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

AITCS

Vol. 6 No. 1 (2025) 1871-1887
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System

Muhammad Danish Ariff Zainal Abidin[1], Suziyanti Marjudi[1]*

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

*Corresponding Author: suziyanti@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2025.06.01.104

Article Info

Received: 12 June 2024
Accepted: 18 June 2025
Available online: 30 June 2025

Keywords

Internet of Things (IoT), Air pollution, gas sensor, web system

Abstract

The use of the internet of things has been widely applied in the daily routine of life and surrounding conditions. AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System was developed based on IoT technology to help its users monitor the quality of surrounding air as there was no system to monitor the air pollution developed for schools. The main objective of this system development is to develop a real-time IoT-Based Air Pollutant Monitor system using a structured approach and using web-based approach and IoT based approach and to test the developed system for user acceptance by using the Agile methodology. The system will show about the air pollution reading and can be used for early precaution for the students, school staff and residents around the school compound. By using this system, it can help students and school staff to gain awareness on air pollution and this system can be integrated with other systems as well.

1. Introduction

The Internet of Things (IoT) is a system of interconnected devices and sensors that collect and share data about their surroundings. It has become a promising technology for addressing societal challenges by connecting smart devices to create smart cities worldwide. IoT for smart cities needs to guarantee the accessibility of open data and cloud services to allow industries and citizens to develop new services and applications. In recent years, the Internet of Things (IoT) has drawn significant research attention [1]. Air pollution happens when there is a mixture of harmful chemicals in the air that will cause damage to humans and the surroundings. Its effects can vary from a higher risk of diseases to rising temperatures [2].

Based on Air Pollutant Index that has been issued by Department of Environmental Malaysia, a lower Air Pollution Index (API) reading of air quality will results in a healthy air which can be safely consume by human being while a higher reading can give huge impact on human health through respiratory [3]. In SMK Senggarang, there is not any system implemented that can monitor the quality of the air around the school that has been developed to help the students and school staff who had a respiratory problem and the case of air pollutant around the school compound are rising as the school location is near to the main road from Batu Pahat to Pontian and near to palm plantation. An IoT Web- Based System AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System can overcome this problem as it is a system that can monitor the air pollution by help them monitoring the air quality of the school area, so it is safe for students and school staff to conduct any activities plus enlighten them on how important on implementing the IoT in their daily life.

The primary goal of implementing such a system is to develop a system that can monitor the air pollution around the school compound. The AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System utilized a structured approach by implementing it into an IoT web-based system. This system helps by monitoring the air

quality of the school area, so it is safe for students and school staff to conduct any activities plus enlighten them on how important it is to implement IoT in their daily life.

The developed system AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System is expected to be able to enlighten people about the importance of monitoring the air pollution around us by implementing the IoT technology in our daily life without needed to depend on news as it is slow and they could get the information faster, and any significant action could be made on the spot by using this system.

2. Literature Review

This section explains the case study of the developed system, research, and analysis of three existing system that consist of their functions and comparison between each system and the proposed system to find any advantage and improvement of the developed system.

2.1 Case Study

The case study investigates the development of the air pollution monitoring system by using the Arduino that has been conducted by a researcher. The researcher has proposed an Air Pollution Monitoring System where the system uses an Arduino microcontroller connected with MQ135 and MQ6 gas sensor which senses the different types of gases present in the environment. It was then connected to the Wi-Fi module which connects to the internet and LCD is used to display the output to the user and buzzer alerts when the ppm crosses certain limits. Their applications were industrial perimeter monitoring, indoor air quality monitoring, site selection for reference monitoring stations, making data available to users [4].

An interview has been conducted with the IT Department of SMK Senggarang to gather every data and requirements for developing this system. During the interview, a study was made where there is no system that has been implemented in SMK Senggarang to detect any air pollution that causes harm to the students and school staff. Based on the information, the alerts on monitoring the air pollution were only made by gaining information from news and from the ministry department. This is a disadvantage for the situation on the school as Malaysia are having high amount of air pollution these days based on the trend of studies that has been conducted which will causes asthma, current wheeze, dry cough at night and wheezing after exercise [5].

2.2 Review On Existing System

The review was focused on three existing systems which is Air Pollution Detection System using Edge Computing [6], An Internet of Things-based Air Pollution Detection System [7] and Real Time Air Quality Monitoring System [8].

2.3 Air Pollution Detection System using Edge Computing

This system was proposed to measure air quality and detect pollution in real-time, which can provide timely information about environmental conditions by using Edge Computing which used the Cloud server as the database to keep all the data and shows the output of the air pollution reading. The systems use the Raspberry Pi as their main device to connect to the system and all the data is collected and kept by using Microsoft Azure. The system also includes an alerting mechanism that triggers notifications when pollution levels reach the danger level which shows the main function of the system after gaining the read of air pollution. The system also used internet connection as the medium to transmit air pollution reading to the system which are related to the developed system.

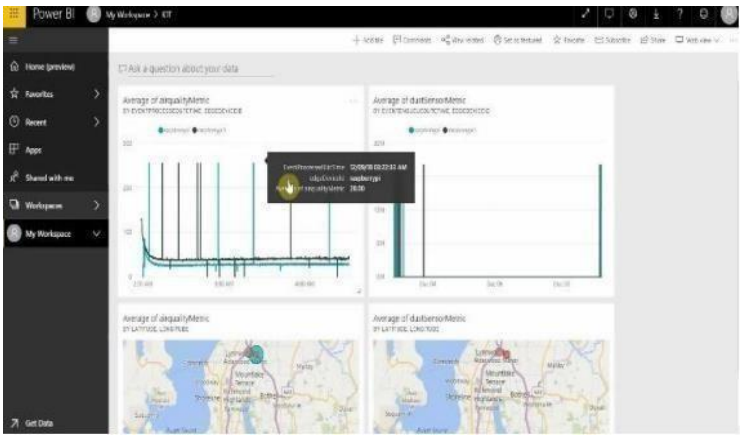
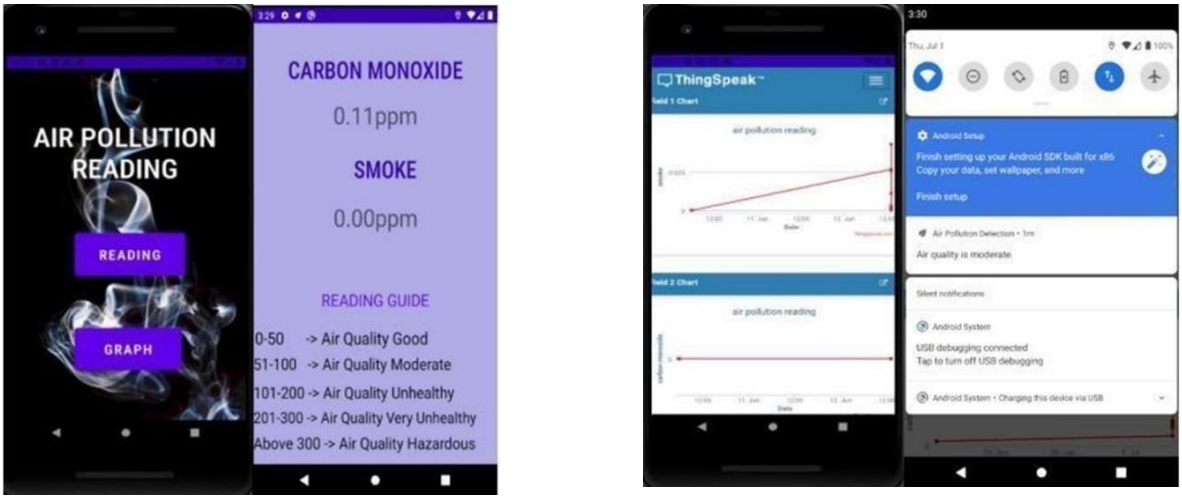


