COLLEGE OF COMPUTER SCIENCE &

INFORMATION SYSTEMS

NAJRAN UNIVERSITY

(SENT) Smart Environmental Navigation and Tracking Robot

Project Members:

1- Mohsen Zaid Hermes Al Jarah

2- Ali Qamzan Ahmed Al Qamzan

3- Ali Hadi Al-abbas

4- Ali Misfer Alhaddad

5- Faris Hussain Mana Al-abbas

6- Mesfer Abdullah Mesfer Alslom

Project Supervisors

Prof. Asadullah Shaikh.

Abstract

The Smart Environmental Navigation and Tracking Robot (SENT) is a mobile safety system designed to work inside the College of Computer Science and Information Systems at Najran University. The robot detects two main hazards: gas leaks and water. It uses a Raspberry Pi 4 and several sensors, including a gas sensor, water sensor, ultrasonic sensor, and an HD camera. A Flask server runs on the robot and works as the main controller.

The system provides live video so users can watch robots in real time. When gas or water is detected, the robot immediately sends alerts to the administrator through email and WhatsApp. The Flutter dashboard has two user types: the administrator, who can see and delete logs, receive snapshots, and get all alerts; and the regular user, who can only Receive notifications and watch the live stream.

The robot can move in two ways: manual control or following a black line using line-tracking sensors. All detected events appear on the dashboard so users can respond quickly and stay aware of the situation.

Acknowledgment

First, we thank God for giving us the ability, knowledge, and patience to complete the Smart Environmental Navigation and Tracking Robot (SENT) project.

We would like to express our gratitude to our supervisor, Prof. Asadullah Shaikh, for his continuous guidance, support, and helpful feedback throughout the development of this project. His direction played a major role in making this work possible.

We also thank the Faculty of Computer Science and Information Systems at Najran University, along with all the instructors who have taught and supported us since the beginning of our academic journey. Their efforts helped us gain the skills needed to complete this project.

Finally, we thank our families for their constant encouragement and support, which motivated us to keep going and give our best.

Table of Contents

List of Figures:

List of Table

Chapter 1: Introduction.

1.1 Introduction of the Project.

Modern monitoring systems increasingly rely on automation to improve safety and reduce the need for manual inspection. In indoor environments such as the College of Computer Science and Information Systems at Najran University, fast detection of environmental hazards is essential, especially in cases of gas leaks or water that may cause electrical risks or accidents. (Reference: [1], [2])

This project presents the Smart Environmental Navigation and Tracking (SENT) Robot, a mobile safety system designed to detect two primary hazards: gas leaks and water on the floor. The robot uses a gas sensor capable of detecting several flammable gases, including LPG, propane, methane, hydrogen, alcohol vapors, smoke, and i-butane, allowing it to identify different types of gas leaks inside indoor spaces. (Reference: [1])

The robot is built on the XiaoR Geek Smart WiFi Robot Kit using a tank-chassis platform equipped with a Raspberry Pi 4 as the main controller. It integrates a gas sensor, water sensor, ultrasonic sensor, and HD camera to provide real-time monitoring and instant alerts when hazards are detected. (Reference: [1], [2])

A Flask-based backend processes sensor data, handles notification logic, and communicates with the dashboard. When gas or water is detected, the system sends immediate alerts to the administrator through WhatsApp using Twilio's API and through email for quick awareness. (Reference: [3], [4], [6])

For continuous real-time monitoring, the robot uses MJPEG-Streamer to broadcast a live video feed accessible to users through a Flutter dashboard. The system includes two user roles: admin, who can clear logs, receive snapshots, and access all alerts; and regular user, who can only Receive notifications and the live camera stream. (Reference: [5])

The robot supports two movement modes: manual remote control and black-line following using infrared sensors, allowing it to operate flexibly in different indoor scenarios. (Reference: [1])

Through the combination of hardware and software components, this project offers a reliable and practical monitoring solution that improves safety inside Najran University and supports fast response to environmental hazards. (Reference: [1], [2])

1.2 Problem Statement.

Monitoring environmental hazards inside buildings often depends on manual inspection, which can be slow, inconsistent, and sometimes unsafe. In areas like hallways, labs, and classrooms, water or gas leaks may go unnoticed, leading to electrical risks, slipping accidents, and equipment damage. Relying only on humans to check these places makes the process less efficient and increases the chance of delayed detection.

Traditional surveillance systems provide only camera feeds and do not actively detect hazards. They cannot identify gas leaks or water on the ground, and they require a human operator to constantly monitor them. This creates gaps in safety, especially in large or busy indoor environments such as university buildings.

To address this issue, the SENT robot provides an automated monitoring solution. It detects gas leaks and water using sensors connected to a Raspberry Pi 4. The robot sends instant notifications to the administrator through a dashboard, email, and WhatsApp, ensuring that hazards are noticed immediately. With live video streaming, manual control, and black-line following, the robot can move inside the building and provide real-time awareness without constant human supervision.

This system reduces the need for manual checks, improves safety, and provides faster response to environmental hazards inside Najran University's College of Computer Science and Information Systems.

1.3 Scope.

This project presents a smart monitoring robot designed to detect water leaks and gas leaks inside the College of Computer Science and Information Systems at Najran University. The robot uses different sensors connected to a Raspberry Pi 4 to monitor the environment in real time. When gas or water is detected, the system immediately sends alerts to the administrator through WhatsApp, email, and the Flutter dashboard. The robot also provides a live camera stream, manual movement control, and a black-line-following mode, which allows it to move safely inside the building. This system reduces the need for manual inspection and helps improve safety inside the college. Future improvements may include fire detection, better navigation, and adding more environmental sensors.

1.4 objectives

The main objective of this project is to develop a smart monitoring robot that can detect gas leaks and water inside the College of Computer Science and Information Systems at Najran University. The system improves safety by providing real-time monitoring and instant alerts to the administrator.

