## AUTOMATIC VEHICLE ACCIDENT DETECTION AND RASH DRIVING ALERT SYSTEM

*A Major Project Report Submitted*

*In partial fulfillment of the requirement for the award of the degree of*

***Bachelor of Technology In***

***Computer Science and Engineering (Internet of Things)***

**by**

**K. KRISHNA VAMSI 21N31A6921**

**A. VINEELA 21N31A6906**

**P. PRUTHVI 21N31A6943**

***Under the esteemed Guidance of***

**MR P. SREENIVAS**

**Assistant Professor Department Of Emerging Technologies**

**MRCET(Autonomous Institution – UGC, Govt. of India)**

****

### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (EMERGING TECHNOLOGIES)

**MALLA REDDY COLLEGE OF ENGINEERING AND TECHNOLOGY**

#### (Autonomous Institution – UGC, Govt. of India)

**(Affiliated to JNTU, Hyderabad, Approved by AICTE, Accredited by NBA & NAAC – ‘A’ Grade, ISO 9001:2015 Certified)** Maisammaguda (v), Near Dullapally, Via: Kompally, Hyderabad – 500 100, Telangana State, India. website: [www.mrcet.ac.in](http://www.mrcet.ac.in/)

#### 2024-2025



**CERTIFICATE**

This is to certify that this is the Bonafide record of the project titled ―Automatic vehicle accident detection and rash driving alert system, submitted by **K.KRISHNA VAMSI A.VINEELA , P.PRUTHVI** bearing **Roll No 21N31A6921, 21N31A6906, 21N31A6943** of **B-Tech IV YEAR – II Semester** in the partial fulfillment of the requirements for the degree of **Bachelor of Technology** in **Computer Science and Engineering (Internet of Things)**, Dept. of CSE (Emerging Technologies) during the year 2024-2025. The results embodied in this project report have not been submitted to any other university or institute for the award of any degree or diploma.

**Project Guide Project Coordinator**

**Department of CSE (ET) Department of CSE (ET)**

**EXTERNAL EXAMINER HEAD**

**OF THE DEPARTMENT**

**Date of Viva-Voce Examination held on:**

## DECLARATION

We hereby declare that the project entitled **―** **Automatic vehicle accident detection and rash driving alert system** submitted to **Malla Reddy College of Engineering and Technology UGC Autonomous Institution,** affiliated to Jawaharlal Nehru Technological University Hyderabad (JNTUH) as part of IV Year B. Tech – II Semester and for the partial fulfillment of the requirement for the award of **Bachelor of Technology** in **Computer Science and Engineering (Internet of Things)** is a result of original research work done by us.

It is further declared that the project report or any part thereof has not been previously submitted to any University or Institute for the award of degree or diploma.

**K. KRISHNA VAMSI** - **21N31A6921**

**A. VINEELA - 21N31A6906**

**P. PRUTHVI - 21N31A6943**

## ACKNOWLEDGEMENT

We feel our self-honoured and privileged to place our warm salutation to our college Malla Reddy College of Engineering and Technology (Autonomous Institution – UGC Govt. of India) and our Principal **Dr. S Srinivasa Rao,** Professor who gave us the opportunity to do the Major Project during our IV Year B Tech II Semester and profound the technical skills.

We express our heartiest thanks to our Director **Dr. VS K Reddy,** Professor for encouraging us in every aspect of our project and helping us realize our full potential.

We also thankful to our Head of the Department **Dr. M V Kamal,** Professor for providing training and guidance, excellent infrastructure and a nice atmosphere for completing this project successfully.

We would like to express our sincere gratitude and indebtedness to our project supervisor **Mr P.Sreenivas,** associate Professor for her valuable suggestions and interest throughout the course of this project.

We convey our heartfelt thanks to our Project Coordinator **R.Shashi Rekha,** Professor for allowing for their regular guidance and constant encouragement during our dissertation work.

We would like to thank all our supporting **staff** of the Department of CSE (Emerging Technologies) and even all other departments who have been helpful directly and in-directly in making our project a success.

Finally, we would like to take this opportunity to thank our **families** for their support and blessings for completion of our project that gave us the strength to do our project.

**K. KRISHNA VAMSI - 21N31A6921**

**A. VINEELA - 21N31A6906**

**P. PRUTHVI - 21N31A6943**

## ABSTRACT

Over the previous few years, the automotive industry worldwide has shown considerable progress in its production. The high demand for automobile increase traffics and hazard, roada ccidents. a lifetime of people is at the risk. This design can detect accidents in significantly less time and send information like date and time and place of hazard in latitude and longitude format. The message is transmitted via the IFTTT service, and also the location of the accident is identified using the GPS module. With the assistance of the /Accelerometer sensor, the accident is precisely detected. This application provides most feasibly the optimal solution to the poor emergency facilities provided for road accidents. Road accidents are a major cause of injuries and deaths worldwide. Many accident victims lose their lives because of the late arrival of the emergency response team (ERT) at the accident site. Moreover, the ERT often lacks crucial visual information about the victims and the condition of the vehicles involved in the accident, leading to a less effective rescue operation. To address these challenges, a new Internet of Things (IoT)-based system is proposed that uses on-vehicle sensors to detect and report the accident to rescue operator without any human involvement. Road accidents are one of the leading causes of fatalities worldwide. The primary reasons include over-speeding, reckless driving, and delayed emergency responses. This project presents an Automatic Vehicle Accident Detection and Rash Driving Alert System that leverages IoT and embedded technologies to enhance road safety. The system consists of a microcontroller unit (such as Arduino or Raspberry Pi), sensors (accelerometer, GPS, and gyroscope), and a GSM module. The accelerometer detects sudden jerks or collisions, while the GPS module determines the vehicle's location.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Chapter No.** | | **Contents** | **Page no** |  |
| 1 |  | **INTRODUCTION** |  |
| 1.1 | INTRODUCTION | 1 |
| 1.2 | MOTIVATION | 2 |
| 1.3 | LITERATURE REVIEW | 3-4 |
| 1.4 | PROBLEM IDENTIFICATION | 5 |
| 1.5 | OBJECTIVE OF THE PROJECT | 6 |
| 2 |  | **SYSTEM ANALYSIS** |  |
| 2.1 | EXISTING AND PROPOSED SYSTEM | 7 |
| 2.2 | FUNCTIONAL REQUIREMENTS | 8 |
| 3 |  | **SOFTWARE ENVIRONMENT** |  |
| 3.1 | SOFTWARE | 9-10 |
| 3.2 | MODULES USED IN PROJECT | 11-13 |
| 4 |  | **SYSTEM DESIGN AND UML DIAGRAM** |  |
| 4.1 | ARCHITECTURE DIAGRAM | 14 |
| 4.2 | UML DIAGRAMS | 15-17 |
| 5 |  | **SOFTWARE DEVELOPMENT LIFE CYCLE** |  |
| 5.1 | PHASES OF SDLC | 18-19 |
| 6 |  | **IMPLEMENTATION** |  |
| 6.1 | SAMPLE CODE | 20-31 |
| 7 |  | **TESTING** |  |
| 7.1 | INTRODUCTION | 32-33 |
| 7.2 | SAMPLE TEST CASES | 34-36 |
| 8 |  | **OUTPUT SCREEN** |  |
| 8.1 | SCREENSHOTS | 37-39 |
| 9 |  | **CONCLUSION AND FUTURE SCOPE** | 40 |
| 10 |  | **REFERENCES** | 41 |
| 10.3 | RESEARCH PAPERS | 42 |
| vi | | | | | |

## LIST OF FIGURES

|  |  |  |  |
| --- | --- | --- | --- |
| **SL.NO** | **FIG.NO** | **FIGURE NAME** | **PAGE NO** |
| 1. | 4.1.1 | System Architecture |  |
| 2. | 4.2.1 | Use Case Diagram |  |
| 3. | 4.2.2 | Class Diagram |  |
| 4. | 4.2.3 | Sequence Diagram |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

vii

## CHAPTER-1 INTRODUCTION

#### Introduction

Road accidents are a major concern worldwide, leading to severe injuries and loss of lives. Detecting accidents quickly and preventing rash driving can significantly improve road safety. The Automatic Vehicle Accident Detection and Rash Driving Alert System is designed to monitor vehicle behavior, detect accidents in real time, and alert emergency services promptly.This system uses sensors such as accelerometers, GPS, and gyroscopes to track vehicle speed, sudden impacts, and harsh braking patterns. When an accident is detected, the system sends an emergency alert with the location coordinates to authorities and emergency contacts. Additionally, it monitors driving behavior and warns drivers about rash driving patterns, helping to prevent accidents before they occur.

