APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 6 No. 2 (2025) 247-264
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

Development of Web-Based IoT Smart Fire and Gas Leak Detection System

Nureen Haziqah Mohd Hairudin[1], Nayef Abdulwahab Mohammed Alduais[1]*,

[1] *Fakulti Sains Komputer dan Teknologi Maklumat, Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA*

*Corresponding Author: nayef@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2025.06.02.014

Article Info

Received: 12 June 2025
Accepted: 3 November 2025
Available online: 30 November 2025

Keywords

Development of Web-Based IoT Smart Fire and Gas Leak Detection System, Sensor, Alarms, Emergency, Gas Leak, Temperature

Abstract

Gas leaks and fires in restaurant settings are frequently unnoticed before it is too late with the emergency options being late to start rescue activities and the risks of losses to employees and patrons increasing. The common forms of detection are traditional where detections are performed manually or easy alarms, which are either have ambiguity, or not detected especially during off-shifts. This is why this project was conducted with the title of Development of Web-Based IoT Smart Fire and Gas Leak Detection System, which addresses these flaws by offering a more efficient and prompt response in a restaurant environment. A combination of Wemos D1 ESP8266, MQ-2 Gas Sensor, IR Flame Sensor, and DHT11 Temperature Sensor is implemented in the system making it monitor continuously. It is programmed using Arduino IDE, data visualization as a cloud-based solution is made using ThingSpeak whereas implementation of web-based platform is made using PHP, Laravel, and MySQL. WhatsApp notifications are real-time so that there is direct alerting. Thorough testing of the system, such as the User Acceptance Testing (UAT) proved that this system contributes a massive collaboration to safety as it diminishes the dangers linked to the fires and gas explosion. In general, the present project demonstrates how it is possible to appropriately utilize IoT technology as a means of enhancing safety in the workplaces and resolving emergency situations at high-risk areas much faster.

1. Introduction

Fire and gas leak detection in high-risk environments like restaurant kitchens is usually checked by observations and safety checks from time to time without any set standard [1]. This reliance often results in slow responses, relatively high risks, and staff and customers' elevated safety risks. The last few research works stress the urgency of implementing technology to combat these problems, lessen the chance of an accident, and limit the loss. This work presents a Smart Fire and Gas Leak Detection System which is an IoT based system to improve the capacity of hazard identification and mitigation. Using Arduino IDE, C++, PHP and cloud storage, the system interfaces fire and gas sensors to monitor the current surrounding environment. They include auto- alarms, simple notification systems and emergency instructions that solve the limitations of old-fashioned methods and guarantee quick action in case of emergency. It is for this reason that the proposed system is to be designed to be a proactive one in that it minimizes hazards and preserves assets as well as to facilitate staff members in the handling of accidents. From the developments noted, the project increases safety and operational stability to restaurant kitchens.

2. Literature Review

This section gives an idea of the constructs, technologies and strategies that would be employed to elicit a smart fire and gas leak detection system suitable for restaurant setting. It uses state of the art sensors and hardware, software and cloud-based solutions in identifying dangers such as gaseous fumes, smoke, fire and heat. These elements are combined so that timely alarms and corrective actions improve staff and public security alongside physical assets and processes.

The principal parts of the proposed system include the MQ2 gas sensor, the IR flame sensor, temperature sensor thermistor and buzzer alarm. Each of the components is intended to cater for distinct risks. The MQ2 gas sensor finds gases such as methane, propane and butane, and gives an alert if there is any leakage [2]. The IR flame sensor characterizes infrared light of fire and ensures the fast recognition of it, especially in the areas like a kitchen [3]. The so-called temperature sensor keeps track of heat in the room advising the staff if temperatures go higher than a specific limit or range and the buzzer ensures that if there is anything within the room that poses a danger that is detected then there will be an audible signal to act on it.

The system is controlled by Arduino Wi-Fi-enabled NodeMCU, which processes signals from the sensors and performs wireless connection. This board enables the data to be transferred without any disruption to cloud platforms such as ThingSpeak which is used in real-time data acquisition, data presentation and data analysis. The IoT functionality of the system enables restaurant managers to have a remote view of the environmental conditions of the procurement area that can help in anticipating and controlling hazards. A development environment for the microcontroller is a structured system, which includes an essential application known as the Arduino IDE to control its programming.

The position and fine-tuning of the sensors also remain of utmost importance to the functioning of the system. Sensors are most placed where high incidence likely to occur this includes kitchens, storage rooms as well as dining areas. Tuning has a positive impact in raising the efficiency of finding the real objects and decreasing the rate of False Acceptance, thereby effectively increasing system performance. This way, safety issues are also taken into consideration at a high level, in order to develop the proposed system that will be able to meet restaurant needs starting from fire detection to gas leak detection.

2.1 Reviewing Existing Systems

A range of systems suggested in the literature is aimed at solving the problem of gas leakage and fire detection with the help of IoT and smart technologies. In the study by Jamadagni et al. [4], IoT monitoring system was designed in a low-cost manner aiming to monitor industrial settings to detect gas and fire hazards and communicate via SMS to reduce threats. Nevertheless, it does not have a web-based dashboard, and it is less convenient in terms of constant or massive monitoring as is needed in restaurants.

Al Towaijri et al. [5] proposed a new and capable IoT-based fire detection system that makes use of wireless cameras and convolutional neural networks (CNN) as a means of identifying high resolution fire objects. The system retains the privacy through only sending key video features instead of raw streams but complex modeling of machine learning and increased costs can make the system less feasible to establishments like restaurants.

Suma et al. [6] also developed an IoT system to be used in domestic context based on the combination of gas sensors and load cells providing the detection of LPG levels and the detection of LPG leaks with the possibility to send the SMS alert and autofill. Although this system can increase household safety, it does not have a multi-point fire detection that is necessary in crowded commercial kitchens.

Recently, Medewar et al. [7] offered an overview of contemporary practices in the development of fire and smoke detection-based on machine learning (with IoT devices incorporated into their frameworks) to enable predictive fire control and safety increase. They point to the utility of ML methods in the detection of improvements in the reliability and mitigation of false positives. However, the systems usually require high-level computational resources and expertise, thereby they are not most appropriate when the restaurants are small to medium size without a specific IT support.

On the contrary, the system that has been created in the current study is proprietary to a restaurant. By combining multi-sensor monitoring capabilities with a web-based dashboard, real-time alerts via WhatsApp messenger, and staff-management functionalities, it has found a balance between the technological savviness and the affordability of using this solution. This makes it a powerful source to enhance safety within a hazardous premise in food preparation.

Table 1: Comparison between existing systems and proposed system

Features/ System	Gas Leakage and Fire Detection Using Raspberry Pi [4]	A Privacy-Preserving Iot-Based Fire Detector [5]	Gas Leakage Detection Based on Iot [6]	A Review on Fire and Smoke Detection With Intelligent Control for Enhanced Safety Using Machine Learning (ML) and Internet of Things (IoT) [7]	Smart Fire and Gas Leak Detection System Iot
Gas Detection	✓	✕	✓	✓	✓
Fire Detection	✓	✓	✕	✓	✓
Temperature Monitoring	✕	✕	✕	✕	✓
Notification Type	SMS	App/Cloud	SMS	IoT/ML	WhatsApp
Staff Registration	✕	✕	✕	✕	✓
Emergency Response	✓	✓	✓	✓	✓
Response Time	Moderate	Fast	Moderate	Moderate	Fast
Use of ML / AI	✕	CNN	✕	ML	✕

3. Methodology

This project was developed using the parallel development approach that aided in the management of the project as it subdivided the project into small manageable segments that could be developed simultaneously. This strategy made the process of development quicker and more structured. One component of the system could be designed whilst another component of the system was already developed or under test. The methodology also contributed to lessening the delays and made it easier to correct the problems at an earlier stage. Development process involved some significant phases namely planning, system design, implementation and testing just to make sure that the final system was functional and that it achieved the objectives of the project and that it was a pleasant experience to use.

