IOT - BASED SUICIDE PREVENTION SYSTEM (FOR HOSTELS)

VULLENGALA RAJ KUMAR
Dept. of Computer Science & Engineering (IOT), Siddhartha Institute of Technology and Sciences)
21tq1a6907@siddhartha.co.in

MOUNIKA
Assistant Professor
(Dept. of Computer Science & Engineering, Siddhartha Institute of Technology and Sciences)
dudalamounika.cse@siddhartha.co.in

NARLAPURAM SRAVAN KUMAR
(Dept. of Computer Science & Engineering (IOT), Siddhartha Institute of Technology and Sciences)
21tq1a6905@siddhartha.co.in

DAMERA SHIVA PRASAD
(Dept. of Computer Science & Engineering (IOT), Siddhartha Institute of Technology and Sciences)
21tq1a6916@siddhartha.co.in

YAYA AVANTHI
(Dept. of Computer Science & Engineering (IOT), Siddhartha Institute of Technology and Sciences)
21tq1a6908@siddhartha.co.in

Abstract—**The project addresses a significant issue, particularly among students in hostels, where mental health struggles often lead to tragic incidents. This project proposes an IoT-based system designed to prevent suicides by enabling early detection and timely intervention. The system employs IR sensors installed near ceiling fans to identify objects such as ropes or cloth that may indicate a suicide attempt. These sensors are connected to an Arduino microcontroller, which processes the data and triggers an alert mechanism when a potential attempt is detected.**

The alert system includes LED indicators, buzzers, and real-time notifications sent via a Wi-Fi module (ESP32) to a central monitoring dashboard. The dashboard visually highlights the affected room, enabling hostel wardens or emergency responders to act promptly. This automated process ensures quick response times while minimizing human error. The system's sensitivity is calibrated to enhance accuracy and reduce false positives, making it reliable and efficient in various settings.

Testing demonstrated that the system could accurately detect objects and trigger immediate alerts, proving its effectiveness. The audible alarms and visual indicators worked seamlessly, while the dashboard provided a user-friendly interface to monitor multiple rooms simultaneously. By leveraging IoT technology, this system offers a scalable and practical solution to mitigate suicide risks, especially in high-risk environments such as hostels.

This project underscores the potential of IoT to address real-world mental health challenges. By enabling early detection and rapid response, the system provides a proactive tool to save lives and foster safer living spaces. With further development, it can be widely adopted to prevent suicides and support mental health awareness in various high-risk environments.

Keywords- IoT Technology, Sensor Integration, PIR Sensor, Ultrasonic Sensor, ESP32 Microcontroller, GSM Module, Real-Time Alerts, Wireless Communication, Cloud Data Storage, Safety Monitoring, Motion Detection, Object Detection, Emergency Response System, Automated Alert Mechanism, Proactive Detection, Environmental Monitoring, Remote Monitoring, Data Logging, Internet of Things Applications, Security System, Intelligent Safety Solutions, Real-Time Notification, Human Presence Detection, IoT-Based Safety System.

INTRODUCTION

This project addresses the increasing issue of self-harm incidents, particularly those involving ceiling fans in residential and institutional settings. Current safety systems lack the capacity for early detection and real-time alerts in such situations, prompting the development of an innovative, IoT-enabled solution. The aim is to design a compact device that can be mounted on ceiling fans to monitor activities and generate alerts in potentially harmful scenarios.

Driven by the urgent need to prevent suicide attempts in hostel environments, this project leverages the Internet of Things (IoT) to create an early detection system aimed at saving lives, reducing emotional distress for families, and enhancing safety for students. Beyond its technological advancements, this initiative also advocates for mental health awareness and proactive support systems, emphasizing the importance of early intervention and prevention. The ultimate goal is to foster a supportive environment where potential self-harm incidents are promptly identified and addressed, creating a safer and more compassionate space for individuals at risk. By integrating technology with mental health support, this project aims to provide holistic solutions to these pressing issues.

LITERATURE SURVEY

Introduction The purpose of this literature survey is to review existing research, technologies, and solutions related to IoT-based safety systems and suicide prevention mechanisms. This survey provides context for the IoT-Based Suicide Prevention System, justifies the technological approach adopted, and identifies gaps in current solutions that the project aims to address.

Academic Research and Studies-

IoT in Safety Monitoring

Studies by Ashton (2009) and Gubbi et al. (2013) highlight

IoT's role in enabling real-time data collection and remote monitoring, crucial for timely intervention in safety applications.

Sensor Technology for Detection

Research by Li et al. (2015) emphasizes the use of PIR, ultrasonic, and load sensors for accurate detection and reduced false alarms in safety systems.

Microcontrollers and IoT Communication

The ESP32 microcontroller is favored for its low power consumption and built-in Wi-Fi, as shown by Espressif Systems (2016), while GSM modules are recommended for SMS alerts in limited connectivity areas.

Real-Time Monitoring and Cloud Integration

Studies by Da Xu et al. (2014) demonstrate the benefits of cloud platforms like Firebase for real-time data logging and analysis in IoT systems.

Gaps in Current Solutions-

Existing suicide prevention systems often rely on manual monitoring or basic automated systems, which lack real-time alert mechanisms. This delay in response can lead to missed opportunities for timely intervention, increasing the risk of harm. IoT-based systems with immediate alert capabilities can address this gap by ensuring quicker reaction times.