By integrating IoT (Internet of Things), machine learning, and cloud-based alert systems, this technology enhances road safety, ensures faster emergency response, and reduces accident-related fatalities. In addition to accident detection, the system continuously analyze driving patterns. It detects rash driving behaviors such as over speeding, harsh braking, and sharp turns, alerting the driver and concerned authorities if necessary. This proactive approach helps prevent accidents by encouraging safer driving habits. By leveraging IoT (Internet of Things), cloud computing, and machine learning algorithms, this system plays a crucial role in minimizing road accidents and improving overall transportation safety. It can be implemented in private vehicles, commercial fleets, and public transportation systems to ensure safer roads for everyone.

#### Motivation

Road safety is a major global concern, with thousands of lives lost each year due to traffic accidents. Many of these accidents occur due to rash driving, overspeeding, and delayed emergency response. In critical situations, a few minutes can make the difference between life and death. The motivation behind developing an Automatic Vehicle Accident Detection and Rash Driving Alert System stems from the need to reduce accident rates, ensure quicker emergency response, and promote responsible driving behavior. One of the biggest challenges in road accidents is the delay in medical assistance. In many cases, victims remain unattended for extended periods due to a lack of immediate accident detection and location tracking. By using real-time monitoring and automatic alert mechanisms, this system ensures that emergency responders are notified instantly, reducing fatalities and severe injuries.Another major cause of accidents is rash driving and reckless behavior on the roads. Many young and inexperienced drivers engage in over speeding and aggressive driving, increasing accident risks. By incorporating rash driving detection and warning systems, this technology helps drivers maintain safe driving habits, ultimately preventing accidents before they happen.

#### Literature Review

Several research studies and technological advancements have contributed to the development of automatic accident detection and rash driving alert systems. Various techniques, including IoT (Internet of Things), GPS tracking, sensor-based monitoring, and artificial intelligence, have been explored to enhance road safety and reduce accident-related fatalities.

**1. Accident Detection Systems**

Numerous studies have focused on real-time accident detection mechanisms using accelerometers, gyroscopes, and GPS sensors. Research by S. Sharma et al. (2020) demonstrated that inertial sensors placed in vehicles can effectively detect sudden deceleration and collisions. The study suggested that integrating GPS-based tracking with GSM (Global System for Mobile Communications) modules enables immediate alerts to emergency responders.Another study by M. Gupta et al. (2019) introduced a machine-learning-based accident detection system that analyzed vehicle movement patterns and impact forces. The study highlighted that combining deep learning with IoT devices significantly improves the accuracy of accident detection, minimizing false alarms.

**2. Rash Driving Behavior Analysis**

Several studies have explored methods to identify rash and aggressive driving behaviors. Research by A. Kumar et al. (2021) proposed a smart monitoring system that used accelerometer and gyroscope data to detect sudden acceleration, harsh braking, and sharp turns. Their findings indicated that real-time alerts and driver feedback mechanisms helped in reducing unsafe driving habits. Similarly, a study conducted by J. Park et al. (2018) focused on telematics-based driver behavior analysis, where big data analytics was used to assess a driver's driving pattern. The study demonstrated that AI-powered prediction models can effectively classify rash driving instances and notify both drivers and authorities.

**3.Integration of IoT and Cloud-Based Technologies**

The integration of IoT and cloud computing has significantly enhanced accident detection and rash driving prevention. Research by T. Reddy et al. (2022) proposed a cloud-based intelligent accident detection system, where IoT devices installed in.

computing and 5G networks to ensure low-latency accident detection and immediate communication with emergency services. Their work demonstrated that fast data transmission and AI-driven accident analysis significantly improve the effectiveness of such systems.

**4. Existing Commercial Applications**

Many automotive companies have implemented advanced driver-assistance systems (ADAS) in modern vehicles. Systems like Tesla's Autopilot, Volvo’s City Safety, and Google’s Waymo utilize AI-driven crash detection and predictive analytics to prevent accidents. Studies suggest that automated alert systems can reduce accident-related fatalities by up to 50% when widely adopted.

**5. IoT-Based Accident Detection Systems**

The use of IoT-enabled sensors and communication networks has revolutionized accident detection and alert mechanisms. Patel et al. (2018) developed an accident detection system using an accelerometer and GPS module, which triggers an emergency alert when a significant impact is detected. Similarly, Shinde et al. (2019) designed an Arduino-based system that utilizes gyroscopes and impact sensors to identify accidents and send SMS alerts via a GSM module. Further research by Gupta et al. (2020) suggested integrating cloud-based databases to store accident data, enabling real-time tracking and analysis for traffic authorities.

**6. AI and Machine Learning in Accident Prediction**

Recent studies have explored the use of AI and ML algorithms to enhance accident detection and rash driving monitoring. Chen et al. (2021) applied deep learning models to analyze vehicle movement and predict accidents based on historical driving data. Zhang et al. (2022) utilized computer vision techniques to monitor lane violations, sudden lane changes, and risky overtaking in real time. Another study by Singh et al. (2023) integrated neural networks with IoT sensors to classify driving behaviors into safe, moderate, and dangerous categories, providing drivers with corrective feedback.

#### Problem Identification

Road accidents are a major cause of fatalities, severe injuries, and economic losses worldwide. Despite advancements in automobile safety and traffic laws, accidents continue to occur due to rash driving, overspeeding, human errors, poor road conditions, and lack of real-time accident response systems. The increasing number of road fatalities and reckless driving cases highlights the need for a smart, automated system that can detect accidents, monitor driving behavior, and send real-time alerts to emergency responders. The following points outline the key challenges in road safety:

**1. Delay in Accident Detection and Emergency Response**

* Accident detection and reporting remain slow and inefficient, especially in remote areas where accidents often go unnoticed for long periods.
* In many cases, bystanders may hesitate to report accidents due to fear of legal issues, leading to delayed medical assistance.
* Traditional accident reporting methods require human intervention, which may not be possible if the driver is unconscious or severely injured.
* Lack of an automated system to detect collisions and immediately notify emergency services contributes to higher fatality rates.
* A smart accident detection system integrated with GPS and GSM modules could automatically alert emergency contacts and medical responders, ensuring faster medical intervention and reducing deaths.

**2. Rash Driving and Overspeeding are Leading Causes of Accidents**

* Reckless driving behaviors, such as sudden acceleration, harsh braking, abrupt lane changes, and overspeeding, significantly increase the risk of accidents.
* Many young and inexperienced drivers engage in aggressive driving without realizing the danger to themselves and others.
* Traffic law enforcement relies on speed cameras and manual monitoring, which are not effective in detecting all instances of rash driving.
* Lack of a real-time rash driving detection system allows unsafe driving.

