**IOT BASED COLLISION DETECTION SYSTEM**

Jyothi Murugan  
StudentVITChennai, India

Cicil Melbin Denny J  
Student VITChennai, India

Mohith S  
StudentVITChennai, India

Manieesh K R  
Student VITChennai, India

Dr.SivaBala Krishnan, Associate Professor, SCOPEVITChennai, India

**ABSTRACT—** The IOT based collision detection system is an innovative solution that employs the use of the SIM 800 C, MPU 6050, and GSM modules to detect collisions in real-time. The system is designed to detect the occurrence of a collision using the MPU 6050 sensor which senses the movement and orientation of the vehicle. The SIM 800 C module is then used to transmit the detected collision data to a cloud-based server via the internet. The GSM module is used for sending SMS alerts to the emergency contact numbers in case of a collision.

The system provides an effective solution for early detection of accidents and ensures prompt response from emergency services. It is also cost-effective, easy to install, and operates in real-time, making it suitable for use in various industries, including transportation, logistics, and personal vehicle safety.

***Keywords:* IoT, MPU6050, GPS, GSM, Health care systems.**

# **introduction**

The implementation of the Internet of Things (IoT) technology has revolutionized the way traffic and healthcare systems operate. In traffic systems, IoT sensors are deployed to collect data on traffic density, road conditions, and environmental factors. The collected data is then analyzed in real-time to provide insights on traffic flow and optimize traffic management. Additionally, IoT sensors can be used to detect accidents and alert emergency services for timely response.

In healthcare systems, IoT technology is used to monitor patient health remotely. IoT devices such as wearables and sensors can collect and transmit patient data to healthcare providers in real-time. This enables early detection of health issues, personalized treatment plans, and reduces the need for hospital visits.

The implementation of IoT technology in traffic and healthcare systems has brought numerous benefits, including reduced congestion, improved safety, and enhanced healthcare delivery. Furthermore, IoT implementation can lead to increased efficiency, cost savings, and improved customer satisfaction. Overall, the application of IoT technology has the potential to revolutionize the way we live, work, and interact with our environment.

The IoT collision detection system is a technological solution designed to detect and alert individuals or organizations of potential collisions in real-time. The system employs various IoT sensors and modules, including SIM 800 C, MPU 6050, and GSM, to detect the occurrence of a collision and transmit the data to a cloud-based server or emergency contacts via SMS. The system is cost-effective, easy to install, and operates in real-time, making it a suitable solution for various industries, including transportation and personal vehicle safety. The IoT collision detection system provides an efficient way of detecting accidents and ensuring prompt response from emergency services, ultimately contributing to the safety of individuals and organizations.

• Automobile accidents are increasing due to human and technological errors.

• The government is working to reduce accidents through initiatives.

• Rescuing injured individuals in remote locations is difficult.

• To address this issue, an IoT system is being developed.

• The system will detect accidents and notify emergency services and contacts.

• The system will use an MPU6500 to identify the accident.

• Authorities will receive an alarm message and use GPS to locate the accident.

• The system aims to improve emergency response times.

• The system can potentially save lives.

• The system can be integrated with existing emergency services.

1. **MOTIVATION AND CONTRIBUTION**

The motivation behind developing an IoT collision detection system was to address the rising number of accidents on roads and the subsequent loss of life and property. The traditional methods of detecting accidents and alerting emergency services have proven to be inadequate, leading to delayed response times and increased fatalities. The development of an IoT collision detection system, therefore, aimed to provide an effective solution that operates in real-time and can alert emergency services promptly.

The IoT collision detection system's contributions are manifold. Firstly, it provides an efficient and cost-effective solution for detecting accidents and alerting emergency services. The system operates in real-time, ensuring that prompt action is taken in case of a collision. Secondly, the system can collect and analyze data on traffic density, road conditions, and environmental factors, providing insights on traffic flow and optimizing traffic management. This feature can help reduce congestion and improve overall road safety. Thirdly, the system can be customized to meet the needs of various industries, including transportation and personal vehicle safety, making it a versatile and scalable solution. Finally, the system can contribute to reducing the number of fatalities and injuries resulting from accidents, ultimately improving the quality of life for individuals and communities.

1. **SCOPE**

Accidents on roads are a leading cause of death and injury worldwide. Rapid response to accidents can save lives and minimize injuries. The IoT-based accident detection system can help in reducing the response time to accidents by automatically detecting and alerting the emergency services. The system can be used in both urban and rural areas and can be integrated with existing emergency services to provide a more efficient response to accidents.