Many traditional safety systems use single-sensor setups, which can lead to high false positive rates. For instance, motion detected by a single PIR sensor might not always indicate a genuine threat. Integrating multiple sensors, such as PIR, ultrasonic, and load sensors, can significantly improve accuracy by providing corroborative data.

Many existing systems do not provide continuous, real-time monitoring, which is essential for detecting and addressing potential self-harm incidents promptly. IoT-enabled systems can offer continuous data collection and real-time analysis, enhancing overall responsiveness and effectiveness.

Relevance to IoT-Based Suicide Prevention System

The IoT-based Suicide Prevention System adopts best practices identified in research on safety, user-centred design, and real-time monitoring. The system is designed to prioritize the user's well-being, ensuring that the technology is non-intrusive while offering timely alerts. Regular testing and improvements based on user feedback ensure that the system is responsive and accessible.

Addressing Gaps The system addresses gaps in current safety solutions by providing proactive monitoring through an intelligent sensor placed on ceiling fans to detect potential hanging scenarios. By integrating IoT technology, it offers real-time notifications to caretakers and emergency responders with accuracy.

Innovative Features The IoT-based Suicide Prevention System introduces innovative features, such as real-time sensor data analytics, automated notifications to caretakers and authorities, and integration with personal devices for immediate response.

Conclusion

The literature survey highlights the importance of integrating IoT, sensor technologies, and real-time communication to enhance safety systems. The IoT-Based Suicide Prevention System leverages these insights to develop a robust, real-time monitoring solution aimed at preventing suicide attempts. This survey underscores the innovative approach taken by the project and its potential to address critical gaps in current safety monitoring systems.

PROBLEM FORMULATION

PROBLEM DEFINITION

The increasing incidence of self-harm in residential and institutional settings, particularly involving ceiling fans, demands innovative safety solutions. Suicide is a significant issue, especially among students living in hostels, where mental health struggles often result in tragic incidents.

PROJECT AIM

The aim of the Iot Based Suicide prevention System is to create a compact, IoT-enabled device that can be mounted on ceiling fans to monitor activities and generate alerts in potentially harmful scenarios. By doing so, it strives to save lives, reduce the emotional distress experienced by families

SCOPE OF PROJECT

This project ultimate goal is to foster a supportive environment where potential self-harm incidents are quickly identified and addressed. thus, providing a safer and more compassionate space for individuals at risk & prevent suicide attempts in hostel environments, this project leverages the Internet of Things (IoT) to establish an early detection system

MOTIVATION

In recent years, the rising number of suicides by hanging in confined spaces such as hostels, dormitories, and hospitals has become a major concern. The Iot based suicide system is designed to detect the presence of individuals near a ceiling fan, measure distances to objects below the fan, and monitor moments changes near the fan rod. If abnormal conditions are detected, alerts are sent in real-time via alert messages such as emergency alarm to alert the concerned authorities .

EXISTING SYSTEM

The increasing incidence of self-harm in residential and institutional settings, particularly involving ceiling fans, demands innovative safety solutions. . By focusing on early detection and leveraging IoT technology, the goal is to save lives, reduce emotional distress for families, and create safer environments for students. Additionally, this initiative supports mental health awareness and proactive support systems

LACK OF EXISTING SYSTEM

1. Delayed Response: Existing suicide prevention systems often rely on manual monitoring or basic automated systems, which lack real-time alert mechanisms. This delay in response can lead to missed opportunities for timely intervention, increasing the risk of harm.
2. No Immediate Notification to Caretakers: In many existing solutions, notifications are delayed or sent to the wrong people, meaning that immediate intervention from caretakers or emergency services is often not possible.
3. Lack of Real-Time Monitoring: Many existing systems do not provide continuous, real-time monitoring, which is essential for detecting and addressing potential self-harm incidents promptly. IoT-enabled systems can offer continuous data collection and real-time analysis, enhancing overall responsiveness and effectiveness.

4. No Automated Monitoring: Traditional systems may require human intervention to detect danger signs, which can delay response times and reduce the overall effectiveness.
5. Failure to Adapt Over Time: Traditional systems do not adapt or improve based on evolving data, which means they may miss important trends or signs over time.

By addressing these gaps, the IoT-Based Suicide Prevention System offers a comprehensive solution that proactively monitors environmental conditions, detects potential threats in real time, and sends immediate alerts to caregivers and emergency responders. This innovative system enhances safety by integrating multiple sensor technologies and advanced communication modules, ensuring timely intervention and preventing tragedies. It sets a new standard in suicide prevention, delivering a seamless and life-saving experience for at-risk individuals and their support network.

PROPOSED SYSTEM

Key Features of the Proposed System

1. Real-Time Monitoring:
   - The system continuously monitors the environment using ultrasonic and PIR sensors to detect unusual activity, such as a student attempting to commit suicide using a ceiling fan "Real-Time Monitoring".
2. Real-Time Alerts and Immediate Response:
   - Buzzer and LED Indicators Provides instant visual and auditory alerts when abnormal conditions are detected.
3. Cloud-Based Monitoring and Data Visualization:
   - ThingSpeak Integration: Sensor data is sent to the cloud, providing real-time updates, visualizations, and historical insights through graphs and charts.
   - Dashboard-Monitoring: Authorities can access the dashboard for continuous monitoring of sensor statuses and immediate actions.
4. Enhanced Data Accuracy and Reliability:
   - Sensor Calibration & precise measurements to minimize false positives and improve detection accuracy.
   - Secure and Reliable Communication.
5. Comprehensive Safety and Security:
   - All data is processed through a secure microcontroller and transmitted to an HTTP server for safe storage and access.
   - Engaging Content: Includes quizzes, videos, and interactive elements to reinforce key concepts.
6. Cloud-Based Insights
   - Historical data analysis allows authorities to track patterns and trends in behaviour over time, improving the system's effectiveness.
   - Immediate Intervention Alerts and data are designed to ensure the fastest possible response from concerned authorities.