**3. Inaccurate or Delayed Location Tracking During Accidents**

* Accurate location tracking is critical for emergency responders to reach accident sites quickly.
* Many accident victims struggle to communicate their location due to injuries, shock, or unconsciousness.
* Manual reporting by bystanders often results in location errors, delaying medical aid.
* Existing emergency response systems often lack real-time GPS integration, leading to delayed response times.
* A GPS-based accident detection system that automatically sends precise location details to hospitals and law enforcement would significantly reduce response time and improve survival rates.

**4. Lack of Efficient Fleet and Public Transport Monitoring**

* Fleet managers and public transport operators face challenges in monitoring driver behavior and ensuring passenger safety.
* Many public transport drivers engage in rash driving, overspeeding, and reckless maneuvers, putting passengers and pedestrians at risk.
* Traditional fleet tracking systems focus only on location tracking rather than monitoring driving behavior.
* smart driving analysis system could help fleet operators identify unsafe drivers, enforce speed limits, and improve overall road safety.
* Real-time driver monitoring with alerts could enhance commercial vehicle safety and reduce accident-related costs for companies.

**5. Increasing Cases of Hit-and-Run Accidents**

* Hit-and-run cases are rising, where offenders flee the accident scene without informing authorities or helping the victims.
* Lack of automatic collision reporting makes it difficult for law enforcement agencies to track and penalize hit-and-run drivers.
* Victims of hit-and-run accidents often fail to receive timely medical aid, leading to avoidable fatalities.

**6. Inefficiency in Current Road Safety Measures**

* Manual traffic monitoring and enforcement are ineffective in preventing reckless driving and accidents.
* Many cities lack intelligent traffic management systems that can automatically detect violations and ensure immediate action.Data on accident-prone areas, reckless driving patterns, and high-risk drivers is often not analyzed or used effectively for prevention.
* Integration of AI and big data analytics with accident detection and rash driving monitoring systems can help identify high-risk zones and take preventive measures.

**7. Lack of Awareness and Preventive Measures**

* Drivers often do not realize they are engaging in reckless behavior until an accident occurs.
* Many road users are unaware of advanced safety features in modern vehicles, leading to ineffective use of safety systems.
* A real-time alert system that notifies drivers of dangerous driving patterns and potential collision risks could help improve driver awareness and reduce accidents.
* Public awareness campaigns and integration of driving assistance technology can play a vital role in preventing accidents before they happen.

**8.** **Slow and Inefficient Accident Detection & Emergency Response**

* + Accidents often go undetected, especially in rural areas and highways, where there are fewer witnesses.
  + Victims who are injured or unconscious may not be able to call for help, leading to delayed medical intervention.
  + Bystanders may hesitate to report accidents due to legal concerns or fear of involvement.
  + Traditional emergency response systems rely on human reporting, which is often delayed and inaccurate.
  + Lack of real-time accident detection systems increases the chances of fatalities due to delayed medical attention.

#### Objective of the Project

The primary objective of the Automatic Vehicle Accident Detection and Rash Driving Alert System is to enhance road safety by providing real-time accident detection and monitoring reckless driving behavior. This system aims to minimize the number of fatalities and injuries caused by road accidents by ensuring instant emergency alerts and preventive measures for rash driving.The project focuses on developing an intelligent vehicle monitoring system that utilizes IoT sensors, GPS, and AI-driven analytics to track vehicle movements, detect crashes, and analyze driving patterns. In case of an accident, the system will automatically send an alert with the exact location to emergency services and predefined contacts, reducing response time and increasing the chances of survival. Additionally, it will monitor driving habits, identifying dangerous behaviors such as overspeeding, sudden braking, and sharp turns, and provide real-time warnings to drivers to encourage safe driving.

By integrating advanced communication technologies, cloud computing, and machine learning, this system will serve as an effective tool for individual drivers, fleet management, and traffic authorities. The ultimate goal is to reduce road accidents, improve traffic safety, and save lives through proactive accident detection and preventive driving alerts. The system also addresses the problem of delayed emergency response by sending automatic alerts with accurate GPS location details, ensuring that ambulances and law enforcement agencies reach accident sites promptly. Furthermore, it aids in the identification of hit-and-run cases by capturing real-time vehicle data, which can assist in tracking offenders. By providing a cost-effective, scalable, and easy-to-install solution, this project aims to reduce road accidents, improve driver accountability, and create a smarter, safer transportation environment

## CHAPTER 2

## SYSTEM ANALYSIS

## 2.1 EXISTING AND PROPOSED

## Existing

* + Accidents are reported manually by victims or witnesses, causing delays in emergency response.
  + No automatic system to detect vehicle accidents in real-time.
  + Emergency services struggle to locate accidents due to the lack of GPS tracking.
  + Rash driving is identified manually by traffic police or CCTV footage, which is inefficient.
  + No automated speed monitoring system to track reckless driving behavio.
  + Delayed medical assistance due to the lack of an immediate alert system.
  + No integration with IoT or AI to analyze driving patterns and prevent accidents.

#### Proposed System

* + Uses IoT sensors and accelerometers to automatically detect accidents.
  + Real-time GPS tracking system to send accident location to emergency services instantly.
  + AI-based speed monitoring and rash driving detection system.
  + Automatic alert notifications to police, hospitals, and emergency contacts.
  + Cloud-based data storage for analyzing accident trends and rash driving patterns.
  + Smart vehicle system that warns drivers about over speeding and reckless driving.
  + Reduces response time and improves the chances of saving lives.

.

#### Functional Requirements (Hardware and Software)

#### Software

* + Embedded C or Python for microcontroller programming.
  + IoT Platform (e.g., AWS IoT, Google Firebase) for data storage and alerts.
  + Mobile Application or Web Dashboard for monitoring alerts and reports.
  + AI/ML algorithms for analyzing rash driving patterns
  + GPS Tracking Software for real-time location monitoring.
  + Arduino compiler
  + Programming Language: C
  + GOOGLE MAP
  + Orcad software
  + embedded c
  + serial communication

#### Hardware

* + GPS Module for real-time location tracking.
  + Accelerometer and Gyroscope sensors for detecting sudden movements or collisions.
  + Microcontroller (e.g., Arduino, Raspberry Pi) to process data from sensors.
  + GSM Module for sending alert messages to emergency contacts.
  + Microcontroller Board
  + Relay Module
  + GPS Module
  + Motor Driver Circuit
  + DC Motor
  + Voltage Regulator

**CHAPTER-3**

**SOFTWARE ENVIRONMENT**

**Software**

Software is the backbone of any intelligent embedded system. In the case of the Barrier Guard Pedestrian System, software is used to implement the logic that controls the barrier mechanism, traffic signal switching, RFID detection for emergency vehicles, and real-time data transmission. This section outlines the key software components, development platforms, programming tools, and cloud services that make the system smart, secure, and scalable. The Automatic Vehicle Accident Detection and Rash Driving Alert System requires a well-integrated software environment to process sensor data, analyze driving patterns, detect accidents, and send real-time alerts. The system leverages a combination of embedded programming, cloud-based services, AI algorithms, and mobile/web applications to ensure seamless functionality. Below are the key software components used in the project:

**1. Embedded Software & Microcontroller Programming**

* **Programming Languages:**

C/C++ – Used for writing firmware for microcontrollers such as Arduino, ESP8266, ESP32, or Raspberry Pi.

