BCSE497J - Project-I

AquaWatch IoT - Real-time automated boiling test solution

21BCT0014	**PRATIK ANAND**
21BCT0094	**ADVAITA SHARMA**
21BCI0228	**SPARSH TEOTIA**

Under the Supervision of

DR. ARUNKUMAR T

Professor Higher Academic Grade

School of Computer Science and Engineering (SCOPE)

B.Tech.

in

Computer Science and Engineering

School of Computer Science and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

November 2024

ABSTRACT

Abstract: Real-Time Temperature Monitoring for Long-Duration Tests

This project aims to develop a web-based mobile application capable of providing real-time temperature monitoring for long-duration tests (specifically, 100-hour boiling tests) conducted at remote locations. The application will enable users to access and visualize temperature data in a convenient and accessible manner, enhancing the efficiency and reliability of the testing process.

The current method of monitoring such tests often involves manual observation and recording of temperature data, which can be time-consuming, labor-intensive, and prone to human error. By automating this process, the proposed application will significantly reduce the workload of researchers and technicians, allowing them to focus on other critical aspects of the experiment.

The application will utilize a temperature sensor to continuously measure the temperature of the boiling solution. This data will then be transmitted wirelessly to a central server, where it will be processed and stored. Users can access the application through a web browser on their mobile devices or computers to view real-time temperature readings, historical data, and graphical representations of the temperature trends.

In addition to providing real-time monitoring, the application will also offer features such as data logging, alarm notifications, and remote control capabilities. Data logging will allow users to store and analyze temperature data for future reference, while alarm notifications can be configured to alert users of any abnormal temperature fluctuations. Remote control capabilities will enable authorized users to start, stop, or adjust the heating process from a distance, enhancing flexibility and convenience.

The proposed application is expected to have a significant impact on the field of long-duration testing by improving efficiency, accuracy, and accessibility. By providing real-time temperature monitoring and data analysis tools, researchers and technicians will be better equipped to conduct and analyze their experiments, leading to valuable insights and advancements in their respective fields.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Background

The production of high-quality insulators is a critical component of various electrical systems, ensuring the safe and efficient transmission of power. One essential aspect of insulator quality assurance is the rigorous testing of their performance under extreme conditions, such as prolonged exposure to high temperatures and corrosive environments. Traditional methods of testing often involve manual monitoring and data recording, which can be time-consuming, labor-intensive, and prone to human error.

To address these challenges and enhance the efficiency and reliability of insulator testing, this project aims to develop a web-based mobile application capable of real-time monitoring and display of temperature data during long-duration boiling tests. By automating the data collection and analysis process, the application will provide valuable insights into the performance of insulators under extreme conditions, facilitating more informed decision-making and quality control measures.

The proposed application will be designed to monitor a specific type of test, the 100-hour boiling test, which is commonly used to assess the durability and resistance of insulators to high temperatures and corrosive environments. This test involves immersing test specimens in a 5% NaCl solution and heating it to boiling for an extended period. After the test, the specimens are analyzed for leakage current under a high voltage, providing a measure of their insulation integrity.

By automating the monitoring of this test, the application will eliminate the need for continuous human presence, allowing researchers and technicians to focus on other critical tasks. Additionally, the application will provide real-time data visualization, enabling users to track the temperature of the boiling solution and identify any anomalies or trends that may indicate potential issues with the test specimens.

1.2 Motivation

The development of a web-based mobile application for real-time temperature monitoring during the 100-hour boiling test is driven by the desire to address the limitations and inefficiencies of traditional manual methods. By automating the monitoring process and providing remote access to temperature data, the application aims to enhance the efficiency, accuracy, and accessibility of the testing process.

Inspiration for this project stems from the recognition that manual monitoring can be time-consuming, labor-intensive, and prone to human error. The application seeks to alleviate these challenges by leveraging modern technology to provide a more reliable and efficient solution. Additionally, the increasing demand for high-quality and reliable insulators in various industries has highlighted the need for improved testing methods.

The project's importance lies in its potential to contribute to advancements in insulator technology and ensure the safe and efficient operation of electrical systems. By providing accurate and accessible temperature data, the application can support research and development efforts, improve quality control, and ultimately enhance the performance and reliability of insulators.

1.3 Scope of the Project

The primary objective of this project is to develop a web-based mobile application capable of real-time monitoring and display of temperature data during the 100-hour boiling test. The application will be designed to ensure accurate and reliable temperature measurement and data transmission, providing a user-friendly interface for accessing and visualizing temperature data.