HARDWARE REQUIREMENTS

1. ESP32 Microcontroller:
   a. Model: ESP32 Dev Kit V1 (or any compatible ESP32 development board).
   b. Processor: Dual-core processor (ESP32, clock speed up to 240 MHz).
   c. Memory: 520 KB SRAM and 4MB Flash storage.
   d. Connectivity: Built-in Wi-Fi and Bluetooth capabilities for cloud communication and device control.
2. PIR Sensor:
   a. Model: HC-SR501 or compatible PIR motion sensor.
   b. Voltage: 5V DC.
   c. Output: Digital signal (HIGH when motion is detected, LOW otherwise).
   d. Field of View: Typically, 120°.
3. Ultrasonic Sensor:
   a. Model: HC-SR04 or compatible ultrasonic distance sensor.
   b. Voltage: 5V DC.
   c. Operating Range: 2cm to 400cm.
   d. Output: Sends and receives ultrasonic pulses to measure the distance between the sensor and nearby objects.
4. LED (Light Emitting Diode):
   a. Color: Any (Typically Red or Green for status indication).
   b. Voltage: 3.3V or 5V (depending on the power supply).
   c. Current: Typically, 20mA for standard LEDs.
   d. Role: Visual indicator for system status or alerts.
5. Buzzer:
   a. Model: Active or passive buzzer.
   b. Voltage: 5V DC.
   c. Current: 15-30mA.
   d. Role: Provides an audible sound for immediate alert when abnormal conditions are detected.
6. Jumper Wires:
   a. Type: Male-to-male, female-to-male, or male-to-female jumper wires.
   b. Length: 10cm to 20cm for flexible connections.
   c. Purpose: Used to connect various components (sensors, ESP32, LED, buzzer) on a breadboard.
7. Breadboard:
   a. Size: Standard size (830 tie-points recommended).
   b. Purpose: Used for prototyping and testing the circuit without soldering.
   c. Power Supply: Can be powered via the breadboard’s built-in power rails (5V and GND).

SOFTWARE REQUIREMENTS

1. Arduino IDE:

- Version: 1.8.19 or later.
- Supported Platforms: Windows, macOS, and Linux.
- Description: The Arduino IDE is used to write, compile, and upload code to the ESP32 microcontroller. It supports C++ programming and

is compatible with various libraries for sensors and cloud integration.

2. ESP32 Board Support:
   - Installation: Required to be added to the Arduino IDE's board manager.
   - Description: The ESP32 Board Support Package (BSP) is necessary to recognize and communicate with the ESP32 microcontroller within the Arduino IDE.
3. ThingSpeak Cloud Platform:
   - Account: A free ThingSpeak account (https://thingspeak.com) is required to store and visualize IoT sensor data.
   - Supported Browsers:
     - Chrome: Latest version.
     - Firefox: Latest version.
   - Description: ThingSpeak is a cloud-based platform used for real-time data logging, visualization, and storage.
4. Required Libraries:
   - ESP32 WiFi Library: Used to establish Wi-Fi connectivity for the ESP32.
   - ThingSpeak Library: Used to send and retrieve data from the ThingSpeak platform.
   - Ultrasonic Library: Used to interface with the ultrasonic distance sensor.
   - PIR Sensor Library: Provides functions to handle the PIR sensor for motion detection.
   - Description: Libraries are used to simplify coding for sensors, communication, and cloud interaction.

SYSTEM DESIGN

SOFTWARE DESIGN

The system design for the IoT-Based Suicide Prevention System focuses on creating a reliable, scalable, and efficient platform for real-time monitoring and alerting. This design integrates multiple hardware components and IoT technologies to ensure accurate detection and timely response. Here's a detailed overview of the system design:

1. Sensor Integration Design
   - PIR Sensors: Detect motion near the ceiling fan, triggering alerts when unusual activity is observed.
   - Ultrasonic Sensors: Measure the distance to objects below the fan to detect items like ropes or cloth.
2. Microcontroller and Data Processing
   - ESP32 Microcontroller: Acts as the central processing unit, collecting and analysing data from all sensors.
   - Data Processing: The ESP32 processes the input data to determine if the conditions meet the criteria for a potential self-harm attempt.
   - Decision-Making: Based on the processed data, the ESP32 triggers alert mechanisms.
3. Communication System Design
   - Wi-Fi Module: Integrated within the ESP32, used to send real-time data to cloud platforms for remote monitoring
   - ThingSpeak Cloud Integration:
     - Data Transmission: The ESP32 sends real-time data to ThingSpeak, a cloud-based platform for IoT data logging and visualization.
     - Graph Visualization: ThingSpeak visualizes sensor data through graphs, allowing easy monitoring of activity trends and immediate detection of anomalies.
   - Local Alerts:
     - LED Indicators: Provide a visual alert when the system detects unusual activity.
     - Buzzer: Emits an audible alert to notify nearby individuals of potential danger.

