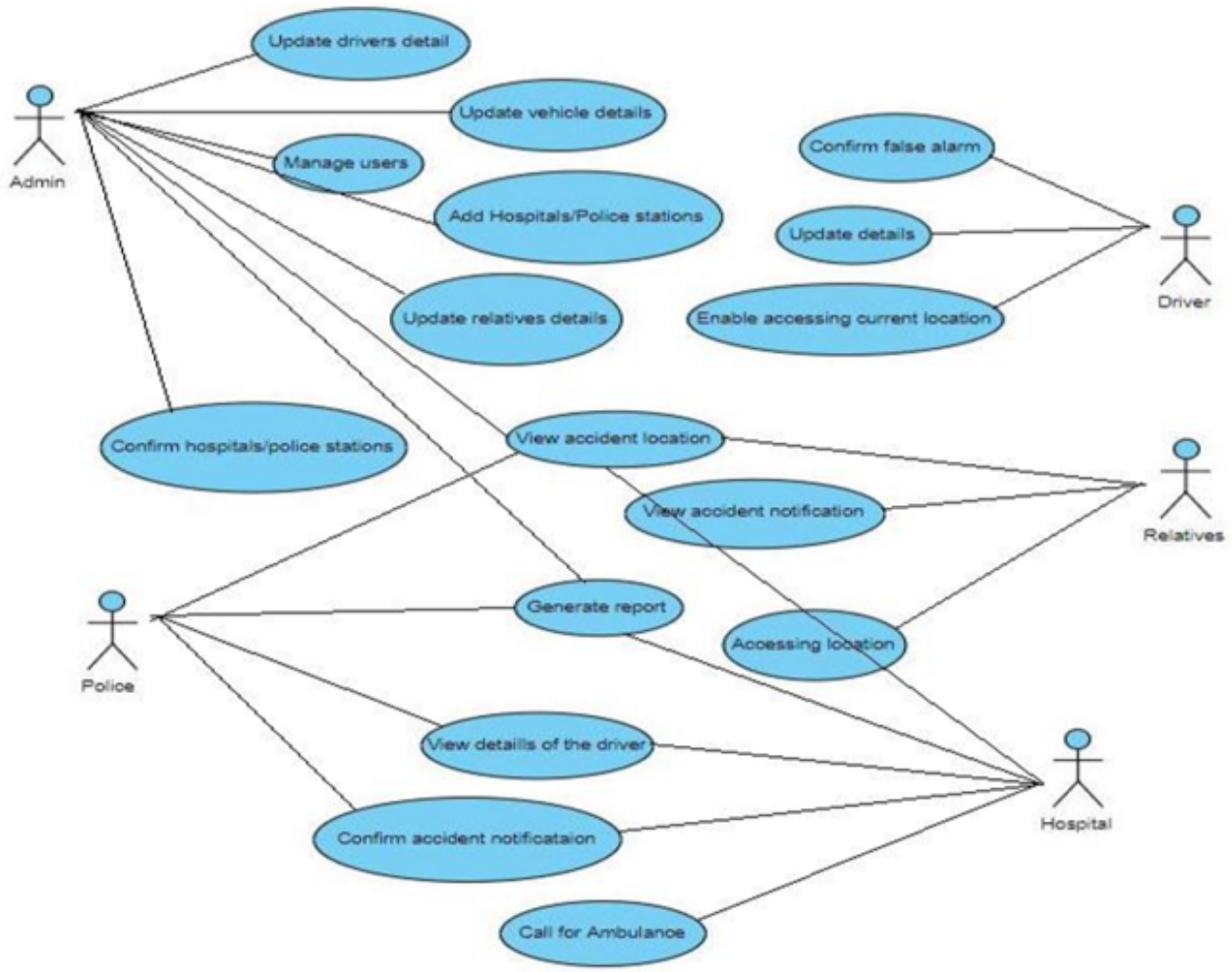


Figure 4.1: Usecase Diagram

The above diagram depicts the usecase diagram for our project.Since the car crash victim uses our system and the emergency authorities and relatives of the victim are notified by the accident alert system, they all are the various actors that are associated with the system. Some of the actions that these actors can perform are view details of the driver, view location, generate report,manage users,etc.