Key features of the application will include the integration of a temperature sensor for continuous measurement, wireless transmission of data to a central server, a mobile app for real-time access, graphical representation of temperature trends, data logging, customizable alarms, and optional remote control capabilities. The application will focus on temperature monitoring and will be specifically designed for the 100-hour boiling test, with potential future enhancements to include additional test parameters and compatibility with other test types.

To ensure data security, appropriate measures will be implemented to protect sensitive information and prevent unauthorized access. The application's functionality will depend on the availability and performance of the temperature sensor and communication hardware.

2. PROJECT DESCRIPTION AND GOALS

2.1 Literature Review

Existing research and theories in several related fields:

1. Temperature Measurement and Sensors:

- Sensor Technology: Research on various temperature sensors, such as thermocouples, thermistors, and infrared sensors, has contributed to the development of accurate and reliable measurement techniques.
- Sensor Calibration: Studies on sensor calibration methods ensure the accuracy of temperature readings and minimize measurement errors.
- Sensor Placement: Research on optimal sensor placement within the boiling solution helps to capture representative temperature data.

2. Wireless Communication:

- Communication Protocols: Existing research on wireless communication protocols (e.g., Bluetooth, Wi-Fi, cellular networks) provides the foundation for transmitting temperature data to a central server.
- Network Reliability: Studies on network reliability and error correction techniques ensure the integrity of data transmission.

3. Data Processing and Analysis:

- Data Storage: Research on efficient data storage techniques enables the long-term preservation of temperature data.
- Data Visualization: Studies on data visualization methods (e.g., line charts, bar charts) facilitate the interpretation and analysis of temperature trends.
- Data Analysis Techniques: Research on statistical analysis techniques (e.g., mean, standard deviation, correlation analysis) can be applied to extract meaningful insights from the temperature data.

4. Mobile Application Development:

- User Interface Design: Research on user interface design principles guides the development of intuitive and user-friendly mobile applications.
- Mobile App Development Frameworks: Existing frameworks (e.g., React Native, Flutter) provide tools and libraries for efficient mobile app development.

5. Real-Time Systems:

- Real-Time Operating Systems: Research on real-time operating systems ensures timely data processing and response to events.
- Synchronization and Timing: Studies on synchronization and timing mechanisms guarantee accurate and consistent data collection.

6. Industrial Automation:

- Automation Technologies: Research on industrial automation technologies (e.g., programmable logic controllers, SCADA systems) provides insights into the integration of automated systems with laboratory equipment.

2.2 Research Gap

While existing research provides a solid foundation for developing a web-based mobile application for real-time temperature monitoring, several research gaps remain to be addressed:

1. Long-Term Sensor Reliability:

- Degradation: Further research is needed to evaluate the long-term reliability and potential degradation of temperature sensors under the harsh conditions of the 100-hour boiling test.
- Calibration Drift: Investigating calibration drift over extended periods is essential to ensure accurate temperature measurements.

2. Data Security and Privacy:

- Data Encryption: Research on robust encryption techniques is required to protect sensitive temperature data from unauthorized access and breaches.
- Privacy Compliance: Ensuring compliance with data privacy regulations (e.g., GDPR, CCPA) is crucial, especially when dealing with sensitive scientific data.

3. Energy Efficiency:

- Power Consumption: Investigating methods to reduce the power consumption of the temperature sensor and communication devices is important for prolonging battery life and minimizing environmental impact.
- Energy-Efficient Algorithms: Exploring energy-efficient algorithms for data processing and transmission can further optimize power usage.

4. Integration with Laboratory Equipment:

- Interoperability: Research on protocols and standards for seamless integration of the monitoring system with existing laboratory equipment (e.g., heating units, data loggers) is necessary.
- Automation: Investigating methods to automate the entire testing process, including heating control and data analysis, can improve efficiency and reduce human error.

5. User Experience and Interface Design:

- Intuitive Interfaces: Research on user-centered design principles and usability testing can help create intuitive and user-friendly mobile applications.
- Customization: Investigating methods to allow users to customize the application's interface and features based on their specific needs.

2.3 Objectives

SMART Objectives for the project :

Goal 1: Accurate and Real-Time Temperature Monitoring

- Specific: Implement a system to accurately measure and display real-time temperature data during the 100-hour boiling test.
- Measurable: Achieve a temperature measurement accuracy of ±1°C and update the display in real-time (within 5 seconds).
- Achievable: Use proven temperature sensors and communication technologies to ensure reliable and accurate measurements.
- Relevant: Accurate temperature data is essential for the successful execution and analysis of the boiling test.

Goal 2: Remote Access and Visualization

- Specific: Develop a web-based mobile application that allows users to access and visualize temperature data remotely.
- Measurable: Ensure the application is accessible on various devices (iOS, Android, web) and provides clear graphical representations of temperature trends.
- Achievable: Utilize existing web development frameworks and mobile app development tools.
- Relevant: Remote access will enhance flexibility and convenience for researchers and technicians.