4. Payment Processing:
   - Payment Gateway Integration: Integrate with secure payment gateways like Paytm or PayPal to handle transactions.
   - Multiple Payment Options: Design the interface to support various payment methods, ensuring a smooth and secure transaction process.

5. User Interaction and Dashboard Design:

ThingSpeak Dashboard:
- Real-Time Data Monitoring: Displays sensor data in real-time, helping users track environmental changes.
- Historical Data Analysis: Provides a record of past activity, enabling pattern recognition and better understanding of high-risk times or areas.
- Cloud Alerts: Based on data trends, ThingSpeak can be configured to send alerts or trigger actions when specific conditions are met, ensuring remote monitoring is effective.

ALERT SYSTEM MECHANISM :

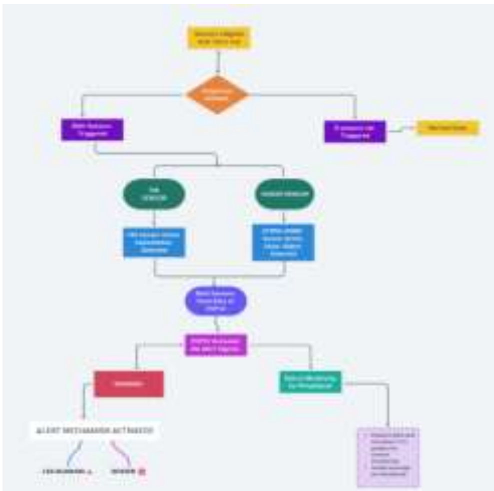

FIG. 1. ALERT MECHANISM AND CONTROL FLOW

Alert System Mechanism Explanation:

1. Sensor Alignment and Monitoring:
   - The system starts with sensors strategically placed near the fan's rod. These sensors continuously monitor the environment for any suspicious activities.
2. Suspicious Moment Detection:
   - Normal State: If no abnormal activity is detected, the system remains in the normal state without triggering any alerts.
   - Suspicious State: When both the PIR (Passive Infrared) and Ultrasonic sensors detect unusual motion or the presence of an object near the fan's rod, the system transitions into the suspicious state.
3. Sensor Activation:
   - PIR Sensor: This sensor is responsible for detecting motion. When it senses movement, it becomes active and sends data indicating motion has been detected.
   - Ultrasonic Sensor: This sensor detects the presence of objects. When it identifies an object, it sends data indicating an object is present near the fan's rod.
4. Data Transmission to ESP32:
   - Both sensors send their respective data to the ESP32 microcontroller. The ESP32 serves as the central processing unit, receiving and analysing the sensor inputs.
5. Activation of Alert Signals:
   - Once the ESP32 processes the data and confirms the detection of a suspicious event, it triggers the alert mechanism:
   - Warning: A warning signal is generated to indicate that the system has detected a potentially dangerous situation.
   - LED Blinking: The LED starts blinking to provide a visual alert to anyone nearby.
   - Buzzer: The buzzer sounds to provide an audible alert, ensuring the alert is noticed even if no one is watching the LED.
6. Data Monitoring and Visualization via ThingSpeak:
   - The ESP32 sends the data to the ThingSpeak platform, which is used for remote monitoring and visualization.
   - ThingSpeak: This cloud-based platform collects the data, visualizes it in real-time graphs, and provides an online dashboard for monitoring.
   - Notifications: The system sends alerts and messages to designated individuals, allowing for quick response to prevent a potential suicide attempt.

Caretaker Dashboard:

1. Home Page
   - Purpose: Entry point for caretakers to log in or register.
     Key Features:
   - A welcoming message about the system's importance and functionality.
   - Options for Login and Register.
2. Register or Login Page
   - Register Page: Allows caretakers to create an account by entering credentials (e.g., email, password) and initial profile details.
   - Login Page: If an account exists, the caretaker logs in with their credentials.
   - On successful login, they are redirected directly to the Caretaker Dashboard, bypassing the setup phase.
3. Caretaker's Dashboard
   - Purpose: The main panel for managing and monitoring the IoT system.
     Sections and Features:
   - Real-time sensor status and graph visualization from ThingSpeak.
   - Alert status: "Active" or "Normal."
     Monitor
   - Live data from sensors.
   - Highlighted graph spikes when alerts trigger.
     Manage
   - Reset alerts with a single button (deactivates LED and buzzer).
   - Adjustable sensor thresholds for trigger sensitivity.
     Resources
   - Guides for troubleshooting and replacing components.
   - Support contact or issue reporting form.
4. Control Flow
   - Caretaker registers/logins on Home Page.
   - New users configure their profiles, while returning users access the Dashboard.
   - The caretaker views real-time sensor data in Monitor and addresses alerts in Manage.
   - Resources provide access to technical support and guides

UML DIAGRAMS

UML (Unified Modelling Language) is a standardized modelling language used in object-oriented software engineering. It helps visualize, design, and document software systems effectively. UML diagrams are especially helpful in organizing project requirements and illustrating the relationships between different system components.

The UML standard is managed by the Object Management Group (OMG) and aims to provide a common language for designing object-oriented systems. UML consists of a meta-model and notations, allowing developers and stakeholders to communicate ideas effectively. Over time, UML has evolved to support software, business, and even non-software system modelling.