Fig. 1 *Air Pollution Detection System using Edge Computing Dashboard*

Applied Information Technology and Computer Science Vol. 6 No. 1 (2025) p. 1871-1887

2.4 An Internet of Things-based Air Pollution Detection System

An Internet of Things-based Air Pollution Detection System is an application for mobile phones that are being used to detect the air pollution around the atmosphere. The system consists of two functions which are viewing air pollution reading and viewing air pollution reading graph but it lacks with Login module so the system can be used by anyone who installed the application via their mobile phones without any security measures. The systemuses the Arduino Uno as the microcontroller of the system but with less sensors applied to the device. The system uses Google Firebase as the database for the system to keep all the data about the air pollution read and the ThingSpeak application to generate the graph of the air pollution to the user with the description of the air pollution content. The system also lacks generating reports for the user which can be described as the disadvantage of the system.



(a) (b)

Fig. 2 *An Internet of Things-based Air Pollution Detection System (a) Main Page; (b) notification alert interface*

2.5 Real Time Air Quality Monitoring System

Real Time Air Quality Monitoring System is developed by Universiti Teknologi Petronas (UTP) to monitor the Air Pollution Index, humidity and temperature of surrounding through internet connection. The system used the Arduino Uno microcontroller that has been connected to the ThingSpeak Cloud. The system is able to monitor the concentration of Carbon Monoxide (CO) and all the data are presented on a graph as a result of reading the air pollution. The system uses the GSM model as a medium to send notification to the user via SMS so it will not require any internet connection to get the notification of the air pollution reading. By using the ThingSpeak website, the system also able to shows the graph and reading of the air pollution, humidity, and temperature to the user.

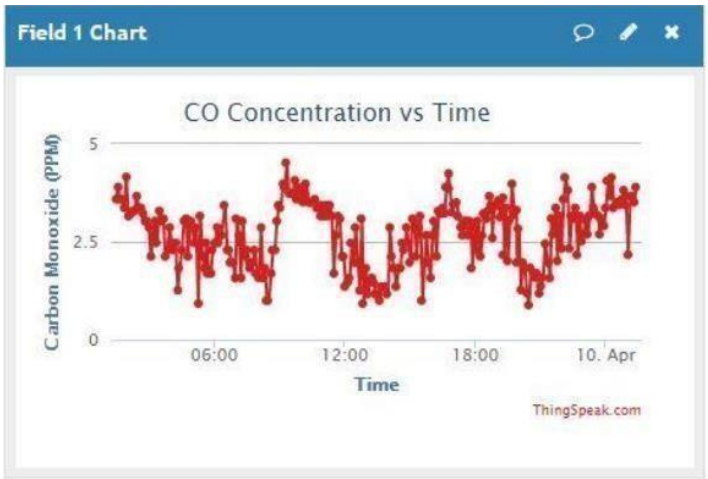


Fig. 3 *Real Time Air Quality Monitoring System Example of CO Monitoring Result Interface*

Table 1 shows the comparison between the existing system with the proposed system.

Table 1 *Comparison between the existing system with the proposed system*

Features/System	Air Pollution Monitoring System using Edge Computing	An Internet of Things-based Air Pollution Detection System	Real Time Air Quality Monitoring System	Air Pollution Monitoring System using Edge Computing	AirGuardian: Real-time IoT-Based Air Pollutant Monitor System
Log In	No	No	Yes	Yes	Log In
Register User/Admin	No	No	Yes	Yes	Register User/Admin
On/Off Remote	No	No	No	Through website	On/Off Remote
Air Monitor Sensors and Microcontroller used	Raspberry Pi, Grove dust sensor and Grove air quality sensor	Arduino Uno, MQ2 and MQ9 gas sensors	Arduino Uno, SHARP DN7C3CA006		Air Monitor Sensors and Microcontroller used
PM2.5 Sensor, SHARP GP2Y1010AU0F					PM2.5 Sensor, SHARP GP2Y1010AU0F
PM10 Sensor	MQ135 Gas sensor, Arduino Uno.				PM10 Sensor
User Interface	Yes	Yes	Yes	Yes	User Interface
Receive/Sending Notification	Yes	Yes	Yes	Yes	Receive/Sending Notification
Database Manipulation	Yes	Yes	Yes	Yes	Database Manipulation
Device Alarm	No	Yes	No	Yes	Device Alarm
Report Generating	No	Yes	Yes	Yes	Report Generating

3. Methodology

The Agile methodology has been used to develop the AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System. It combines the iterative and incremental approach for software development and project management as it emphasizes flexibility, collaboration, customer feedback, and rapid, incremental delivery of software to ensure the system works properly. The Agile methodology produces results in higher quality software while allowing flexibility to respond to evolving user requirements [9].