The main reason for this ideology was the 2-wheeler and 4-wheeler accidents been recorded ever day across India. The major cause effect will be on the middle-class people who’re exposed to these kinds of problems in their lifestyle. And, while considering the situation of traffic police and medical service unavailability in remote areas for rescue (i.e., considering hill station pathways, remote area accidents, etc). These were the motivation for this project.

1. **PROJECT OBJECTIVE**

**Issue identified by us: -**

The progress of technology and motor vehicle production has led to a rise in road accidents. The lack of adequate emergency facilities has contributed to a low survival rate after accidents. To address this issue, our proposal involves detecting accidents and identifying their location, which will then be communicated to both the rescue team and emergency contacts of the riders. This concept aims to enhance the response time of emergency services, potentially saving lives in the event of an accident.

**Solution for proposed system: -**

So, in accordance with the problem statement, this project focusses on accident detection (collisions) and informing the rescue team and the passenger's emergency contacts of that specific location. That is, merely detection and informing the appropriate authorities of identification.

**Technologies implemented: -**

For this, GMS Module mounted on the vehicle's bumper will be used, and an MPU6050 will be used to determine the vehicle's orientation. Additionally, an alarm message will be sent to the appropriate authorities so they can use a GPS module to pinpoint the accident's location.

1. **LITERATURE REVIEW**





























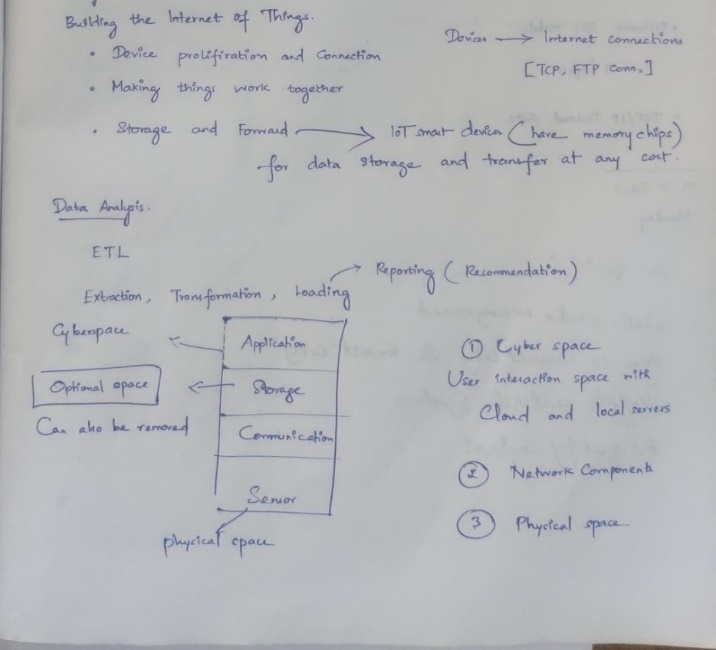


1. **ISSUES IN EXISTING SYSTEM**

Existing accident detection systems have several issues that need to be addressed. Some of the main issues include high false positive rates, unreliable communication, and high cost. The high false positive rates can lead to unnecessary alerts and waste of emergency services' resources. The unreliable communication can delay the response time to accidents, and high cost can limit the system's deployment in low-income areas.

1. **PROPOSED ARCHITECTURE**

A general three-layered architecture for an IoT-based detection system can be structured as follows:



• **Perception layer:** This layer is responsible for acquiring data from various sensors and devices connected to the IoT system. The sensors could be temperature sensors, pressure sensors, motion sensors, cameras, etc. The perception layer processes the sensor data and sends it to the next layer for further processing.

• **Network layer:** This layer is responsible for data transmission and communication between different devices in the IoT system. The network layer could include gateways, routers, switches, and protocols such as MQTT, CoAP, HTTP, etc. The network layer receives data from the perception layer and sends it to the application layer or other devices.

• **Application layer:** This layer is responsible for processing and analyzing the data received from the perception layer and network layer. The application layer could include machine learning algorithms, data analytics tools, and decision-making modules. The application layer generates insights and actionable information that can be used to improve system performance, optimize resources, or trigger alarms and notifications.

Overall flow of data in this architecture would be as follows:

• The MPU6050 Module detects any sudden movements or impacts and send the data to the sensor layer.