Goal 3: Improved Data Reliability and Efficiency

- Specific: Reduce human error and improve the efficiency of the testing process through automation.
- Measurable: Achieve a reduction in manual data recording errors by 90% and a 20% increase in overall testing efficiency.
- Achievable: Implement automated data collection and storage mechanisms.
- Relevant: Improved data reliability and efficiency will contribute to more accurate test results and faster research outcomes.

Goal 4: Enhanced Data Security and Privacy

- Specific: Implement robust security measures to protect sensitive temperature data.
- Measurable: Ensure compliance with relevant data privacy regulations (e.g., GDPR, CCPA) and implement encryption mechanisms to protect data in transit and at rest.
- Achievable: Utilize industry-standard security practices and tools.
- Relevant: Protecting data privacy is essential to maintain trust and comply with legal requirements.

2.4 Problem Statement

The manual monitoring of the 100-hour boiling test requires continuous human presence, which is time-consuming and prone to errors. A web-based mobile application that displays real-time temperature data can automate the monitoring process and improve efficiency.

2.5 Project Plan

Phase 1: Project Planning and Initialization (July 21 - August 4)

<u>Week 1 (July 21-27):</u>

- Define project scope and objectives
- Identify team members and assign roles
- Conduct initial meetings with the advisor/supervisor
- Develop a detailed project plan and timeline

This phase will include engagement with BHEL employees, virtual or physical, to gain a deeper understanding of the project and successfully form a comprehensive document with requirements and scope.

Week 2 (July 28 - August 4):

● Gather necessary resources and tools (temperature sensors, microcontroller, etc.)

● Conduct literature review and research on temperature monitoring and data logging

● Refine project objectives based on initial research

● Develop and finalise project proposal

● Submit project proposal for approval

This phase will include researching components, technologies and information essential for the project's needs and a proposed design of the prototype.

Phase 2: Research and Requirements Gathering (August 5 - August 18)

Week 3 (August 5-11):

● Conduct in-depth research on temperature monitoring and data logging

● Gather and document detailed project requirements (hardware, software, data storage,

etc.)

This phase will commence the discussion of preferred tech stack for handling data based on our understanding of the project's requirement.

Week 4 (August 12-18):

● Perform feasibility study (if necessary)

● Develop initial project plan and timeline

● Review project plan with advisor/supervisor

● Make necessary adjustments to the project plan

This phase assumes a finished structure for the project, and may even take place earlier than intended depending on the pace of the project.

Phase 3: Design Phase (August 19 - September 8)

Week 5 (August 19-25):

● Develop initial design concepts and prototypes (hardware setup, data logging

mechanism)

● Review designs with team and advisor/supervisor

This phase begins construction of the designs for the working model, starting from the hardware end of the project.

Week 6 (August 26 - September 1):

- Gather feedback on designs
- Refine and finalise designs

This phase depends on the constructive feedback from the supervisor.

Week 7 (September 2-8):

- Prepare detailed design documents (hardware circuit diagrams, software architecture)
- Plan for the development phase

Phase 4: Development Phase (September 9 - October 13)

Week 8 (September 9-15):

- Begin development of core components/features (temperature sensor integration, data

logging software)

- Conduct regular code reviews and testing

Week 9 (September 16-22):

- Continue development and integration of components (mobile app development,

graphical data display)

- Ensure functionality and address any issuesWeek 10 (September 23-29):
- Further development and testing
- Integration of all components
- Initial deployment on testing environment

Week 11 (September 30 - October 6):

- Full system testing (real-time data monitoring, remote access functionality)
- Fix any bugs and issues identified during testing

Week 12 (October 7-13):

- Prepare the system for the 100-hour test
- Conduct final pre-test checks and validation

Phase 5: Testing and Finalization (October 14 - November 6)

Week 13 (October 14-20):

- Conduct the 100-hour test (start the test, monitor the setup)
- Ensure continuous data logging and remote monitoring

This phase may not actually involve a 100-hour test due to limited resources and results may be judged on simulated sessions.