3.1 Parallel Development Methodology

The parallel development methodology of the product under development involves the division of the project into many component modules that are not dependent on the other, thus implying that in case one or some of the many developed modules makes use of a certain feature, other modules with similar requirements could be developed at the same time [8]. This was adopted in the Smart Fire and Gas Leak Detection System IoT due to the features that allows the separation of tasks in the various modules, which encompass monitoring and display, notification and alert subsystem, response mechanism and staff enrollment. The modules are fully self- contained and therefore can be worked on in parallel with no application degradation conundrums here. The methodology also makes use of consistent software tools including Arduino IDE in the sensor programming aspect, ThingSpeak in data log, analysis, and cloud notification, making the programming of the various modules easier. This method saves time since the modules are developed and tested in stages, yet it is accurate and reliable. Some of the crucial phases include design where the general framework and the specifications of specific modules are developed, implementation which comprises writing of code and integrating the functionalities while testing involves determining the performance and checking whether they meet safety standards. It is in terms of this efficiency that one can affirm the potential of developing a system that would address the more basic, but essential safety requirements.

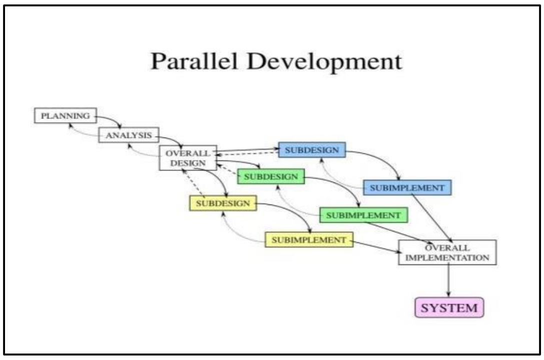


Figure 1: Parallel Development Methodology Diagram [8]

3.2 System Development Workflow

The System Development Workflow makes an outline of all the phases which are involved in the Software development process. Each phase also describes certain activities and deliverables that help bring implementation of the system to reality [9]. Table 2 below outlines these phases in detail.

Table 2: Task and Output for each Phase

Phase	Task	Output
Monitoring	Deploy IoT with sensors gas and fire sensors in key areas of the restaurant	Continuous monitoring of gas levels and fire detection. Data sent to a central database for real-time monitoring
Reporting	Access historical and real-time data on gas levels and fire incidents	Detailed reports on gas concentration levels, fire incidents, and system status. Optimized safety measures.
Visualization	Provide a user-friendly webbased dashboard for administrators and users	Visualized gas and fire detection data in graphs and charts. Insights into safety trends for informed actions.
Notification	Send alerts and notifications based on predefined conditions such as gas leak or fire detected	Alerts via WhatsApp for gas leaks or fire detection. Ensures timely response and preventive measures.
Staff Management	Provide tools for assigning tasks and managing safety checks and response procedures	Task management and tracking of safety checks by staff. Ensures swift response in emergencies
Registration	Allow staff to register and manage their profiles	Registered users and staff profiles for efficient management of safety protocols.
Time Tracking	Implement a time tracking system for staff working on safety checks and monitoring	Accurate recording of staff works hours. Streamlined task assignment and monitoring of shifts

4. Analysis and Design

This section proceeds to a detailed evaluation and design of the proposed Smart Fire and Gas Leak Detection System (IoT). It begins with a survey of the system structure and covers how the parts of the system interact to achieve the setup's goals. The specifications for the system are then provided by defining both what the system is to accomplish as well as how it should accomplish it. For clarity of understanding of the design and the flow an activity, class, use case and sequence diagram is shown to demonstrate the overall structure of the above concept. Lastly, this section delivers a practical description of how users will interact with the system by providing a description of its interface and its functions.

4.1 General System Architecture

The system architecture design diagram was illustrated on Figure 2 below, the diagram IoT-based Smart Fire and Gas Leak Detection System enhances safety and functionality through a harmonious connection of its parts. It is based on General Systems Theory to enforce the global interactions and concerted behavior of this system [10]. There are necessarily installed sensors at accurate areas that can identify the presence of gas, heat or fire and collect data and send them to cloud server named as Thing Speak for further analysis and storage. This has incorporated multiple interfaces monitor where staff can view updates on sensor and alerts options to call emergency services where necessary. System administrators control all the processes, such as sensors, employees, and static and dynamic reports, guaranteeing that alarms are resolved to avoid emergency situations. This architecture uses technology to provide reliable safety and monitoring as much as possible.