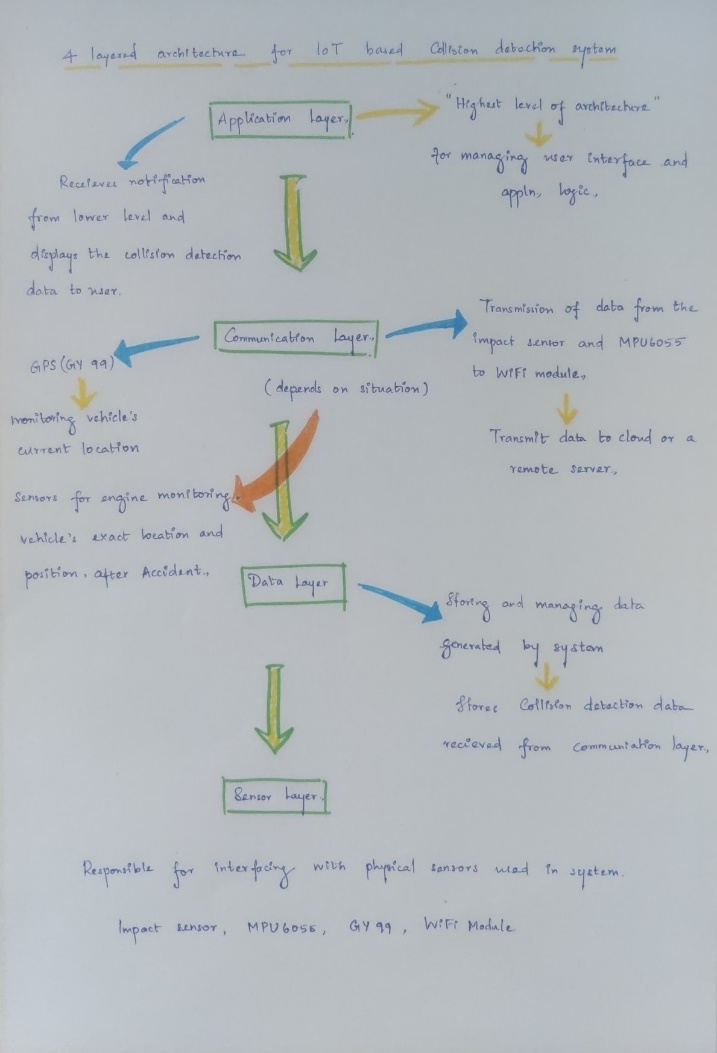
• The sensor layer processes the data and sends it to the communication layer.

• The communication layer transmits the data to the GSM module, which then transmits the data to the cloud or a remote server.

• The data layer stores the accident detection data received from the communication layer.

• The application layer retrieves the data from the data layer and displays it to the user.

Overall, this architecture provides a structured and organized approach to developing an IoT-based collision detection system using MPU6050 and GSM module.



1. **HARDWARE AND SOFTWARE REQUIREMENTS**

The hardware components required for the system include an MPU6050 sensor, microcontroller, GSM Module and power source. The software components required include embedded software for the microcontroller, cloud-based software for the server, and user interface software. The microcontroller used in this system is Arduino, and the cloud-based software is hosted on a web server. The user interface can be accessed through a web browser or a mobile application.

1. **COMPONENTS OF THE COLLISIO DETECTION SYSTEM**

The collision detection system consists of the following components:

• MPU 6050 sensor: This sensor is used to detect the acceleration and rotational movement of the vehicle. It is capable of detecting sudden changes in movement that could indicate a potential collision.

• Arduino UNO: The Arduino UNO is a microcontroller board that acts as the brain of the system. It receives input from the MPU 6050 sensor and processes the data to detect potential collisions.

• GPS module: The GPS module is used to determine the location of the vehicle. It provides real-time location data that can be used to send alerts to emergency services.

• GSM 800A: The GSM 800A is a communication module that is used to send alerts to the driver and emergency services. It can send SMS messages and make calls to predefined numbers.

• Alert sensor: The alert sensor is a device that provides a visual or audible warning to the driver when a potential collision is detected. It can be a buzzer, LED, or any other device that can provide a warning.

1. DESIGN AND IMPLEMENTATION

The collision detection system can be designed and implemented in the following steps:

• Assemble the hardware components: The MPU 6050 sensor, Arduino UNO, GPS module, GSM 800A, and alert sensor need to be assembled together to create the collision detection system.

• Write and upload the code: The code needs to be written and uploaded to the Arduino UNO using the Arduino IDE. The code should include functions to read data from the MPU 6050 sensor, GPS module, and GSM 800A module. It should also include functions to process the data and send alerts when a potential collision is detected.