Week 14 (October 21-27):

- Complete the 100-hour test
- Take out specimens and measure leakage current
- Analyse test data and validate results

Week 15 (October 28 - November 3):

- Finalise all project documentation (project report, user manual)
- Prepare for project presentation and defence

Week 16 (November 4-6):

- Submit project for review
- Conduct project presentation and defence

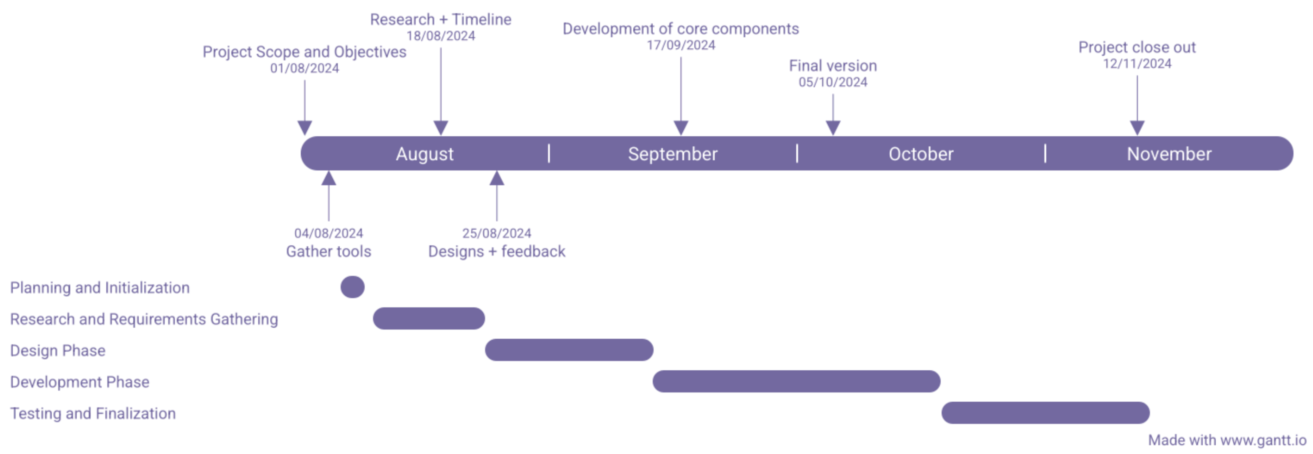


Fig. 1. Gantt chart

3. TECHNICAL SPECIFICATION

3.1 Requirements

3.1.1 Functional

These describe the specific behaviors or functions the system must exhibit.

1. **Temperature Monitoring**
   - The system must continuously measure the boiling water temperature using thermocouples.
   - It must record temperature data at regular intervals (e.g., every minute or hour) for the entire duration of the test.
2. **Data Logging**
   - The Raspberry Pi must process and log temperature data from the thermocouples.
   - NodeMCU should handle data transmission from the Raspberry Pi to the Thingspeak cloud.
3. **Data Transmission**
   - The system should transmit recorded data to the Thingspeak cloud in real-time via Wi-Fi, using NodeMCU.
4. **Real-time Graphical Representation**
   - Thinkspeak cloud should display the recorded temperature data in graphical form, updated in real-time.
   - The graph must show the current temperature, historical data, and elapsed time during the test.
5. **Mobile App Access**
   - The system should provide remote access to the data via a mobile app, allowing users to monitor the temperature and elapsed time from anywhere.
6. **Data Storage and Retrieval**
   - After the test completion, users must be able to access and verify historical temperature data from Thingspeak.
7. **Alerts and Notifications (Optional for prototype)**
   - The system may trigger alerts or notifications via the mobile app if the temperature exceeds or drops below predefined thresholds.

3.1.2 Non-Functional

These describe how the system should perform under various conditions and constraints.

1. **Reliability**

   The system must reliably record and transmit temperature data without frequent disconnections or data loss over a period of several days.

2. **Accuracy**

   Temperature readings must have high precision, with the thermocouples capable of detecting small changes in the boiling water's temperature.

3. **Scalability**

   The system should be scalable to support future upgrades, such as monitoring multiple boiling tests simultaneously.

4. **Performance**

   The system must process temperature readings and transmit them to the cloud with minimal delay (real-time or near-real-time).

   The mobile app should provide a smooth user experience, loading data and graphs quickly.

5. **Security**

   Data transmitted to the Thingspeak cloud should be secure, and user access to the data via the mobile app should require authentication.

6. **Usability**

   The mobile app and Thinkspeak interface must be easy to navigate, allowing users to access real-time and historical data effortlessly.

7. **Maintainability**

   The system must be easy to maintain, allowing for quick updates or repairs if any hardware or software issues arise.

8. **Energy Efficiency**

   The system should consume minimal power, especially during data transmission, to ensure the Raspberry Pi and NodeMCU can operate continuously for extended periods.

9. **Scalability**

   While this is a prototype, the system design should allow scaling for more extensive data collection and monitoring in the future, possibly with multiple sensors.

10. **Modularity**

    The design should be modular, allowing components (such as thermocouples, Raspberry Pi, NodeMCU) to be replaced or upgraded independently.