UML played a pivotal role in designing, visualizing, and managing the interactions within the IoT-Based Suicide Prevention System. The system connects sensors to cloud data storage and enables the caretaker's dashboard for real-time monitoring.

TYPES OF UML DIAGRAM

By leveraging UML diagrams, we were able to effectively visualize the flow of information and interactions across different system components. Here's how UML facilitated various aspects of the project:

A. USE CASE DIAGRAM

The use case diagram provides an overview of system functionality from the Caretaker's perspective.

Actors:

- *Caretaker*
- *Sensors*
- *ThingSpeak Cloud*

Use Cases:

- Caretaker: Can monitor real-time data, view alerts, and reset the system when necessary
- Sensors: Trigger alerts based on specific conditions (e.g., motion, vital signs).
- ThingSpeak Cloud: Receives data from sensors, processes it, and displays the alert spike on the dashboard.

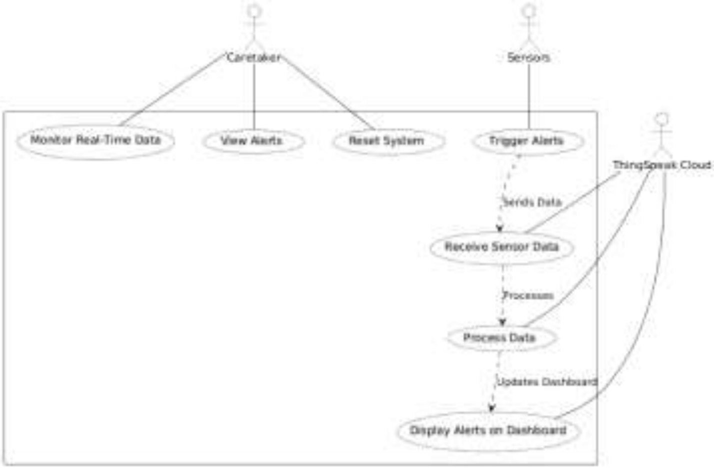

FIG. 2. USE CASE DIAGRAM

B. CLASS DIAGRAM

A Class Diagram defines the structure of the system and its classes, including attributes and methods.

Classes:

Caretaker:

- Attributes: username, password, reset_count, threshold_preferences
- Methods: login(), resetAlert(), monitorData()

Sensor:

- Attributes: sensorID, sensorType, status
- Methods: readData(), triggerAlert()

CloudService (ThingSpeak API):

- Attributes: apiKey, channelID, dataStorage
- Methods: uploadData(), retrieveData()

AlertSystem:

- Attributes: ledStatus, buzzerStatus
- Methods: activateAlert(), deactivateAlert()

Guided System Design:

The class diagram helped design the structure of our system, defining important objects like Caretaker, Sensors, CloudService, and the AlertSystem.

C. ACITIVITY DIAGRAM

An Activity Diagram shows the workflow of an activity. For instance, when both sensors are triggered*:*

- *Start*
- *Sensors detect event and trigger an alert*
- *Alert notification is displayed on the dashboard*
- *Caretaker clicks "Reset"*
- *System deactivates alert and resets data to normal*
- *End*

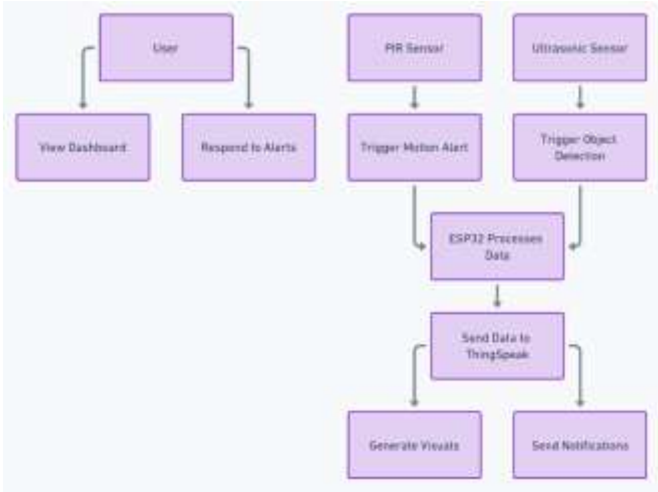

FIG. 3. CLASS & ACTIVITY DIAGRAM

C. SEQUENCE DIAGRAM

A Sequence Diagram visualizes the interaction between components over time.

Scenario: Caretaker Resetting Alert

- Caretaker: Initiates a reset via the web page.
- System: Receives reset command and triggers a reset to the microcontroller.
- Microcontroller: Deactivates the LED and Buzzer, updates the status in the cloud.
- Cloud: Updates alert status in the dashboard for real-time monitoring.

FIG. 4. SEQUENCE DIAGRAM

IMPLEMENTATION

The IoT-Based Suicide Prevention System is a comprehensive solution that integrates multiple hardware components, including sensors, a microcontroller, and communication modules, to detect abnormal activity near ceiling fans. Here's a detailed step-by-step implementation of the system:

1. <u>Powering the System:</u>
   - The system is powered by a 3.7V LiPo battery.

- The AMS1117 3.3V voltage regulator is used to step down the voltage from the LiPo battery to 3.3V, which powers the ESP32 microcontroller and other low-voltage components (like the PIR sensor and load sensor).

