

College of Engineering and Technology

B.Sc. Computer Engineering

***Graduation Project II***

***Final Report***

**Augmented Reality (AR) in Smart Buildings**



***Academic Year:*** *2024 – 2025*

***Semester:*** *Fall 2024*

***Course Title and Code:*** *CE477*

|  |  |  |
| --- | --- | --- |
| Authors’ Names | IDs | Majors |
| Hamad Alatel | 59070 |  |
| Saoud Aldhuwaihi | 59620 |  |
| Ali Almataani | 52439 |  |
| Hamad Alajmi | 48898 |  |
| Hussain Sadeq | 33921 |  |
| Supervisor: Dr. | | |

*Abstract*:

The main goal of this project is to bridge the gap between raw data collecting and user engagement by creating a visually stimulating and interactive experience. By combining IoT with AR, we show how technology can make data more accessible, increase user engagement, and support informed decision-making in a variety of sectors, including energy management, education, and environmental awareness. The idea is simple but effective, providing a scalable solution for small-scale monitoring systems while laying the groundwork for future advances in smart building technologies. Our approach prioritizes ease of deployment, cost-effectiveness, and adaptability, demonstrating that advanced technologies like AR and IoT can be made accessible and useful for everyday use.

In this project, we created a unique system that combines Internet of Things (IoT) and Augmented Reality (AR) technologies to revolutionize how environmental data is monitored and visualized in smart buildings. The device captures real-time environmental data by pairing an ESP32 microcontroller with a DHT11 temperature and humidity sensor. The data is then securely transmitted to ThingSpeak, a cloud-based platform, where it is stored and visualized using dynamic, interactive graphs. To make this data more interesting and user-friendly, an AR application was created with Unity and Vuforia. This software detects particular building photos and uses the AR interface to overlay real-time environmental data directly on the user's screen.

Table of Contents

[I. Introduction 7](#_Toc185163185)

[1. Problem Statement 7](#_Toc185163186)

[2. Motivation 8](#_Toc185163187)

[3. Main Objectives 8](#_Toc185163188)

[4. Engineering Principles 9](#_Toc185163189)

[II. Design Development 10](#_Toc185163190)

[1. Comparison with Existing Solutions and Literature Review 10](#_Toc185163191)

[2. Unique Advantages of the Proposed System 11](#_Toc185163192)

[3. Proposed Design 11](#_Toc185163193)

[a) Problem Analysis and Engineering Challenges 11](#_Toc185163194)

[b) Solution Requirements 11](#_Toc185163195)

[c) Design Criteria 12](#_Toc185163196)

[d) Constraints 13](#_Toc185163197)

[e) Solution Overview 13](#_Toc185163198)

[4. Detailed High-Level Specifications 13](#_Toc185163199)

[a) IoT Sensor Block 13](#_Toc185163200)

[b) Prototype Inputs and Outputs 14](#_Toc185163201)

[c) Cloud Platform Block. 16](#_Toc185163202)

[d) AR Technology Block 16](#_Toc185163203)

[e) Interactive Application Block. 16](#_Toc185163204)

[f) Integration and Communication Block 17](#_Toc185163205)

[g) Vuforia Target Image and Feature Extraction 17](#_Toc185163206)

[5. Detailed Low-Level Specifications 19](#_Toc185163207)

[a) IoT Sensor 19](#_Toc185163208)

[b) Cloud Platform 20](#_Toc185163209)

[c) AR Technology 20](#_Toc185163210)

[d) Interactive Application 21](#_Toc185163211)

[e) Link Between Unity, ThingSpeak Server, and Sensor 22](#_Toc185163212)

[f) Integration and Communication 22](#_Toc185163213)

[g) Vuforia Target Image and Feature Extraction 23](#_Toc185163214)

[6. Performance Specifications: 23](#_Toc185163215)

[7. Flowchart 24](#_Toc185163216)

[III. Project Realization and Performance Optimization 25](#_Toc185163217)

[1. Planned implementation and experiments 25](#_Toc185163218)

[a) Hardware Implementation: 25](#_Toc185163219)

[b) Software Implementation: 25](#_Toc185163220)

[c) Simulation and Testing: 25](#_Toc185163221)

[d) Planned Experiments: 25](#_Toc185163222)