Key Objectives:

1. **Water Detection** – Detecting floodwater and protecting the building and flooring
2. **Gas Detection** – Identify different types of gas leaks using a multi-gas sensor that can detect LPG, propane, methane, hydrogen, alcohol vapors, smoke, and i-butane.
3. **Live Monitoring** – Providing continuous live video streaming from the robot's camera via the Flutter control panel.
4. **Real-time Alerts** – Send instant notifications via SMTP/HTTP for quick maintenance response and Twilio Wahhtsap.
5. **Role-Based Dashboard** – Provides two user roles: Administrator: Full permissions (Receive /delete notifications + Send photo to email+ live stream) User: Limited permissions (Receive notifications + live stream).
6. **Local Data Logging** – Saving detected events in the system for faster review and response.
7. **Energy-efficient Design** – Ensure low power consumption and cost effectiveness.
8. **Future Scalability** – Support modular upgrades for fire, gas leak, and security detection.

This system improves campus safety by automating the detection of gas leaks and water , reducing the need for manual inspection and helping maintain a safer environment for everyone

1.5 Project Features

This project includes several features that help improve safety inside the College of Computer Science and Information Systems. The SENT robot combines sensors, live monitoring, and automated alerts to detect gas and water hazards efficiently.

1. **Water Detection**– The robot uses a floor-contact water sensor to detect moisture and trigger an alert.
2. **Gas Detection** – MQ-2 sensor identifies multiple gases and sends hazard events to the server.
3. **Line-Tracking Mode** – The robot can follow a black line using follow line sensors when line-following is enabled.
4. **Real-Time Alerts**– The system sends notifications through WhatsApp, email, and dashboard logs when gas or water is detected.
5. **Robotic Arm** – Allows object manipulation when required
6. **Built-in Wi-Fi Connectivity**– The Raspberry Pi 4 connects the robot with the dashboard and live video stream.
7. **Energy-efficient & Cost-effective** – Low power consumption for sustainable use.

This system automates surveillance, improving campus safety and hygiene efficiently.

1.6 Project Advantages and Disadvantages

1.6.1 Advantages

A. Improved Safety: Detects gas leaks and water early, helping prevent accidents and protecting equipment.
B. Reduces Manual Inspection: Automates hazard detection, which saves time and reduces the workload on staff.
C. Real-Time Alerts: Sends instant notifications through WhatsApp, email, and the dashboard, allowing quick response to hazards.
D. Live Monitoring: Provides a continuous camera stream so users can monitor the robot's surroundings in real time.
E. Role-Based Access: Admin and user roles improve system organization and prevent unauthorized actions.
F. Dual Movement Modes: Supports manual control and line-following, giving flexibility in different indoor environments.
G. Scalable Design: The system can be expanded in the future with additional sensors such as fire detection or better navigation.
H. Local Data Logging: Keeps a record of detected events for review and improved incident tracking.

1.6.2 Disadvantages

a) Initial Hardware Cost: Components such as Raspberry Pi, sensors, and camera increase the starting cost of the project.
b) Indoor Limitation: Designed mainly for indoor environments; outdoor operation would require stronger sensors and protection.
c) Sensor Sensitivity: Gas and water sensors may occasionally produce false readings due to environmental noise or humidity.
d) Maintenance Requirements: Sensors need periodic calibration, and the robot requires physical checks to ensure proper movement.
e) Network Dependency: Real-time streaming and alerts depend on a stable Wi-Fi connection; weak signals affect performance.

Chapter 2: Background Study.

2.1 Overview

The rapid advancement of technology has enabled the creation of automated systems that improve safety, monitoring, and operational efficiency. In environments such as educational institutions, maintaining safety from hazards like water and gas leaks is essential. This project aims to utilize a Raspberry Pi–based robotic platform equipped with environmental sensors to detect water and gas leakage within the College of Computer Science and Information Systems. By applying modern automation techniques, the system provides real-time monitoring and instant alerts, reducing the need for manual inspection and helping prevent potential risks [9], as shown in Figure 1.

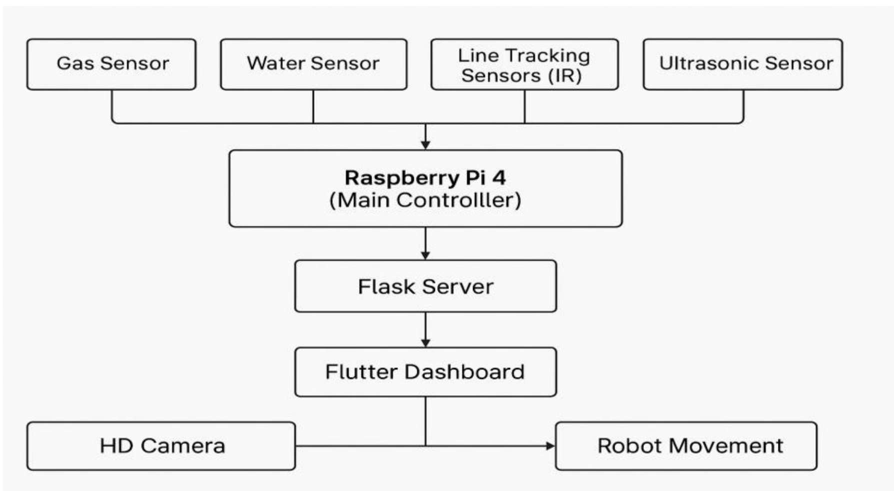

Figure 1 Overview system

2.2 Related Systems and Applications

Several existing systems use different technologies for object detection, environmental monitoring, and automated cleaning. These systems employ various sensors, AI algorithms, and robotics to achieve their goals.