4.1.2 Class Diagram:

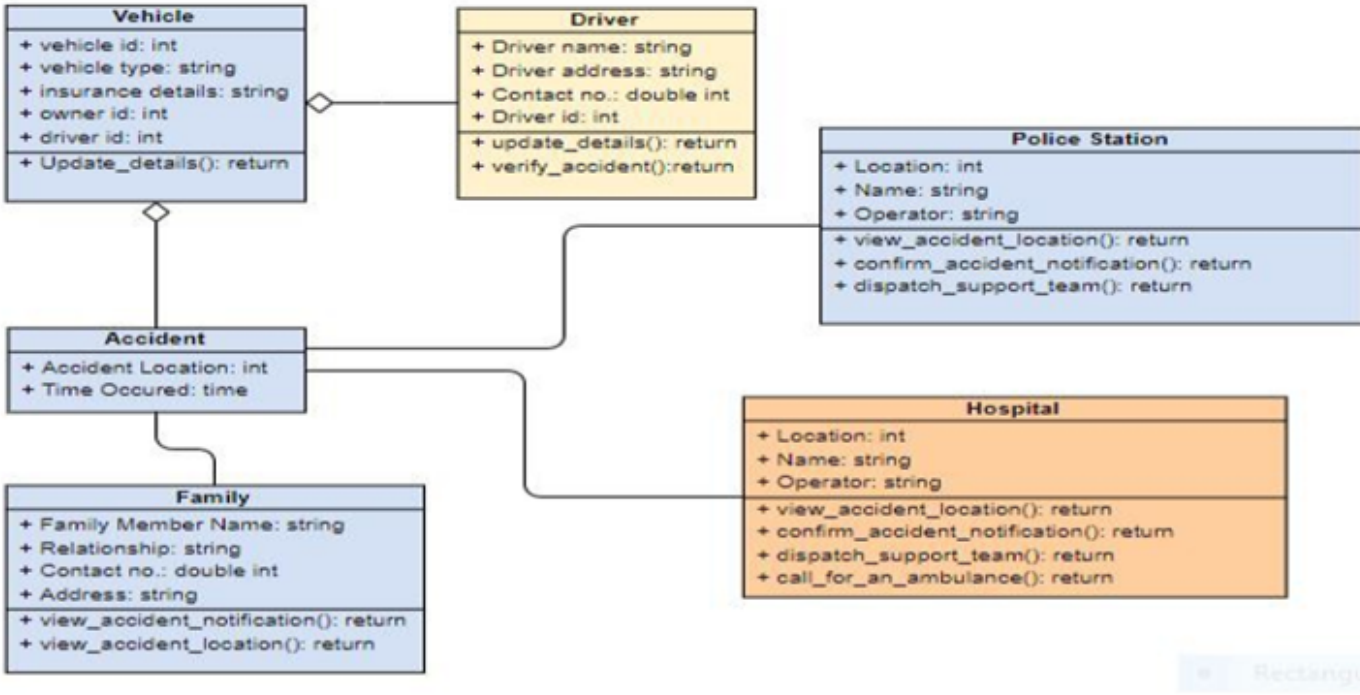


Figure 4.2: Class Diagram

The various entities in this environment are:

1. **Driver:** The registered user that makes use of the Accident alert system.

2. **Vehicle:** The vehicle that is driven by the registered user.

3. **Accident:** The time and location of the car crash.

4. **Family:** Relatives/friends of the registered user.They will be notified at the time of crash.

5. **Police Station:** An emergency service authority.

6. **Hospital:** An emergency service authority.

4.1.3 Activity Diagram:

Activity diagram is another important diagram in UML to describe the dynamic aspectsof the system.Activity diagram is basically a flowchart to represent the flow from one ac-tivity to another activity.The control flow is drawn from one operation to another.Activitydiagrams deal with all type of flow control by using different elements such as fork, join, etc.Activity is a particular operation of the system.Activity diagrams are not only used for vi-sualizing the dynamic nature of a system, but they are also used to construct the executablesystem by using forward and reverse engineering techniques.