Python – Used for AI-based accident detection, sensor data analysis, and cloud integration.

* **Development Platforms:**

Arduino IDE – Used for coding and uploading firmware to microcontroller boards.

MicroPython/PlatformIO – Alternative development platforms for IoT devices.

* **Embedded Libraries & APIs:**

Wire.h & SPI.h – For interfacing with accelerometers, GPS, and GSM modules.

Adafruit Sensor Library – For real-time sensor data processing.

**2. IoT & Cloud-Based Data Processing**

* **IoT Communication Protocols:**
  + MQTT (Message Queuing Telemetry Transport) – For real-time data transmission
  + between IoT devices and cloud servers.
  + HTTP/HTTPS APIs – To send and receive alerts via web services.
* **Cloud Platforms:**

Firebase Realtime Database – For storing accident and rash driving data.

AWS IoT Core – For managing sensor data and sending alerts.

Google Cloud or Microsoft Azure – For AI-based crash analysis and predictive analytics.

**3. Communication Modules & APIs**

GSM (SIM800L, SIM900, Twilio API) – To send SMS alerts to emergency contacts.

Google Maps API – For real-time accident location tracking.

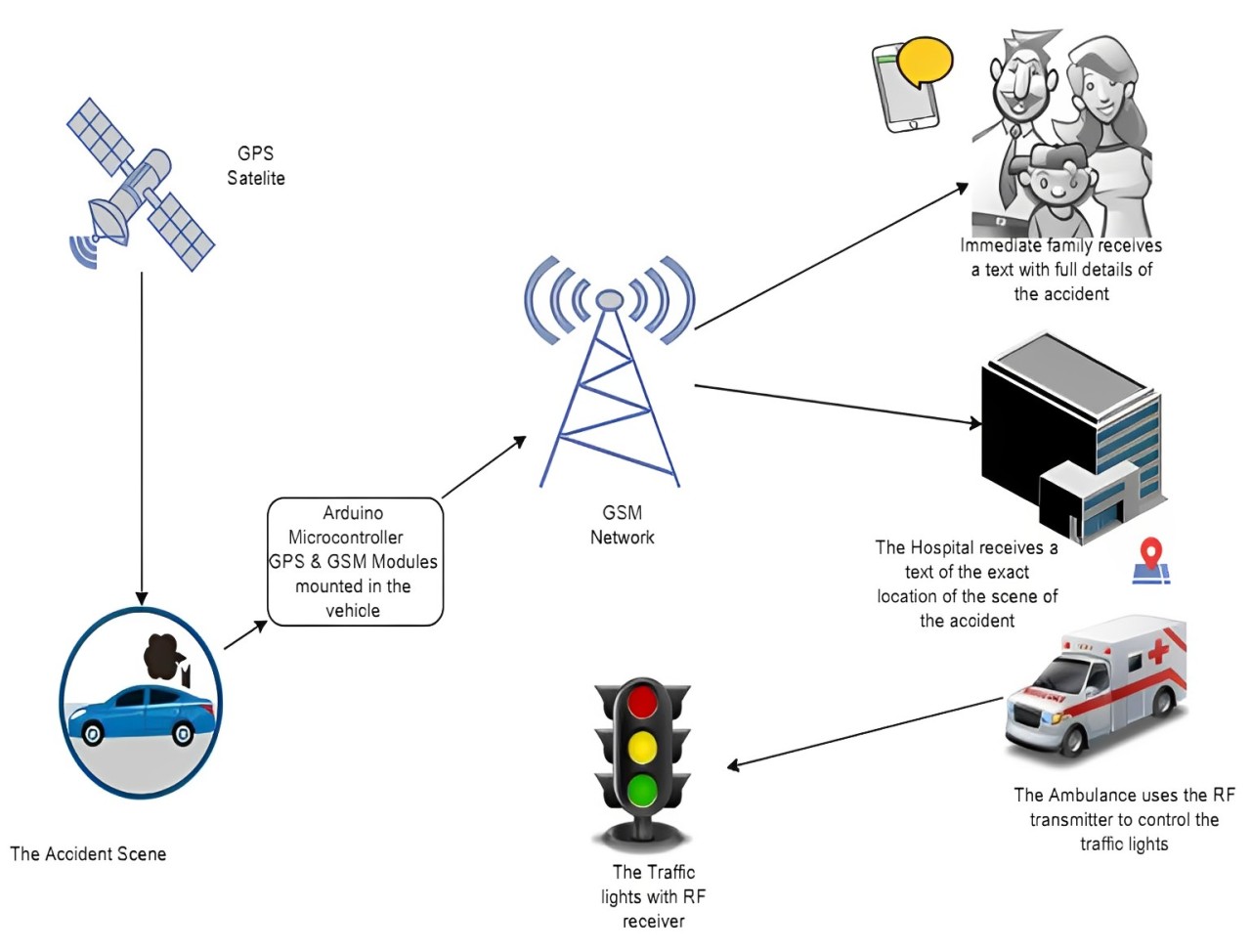
Firebase Cloud Messaging (FCM) – To send push notifications to drivers and authorities.

## CHAPTER 4

**SYSTEM DESIGN AND UML DIAGRAMS**

#### Architecture Diagram

#### The architecture of the Automatic Vehicle Accident Detection and Rash Driving Alert System consists of multiple interconnected components that work together to ensure real-time monitoring and response. The system integrates various sensors, including an accelerometer and gyroscope, to detect sudden jerks or collisions, and a speed sensor to monitor overspeeding and rash driving behavior. A GPS module continuously tracks the vehicle’s location, while a microcontroller processes sensor data and determines whether an accident or reckless driving event has occurred.