2.2.1 Smart Cleaning Robots

Smart cleaning robots such as iRobot Roomba and Xiaomi Roborock use AI-powered image processing and sensor navigation to autonomously sweep floors. The cleaning robots use LiDAR, infrared sensors, and cameras to identify dirt, navigate spaces, and optimize their cleaning paths [9][10] as shown in figure 2.

Figure 2 Smart Cleaning Robots [10]

2.2.2 Water Leak Detection Systems

Water leak detection systems, widely used in smart buildings, rely on IoT-sensors to identify and alert users of water spillage or leakage. They make use of WiFi-based alerts and can be incorporated into home automation systems [11] as shown in figure 3.

Figure 3 Water Leak Detection System [11]

2.2.3 Industrial Surveillance Systems

Industries use AI-driven monitoring systems with real-time image recognition to monitor work environment conditions. The systems watch over video feeds and sensor data to detect hazardous conditions and alert against accidents [12] as shown in figure 4.

Figure 4 Industrial Surveillance [12]

2.2.4 IoT-Based Waste Management Systems

There are some cities with IoT-based waste management systems where smart bins monitor the levels of garbage and optimize waste collection routes. They use ultrasonic sensors and cloud monitoring to maximize efficiency [13] as shown in figure 5.

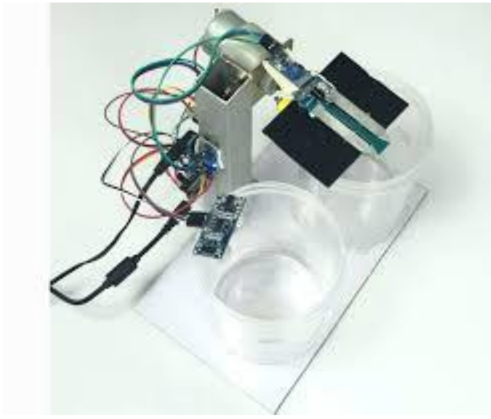

Figure 5 IoT-Based Waste Management Systems [13]

2.2.5 SENT Project Robot

Our system integrates multiple sensing and monitoring functions to improve safety inside educational facilities. Unlike traditional monitoring systems, it specifically detects **water** and **gas leaks** in classrooms and hallways using real-time sensor data and live camera streaming. This helps maintain a safer environment and reduces the need for manual inspection, as shown in Figures 6 and Table 1.

Figure 6 SENT

2.3 A Comparison of Existing Systems with SENT

This table compares existing monitoring systems with the SENT robot to show how our system combines multiple safety features into one integrated solution.

Table 1 Comparison Table

Feature	Smart Cleaning Robots	Water Leak Detection	Industrial Surveillance	IoT-Based Waste Management	SENT
Uses Sensors for Monitoring	Yes	Yes	Yes	Yes	Yes
Real-time Alerts	No	Yes	Yes	Yes	Yes
Mobile App & Dashboard	Yes	Yes	No	Yes	Yes
Focused on Educational Environments	No	No	No	No	Yes
Multiple objective detection technique	No	No	No	No	Yes
Live Video Streaming	No	No	No	NO	Yes

Chapter 3: Methodology and Design.

The methodology adopted for this project follows the **Waterfall Model**, which is a linear and sequential approach to software development. This model ensures that each phase is completed before moving on to the next, making it ideal for structured and well-documented projects. The Waterfall Model consists of five primary phases: Requirement Analysis, System Design, Implementation, Testing, and Deployment [14] as shown in figure 7.

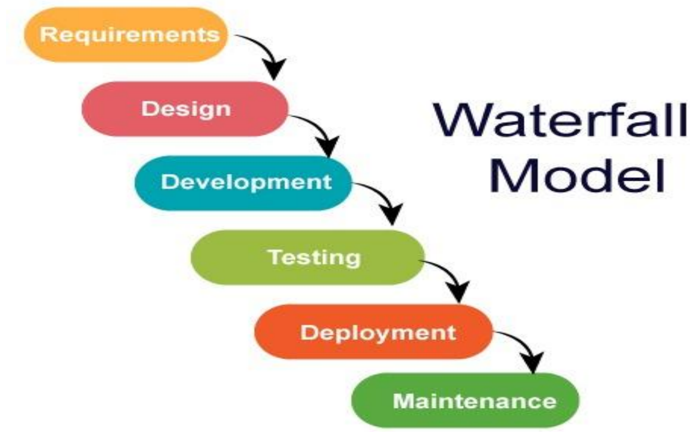

Figure 7 Waterfall Model diagram

3.1 Waterfall Model Phases

3.1.1 Requirement Analysis

In this phase, we define the key objectives and requirements of the project. This includes:

- Detecting **water** using a water sensor.
- Detecting **gas leaks** using a multi-gas sensor (LPG, propane, methane, hydrogen, alcohol vapors, smoke, i-butane).
- Providing **real-time video streaming** using an HD camera.
- Allowing **manual control** and **black line following** using IR sensors.
- Sending **instant alerts** through WhatsApp, email, and dashboard notifications.
- Using **Flask** as the server backend and **Flutter** as the dashboard interface.
- Ensuring the system works reliably inside **Najran University – College of Computer Science and Information Systems**.

3.1.2 System Design

The system architecture is designed based on the gathered requirements. The main components include:

- **Hardware Design:**
    - Raspberry Pi 4 (main controller)
    - Gas sensor
    - Water sensor
    - Line-tracking IR sensors
    - Ultrasonic sensor (optional distance readings)
    - HD camera
    - Motor driver + tank chassis

- **Software Design:**
    - Python scripts for sensor reading, motor control, and notifications
    - Flask server to handle API requests and data logging
    - Flutter dashboard for admin/user interaction
    - MJPEG-Streamer for real-time video feed

- **Data Flow Design:**
- Sensors send readings to the Raspberry Pi
- Flask server processes events
- Alerts (WhatsApp/email) are triggered when hazards are detected
- Data is displayed on the Flutter dashboard

3.1.3 Implementation

In this phase, the actual coding and hardware assembly take place:

- Writing Python scripts to read sensor data, detect gas and water, control motors, and manage robot actions.
- Integrating the HD camera stream using MJPEG-Streamer for real-time monitoring.
- Building Flask HTTP APIs to handle alerts, notifications, snapshots, and communication with the Flutter dashboard.
- Connecting the Flutter application with the backend to display live video, notifications, and system status.