[2. Design Analysis and Feedback 26](#_Toc185163223)

[a) Experiment 1: Sensor Accuracy Testing 26](#_Toc185163224)

[b) Experiment 2: Data Transmission Reliability 26](#_Toc185163225)

[c) Experiment 3: AR Tracking Accuracy 26](#_Toc185163226)

[d) Experiment 4: Unity Scene Optimization 26](#_Toc185163227)

[3. Design Optimization and Improvements [PI-6.c] 27](#_Toc185163228)

[a) Sensor Integration: 27](#_Toc185163229)

[b) Data Transmission: 27](#_Toc185163230)

[c) AR Experience: 27](#_Toc185163231)

[d) System Scalability: 27](#_Toc185163232)

[IV. General Discussion 28](#_Toc185163233)

[1. Final Cost Analysis and Discussion 28](#_Toc185163234)

[a) Hardware 28](#_Toc185163235)

[b) Software 28](#_Toc185163236)

[2. Comparison with Existing Solutions 29](#_Toc185163237)

[a) Raspberry Pi-Based Systems: 29](#_Toc185163238)

[b) Blynk or Adafruit IO Platforms: 29](#_Toc185163239)

[c) Commercial IoT Solutions: 29](#_Toc185163240)

[3. Cost Effectiveness Discussion 29](#_Toc185163241)

[4. Commercializing the Project and Relevance to Region 29](#_Toc185163242)

[a) Social Impact 30](#_Toc185163243)

[b) Cultural Relevance 30](#_Toc185163244)

[c) Environmental Benefits: 30](#_Toc185163245)

[d) Political and Economic Context 30](#_Toc185163246)

[e) Commercialization Potential 30](#_Toc185163247)

[f) Local Relevance and Industrial Applications 31](#_Toc185163248)

[5. Related Engineering Standards 31](#_Toc185163249)

[6. Engineering Considerations: 32](#_Toc185163250)

[a) Relevant Ethical aspects and Codes of Ethics 32](#_Toc185163251)

[b) Impact of proposed solution 33](#_Toc185163252)

[c) Alignment of proposed solution with the Engineering codes of ethics and taking into consideration the impacts of the design 33](#_Toc185163253)

[V. Project Management 35](#_Toc185163254)

[1. The Schedule of GP2 Tasks 35](#_Toc185163255)

[2. Encountered Problems and Proposed Solutions 36](#_Toc185163256)

[3. Students’ Responsibilities and Contribution 37](#_Toc185163257)

[VI. Conclusion and Future works 38](#_Toc185163258)

[VII. References 39](#_Toc185163259)

[VIII. Appendices 40](#_Toc185163260)

**Table of Figures:**

[Figure 1: : Augmented Reality in Educational Institutions 7](#_Toc185163301)

[Figure 2: ESP32 Microcontroller with DHT11 Temperature Sensor. 14](#_Toc185163302)

[Figure 3: ThingSpeak IoT PlatformFigure 4: ESP32 Microcontroller with DHT11 Temperature Sensor. 14](#_Toc185163303)

[Figure 5: ThingSpeak IoT Platform 16](#_Toc185163304)

[Figure 6: AR System (Unity & Vuforia).Figure 7: ThingSpeak IoT Platform 16](#_Toc185163305)

[Figure 8: AR System (Unity & Vuforia). 16](#_Toc185163306)

[Figure 9: C# Code to Display The Text in Unity. 17](#_Toc185163307)

[Figure 10: Vuforia database image recognition 17](#_Toc185163308)

[Figure 11: ESP32 Microcontroller. 19](#_Toc185163309)

[Figure 12: DHT11 Temperature Sensor 19](#_Toc185163310)

[Figure 13: ESP32 Microcontroller with DHT11 Temperature Sensor. 19](#_Toc185163311)

[Figure 14: ThinkSpeak displays data received through API key 20](#_Toc185163312)

[Figure 15: Unity Editor Interface Window. 20](#_Toc185163313)

[Figure 16: Vuforia Dataset Page 21](#_Toc185163314)

[Figure 17: C# used with Unity Editor. 21](#_Toc185163315)

[Figure 18: Project’s Gantt Chart.Figure 19: C# used with Unity Editor. 21](#_Toc185163316)

[Figure 20: Feature extraction. 23](#_Toc185163317)