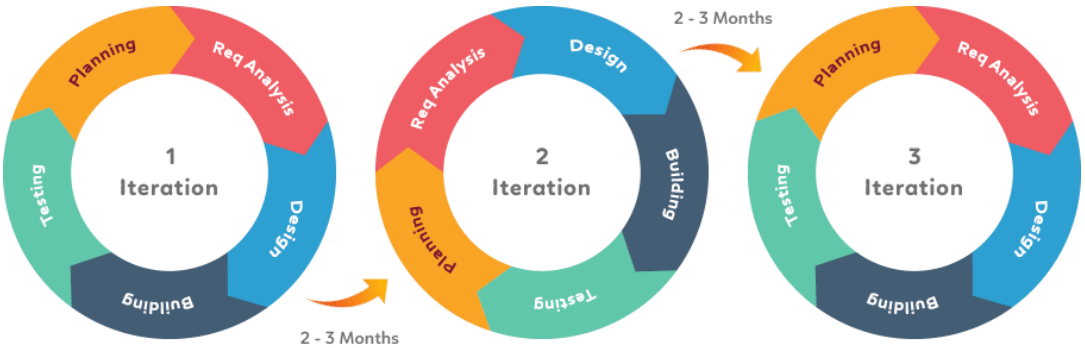


Fig. 4 *Agile Model Methodology*

Applied Information Technology and Computer Science Vol. 6 No. 1 (2025) p. 1871-1887

3.1 Planning Phase

The planning phase is conducted to gain information and requirements for developing the system from the stakeholder. For developing AirGuardian: Real-Time IoT-Based Air Pollutant Monitor system, an interview has been conducted to get all the data and requirements for the system development. A study on reviewing the existing system has also been in this phase to learn more about any features that should have been included inside the developed system that can be used as reference. Furthermore, a Gantt Chart has also been made as a guide timeline for the system development so the project could be finished on time.

3.2 Analysis Phase

In the Analysis phase, an analysis for the developed system is made and involves gathering and documenting every detailed requirement. The analysis will produce the Data Flow Diagrams (DFD) which shows the data flow on the system, Entity-Relationship Diagrams (ERD) which shows about the relationship between each entity inside the system and the flowcharts which shows the flow of the system. This helps with the system development that will meet the requirements.

3.3 Design Phase

The design phase is a phase of designing the system based on the stakeholder requirement from the data gathered from the interview. In the design phase, the device is designed based on portability and their functions so there would not be any problems for every module to work properly. In this phase, a system architecture is created to foresee the whole system function. Additionally, this phase also includes the designing of the database which consists of the data dictionary and the database scheme which will work as a guidance during database development.

3.4 Implementation Phase

The Implementation phase in Agile Model is focusing on the system development for the proposed system. The development includes writing the codes for the system interface by using HTML, PHP and CSS, database construction by using MySQL and the Arduino device that consist of gas sensor are written by using Arduino IDE. This phase will bring the proposed system to life based on the user requirements and the design is based on the plan.

3.5 Testing Phase

The testing phase is the final phase in the Agile Model. A test is conducted to ensure every function of the system is operated properly without any problems. The tests include the functional and non-functional requirements of the system which is to identify if there is any glitches or error on the system operation. A User Acceptance Testing is also performed to ensure the proposed system meets user expectations before delivered. This phase is important to ensure the developed system reaches the goals of the project.

3.6 System Development Workflow

Table 1 *System Development Workflow*

Phase	Task	Output
Planning	• Proposed the project. • Determine the project schedule, activities, and output. • Interviewing stakeholders.	• Project proposal. • Develop Gantt chart. • User requirements of the system.
Analysis	• Analyze user requirements of system. • Analyze existing system. • Compare the proposed system with existed system.	• Data Flow Diagrams (DFD). • Entity-Relationship Diagrams (ERD). • Flowcharts.
Design	• Designing the device. • Designing the interface. • Designing the database.	• Internet of Things (IoT) System Architecture. • Data schema. • Data Dictionary. • System interface.
Implementation	• Writing code for Arduino. • Writing code for the webpage interface. • Writing code for the database. • Connecting the device with the webpage and database.	• Device functionalities. • Webpages interface. • Database.
Testing	• Testing the sensor function. • Functional testing • Testing the database to record data. • User Acceptance Testing.	• Error code modifying. • System tested.

4. Analysis And Design

This section discusses the analysis and design which includes the system requirement consist of functional and non- functional requirements, Data Flow Diagram (DFD), Entity Relationship Diagram (ERD), flowcharts, the system architecture, and the user interface design.

4.1 System Analysis and Design

Requirement analysis involves identifying user expectations and needs for a completely new or modified system. This analysis is important in system development as it serves to determine the system success. System requirements can be categorized into two types which are functional requirements and non-functional requirements.

Applied Information Technology and Computer Science Vol. 6 No. 1 (2025) p. 1871-1887

4.2 Functional and Non-Functional Requirements

Table 3 *Functional Requirements*

Modules	Functions
Registration and Login/Logout Module	The system will allow the user to register their account while the administrator monitor every recorded data and allow the administrator and user to Login and Logout into the system by using the ID and password.
Login/Logout Module	The system will allow the admin and user to Login and Logout into the system by using the ID and Password.
Homepage Modul	The system will display the user about the air quality of the school compound.
Start On/Off Module	The admin can set up the starting time for the device to turn on and turn off.
Notification Module	The system will send a notification the user through E-Mail if the air pollution reading is high.
Generating Report Module	The system will allow the admin to generate report from the gathered data

4.3 Data Flow Diagram (DFD) Level 0

A data flow diagram is a logical model of the flow of data through a system that shows how the system's boundaries, processes, and data entities are logically related [10]. For AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System, it has 8 processes which are registration, login, profile management, air pollution read, device status, user account management, air reading report generating and notification and alert. In addition, it also shows 6 database tables which are the admin, student, gas_read, reports and device.