• Test the system: The system needs to be tested to ensure that it is working correctly. The system should be tested under different conditions to ensure that it is reliable and accurate.

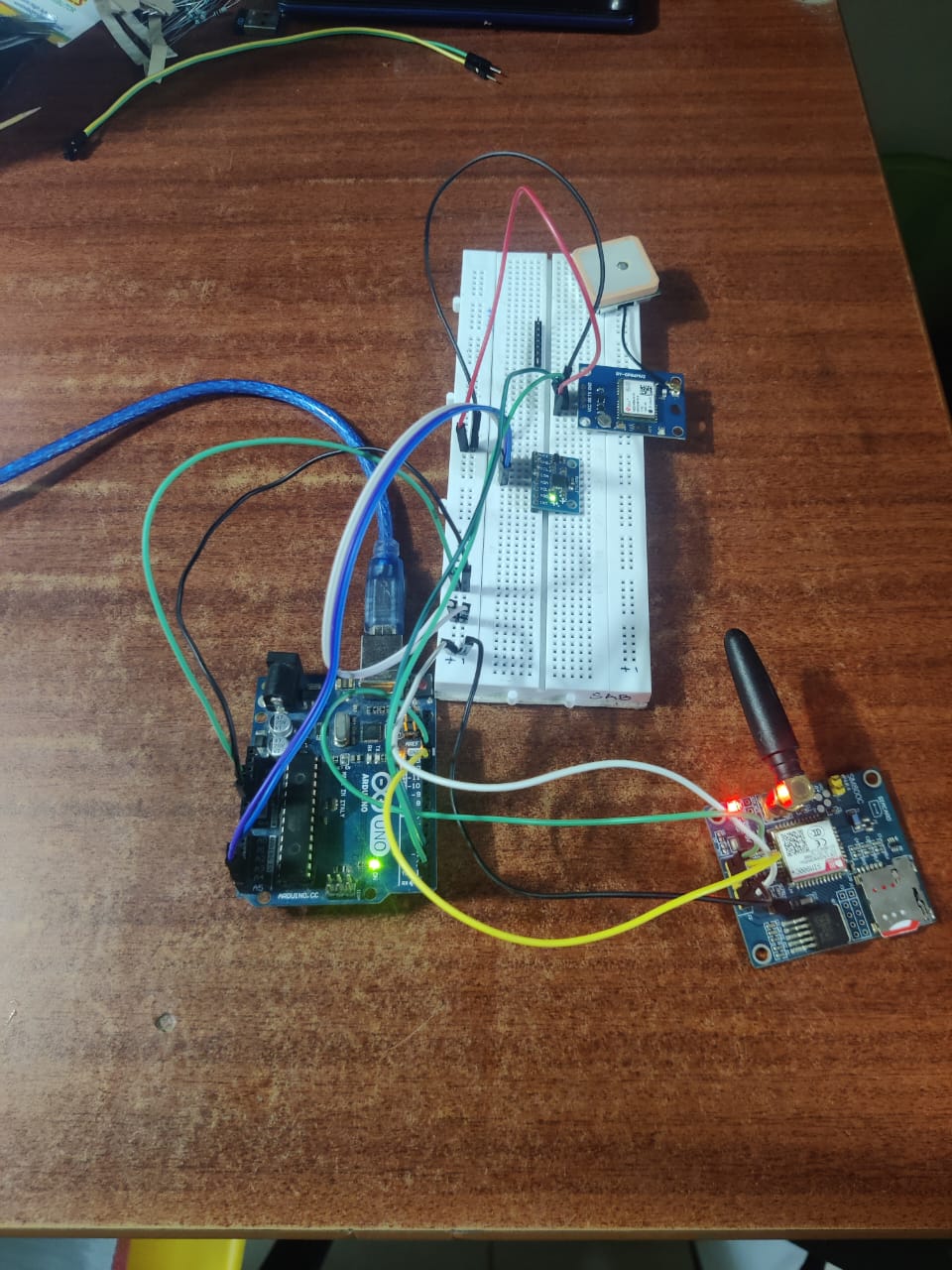
1. **METHODOLOGIES**

The accident detection system uses MPU6050 sensor to detect accidents. This detects sudden changes in acceleration, while the MPU6050 sensor measures the angular velocity and orientation of the vehicle. The sensor data is processed by the embedded software on the microcontroller, which determines whether an accident has occurred. If an accident is detected, the system sends an alert to emergency services using the GSM module. The cloud-based software receives the alert and sends it to the emergency services. The user interface allows users to view and manage alerts and system settings.

