1. **INTRODUCTION**
   1. **PROJECT DESCRIPTION**

A fire tragedy is now a frequent incident in factories, homes, markets in any country that often occurs for leakage of LPG, or liquefied petroleum gas was first discovered in 1910 by scientist Dr. Walter Snelling, a mixture of industrial-grade gases that are typically stored in liquid form in pressurized cylinders and vaporize at normal temperatures. But it may lead to major fire incidents when LPG is leaked and both material damage and human injury or death occur, which are growing day by day for so many reasons.

From 0.72% of all kitchen accidents, LPG gas leaks have risen to 10.74% of all kitchen accidents. Therefore, gas leakage detection is the most critical safety problem. This project, therefore, represents the detection and alarm system for LPG leakage to prevent fire accidents and to provide house protection to create an alert warning Global System for Mobile as its foundation. The existence of gases can be detected using it.

The sensors used in the circuit will detect that it turns on the LED whenever gas leakage, and at this time the GSM modem will send an SMS alert to the mobile user not that the gas has been leaked! "Please check" which provides notification in real-time. This device can be mounted in kitchens, storage rooms for LPG, factories, or any place that is considered appropriate.

Additionally, the system includes a sophisticated power cut-off mechanism, further reducing the risk of ignition or explosion. This feature enables swift disconnection of power sources upon gas detection, minimizing the possibility of sparks or other ignition sources causing harm. Moreover, the integration of a Telegram chatbot enhances the system's capabilities by providing remote notifications to users. Through the Telegram platform, users receive instant alerts on their mobile devices, allowing them to take necessary precautions or promptly inform authorities, even when they are not physically present at the location of the gas leak.

1. **LITERATURE SURVEY**

**2.1 EXISTING AND PROPOSED SYSTEM**

The existing system for weather monitoring typically relies on traditional methods and technologies, which may have several limitations in terms of accessibility, accuracy, and real-time data delivery. Some common components and characteristics of the existing systems include:

**1. Conventional Weather Stations:**

* Many weather monitoring systems employ conventional weather stations equipped with various sensors such as thermometers, barometers, anemometers, and hygrometers.
* These stations are often large, stationary structures installed at specific locations, such as airports, meteorological stations, or research facilities.
* Data collected by these stations are transmitted to centralized databases for analysis and dissemination to the public through weather forecasting agencies or websites.

**2. Wired Sensor Networks:**

* Some weather monitoring systems utilize wired sensor networks deployed in specific geographic areas to collect data on temperature, humidity, rainfall, wind speed, and other meteorological parameters.
* Usually, these networks are made up of a number of sensors that are coupled to a central control unit.
* The information these sensors gather is transmitted to the central control unit for processing and analysis.

**3. Remote Sensing Technologies:**

* Remote sensing technologies, such as satellites and radar systems, are often used for large-scale weather monitoring and forecasting.
* Satellites capture images of the Earth's surface and atmosphere, providing valuable data on cloud cover, precipitation, temperature, and other weather-related phenomena.
* Radio waves are used by radar devices to identify precipitation and measure its intensity and movement.

**4. Data Analysis and Forecasting:**

* Weather monitoring systems typically involve data analysis and forecasting techniques to predict future weather conditions based on historical data and current observations.
* Meteorologists use sophisticated mathematical models and computer simulations to generate weather forecasts, which are then disseminated to the public through various channels, including television, radio, and the internet.

**5. Limitations of the Current System:**

* Limited Accessibility: Traditional weather monitoring systems are often centralized and may not provide real-time data access to individuals or small-scale users.
* Costly Infrastructure: Deploying and maintaining conventional weather stations and sensor networks can be expensive, limiting their widespread adoption, especially in remote or developing regions.
* Reliance on Wired Communication: Wired sensor networks require extensive infrastructure for deployment and may be susceptible to damage or disruption due to physical constraints.
* Lack of Personalization: Existing systems may not offer personalized weather updates tailored to individual users' preferences or locations.

**2.2 FEASIBILITY STUDY**

**2.2.1. Technical Feasibility**

**Hardware Components:**

***MQ-2 Gas Sensor***: This sensor is crucial for detecting various gases including, carbon monoxide, methane, propane, hydrogen, smoke, and alcohol. It is reliable, cost-effective, and widely used in gas detection applications, ensuring its suitability for this project.

***NodeMCU* (*ESP8266*)**: This microcontroller serves as the system's brain, processing sensor inputs and controlling alert mechanisms. It offers sufficient processing power, built-in Wi-Fi capabilities for internet connectivity, and is highly cost-effective for IoT applications.

***Buzzer***: This component is employed to supply an audible alert in the event of gas detection. Its simplicity and effectiveness in alerting make it an essential a portion of the alert system.

***Power Cut-off* *Mechanism***: This component is made to turn off the power source to prevent ignition when a gas leaks, enhancing the system's safety.

***LED Indicators***: These provide visual status updates, indicating system operation and alert conditions to the users.

**Integration and Implementation:**

***Sensor Placement***: Strategic placement of the gas sensor MQ-2 in areas prone to outbursts ensures accurate detection of gas presence. Calibration and testing will be essential to guarantee precise readings under various environmental conditions.

***Microcontroller Programming***: The NodeMCU will be programmed using the Arduino IDE, leveraging its extensive libraries for sensor and GSM module interfacing. The programming logic will involve setting conditions based on sensor readings to dynamically trigger alerts and the power cut-off mechanism.

***Alert Integration***: The buzzer and GSM module will be connected to the NodeMCU to provide real-time audible and SMS alerts. The Raspberry Pi will handle more advanced tasks like remote monitoring and logging data for analysis.

***LED Display Integration***: LED indicators will be employed to supply real-time visual alerts and system status updates, aiding user awareness and response.

**Scalability and Maintenance**

***Scalability****:* The system can be easily scaled to include more sensors or integrate with additional safety mechanisms as required. The modular design allows for seamless integration of new components, making it adaptable to various settings.

***Maintenance***: Regular maintenance involves periodic calibration of the MQ-2 sensors to ensure accuracy, updating the microcontroller firmware, and routine checks among the hardware components. This might be managed with basic technical skills and regular intervals.

**Reliability and Durability**

***Component Reliability***: The chosen components (MQ-2 gas sensor, NodeMCU, buzzer, LED Lights and Exit fans) have proven reliability in similar applications, ensuring the system's overall durability and effectiveness.

***Environmental Conditions***: The system's components are designed to operate in various environmental conditions. However, additional measures like weatherproof housings for sensors might be necessary in harsh environments to ensure continuous operation and protection from external factors.

**2.2.2. Economic Feasibility**

**Initial Costs:**

***Hardware Costs***: The cost of the components, including the NodeMCU microcontroller, MQ-2 gas sensors, buzzer, and LED indicators, is relatively low. Bulk purchasing and vendor negotiations can further reduce these costs, making the system economically viable.

***Installation Costs***: Installation involves positioning sensors in strategic locations, connecting components, wiring, and initial calibration. These tasks can be performed with minimal infrastructure changes, keeping installation costs low. The use of easily available and standard components further reduces the complexity and cost of installation

**Operational Costs**

***Maintenance Costs***: Routine maintenance involves sensor calibration, firmware updates, and occasional hardware checks. These activities are low-cost and may manage with basic technical skills. The application of widely supported components ensures availability of parts and ease of maintenance.

***Energy Consumption***: The system components are energy-efficient, ensuring minimal impact on operational costs. The NodeMCU and Raspberry Pi have low power requirements, making the system cost-effective to run continuously.

**Cost-Benefit Analysis**

***Accident Prevention****:* By detecting gas leaks early and alerting users in real-time, the system can prevent potential accidents, saving lives and reducing property damage. This translates to significant cost savings in emergency response and healthcare costs.

***Insurance Benefits***: Implementation of a reliable gas detection system can lead to lower insurance premiums for homes and businesses due to reduced risk of gas-related incidents.

***Economic* *Efficiency***: Enhanced safety through timely alerts improves overall economic productivity by preventing disruptions caused by gas leaks. This benefits businesses by minimizing downtime and ensuring a safe working environment.

**2.2.3. Operational Feasibility**

**Implementation Process**

***Phased Implementation***: A phased approach, starting with a pilot implementation in a single location, allows for testing, refinement, and validation before wider deployment. This minimizes risk and ensures the system functions as expected. Feedback from the pilot phase can be used to make necessary adjustments.

***Training***: Minimal training is required for users to understand the system's operation and perform basic troubleshooting. Detailed manuals and training sessions will be provided to ensure users can effectively manage the system.

**System Reliability**

***Sensor Accuracy***: The gas sensors MQ-2 are reliable for detecting various gases and providing accurate readings. Regular calibration and testing ensure continued accuracy and reliability of the system.

***Microcontroller Performance****:* The NodeMCU and Raspberry Pi have proven track records in various IoT applications, ensuring reliable performance in processing sensor inputs and managing alerts. Their robust performance ensures the system operates smoothly under various conditions.

**User Acceptance**

***Alert Mechanisms***: The system’s use of audible (buzzer) and visual (LED indicators) alerts, along with SMS notifications via the Telegram Bot, enhances user acceptance by providing multiple layers of alerting. Integration with a Telegram chatbot for remote monitoring further improves user convenience and satisfaction.

***Safety Authorities***: The system provides real-time data and dynamic control, aiding safety authorities in better decision-making and efficient incident management.

**Operational Challenges and Mitigation**

***Environmental Conditions***: Ensuring accurate sensor readings in adverse environmental conditions is critical. Protective housings for sensors and regular maintenance can mitigate these challenges, ensuring consistent performance.

***Integration with Existing Systems***: Integrating the new system with existing safety and monitoring systems requires careful planning and coordination with relevant authorities to ensure seamless operation. This includes ensuring compatibility with other safety protocols and communication systems.

**2.2.4. Legal and Regulatory Feasibility**

**Compliance:**

***Safety Standards***: All components and their installation will comply with safety standards to ensure the system poses no risk to public safety. This includes ensuring secure electrical installations and that the system does not interfere with other safety operations.

***Regulatory Approval Process***: A detailed plan outlining the system's functionality, benefits, and compliance with regulations will be presented to safety authorities for approval. This includes risk assessments and mitigation strategies to address potential concerns.

**Data Privacy:**

***Data Collection****:* The system does not collect personal data, ensuring compliance with data privacy regulations. Only gas concentration data are collected and processed, ensuring user privacy is maintained.

**2.2.5. Environmental Feasibility**

**Emission Reduction**

***Safety Enhancements****:* By detecting gas leaks early and preventing accidents, the system contributes to reducing hazardous emissions from potential gas explosions or fires, thereby improving urban air quality.

**Energy Efficiency**

***Component Efficiency*:** The system components are energy-efficient, minimizing their environmental impact. Solar-powered options for sensors and alert mechanisms can further enhance energy efficiency and sustainability.

**Environmental Impact Assessment**

***Assessment:*** An environmental impact assessment will be conducted to ensure that the system implementation does not adversely affect the local environment. This includes evaluating the impact on air quality, noise levels, and urban aesthetics.

**Long-term Sustainability**

***Sustainability Measures***: The system promotes long-term sustainability by preventing gas leaks and subsequent accidents, contributing to the overall safety and environmental goals of urban areas. Continuous monitoring and optimization will ensure sustained environmental benefits.

**2.3 Tools and technologies**

**2.3.1. Hardware Components**

**Sensors:**

***MQ-2 Gas Sensor:***

* ***Function*:** The MQ-2 sensor is capable of identifying gases including hydrogen, methane, propane, and LPG. In accordance to the gas concentration, it produces an analog output.
* ***Features*:** It features a broad detecting range, a straightforward drive circuit, great sensitivity, and quick reaction time. The sensor is robust and appropriate for a range of settings.
* ***Application*:** This sensor is utilized in commercial, industrial, and residential settings to keep an eye out for gas leaks. Accurate detection in various environmental circumstances is ensured by proper calibration.

**Microcontroller:**

***NodeMCU ESP8266:***

* ***Function*:** As the system's central processing unit, the NodeMCU ESP8266 microcontroller manages inputs from the gas sensor and regulates the alert mechanisms.
* ***Features***: It contains digital and analog I/O pins, supports many communication protocols (I2C, SPI, and UART), and is compatible with a range of sensors and modules. It also has built-in Wi-Fi for wireless connection. It provides simple integration and minimal power consumption.
* ***Application*:** In addition to processing data from the gas sensor in real time and running control algorithms, the NodeMCU ESP8266 also communicates with alert systems like buzzers, LEDs, and the Telegram bot.

**Alert Mechanisms:**

***Buzzer*:**

* ***Function*:** In case of a gas leak, the buzzer sounds an alert, guaranteeing prompt attention.
* ***Features*:** It is small, provides a loud, clear sound, and is simple to interface with microcontrollers.
* ***Application*:** When a gas leak is discovered, the NodeMCU ESP8266 triggers the buzzer to provide a prompt and efficient warning.

***LED Indicator:***

* ***Function*:** The LED indicator provides a visual alert, indicating the status of the gas leakage monitoring system.
* ***Features*:** It is bright, energy-efficient, and easy to integrate with the microcontroller.
* ***Application*:** The LED indicator is activated alongside the buzzer to provide a visual cue of a gas leak, enhancing the alert system's effectiveness.

**Communication Module:**

***Telegram Bot:***

* ***Function*:** The Telegram bot sends notifications to users via the Telegram messaging app, providing remote alerts about gas leaks.
* ***Features*:** It supports real-time communication, is easy to set up, and allows for automated messaging.
* ***Application*:** When a gas leak is discovered, the NodeMCU ESP8266 causes the Telegram bot to notify users, allowing for distant monitoring and prompt action.

**Additional Hardware:**

***Power Supply Units:***

* ***Function*:** To power the microcontroller, sensors, and alert mechanisms.
* ***Features*:** These can be standard AC adapters or battery packs. For sustainable energy use, solar power units can also be considered.
* ***Application*:** A reliable power source is essential for the continuous operation of the gas leakage monitoring system.

**Mounting Accessories:**

* ***Function*:** To securely mount sensors and alert mechanisms.
* ***Features*:** Brackets, housings, and poles designed for durability and ease of installation. Weatherproof housings protect components from environmental factors.
* ***Application*:** Proper mounting ensures the optimal placement and protection of the gas sensor and alert mechanisms.

**Cabling and Connectors:**

* ***Function*:** To ensure stable and reliable connections between all system components.
* ***Features*:** High-quality cables and connectors that support the necessary electrical and communication requirements.
* ***Application*:** Reliable cabling and connectors are crucial for the efficient operation of the gas leakage monitoring system.

**2.3.2. Software Components**

**Development Environment:**

***Arduino IDE:***

***Function*:** The Arduino Integrated Development Environment (IDE) is used for writing, compiling, and uploading code to the Arduino microcontroller.

***Features*:** It offers a user-friendly interface, supports extensive libraries for various sensors and modules, and is cross-platform compatible (Windows, macOS, Linux).

***Application*:** The Arduino IDE is used to develop the control logic, handle sensor inputs, and manage traffic signal timings.

**Programming Languages:**

***C/C++:***

***Function*:** Primary programming languages used in the Arduino IDE.

***Features*:** These languages are efficient and widely used for embedded systems programming, providing extensive community support and libraries.

***Application*:** Writing firmware for the Arduino Mega 2560 to process sensor data, control signal timings, and manage display outputs.

**Libraries and Frameworks:**

***Arduino Libraries:***

Wire.h: For I2C communication with sensors and displays.

**2.3.3. Network and Communication Technologies**

1. **Wi-Fi:** Wi-Fi is used for connecting the NodeMCU to the internet. It provides high-speed internet connectivity, enabling the NodeMCU to communicate with the Telegram servers and send real-time notifications.
2. **HTTP/HTTPS:** The NodeMCU uses HTTP/HTTPS protocols to interact with the Telegram Bot API. HTTPS ensures that the communication is secure and encrypted, protecting the data exchanged between the system and the Telegram servers.
3. **Telegram Bot API:**The Telegram Bot API is a powerful interface provided by Telegram to create and manage bots. It allows the NodeMCU to send messages, notifications, and alerts to users. The API also supports receiving commands from users, enabling two-way communication.