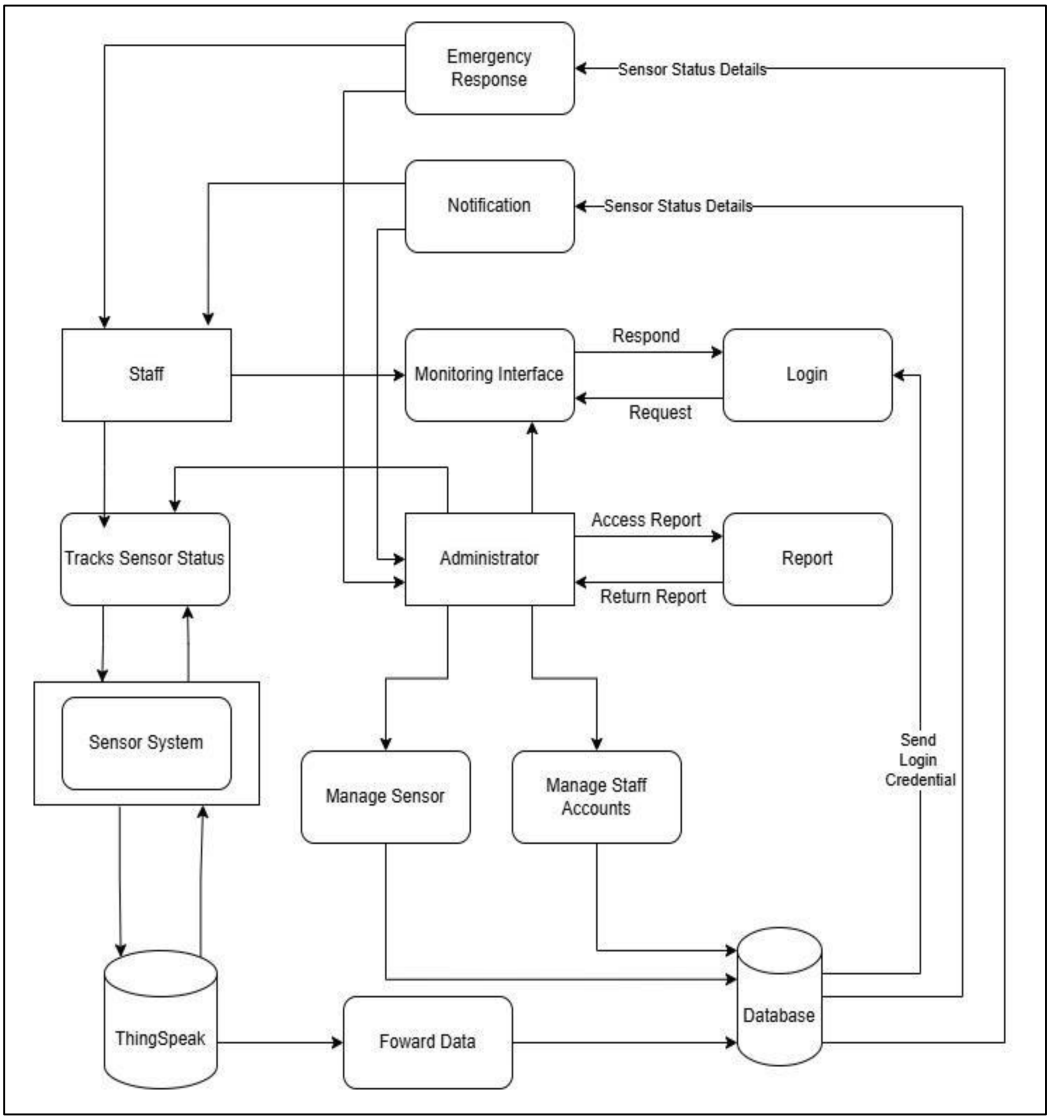


Figure 2: System Architecture Diagram

4.2 Requirement Analysis

Requirement analysis is a formal process aimed at understanding the requirements of the user and the specifications of the project to determine what it is that the project needs to accomplish and what it cannot do. It includes functional requirements on Table 3, which specifies how the identified system needs to behave or act and non-functional requirement on Table 4, which describes how the identified system should be used or operated.

Table 3: Functional Requirements for the Proposed System

No	Module	Functionality

1.	Surveillance and Data Visualization	Deploys IoT sensors to monitor fire and gas levels in real-time. Sensor data is sent to a central database and displayed on a dashboard for visual analysis.
2.	Notification	Sends real-time alerts to users via WhatsApp when gas leaks or fires are detected. Notifications include preventive measures to mitigate risks.
3.	Emergency	Provides an interface for quick-call options to contact emergency services such as the police, fire station or ambulance, ensuring rapid response during incidents.
4.	Registration and Staff Management	Enables user registration and login functionalities. Administrators can manage staff profiles, assign roles, and control access levels for secure system operations
5.	Reporting	Offers access to historical and real-time data related to fire and gas incidents. Users can generate detailed reports for analysis, including trends and risk areas.
6.	Visualization	Provides a user-friendly web-based dashboard that visualizes fire and gas sensor data through charts and graphs, aiding stakeholders in identifying risks effectively.

Table 4: Non-Functional Requirements for the Proposed System

No	Function	Functionality
1.	Performance	The system must provide real-time monitoring of fire and gas levels with minimal latency and handle large volumes of sensor data efficiently.
2.	Reliability	The system must ensure continuous operation for fire and gas detection with backup mechanisms to handle hardware or software failures.
3.	Scalability	The system should be scalable to accommodate an increasing number of sensors and users without compromising performance.
4.	Security	Robust security measures should protect user data and system integrity. Access control mechanisms must manage permissions and prevent unauthorized access.
5.	Usability	The user interface must be intuitive and easy to use, allowing stakeholders, including administrators and staff, to operate the system with minimal training.
6.	Compatibility	The system must be compatible with various devices and web browsers, supporting common operating systems and platforms.
7.	Maintainability	The system should be designed for easy maintenance and updates, with comprehensive documentation provided for administrators and users.
8.	Environmental Impact	The system should contribute to reducing environmental risks by preventing fires and mitigating gas leaks, ensuring safer and more sustainable operations.

4.3 Object Oriented

This section makes use of Object-Oriented Analysis and Design to model the functionality and structure of Smart Fire and Gas Leak Detection System (IoT) using real life entities and their interactions. Diagrams are used to illustrate key aspects such as, the use case diagram displays how the different users of the system engage with the system, the sequence diagram depicts on how the different messages in one process interrelate, the activity diagram depicts the different activities in the system and the class diagram describes the system using classes, attributes and connections [11]. Altogether, these diagrams give concise and rational vision of architecture and development of the system.

4.4 Use Case Diagram

The use case diagram for the Smart Fire and Gas Leak Detection System (IoT) is found in the Figure 3, outlines the roles of two main actors which are administrators and staff. The system includes eight core functions: observe areas, enter logs, create accounts, get notifications, act during emergencies, deal with accounts, remove accounts, and obtain reports. Companies' administrators manage everything, including accounts, reports and emergencies. Users can log in, register, view conditions, be notified and attend to occurrences but cannot cancel an account or produce a report. Specialization of work is well illustrated in the diagram indicating the distinct tasks for each of the constituents in the system.

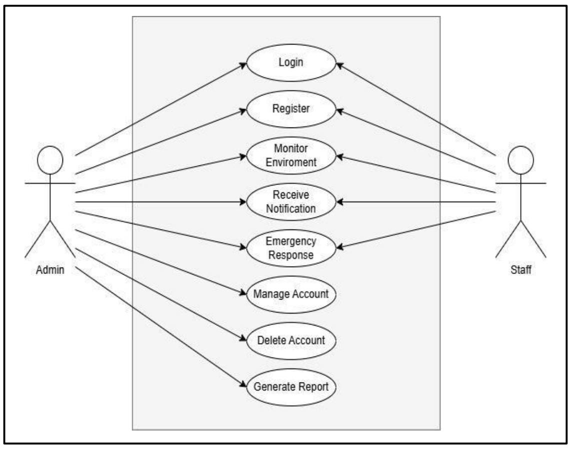
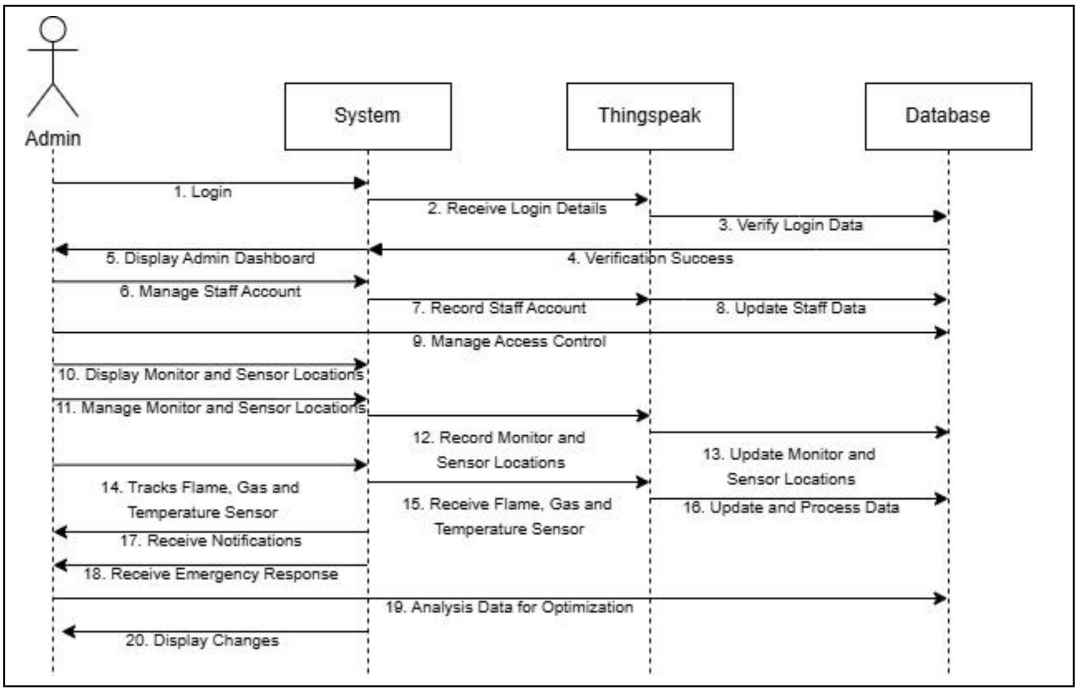


Figure 3: Use Case Diagram

4.5 Sequence Diagram

The Smart Fire and Gas Leak Detection System (IoT) sequence diagrams presented in the Figure 4 include some of the primary processes of data exchange, staff and administrator interference. In data transfer, sensors stream the raw data directly to ThingSpeak for ingestion and analysis, where live monitoring and system dynamic updates are accomplished, with cyclic request for testing the functionality of the sensors. Regarding the staff, the diagram indicates where they login, what they use to monitor, and the alerts, whether normal, abnormal, or emergency. The administrator's sequence includes an input of the login authorization, the ability to open the admin panel, and authorization of such functions as staff account managing, privilege setting, sensor monitoring, and emergencies. These diagrams illustrate how the overall system is live, efficient and secure for its purpose.