3.2 Feasibility Study

3.2.1 Technical Feasibility

- **Technology Availability:** The project relies on readily available technologies such as thermocouples for temperature sensing, Raspberry Pi for data processing, and cloud services like ThingSpeak for data visualization. All components are well-documented, widely used, and accessible.
- **Technical Expertise:** A basic understanding of embedded systems, IoT communication, and cloud integration is required. The project team should have experience with programming the Raspberry Pi (Python, Linux), interfacing sensors, and using cloud platforms like ThingSpeak for data monitoring.
- **Infrastructure:** The project requires standard hardware, including thermocouples, Raspberry Pi, and networking equipment to connect the system to the internet. Cloud infrastructure for data storage and visualization is handled by ThingSpeak, which reduces the need for in-house server maintenance.
- **Integration:** The system must ensure smooth integration between the temperature sensors, Raspberry Pi, and the ThingSpeak cloud platform. Data should be collected in real-time, transmitted without interruption, and displayed on the cloud dashboard. Mobile app access for remote monitoring must also be tested.

3.2.2 Economic Feasibility

- **Cost-Benefit Analysis:** Initial costs include purchasing Raspberry Pi boards, thermocouples, and any necessary networking components. The ThingSpeak cloud service offers free tiers, but paid options may be necessary for extended data logging. The automation of the boiling test eliminates the need for continuous human monitoring, reducing labor costs and increasing precision, which justifies the initial investment.
- **Budget:** A detailed budget should account for the cost of Raspberry Pi, thermocouples, Wi-Fi modules (if necessary), power supplies, and cloud subscription fees (if applicable). Personnel costs for developing, deploying, and maintaining the system should also be included.
- **Return on Investment (ROI):** The ROI comes from improved efficiency in test monitoring, reduced labor costs, and more accurate data collection. Long-term, the automation allows for scaling, improving the overall productivity of the testing process and potentially reducing errors.
- **Funding:** Depending on the scope, the project could seek internal funding or external grants, particularly if it is part of a larger initiative focused on automating industrial or scientific processes.

3.2.3 Social Feasibility

- **User Acceptance:** The system must be user-friendly, with clear data presentation on the ThingSpeak cloud and mobile app. The simplicity of monitoring and the elimination of the need for manual supervision should increase user satisfaction and acceptance.
- **Training and Support:** Adequate training materials should be provided to familiarize users with setting up and managing the system. Basic support for troubleshooting and using the cloud dashboard should also be provided.
- **Ethical Considerations:** The project involves data collection on non-personal metrics (boiling water temperature), so ethical concerns like data privacy are minimal. However, the system should ensure reliable and secure data transmission to avoid potential tampering or data loss.
- **Impact on Workforce:** The introduction of an automated system could reduce the need for continuous manual supervision, potentially shifting job roles toward more analytical or supervisory tasks. Proper communication and transition planning will ensure that workers adapt to these changes without disruption.

3.3 System Specification

3.3.1 Hardware Specification

- **Processor**:

  Raspberry Pi 4 Model B with a Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz.

- **Memory (RAM)**:

  2GB LPDDR4 SDRAM (based on model choice).

- **Storage**:

  16GB or 32GB microSD card for the Raspberry Pi OS, application scripts, and temporary data caching. (**optional** if device is completely reliant on cloud services)

- **Graphics Processing Unit (GPU)**:

  Broadcom VideoCore VI @ 500MHz (in Raspberry Pi 4)

- **Monitor**:

  Any **HDMI-compatible monitor (optional)** for real-time monitoring when needed for local troubleshooting or setup since data will be viewed remotely via a mobile app or web interface connected to the cloud.

3.3.2 Software Specification

- **Operating System**:
  Raspberry Pi OS (formerly Raspbian), a Linux-based OS optimized for Raspberry Pi.
- **Programming Languages:**
  - Python: Used for sensor data collection and communication with the cloud (ThingSpeak).
  - JavaScript/HTML/CSS: For developing the web interface.
- **Development Environment:**
  - Visual Studio Code: IDEs for writing Python scripts to handle sensor input, data processing, and cloud communication.
  - ThingSpeak API: For data integration and visualization on the cloud.
- **Libraries and Frameworks:**
  - PySerial: For interfacing with the thermocouple sensors.
  - Requests/HTTP: For sending data to the ThingSpeak cloud using HTTP requests.
  - Flask or Django: If developing a custom web interface to display the sensor data.
- **Database**:
  ThingSpeak Cloud (or similar cloud database): This will store and visualize real-time sensor data, negating the need for a local database.
- **Security Tools:**
  - SSL/TLS: For securing communication between the Raspberry Pi and the cloud.
  - API Key Authentication: Used in ThingSpeak to ensure secure data transmission from the Raspberry Pi to the cloud.