- The 5V power supply is used to power the ultrasonic sensor, PIR sensor and other components requiring 5V.

2. Sensor Setup:

PIR Sensor (Motion Detection)

- The PIR sensor is connected to GPIO 23 of the ESP32.

- The sensor detects the presence of a person near the ceiling fan by sensing changes in infrared radiation. If movement is detected, the PIR sensor sends a HIGH signal to the ESP32, triggering further actions

Ultrasonic Sensor (Object Detection)

- The TRIG pin of the ultrasonic sensor is connected to GPIO 19, and the ECHO pin is connected to GPIO 18 of the ESP32.

- The ultrasonic sensor sends pulses to measure the distance between the sensor and objects below the fan. If an object is detected within a predefined range (indicating potential hanging), the sensor triggers the ESP32 to process this data.

3. ESP32 Microcontroller Setup:

- The ESP32 acts as the central processing unit of the system, handling data processing from all the connected sensors (PIR, ultrasonic, and load).

- GPIO pins on the ESP32 receive input signals from the sensors.

- The ESP32 processes these signals to detect abnormal activity, such as motion, objects near the fan, or unusual weight on the fan rod, and makes a decision based on predefined conditions (e.g., motion + abnormal weight).

4. Alert Mechanisms:

Local Alerts (Buzzer and LED)

- When abnormal activity is detected, the ESP32 triggers the buzzer and LED for local alerts.

- The buzzer is connected to GPIO 4 and the LED to GPIO 2, providing both audio and visual alerts to anyone nearby.

- When a safety risk is detected, the ESP32 sends an SMS alert to registered emergency numbers, informing them of a potential suicide attempt.

5.Cloud-Based Alerts

- The ESP32 sends sensor data to a cloud platform (e.g., Firebase or ThingSpeak) via Wi-Fi.

- The cloud platform logs sensor data, enabling real-time monitoring, trend analysis, and historical data review by authorities or caretakers. Notifications can be sent to a mobile app or dashboard, allowing remote monitoring of the environment.

6. Data Logging and Monitoring:

- Data from the PIR sensor, ultrasonic sensor, and load sensor is continuously logged in the cloud database for analysis.

- The cloud platform stores data on movement, distance, and weight changes, allowing caretakers to access real-time and historical information.

- The data is used to monitor potential safety risks and recognize patterns, such as frequent disturbances or high-risk times of the day.

7. Power Management:

- The 3.7V LiPo battery provides power to the system, and the AMS1117 voltage regulator ensures that the system operates at a steady 3.3V.

- The ESP32 and sensors are designed to consume minimal power when idle. The system utilizes sleep modes to save energy during periods of inactivity.

- The battery can be recharged when needed, ensuring that the system remains operational for extended periods.

8. Integration and Mounting:

- The PIR sensor, ultrasonic sensor, and load sensor are mounted securely near the ceiling fan.

- The ESP32 microcontroller, GSM module, and other components are housed in a compact, portable enclosure that can be easily mounted to the fan rod.

- All components are connected via jumper wires or a breadboard for testing, which will later be replaced with a PCB (Printed Circuit Board) for permanent installation.

9. System Workflow:

1. Sensor Data Collection:
    - o The PIR sensor detects motion, the ultrasonic sensor measures the distance, and the load sensor monitors the weight on the fan rod.
2. Data Processing:
    - o The ESP32 processes the sensor data and compares it against predefined thresholds to determine if there is a potential risk.
3. Alert Triggering:
    - o If a safety risk is detected, the ESP32 triggers local alerts (buzzer and LED) and sends remote notifications (SMS via GSM or cloud-based alerts).
4. Data Logging:
    - o The sensor data is logged to the cloud platform for historical analysis, providing

valuable insights into patterns of risky behaviour.

RESULT

The implementation of the IoT-Based Suicide Prevention System demonstrated successful integration of IoT components for real-time monitoring, efficient alerting, and caretaker intervention. The system achieved its objective of identifying potential emergencies based on sensor data and notifying the caretaker for timely response. Below are the key results observed during the project execution:

1. Real-Time Data Monitoring

The system successfully gathered data from sensors (e.g., motion and heart rate) and transmitted it to the ThingSpeak cloud. This data was visualized on a graph, enabling the caretaker to track real-time conditions. The graph showed clear spikes whenever both sensors were triggered simultaneously, making it easier to identify critical situations.

2. Efficient Alert Mechanism

- When both sensors detected anomalies, the system activated the alert mechanisms, which included an LED and a buzzer.
- The alerts remained active until the caretaker manually reset the system via the web dashboard.
- This ensured that no critical situation went unnoticed, reducing the risk of delays in caretaker intervention.

3. Functional Caretaker Dashboard

The web-based dashboard provided a user-friendly interface for monitoring and managing alerts.

- Key Features:
  - Display of real-time sensor data fetched from the ThingSpeak cloud.
  - Visual indication of the alert status when both sensors were triggered.
  - Reset button for the caretaker to deactivate alerts after ensuring safety.

4. Reliable Data Transmission

The communication between the hardware (sensors and microcontroller) and the cloud platform was stable and seamless.

- The system used Wi-Fi for connectivity, ensuring consistent data transfer without significant delays.
- Data updates were reflected on the dashboard in near real-time, ensuring the caretaker had accurate and timely information.