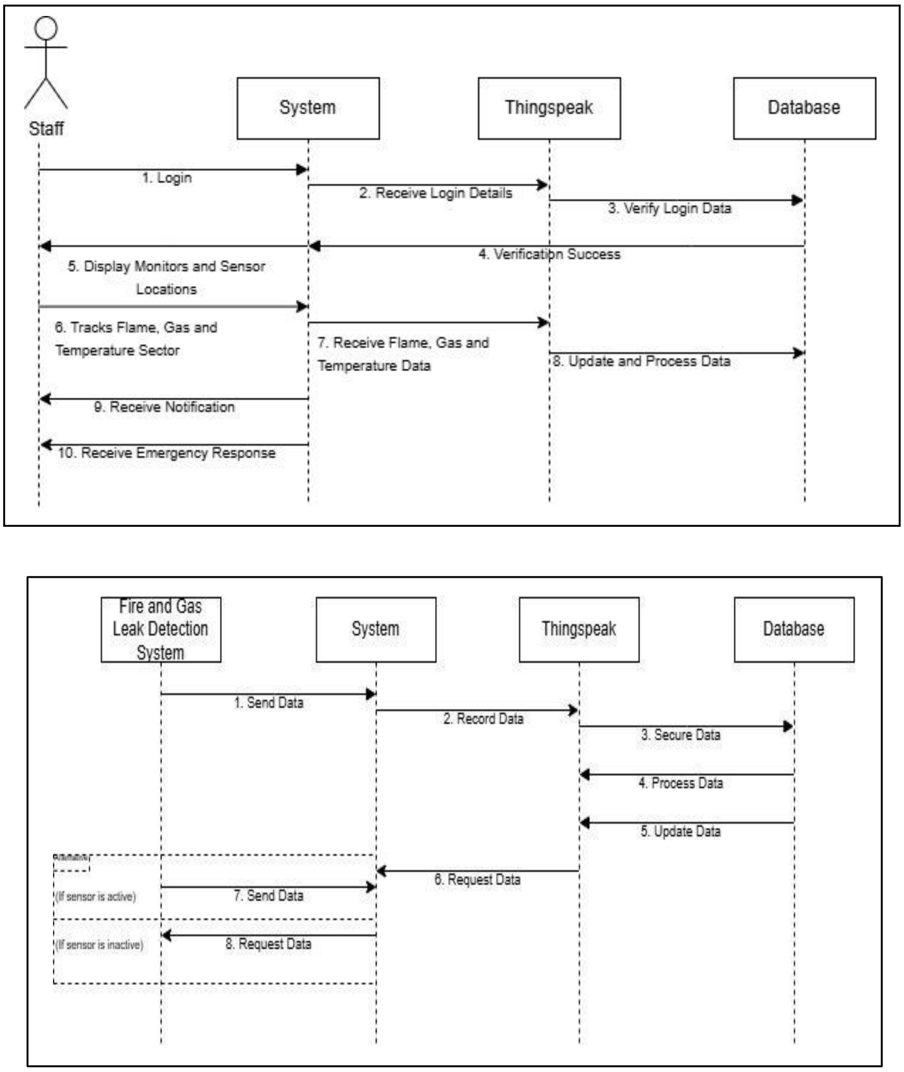


Figure 4: Sequence Diagram

4.6 Activity Diagram

The activity diagram demonstrating the operation flow of the Development of Web-Based IoT Smart Fire and Gas Leak Detection System is given in Appendix B. The following diagram describes the process through which site visitors start with logging in account accounts using identification and registration information so the new users create registration accounts whereas the returning users create logins. After being authenticated, users will be defined as either an administrator or its staff. Administrators manage sensors, user accounts and reports so as to make the system run smoothly. The main work of the staff members is to monitor sensors, respond to alerts and deal with emergencies. The process is over at the moment when all required assignments are fulfilled, which allows, in fact, the prediction and elimination of the occurrences involving fire and gas leaks.

4.7 Class Diagram

The class diagram in Figure 6 offers a representation of the relationships between elements of the Smart Fire and Gas Leak Detection System. Other components include System, Administrator,Staff, NotificationSystem, EmergencyResponse, SensorArea and gas, flame and temperature Sensors. The System is managed by an administrator and has the status of staff to track processes. It notifies through the NotificationSystem and deals with emergencies through the EmergencyResponse. Several types of SensorAreas monitor the state using concrete Sensors that are inherited from a universal Sensor. The diagram under consideration also points to certain areas, how the system collects information, alerts users about threats and prevents or minimizes their occurrence.

Applied Information Technology and Computer Science Vol. 6 No. 2 (2025) p. 247-264