4. DESIGN APPROACH AND DETAILS

4.1 System Architecture

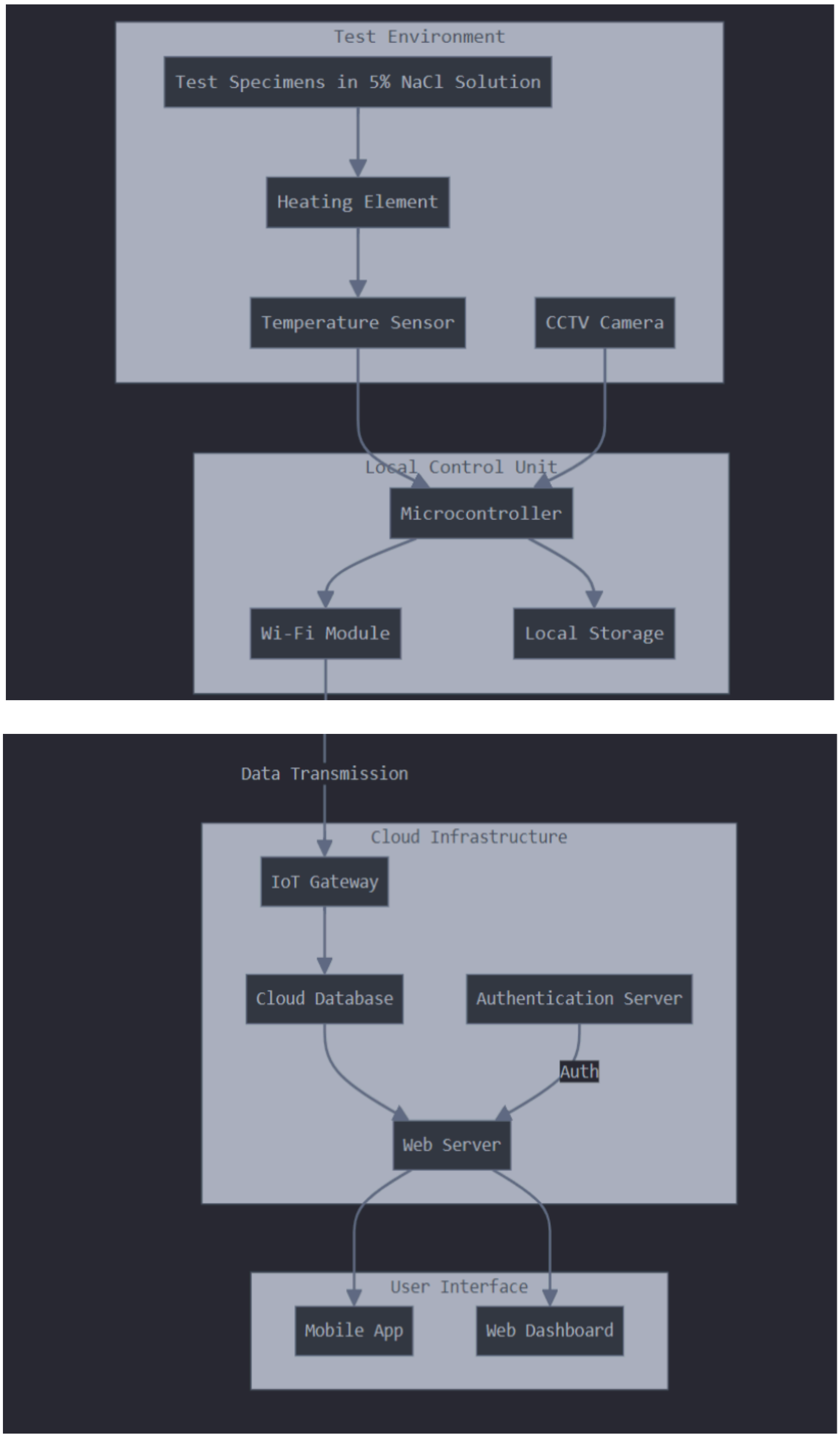


Fig. 2. System Architecture

1. Test Environment:

- Test Specimens in 5% NaCl Solution: This is where the actual test takes place. The specimens are submerged in the solution for 100 hours.
- Heating Element: This could be an electric heating coil controlled by the microcontroller. It maintains the solution at boiling temperature.
- Temperature Sensor: A waterproof, high-temperature sensor (like a PT100 or K-type thermocouple) that can withstand the boiling salt solution.
- CCTV Camera: A waterproof, heat-resistant camera positioned to capture the entire setup.