3.1.4 Testing

The system undergoes rigorous testing to ensure proper functionality:

- **Unit Testing:** Each sensor and module is tested individually.
- **Integration Testing:** Testing the interaction between different components.
- **Performance Testing:** Evaluating response time and accuracy of object detection.
- **User Testing:** Ensuring ease of use and effectiveness in detecting water and gas.

3.1.5 Deployment and Maintenance

Once testing is completed, the system is deployed within the college. Continuous monitoring and maintenance are carried out to:

- Ensure sensors function correctly over time.
- Fix any software bugs or hardware issues.
- Maintaining the hardware components (motors, sensor cables, Raspberry Pi)

3.2 System Workflow

The system follows a structured workflow:

1. Sensors continuously monitor the environment for gas leaks and water .
2. The Raspberry Pi processes the sensor data and manages the live video stream.
3. When a hazard is detected, the Flask server sends instant alerts through WhatsApp, email, and the dashboard.
4. Users monitor the robot and Receive alerts through the Flutter dashboard

3.3 Hardware and Software Integration

To achieve high accuracy, the integration of hardware and software is crucial:

1. The Raspberry Pi 4 controls all sensors and handles movement and monitoring.
2. The HD camera provides a live video stream displayed on the dashboard.
3. The Flask server processes sensor data and sends real-time alerts.
4. Wi-Fi enables communication between the robot and the dashboard.

3.4 Technologies Used

3.4.1 Front End

- **HTML, CSS, and JavaScript**
  Basic HTML, CSS, and JavaScript files are used inside the Flask server for simple interface testing and small utility pages. They support the backend by enabling basic layouts or static pages when needed, but the primary user interface remains the Flutter dashboard.

- **Flutter.**
  Flutter is used to build the main dashboard interface for both admin and regular users. It displays the live stream, notifications, and robot status through a clean and responsive UI.

3.4.2 Back End

- **Python.**
  Python is used on the Raspberry Pi to read sensor data, control the robot's movements, send alerts, and run backend scripts. It handles all the core logic of the SENT system.

- **Flask**.
  Flask is used as the backend web framework. It manages APIs, notifications, log storage, and communication between the Raspberry Pi and the Flutter dashboard.

- **MJPEG-Streamer**
  Used to provide real-time video streaming from the HD camera to the dashboard.

3.4.3 Documentation & Presentation.

- **Microsoft Word.**
  Microsoft Word is used to write and organize all project documentation. It allows creating structured reports, adding figures, tables, and formatting the content clearly. We relied on Word to prepare the full documentation for the SENT project.

- **Microsoft PowerPoint.**
  Microsoft PowerPoint is used to design and present the project during the final evaluation. It provides ready-made templates and custom design tools to create clear and visually appealing presentation slides.

- **Draw.io.**
  Draw.io is used to design system diagrams such as workflow diagrams, system architecture diagrams, and data flow diagrams. It helps visualize the structure of the project before implementation.

- **plantUML.**
  PlantUML is a tool that generates diagrams from text-based descriptions. It can be used to create sequence diagrams, use case diagrams, class diagrams, and activity diagrams in a simple and efficient way.

3.5 Summary

The project follows the Waterfall Model, where each phase—requirements, design, implementation, testing, and deployment—is completed in order to ensure a clear and organized development process. By combining the sensors, Flask backend, and Flutter dashboard, the SENT robot reliably detects gas leaks and water while providing real-time alerts inside the college environment.

Chapter 4: Proposed Solution

4.1 Feasibility Study

4.1.1 Why This Idea?

The Smart Environmental Navigation and Tracking Robot (SENT) is designed to provide an automated monitoring solution that improves safety inside educational environments. The idea emerged from the need for faster hazard detection in buildings, where traditional human-based monitoring can be slow, inconsistent, and prone to errors [12][15].The reasons for selecting this concept include:

- **Safety Enhancement:**
  - Water in hallways and classrooms can cause slip-and-fall accidents..
  - Automated detection improves response time and reduces risks.
- **Automation & Efficiency:**
  - Eliminates the need for constant human monitoring.
  - Reduces dependency on manual inspection processes.
- **Scalability & Adaptability:**
  - Can be modified to detect other environmental hazards like fire.
  - Can be adapted for use in industrial settings, hospitals, and public spaces.
- **Cost-Effectiveness:**
  - One-time implementation cost with **low operational expenses**.
  - Reduces long-term maintenance costs associated with hazard monitoring.

4.2 Requirements

Requirements are the needs or conditions that a system must meet or conform to. They define what the system is required to do and how it is expected to function from a system and user perspective.

4.2.1 Functional Requirements

- The system must detect water using a water sensor and notify the admin immediately.
- When a hazard is detected, the robot must send instant notifications through WhatsApp, email, and the dashboard.All detected hazards, including their location and timestamp, must be stored in a centralized database to allow future retrieval and analysis of historical incidents.
- The system must provide a live camera stream so users can monitor the robot's surroundings in real time.
- The dashboard must support two roles:
  - Admin: Receive/delete logs, request snapshots, full access
  - User: Receive notifications + live stream
- The robot must support two movement modes:
  - Manual control
  - Line-following mode using IR sensors
- All detected hazards must be logged (title, message, timestamp) and displayed on the dashboard.