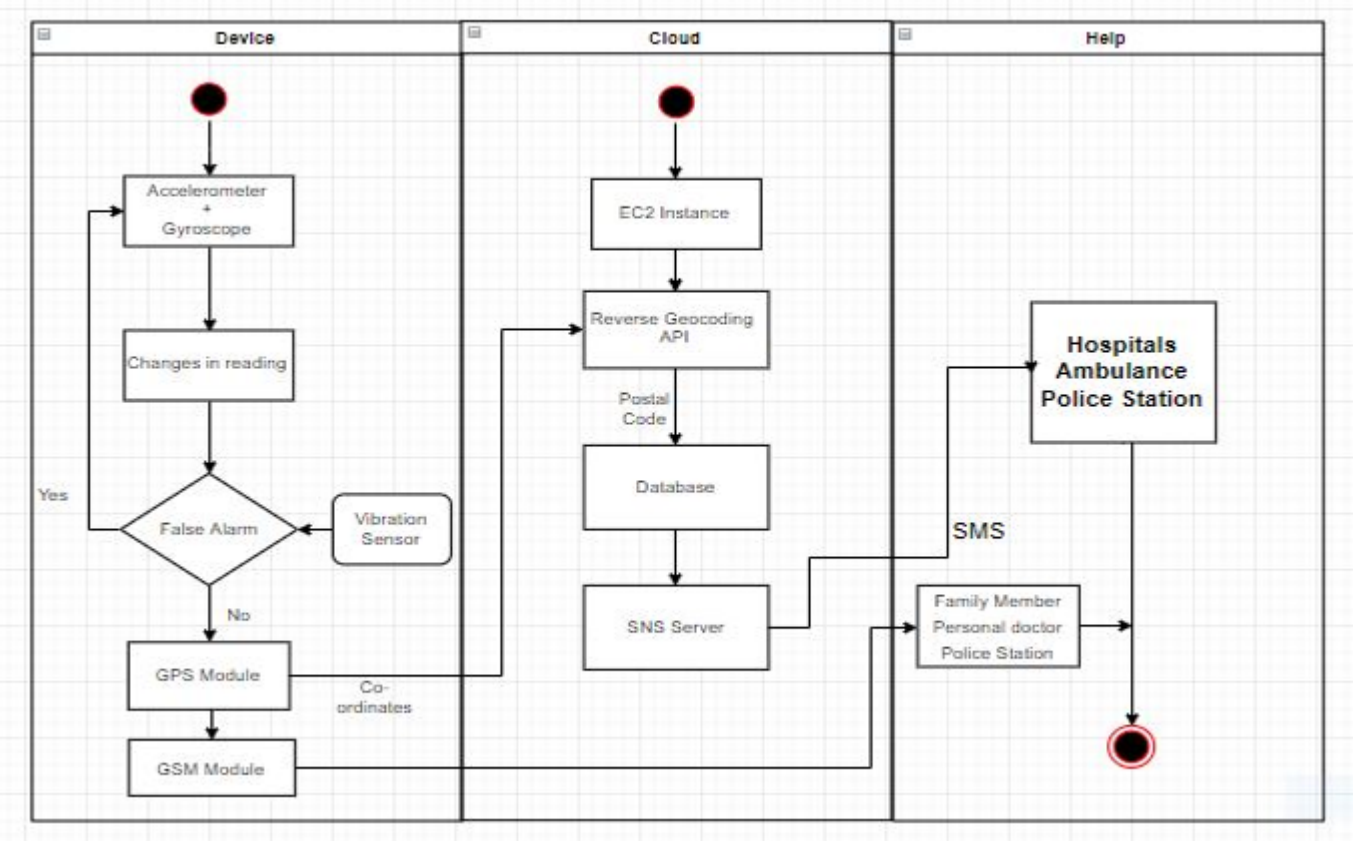


Figure 4.3: Activity Diagram

When a sudden change in the value of gravitational constant g is recorded,it is cross checked with the readings of the vibration sesnor to check for a false alarm.If it's a false alarm, the user can manually turn it off,however if a genuine crash is detected, the GPS module send-sthe location coordinates to the cloud. The received data is first stored in an S3 bucket and then the Reverse Geocoding API acts on it and converts into a human readable format.This data is then stored in a Database and notification is sent to the emergency services via the SNS service. If at all the internet connectivity of the area where the crash was recorded is poor, the GSM module is used to send normal SMS text messages to the contacts saved in the sim card.

Chapter 5

Project Implementation

5.1 Working and Code Snipettes

The working of IOT Based Accident Alert System begins with the accelerometer which is constantly monitoring the vehicle's acceleration with respect to the 3 axes. This data that is generated is constantly being uploaded to the cloud.

```
void setup() {
  Serial.begin(9600);
  Wire.begin(sda, scl);
  MPU6050_Init();
}

void loop() {
  double Ax, Ay, Az, T, Gx, Gy, Gz;

  Read_RawValue(MPU6050SlaveAddress, MPU6050_REGISTER_ACCEL_XOUT_H);

  //divide each with their sensitivity scale factor
  Ax = (double)AccelX/AccelScaleFactor;
  Ay = (double)AccelY/AccelScaleFactor;
  Az = (double)AccelZ/AccelScaleFactor;
  T = (double)Temperature/340+36.53; //temperature formula
  Gx = (double)GyroX/GyroScaleFactor;
  Gy = (double)GyroY/GyroScaleFactor;
  Gz = (double)GyroZ/GyroScaleFactor;

  Serial.print("Ax: "); Serial.print(Ax);
  Serial.print(" Ay: "); Serial.print(Ay);
  Serial.print(" Az: "); Serial.print(Az);
  Serial.print(" T: "); Serial.print(T);
  Serial.print(" Gx: "); Serial.print(Gx);
  Serial.print(" Gy: "); Serial.print(Gy);
  Serial.print(" Gz: "); Serial.println(Gz);

  delay(100);
}

void I2C_Write(uint8_t deviceAddress, uint8_t regAddress, uint8_t data){
  Wire.beginTransmission(deviceAddress);
  Wire.write(regAddress);
```