5. Enhanced Caretaker Response Efficiency

The combination of real-time monitoring, instant alerts, and an intuitive dashboard significantly improved the caretaker's ability to respond quickly to emergencies.

- The immediate activation of visual and audio alerts drew attention to the situation.
- The availability of detailed sensor data on the dashboard helped caretakers understand the context and take appropriate action.

6. Scalability and Potential Applications

The success of the project highlighted the system's scalability and potential for future improvements.

- Additional sensors, such as environmental monitors or wearable devices, can be integrated to enhance detection capabilities.
- Automated notification systems using SMS or email can further improve emergency response efficiency.
- This project not only achieved its intended functionality but also demonstrated the potential of IoT-based solutions in improving safety and reducing risks.

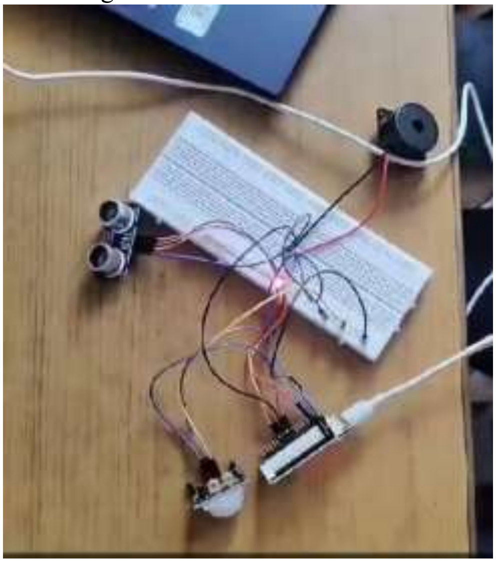

FIG. 5. SENSORS ASSEMBLED ON BREADBOARD

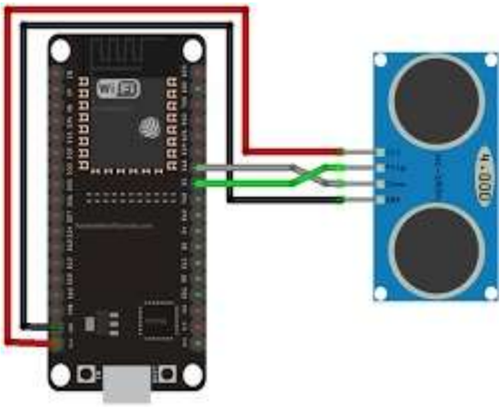

FIG. 6. ULTRASONIC SENSOR INTEGRATED WITH ESP32

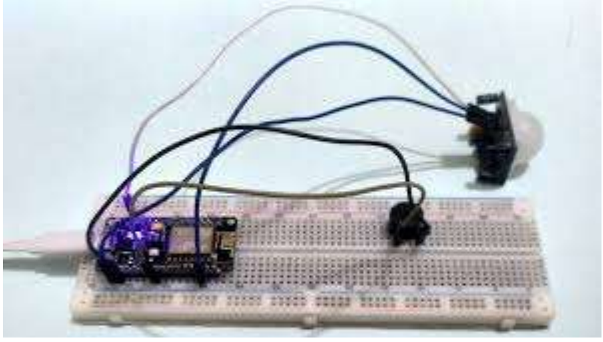

FIG. 7. PIR SENSOR ALIGNMENT

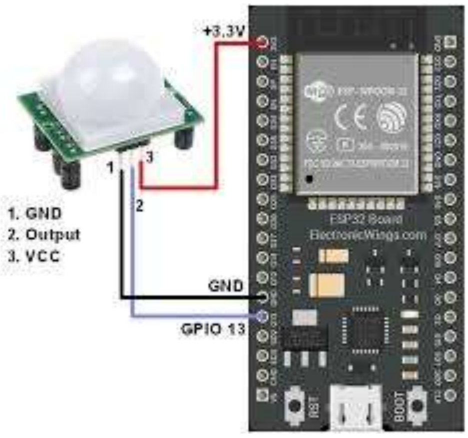

FIG. 8. PIR SENSOR INTEGRATED WITH ESP32

FIG. 9. ALERT MESSAGE WHEN SENSORS TRIGGERED

FIG. 10. THINGSPEAK GRAPHICAL VISUALIZATION BASED ON OBJECT DETECTION BY SENSORS

FIG. 11. SENSORS DISPLAYING THE OBJECT DISTANCE

FIG. 12. ALERT MESSAGE AND DATA EXPORTED TO THINGSPEAK CLOUD

FUTURE SCOPE

The project showcases immense potential for advancements and scalability to improve its functionality, reliability, and impact.

The future iterations of this system can provide even greater assistance in suicide prevention efforts by integrating more advanced features and technology, making it more robust, efficient, and accessible to a broader audience.

Below are some areas of future scope:

- Develop an emergency notification system that sends SMS, emails, or app alerts to caretakers, family members, or medical professionals during critical situations.
- Integration with services like Twilio or Firebase for real-time alerts.
- Implement predictive analytics using AI/ML algorithms to analyze sensor data over time and predict potential emergencies before they occur.
- Use anomaly detection models to identify unusual patterns in sensor data.
- Upgrade to a more sophisticated cloud platform like AWS IoT Core or Google Cloud IoT to enhance scalability and performance.
- Enable the storage of historical data for long-term analysis and visualization.
- Extend the web-based dashboard to support multi-user access, allowing several caretakers to monitor the system in shifts or remotely.
- Add user roles and permissions for enhanced security and task delegation.
- Collaborate with local healthcare providers or emergency response teams to provide instant assistance in critical situations.
- Enable automated calls to emergency services when a persistent alert is detected.
- Implement IoT modules that support cellular communication (e.g., LTE-M, NB-IoT) to enable global connectivity where Wi-Fi isn't available.
- Allow the system to operate in rural or geographically remote areas effectively.