4.2.2 Non-Functional Requirements

- The system must continue operating steadily and send alerts without interruption.
- Sensor readings and notifications should be processed quickly to ensure timely detection.
- The dashboard interface should be simple, clear, and easy for both admin and regular users.
- The Raspberry Pi scripts, Flask server, and Flutter interface should be easy to update and modify.
- The system should allow adding new sensors in the future, such as fire or smoke detectors.
- The robot must communicate reliably over Wi-Fi for live streaming and alerts.

4.3 Diagrams

Below are the descriptions for the system's **diagrams**, including **attributes and relationships**, to guide AI-generated diagram creation.

4.3.1 Use Case Diagram

A Use Case Diagram shows the **interactions between users (actors)** and the system. It helps visualize the system's **functionalities** and how different users interact with those functions as shown in figure 8.

- **Admin:**

  Login: The admin logs into the system to access administrative functionalities.
  Take Photo: The admin can capture a real-time photo from the robot's camera for documentation or inspection.
  Clean Notifications: The admin can delete and manage the stored notifications to keep the system organized.
  Receive Notifications: The admin can receive all alerts generated by the robot, including gas leaks, water detection, or system warnings.
  Live Stream: The admin can access the robot's live video feed for monitoring.

- **User**:

  Login: The User logs into the system to access the interface.
  Receive Notifications: The User can Receive system alerts sent by robots, such as gas or water detection.
  Live Stream: The User can watch the robot's live video feed for real-time monitoring.

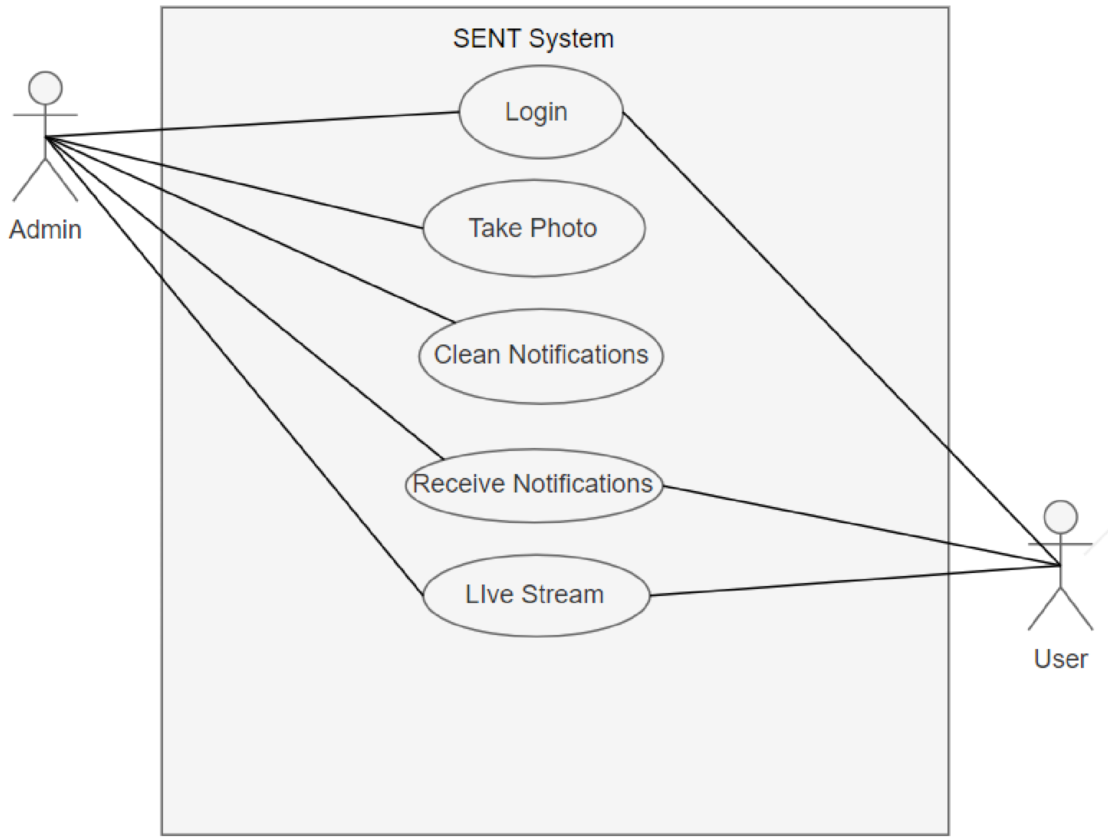

Figure 8 Use Case Diagram

4.3.2 Class Diagram

A Class Diagram is employed to illustrate of the organization of a **system** by showing its **classes, attributes, methods**, and the **relationships between objects** in the system as shown in figure 9.

- **User:**
  Attributes: User ID, username, password, role
  Methods: Login, receive notifications, receive live stream
- **Admin:**
  Attributes: Admin ID, email.
  Methods: Take photo, clean notifications, send alerts
- **Robot:**
  Attributes: Robot ID, gas level, water status, camera feed
  Methods: Move, detect hazards, capture photo, send alert
- **SENT System:**
  Attributes: Notifications log, connected users, robot instance
  Methods: Authenticate user, store notification, provide live stream, send email, send WhatsApp alert, logout