The vibration sensor is also monitoring data such as displacement, change in temperature,etc. This is because at the time of crash, the values of displacement and temperature also change abruptly along with the change in acceleration. So, to check for false alarms, the vibration sensor is essential.

```
#include <ESP8266Wifi.h>
#include <EEPROM.h>
int val;

#define USE_SERIAL Serial
#define Vib A0

void setup()
{
  Serial.begin(115200);
  delay(10);

}

void loop()
{
  val=analogRead(Vib);
  Serial.print(val);
  Serial.println(" ");

}
```

When a crash occurs, and it is confirmed by the accelerometer as well as the vibration sensor, the geographical coordinates of the location are mapped and uploaded to the cloud via the MQTT Cloud service provided by Amazon AWS.

The following figure shows the code for GPS sensor using which it fetches the coordinates and sends the to Cloud:

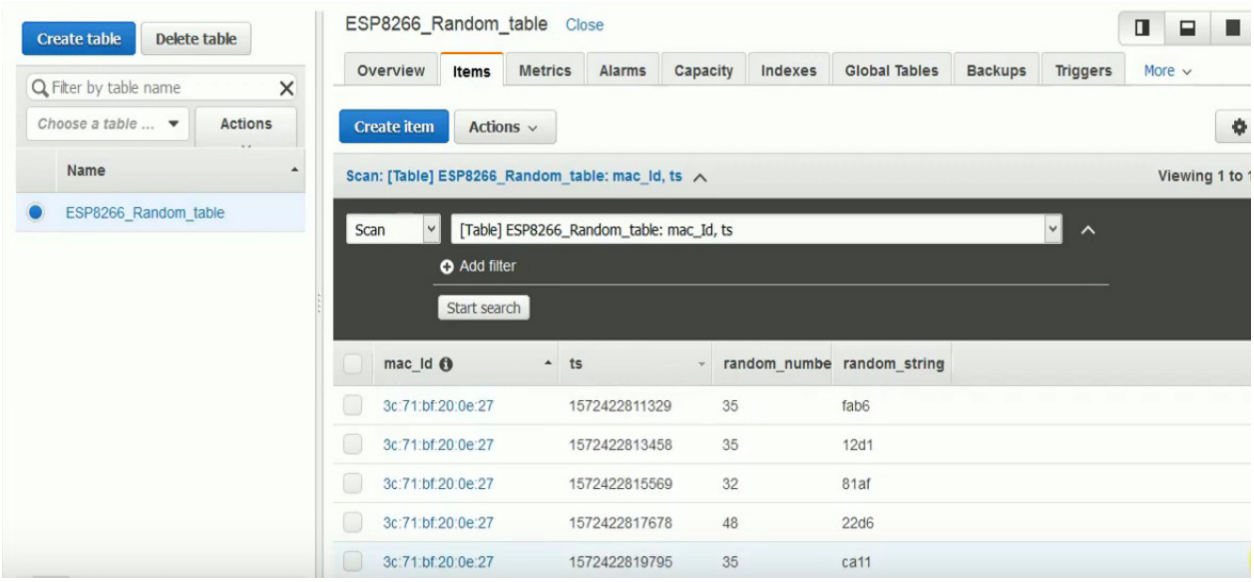
```
#ifdef ESP8266
extern "C" {
#include "user_interface.h"
}
#endif

#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>

ADC_MODE(ADC_VCC);

WiFiClientSecure client;

const char* host = "u3v20krp3l.execute-api.us-east-1.amazonaws.com";
String ssid = "Oneplus7 Pro";
String pass = "123456";

void setup() {
  Serial.begin(115200);
  wifiConnection();
}

void loop() {
    String url = "/dev/map";
    String dataLocation = "{ \"lat\": \"19.222" + String(rand() % 10) + "45\", \"lng\": \"72.960" + String(
    Serial.println("POST to https://" + String(host) + url);
    Serial.println("Data: " + String(dataLocation));
    Serial.print("Result(response): ");
    Serial.println(httpsPost(url, dataLocation));
    delay(10000); //delay 10s
}
```

These coordinates are then sent to the emergency services via an email notification using the AWS Simple Notification Service. For the emergency services to receive an alert regarding the crash, they should be subscribed to the SNS service.