**System Operation**

1. **Detection:** The MQ-2 gas sensor continuously monitors the air for the presence of gas. When the gas concentration exceeds a predefined threshold, the sensor outputs a signal to the NodeMCU. This constant monitoring ensures that any gas leak is detected as soon as it occurs, providing the earliest possible warning.
2. **Alert** **Activation:** Upon detecting a high concentration of gas, the NodeMCU processes the signal from the MQ-2 sensor and activates the buzzer and LED indicator. This dual alert system ensures that both auditory and visual warnings are provided, making it highly likely that anyone nearby will notice the alert and take immediate action.
3. **Communication:** The NodeMCU uses its Wi-Fi connection to access the internet and send an HTTP/HTTPS request to the Telegram Bot API. This request triggers the bot to send an alert message to the user’s Telegram account. The message includes details about the gas leak, such as the location and concentration of the gas, allowing the user to take appropriate action remotely.
4. **Safety** **Response:** The power cutting mechanism is engaged to shut off the gas supply automatically. This immediate response helps to reduce the possibility of fire or explosion by preventing further gas leakage. The automatic shutdown of the gas supply is a critical safety measure that significantly reduces the potential for harm case a gas leak occurs.

**2.4 Hardware and software requirements**

**2.4.1 Hardware Requirements**

|  |
| --- |
| * NodeMCU ESP8622 |
| * LED lights |
| * 5V DC Exit Fans |
| * Wi-Fi or GSM Module |
| * Power Supply Components |
| * Breadboard and Jumper Wires |
| * Resistors and Capacitors |

**2.4.2 Software Requirements**

|  |  |
| --- | --- |
|  | * Arduino IDE |
|  | * Arduino Libraries |
|  | * Sensor Calibration Software |

1. **SOFTWARE REQUIREMENTS SPECIFICATION**

**3.1 Users**

**Homeowners:** Homeowners are responsible for maintaining safety within their residential environments. Their primary responsibilities include regularly checking the system status through notifications received on their devices, taking immediate action upon receiving alerts (such as evacuating the area or calling emergency services), and performing basic maintenance tasks, like checking sensor placement and ensuring the system is powered. Homeowners need an intuitive mobile interface that provides clear and concise notifications and system status updates. They also require real-time alerts with detailed information about the location and gas concentration, a reliable system with minimal false alarms and consistent performance, and the assurance that the system will detect leaks and automatically shut off the gas supply to prevent hazards.

**Building Managers:** Building managers oversee the safety and maintenance of commercial or residential buildings. They continuously monitor the status of gas leakage systems in multiple units or buildings, work with maintenance teams to address any technical issues or perform routine checks, and coordinate evacuation procedures and what to do in case of a gas leak in terms of emergencies. Building managers need a centralized dashboard for monitoring multiple systems across different buildings, up-to-date information on gas levels, system status, and alert history, analytical tools to analyze trends, generate reports, and identify potential maintenance needs, and secure access to system data to protect sensitive information and make sure that only individuals have permission can manage the system.

**Emergency Services:** Emergency services personnel, including police, fire, and medical responders, need to be aware of gas leaks to ensure public safety. They quickly respond to gas leak incidents, manage the affected area, work closely with building managers to manage evacuations, and ensure the area is safe. Emergency responders need immediate alerts with precise location details, reliable system communication to ensure prompt receipt of alerts, comprehensive data on the nature of the gas leak, including gas type and concentration levels, and seamless integration with existing emergency response systems for coordinated efforts.

**General Public:** Residents and visitors within the monitored area are responsible for following safety instructions and evacuation procedures during a gas leak and being aware of and responding to alarms and notifications from the system. The general public needs clear and audible alerts to ensure everyone in the vicinity is aware of a gas leak, efficient evacuation instructions and guidance, and the assurance that the system will detect and manage gas leaks effectively to ensure their safety.

**3.1.1 System components and operations**

***Gas Sensor (MQ-2):*** The MQ-2 gas sensor is a semiconductor sensor capable of detecting gases such as propane, methane, and butane. It continuously monitors the air and outputs an analog signal proportional to the gas concentration. This sensor is highly sensitive and can detect even small amounts of gas, making it crucial for early leak detection.

***NodeMCU:*** NodeMCU is a microcontroller with built-in Wi-Fi capabilities, based on the ESP8266 or ESP32 chip. It processes signals from the MQ-2 sensor, controls other components (buzzer, LED, power cutting mechanism), and communicates with the Telegram bot API to send alerts. The NodeMCU is integral in managing the system's operations and ensuring timely communication with users.

***Telegram Bot*:** A bot created using the Telegram Bot API facilitates communication between the gas leakage monitoring system and users via the Telegram messaging app. The bot sends real-time notifications to users about gas leaks, including detailed information about the leak, and can also receive commands from users for system status checks and other interactions. This ensures users are always informed and can take necessary actions promptly.

***Buzzer*:** The buzzer provides an audible alarm when the gas concentration exceeds a predefined threshold. It emits a loud sound to warn anyone nearby of the potential danger, ensuring immediate awareness of the situation.

***LED Indicator*:** The LED indicator offers a visual alert to complement the audible alarm. When a gas leak is detected, the LED lights up or blinks, providing a clear visual indication of danger. This ensures that anyone in the vicinity can see the warning, even from a distance, thereby enhancing the overall effectiveness of the alert system.

***Power Cutting Mechanism*:** The power cutting mechanism is connected to the gas supply and automatically shuts off the gas flow when a leak is detected. This immediate response helps to reduce the possibility of fire or explosion by preventing further gas leakage. The automatic shutdown of the gas supply is a critical safety measure that significantly reduces the potential for harm in the situation of a gas leakage.

**3.1.2 Network and communication technologies**

***Wi-Fi:*** Wi-Fi is used for connecting the NodeMCU to the internet, providing high-speed internet connectivity, enabling the NodeMCU to communicate with the Telegram servers and send real-time notifications.

***HTTP/HTTPS:*** The NodeMCU uses HTTP/HTTPS protocols to interact with the Telegram Bot API. HTTPS ensures that the communication is secure and encrypted, protecting the data exchanged between the system and the Telegram servers.

***Telegram Bot API*:** The Telegram Bot API is a powerful interface provided by Telegram to create and manage bots. It allows the NodeMCU to send messages, notifications, and alerts to users and supports receiving commands from users, enabling two-way communication.

**3.2 Functional requirements**

**3.2.1 Gas Leakage Detection and Alert System**

The system must accurately detect the presence of hazardous gases like propane, methane, and butane using the MQ-2 gas sensor. The MQ-2 sensor will continuously monitor the air for gas concentrations and provide real-time data. It is capable of detecting gas concentrations from 200 to 10000 ppm (parts per million), and this data will be processed by the NodeMCU to determine if gas levels exceed a predefined safety threshold. To ensure accuracy, the system must filter out false positives.

When gas is detected, the system must trigger multiple alert mechanisms, including visual, auditory, and remote notifications. A buzzer will sound an audible alarm when gas levels exceed the safety threshold, providing an immediate warning to occupants. An LED indicator will provide a visual alert by lighting up or blinking to indicate the presence of gas. Additionally, the system will send real-time notifications via a Telegram bot to inform users on their internet-enabled devices about the gas leak, including detailed information about the gas type and concentration level to assist in quick and effective responses.

**3.2.2 Sensor Data Collection**

The system must collect accurate gas concentration data using the MQ-2 sensor. The sensor will detect gas concentrations within its specified range and provide continuous data to the NodeMCU for real-time processing and decision-making. It is crucial to ensure that sensor readings are precise and reliable to minimize the chances of false alarms.

Furthermore, the system must monitor the environmental conditions around the gas sensor to maintain accuracy. This involves accounting for factors such as temperature and humidity that could affect gas sensor readings. Data from environmental monitoring will be used to calibrate the gas sensor and adjust detection thresholds as necessary.

**3.2.3 Data Processing and Decision Making**

The system must process sensor data in real-time to make prompt and accurate decisions. The NodeMCU will process data from the MQ-2 sensor to determine current gas concentration levels. This data will be evaluated against predefined safety thresholds to identify potential gas leaks, and the data processing must be efficient to ensure timely activation of alerts and safety mechanisms.

Predefined algorithms will be used to determine the appropriate response to detected gas levels. These algorithms must consider current gas concentration, historical data, and environmental conditions to decide on the activation of alerts. The system must also refine its decision algorithms based on new data to improve accuracy over time. Immediate safety measures, such as triggering alerts and activating the power cutting mechanism, must be prioritized upon detection of dangerous gas levels.

**3.2.4 Communication and Monitoring**

The system must enable remote monitoring and control via the Telegram bot. It will send real-time notifications to users' Telegram accounts, providing detailed information about gas leaks. Users must be able to interact with the system through the Telegram bot to check system status and receive updates, and communication between the system and the Telegram bot must be secure and encrypted to protect data integrity and privacy.

Additionally, the system must generate alerts for significant events such as gas leaks or sensor malfunctions. These alerts must be sent to users in real-time via the Telegram bot, enabling swift responses to emergencies. Notifications will include detailed information about the nature of the alert, such as gas type, concentration level, and location. The system will categorize alerts by severity to prioritize responses to critical issues.

**3.2.5 User Interface**

The system must provide a user-friendly interface for monitoring and managing gas leakage alerts. This centralized dashboard will display real-time data on gas concentrations, system status, and alert history in an intuitive format. The interface must allow users to check system status, view historical data, and manage alert settings. To ensure security, user authentication and role-based access control must be implemented, allowing only authorized personnel to make changes to the system settings.

Lastly, the system must provide clear and effective notifications to users through the Telegram bot. Notifications must be concise and include essential details to help users understand and respond to gas leaks promptly. The Telegram bot will support interactive commands, allowing users to query system status and receive real-time updates. Notifications will be prioritized and categorized to ensure that critical alerts receive immediate attention.

**3.3 Non-functional requirements**

**3.3.1 Performance Requirements**

**Response Time**

The system must promptly detect gas leaks and trigger appropriate responses to ensure safety and mitigate risks. The system must process sensor data and trigger alarms within 1 second of gas detection to ensure timely response and mitigation. The control unit must immediately activate ventilation systems and shut off gas valves to prevent escalation. Notifications to the monitoring station and relevant authorities must be sent within 2 seconds of leak detection.

**Throughput**

The system must handle continuous monitoring and rapid response for multiple detection points simultaneously. It must support real-time data processing from up to 100 sensors within an industrial setting without delays. The control unit must manage up to 1,000 alerts per day without performance degradation. The communication modules must handle up to 50 remote commands per hour to ensure robust remote monitoring and control capabilities.

**3.3.2 Reliability and Availability**

**System Uptime**

The system must ensure continuous operation to provide uninterrupted gas leakage detection and response. It must achieve an uptime of 99.9%, allowing for minimal downtime for maintenance and updates. Backup power supplies must ensure uninterrupted operation for at least 24 hours during power outages. Redundant components and failover mechanisms must prevent single points of failure and ensure system resilience.

**Fault Tolerance**

The system must maintain its functionality even in the presence of hardware or software failures. It must automatically detect and isolate faulty components, switching to backup systems when necessary to maintain operation. Sensor data must be cross-verified to detect and compensate for sensor malfunctions, ensuring accurate gas leak detection. The system must log and report faults in real-time to enable quick diagnostics and repairs, minimizing downtime.

**3.3.3 Scalability**

**Horizontal Scalability**

The system must be scalable to accommodate additional detection points and increased monitoring needs without significant reconfiguration. It must support the addition of more sensors with minimal reconfiguration, enabling easy expansion. Communication protocols must handle increased data traffic from multiple detection points, ensuring smooth data flow. The system must scale to manage up to 200 sensors simultaneously without significant performance impact.

**Vertical Scalability**

The system must support enhancements in processing power and data handling capabilities to adapt to future needs. It must allow upgrading control units to more powerful versions as needed, enhancing processing capabilities. Sensor interfaces must support higher sensitivity sensors for improved detection accuracy. Data storage and processing units must be upgradable to handle increased data volumes from expanded system deployment.

**3.3.4 Usability**

**User Interface**

The system's interface must be user-friendly and intuitive for monitoring personnel, ensuring ease of use. The web-based dashboard must present real-time gas concentration data, sensor statuses, and alerts clearly to facilitate monitoring and control. The interface must support easy configuration and manual override of response actions, allowing flexibility in emergency management. Training materials and user guides must be provided to ensure effective system use by monitoring personnel.

**Accessibility**

The system must be accessible to users with different needs and abilities, ensuring inclusivity. The interface must comply with accessibility standards such as WCAG 2.1 to ensure usability for all users. Visual indicators must be supplemented with auditory alarms for users with visual impairments. The system must offer multiple language support for diverse user groups, ensuring wider usability.

**3.3.5 Security**

**Data Security**

The system must ensure the confidentiality, integrity, and availability of data, protecting it from unauthorized access and tampering. All data transmissions must be encrypted using SSL/TLS protocols to protect data in transit. Stored data must be protected with encryption and access control mechanisms to prevent unauthorized access. Regular security audits must be conducted to identify and mitigate vulnerabilities, ensuring robust security.

**Access Control**

The system must restrict access to authorized personnel only, ensuring controlled and secure usage. User authentication mechanisms, such as username and password, must be in place to secure access. Role-based access control must ensure that users can only perform actions within their permissions, maintaining operational integrity. Logs of user actions must be maintained for accountability and traceability, enabling oversight and auditing.

**3.3.6 Maintainability**

**Modularity**

The system must be designed with a modular architecture to simplify maintenance and updates, ensuring long-term operability. Components such as sensors, control units, and communication modules must be easily replaceable, facilitating quick repairs. The system software must be modular, allowing updates and bug fixes without affecting overall functionality. Documentation must be provided for each module to facilitate maintenance and troubleshooting.

**Documentation**

Comprehensive documentation must be available for system maintenance and upgrades, ensuring smooth operation and knowledge transfer. Installation guides, user manuals, and maintenance procedures must be provided to support users and technicians. System architecture and design documents must be available for reference, ensuring clear understanding of the system. Documentation must be regularly updated to reflect changes and improvements.

**3.3.7 Compliance**

**Regulatory** **Compliance**

The system must comply with relevant industry standards and regulations, ensuring legal and operational adherence. It must meet safety standards set by local and national authorities, ensuring regulatory compliance. Environmental regulations concerning emissions and energy consumption must be adhered to, ensuring environmental responsibility. Data privacy regulations such as GDPR or CCPA must be followed to protect user data.

**Environmental Impact**

The system must minimize its environmental impact, contributing to sustainable operations. Energy-efficient components must be used to reduce power consumption. The system must aim to prevent environmental hazards by detecting gas leaks promptly, contributing to safety. Disposal and recycling guidelines for electronic components must be followed, ensuring responsible e-waste management.

1. **SYSTEM DESIGN**

**4.1 System perspective**

The architecture of the proposed gas leakage detection and response system is designed to effectively monitor and respond to gas leaks in residential and commercial environments. At its core, the system is built around a NodeMCU, which serves as the central processing unit. This microcontroller is responsible for collecting and processing data from a gas sensor, managing alerts, and controlling safety mechanisms. The NodeMCU is chosen for its Wi-Fi capabilities, sufficient I/O capabilities, and compatibility with various sensors and modules, making it well-suited for real-time gas leak detection and response.

The sensor network comprises an MQ-2 gas sensor that detects the presence of gases such as propane, methane, and butane. The data from this sensor is transmitted to the NodeMCU, which processes the inputs using pre-programmed algorithms to determine the appropriate response. This ensures timely detection and response to gas leaks, significantly enhancing safety measures.

The output of the processed data is communicated to various components: an audible alarm via a buzzer, a visual alert through an LED indicator, and a notification system using a Telegram bot. The Telegram bot sends real-time alerts to users' internet-enabled devices, allowing for remote monitoring and prompt response to gas leaks. Additionally, the system includes a power cutting mechanism that automatically shuts off the gas supply when a leak is detected, preventing potential hazards such as explosions or fires.

The power supply architecture is designed to ensure uninterrupted operation of the system. It includes a 9V battery as the primary power source, complemented by a linear regulator (7805) to stabilize the voltage supplied to the electronic components. This robust power management system ensures that the gas leakage detection and response system remain operational even in the event of external power failures, maintaining continuous safety.