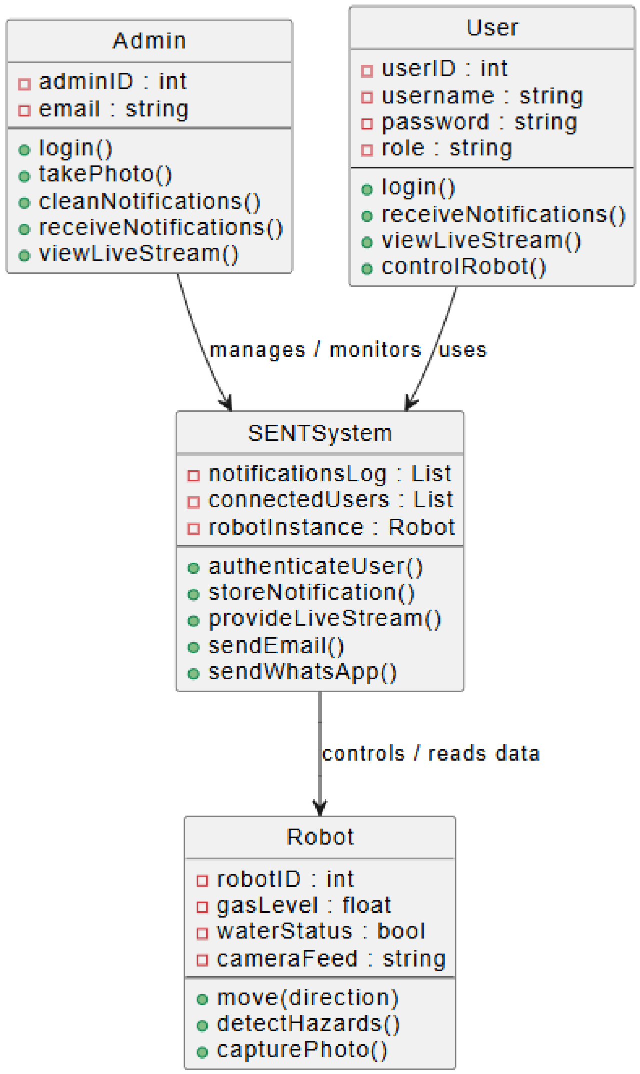

Figure 9 Class Diagram

4.3.3 User Sequence Diagram

The **User Sequence Diagram** illustrates the interactions within the **Smart Environmental Navigation and Tracking Robot (SENT)** system, showing the order of operations between system elements over time to achieve a specific function as shown in figure 10.

- **Log in**: The user submits login credentials through the dashboard. The backend authenticates the user and returns the login status.
- **Receive Notifications**: The user requests all stored notifications. The backend retrieves the notifications and sends them back to the dashboard.

- **Display Alerts:**: The Dashboard displays the received notifications to the user for monitoring system activity.

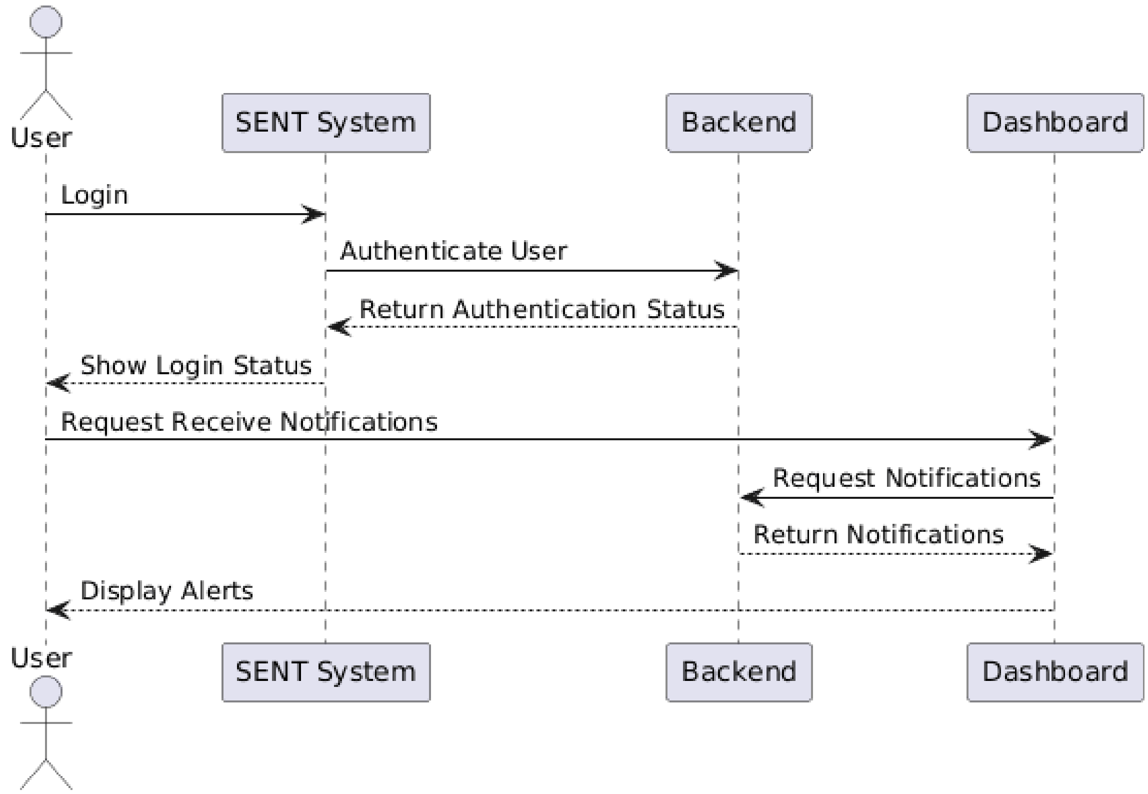

Figure 10 User Sequence Diagram

4.3.4 Admin Sequence Diagram

The Admin Sequence Diagram outlines the main interaction steps between the admin and the SENT system during operation. It shows how the system handles login, viewing notifications, capturing images, and sending email alerts, as shown in Figure 11.

- **Log In**: The admin submits login credentials. The backend authenticates the admin and returns the authentication status.
- **Receive Notifications:** The admin requests stored notifications. The backend retrieves and returns all hazard alerts (gas, water, system warnings).
- **Display Alerts:** The system displays the received notifications to the admin for monitoring.
- **Take Picture of Leak:** When the admin requests a snapshot, the backend triggers the robot to capture an image using its camera.
- **Send Image via Email:** The captured image is returned to the backend, which then forwards it to the admin's email for documentation.