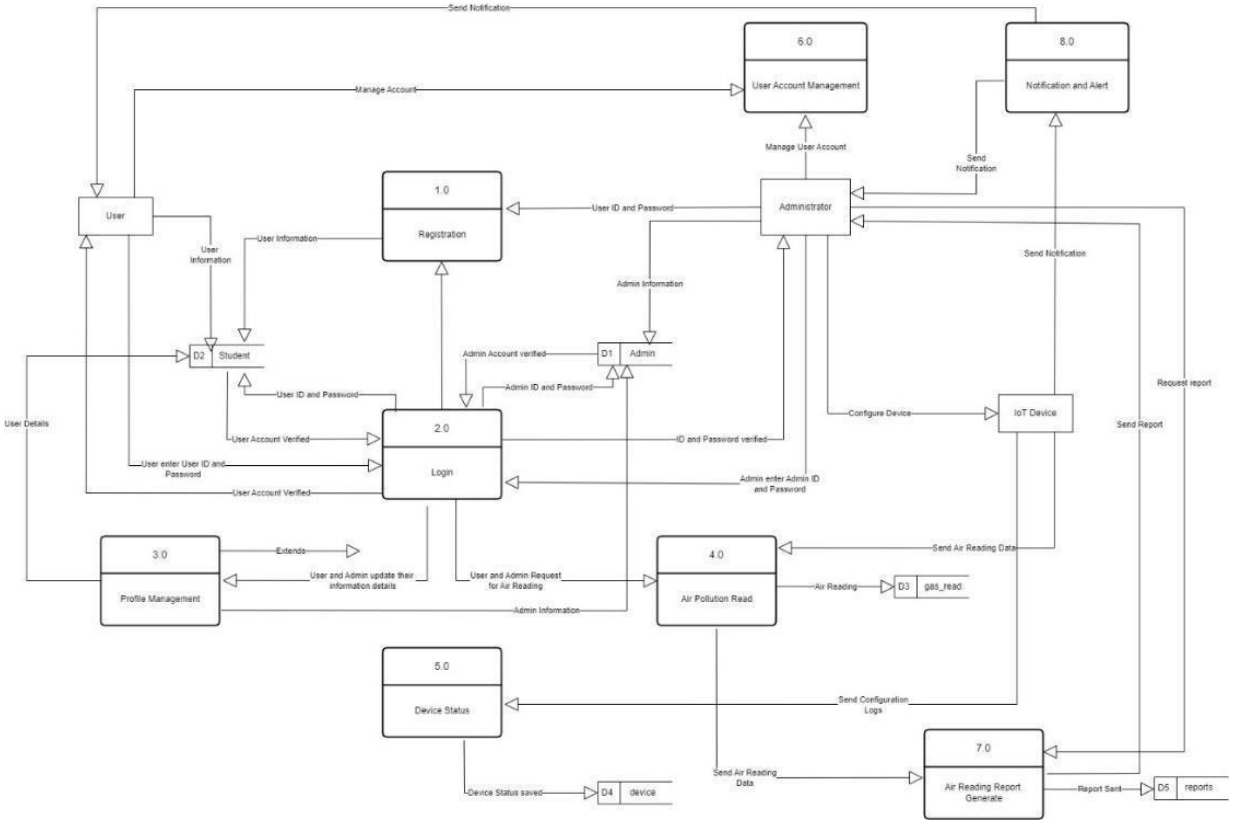


Fig. 5 *Data Flow Diagram Level 0 for AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System*

4.4 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) is a graphical representation used during system analysis to shows the relationship between the entities stored inside the database. It is a methodology that describes how the data of a system is stored at a high level of detail [11]. The Entity Relationship Diagram (ERD) is being used to visual representation of the entities, relationships, and attributes within a database system. For the AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System, it has 6 entities in the design of the system database that will be used in the database, which is Admin, User, info_details, air_read, reports and Device_status.

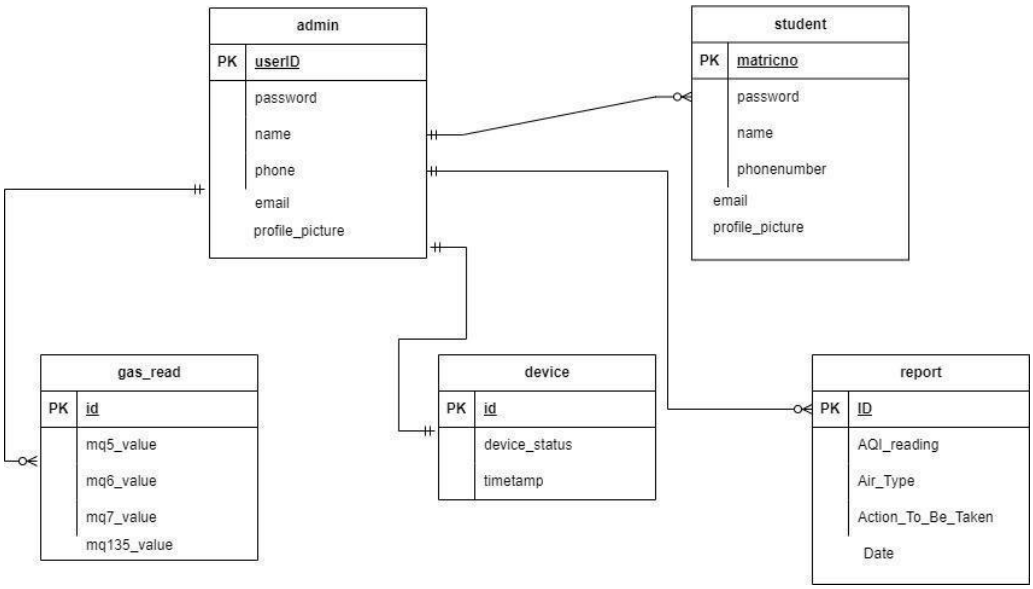
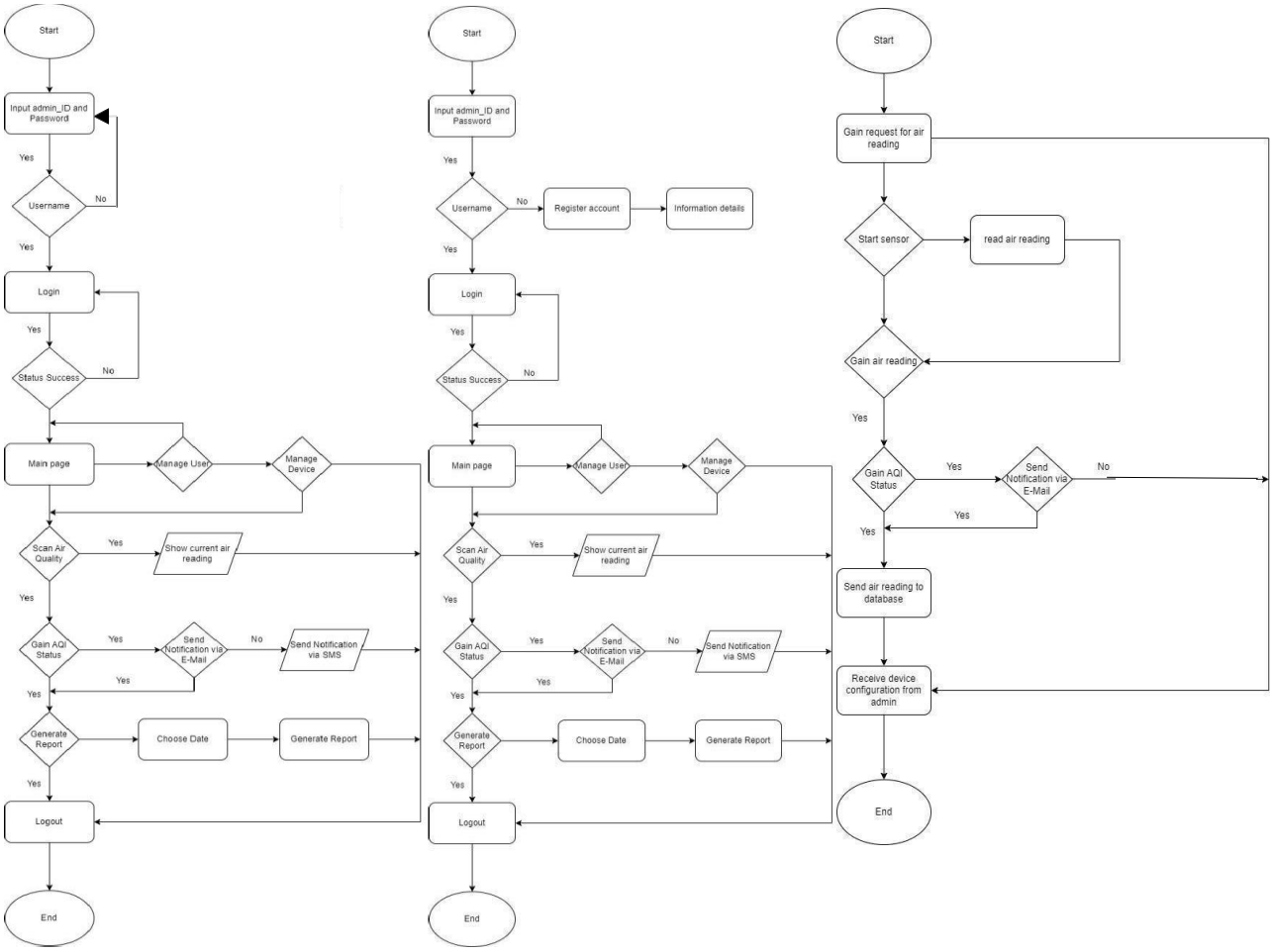