The future iterations of this system can provide even greater assistance in suicide prevention efforts by integrating more advanced features and technology, making it more robust, efficient, and accessible to a broader audience.

The IoT-Based Suicide Prevention System successfully demonstrates how modern technologies like IoT, cloud computing, and real-time data visualization can be harnessed to address critical social issues. The project ensures continuous monitoring, instant alert mechanisms, and an intuitive caretaker dashboard, all of which collectively enable rapid response to potential emergencies. By integrating sensors, microcontrollers, and cloud services, the system creates a reliable framework that enhances safety and minimizes risks.

This project highlights the transformative power of IoT in fostering proactive solutions for human welfare. Its scalability and potential for future advancements, such as AI integration and automated notifications, pave the way for further innovation. As a proof of concept, it reinforces the role of technology in making communities safer, underlining its potential to save lives and support mental health initiatives.

REFERENCES

The development of the IoT-Based Suicide Prevention System was guided by extensive research, practical resources, and inspiration from related projects. Each component of the system, from hardware selection to software integration, was built upon established concepts and tested methodologies.

In particular, a significant reference was drawn from an IoT Smart Fan Project, which addressed a similar objective of preventing suicides. This project utilized a sensor-based system to detect weight thresholds, triggering an automated mechanism to lower the fan rod. Similarly, our project adopted a threshold-triggered approach to activate alerts, ensuring timely intervention.

Additionally, technical manuals, academic resources, cloud platforms like ThingSpeak, and programming tools were pivotal in ensuring the system's reliability and scalability. These references provided the foundation for integrating IoT technology into a practical and impactful solution.

Below is a detailed list of the tools, projects, and materials referenced during the project's design and implementation phases.

Reference Project:

- IoT Smart Fan Project: This project inspired the concept of triggering emergency mechanisms based on threshold conditions. Specifically, it involved using a sensor to detect an increase in weight on a fan. Upon detection, the fan's rod was automatically extended to the ground level to prevent harm.
  - Key Concept Derived: Using threshold-based triggers for emergency interventions

Hardware Resources

- Microcontroller: ESP32 or Raspberry Pi for sensor interfacing and cloud communication.
  - ESP32 Datasheet
- Motion Sensor (HC-SR501): Used for detecting movement as an indicator of physical activity.
  - HC-SR501 Motion Sensor Manual
- Heart Rate Sensor (Pulse Sensor): For detecting heart rate anomalies.
  - Pulse Sensor Documentation
- Alert Mechanisms: LED and buzzer for real-time notification of emergencies.
  - Circuit guides from Electronics Tutorials

Protocols and Communication

- MQTT Protocol: To facilitate lightweight communication between devices and the cloud.
  - Introduction to MQTT
- HTTP: REST API for sending and receiving data from ThingSpeak.
  - HTTP Overview

5. Web Dashboard

- Frontend Development:
  - HTML, CSS, and Bootstrap for building a clean and responsive interface for the caretaker.
    - Bootstrap Guide
- API Integration: Embedded ThingSpeak data into the dashboard using JavaScript Fetch API.
  - Fetch API Documentation

---

6. Literature and Research Material

- "Internet of Things: A Hands-on Approach" by Arshdeep Bahga and Vijay Madisetti
  - For conceptual understanding and implementation techniques.
- Blogs and Tutorials:
  - Sample code and hardware setups from IoT-related GitHub repositories.
  - Tutorials on real-time cloud integration and sensor interfacing.

Additional Acknowledgments

- Inspiration from real-world projects addressing emergency scenarios in IoT-based systems.
- Guidance from instructors and peers who provided constructive feedback during development.

3. Software Resources

- ThingSpeak Cloud: For real-time data storage and visualization.
  - ThingSpeak API and Documentation
- Programming Tools:
  - Arduino IDE: For programming the ESP32.
    - Arduino Official Page
  - Python and Flask: Used for creating the caretaker dashboard for monitoring and alert reset.
    - Flask Documentation
    - Python Documentation

ABOUT THE AUTHORS

VULLENGALA RAJ KUMAR
Dept. of Computer Science and Engineering (IOT), Siddhartha Institute of Technology and Sciences
21tq1a6907@siddhartha.co.in

MOUNIKA
Dept. of Computer Science and Engineering, Siddhartha Institute of Technology and Sciences
yaminichouhan_cse@siddhartha.co.in

NARLAPURAM SRAVAN KUMAR
Dept. of Computer Science and Engineering (IOT), Siddhartha Institute of Technology and Sciences
21tq1a6905@siddhartha.co.in

DAMERA SHIVA PRASAD
Dept. of Computer Science and Engineering (IOT), Siddhartha Institute of Technology and Sciences
21tq1a6916@siddhartha.co.in

YAYA AVANTHI
Dept.of Computer Science and Engineering (IOT), Siddhartha Institute of Technology and Sciences
21tq1a6908@siddhartha.co.in