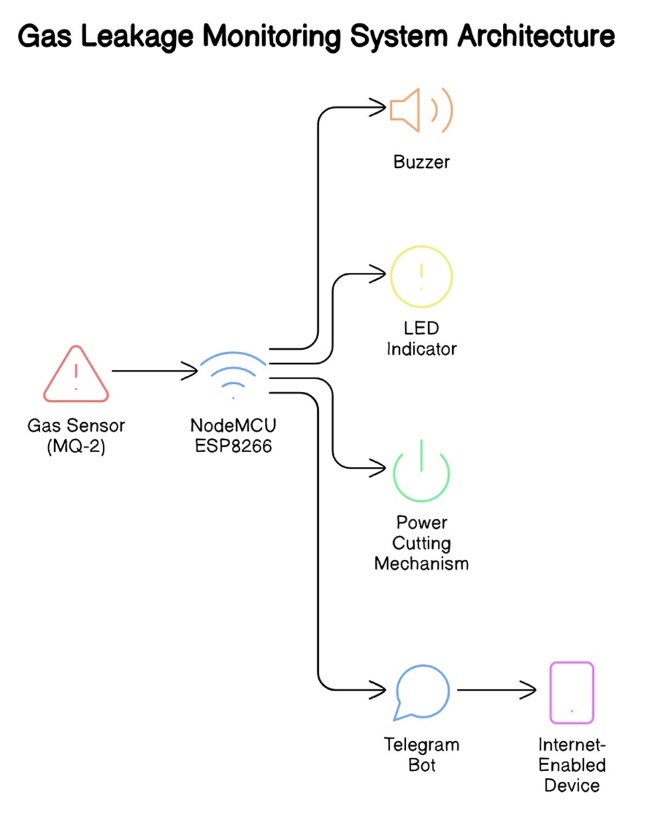


Figure 1: Architecture of Gas Leakage Monitoring system

**4.1.1 System Architecture**

The architecture of the gas leak detection system that makes use of a Telegram bot is depicted in the schematic diagram. The system consists of a number of essential parts, each performing a crucial part in the detection and alert process:

**Gas Sensor (MQ-2)**: The MQ-2 gas sensor is a semiconductor sensor that detects the presence of gases such as propane, methane, and butane. It outputs an analog signal proportional to the gas concentration. The sensor is highly sensitive and able to detect even low levels of gas leaks, establishing it as an essential component for early detection.

**NodeMCU**: NodeMCU is a low-cost open-source IoT platform that includes firmware that runs on the ESP8266 Wi-Fi SoC and hardware according to the ESP-12 module. It processes the gas sensor's signal and controls the other components in the system. The NodeMCU's built-in Wi-Fi capability allows it to connect to the internet, enabling remote monitoring and control.

**Buzzer**: The buzzer serves as an audible alarm that activates when the gas concentration exceeds a predefined threshold, providing an immediate warning to anyone nearby. The sound emitted by the buzzer is loud enough to alert occupants, ensuring that they are aware of the potential danger even if they are not in the immediate vicinity of the sensor.

**LED Indicator**: The LED provides a visual indication of a gas leak, complementing the audible alarm to ensure the alert is noticeable even in noisy environments. The LED indicator is strategically placed to be visible from different angles, ensuring that the visual alert can be seen from various parts of the room or area.

**Telegram Bot**: Instead of using a GSM module, the system employs a Telegram bot to send notifications to users' internet-enabled devices (e.g., smartphones, tablets). This bot provides real-time alerts about the gas leak, allowing users to monitor the situation remotely and respond promptly. The use of a Telegram bot offers a flexible and user-friendly way to receive notifications, as it can be easily integrated into existing communication channels on users' devices.

**Power Cutting Mechanism**: This mechanism is connected to the gas supply and is triggered to shut off the gas flow automatically when a leak is detected, thereby preventing potential hazards such as explosions or fires. The power cutting mechanism is a critical safety feature that ensures immediate action is taken to mitigate the risk of further gas accumulation and potential ignition sources.

**3.2 Operation**

The operation of the monitoring system for gas leaks employing a Telegram bot can be summarized in the following steps:

**Detection**: The MQ-2 gas sensor continuously monitors the air for the presence of gas. When the gas concentration exceeds a predefined threshold, the sensor outputs a signal. The continuous monitoring ensures that any gas leak is detected as soon as it occurs, providing an early warning system.

**Alert Activation**: Upon detecting a high concentration of gas, the NodeMCU processes the sensor data and activates the buzzer and LED indicator to provide immediate auditory and visual alerts. This dual alert system ensures that the warning is noticeable regardless of the environment, whether it is quiet or noisy.

**Communication**: Simultaneously, the NodeMCU sends a notification to the Telegram bot. The bot then sends an alert message to the user's internet-enabled device. This message includes details about the gas leak, allowing the user to take appropriate action remotely. The alert message can contain information such as the time of detection, the concentration of gas detected, and suggested actions for the user.

**Safety Response**: The power cutting mechanism is engaged to shut off the gas supply automatically, minimizing the risk of fire or explosion. This immediate response to detected gas leaks significantly reduces the potential for hazardous situations, protecting both property and lives.

**Extended Features**:

1. ***Data Logging***: The system can include a feature for logging the gas concentration data over time. This data can be stored locally on the NodeMCU or sent to a remote server for further analysis. Historical data can be useful for identifying patterns in gas leaks, which can inform preventative measures and maintenance schedules.
2. ***Battery Backup***: To ensure continuous operation during power outages, the system can be equipped with a battery backup. This backup power supply ensures that the gas leak detection and response system remain functional even during electrical disruptions, providing uninterrupted protection.
3. ***Periodic Testing***: The system can be programmed to perform periodic self-tests to ensure that all components are functioning correctly. Regular testing of the sensor, buzzer, LED indicator, and power cutting mechanism can help in early identification of any faults, ensuring the system's reliability.
4. ***User Interface***: An optional user interface can be developed to allow users to interact with the system. This interface can provide real-time status updates, allow users to view historical data, configure alert thresholds, and manually test system components.
5. ***Integration with Smart Home Systems***: The gas leakage monitoring system can be integrated with existing smart home systems for enhanced functionality. For example, the system can trigger smart home devices to open windows or activate ventilation systems when a gas leak is detected, further improving safety measures.

**4.2 Context diagram**

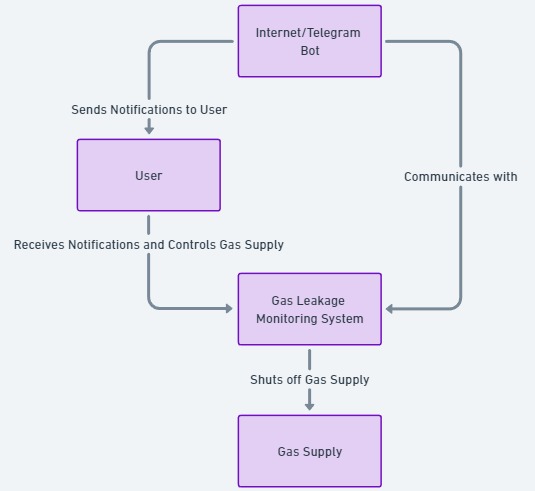


Figure 2: Context Diagram of whole model

The context diagram for the gas leak detection system that makes use of a Telegram bot illustrates the interactions between the various components and actors involved in the system. This diagram provides a high-level overview of how information and control signals flow within the system, ensuring a comprehensive understanding of its operation.

**Internet/Telegram Bot**

The Internet/Telegram bot is a crucial component in the gas leakage monitoring system. It serves as the communication bridge between the monitoring system and the user. When a gas leak is discovered by the gas leakage monitoring system, it transmits a notification to Telegram bot. The bot then promptly sends this alert to the user’s internet-enabled device, such as a smartphone or tablet. This ensures that the user receives real-time updates about the gas leak, allowing them to respond quickly to potential dangers. The use of the Telegram bot leverages internet connectivity to provide a reliable and efficient means of communication, ensuring users are always informed regardless of their location.

**User**

The user is the primary actor in this system serving as a key component in responding to gas leak notifications. Upon receiving an alert from the Telegram bot, the user can take necessary actions to mitigate the risk. These actions might include contacting emergency services, manually shutting off the gas supply, or evacuating the area. The user’s ability to receive and act on real-time notifications significantly enhances the safety and effectiveness of the gas leakage monitoring system. By ensuring that the user is always informed of any gas leaks, the system empowers them to make timely decisions that can prevent accidents and protect property and lives.

**Gas Leakage Monitoring System**

The gas leakage monitoring system is the core of the setup, responsible for detecting gas leaks and initiating safety protocols. It continuously monitors gas levels using sensors, and when a leak is detected, it activates an alert mechanism. This system sends a notification to the Internet/Telegram bot, which then informs the user. Additionally, the gas leakage monitoring system can autonomously shut off the gas supply to prevent the accumulation of gas, thereby reducing the risk of fire or explosion. This automated response is crucial for immediate hazard mitigation, especially in scenarios where the user might not be able to respond instantly.

**Gas Supply**

The gas supply is the source that the system monitors and controls. In the event of a gas leak, the gas leakage monitoring system sends a signal to shut off the gas supply. This automatic shut-off mechanism is a critical safety feature designed to prevent further gas leakage, thereby minimizing the risk of dangerous incidents such as explosions or fires. The gas supply's integration into the system ensures that any detected leak is promptly addressed by cutting off the source, thus providing an additional layer of safety and preventing potential hazards from escalating.

**Flow of Information and Control**

The flow of information and control in this system begins with the gas leakage monitoring system detecting a gas leak. Upon detection, the system sends a control signal to shut off the gas supply, ensuring immediate mitigation of the hazard. Simultaneously, the monitoring system communicates with the Internet/Telegram bot, transmitting detailed information about the leak. The Telegram bot then relays this information to the user, sending real-time notifications to their internet-enabled devices. The user, informed about the gas leak, can then take further actions, such as contacting emergency services or manually controlling the gas supply. This seamless flow of information and control ensures that gas leaks are promptly detected, communicated, and mitigated, enhancing overall safety and responsiveness.

In summary, the context diagram showcases a well-integrated system where the gas leakage monitoring system, Internet/Telegram bot, and user work in tandem to ensure safety and prompt response to gas leaks. This comprehensive approach not only enhances safety but also provides users with the convenience of remote monitoring and control, making it highly effective for modern smart homes and industrial applications.

1. **DETAILED DESIGN**

**5.1 Use case diagram**

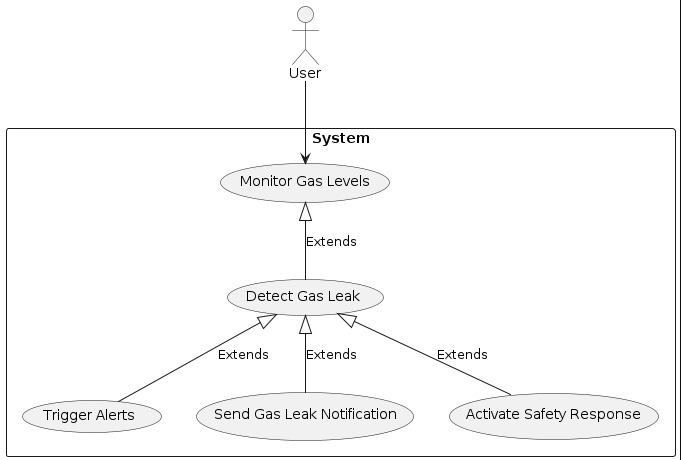
****

Figure 3: A complete Use case Diagram

The use case diagram for the gas leak detection system that makes use of a Telegram bot outlines the interactions between various actors and the system's functionality.

**5.1.1 Actors:**

**User**:

The User represents the individuals or entities responsible for receiving notifications about gas leaks and taking appropriate action. Users receive gas leak notifications via the Telegram bot, which informs them remotely about the gas leak. This allows them to take necessary actions such as contacting emergency services or manually shutting off the gas supply to mitigate any potential hazards.

**Maintenance Personnel (MP):**

Maintenance Personnel are responsible for maintaining and ensuring the proper functioning of the gas leakage monitoring system. Their tasks include calibrating sensors, repairing system components, and performing software updates. Maintenance Personnel regularly check and maintain the gas sensor and other hardware components to ensure accurate data processing. They address any system alerts related to hardware malfunctions or required updates to keep the system reliable and accurate.

**5.1.2 Use Cases:**

**Monitor Gas Levels:**

This use case involves the continuous monitoring of gas concentration in the environment using the MQ-2 gas sensor. The sensor detects gases such as propane, methane, and butane, ensuring early detection of leaks. This continuous monitoring is crucial for maintaining a safe environment.

**Detect Gas Leak:**

In this use case, the system identifies when the gas concentration exceeds predefined safety thresholds. Upon detection of high gas levels, the system triggers alerts and engages safety mechanisms to prevent potential hazards.

**Trigger Alerts:**

When a gas leak is detected, the system activates the buzzer and LED indicator to provide immediate auditory and visual alerts. This ensures that nearby individuals are quickly informed of the danger, allowing them to take immediate action to ensure their safety.

**Send Gas Leak Notification:**

Simultaneously with triggering local alerts, the system sends a notification to the user’s internet-enabled device via the Telegram bot. This notification provides real-time information about the gas leak, enabling the user to respond promptly from a remote location.

**Activate Safety Response:**

This use case involves the activation of the power cutting mechanism, which automatically shuts off the gas supply when a leak is detected. This safety measure is designed to prevent potential hazards such as explosions or fires by stopping the flow of gas immediately.

**5.1.3 Relationships:**

**Include:**

***Monitor Gas Levels includes Detect Gas Leak:***

Monitoring gas levels involves detecting when the gas concentration exceeds safety thresholds.

***Detect Gas Leak includes Trigger Alerts:***

Detection of a gas leak necessitates the triggering of immediate alerts.

***Detect Gas Leak includes Send Gas Leak Notification:***

Detection of a gas leak also involves sending notifications to users for remote awareness and action.

***Detect Gas Leak includes Activate Safety Response:***

Detection of a gas leak triggers the power cutting mechanism to ensure safety.

**Communicates/Interacts:**

***User:***

Receives notifications to stay informed about gas leaks.

May manually engage safety responses if automated systems fail.

***Maintenance Personnel (MP):***

* Ensures sensors are properly calibrated and functioning.
* Verifies that detection systems are accurate.
* Ensures the buzzer and LED indicators are operational.
* Confirms that the notification system is functioning correctly.
* Maintains the power cutting mechanism for reliability.

**5.1.4 Expanded Workflow and System Functionality:**

**Initialization and Calibration:**

At the start, the system initializes all components and calibrates sensors to ensure accurate data collection. This step is crucial for preparing the system for accurate and reliable operation. Maintenance Staff Members are in charge of this process.

**Real-Time Monitoring and Data Collection:**

The system continuously monitors gas levels using the MQ-2 sensor, collecting real-time data on gas concentration. Maintenance Personnel oversee this process to ensure that the data collection remains consistent and accurate.

**Data Processing and Analysis:**

The collected sensor data is processed and analyzed to ascertain if gas levels exceed safety thresholds. Maintenance Personnel make certain the data processing systems function correctly to translate raw data into actionable insights for gas leak detection.

**Dynamic Safety Response:**

Based on the processed data, safety mechanisms such as the power cutting mechanism are activated to shut off the gas supply. This step is crucial for preventing potential hazards. Users benefit indirectly from this safety measure as it reduces the risk of fire or explosion.

**User Interaction:**

Notifications are sent to users' devices via the Telegram bot to inform them of gas leaks. Users respond to these notifications and take necessary actions, such as contacting emergency services or manually shutting off the gas supply.

**Alert and Notification System:**

The system sends alerts via buzzer, LED indicator, and Telegram bot in case of detected gas leaks. Users receive and act on these alerts to ensure timely interventions and maintain safety.

**Maintenance and Updates:**

Regular maintenance and updates of the system ensure continued accuracy and functionality. Maintenance Staff Members are in charge of preventing system failures and ensuring reliable operation through regular checks and updates.

**5.2 Sequence diagram**

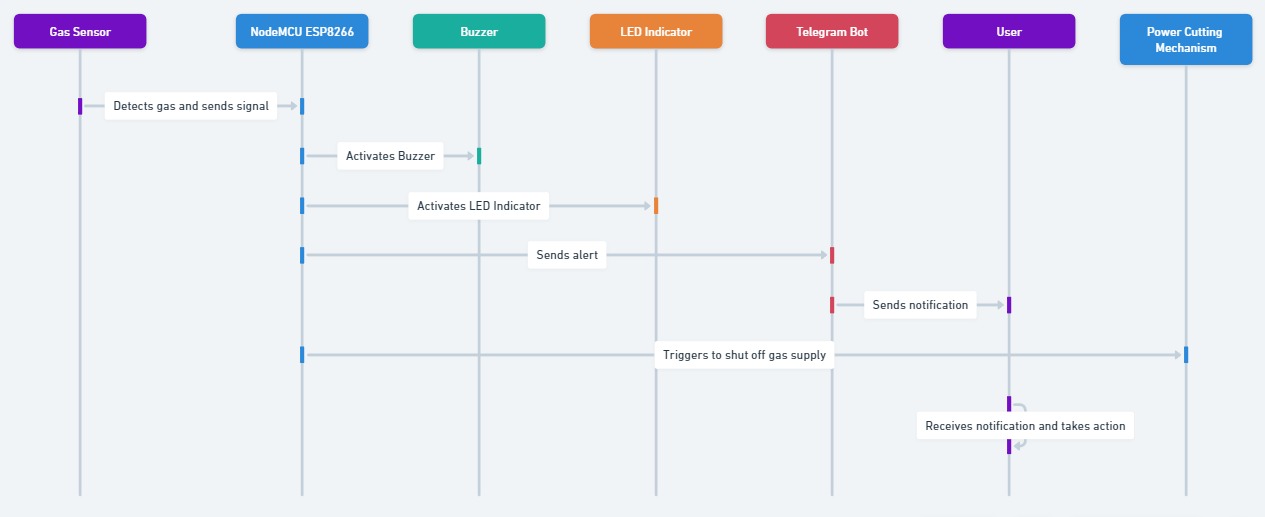
****

Figure 4: Sequence Diagram

The sequence diagram for the gas leakage monitoring and alert system provides a detailed visual representation of the interactions between various components over time. This analysis explains how each component and actor within the system contributes to detecting gas leaks and ensuring safety through notifications and automated responses.