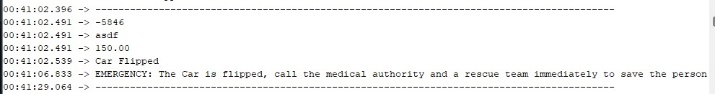
1. **RESULTS AND ANALYSIS**

Results and analysis are critical components of any research or project. It is through the analysis of the results that meaningful insights can be gleaned, and conclusions can be drawn. In the context of IoT-based collision detection using MPU 6050 sensor, Arduino UNO, GPS module, GSM 800A, and alert sensor, results and analysis play a crucial role in determining the effectiveness of the system.

To conduct a proper analysis, the system needs to be tested in different scenarios and conditions. The tests should be designed to mimic real-world situations where collisions are likely to occur. The tests should be conducted on different types of roads and traffic conditions, such as highways, urban roads, and rural roads.



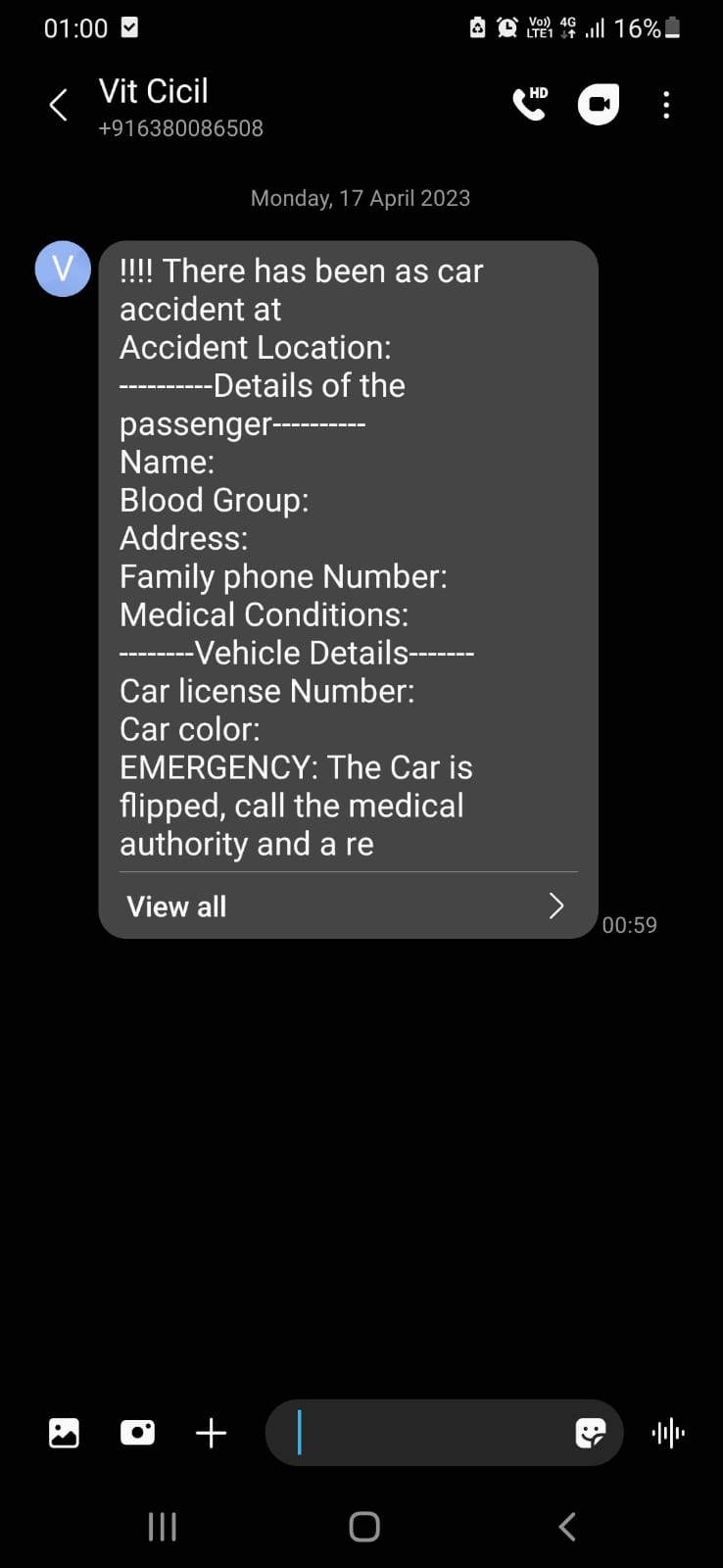
**COMPONENTS CONNECTION**



**CAR FLIPPED**



**CAR NOT FLIPPED**



**MESSAGE INTIMATION**

Once the tests have been conducted, the results need to be analyzed to determine the accuracy and reliability of the system. The following are some of the key metrics that need to be evaluated:

• Detection accuracy: The accuracy of the system in detecting potential collisions needs to be evaluated. This can be done by comparing the system's output with actual collision events that were recorded during the tests.

• Response time: The response time of the system in detecting potential collisions and sending alerts needs to be evaluated. This can be done by measuring the time between the detection of a potential collision and the sending of an alert.

• False positive rate: The false positive rate of the system needs to be evaluated. This can be done by measuring the number of times the system generates alerts when no potential collision is detected.

• Robustness: The robustness of the system needs to be evaluated by subjecting it to different environmental conditions, such as extreme temperatures, humidity, and vibrations.

• Usability: The usability of the system needs to be evaluated by testing it with different users and assessing their feedback on the system's ease of use and effectiveness.

Once the metrics have been evaluated, the results need to be analyzed to determine the system's effectiveness. Our analysis includes a comparison of the system's performance against established benchmarks and standards. This analysis also includes an evaluation of the system's strengths and weaknesses and recommendations for future improvements.

1. **MONETIZATION**

**PRODUCTION PLAN:**

In the Monetization section of the production plan for the IoT collision detection system, the following strategies can be adopted:

Direct Sales: The IoT collision detection system can be sold directly to customers or businesses that require the technology. This can be done through online marketplaces, trade shows, or through direct sales teams. This strategy allows for a quick return on investment.

Subscription-based Model: A subscription-based model can be adopted, where customers pay a monthly or annual fee for access to the system. This can be appealing to businesses that want to avoid large upfront costs and prefer a more predictable and manageable expense structure.

Licensing: The IoT collision detection system can be licensed to third-party companies, allowing them to use the technology under certain terms and conditions. This can be a good option for companies that already have an established customer base or distribution network.

Value-Added Services: Value-added services such as maintenance, support, and upgrades can be offered to customers as an additional revenue stream. This strategy can help build customer loyalty and increase the lifetime value of customers.

Overall, the IoT collision detection system has significant potential for monetization. The versatility and scalability of the technology make it appealing to various industries, and the adoption of multiple monetization strategies can ensure a steady revenue stream.

**II) TARGET AUDIENCE:**

The targeted audience for the IoT collision detection system includes individuals, businesses, and organizations that operate vehicles, including cars, trucks, buses, and other transportation means. This can include personal vehicle owners, fleet managers, transportation and logistics companies, and emergency services providers.

**III) MARKETING PLAN:**

The marketing plan for the IoT collision detection system should focus on educating potential customers about the technology's benefits and how it can improve their safety and efficiency. Some effective marketing strategies that can be adopted include:

Content Marketing: Creating high-quality content that educates potential customers about the technology's features and benefits. This can include blog posts, infographics, videos, and case studies.

Social Media Marketing: Promoting the technology on social media platforms such as LinkedIn, Twitter, and Facebook. This can include paid advertising, sponsored content, and influencer marketing.

Search Engine Optimization (SEO): Optimizing the website and other online content to rank higher in search engine results pages. This can help potential customers find the technology when searching for related keywords.

Direct Mail Marketing: Sending promotional material such as brochures, flyers, and other printed materials directly to potential customers.

Trade Shows and Events: Participating in industry-related trade shows and events to showcase the technology and network with potential customers and partners.

Referral Marketing: Encouraging existing customers to refer their friends, colleagues, and associates to the technology by offering incentives or discounts.

By adopting a targeted and integrated marketing approach, the IoT collision detection system can reach a broader audience and generate significant interest and demand from potential customers.

1. **CONCLUSION**

The IoT-based accident detection system can help in reducing the response time to accidents by automatically detecting and alerting the emergency services. The system uses MPU6050 sensor and GSM module to detect and transmit information about the accident to the cloud. The proposed layered architecture allows for modular development and easy integration of different components. The system can be used in both urban and rural areas and can be integrated with existing emergency services to provide a more efficient.