Fig. 6 *Entity Relationship Diagram for AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System*

4.5 Flowchart

A flowchart is a diagram that is being used to illustrate and shows how the data flows in the system and used to shows every process that occurs in the system. The flowchart serves as the central design document around which systems analysts, computer programmers, and end users communicate, negotiate, and represent complexity [12]. Therefore, a flowchart will simplify the complex process into a diagram that can be easily understood by anyone. In the AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System, the flowchart will be divided into three parts which are the user, the administrator and the IoT device itself to outline the development process of the system.

Applied Information Technology and Computer Science Vol. 6 No. 1 (2025) p. 1871-1887



(a) (b) (c)

Fig. 7 *AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System (a) User flowchart; (b) Admin flowchart; (c) IoT Device flowchart*

4.6 Internet of Things (IoT) System Architecture

Internet of Things (IoT) system architecture is a description on how the system will interact to each other. An open and comprehensive network of intelligent objects that have the capacity to auto-organize, share information, data and resources, reacting and acting inface of situations and changes in the environment [13]. It shows the interaction between each entity to the system. The user and administrator can use the system by using the pcand gain notification by email through their device. The IoT device will interact with the system to display the air reading and the system will interact with the database to store data.

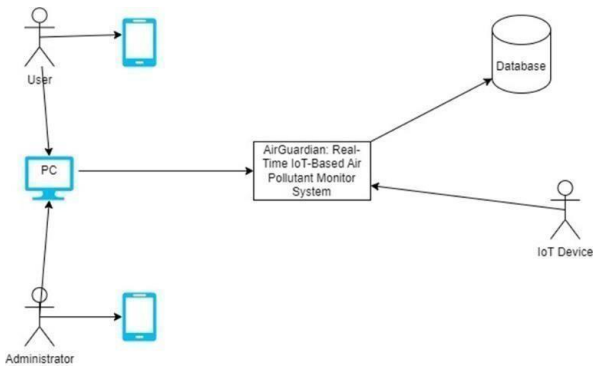


Fig. 8 *AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System IoT System Architecture*

4.7 Interface Design

Interface design is developed to ease the user of the system to navigate and use this system without any problems. The design should be user-friendly and easy to navigate from page to page. The system interface should be designed and developed in the system development especially on AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System.

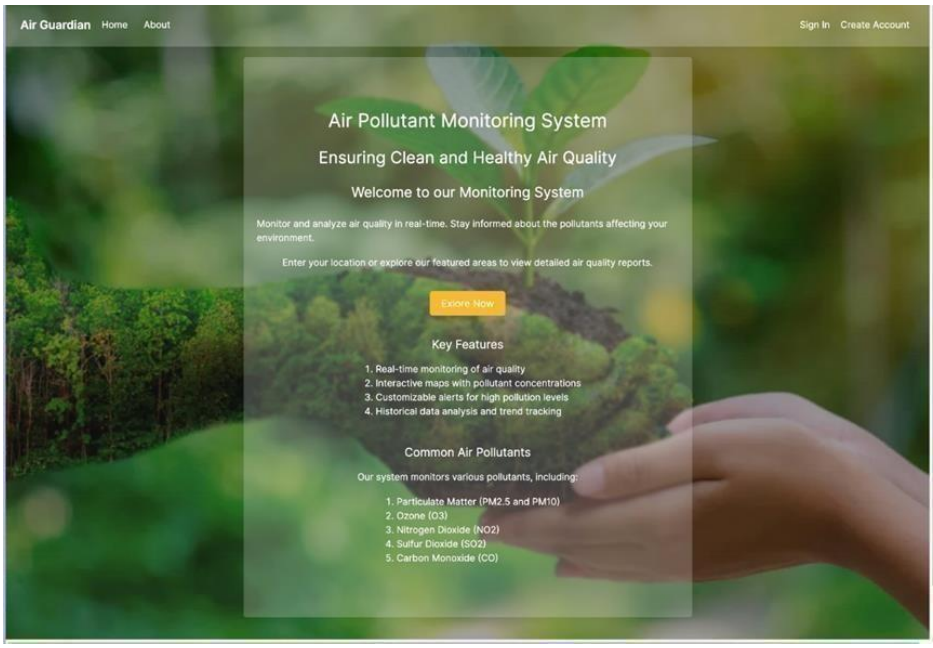


Fig. 9 *AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System Home Interface*

5. Result and Discussion

This section will discuss the results of the development for AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System which includes all the modules interface, the hardware assembly, circuit diagram and the testing of the system. This section is crucial as to determine whether the system meets the user requirements and able to operate properly without any problems. The user also shows feedback to evaluate the system performance and ready to be deployed.

5.1 User Account Registration

Figure 10 below shows the interface for user registration. Users are requested to enter a student number, full name, e-mail, phone and password. If the user has not yet registered, they can do so through the user registration to log into the system as an administrator.