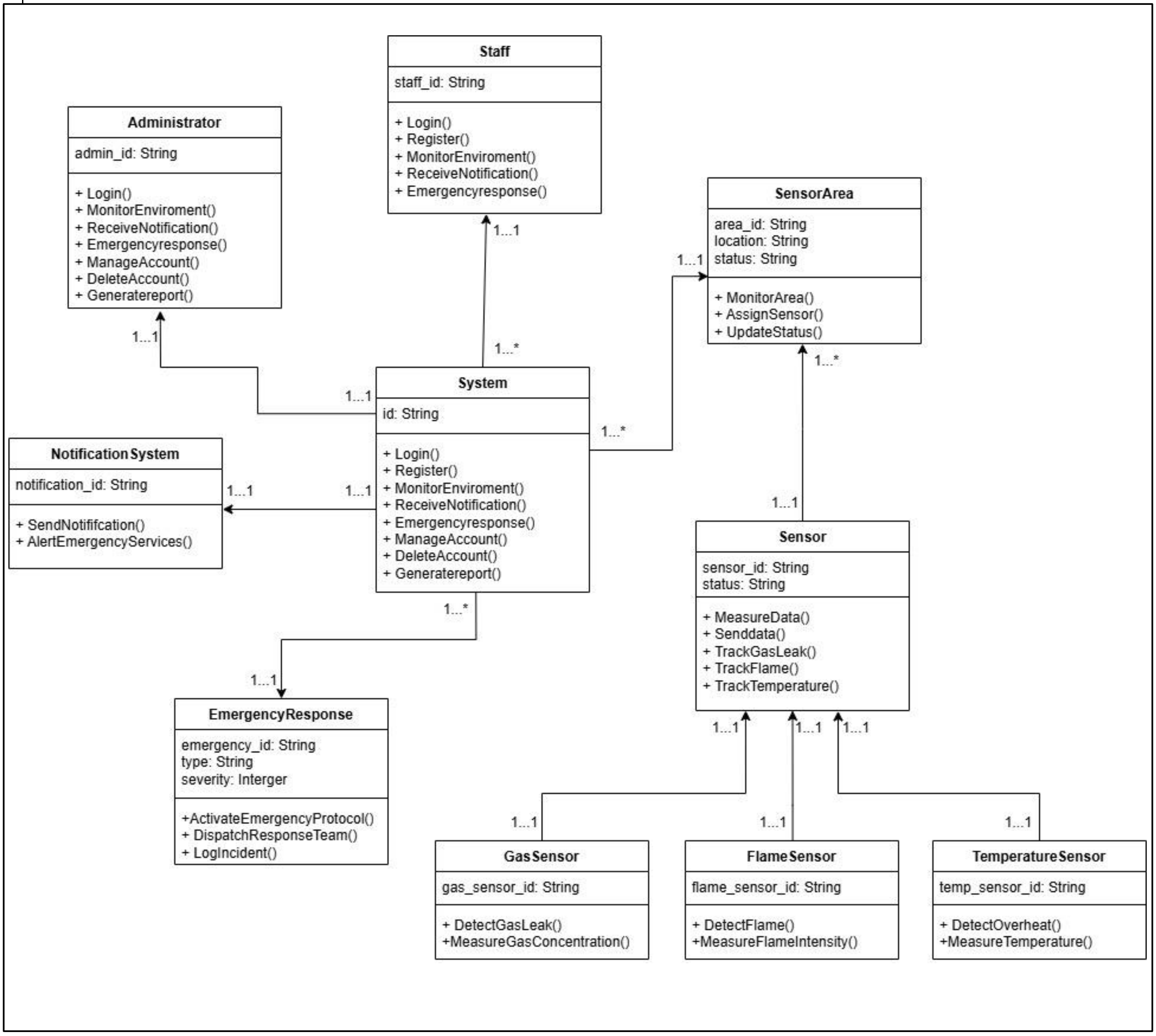


Figure 6: Class Diagram

4.8 User Interface

Interface design or user interface (UI) design is the process of developing the visual appearance and interactive behaviour of the system. UI design is meant to ensure that the system is user friendly, performs well and creates a positive experience to the user. In this project, the interface was developed to have some significant aspects like registration, login, monitoring dashboard, profile settings, staff management, sensor management, notifications and viewing reports so that users could interrelate with the system conveniently and effectively.

5.0 Results and Discussion

The web-based interface of the Smart Fire and Gas Leak Detection System (IoT) is created in Visual Studio Code, Laravel, PHP and MySQL. Programming code for the IoT components is written in the Arduino IDE in C++. Key hardware supports the system, for example MQ-2 Gas, IR (Infrared) Flame and DHT-11 Temperature Sensors, which are all attached to the Wemos D1 ESP8266 microcontroller. For live data transmission and viewing, ThingSpeak is the platform and to receive immediate notifications, WhatsApp is included. The system relies on MySQL to keep user details, readings from sensors and emergency notices, making certain that all data is handled properly.

5.1 System Implementation

Figure 7 presents the prototype of the IoT-based Smart Fire and Gas Leak Detection system placed in a simulated kitchen. Multiple sensors are placed in different parts such as close to the stove and gas pipes to spot any leaks, flames, and dramatic temperature changes. The sensors are linked to a Wemos D1 ESP8266 microcontroller that gathers the data and sends it remotely for review. It is designed to function alongside alarms, an internet dashboard, and real-time alerts so users can respond quickly during emergencies and boost safety.

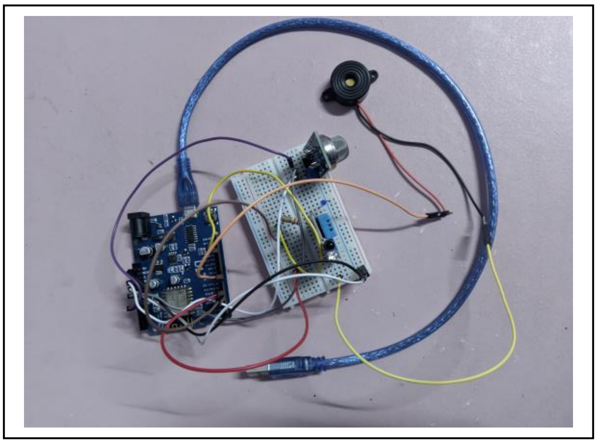


Figure 7: Wiring Diagram of Iot-Smart Fire and Gas Leak Detection System

The login interface displays how to log in to the Smart Fire and Gas Leak Detection System through the IoT. The first image displays the Admin Login page, where administrators can safely enter their email and password to open the system's management section. The second image shows the Staff Login page, where staff members can view the sensor data and receive warnings when needed. Users can use the "Forgot Password" link on either interface when they have forgotten their account information. The design is clear and user-friendly, allowing admins and site staff to use the system with no difficulty.

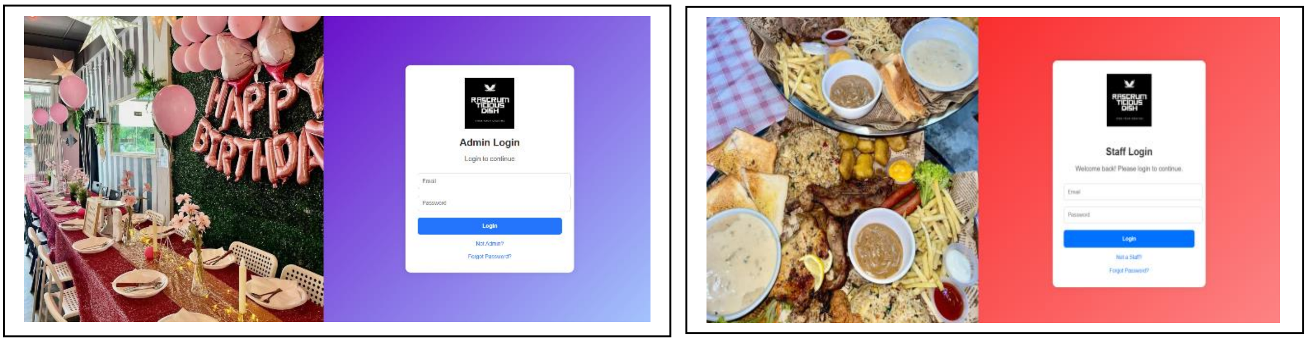
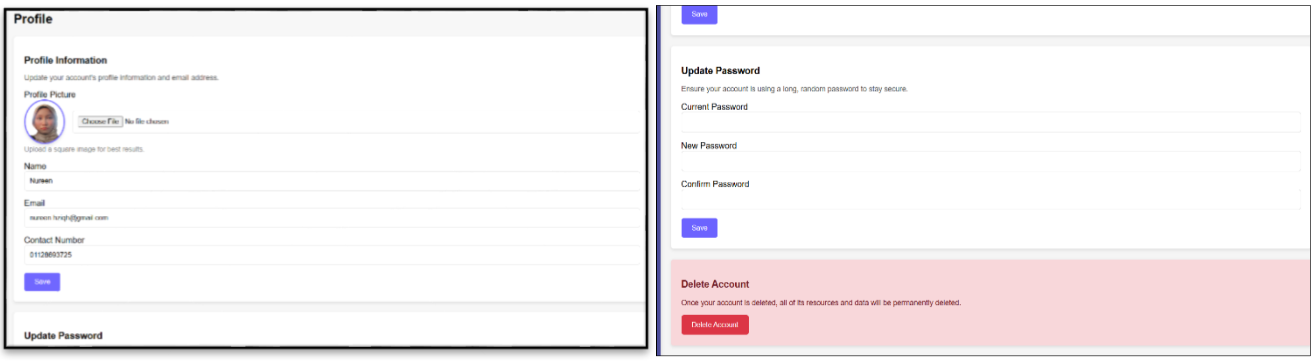


Figure 8: Admin and Staff Login Page

The profile management pages for both admin users and staff are displayed in the Smart Fire and Gas Leak Detection System (IoT) as demonstrated by Figure 9. The first image represents Nureen's Admin Profile, in which the administrator can change the email address, password, phone number and profile picture. The next picture presents the Staff Profile page for "Dora" and supports the same features for managing personal email, password, phone number and profile image. The design ensures both admin and staff can manage their account data safely, making it easy for them and maintaining valid contact information in the system.