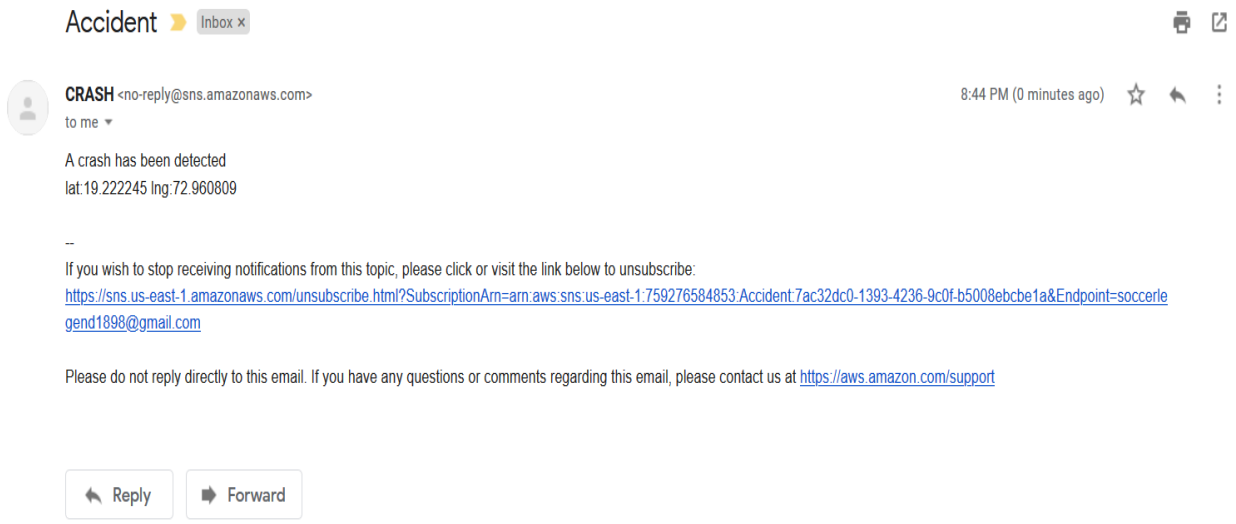
```java
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.regions.Region;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.sns.AmazonSNSClient;
import com.amazonaws.services.sns.model.*;
import org.jetbrains.annotations.NotNull;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import java.util.concurrent.TimeUnit;


public class Main {

    private static final Logger LOGGER = LoggerFactory.getLogger(Main.class);


    public static void main(@NotNull String[] args) throws InterruptedException {

        if(args.length == 0)
        {
            System.out.println("Enter a valid email address as argument");
            System.exit(-1);
        }

        final String email = args[0];

        AmazonSNSClient snsClient = new AmazonSNSClient(new DefaultAWSCredentialsProviderChain());
        snsClient.setRegion(Region.getRegion(Regions.us-east-1));


        CreateTopicRequest createTopicRequest = new CreateTopicRequest().withName("Accident");
        CreateTopicResult createTopicResult = snsClient.createTopic(createTopicRequest);
        final String topicArn = createTopicResult.getTopicArn();
        LOGGER.info("Topic ARN : "+topicArn);

        SubscribeRequest subscribeRequest =
                new SubscribeRequest()
                .withTopicArn(topicArn)
                .withProtocol("soccerlegend1898@gmail.com")
                .withEndpoint(email);

        SubscribeResult subscribeResult = snsClient.subscribe(subscribeRequest);
        String subscriptionArn = subscribeResult.getSubscriptionArn();
        LOGGER.info("Subscription ARN : {}, for Endpoint {}",subscriptionArn, email);
        if(subscriptionArn.equals("confirmation pending")) {
            LOGGER.info("Subscription confirmation pending. Waiting 60s for confirmation ...");
            Thread.sleep(TimeUnit.SECONDS.toMillis(60));
        }


        String msg = "A crash has been detected";
        PublishRequest publishRequest = new PublishRequest().withMessage(msg).withTopicArn(topicArn);
        PublishResult publishResult = snsClient.publish(publishRequest);
        LOGGER.info("Message send with id {}."+publishResult.getMessageId());
```

5.2 Output

1.Sensor data stored in DynamoDB on Cloud



2. Email notification received via SNS



Chapter 6

Testing

For the testing purpose we opted to go for the functional testing methods. Functional testing involves testing the application against the business requirements. It incorporates all test types designed to guarantee each part of a piece of software behaves as expected by using cases provided by the design team or business analyst. Function testing includes :
1. Unit Testing
2. Integration Testing
3. System Testing
4. Acceptance Testing