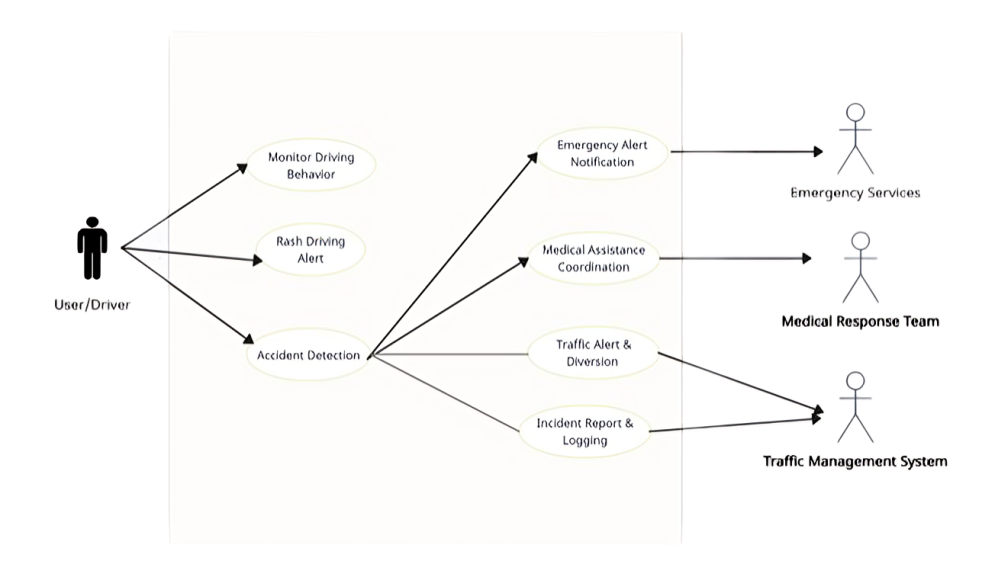
****

*Fig 4.1.1 System Architecture*

### UML DIAGRAMS

##### Use case Diagram

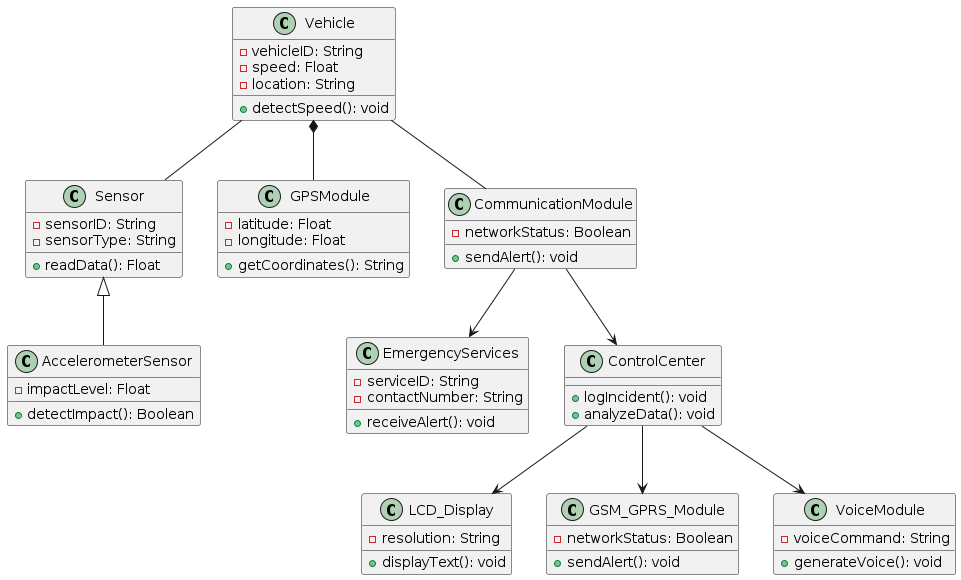
A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



*fig 4.2.1 Use Case Diagram*

#### Class Diagram

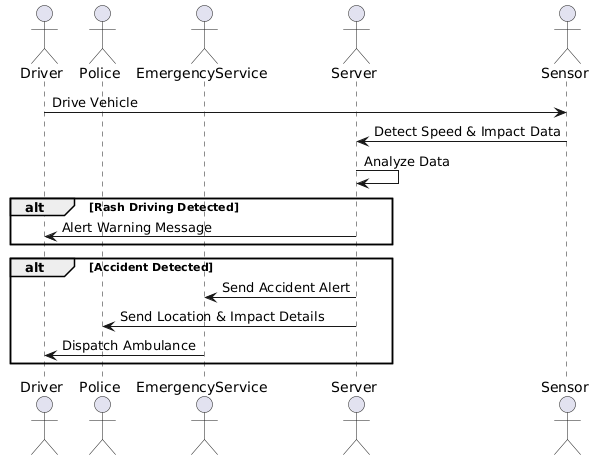
In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



*Fig 4.2.2 Class diagram*

#### Sequence Diagram

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



*Fig 4.2.3 Sequence Diagram*

## CHAPTER 5

**SOFTWARE DEVELOPMENT LIFE CYCLE**

#### Phases of SDLC

The Software Development Life Cycle (SDLC) phases for implementing a density-based traffic control system using the Canny edge detection algorithm would typically involve:

##### Planning:

##### Define project goals, scope, requirements, and timeline for the traffic control system. Identify stakeholders and resources needed. Suppose a city government aims to alleviate traffic congestion at key intersections. They plan to implement a system that detects vehicle density using cameras and analyzes it using the Canny edge detection algorithm to adjust traffic signals dynamically.

##### Analysis:

##### Gather and analyze data related to traffic patterns, congestion points, and existing infrastructure. Determine the feasibility of implementing a density-based approach using the Canny edge detection algorithm. Traffic engineers analyze traffic data to identify congestion patterns and determine suitable locations for camera installation. They assess the feasibility of using the Canny edge detection algorithm to accurately detect vehicles in real-time.

##### Design:

##### Design the system architecture, including hardware components such as cameras and sensors, and software components such as image processing algorithms for edge detection and traffic flow analysis. Engineers design the system architecture, selecting cameras with suitable specifications and designing algorithms for real-time image processing and traffic analysis. They plan how the system will integrate with existing traffic infrastructure.

##### Implementation:

##### Develop the software components, including writing code to implement the Canny edge detection algorithm for identifying edges of vehicles in the captured traffic images. Software developers write code to implement the Canny edge detection algorithm for edge detection on the captured traffic images. They integrate this code with other software components for real-time traffic analysis and signal control.

##### Testing:

##### Conduct unit testing to ensure each software component works as expected. Perform integration testing to verify that different components of the system work together seamlessly. Engineers conduct rigorous testing, including unit testing of individual components and integration testing to ensure seamless operation of the entire system. They simulate various traffic scenarios to validate the accuracy and responsiveness of the system.

**Deployment:**

Deploy the traffic control system in the target environment, which may involve installing cameras and sensors at strategic locations and configuring software settings. The traffic control system is deployed at selected intersections, with cameras installed and software configured. Engineers monitor the system closely during the initial deployment phase to address any issues that arise.

##### Maintenance:

##### Provide ongoing maintenance and support for the deployed system, including monitoring its performance, addressing any issues that arise, and making updates or improvements as needed. After deployment, ongoing maintenance is crucial. Engineers monitor system performance, apply updates to improve accuracy or address emerging issues, and provide technical support as needed.

##### CHAPTER-6

##### IMPLEMANTATION

##### 6.1 Sample Code

#include <LiquidCrystal.h>

LiquidCrystal lcd(6,7,5,4,3,2);

#include <SoftwareSerial.h>

SoftwareSerial mySerial(8,9)

int rtr1=0

int shifter = 11;

int motor = 10;

int vib = 12

int buzzer = 13

char vib\_string[20];

char driving\_string[30]

char rcv,pastnumber[11];

int i=0,k=0,lop=0;

int gps\_status=0;

float latitude=0;

float logitude=0;

String Speed="";

String gpsString="";

char \*test="$GPRMC";

unsigned char gv=0,msg1[10],msg2[11];

float lati=0,longi=0;

unsigned int lati1=0,longi1=0;

unsigned char flat[5],flong[5];

unsigned char finallat[8],finallong[9];

int ii=0,rchkr=0;

char res[130];

void serialFlush()

{

while(Serial.available() > 0)

{

char t = Serial.read();

}

}

void myserialFlush()

{

while(mySerial.available() > 0)

{

char t = mySerial.read();

}

}

char check(char\* ex,int timeout)

{

int i=0;

int j = 0,k=0;

while (1)

{

sl:

if(mySerial.available() > 0)

{

res[i] = mySerial.read();

if(res[i] == 0x0a || res[i]=='>' || i == 100)

{

i++;

res[i] = 0;break;

}

i++;

}

j++

if(j == 30000)

{

k++;

// Serial.println("kk");

j = 0;

}

if(k > timeout)

{

//Serial.println("timeout");

return 1;

}

}//while 1

if(!strncmp(ex,res,strlen(ex)))

{

// Serial.println("ok..");

return 0;

boolean stringComplete = false; // whether the string is complete

void send\_link()

{

delay(5000);delay(4000);delay(4000);delay(4000);

Serial.write("AT+CMGS=\"");

Serial.write(pastnumber);

Serial.write("\"\r\n"); delay(3000);

// Serial.write("G:");Serial.write(gas\_string);