**Actors and System Components:**

***Gas Sensor*:** The gas sensor is responsible for detecting the presence of gas. Once gas is detected, the sensor sends a signal to the NodeMCU ESP8266 microcontroller. This detection is the critical first step in the sequence, triggering the subsequent actions aimed at ensuring safety.

***NodeMCU ESP8266:*** The NodeMCU ESP8266 microcontroller acts as the central processing unit of the system. Upon receiving the signal from the gas sensor, it activates both the buzzer and the LED indicator. Additionally, it sends an alert to the Telegram bot, initiating the notification process to inform the user. The NodeMCU also triggers the mechanism to shut off the gas supply, ensuring immediate hazard mitigation.

***Buzzer:*** The buzzer provides an audible alert when activated by the NodeMCU ESP8266. This sound alert serves as an immediate warning to anyone nearby, prompting them to take necessary actions to ensure their safety.

***LED Indicator*:** The LED indicator provides a visual alert when activated by the NodeMCU ESP8266. This visual cue complements the buzzer's audible alert, ensuring that even in noisy environments, the warning is noticeable.

***Telegram Bot*:** The Telegram bot serves as the communication interface between the system and the user. Upon receiving the alert from the NodeMCU ESP8266, it sends a notification to the user's internet-enabled device. This ensures that the user is promptly informed about the gas leak, regardless of their location.

***User*:** The user receives the notification from the Telegram bot and is responsible for taking appropriate action. This might include contacting emergency services, evacuating the area, or manually shutting off the gas supply if the automatic mechanism fails.

***Power Cutting Mechanism*:** The power cutting mechanism is triggered by the NodeMCU ESP8266 to shut off the gas supply. This automatic response is crucial for preventing the accumulation of gas, thereby reducing the risk of explosions or fires.

**Sequence of Interactions:**

1. ***Gas Detection:***

The gas sensor continuously monitors the environment for the presence of gas. When it detects gas, it sends an immediate signal to the NodeMCU ESP8266.

1. ***Signal Processing:***

The NodeMCU ESP8266 receives the signal from the gas sensor and processes it. This microcontroller is programmed to recognize the gas detection signal and initiate the next steps in the sequence to ensure safety.

1. ***Activation of Alerts:***

**Activating the Buzzer:** The NodeMCU ESP8266 sends a signal to the buzzer to activate it. The buzzer emits a loud sound to alert anyone nearby of the gas leak. This immediate audible alert is crucial for quickly drawing attention to the potential danger.

**Activating the LED Indicator:** Simultaneously, the NodeMCU ESP8266 sends a signal to the LED indicator to turn it on. The LED provides a visual alert, which is especially useful in noisy environments where the buzzer might not be heard clearly.

1. ***Sending Notifications:***

**Alert to Telegram Bot:** The NodeMCU ESP8266 sends an alert to the Telegram bot, which is integrated into the system for remote notifications. The alert includes details about the detected gas leak.

**Notification to User:** The Telegram bot processes the alert and sends a notification to the user’s device. This notification ensures that the user, regardless of their location, is informed about the gas leak in real-time. The notification includes crucial details such as the location and time of the gas detection.

1. ***Automatic Gas Supply Shut-Off:***

The NodeMCU ESP8266 triggers the power cutting mechanism, which is designed to automatically shut off the gas supply. This automatic shut-off is a critical safety measure to prevent further gas leakage and potential hazards. The mechanism is activated promptly to mitigate any risk of explosion or fire.

1. ***User Actions:***

**Receiving the Notification:** The user receives the notification on their internet-enabled device. The prompt notification allows the user to quickly become aware of the gas leak.

**Taking Necessary Actions:** Upon receiving the notification, the user takes appropriate actions. These actions may include contacting emergency services, evacuating the premises, and manually verifying the gas shut-off if necessary. The user’s response is crucial for ensuring the safety of individuals in the vicinity and preventing any potential disaster.

1. **Continued Monitoring and Safety Measures:**

The gas sensor continues to monitor the environment even after the initial detection and response. This ongoing monitoring ensures that if the gas leak persists or reoccurs, the system will repeat the sequence of interactions to maintain safety. Maintenance personnel may be dispatched to check and repair any system components as needed, ensuring the system's reliability and accuracy over time.

**Notable Interactions:**

**Simultaneous Alerts:**

The NodeMCU ESP8266 simultaneously activates the buzzer and LED indicator to ensure immediate and effective alerting through both audible and visual means. This redundancy ensures that the alert is noticeable in various conditions.

**Automated Response:**

The automatic shut-off of the gas supply by the NodeMCU ESP8266 is a critical safety feature that operates independently of user actions, ensuring immediate hazard mitigation. This automated response reduces the time lag between gas detection and hazard prevention.

**Real-Time Notifications:**

The integration of the Telegram bot ensures that notifications are sent in real-time, enabling the user to take prompt actions regardless of their physical proximity to the system. This remote alerting system is crucial for ensuring timely responses in emergency situations.

**Continuous Monitoring:**

The gas sensor continuously monitors gas levels, providing real-time data to the NodeMCU ESP8266, which processes and responds to potential hazards dynamically. This continuous monitoring ensures that the system remains vigilant and responsive at all times.

**Alternative Path:**

If the automatic shut-off mechanism fails, the system's design ensures that the user is still

notified and can take manual actions to ensure safety. This alternative path provides an

additional layer of security and reliability

**5.3 Collaboration diagram**

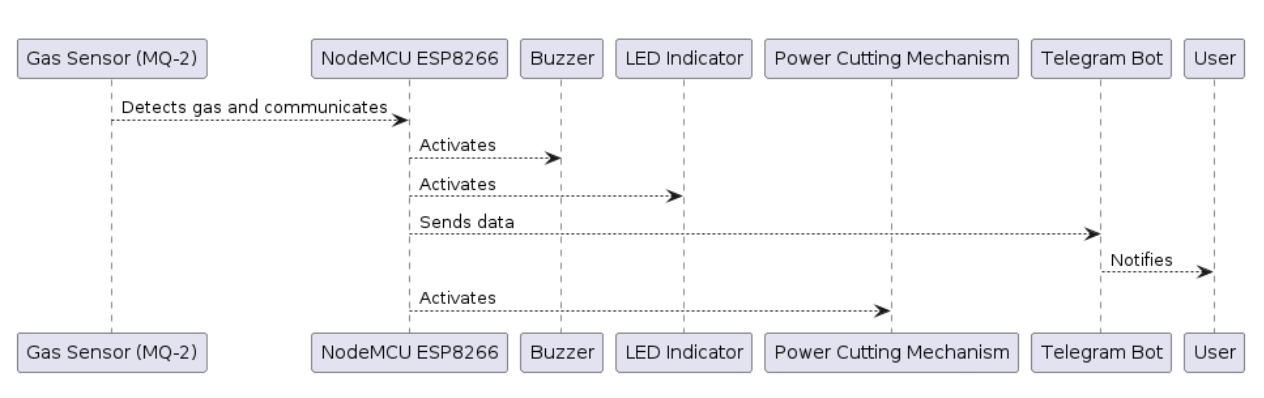
****

Figure 5: Collaboration Diagram

**5.3.1 Actors and System Components:**

**Gas Sensor (MQ-2):** The Gas Sensor (MQ-2) detects the presence and concentration of various gases, such as LPG, methane, and smoke. It continuously monitors the environment and sends real-time gas level data to the NodeMCU ESP8266. Typically installed in areas where gas leaks are a potential hazard, like kitchens or industrial sites, it ensures that any gas presence is promptly communicated for safety measures.

**NodeMCU ESP8266:** The NodeMCU ESP8266 acts as the central processing unit for the system, receiving gas data from the sensor, processing this data, and taking necessary actions. Equipped with Wi-Fi, it can send notifications over the internet. It determines whether the gas concentration exceeds predefined safety thresholds and activates safety mechanisms accordingly.

**Buzzer**: The Buzzer provides an audible alarm to alert people in the immediate vicinity of a gas leak. Triggered by the NodeMCU ESP8266 when gas levels are detected above the safety threshold, it ensures immediate attention to the gas leak, allowing people to evacuate or take necessary actions to mitigate the leak.

**LED Indicator:** The LED Indicator provides a visual alert to indicate the presence of a gas leak. Triggered by the NodeMCU ESP8266 in conjunction with the buzzer, it serves as a silent alert mechanism for environments where loud noises might not be ideal or might be missed.

**Power Cutting Mechanism:** The Power Cutting Mechanism cuts off the power supply to electrical devices to prevent potential ignition sources in the presence of gas. Engaged by the NodeMCU ESP8266 when dangerous gas levels are detected, it reduces the risk of fire or explosions by eliminating electrical sparks or heat sources.

**Telegram Bot:** The Telegram Bot is a messaging service bot that communicates with the user through the Telegram app. It receives real-time data from the NodeMCU ESP8266 and sends notifications to the user, allowing the user to receive immediate alerts and possibly control the system remotely via commands sent through the Telegram app.

**User:** The User is the end recipient of notifications and alerts sent by the system. The user takes necessary actions based on the alerts, such as evacuating the area, contacting emergency services, or performing safety checks. They receive notifications via the Telegram Bot, ensuring they are informed regardless of their location.

* + 1. **Sequence of Interactions:**

1. **Detect Gas and Communicate**: The Gas Sensor (MQ-2) continuously senses the air for gas particles. When it detects gas concentrations above a certain threshold, it sends a signal to the NodeMCU ESP8266 through analog or digital signals representing the concentration of detected gas.
2. **Activate Buzzer**: Upon receiving data indicating high gas levels, the NodeMCU ESP8266 processes this information and activates the Buzzer. The activation is immediate to ensure rapid alerting of individuals in the vicinity. The Buzzer emits a loud, continuous sound to draw attention to the gas leak situation.
3. **Activate LED Indicator**: Along with the Buzzer, the NodeMCU ESP8266 also activates the LED Indicator. The LED lights up or flashes in a distinct pattern to indicate the severity of the gas leak. This provides an additional layer of alert for those who may not hear the buzzer or for environments where noise levels are already high.
4. **Send Data to Telegram Bot**: The NodeMCU ESP8266 formats the gas detection data and sends it via Wi-Fi to the Telegram Bot. The message includes details such as gas concentration levels, location, and time of detection. This utilizes the internet connection to ensure that the user receives notifications no matter where they are.
5. **Activate Power Cutting Mechanism**: If the gas concentration exceeds a critical level, the NodeMCU ESP8266 triggers the Power Cutting Mechanism. This action cuts off the power to prevent any electrical sources from causing an explosion or fire. The power cut-off system is integrated into the main power supply line or specific high-risk devices.
6. **Notify User**: The Telegram Bot sends an immediate notification to the user’s mobile device. The alert includes comprehensive information about the gas leak, like the kind of gas detected, concentration, and safety instructions. This enables the owner to take prompt action, such as vacating the premises, shutting off gas valves, or contacting emergency services.

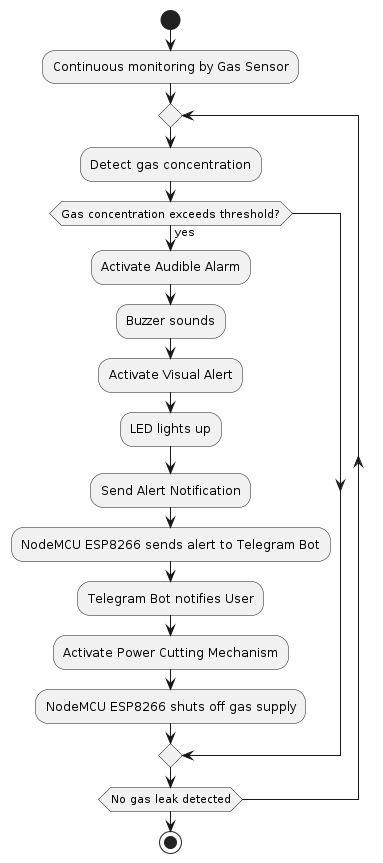
**5.4 Activity diagram**

Figure 6: Activity Diagram

**5.4.1 Actors and System Components:**

**Gas Sensor (MQ-2):** The Gas Sensor (MQ-2) is in charge of identifying the presence and concentration of various gases, such as LPG, methane, and smoke. It continuously monitors the environment and sends real-time gas level data to the NodeMCU ESP8266. These sensors are typically installed in areas where gas leaks are a potential hazard, such as kitchens or industrial sites, ensuring prompt detection and communication of gas presence for safety measures.

**NodeMCU ESP8266:** The NodeMCU ESP8266 acts as the central processing unit for the system. It receives gas data from the sensor, processes this data, and takes necessary actions. Equipped with Wi-Fi, it can send notifications over the internet and determines whether the gas concentration exceeds predefined safety thresholds, activating safety mechanisms accordingly.

**Buzzer:** The Buzzer provides an audible alarm to alert people in the immediate vicinity of a gas leak. Triggered by the NodeMCU ESP8266 when gas levels are detected above the safety threshold, it ensures immediate attention to the gas leak, allowing people to evacuate or take necessary actions to mitigate the leak.

**LED Indicator:** The LED Indicator provides a visual alert to indicate the presence of a gas leak. Triggered by the NodeMCU ESP8266 in conjunction with the buzzer, it serves as a silent alert mechanism for environments where loud noises might not be ideal or might be missed.

**Power Cutting Mechanism:** The Power Cutting Mechanism cuts off the power supply to electrical devices to prevent potential ignition sources in the presence of gas. Engaged by the NodeMCU ESP8266 when dangerous gas levels are detected, it reduces the risk of fire or explosions by eliminating electrical sparks or heat sources.

**Telegram Bot:** The Telegram Bot is a messaging service bot that communicates with the user through the Telegram app. It receives real-time data from the NodeMCU ESP8266 and sends notifications to the user, allowing the user to receive immediate alerts and possibly control the system remotely via commands sent through the Telegram app.

**User:** The User is the end recipient of notifications and alerts sent by the system. The user takes necessary actions based on the alerts, such as evacuating the area, contacting emergency services, or performing safety checks. They receive notifications via the Telegram Bot, ensuring they are informed regardless of their location.

* + 1. **Sequence of Interactions:**

1. **Continuous Monitoring**: The Gas Sensor (MQ-2) continuously monitors the environment for gas particles.
2. **Detect Gas Concentration**: When gas is detected, the Gas Sensor sends the concentration data to the NodeMCU ESP8266.no
3. **Evaluate Gas Concentration**: The NodeMCU ESP8266 checks if the gas concentration exceeds the predefined safety threshold.
4. **Activate Audible Alarm:** If the gas concentration exceeds the threshold, the NodeMCU ESP8266 activates the Buzzer, which sounds an audible alarm.
5. **Activate Visual Alert:** Concurrently, the NodeMCU ESP8266 activates the LED Indicator, which lights up to provide a visual alert.
6. **Send Alert Notification:** The NodeMCU ESP8266 sends the gas detection data to the Telegram Bot.
7. **Telegram Bot Notification:** The Telegram Bot sends an alert notification to the user’s mobile device, detailing the gas leak and safety instructions.
8. **Activate Power Cutting Mechanism:** The NodeMCU ESP8266 triggers the Power Cutting Mechanism, cutting off the power supply to prevent potential ignition sources.
9. **Monitor for No Gas Leak:** The system returns to monitoring mode if no gas leak is detected after the safety measures are executed.
10. **IMPLEMENTATION**

**6.1 Screenshot**

****

Figure 7: Miniature model of Home

The implementation of a gas leakage detection and prevention system involves several key steps: selecting and connecting appropriate hardware components, developing and deploying software for data collection and processing, and establishing communication between the system and a remote user via a Telegram bot.

**Setting Up the Hardware**

**Connect Gas Sensor to NodeMCU ESP8266:**

* Connect the gas sensor (e.g., MQ-2) to the analog pin (A0) of the NodeMCU ESP8266.
* Connect the VCC pin of the gas sensor to the 3.3V pin on the NodeMCU.
* Connect the GND pin of the gas sensor to the GND pin on the NodeMCU.