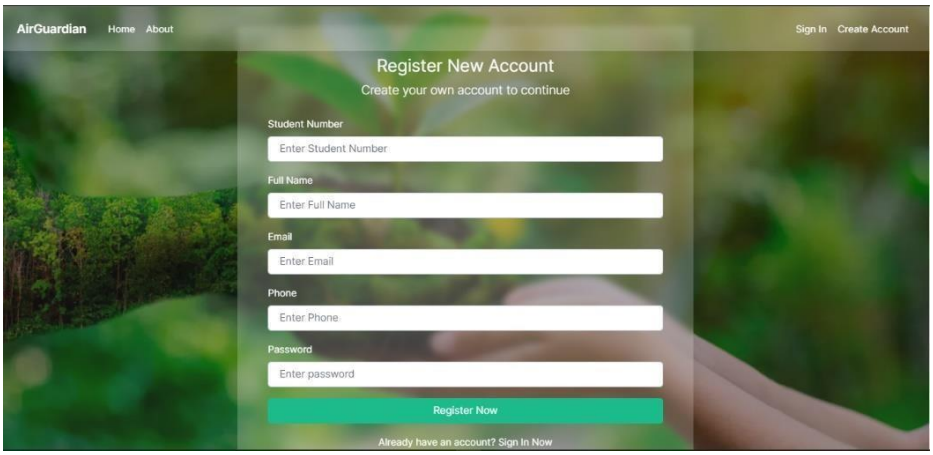


Fig. 10 *User Registration Interface*

Applied Information Technology and Computer Science Vol. 6 No. 1 (2025) p. 1871-1887

5.2 User login Interface

Figure 11 below shows the user login page of the system intended to use the AirGuardian: Real-Time IoT Based Air Pollutant Monitor System. Users need to log in first to use the system by entering their ID and password. The user also can switch between ordinary user login or administrator login.

Fig. 11 *User and Admin Login Interface*

5.3 System Dashboard Interface

Figure 12 below shows the dashboard interface of AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System. The dashboard shows the AQI Reading, Air Type, Action To Be Taken, Monitoring Chart to visualize the air reading graph, the sensor reading and even maps to locate the system user location. Then, the button Report is to navigate to report page for download the report.

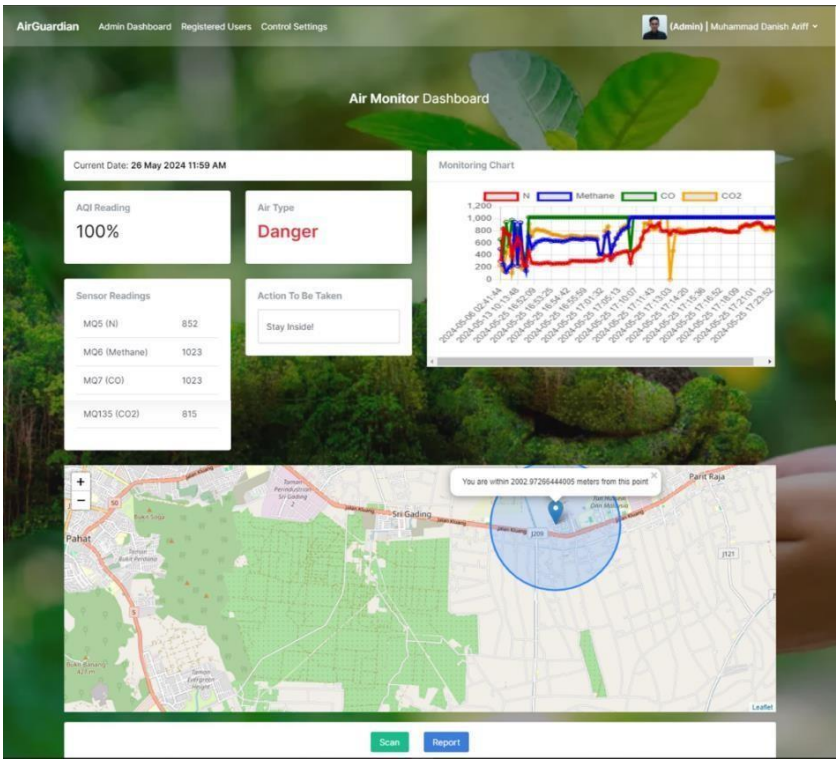


Fig. 12 *AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System Dashboard Interface*

5.4 Start On/Off Interface

Figure 13 shows the Start On/Off Interface. The function of this module is to manage and configure the device either to turn it on or off the air reading on the IoT device remotely. This can ease the admin to manage the device in a single system.

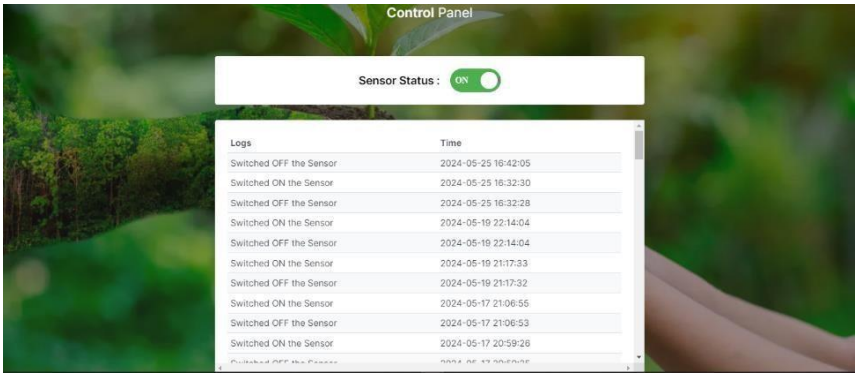


Fig. 13 *AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System Start On/Off Interface*

5.5 Notification System

Figure 14 below shows the system notification for AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System which shows the alert when the system detected for air pollution.

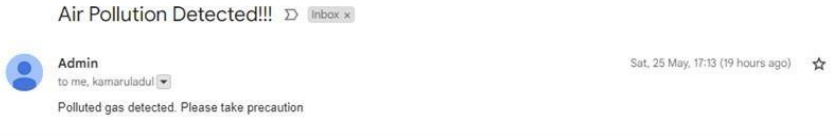


Fig. 14 *System Notification*

5.6 Generate Report Interface

Figure 15 below shows the generated report for AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System. The dashboard shows the daily reading chart and the record of the air reading. Users can download the file in .csv through the download button for further research or data visualization.