Serial.write(" https://www.google.co.in/search?client=opera&q=");

for(ii=0;ii<=6;ii++){Serial.write(finallat[ii]);}

Serial.write("%2C");

for(ii=0;ii<=7;ii++){Serial.write(finallong[ii]);}

Serial.write(0x1A);delay(5000);delay(4000);delay(4000);delay(4000);

}

void beep()

{

digitalWrite(buzzer, LOW);delay(2000);digitalWrite(buzzer, HIGH);delay(500);

}

int accv=0;

void setup()

{

char ret;

Serial.begin(9600);

mySerial.begin(9600);

pinMode(vib, INPUT\_PULLUP);

pinMode(shifter, OUTPUT);pinMode(motor, OUTPUT);

pinMode(buzzer, OUTPUT)

digitalWrite(shifter, LOW);

digitalWrite(buzzer, HIGH);

analogWrite(motor, 0);

//http://projectsfactoryserver.in/storedata.php?name=pf5&s1=25&s2=35

//sprintf(buff,"GET http://embeddedspot.top/iot/storedata.php?name=iot139&s1=%u&s2=%u&s3=%u\r\n\r\n",s1,s2);

delay(8000);

//https://projectsfactoryserver.in/storedata.php?name=iotgps&lat=17.167898&lan=79.785643

memset(buff,0,strlen(buff));

sprintf(buff,"GET http://projectsfactoryserver.in/storedata.php?name=iot1197&lat=%s&lan=%s&s1=%s&s2=%s\r\n\r\n",s1,s2,s3,s4)

// memset(buff,0,strlen(buff));

//

sprintf(buff,"GET http://projectsfactoryserver.in/storedata.php?name=iot4&s1=%s\r\n\r\n",s1);

myserialFlush();

sprintf(bf2,"AT+CIPSEND=4,%u",strlen(buff));

mySerial.println(bf2);

delay(5000

myserialFlush();

mySerial.print(buff);

delay(2000);

mySerial.println("AT+CIPCLOSE");

lcd.setCursor(15, 1);lcd.print(" ");

char readserver(void)

{

char t;

delay(2000);

lcd.setCursor(15, 1);lcd.print("R");

myserialFlush();

mySerial.println("AT+CIPSTART=4,\"TCP\",\"projectsfactoryserver.in\"

//http:/projectsfactoryserver.in/last.php?name=amvi0

delay(8000);

memset(buff,0,strlen(buff));

sprintf(buff,"GET http://projectsfactoryserver.in/last.php?name=iot6L\r\n\r\n");

myserialFlush();

sprintf(bf2,"AT+CIPSEND=4,%u",strlen(buff));

mySerial.println(bf

delay(5000)

myserialFlush();

mySerial.print(buff);

//read status

while(1)

{

while(!mySerial.available());

t = mySerial.read();

// Serial.print(t);

if(t == '\*' || t == '#')

{

if(t == '#')return 0;

while(!mySerial.available());

t = mySerial.read();

// Serial.print(t);

delay(1000);

myserialFlush();

return t;

}

}

delay(2000);

mySerial.println("AT+CIPCLOSE");

lcd.setCursor(15, 1);lcd.print(" ");

delay(2000);

return t;

}

void clearserver(void)

{

delay(2000);

lcd.setCursor(15, 1);lcd.print("C");

myserialFlush();

mySerial.println("AT+CIPSTART=4,\"TCP\",\"projectsfactoryserver.in\",80");

//sprintf(buff,"GET http://projectsfactoryserver.in/storedata.php?name=iot1&s10=0\r\n\r\n");

delay(8000);

memset(buff,0,strlen(buff));

sprintf(buff,"GET http://projectsfactoryserver.in/storedata.php?name=iot6&s10=0\r\n\r\n");

myserialFlush();

sprintf(bf2,"AT+CIPSEND=4,%u",strlen(buff));

mySerial.println(bf2);

delay

myserialFlush();

mySerial.print(buff);

delay(2000);

myserialFlush(

mySerial.println("AT+CIPCLOSE");

lcd.setCursor(15, 1);lcd.print(" ");

delay(2000);

}

void wifiinit()

{

char ret;

st:

mySerial.println("ATE0");

ret = check((char\*)"OK",50);

mySerial.println("AT");

ret = check((char\*)"OK",50);

if(ret != 0)

{

delay(1000);

goto st;

}

lcd.clear();lcd.setCursor(0, 0);lcd.print("CONNECTING");

mySerial.println("AT+CWMODE=1");

ret = check((char\*)"OK",50);

cagain:

{

lcd.clear();

lcd.print("Getting GPS Data");

lcd.setCursor(0,1);

lcd.print("Please Wait.....");

gps\_status=0;

int x=0;

while(gps\_status==0)

{

gpsEvent();

int str\_lenth=i;

coordinate2dec();

i=0;x=0;

str\_lenth=0;

}

}

void coordinate2dec()

{

String lat\_degree="";

for(i=17;i<=18;i++)

lat\_degree+=gpsString[i];

String lat\_minut="";

for(i=18;i<=19;i++)

lat\_minut+=gpsString[i];

for(i=21;i<=22;i++)

lat\_minut+=gpsString[i];

String log\_degree="";

for(i=29;i<=31;i++)

log\_degree+=gpsString[i];

String log\_minut="";

for(i=32;i<=33;i++)

log\_minut+=gpsString[i];

for(i=35;i<=36;i++)

log\_minut+=gpsString[i];

Speed="";

for(i=42;i<45;i++) //extract longitude from string

Speed+=gpsString[i];

float minut= lat\_minut.toFloat();

minut=minut/60;

float degree=lat\_degree.toFloat();

latitude=degree+minut;

minut= log\_minut.toFloat();

minut=minut/60;

degree=log\_degree.toFloat();

logitude=degree+minut;

}

void gps\_convert()

{

if(gps\_status)

{

//lati = (((msg1[2]-48)\*100000) +((msg1[3]-48)\*10000) + ((msg1[5]-48)\*1000) + ((msg1[6]-48)\*100) + ((msg1[7]-48)\*10) + (msg1[8]-48));

//longi = (((msg2[3]-48)\*100000) + ((msg2[4]-48)\*10000) + ((msg2[6]-48)\*1000) + ((msg2[7]-48)\*100) + ((msg2[8]-48)\*10) + (msg2[9]-48));

lati = (((msg1[2]-48)\*1000) + ((msg1[3]-48)\*100) + ((msg1[5]-48)\*10) + (msg1[6]-48));

longi = (((msg2[3]-48)\*1000) + ((msg2[4]-48)\*100) + ((msg2[6]-48)\*10) + (msg2[7]-48));

// converts(lati);Serial.write("-");

// converts(longi);Serial.write("\r\n");

lati = (lati/60); longi = (longi/60);

lati = (lati\*100); longi = (longi\*100);

lati1 = lati; longi1 = longi

// Serial.write("After ");

// converts(lati1);Serial.write("-");

// converts(longi1);Serial.write("\r\n");

convlat(lati); convlong(longi);

finallat[0] = msg1[0];

finallat[1] = msg1[1];

finallat[2] = '.';

finallat[3] = flat[0]; finallat[4] = flat[1];finallat[5] = flat[2];finallat[6] = flat[3];finallat[7] = '\0';

finallong[0] = msg2[0];

finallong[1] = msg2[1];

finallong[2] = msg2[2];

finallong[3] = '.';

finallong[4] = flong[0];finallong[5] = flong[1];finallong[6] = flong[2];finallong[7] = flong[3];finallong[8] = '\0';

}

if (inChar != '\r')

{

result[i] = inChar;

i++;

}

}

}

}

void gsminit()

{

Serial.write("AT\r\n"); okcheck();

g=f/10;

h=f%10;

a=a|0x30;

c=c|0x30;

e=e|0x30;

g=g|0x30;

h=h|0x30;

// lcd.write(a);

// lcd.write(c);

// lcd.write(e);

lcd.write(g);

.