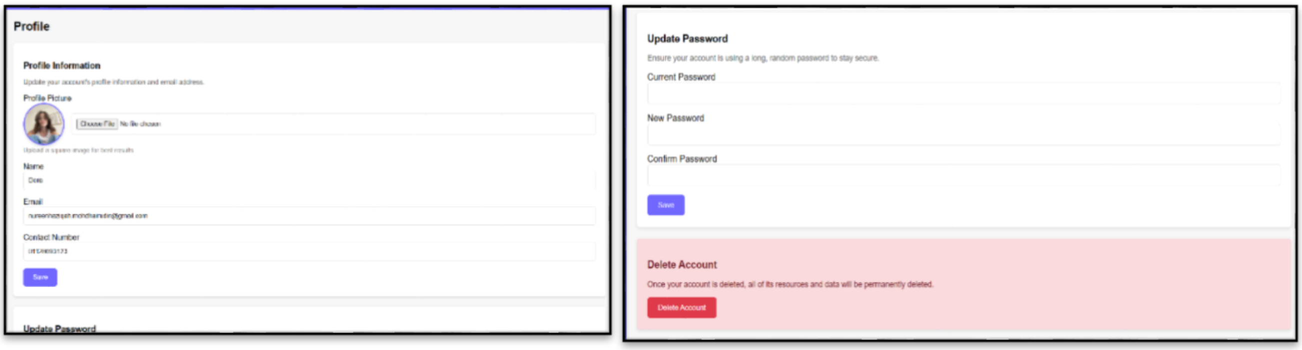


Figure 9: Admin and Staff Profile

The monitoring page of the Smart Fire and Gas Leak Detection System (IoT) is displayed in the Figure 10 and gives up-to-date information about environmental conditions in different areas of the restaurant. For each area, the system shows important details about temperature, humidity, gas level and fire. The arrangement enables personnel to keep watch over various areas at once and spot any unusual occurrences or dangers that might arise. The well-structured interface allows for faster decisions, which improves safety at work.

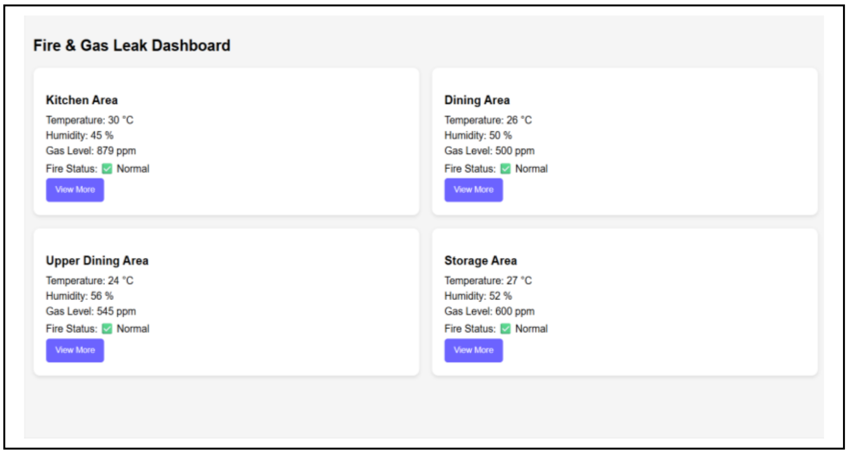


Figure 10: Monitoring Page

Figure 11 demonstrates the page where administrators manage staff in the Smart Fire and Gas Leak Detection System. Here, the admin has the ability to check and handle existing staff accounts. Administrators may update employee records and eliminate staff accounts to maintain the necessary level of security. This function ensures the safety of the system and keeps all employee-related information neat and current.

Figure 11: Staff Management on Admin Page

Figure 12 illustrates the Sensor Management part of the administrator panel in the Smart Fire and Gas Leak Detection System. Admins can use this page to include new sensor areas by giving information like the name, the assigned sensor ID and when it was installed. The information provided covers available sensor areas with their IDs, when they were installed, and ways to manage them. For each sensor, there is an Edit button and a Delete button, so the administrator may make any necessary updates or remove a sensor configuration. This makes the system easy to organize, maintain accuracy and adjust whenever the surrounding environment changes.

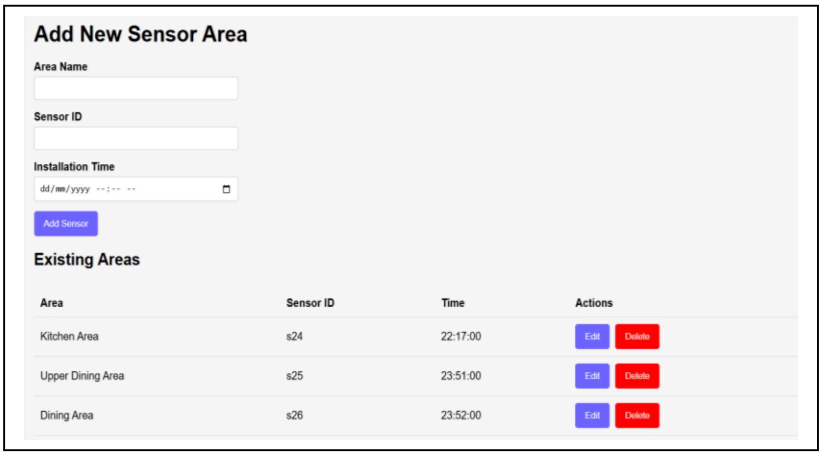


Figure 12: Sensor Management on Admin Page

Figure 13 displays the Report Page of the admin in the Smart Fire and Gas Leak Detection System (IoT). It shows the sensor data in graph and table view such as temperature, humidity, gas level, and fire status. Admins are able to filter the data based on the date to see particular records. This page assists admins to monitor the environment and alert any rapid changes.

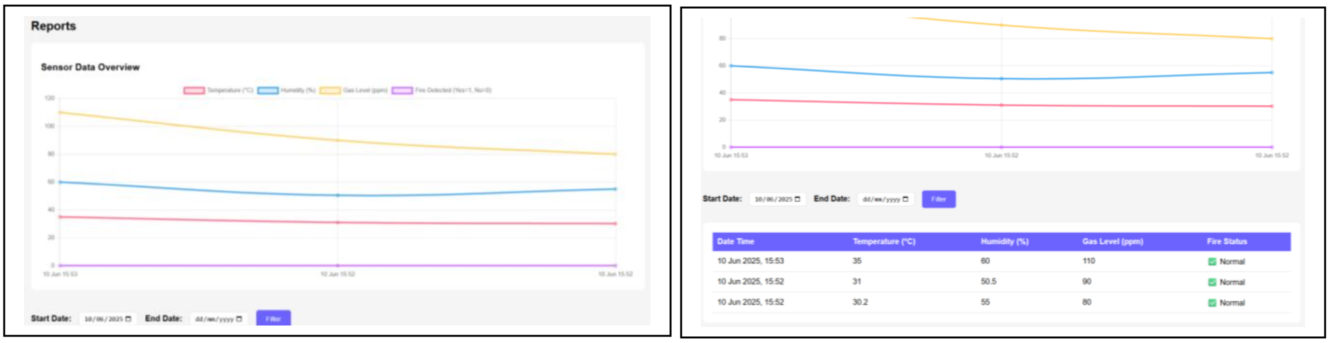


Figure 13: Report on Admin Page

To notify users in the event of a fire or gas leak, the Smart Fire and Gas Leak Detection System has a notification efficiency. Figure 14 show that WhatsApp is used to send notifications in real time, ensuring that employees are alerted to emergencies right away. Along with necessary emergency contact information and evacuation directions, these notifications give clear guidance on what to do. In emergency scenarios, this functionality lets users react quickly and remain secure.