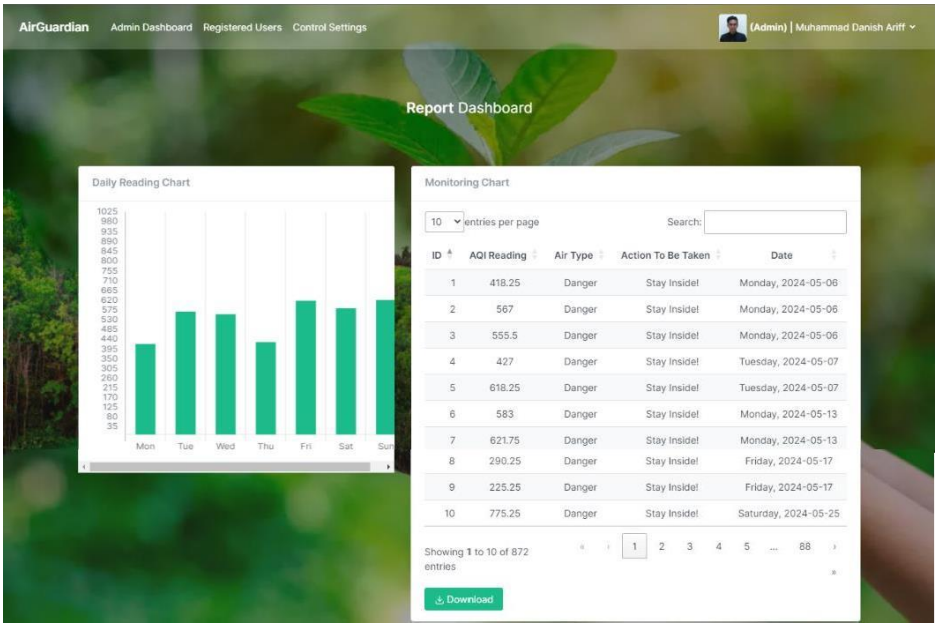


Fig. 15 *Report Dashboard for AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System*

5.7 Device Circuit Diagram

Figure 16 shows the circuit diagram of AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System. The circuit can be by combining the Arduino Uno with ESP8266 NodeMCU. The sensors are connected to Arduino Uno via Analog pin. The relay is used as a circuit breaker to turn off the device when get signal from the system. All data are sent to the NodeMCU and then will be transmitted to the system via Wi-Fi Connection and will be displayed on the webpage and LCD panel. Figure 17 shows the final product of AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System device.

Applied Information Technology and Computer Science Vol. 6 No. 1 (2025) p. 1871-1887

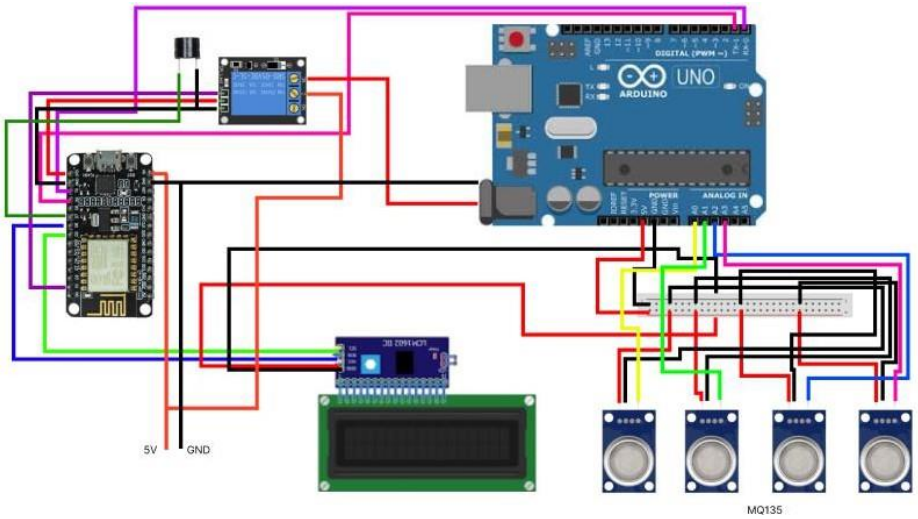


Fig. 16 *AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System Circuit Diagram*

Fig. 17 *AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System device*

5.8 System Functional Test

System Functional Test is a critical phase in software development where the complete integrated system is tested to verify that it meets the specified requirements. The AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System has five (5) main modules which are the Registration and Login/Logout module, the dashboard module, the Start On/Off module, Notification module and Generate Report module. Each module in this system has been tested to ensure that every module functions well. Functional testing of the system was carried out repeatedly to ensure that the system can produce accurate results.

Table 4 *Test Casefor AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System*

Test Modules	Description	Expected Result	Pass/Fail
User Registration and Login/Logout	• To check whether user can register for an account and Login into system	• The user should be able to create for an account	• Pass
Dashboard	• To check whether the system displayed the air reading in the dashboard	• The system should be able to display the air reading in the dashboard	• Pass
Start On/Off	• To check whether the system can turn on and off the air reading on IoT device remotely.	• The system should be able to turn on and off the air reading on IoT device remotely.	• Pass
Notification	• To check whether the system able to send notification to user through email if the air reading is high	• The system should be able to send notification to user through email if the air reading is high	• Pass
Generate Report	• To check whether the system able to display the record	• The system should be able to display the record	• Pass

5.9 User Acceptance Test (UAT)

User acceptance testing or end-user testing is the final stage of testing in application software development where the software is tested in the real world by the target users. [14]. The figures (Figure 18-Figure 24) below shows the results of user testing collected through Google Forms. The development of this system received very positive feedback as users were able to successfully register accounts and log into the system using their registered ID and passwords. Additionally, users gave excellent responses regarding the login and registration, profile section, which can be edited and updated, the dashboard shows the air reading, the notification section, which sends alerts to users when the air quality is unhealthy, and the generate report. Lastly, users rated highly the system AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System.