1. **FUTURE WORK**

Future work will be focusing on implementing this game play through another VR Controller and making it platform independent (VR Controller). And also, implementing a greater number of puzzles and levels with increasing difficulty levels and making the game more interesting and effective one.

1. **REFERENCES**

<https://www.bing.com/ck/a?!&&p=17dd108067f9f5f9JmltdHM9MTY3MzgyNzIwMCZpZ3VpZD0wNzBiNDc0MS01ZTRhLTY5MmQtMWIwOC01NjcxNWZlNzY4Y2EmaW5zaWQ9NTE5MQ&ptn=3&hsh=3&fclid=070b4741-5e4a-692d-1b0856715fe768ca&psq=accident+detection+app&u=a1aHR0cHM6Ly9uZXZvbnByb2plY3RzLmNvbS9hY2NpZGVudC1kZXRlY3Rpb24tYWxlcnQtYW5kcm9pZC1hcHAv&ntb=1>

<https://www.bing.com/ck/a?!&&p=a80cd0ed9539b7e4JmltdHM9MTY3MzgyNzIwMCZpZ3VpZD0wNzBiNDc0MS01ZTRhLTY5MmQtMWIwOC01NjcxNWZlNzY4Y2EmaW5zaWQ9NTIwOQ&ptn=3&hsh=3&fclid=070b4741-5e4a-692d-1b08-56715fe768ca&psq=accident+detection+app&u=a1aHR0cHM6Ly9kZXZmb2xpby5jby9wcm9qZWN0cy9hY2NpZGVudC1kZXRlY3Rpb24tYXBw&ntb=1>

<https://www.bing.com/ck/a?!&&p=fd54eb426a950633JmltdHM9MTY3MzgyNzIwMCZpZ3VpZD0wNzBiNDc0MS01ZTRhLTY5MmQtMWIwOC01NjcxNWZlNzY4Y2EmaW5zaWQ9NTE5Mw&ptn=3&hsh=3&fclid=070b4741-5e4a-692d-1b08-56715fe768ca&psq=mpu+6500+arduino&u=a1aHR0cDovL2FyZHVpbm9sZWFybmluZy5jb20vY29kZS9hcmR1aW5vLW1wdTY1MDAtNi1heGlzLW1vdGlvbi10cmFja2luZy1kZXZpY2UucGhw&ntb=1>

**APPENDIX:**

**Flip.ino**

#include <SoftwareSerial.h>

#include<Wire.h>

int Rx\_pin = 9;

int Tx\_pin = 8;

SoftwareSerial SerialGPS(Rx\_pin, Tx\_pin);

SoftwareSerial GSM(10, 11);

#define BUZ 7

#define LED 3

int button = 2;

double roll;

#include <math.h>

int minVal = 265;

int maxVal = 402;

double x;

double y;

double z;

const int MPU\_addr = 0x68;

int16\_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;

const int x\_out = A1; /\* connect x\_out of module to A1 of UNO board \*/

const int y\_out = A2; /\* connect y\_out of module to A2 of UNO board \*/

const int z\_out = A3; /\* connect z\_out of module to A3 of UNO board \*/

void setup() {

pinMode(button, INPUT); //Set the sensor pin as INPUT

Serial.begin(9600);//Set baud rate for serial communication

SerialGPS.begin(9600);

GSM.begin(9600); // Setting the baud rate of GSM Module

delay(100);

pinMode(BUZ, OUTPUT);

Wire.begin();

Wire.beginTransmission(MPU\_addr);

Wire.write(0x6B);// PWR\_MGMT\_1 register

Wire.write(0); // set to zero (wakes up the MPU-6050)

Wire.endTransmission(true); Wire.begin();

Serial.begin(9600);

delay(1000);

}

void prntmpu()

{

Serial.print(AcX);

Serial.print(" ");

Serial.print(AcY);

Serial.print(" ");

Serial.print(AcZ);

Serial.print(" ");

Serial.print(Tmp);

Serial.print(" ");

Serial.print(GyX);

Serial.print(" ");

Serial.print(GyY);

Serial.print(" ");

Serial.print(GyZ);

Serial.println(" ");

Serial.println("--------------------------------------------------");

Serial.println(AcX+ AcY+ AcZ+ Tmp+ GyX+ GyY+ GyZ);

Serial.println("-----------------------------------------------------");

}

void roll\_check()

{

int x = map(AcZ, minVal, maxVal, -90, 90);

Serial.println(x);

if ( AcZ<-1000)

{

Serial.println("asdf");

roll = 150;

}

else

roll = 210;