**Connect Relays for Alarm and Safety Devices:**

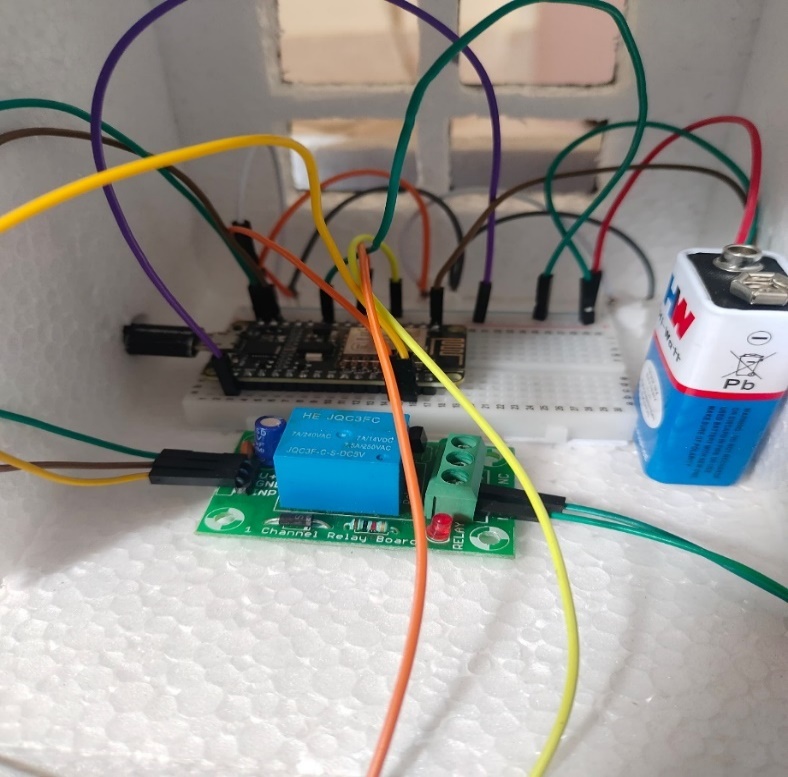
* Connect the inputs of the relays to digital pins D0, D1, and D2 on the NodeMCU ESP8266.
* Relay 1 (D0) will be linked to the alarm light.
* Relay 2 (D1) will be linked to the siren.
* Relay 3 (D2) will be linked to the fan or power-cutting mechanism.
* Ensure relays are properly powered and set up to manage the connected devices.

Figure 7: Connection of the model

**Power Supply:**

* Connect a suitable power supply to the NodeMCU ESP8266 plus additional elements. Ensure a power source that can supply enough current for all connected devices.



Figure 9: Components Arranged inside

**Setting Up the Software**

**Install Arduino IDE:**

* Download and install the Arduino IDE from the official Arduino website.
* Install the necessary libraries: ESP8266WiFi, WiFiClientSecure, UniversalTelegramBot, and ArduinoJson.

**Write the Code:**

Develop code to monitor gas levels, activate safety devices, and send notifications via Telegram. Below is an example code snippet:

#include <ESP8266WiFi.h>

#include <WiFiClientSecure.h>

#include <UniversalTelegramBot.h>

#include <ArduinoJson.h>"

int relayInput1 = 16; // D0

int relayInput2 = 5; // D1

int relayInput3 = 4; // D2

// Initialize Telegram BOT

#define CHAT\_ID "6560079822"

X509List cert(TELEGRAM\_CERTIFICATE\_ROOT);

WiFiClientSecure client;

UniversalTelegramBot bot(BOTtoken, client);

void setup() {

Serial.begin(9600);

pinMode(relayInput1, OUTPUT); // initialize pin as OUTPUT

pinMode(relayInput2, OUTPUT); // initialize pin as OUTPUT

pinMode(relayInput3, OUTPUT); // initialize pin as OUTPUT

configTime(0, 0, "pool.ntp.org"); // get UTC time via NTP

client.setTrustAnchors(&cert); // Add root certificate for api.telegram.org

}

Serial.println("");

Serial.println("WiFi connected");

Serial.print("IP address: ");

Serial.println(WiFi.localIP());

bot.sendMessage(CHAT\_ID, "Bot started up", "");

bot.sendMessage(CHAT\_ID, "Gas level monitoring", "");

}

void loop() {

Serial.println("Gas level monitoring is on");

Serial.print("Gas Level : ");

Serial.println(analogRead(A0));

if(analogRead(A0) > 300) {

Serial.print("Gas Is Leaking, Plz take Care ");

digitalWrite(relayInput1, HIGH); // Light On

digitalWrite(relayInput2, HIGH); // Siren On

digitalWrite(relayInput3, HIGH); // Fan On

bot.sendMessage(CHAT\_ID, "Gas Is Leaking, please take care", "");

delay(5000);

} else {

digitalWrite(relayInput1, LOW); // Light Off

digitalWrite(relayInput2, LOW); // Siren Off

digitalWrite(relayInput3, LOW); // Fan Off

}

delay(1000);

}

**Deploying the Code**

**Upload the Code:**

* Connect the NodeMCU ESP8266 to your computer via USB.
* Select the correct board and port in the Arduino IDE.
* Click on the upload button to compile and upload the code to the NodeMCU ESP8266.

**Monitor Serial Output:**

* Open the Serial Monitor in the Arduino IDE to view the gas level measurements and Wi-Fi connection status.

**Testing and Calibration:**

**Test the System:**

* Verify the gas detection and alert functionality by exposing the sensor to a gas source.
* Ensure the system responds correctly when gas concentration exceeds the threshold (300).

**Calibrate Sensors:**

* Adjust the sensitivity of the gas sensor to ensure accurate gas detection.
* Regularly test the sensor and adjust parameters in the code if necessary.

**Perform Maintenance:**

* Periodically check the sensor and other components for proper functioning.
* Replace faulty components and update the firmware as needed.

1. **SOFTWARE TESTING**

System testing is a crucial phase in the development of the gas leakage detection and prevention system. It aims to identify and correct errors, ensuring that the system satisfies its predetermined specifications and user expectations. Testing involves evaluating the functionality of the gas sensor, NodeMCU ESP8266, relays, and communication components to guarantee the system operates without critical failures. Various types of tests address specific testing requirements and play an essential part in the overall testing strategy.

**7.2 Types of Testing**

**7.2.1 Unit Testing**

It involves designing test cases to validate the logic of the internal program of individual components, like the gas sensor and relays. It guarantees that the sensor accurately detects gas levels, the relays respond correctly to control signals, and the NodeMCU executes its code as expected. This testing is conducted following the conclusion of each unit and earlier than integration. It relies on the tester's comprehension of the design of the hardware and software and is considered disruptive. Unit tests carry out fundamental checks at the component level to make certain that every distinct route of the gas detection and alert procedure operates with accuracy according to documented specifications.

**7.2.2 Integration Testing**

Integration testing focuses on testing the integrated components of the gas leakage detection system to ensure they function together as a cohesive unit. It verifies that the gas sensor, NodeMCU, relays, and the Telegram bot work correctly and consistently when integrated. This testing aims to expose problems that arise from the combination of components, such as communication issues between the NodeMCU and the relays or between the NodeMCU and the Telegram bot.

**7.2.3 Functional Testing**

Functional tests offer methodical proofs that the gas leakage detection system functions as outlined by the project requirements. This testing focuses on validating the detection of gas leaks, activation of alarms, and sending notifications via Telegram. Functional testing guarantees that the system correctly identifies gas leaks, activates the appropriate safety measures (lights, sirens, fans), and notifies the user through the Telegram bot.

**7.2.4 System Testing**

System testing guarantees that all integrated gas leakage detection system fulfils its requirements. This testing involves configuring the framework to achieve recognized and dependable results, such as simulating a gas leak to confirm that the sensor detects it and triggers the appropriate responses. It emphasizes pre-defined process links and integration points, ensuring the system functions as a whole and reacts appropriately to gas leaks.

**7.2.5 White Box Testing**

White box testing involves evaluating the setup with awareness of its internal workings, structure, and code. This testing examines regions that a black box cannot access level, allowing for a thorough examination of the software's internal logic, such as the decision-making process within the NodeMCU's code when a gas leak is detected.

**7.2.6 Black Box Testing**

Black box testing treats the system as a "black box," where the tester does not have knowledge of the internal workings, structure, or code of the testable module. Tests are based on input (gas concentration) and output (activation of alarms and notifications) Without taking into account the internal logic within the system. This testing is derived from the system's specifications and requirements documents.

**Real-Time Data Handling**

Given that the system relies on real-time information from the gas sensor, it is crucial to test the system's ability to handle and process data continuously without lag or errors. This ensures the system can provide timely alerts and activate safety measures promptly.

**Scalability Testing**

Ensure the system can handle varying gas concentrations and potential multiple sources of gas leaks, verifying that it can maintain performance without degradation under different environmental conditions.

**Reliability****and****Fault****Tolerance**

Test the system's reliability and ability to tolerate faults, such as sensor failures or communication disruptions. Ensure the system can continue to function correctly or recover gracefully, maintaining its ability to detect gas leaks and alert users.

**Security Testing**

Verify that the system is secure from unauthorized access, ensuring that only authorized users can receive alerts and control the system. Maintain data integrity, particularly given the critical nature of gas leak detection and prevention.

**Usability Testing**

Evaluate the user interface for ease of use by the end-users, ensuring that the system is intuitive and provides clear, actionable information. This includes testing the Telegram bot interface to ensure users receive timely and understandable alerts.

**Test Cases for Gas Leakage Detection and Monitoring System**

|  |  |  |  |
| --- | --- | --- | --- |
| **Section** | **Test Case ID** | **Test Case** | **Expected Results** |
| Unit Testing | 1 | Gas Sensor Data Handling | Function provides precise gas concentration readings with a tolerable error margin. |
| Unit Testing | 2 | Telegram Bot Message Sending | Telegram bot sends messages correctly when triggered. |
| Unit Testing | 3 | Buzzer Activation Logic | The buzzer correctly sounds when a gas leak is found. |
| Unit Testing | 4 | LED Indicator Logic | When a gas leak is found, the LED becomes red; when none is found, it turns green. |
| Unit Testing | 5 | Exhaust Fan Activation Logic | Exhaust fan activates correctly when a gas leak is detected. |
| Integration Testing | 6 | Sensor and Telegram Bot Integration | When the gas concentration rises beyond the threshold, a message is sent by the Telegram bot. |
| Integration Testing | 7 | Sensor and Buzzer Integration | Buzzer sounds when gas concentration exceeds the threshold. |
| Integration Testing | 8 | Sensor and LED Integration | When the gas concentration is over the threshold, the LED lights red; when it is below the threshold, the LED turns green. |
| Integration Testing | 9 | Sensor and Exhaust Fan Integration | Exhaust fan activates when gas concentration exceeds the threshold. |
| Functional Testing | 10 | Complete Alert Process | Accurately, the system detects a gas leak, sounds an alarm, activates an exhaust fan, switches the LED to red, and sends a Telegram message. |
| Functional Testing | 11 | Power Shut-off Mechanism | Power supply is cut off when gas concentration exceeds the threshold. |
| Functional Testing | 12 | Exhaust Fan and LED Deactivation | When the gas concentration falls below the threshold, the exhaust fan deactivates and the LED turns green. |
| Performance Testing | 13 | Response Time Under Load | System handles multiple gas leak detections without significant delays. |
| Performance Testing | 14 | Long Duration Stability | The system runs continuously for a long time without experiencing serious performance problems or crashes. |
| Error Handling Testing | 15 | Handling Invalid Sensor Data | The system continues to function properly even when it receives erroneous data. |
| Error Handling Testing | 16 | Recovery from Power Failure | After a power outage, the system correctly initializes and continues to function normally. |
| Error Handling Testing | 17 | Telegram Bot Connection Loss | The system sends messages again until it succeeds or logs the error correctly. |
| Security Testing | 18 | Unauthorized Access Attempts | The system stops illegal control activities and logs attempts at unauthorized access. |
| Security Testing | 19 | Data Transmission Security | Sent data to remote monitoring systems or the cloud is encrypted to prevent interception. |

* 1. **CONCLUSION**

In conclusion, the installation of a gas leak detection system and alerting system utilizing the MQ-2 sensor represents a significant advancement in enhancing safety measures within residential, commercial, and industrial environments. By leveraging the capabilities of the MQ-2 sensor, coupled with proactive alerting mechanisms and integration with internet-enabled devices, the system offers a reliable means of identifying and responding to gas leakage incidents promptly. The MQ-2 sensor is particularly effective due to its high sensitivity to various gases such as methane, propane, and butane, making it versatile for detecting leaks from different gas sources. The sensor's integration with the NodeMCU ESP8266 microcontroller facilitates real-time monitoring and immediate response through IoT-enabled alerts, such as notifications sent via the Telegram bot. This real-time capability ensures that users are quickly informed of any potential gas hazards, allowing for prompt action.Through the development of a user-friendly interface and proactive risk mitigation measures such as automatic power cutting upon detection of gas leakage, the system empowers users to take swift and effective action to prevent potential hazards or accidents. The inclusion of visual (LED) and audible (buzzer) alarms further enhances the system's effectiveness by providing immediate, local alerts in addition to remote notifications. Furthermore, the system’s ability to automatically shut off the gas supply through relays adds a critical layer of safety by mitigating the risk of fire or explosion. The system's scalability, reliability, and adherence to safety standards ensure its suitability for various applications and environments. It can be easily adapted to different settings, whether in small residential homes, large industrial facilities, or commercial establishments, due to its flexible design and configurable parameters. The application of Wi-Fi for communication also means that the system able to be kept an eye on and managed remotely, providing convenience and peace of mind to users. Despite its numerous advantages, further research and development are warranted to address challenges such as sensor calibration, reliability in real-world conditions, integration with existing infrastructure, and user acceptance. Sensor calibration is crucial for ensuring accuracy and minimizing false alarms, while testing under varied environmental conditions can enhance the system's robustness and reliability. Integration with existing building management systems and infrastructures would streamline deployment and maximize the utility of the system. Additionally, addressing user acceptance involves making the system intuitive and simple to use, thereby encouraging widespread adoption. Further enhancements could also include incorporating advanced data analytics and machine learning algorithms to predict potential gas leaks based on historical information and patterns. This predictive capability could provide an additional layer of safety by identifying vulnerabilities before they result in actual leaks. Furthermore, expanding the system's compatibility with other IoT devices and smart home technologies could create a more comprehensive safety network within residential and commercial properties.

* 1. **FUTURE ENHANCEMENTS**

Future enhancements for the detection and notification of gas leaks system utilizing the MQ-2 sensor could significantly improve its performance, reliability, and user experience. One critical area of development is extending battery life and implementing advanced power management. By integrating a larger, rechargeable battery and a power management system that alerts users when the battery is low, the system's operational duration and reliability can be enhanced. Additionally, incorporating sensors capable of measuring gas concentration levels would provide users with detailed information about the severity of a gas leak, allowing for more informed decision-making.

Enhancing the user interface and alert mechanisms is another essential improvement. Developing a more user-friendly interface with clear visual and auditory alerts, and integrating the system with mobile applications for remote monitoring, would significantly improve user interaction and ensure alerts are effective and actionable. Integration with smart home ecosystems such as Google Home or Amazon Alexa could create a more comprehensive safety and automation solution. This would enable coordinated responses, such as automatically shutting off gas valves and activating venting systems when a leak is detected.

Implementing machine learning algorithms for predictive analytics could enhance the system's accuracy and provide early warnings by identifying patterns that precede gas leaks. This predictive capability can help prevent potential leaks before they become critical. Additionally, focusing on cost reduction and miniaturization of components can make the system more affordable and accessible to a broader range of users, making it easier to install in various environments. Ensuring the system complies with the highest safety and environmental standards through thorough testing and obtaining relevant certifications will increase user confidence and promote wider adoption. By addressing these areas, the method for detecting and alerting gas leaks can include significantly improved, providing users with a more reliable, efficient, and user-friendly solution for gas leak detection and prevention.