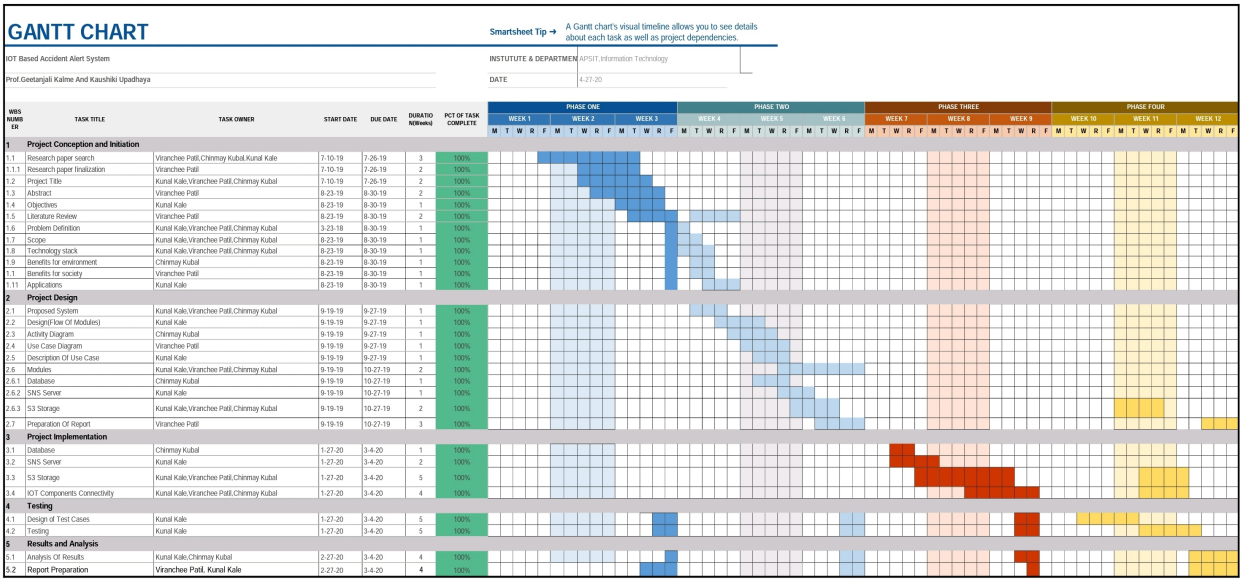
Unit Testing:
Unit testing is the first level of testing and is often performed by the developers themselves.It is the process of ensuring individual components of a piece of software at the code level are functional and work as they were designed to. Developers in a test-driven environment will typically write and run the tests prior to the software or feature being passed over to the test team. Unit testing also makes debugging easier because finding issues earlier means they take less time to fix than if they were discovered later in the testing process.Therefore, opting for the unit testing method in our project played a crucial role in assessing each module of the application separately. This testing method best suited our project as we had various modules at the start which were to be tested and verified. It made the testing process easier by helping us discover the minute errors in each module and therefore we could rectify them efficiently.

Integration Testing:
After each unit is thoroughly tested, it is integrated with other units to create modules or components that are designed to perform specific tasks or activities. These are then tested as group through integration testing to ensure whole segments of an application behave as expected (i.e, the interactions between units are seamless). Integrated tests can be conducted by either developers or independent testers and are usually comprised of a combination of automated functional and manual tests.Integration testing was a necessity to check whether each individual module/unit was working well in synchronisation with one another. There were multiple problems while integrating the various modules which were only discovered with the help of integration testing methodology.

System Testing:
System testing is a black box testing method used to evaluate the completed and integrated system, as a whole, to ensure it meets specified requirements. The functionality of the software is tested from end-to-end and is typically conducted by a separate testing team than the development team before the product is pushed into production. Finally, the entire system was tested as a whole using the System testing Methodology.
Here, the functional requirements of our applications that include coordinates collection,uploading data to cloud, sending notifications via cloud based services were checked altogether.

Acceptance Testing:
Acceptance testing is the last phase of functional testing and is used to assess whether or not the final piece of software is ready for delivery. It involves ensuring that the product is in compliance with all of the original business criteria and that it meets the end user's needs.This requires the product be tested both internally and externally, meaning you'll need to get it into the hands of your end users.Acceptance Testing method was adopted for our application to get a final review of our application by the actual end users. The application was given to fellow peers to try hands on and get their opinions and recommendations which were worked upon. Also, it was seen that all end user needs were met to their complete satisfaction after using the application.

Project Timeline



GANTT CHART

Smartsheet Tip → A Gantt chart's visual timeline allows you to see details about each task as well as project dependencies.