}

void loop(){

Serial.println("------------------------------------------------------------------------------------------");

GetMpuValue1(MPU\_addr);

// prntmpu();

roll\_check();

//

// if (digitalRead(button) == LOW){

// Serial.println("Collision");//print collision

// delay(300);

//// SendMessage();

//// buzled();

// delay(500);

//// exit(0);

// }

// axc();

Serial.println(roll);

if ((roll < 180)||(roll>240)){

Serial.println("Car Flipped");//print collision

delay(300);

SendMessage();

buzled();

delay(500);

// exit(0);

}

else{

Serial.println("Car isnt Flipped");//print collision

}

delay(1000);

}

void buzled(){

int i=0;

while (i<2) {

i+=1;

delay(1000);

digitalWrite(LED, HIGH);

digitalWrite(BUZ, HIGH);

delay(1000);

digitalWrite(BUZ, LOW);

digitalWrite(LED, LOW);}

}

void SendMessage(){

//GSM.println("ATD8825413527");

//delay(300);

GSM.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode

delay(300);

GSM.println("AT+CMGS=\"8248469940\"\r");

delay(300);

GSM.println("!!!! There has been as car accident at");// The SMS text you want to send

delay(300);

gps();

delay(300);

GSM.println("----------Details of the passenger----------");

delay(300);

GSM.println("Name: ");

delay(300);

GSM.println("Blood Group: ");

delay(300);

GSM.println("Address: ");

delay(300);

GSM.println("Family phone Number: ");

delay(300);

GSM.println("Medical Conditions: ");

delay(300);

GSM.println("--------Vehicle Details-------");

delay(300);

GSM.println("Car license Number:");

delay(300);

GSM.println("Car color:");

if ((roll < 180)||(roll>240)) {

GSM.println("EMERGENCY: The Car is flipped, call the medical authority and a rescue team immediately to save the person from the car");

Serial.println("EMERGENCY: The Car is flipped, call the medical authority and a rescue team immediately to save the person from the car");}

else if((roll > 180)||(roll<240)) {

GSM.println("The Car isnt flipped");

Serial.println("The Car isnt flipped");}

delay(300);

GSM.println("!!!!Kindly inform the authorized person to take further actions to save a life!!!! ");

delay(300);

GSM.println((char)26);// ASCII code of CTRL+Z

}

void GetMpuValue1(const int MPU) {

Wire.beginTransmission(MPU);

Wire.write(0x3B); // starting with register 0x3B (ACCEL\_XOUT\_H)

Wire.endTransmission(false);

Wire.requestFrom(MPU, 14, true); // request a total of 14 registers

AcX = Wire.read() << 8 | Wire.read(); // 0x3B (ACCEL\_XOUT\_H) & 0x3C (ACCEL\_XOUT\_L)

AcY = Wire.read() << 8 | Wire.read(); // 0x3D (ACCEL\_YOUT\_H) & 0x3E (ACCEL\_YOUT\_L)

AcZ = Wire.read() << 8 | Wire.read(); // 0x3F (ACCEL\_ZOUT\_H) & 0x40 (ACCEL\_ZOUT\_L)

Tmp = Wire.read() << 8 | Wire.read(); // 0x41 (TEMP\_OUT\_H) & 0x42 (TEMP\_OUT\_L)

int xAng = map(AcX, minVal, maxVal, -90, 90);

int yAng = map(AcY, minVal, maxVal, -90, 90);

int zAng = map(AcZ, minVal, maxVal, -90, 90);

GyX = Wire.read() << 8 | Wire.read(); // 0x43 (GYRO\_XOUT\_H) & 0x44 (GYRO\_XOUT\_L)

GyY = Wire.read() << 8 | Wire.read(); // 0x45 (GYRO\_YOUT\_H) & 0x46 (GYRO\_YOUT\_L)

GyZ = Wire.read() << 8 | Wire.read(); // 0x47 (GYRO\_ZOUT\_H) & 0x48 (GYRO\_ZOUT\_L)

x = RAD\_TO\_DEG \* (atan2(-yAng, -zAng) + PI) + 4; //offset by 4 degrees to get back to zero

y = RAD\_TO\_DEG \* (atan2(-xAng, -zAng) + PI);

z = RAD\_TO\_DEG \* (atan2(-yAng, -xAng) + PI);

}

void gps(){

GSM.println("Accident Location: 10.9027N, 76.9006E");

while(false){

while (SerialGPS.available() > 0)

Serial.write(SerialGPS.read());

}

}