## APPENDIX A

1. Alkhafaji, A. N., Hassan, R., & Ali, N. M. (2020). Smart Gas Leakage Detection and Alert System using IoT and Mobile Application. In 2020 11th International Conference on Information Technology (ICIT) (pp. 189-194). IEEE.
2. Alsultan, A., & Alali, A. (2018). Smart Gas Leakage Detection System using Arduino Microcontroller. Journal of Computer Science, 14(4), 461-467.
3. Bae, J. S., Lee, S. J., Lee, J. S., Park, S. W., & Bae, J. H. (2014). Real-time gas leakage detection system using wireless sensor networks. Journal of Sensor Science and Technology, 23(1), 59-64.
4. Banerjee, S., Chakraborty, T., & Roy, S. (2019). A novel design of IoT based gas leakage detection and alerting system. In 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN) (pp. 451-455). IEEE.
5. Banerjee, S., Das, S., Chatterjee, S., & Ghosh, A. (2020). A cost-effective approach for IoT-based gas leakage detection and alerting system. Journal of Ambient Intelligence and Humanized Computing, 11(4), 1447-1458.
6. Basri, M. N., Anuar, N. B., Suhaimi, M. H., & Abdullah, S. M. A. (2019). Development of Gas Leakage Detection System using MQ-7 Gas Sensor for Smart Home Application. Journal of Advanced Research in Dynamical and Control Systems, 11(1), 507-513.
7. Bhattacharya, A., & Tripathi, M. (2019). Design and development of gas leakage detection and alert system using Arduino and IoT. In 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 26-31). IEEE.
8. Chan, C. Y., & Wong, J. M. (2018). Gas Leakage Detection System using MQ-5 Gas Sensor. In 2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA) (pp. 1-5). IEEE.
9. Chen, J., & Zong, J. (2019). Research on Gas Leakage Monitoring and Control System Based on ZigBee Technology. In 2019 International Conference on Smart Grid and Electrical Automation (ICSGEA) (pp. 62-66). IEEE.
10. Chen, L., Liao, Y., Xu, H., Wang, Z., & Liu, J. (2019). Gas Leakage Detection and Alarm System Based on IoT Technology. In 2019 International Conference on Computer Science, Engineering and Applications (ICCSEA) (pp. 1-6). IEEE.
11. Dhole, P. N., & Gedam, S. S. (2019). IoT based LPG gas leakage detection and alerting system using MQ-2 and Blynk. In 2019 2nd International Conference on Computing, Communication, Control and Automation (ICCUBEA) (pp. 1-4). IEEE.
12. Dixit, S., Trivedi, K., Raval, M., & Patel, R. (2019). IoT Based Gas Leakage Detection and Alerting System. In 2019 International Conference on Computing, Communication, Control and Automation (ICCUBEA) (pp. 1-5). IEEE.
13. Dutta, S., Debnath, S., & Maitra, S. (2021). Design and development of a real-time LPG gas leakage detection and alerting system using IoT. In 2021 12th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.
14. Elmahi, S. S., Essaaidi, M., & El Hajjaji, A. (2020). Gas leakage detection system based on the internet of things and smartphones. Journal of Sensors and Actuator Networks, 9(3), 51.
15. Gaikwad, R., & Gulhane, V. (2019). Development of IoT based Gas Leakage Detection and Alerting System. In 2019 5th International Conference on Computing Communication Control and Automation (ICCUBEA) (pp. 1-5). IEEE.
16. Gao, C., & Zhu, Q. (2019). Research on Gas Leakage Detection System Based on LabVIEW. In 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT) (pp. 1299-1304). IEEE.
17. Gautam, R., & Chaudhary, S. (2018). IoT based gas leakage detection and alert system. In 2018 IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON) (pp. 1-5). IEEE.
18. Ghayyur, S. A., Abbas, M., & Khan, I. A. (2019). Gas Leakage Detection System using Arduino and IoT. In 2019 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET) (pp. 1-5). IEEE.
19. Gill, A. S., & Gill, N. S. (2020). Gas Leakage Detection and Alerting System using IoT and Cloud Computing. In 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 334-339). IEEE.
20. Goel, M., & Chhikara, R. (2020). Design and Development of Gas Leakage Detection and Alerting System using Arduino. International Journal of Advanced Computer Science and Applications, 11(7), 169-174.
21. Gupta, A., Jain, V., & Sharma, P. (2018). Gas Leakage Detection and Alerting System Using IoT. In 2018 4th International Conference on Computational Intelligence & Communication Technology (CICT) (pp. 1-6). IEEE.
22. Gupta, P., & Singh, J. (2019). IoT-based Smart Gas Leakage Detection and Alert System. In 2019 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET) (pp. 215-219). IEEE.
23. Gupta, V., Arora, D., Sharma, A., & Agarwal, R. (2020). IoT based Gas Leakage Detection and Alerting System. In 2020 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) (pp. 29-34). IEEE.
24. Harish, V., & Vinay, V. (2019). Gas Leakage Detection and Alert System using IoT. In 2019 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA) (pp. 1-5). IEEE.
25. Khamkham, A., Bahtout, A., & Eloutmani, M. (2019). Development of IoT-Based Gas Leakage Detection System Using MQ-2 Gas Sensor. In 2019 International Conference on Computer and Information Sciences (ICCIS) (pp. 1-4). IEEE.
26. Khandelwal, N., Jadhav, N., & Jadhav, S. (2019). IoT Based Gas Leakage Detection System using MQ2 Sensor. In 2019 International Conference on Communication Information and Computing Technology (ICCICT) (pp. 1-4). IEEE.
27. Khanna, A., & Baluja, S. (2018). Implementation of IoT based gas leakage detection system. In 2018 2nd International Conference on Inventive Systems and Control (ICISC) (pp. 1057-1061). IEEE.
28. Kumar, A., & Kumar, R. (2019). Design of Wireless Gas Leakage Detection and Alert System. In 2019 IEEE 5th International Conference for Convergence in Technology (I2CT) (pp. 1-5). IEEE.
29. Kumari, N., Panchal, A., & Chourasia, S. (2018). IoT Based Gas Leakage Detection and Alert System using NodeMCU. In 2018 International Conference on Computing, Power and Communication Technologies (GUCON) (pp. 129-132). IEEE.
30. Lajevardi, S. M., Al-Shourbaji, I., & Alhadidi, D. (2020). IoT-based gas leakage detection system using Raspberry Pi. Journal of Sensors and Actuator Networks, 9(4), 75

**APPENDIX B**

**Introduction**

The Gas Leakage Detection and Alert System is designed to detect and alert users about LPG gas leaks using the MQ-2 gas sensor. This system is crucial for enhancing safety in residential, commercial, and industrial settings by providing timely alerts and triggering preventive actions to mitigate the risks associated with gas leaks.

**System Operation**

**System Start-Up:** Ensure all components are properly connected as per the setup instructions. Connect the power supply to the Arduino Mega 2560 and switch on the power supply. The system will perform an initial self-check to ensure all components are operational.

**WiFi Connection:** The system will attempt to connect to the specified WiFi network using the credentials provided in the code. Upon successful connection, the system will send a "Bot started up" message to the specified Telegram chat.

**Sensor Calibration:** The MQ-2 sensor will automatically calibrate itself to establish a baseline gas concentration level. Ensure there are no gas leaks or unusual environmental conditions during this calibration period for accurate readings.

**Gas Detection:** The MQ-2 sensor continuously monitors the gas concentration in the environment. The system displays real-time gas levels on the serial monitor.

**Leak Detection and Alerts:** If the gas concentration exceeds the predefined threshold (e.g., 300 units), the system will activate a warning light (relayInput1), an alarm siren (relayInput2), and an exhaust fan (relayInput3). A notification message will be sent to the Telegram chat indicating the presence of a gas leak.

**Resetting Alerts:** If the gas concentration falls below the threshold, the system will automatically deactivate the warning light, siren, and exhaust fan.

**Remote Monitoring:** All critical alerts and status updates are sent to the specified Telegram chat. Users can monitor the system's performance and receive real-time updates remotely.