IOT Based Accident Alert System

INSTUTUTE & DEPARTMEN APSIT,Information Technology

Prof.Geetanjali Kalme And Kaushiki Upadhaya

DATE 4-27-20

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION(Weeks)	PCT OF TASK COMPLETE
1	**Project Conception and Initiation**					
1.1	Research paper search	Viranchee Patil,Chinmay Kubal,Kunal Kale	7-10-19	7-26-19	3	100%
1.1.1	Research paper finalization	Viranchee Patil	7-10-19	7-26-19	2	100%
1.2	Project Title	Kunal Kale,Viranchee Patil,Chinmay Kubal	7-10-19	7-26-19	2	100%
1.3	Abstract	Viranchee Patil	8-23-19	8-30-19	2	100%
1.4	Objectives	Kunal Kale	8-23-19	8-30-19	1	100%
1.5	Literature Review	Viranchee Patil	8-23-19	8-30-19	2	100%
1.6	Problem Definition	Kunal Kale,Viranchee Patil,Chinmay Kubal	3-23-18	8-30-19	1	100%
1.7	Scope	Kunal Kale,Viranchee Patil,Chinmay Kubal	8-23-19	8-30-19	1	100%
1.8	Technology stack	Kunal Kale,Viranchee Patil,Chinmay Kubal	8-23-19	8-30-19	1	100%
1.9	Benefits for environment	Chinmay Kubal	8-23-19	8-30-19	1	100%
1.1	Benefits for society	Viranchee Patil	8-23-19	8-30-19	1	100%
1.11	Applications	Kunal Kale	8-23-19	8-30-19	1	100%
2	**Project Design**					
2.1	Proposed System	Kunal Kale,Viranchee Patil,Chinmay Kubal	9-19-19	9-27-19	1	100%
2.2	Design(Flow Of Modules)	Kunal Kale	9-19-19	9-27-19	1	100%
2.3	Activity Diagram	Chinmay Kubal	9-19-19	9-27-19	1	100%
2.4	Use Case Diagram	Viranchee Patil	9-19-19	9-27-19	1	100%
2.5	Description Of Use Case	Kunal Kale	9-19-19	9-27-19	1	100%
2.6	Modules	Kunal Kale,Viranchee Patil,Chinmay Kubal	9-19-19	10-27-19	2	100%
2.6.1	Database	Chinmay Kubal	9-19-19	10-27-19	1	100%
2.6.2	SNS Server	Kunal Kale	9-19-19	10-27-19	1	100%
2.6.3	S3 Storage	Kunal Kale,Viranchee Patil,Chinmay Kubal	9-19-19	10-27-19	2	100%
2.7	Preparation Of Report	Viranchee Patil	9-19-19	10-27-19	3	100%
3	**Project Implementation**					
3.1	Database	Chinmay Kubal	1-27-20	3-4-20	1	100%
3.2	SNS Server	Kunal Kale	1-27-20	3-4-20	2	100%
3.3	S3 Storage	Kunal Kale,Viranchee Patil,Chinmay Kubal	1-27-20	3-4-20	5	100%
3.4	IOT Components Connectivity	Kunal Kale,Viranchee Patil,Chinmay Kubal	1-27-20	3-4-20	4	100%
4	**Testing**					
4.1	Design of Test Cases	Kunal Kale	1-27-20	3-4-20	5	100%
4.2	Testing	Kunal Kale	1-27-20	3-4-20	5	100%
5	**Results and Analysis**					
5.1	Analysis Of Results	Kunal Kale,Chinmay Kubal	2-27-20	3-4-20	4	100%
5.2	Report Preparation	Viranchee Patil, Kunal Kale	2-27-20	3-4-20	4	100%

PHASE ONE: WEEK 1, WEEK 2, WEEK 3 — PHASE TWO: WEEK 4, WEEK 5, WEEK 6 — PHASE THREE: WEEK 7, WEEK 8, WEEK 9 — PHASE FOUR: WEEK 10, WEEK 11, WEEK 12

Figure 6.1: Gantt Chart

Chapter 7

Conclusions and Future Scope

The development of IOT based Accident Alert System can prove to be vital in deciding the life and death of a car crash victim. Not only does it remove the need for the victim to call emergency services, but it also helps save time by automatically alerting the emergency help providers, and saving precious time during which the victim can proceed with first aid,etc.With this,our goal to develop a cheap accident detection system for low to mid end cars has been achieved.

For our future work, we would like to make an application on all mobile phone platforms, wherein the user can gain access to all the services offered by our current system, but with a GUI. The application will also monitor the speed, routes, nearby destinations,etc and provide the driver with a complete driving experience.