2. Local Control Unit:

- Microcontroller: This could be an Arduino or Raspberry Pi. It performs several crucial functions: a) Reads temperature data from the sensor at regular intervals (e.g., every minute). b) Controls the heating element to maintain boiling temperature. c) Captures images or video feed from the CCTV camera. d) Stores data locally and sends it to the cloud.
- Wi-Fi Module: Enables wireless communication. It could be built into the microcontroller (like in Raspberry Pi) or a separate module (like ESP8266 for Arduino).
- Local Storage: This could be an SD card or built-in storage of the microcontroller. It stores data temporarily if the internet connection is lost.

3. Cloud Infrastructure:

- IoT Gateway: This is a software component that receives data from multiple devices (useful if you scale up to multiple test setups). It handles device authentication and data preprocessing.
- Cloud Database: A database like MongoDB or PostgreSQL to store time-series data efficiently. It will contain temperature readings, timestamps, test metadata, and references to CCTV footage.
- Web Server: Hosts the backend application (could be built with Node.js, Python Flask, or similar frameworks). It serves APIs for the mobile app and web dashboard.
- Authentication Server: Manages user accounts, roles, and access tokens. This ensures that only authorized personnel can view data or control the system.

4. User Interface:

- Mobile App: A cross-platform app (perhaps built with React Native or Flutter) that allows users to:
  - a) View real-time temperature data.
  - b) See historical data in graphical format.
  - c) Receive alerts if temperature goes out of range.
  - d) View live or recorded CCTV footage.
- Web Dashboard: A more comprehensive interface (could be built with React or Vue.js) that includes all mobile app features plus: a) Detailed data analysis tools. b) Test setup and control (for future enhancements). c) User management for administrators.

4.2 Design

4.2.1 Data Flow Diagram

1. The temperature sensor continuously measures the solution temperature.

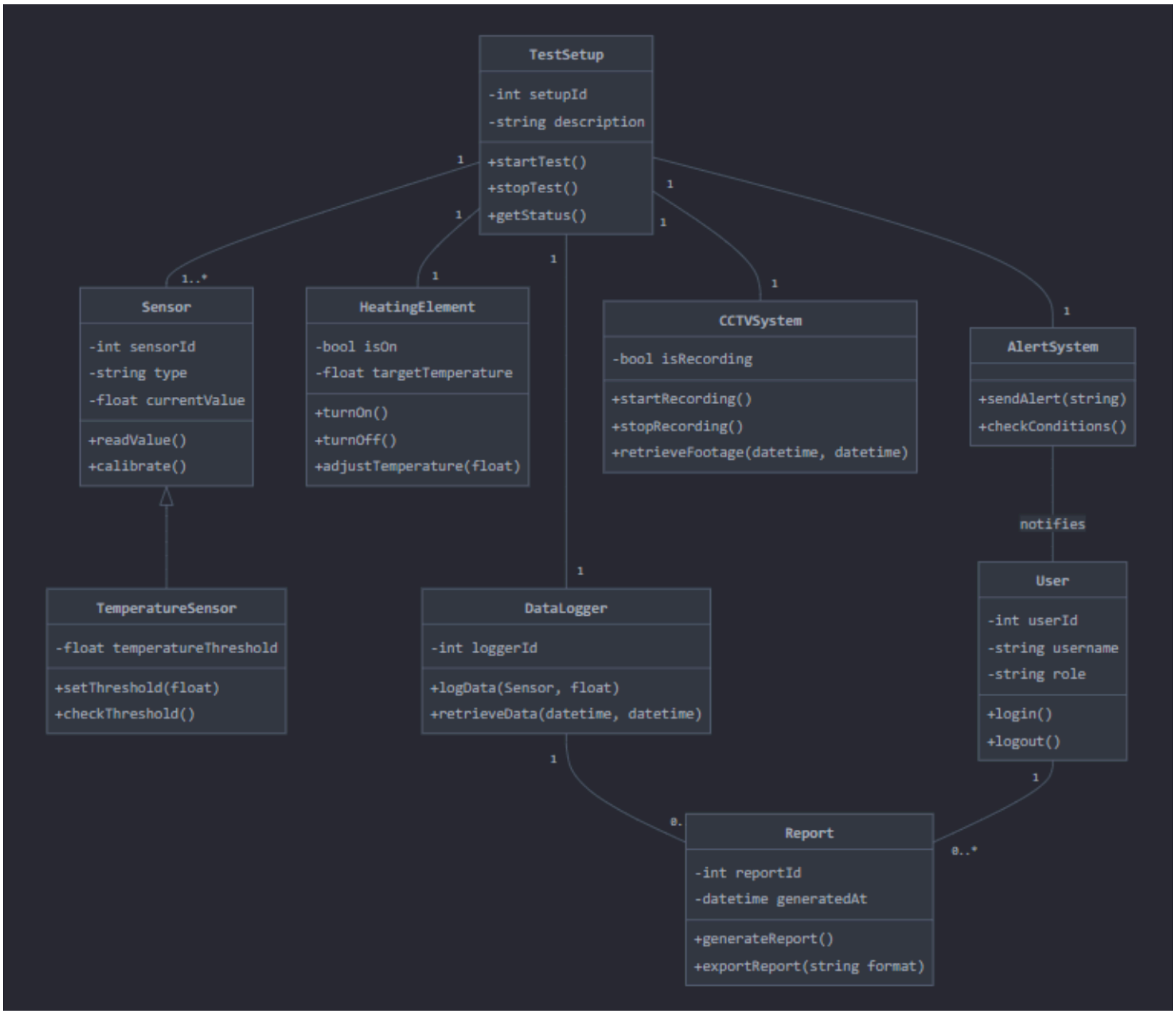
2. The microcontroller reads this data, stores it locally, and sends it to the cloud via the Wi-Fi module.