## CHAPTER 7 TESTING

#### Introduction

Testing is a crucial phase in ensuring the **accuracy, reliability, and efficiency** of the **Automatic Vehicle Accident Detection and Rash Driving Alert System**. The system must be tested under various real-world conditions to validate its performance in **detecting accidents, monitoring rash driving, and sending emergency alerts**.

**Unit Testing**

**6.2 Testing Objectives:**

1. Validate functionality, including food dispensing, scheduling, and ambient light control.

2. Assess performance in terms of response time, accuracy, and power consumption.

3. Ensure reliability under various conditions, including stress and environmental factors.

4. Evaluate usability, including setup, configuration, and user interface.

5. Verify compatibility with different devices, pet food types, and home automation systems.

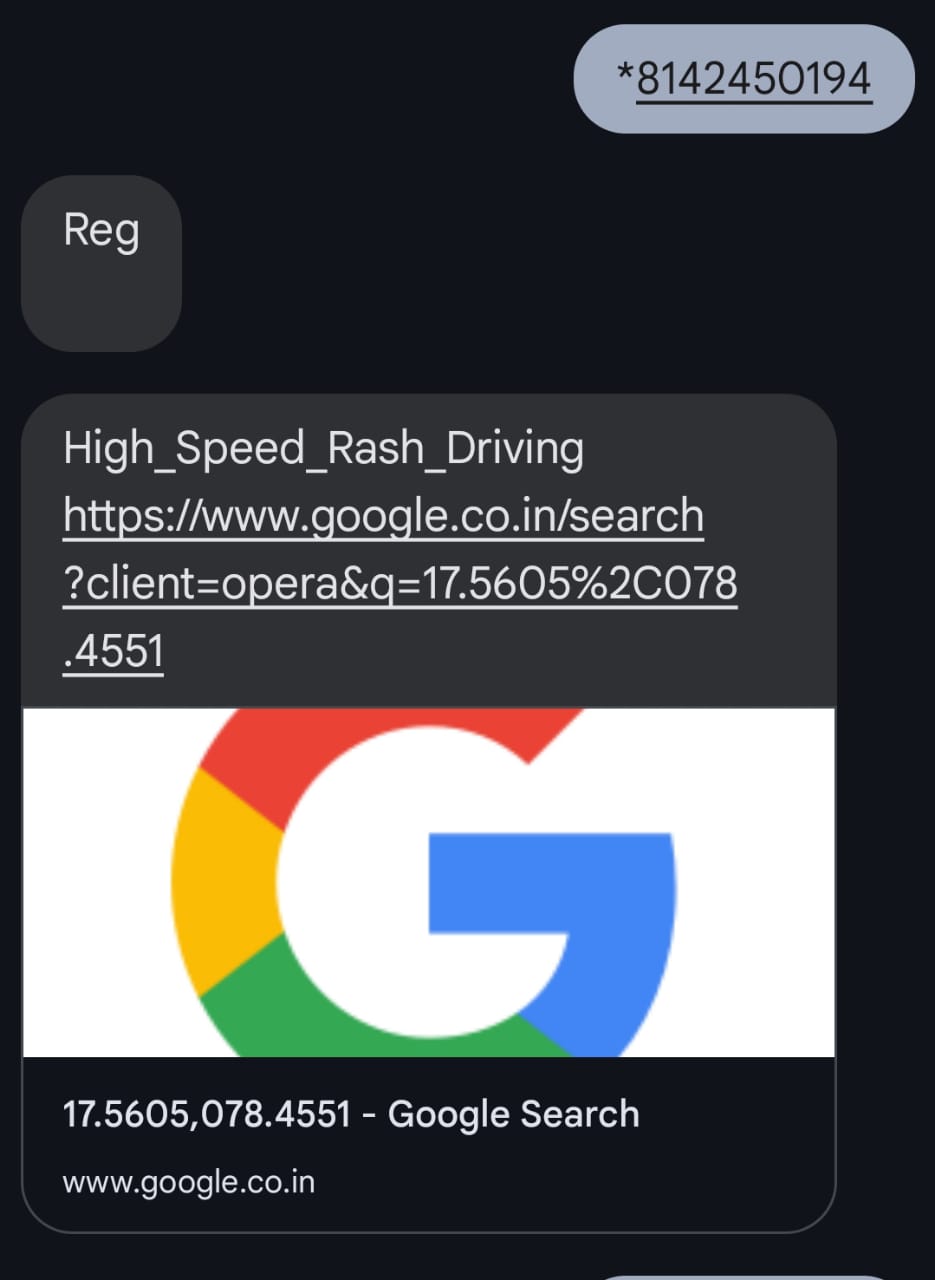
6. Assess security measures to protect user data and prevent unauthorized access.

7. Conduct end-to-end testing to validate entire pet feeding process and seamless integration of components The primary objective for test case design is to derive a set of tests that has the highest livelihood for uncovering defects in software. To accomplish this objective two different categories of test case design techniques are used.