Fig. 18 *User testing for Registration*

Applied Information Technology and Computer Science Vol. 6 No. 1 (2025) p. 1871-1887

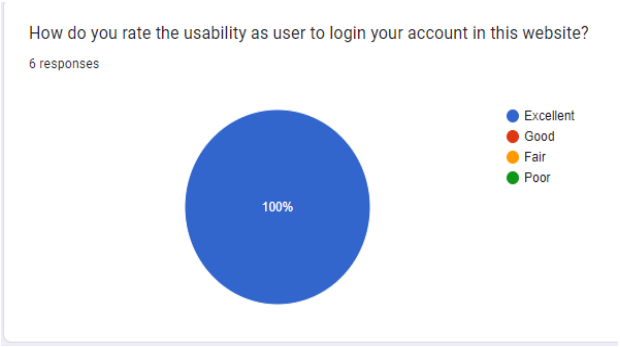


Fig. 19 *User testing for Login*

Fig. 20 *User testing for Dashboard*

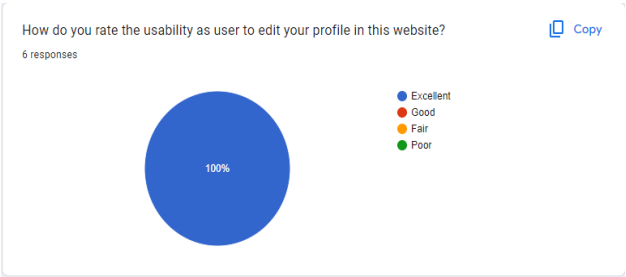


Fig. 21 *User testing for Profile Edit*

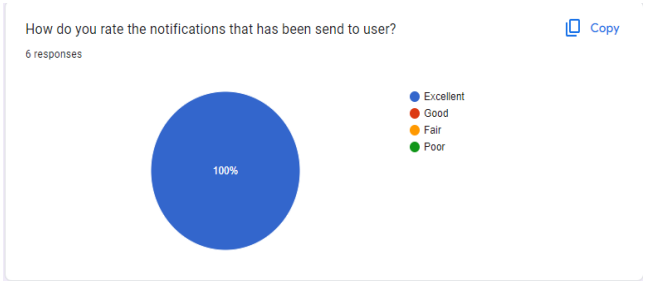


Fig. 22 *User testing for Notifications*

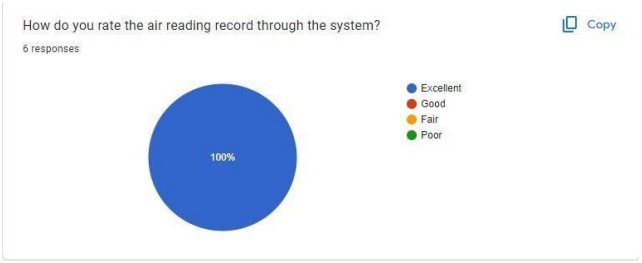


Fig. 23 *User testing for Report*

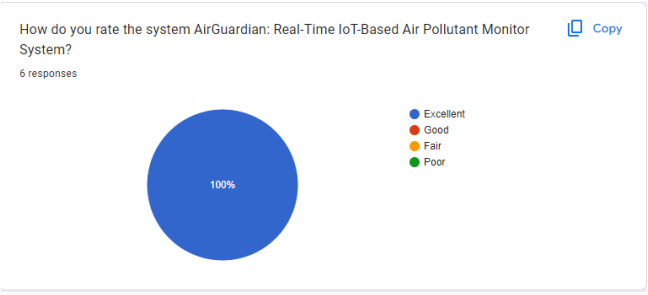


Fig. 24 *User testing for overall system performance*

6. Conclusion

To conclude, the development of AirGuardian: Real-Time IoT-Based Air Pollutant Monitor System will help the school especially the students and the school staffs to be aware on the air pollution and the system could help on monitoring and able to notify everyone if the air pollution occurs as any counter measures could be taken quickly. Additionally, the system also could enhance the user awareness of the importance of implementing Internet of Things (IoT) technology in our daily life.

Acknowledgement

The authors would like to thank 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:*** *Muhammad Danish Ariff Zainal Abidin, Suziyanti Marjudi;* ***data collection:*** *Muhammad Danish Ariff Zainal Abidin;* ***analysis and interpretation of results:*** *Muhammad Danish Ariff Zainal Abidin, Author Y, Author Z;* ***draft manuscript preparation:*** *Muhammad Danish Ariff Zainal Abidin. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] A. Rejeb, K. Rejeb, S. Simske, H. Treiblmaier, and S. Zailani, "The big picture on the internet of things and the smart city: a review of what we know and what we need to know," *Internet of Things*, vol. 19, p. 100565, 2022.

[2] A. Sadatshojaie and M. R. Rahimpour, "CO2 emission and air pollution (volatile organic compounds, etc.)–related problems causing climate change," in *Current Trends and Future Developments on (Bio-) Membranes*, Elsevier, 2020, pp. 1–30.

[3] N. L. A. Rani, A. Azid, S. I. Khalit, H. Juahir, and M. S. Samsudin, "Air Pollution Index Trend Analysis in Malaysia, 2010–15," *Polish Journal of Environmental Studies*, vol. 27, no. 2, 2018.

[4] M. Singh, M. Kumari, and P. K. Chauhan, "IoT Based Air Pollution Monitoring System using Arduino," *International Research Journal of Engineering and Technology (IRJET)*, 2019.

[5] E. Z. Abidin, S. Semple, I. Rasdi, S. N. S. Ismail, and J. G. Ayres, "The relationship between air pollution and asthma in Malaysian schoolchildren," *Air Quality, Atmosphere & Health*, vol. 7, pp. 421–432, 2014.
[6] K. Biondi *et al.*, "Air pollution detection system using edge computing," in *2019 International Conference in Engineering Applications (ICEA)*, pp. 1–6, IEEE, 2019.
[7] N. A. M. Saari, H. Kamaludin, N. Z. M. Safa, and I. R. A. Hamid, "An Internet of Things-based Air Pollution Detection System," *Journal of Soft Computing and Data Mining*, vol. 3, no. 1, pp. 78–85, 2022.
[8] A. A. Zaini, *Development of Real Time Air Quality Monitoring System*, 2017.
[9] N. A. Bonner, J. T. Teng, and S. Nerur, "The perceived advantage of agile development methodologies by software professionals: Testing an innovation-theoretic model," 2010. [Online]. Available: https://aisel.aisnet.org/amcis2010/93/
[10] W. S. Davis, "Data flow diagrams," in *The Information System Consultant's Handbook*, CRC Press, 2019, pp. 175–188.
[11] J. Frantiska Jr and J. Frantiska, "Entity-relationship diagrams," in *Visualization Tools for Learning Environment Development*, pp. 21–30, 2018.
[12] N. Ensmenger, "The multiple meanings of a flowchart," *Information & Culture*, vol. 51, no. 3, pp. 321–351, 2016.
[13] S. G. H. Soumyalatha, "Study of IoT: understanding IoT architecture, applications, issues and challenges," in *1st International Conference on Innovations in Computing & Networking (ICICN16)*, *International Journal of Advanced Networking & Applications*, 2016, p. 478.
[14] H. K. Leung and P. W. Wong, "A study of user acceptance tests," *Software Quality Journal*, vol. 6, pp. 137–149, 1997.