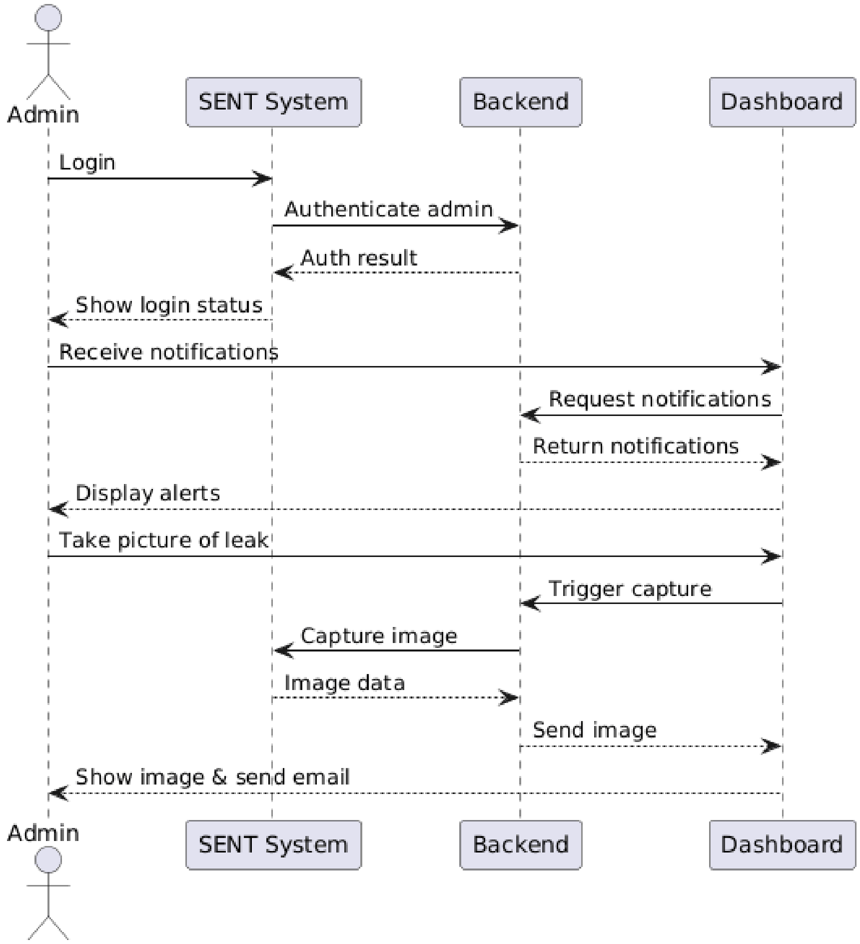

Figure 11 Admin Sequence Diagram

4.3.5 Admin & User Activity Diagram

The Admin and User Activity Diagram illustrate the flow of actions performed by both roles inside the SENT system. It shows how each user proceeds from login to performing their allowed operations, as shown in Figure 12.

- **Admin Workflow:**
    - **Login:** The admin enters login credentials. If authentication succeeds, access is granted; otherwise, an error message is shown.
    - **Receive Notifications**: The admin receives all stored hazard alerts such as gas or water detection.
    - **Clean Notifications:** The admin can clear the stored notifications from the system.
    - **Take Photo:** The admin requests the robot to capture an image, which is then sent via email if needed.
    - **Live Stream**: The admin can access the robot's live video feed for monitoring.

- **Log Out:** After completing tasks, the admin logs out

- **User Workflow:**
    - **Login:** The user enters login credentials. If correct, access is granted; otherwise, an error message is displayed.
    - **Receive Notifications:** The user receive system alerts stored in the dashboard.
    - **Live Stream:** The user accesses the live video stream for monitoring.
    - **Log Out:** Once finished, the user logs out of the system.

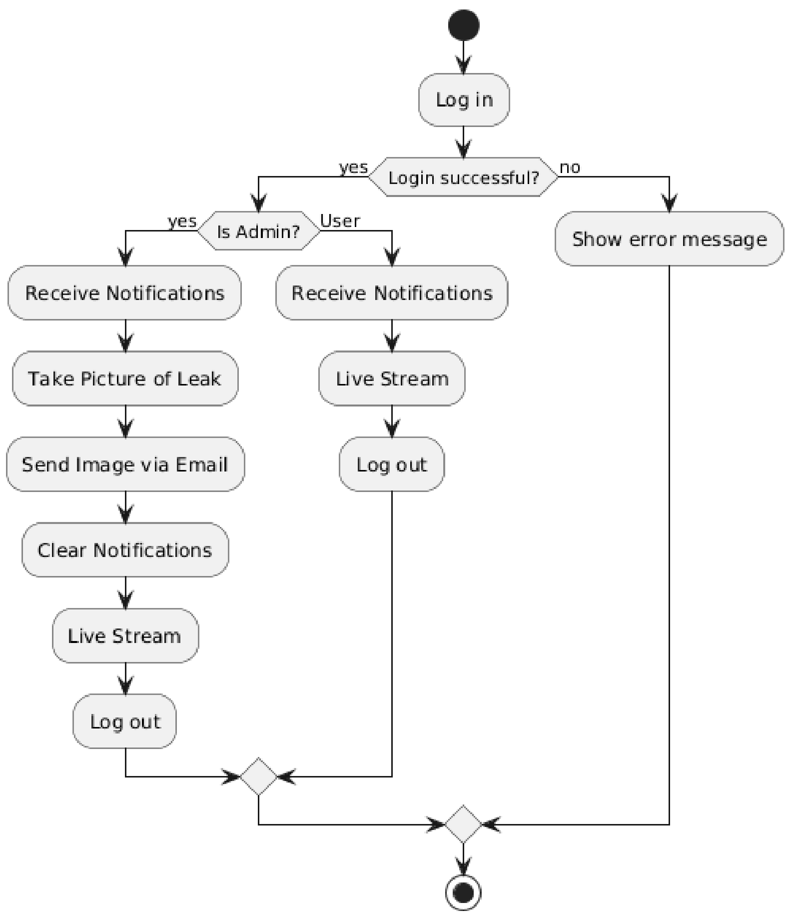

Figure 12 Admin & User Activity Diagram

Chapter 5: Implementation & Future Work.