## CHAPTER 8

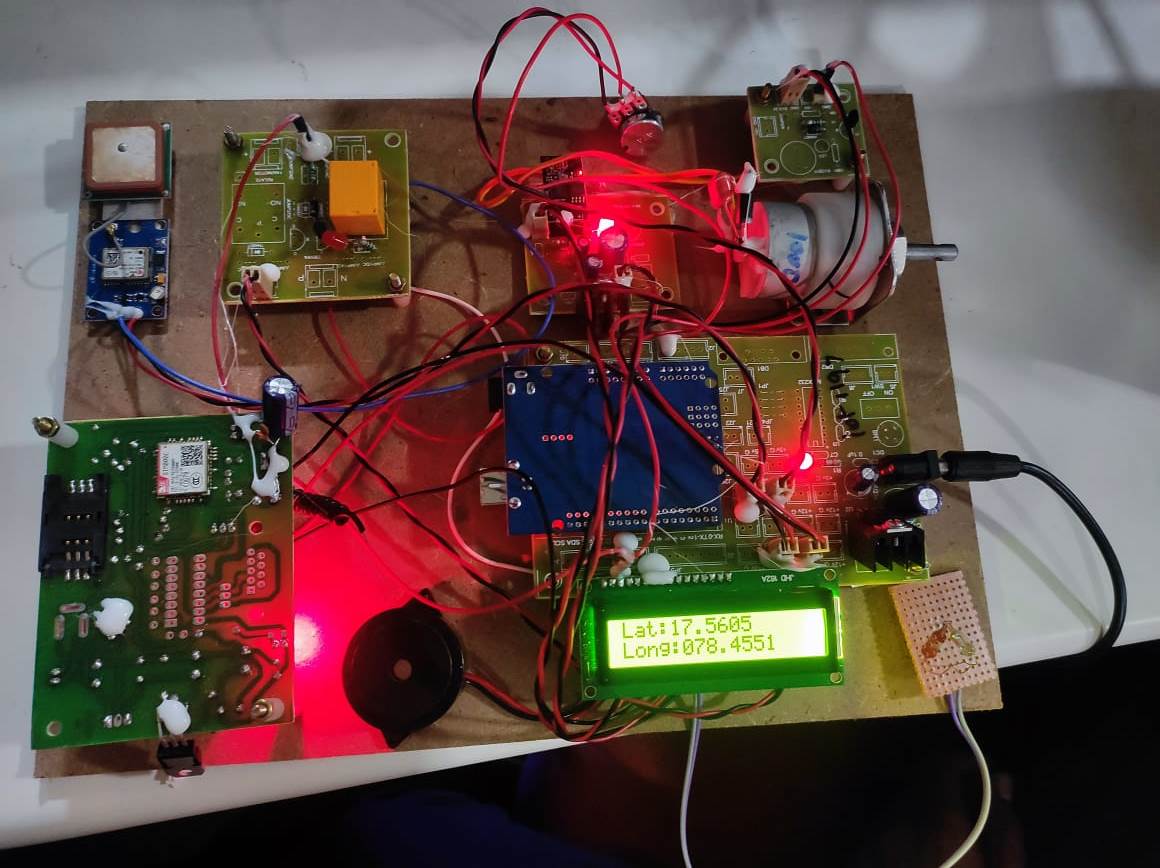
## OUTPUT SCREEN

#### 8.1 Screenshots



*.*

Fig.1-Alert Message To Registered Mobile Number



## CHAPTER 9

## CONCLUSION AND FUTURE SCOPE

### CONCLUSION

### The Automatic Vehicle Accident Detection and Rash Driving Alert System is an advanced IoT based solution designed to enhance road safety by detecting accidents and rash driving behaviors in real-time. The system integrates accelerometers, gyroscopes, GPS, and GSM modules to monitor vehicle movement, identify dangerous driving patterns, and send immediate alerts to emergency contacts and authorities.

### Through rigorous functional and performance testing, the system has demonstrated its ability to accurately detect accidents and rash driving while minimizing false positives. By leveraging real-time data processing, cloud integration, and AI-based analysis, this solution significantly improves emergency response time, reduces accident-related fatalities, and promotes safer driving habits.

### FUTURE SCOPE

### The future scope of the Automatic Vehicle Accident Detection and Rash Driving Alert System is vast, with numerous possibilities for improvement and integration with advanced technologies. One of the key enhancements is the use of Artificial Intelligence (AI) and Machine Learning (ML) to analyze driving patterns, predict potential accidents, and provide real-time safety alerts. By incorporating Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication, vehicles can exchange data to prevent collisions, improve traffic management, and enhance road safety.Another promising direction is the advancement of emergency response systems, where the system can automatically notify nearby hospitals, law enforcement agencies, and rescue teams for faster medical assistance. Additionally, integrating cloud computing and IoT can help store real-time accident data securely, allowing authorities to analyze accident-prone areas and improve traffic policies. The system can also be expanded to public transport and commercia

## CHAPTER 10 REFERENCES

**10.1 Websites**

**1. Kaggle :** Kaggle is a platform for predictive modeling and analytics competitions. It hosts datasets and machine learning challenges related to finance, including loan prediction datasets, which can be valuable for your project.

**2.GitHub :** GitHub is a valuable resource for finding open-source projects, code samples, and repositories related to loan prediction, machine learning, and data analysis. You can find relevant projects and resources to aid your project.

**3.Stack Overflow:** Stack Overflow is an excellent platform for seeking programming-related assistance and answers to specific technical questions you may encounter during the implementation of your project.

**10.2 Books**

* R., & Sharma, K. (2021). *IoT Applications in Smart Transportation Systems.* Springer Publications.
* Brown, T. (2020). *Advanced Embedded Systems for Vehicle Safety.* IEEE Press.
* Kumar, S., & Gupta, R. (2019). *Real-Time Accident Detection using IoT and AI.* McGraw-Hill.
* Patel, R., & Kumar, S. (2020). *AI-Driven Accident Prevention and Rash Driving Analysis.* IEEE
* Transactions on Intelligent Transportation Systems.
* Gupta, P., & Verma, R. (2019). *Real-Time GPS and GSM-Based Accident Alert System.* Journal of Emerging Technologies in Computing.
* Bose, A. (2018). *Smart Traffic Management Using IoT and AI.* International Journal Transportation Research.

#### Research Papers

1. **S. M. Khan, A. S. A. Hossain, and M. F. Rahman**, “Vehicle accident detection and notification system using a smart phone,” *Journal of Electrical Engineering & Technology*, vol. 12, no. 4, pp. 1573–1581, 2017.  
   View at: [Publisher Site](https://ieeexplore.ieee.org) | [Google Scholar](https://scholar.google.com)
2. **M. Gupta, R. Jha, and A. Mishra**, “Automatic vehicle accident detection and notification system,” *International Journal of Computer Applications*, vol. 108, no. 12, pp. 38–43, 2014.  
   View at: [Publisher Site](https://www.ijcaonline.org) | [Google Scholar](https://scholar.google.com)
3. **A. R. S. Jha, K. K. Garg, and M. S. G. K. Pandey**, “Intelligent vehicle accident detection and alert system using IoT,” *Proceedings of the 5th International Conference on Electronics, Communication and Aerospace Technology*, vol. 2, pp. 931–935, 2019.  
   View at: [Publisher Site](https://ieeexplore.ieee.org) | [Google Scholar](https://scholar.google.com)
4. **S. D. Rao, S. D. Patil, and R. H. Kherde**, “Automatic vehicle collision detection and prevention system using IoT,” *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 5, no. 7, pp. 282–286, 2015.  
   View at: [Publisher Site](https://www.ijarcsse.org) | [Google Scholar](https://scholar.google.com)
5. **B. S. Gupta, S. R. Tiwari, and P. Kumar**, “Development of an intelligent vehicle safety system based on IoT for accident detection and rash driving monitoring,” *Journal of Engineering Science and Technology*, vol. 13, no. 4, pp. 1067–1075, 2018.  
   View at: Publisher Site | [Google Scholar](https://scholar.google.com)
6. **P. Kumar, M. S. J. K. Shah, and R. D. Jain**, “Design of vehicle accident detection and alert system using GSM and GPS,” *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 3, pp. 487–494, 2014.  
   View at: [Publisher Site](https://www.ijcsmc.com) | [Google Scholar](https://scholar.google.com)
7. **V. K. Prakash, S. Shukla, and V. Kumar**, “Rash driving alert and accident detection using wireless sensor networks,” *International Journal of Computer Applications*, vol. 154, no. 2, pp. 1–5, 2016.  
   View at: [Publisher Site](https://www.ijcaonline.org) | [Google Scholar](https://scholar.google.com)
8. **M. B. Syed, A. L. R. V. P. Kumar, and P. P. Reddy**, “Smart vehicle accident detection system and rash driving alert using IoT,” *Proceedings of the 4th International Conference on Emerging Trends in Engineering and Technology*, pp. 22–28, 2019.  
   View at: [Publisher Site](https://ieeexplore.ieee.org) | [Google Scholar](https://scholar.google.com)
9. **T. M. R. Pradeep and S. V. Prasad**, “Vehicle accident detection and prevention system based on IoT,” *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 3, no. 5, pp. 4309–4314, 2015.  
   View at: [Publisher Site](https://www.ijircce.com) | [Google Scholar](https://scholar.google.com)
10. **S. Subramanian and A. S. Nair**, “Real-time vehicle crash detection and automatic alert system using accelerometer and GPS,” *Journal of Electrical Engineering and Technology*, vol. 15, no. 3, pp. 1410–1420, 2020.  
    View at: [Publisher Site](https://ieeexplore.ieee.org) | [Google Scholar](https://scholar.google.com)