3. The IoT Gateway receives this data and forwards it to the cloud database.

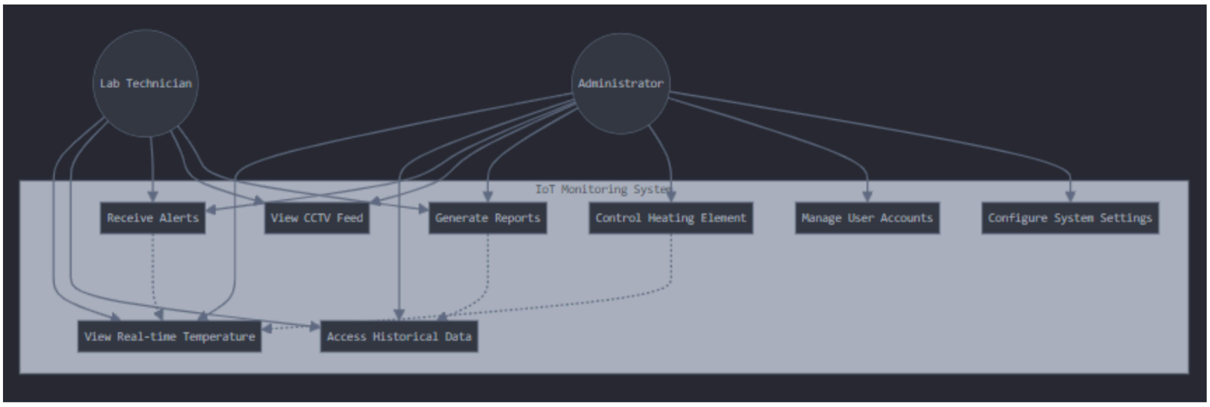
4. The web server retrieves data from the database and serves it to the mobile app and web dashboard.

5. Users access this data through the mobile app or web dashboard, authenticated by the authentication server.

4.2.2 Class Diagram



4.2.3 Use Case Diagram



Future Enhancements:

1. Remote Control: Add relays controlled by the microcontroller to allow remote starting/stopping of the heating element.

2. Multiple Parameters: Include sensors for other parameters like pH or conductivity.

3. Machine Learning: Implement predictive maintenance or anomaly detection algorithms to alert users of potential issues before they occur.

5. REFERENCES

Weblinks:

1. https://www.techtarget.com/iotagenda/resources/Internet-of-Things-IoT-Security](https://www.techtarget.com/iotagenda/resources/Internet-of-Things-IoT-Security)
2. https://www.ibm.com/cloud/mobile](https://www.ibm.com/cloud/mobile)

Journals: *<IEEE Format>*

1. Remote Monitoring and Control of Industrial Processes Using IoT and Cloud Computing"* by A. Kumar and S. Kumar in International Journal of Advanced Research in Computer Science and Engineering (IJARCSE). This paper discusses the use of IoT and cloud computing for remote monitoring and control, which could be applied to a boiling test. https://iopscience.iop.org/article/10.1088/1757-899X/1055/1/012034/pdf

2. Real-Time Monitoring of Industrial Processes Using Mobile Applications"* by B. Lee and C. Park in Sensors and Actuators A: Physical. This paper focuses on the use of mobile applications for real-time monitoring of industrial processes, which could be relevant to the boiling test.https://www.mdpi.com/1424-8220/20/19/5687