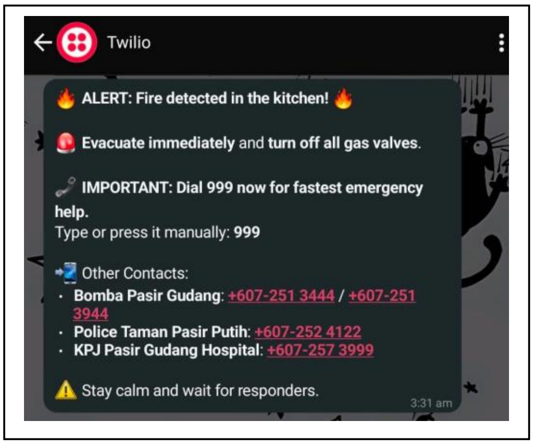


Figure 14: Notification

5.2 System Testing

In order to validate that the system functions well and has no significant issues, it has to complete proper evaluation before being made available for usage. Therefore, to ensure that the system operates as intended, this stage is important. A thorough test strategy and User Acceptance Testing (UAT) were used to test the system in order to verify this.

5.2.1 Test Plan

The purpose of this test plan is to ensure that the system functions as expected. All parts of the system have undergone testing to ensure that it performs as wanted. Testing ensures that the system is ready for usage and helps identify any issues.

Table 5: Surveillance & Data Visualization Module

Test	Expected Result	Actual Result
Display of real-time sensor data	Sensor data (temperature, humidity, gas, fire status) updates in real-time.	Pass
Display accurate sensor area	Every sensor shows data relevant to its assigned area.	Pass
Data synchronization with database	Sensor data is accurately stored in the database	Pass

Table 6: Notification Module

Test	Expected Result	Actual Result
WhatsApp notification on gas detection	WhatsApp alert received when gas level is exceeded	Pass
WhatsApp notification on fire detection	WhatsApp alert received when flame is detected	Pass
Message includes preventive instructions	Instructions or reminders for preventative action are included in the notification	Pass

Table 7: Emergency Module

Test	Expected Result	Actual Result
Quick call button triggers dialer	Call button redirects to phone dialer with emergency number	Pass
Emergency contact details display	Contact numbers for fire station, police, hospital are visible	Pass

Table 8: Registration & Staff Management Module

Test	Expected Result	Actual Result
User registration (staff)	Staff can register successfully	Pass
Admin login	Admin can log in successfully	Pass
Staff login	Staff can log in successfully	Pass
Admin can add/edit/delete staff profiles	Admin can fully manage staff details	Pass

Table 9: Reporting Module

Test	Expected Result	Actual Result
Real-time report generation	Reports show current fire/gas incidents accurately	Pass
Historical report display	Previous incident records are accessible and complete	Pass

Table 10: Visualization Module

Test	**Expected Result**	**Actual Result**
Graphical display of data	Sensor data displayed as clear, readable charts or graphs	Pass

5.2.2 User Acceptance Testing (UAT)

The aim of User Acceptance Testing (UAT) was to determine that the system was functional to the requirements of the users and that it performed well under real life conditions, particularly within a restaurant setting. The system was tested with real users to assess the functionality, the ease of use as well as the performance. The users also gave their feedback which indicated that the system was realistic, convenient, and functional as anticipated. The results of UAT are presented in Table 11 and Figure 15 form and give a clear picture of the test results and show that the system sufficiently met user requirements.

Table 11: Result of User Acceptance Testing for Admin User

No	**Acceptance Requirement**	**Actual Result (Ranking)**				
		1	**2**	**3**	**4**	**5**
User Interface						
I	Easy to understand				✓	
II	Navigation				✓	
III	Interface (e.g., color, background, font)					✓
System Function						
IV	Login/Registration function				✓	
V	Account management function				✓	
VI	Data visualization function				✓	
VII	Notification function (WhatsApp)				✓	
VIII	Report function				✓	
IX	Sensor management function					✓
X	Sensor status and detail update				✓	

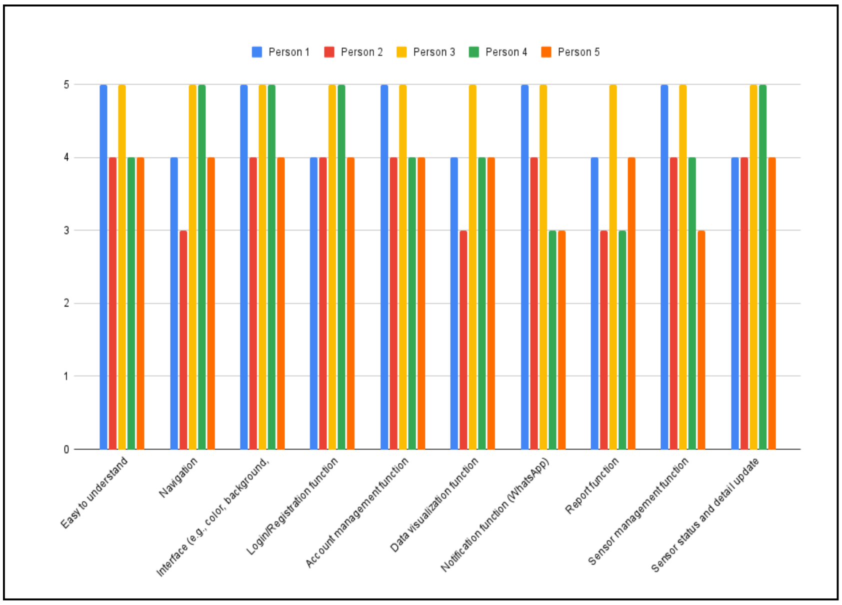


Figure 15: Result of User Acceptance Testing for Admin User

6.0 Conclusion

To summarize, the Smart Fire and Gas Leak Detection System (IoT) has proven to be a successful and practical solution for increasing safety in locations with fire and gas risks, particularly in restaurant settings. Using IoT technologies such as sensors, Wemos D1 ESP8266, ThingSpeak and WhatsApp notifications, the system was able to deliver real-time alerts and address critical difficulties such as delayed detection and slow emergency response.

However, as with every system, it has limitations. The system relies on a stable internet connection to ensure that data is delivered to the cloud and that users receive timely notifications. If the connection to the internet is unstable or unavailable, alerts may be delayed. The sensors and hardware also require frequent maintenance to ensure that they function effectively over time. Furthermore, if the sensors malfunction, the readings' accuracy may be compromised.

For future enhancements, SMS alerts can be added as an alternate alert option in the event of an internet outage. The system can potentially be upgraded by incorporating AI or machine learning to anticipate future threats based on sensor trends. Expanding the system to monitor wider areas or different types of buildings can improve its utility in a variety of settings. With these enhancements, the system will become more dependable, intelligent, and capable of offering improved safety in a broader range of workplaces.