5.1 Overview

The Smart Environmental Navigation and Tracking Robot (SENT) was developed to provide a new, automated solution for detecting indoor hazards using IoT technologies and smart sensor integration. The system combines real-time monitoring, camera inspections, and automated alerts to enhance safety at the College of Computer Science and Information System [1][16]. **5.**6 Conclusion

Safety inside educational buildings is a critical requirement, especially in areas where hazards such as water leaks or gas exposure may lead to accidents or interruptions in daily operations. The Smart Environmental Navigation and Tracking Robot (SENT) provides a practical solution by integrating real-time sensing, automated monitoring, and instant alerting to improve safety within the College of Computer Science and Information Systems [1].

The system combines a Raspberry Pi 4 with gas and water sensors, an ultrasonic module, and a live streaming camera to monitor the environment continuously. When a hazard is detected, SENT automatically sends notifications through email and WhatsApp to the administrator while displaying alerts on the dashboard. This automation reduces the need for manual inspection and ensures faster responses to potential risks [14].

While the robot performs effectively in indoor environments, some limitations remain. The system relies on stable lighting and flat surfaces, and the sensors may require recalibration over time to maintain accuracy. Additionally, the platform is not yet optimized for outdoor use or large-scale deployment [16].

In conclusion, the SENT project demonstrates how low-cost hardware, real-time monitoring, and automated alerts can work together to create a safer and more efficient environment. It establishes a strong foundation for future development in intelligent hazard detection and IoT-based safety systems [14][16].

5.2 Future Work

Future enhancements to the SENT system will focus on expanding its sensing, intelligence, and mobility capabilities. One important improvement is non-contact water detection, since the current system only detects water upon direct contact with the sensor (partial detection). Future versions can adopt capacitive or IR-based moisture sensing to identify water from a distance without requiring physical contact.

Additional planned enhancements include integrating fire-detection modules, advanced multi-gas classification, and thermal imaging to detect hidden heat sources. Cloud integration may enable remote monitoring, long-term event recording, and predictive analytics. Features such as audible alerts, multi-angle surveillance cameras, and robotic-arm waste collection could further enhance the robot’s practical applications. Support for outdoor operation, kinetic energy–based self-charging, and communication between multiple SENT robots would transform the system into a fully autonomous safety and surveillance platform.

5.3 References

[1] XiaoR GeekSmart WiFi Robot Kit, "XiaoR GeekSmart WiFi Robot Kit." [Online]. Available: https://www.xiaorgeek.com. Accessed: Feb. 3, 2025.

[2] Raspberry Pi Foundation, "Raspberry Pi 4 Official Documentation." [Online]. Available: https://www.raspberrypi.org/documentation/. Accessed: Feb. 3, 2025.

[3] MQTT, "MQTT Protocol Specification." [Online]. Available: https://mqtt.org/. Accessed: Feb. 3, 2025.

[4] OpenCV, "OpenCV Documentation." [Online]. Available: https://docs.opencv.org/. Accessed: Feb. 3, 2025.

[5] Flask, "Flask Web Framework Documentation." [Online]. Available: https://flask.palletsprojects.com/. Accessed: Feb. 3, 2025.

[6] FastAPI, "FastAPI Documentation." [Online]. Available: https://fastapi.tiangolo.com/. Accessed: Feb. 3, 2025.

[7] IEEE, "Security Automation in Smart Buildings," IEEE Journal of Security Systems, vol. 12, no. 3, pp. 45–58, 2022.

[8] J. Smith and A. Kumar, "Obstacle Detection Using Ultrasonic Sensors," International Journal of Robotics Research, vol. 40, no. 2, pp. 120–135, 2021.

[9] Robotics Vision Lab, "How Robots Use Cameras for 3D Object Detection and Tracking," RoboticsVisionLab.org, 2023. [Online]. Available: https://roboticsvisionlab.org/3d-object-detection.

[10] Y. Wang, "AI-Based Smart Surveillance Systems: Applications and Challenges," Journal of Artificial Intelligence Research, vol. 25, no. 4, pp. 201–215, 2023.

[11] T. Brown and M. Green, "A Comparative Study of Smart Monitoring Systems for Water Leak Detection," IEEE Transactions on Smart Systems, vol. 30, no. 6, pp. 78–89, 2023.

[12] UML Diagrams for Software Design, "Understanding Use Case, Sequence, and Activity Diagrams," IEEE Software Engineering Journal, vol. 18, no. 4, pp. 55–72, 2024.

[13] P. Roberts, "IoT-Enabled Robotics for Smart Environments," International Journal of AI & Robotics, vol. 33, no. 2, pp. 89–105, 2023.

[14] M. Zhang and R. Lee, "AI-Driven Safety and Surveillance in Educational Institutions," AI & Machine Learning Journal, vol. 21, no. 6, pp. 150–175, 2022.

[15] UML.org, "Unified Modeling Language (UML) Official Guide." [Online]. Available: https://www.uml.org/. Accessed: Mar. 25, 2025.

[16] M. Patel, "The Future of AI-Powered Robots in Hazard Detection," Robotics Today, vol. 10, no. 8, pp. 205–220, 2024.