**Maintenance and Troubleshooting**

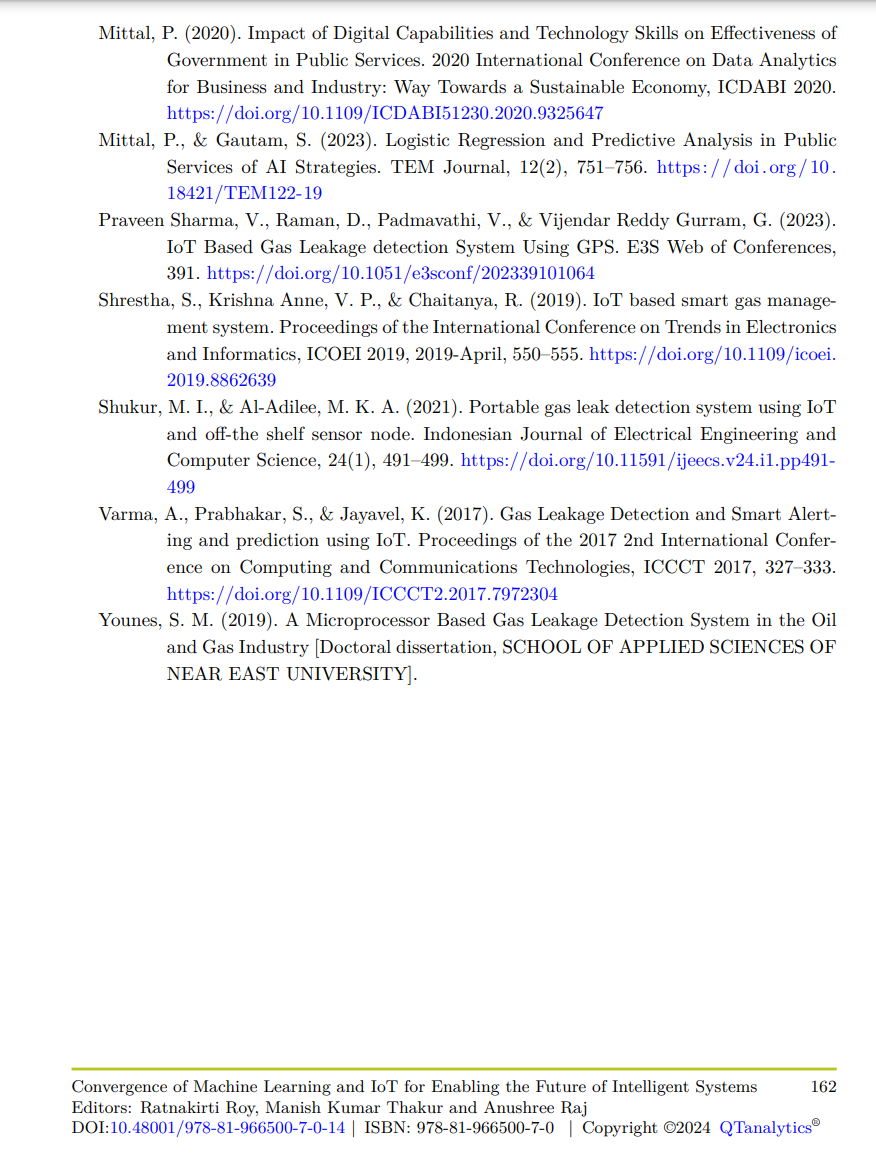
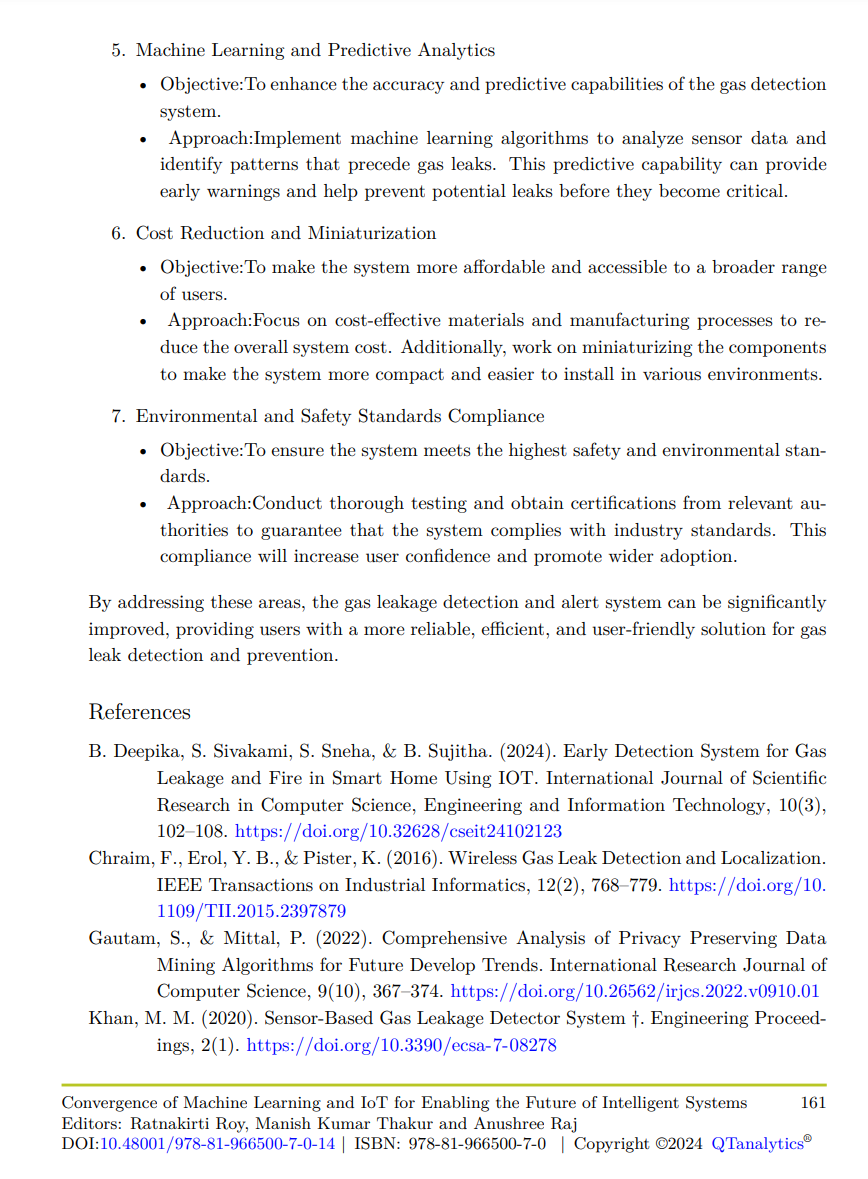
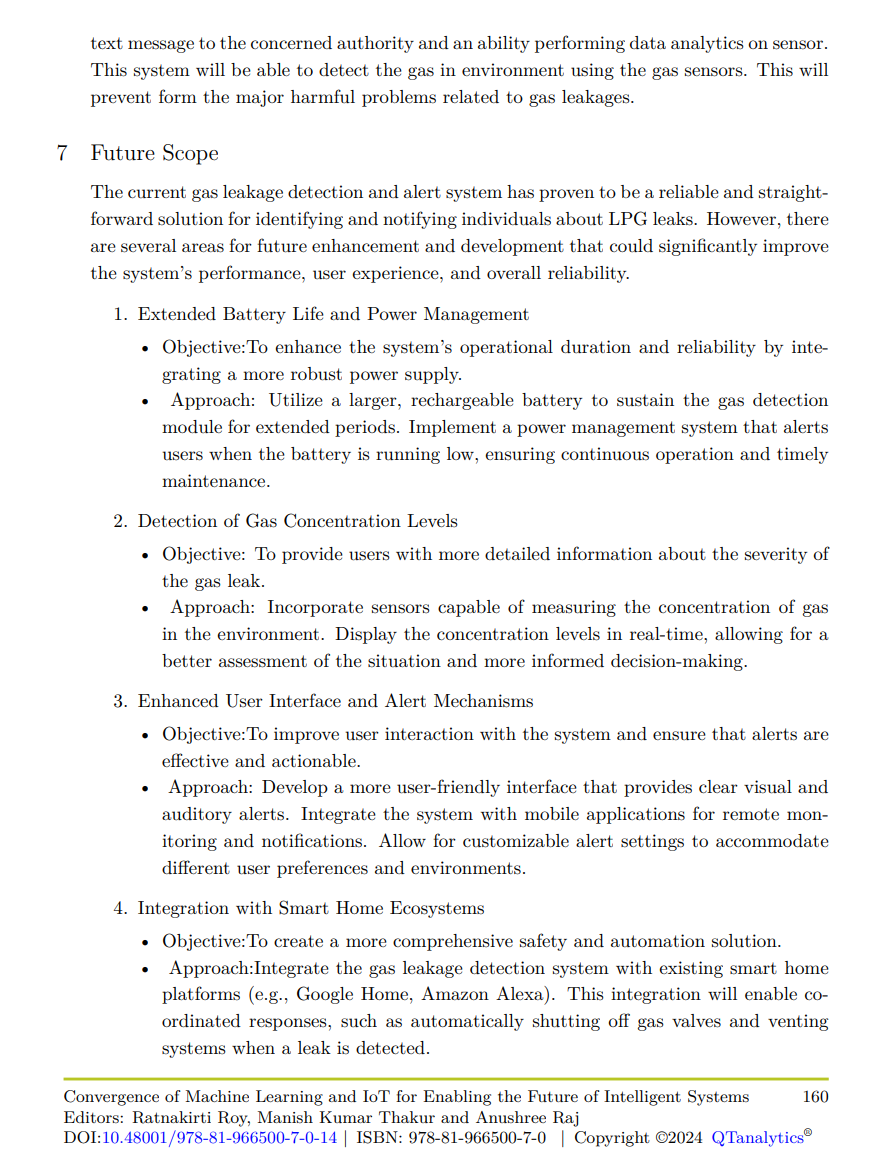
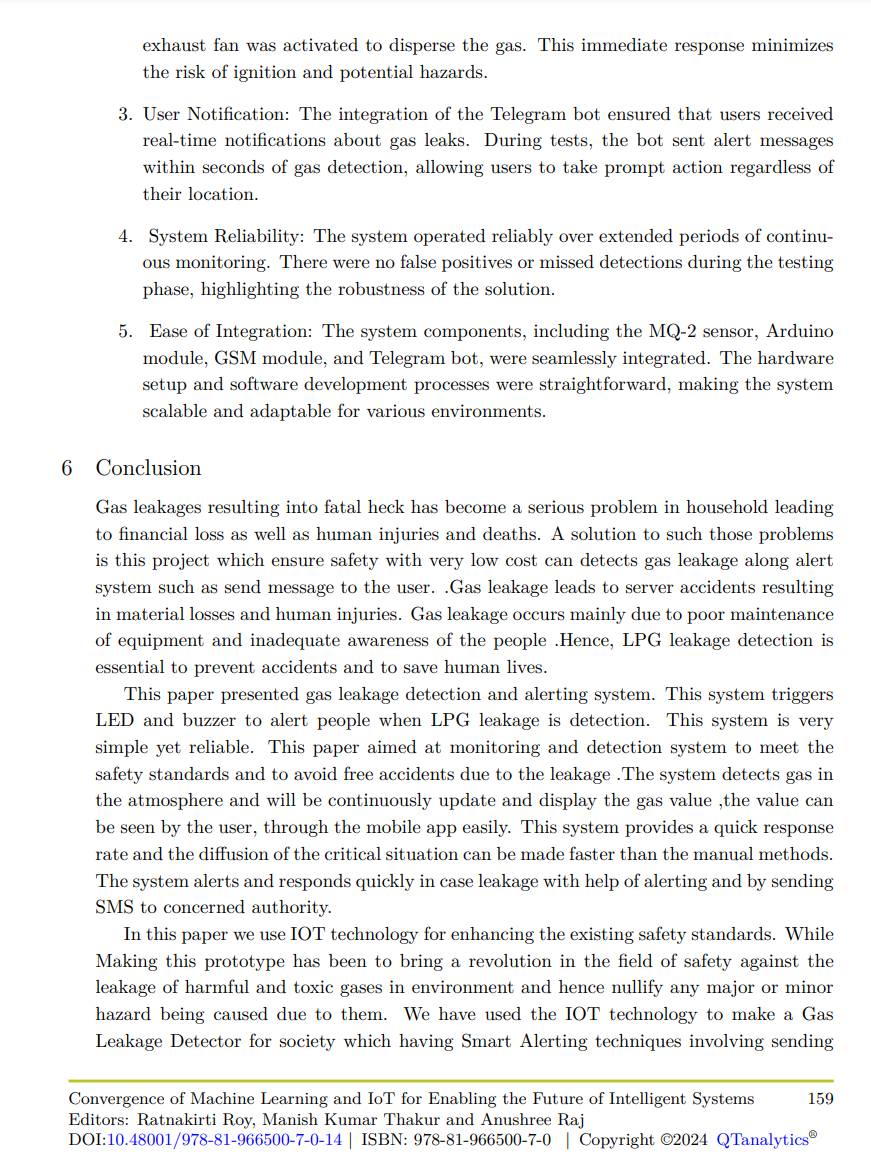
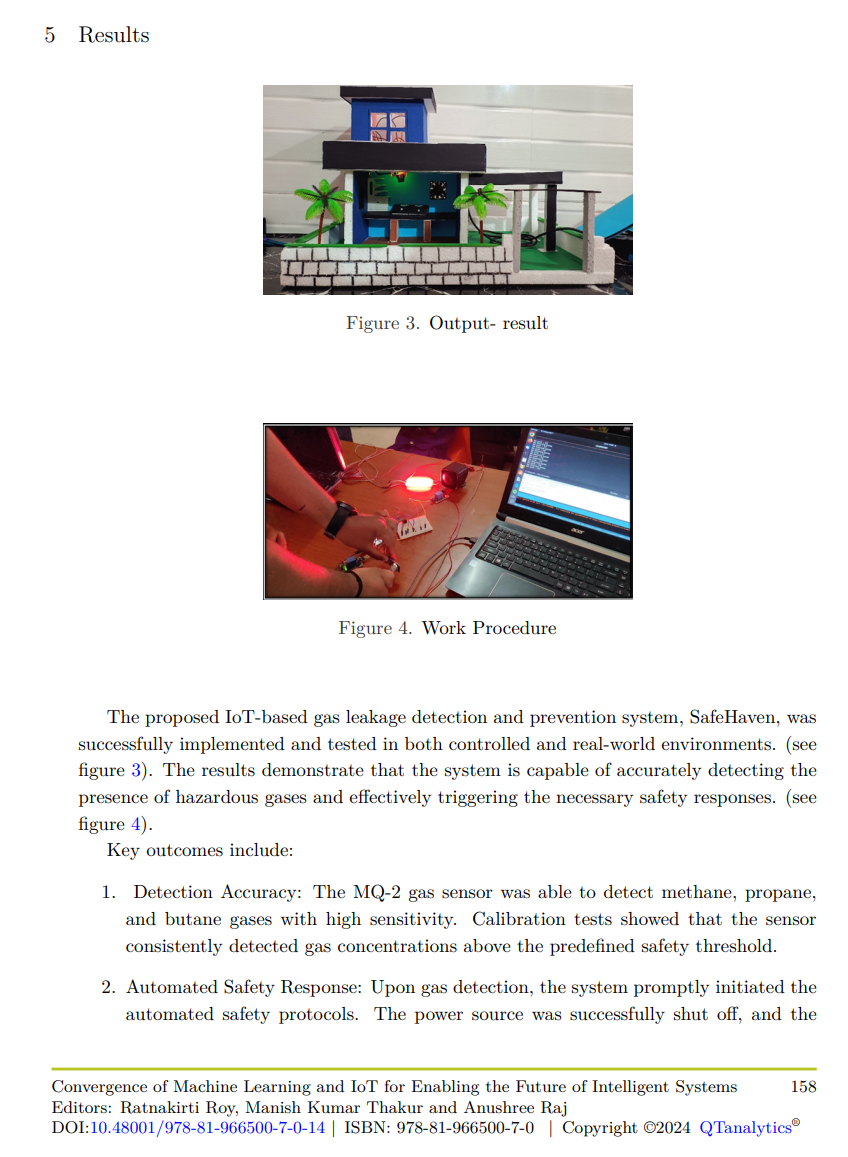
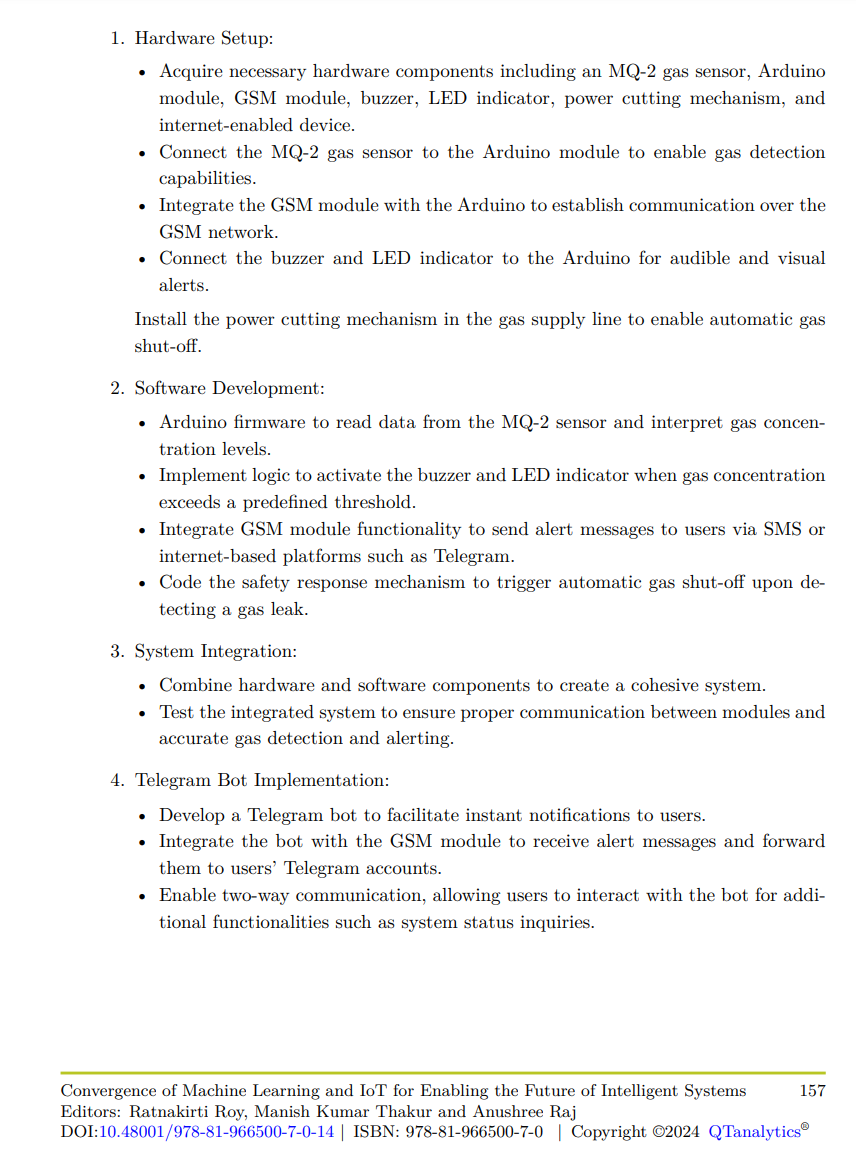
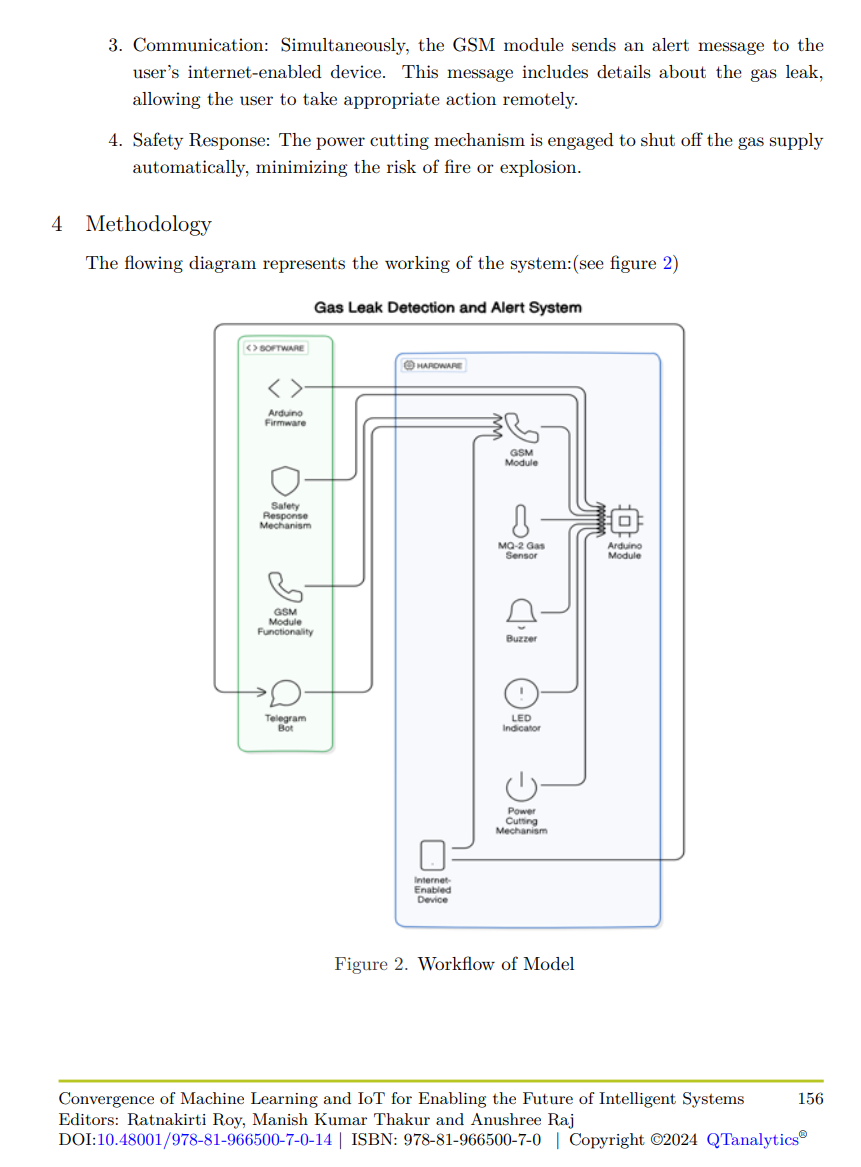
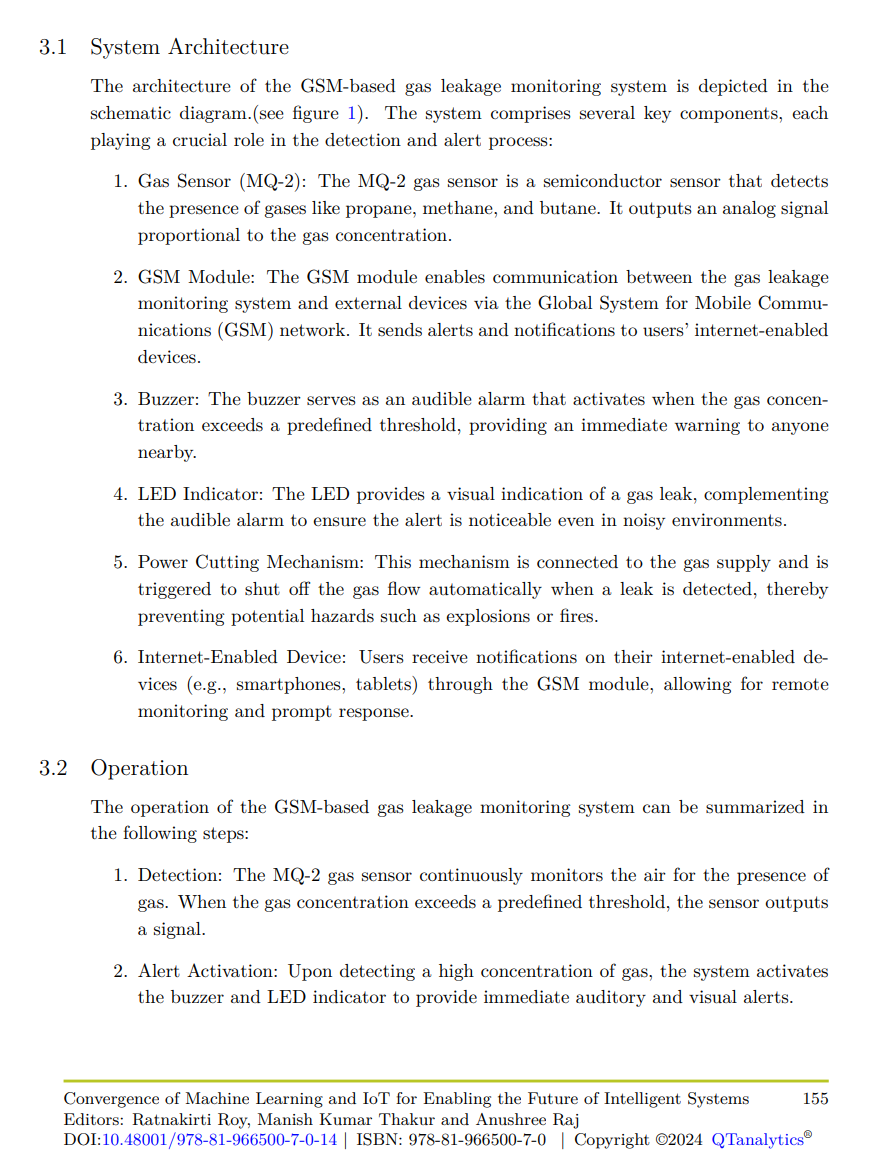
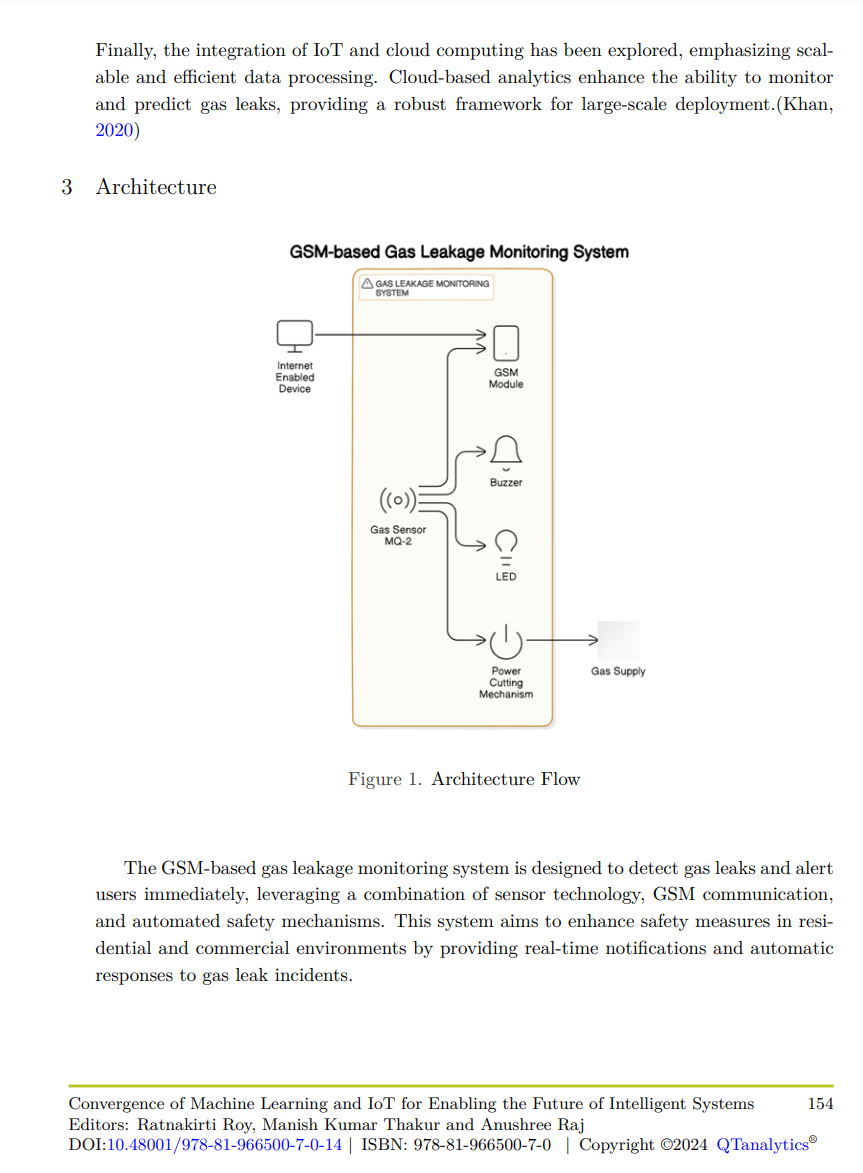
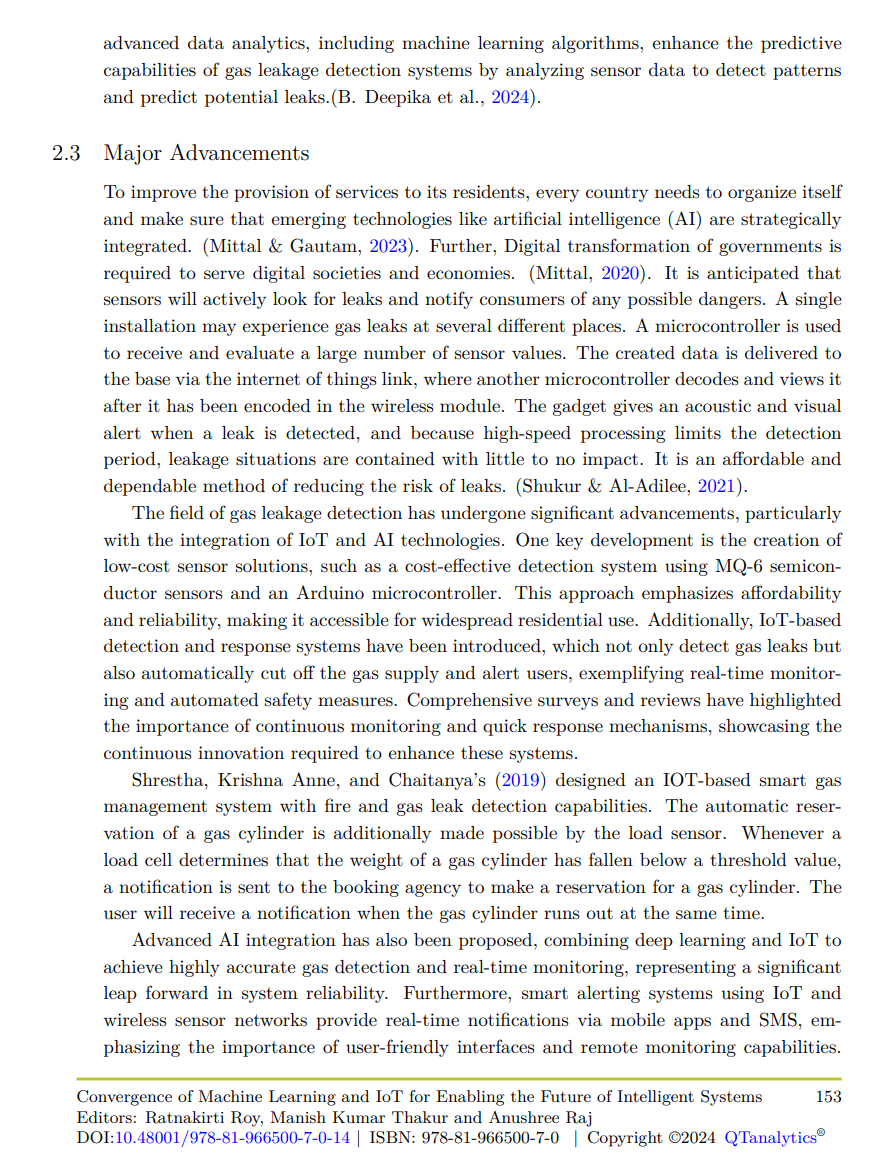
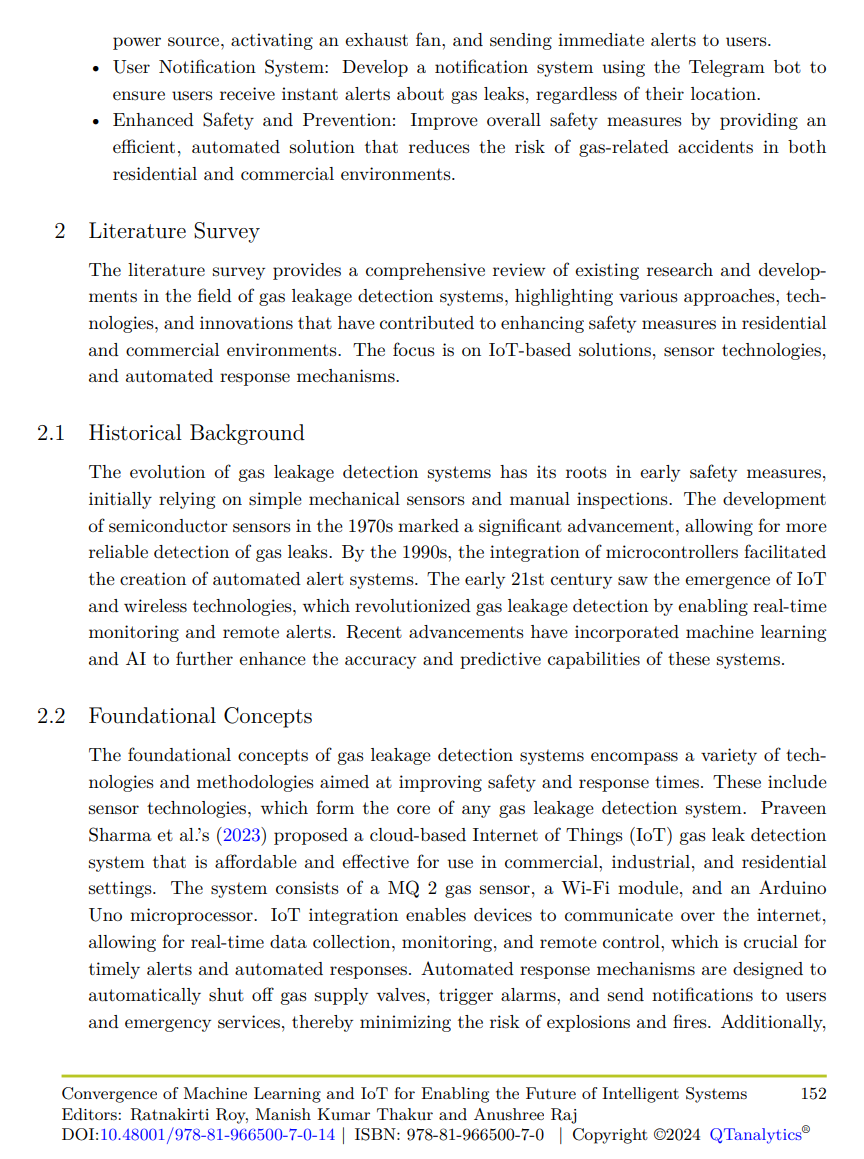
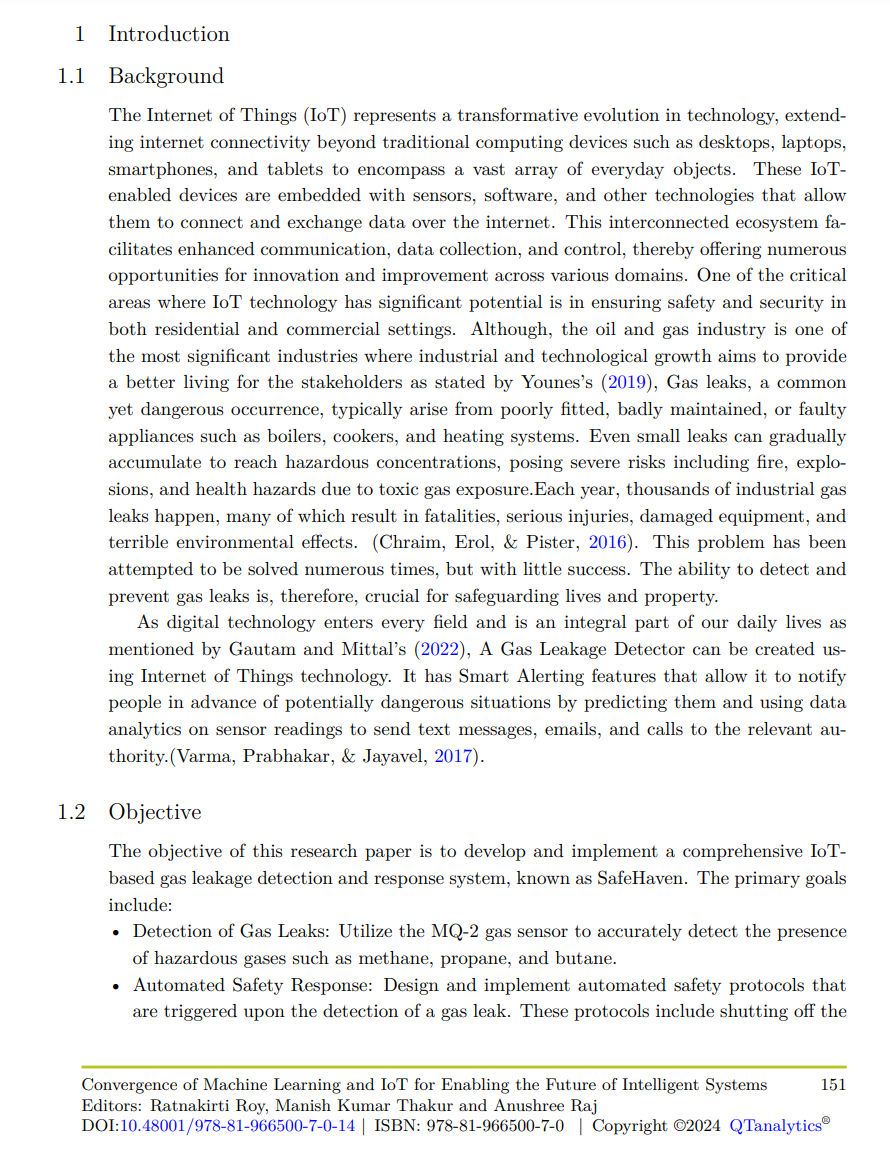
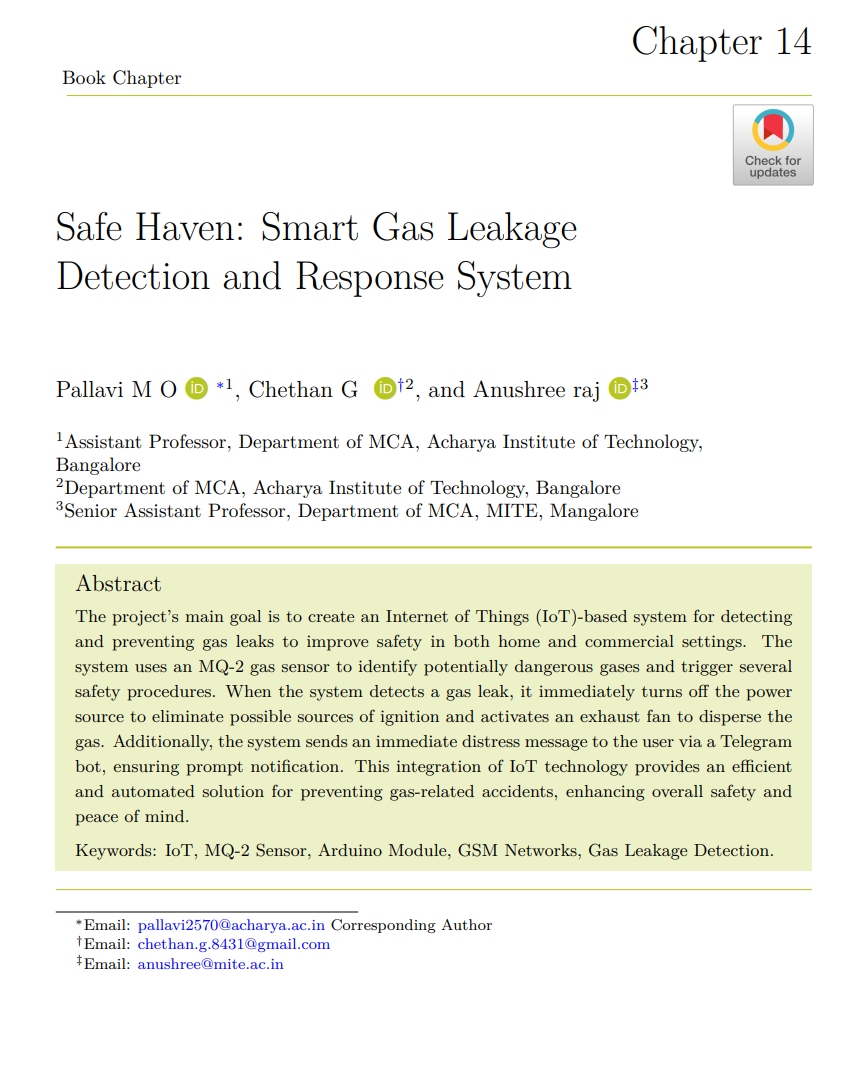
**Sensor Cleaning:** Periodically clean the MQ-2 sensor to ensure it remains free of dust and debris. Use a soft, dry cloth for cleaning. Avoid using water or any cleaning solutions that may damage the sensor.

**System Check:** Regularly perform system checks to ensure all components are functioning correctly. Monitor the serial output and Telegram notifications for any unusual activities or errors.

**Troubleshooting Common Issues:** For WiFi connection issues, verify that the WiFi credentials in the code are correct and ensure the WiFi network is operational and within range. For sensor malfunctions, check the sensor connections and ensure they are secure. Replace the sensor if it appears to be faulty or damaged. If no alerts are sent, ensure the Telegram Bot Token and Chat ID are correctly entered in the code and check the internet connection for stability.

**Safety Precautions**

Always switch off the power supply before performing any maintenance or adjustments to the system. Periodically test the system in a controlled environment to ensure it is functioning correctly. Do not expose the system components to excessive moisture or extreme temperatures.

**PUBLICATION DETAILS  
**