Acknowledgement

The authors would like to thank family members, friends and the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

This journal requires that all authors take public responsibility for the content of the work submitted for review. The contributions of all authors must be described in the following manner:

The authors confirm contribution to the paper as follows: **study conception and design:** Nureen Haziqah binti Mohd Hairudin, Nayef Abdulwahab Mohammed Alduais; **data collection:** Nayef Abdulwahab Mohammed Alduais; **analysis and interpretation of results:** Nureen Haziqah binti Mohd Hairudin, Nayef Abdulwahab Mohammed Alduais; **draft manuscript preparation:** Nureen Haziqah binti Mohd Hairudin, Nayef Abdulwahab Mohammed Alduais. All authors reviewed the results and approved the final version of the manuscript.

References

[1] Hernandez, A., & McDonald's, M. P. (2023). HEAT, HAZARDS AND INDIFFERENCE TO SAFETY IN CALIFORNIA'S FAST FOOD RESTAURANTS.

[2] Staff, L. E. (2022, October 12). *In-depth: How MQ2 gas/smoke sensor works? & interface it with Arduino*. Last Minute Engineers. https://lastminuteengineers.com/mq2-gas-senser-arduino-tutorial/

[3] Admin. (2020, November 18). *Ir Infrared Flame Sensor Module: How fire sensor works " electroduino*. ElectroDuino. https://www.electroduino.com/ir-infrared-flame-sensor-module/

[4] Jamadagni, S., Sankpal, P., Patil, S., Chougule, N., & Gurav, S. (2019, March). Gas leakage and fire detection using raspberry Pi. In *2019 3rd International conference on computing methodologies and communication (ICCMC)* (pp. 495-497). IEEE.

[5] Altowaijri, A. H., Alfaifi, M. S., Alshawi, T. A., Ibrahim, A. B., & Alshebeili, S. A. (2021). A privacy-preserving iot-based fire detector. *IEEE Access*, *9*, 51393-51402.

[6] Medewar, A. G., Sawarkar, A. D., Kshirsagar, U. V., Mr, A. G. M., & Kshirsagar, U. (2024). A review on fire and smoke detection with intelligent control for enhanced safety using machine learning (ML) and Internet of Things (IoT). *Cureus Journals*, *1*(1).

[7] Suma, V., Shekar, R. R., & Akshay, K. A. (2019, June). Gas leakage detection based on IOT. In *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)* (pp. 1312-1315).

IEEE.

[8] Danelutto, M., Di Meglio, R., Orlando, S., Pelagatti, S., & Vanneschi, M. (1992). A methodology for the development and the support of massively parallel programs. *Future Generation Computer Systems*, *8*(1-3), 205-220.

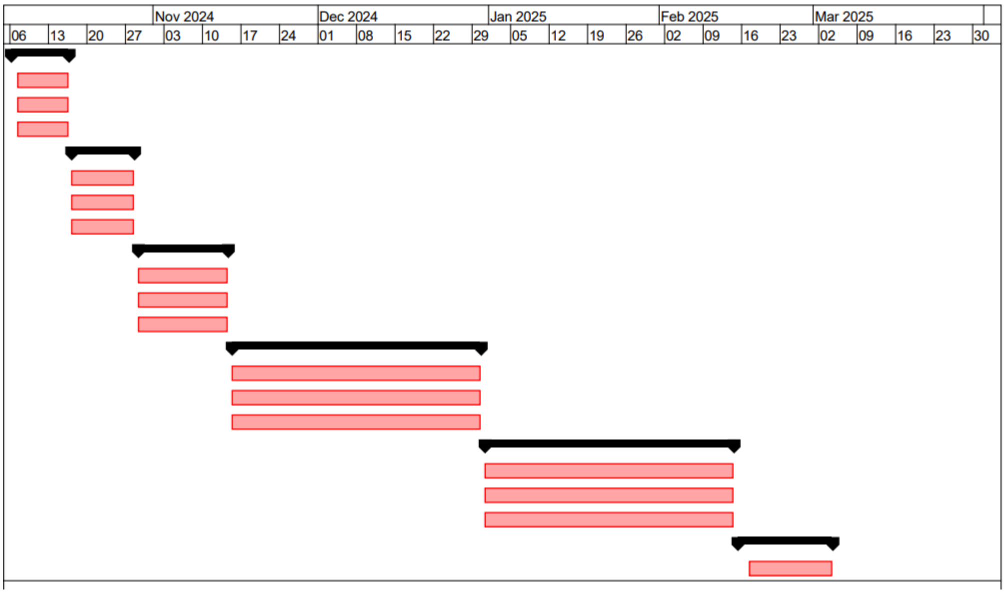
[9] Jeremy-Powers. (2014, November 19). *PPT - System Development Methodologies PowerPoint presentation, free download - ID:6831448*. SlideServe. https://www.slideserve.com/jeremy-powers/system-development-methodologies

[10] Kim, J., & Robert Carlson, C. (2002, December). A design methodology for workflow system development. In *International Workshop on Databases in Networked Information Systems* (pp. 15-28). Berlin, Heidelberg: Springer Berlin Heidelberg.

[11] Lv, W., Meng, F., Zhang, C., Lv, Y., Cao, N., & Jiang, J. (2017, July). A general architecture of IoT system. In 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC) (Vol. 1, pp. 659- 664). IEEE.

Appendix A: Gantt Chart

		Name	Duration	Start	Finish
1		**Project Planning Phase**	**8 days**	**10/6/24 8:00 AM**	**10/16/24 5:00 PM**
2		Define Project Scope	8 days	10/6/24 8:00 AM	10/16/24 5:00 PM
3		Resource Allocation	8 days	10/6/24 8:00 AM	10/16/24 5:00 PM
4		Create Gantt Chart	8 days	10/6/24 8:00 AM	10/16/24 5:00 PM
5		**Analysis Phase**	**8 days**	**10/17/24 8:00 AM**	**10/28/24 5:00 PM**
6		Requirement Analysis	8 days	10/17/24 8:00 AM	10/28/24 5:00 PM
7		Problem Identification	8 days	10/17/24 8:00 AM	10/28/24 5:00 PM
8		Feasibility Study	8 days	10/17/24 8:00 AM	10/28/24 5:00 PM
9		**Design Phase**	**13 days**	**10/29/24 8:00 AM**	**11/14/24 5:00 PM**
10		System Architecture	13 days	10/29/24 8:00 AM	11/14/24 5:00 PM
11		Module Design	13 days	10/29/24 8:00 AM	11/14/24 5:00 PM
12		UI/UX Design	13 days	10/29/24 8:00 AM	11/14/24 5:00 PM
13		**Implementation Phase**	**32 days**	**11/15/24 8:00 AM**	**12/30/24 5:00 PM**
14		Backend Development	32 days	11/15/24 8:00 AM	12/30/24 5:00 PM
15		Frontend Development	32 days	11/15/24 8:00 AM	12/30/24 5:00 PM
16		Integration	32 days	11/15/24 8:00 AM	12/30/24 5:00 PM
17		**Testing Phase**	**34 days**	**12/31/24 8:00 AM**	**2/14/25 5:00 PM**
18		Unit Testing	34 days	12/31/24 8:00 AM	2/14/25 5:00 PM
19		Integration Testing	34 days	12/31/24 8:00 AM	2/14/25 5:00 PM
20		System Testing	34 days	12/31/24 8:00 AM	2/14/25 5:00 PM
21		**Finalization**	**12 days**	**2/15/25 8:00 AM**	**3/4/25 5:00 PM**
22		Documentation and Han...	12 days	2/15/25 8:00 AM	3/4/25 5:00 PM



Appendix B: Activity Diagram