Bibliography

[1] Akriti Singhal, Sarishma, Ravi Tomar, "Intelligent Accident Management System using IoT and Cloud Computing" , 2nd International Conference on Next Generation Computing Technologies (NGCT-2016) Dehradun, India 14-16 October 2016

[2] Harit Sharma, Ravi Kanth Reddy, Archana Karthik, "S-CarCrash: Real-time crash detection analysis and emergency alert using smartphone" , International Conference on Connected Vehicles and Expo (ICCVE), 2016

[3] Md. Saeef Abdul Hadi, Abhijit Saha, Faysal Ahmad, Mohammad ShahriyarHasan, Mehebub Hasan Milon, "A Smart Accident Detection and Control System in Vehicular Networks" , 5th International Conference on Networking, Systems and Security (NsysS), 2018

[4] Hari Sankar S, Jayadev K, Suraj B, Aparna P, "A Comprehensive Solution to Road Traffic Detection and Ambulance Management" , International Conference on Advances in Electrical, Electronic and System Engineering, 14-16 Nov 2016

Appendices

Appendix-A: Installation and Configuration of Arduino IDE

Step 1: Download
1. Download the Arduino IDE file from Official Arduino website
https://www.arduino.cc/en/Main/Software.

Step 2: Install
1. Extract the downloaded folder and run the **arduino.exe** file to begin the installation process.
2. Accept all terms and conditions and wait to installation to complete.
3. Run **Arduino IDE** once the installation is finished.

Step 3: Adding NodeMCU in Arduino IDE
1. In Arduino IDE, go to File¿Preferences
2. Enter **http://arduino.esp8266.com/stable/package-esp8266com-index.json** into the "Additional Boards Manager" URLs field then, click the "OK" button.
3. Open the Boards Manager. Go to Tools ¿ Board ¿ Boards Manager.
4. Search for **ESP8266** and press install button for the "ESP8266 by ESP8266 Community"
5. Locate the newly added board **Nodemcu 1.0** to confirm completion of the process.

Appendix-B: Configuration of Amazon Web Services(AWS)

Launching EC2 Instance
1. Firstly, sign in to view the AWS Console.
2. Open the Amazon EC2 console by choosing **EC2** under Compute.
3. From the Amazon EC2 dashboard, choose **Launch Instance**.
4.On the Choose an Instance Type page, choose **t.2micro** as the hardware configuration of your instance and Review and Launch.
5. On the Review Instance Launch page, choose Launch.
6. In the Select an existing key pair or create a new key pair dialog box, choose **Create a**

new key pair, enter a name for the key pair, and then choose **Download Key Pair**. This is the only chance for you to save the private key file, so be sure to download it. Save the private key file in a safe place.
7. A confirmation page lets you know that your instance is launching.

Setting up Simple Notification Service in AWS

Step 1: Create a topic

1. Sign into the Amazon SNS Console.
2. In the Create topic section, enter a Topic name.
3. The topic is then created and it's ARN,Display name and Topic Owner's AWS ID are displayed in the Details section. Copy the displayed ARN.

Step 2: Create a subscription for an endpoint to the topic

1. On the navigation panel, choose Subscriptions¿Create subscription.
2. On the Create subscription page, do the following:

i) Enter the **Topic ARN** of the topic you created earlier.
ii) For **Protocol**, choose an endpoint type, for example Email (This is the type of alert that the emergency services and relatives of the user will receive)
iii) For **Endpoint**, enter an email address that can receive notifications.
iv) Choose **Create subscription**.

3. In your email client, check the email address that you specified and choose **Confirm subscription** in the email from Amazon SNS.
4. In your web browser, a subscription confirmation with your subscription ID is displayed.

Step 3: Publish a message to the topic

1. On the navigation panel, choose **Topics¿Publish Message**.
2. On the Publish message to topic page, do the following:

i) In the Message details section, enter the Subject.
ii) In the Message body section, choose **Identical payload for all delivery protocols**.
iii) Choose **Publish message**.

3. The entered email id will receive an email.

Acknowledgement

We have great pleasure in presenting the report on **IOT Based Accident Alert System** We take this opportunity to express our sincere thanks towards our guide **Prof. Kaushiki Upadhyaya** & Co-Guide **Prof. Geetanjali Kalme** Department of IT, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Prof. Kiran B. Deshpande** Head of Department,IT, APSIT for his encouragement during progress meeting and providing guidelines to write this report.

We thank **Prof. Anagha Aher** BE project co-ordinator, Department of IT, APSIT for being encouraging throughout the course and for guidance.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Student Name1: Kunal Kale
Student ID1: 16104027

Student Name2: Viranchee Patil
Student ID2: 16104044

Student Name3: Chinmay Kubal
Student ID3: 16104023