[Figure 21: System Flowchart for Real-Time Data Acquisition and AR Visualization. 24](#_Toc185163318)

[Figure 22: Project’s Gantt Chart. 35](#_Toc185163319)

[Figure 23: ESP32 Microcontroller with DHT11 Temperature Sensor.Figure 24: Project’s Gantt Chart. 35](#_Toc185163320)

**Table of Tables:**

[Table 1: Hardware Costs Table 28](#_Toc185108721)

[Table 2: Software Costs Table 28](#_Toc185108722)

[Table 3: Schedule Time Table 35](#_Toc185108723)

[Table 4: Student's Responsibilities table. 37](#_Toc185108724)

# Introduction

This project intends to transform smart building systems by integrating Augmented Reality (AR) technology with Internet of Things (IoT) sensors, utilizing the Unity and Vuforia platforms. By integrating immersive AR experiences, the system improves environmental awareness and encourages efficient behavior among students and educators. Traditional methods employed in smart buildings usually fail to create long-term behavior change due to their static nature, lack of engagement, and limited real-time feedback. This gap highlights the importance of a transformational approach that blends interactive AR technologies with IoT devices to give actionable insights and real-time communication.



Figure 1: : Augmented Reality in Educational Institutions

The suggested method addresses this gap through the integration of Unity and Vuforia to produce user-focused AR applications that work smoothly with IoT devices. These apps provide dynamic, real-time environmental data visualizations, making complex information more accessible and entertaining. This paradigm change improves the user experience while simultaneously promoting sustainability and energy efficiency in smart buildings. This project demonstrates how modern technology may be used to develop meaningful, practical solutions that excite and educate, laying the groundwork for future advancements in smart building management.

## Problem Statement

The traditional approach to track and inform users about smart building technologies has major drawbacks. These methods rely mainly on static displays, isolated monitoring systems, and non-interactive learning materials, which do not engage users or develop long-term sustainable behavior. In addition, the lack of real-time feedback and attractive visualization tools underutilizes crucial environmental data, limiting its ability to provide actionable insights and meaningful decision-making.

As buildings become more related and data-driven, the challenge is to bridge the gap between raw data collection and intuitive, user-friendly engagement. Without a paradigm change toward immersive and dynamic technologies, smart buildings could come up failing in terms of energy efficiency, sustainability, and user engagement. This necessitates a creative solution that not only captures and transfers data seamlessly, but also converts it into a compelling experience that motivates users to take educated decisions. The lack of such a solution restricts the full potential of smart building ecosystems, leaving crucial chances for learning, efficiency, and environmental impact untouched.

## Motivation

The combination of IoT and AR technologies has the potential to bridge this gap by providing dynamic, real-time presentation of environmental data in a compelling and intuitive format. IoT devices like the ESP32 and DHT11 offer seamless data gathering and transmission, while augmented reality technologies like Unity and Vuforia enable immersive data interaction. By merging these technologies, we hope to improve data accessibility and user engagement, making complicated environmental information easier to comprehend and act on. This research is motivated primarily by the possible influence on energy conservation, instructional tools, and smart building management.

## Main Objectives

The main objectives of the project are:

* To create and build a system that collects real-time temperature and humidity data via IoT devices.
* To securely send and visualize this data, use a cloud-based platform such as ThingSpeak.
* Create an augmented reality application that leverages Vuforia and Unity to overlay environmental data onto specific building photos, resulting in an interactive and interesting user experience.
* To integrate engineering, physics, and mathematics ideas into system design to ensure efficiency, accuracy, and scalability.

## Engineering Principles

This project uses key engineering concepts like:

* Data Acquisition: Accurate environmental data collection using IoT devices such as the ESP32 and DHT11.
* Signal processing is the process of converting sensor readings into digital data that may be transmitted to the cloud.
* AR integration involves using computer vision techniques in Vuforia for picture identification and Unity for real-time rendering.
* System Scalability: Creating a modular architecture that can be expanded to accommodate future functionalities.

# Design Development

The suggested system combines IoT sensors, augmented reality technology, and interactive applications to provide an intelligent building management solution. The key components are:

* Hardware: ESP32 microcontroller combined with DHT11 sensors to collect environmental data such as temperature and humidity.
* Software: Unity and Vuforia platforms for creating AR interfaces and managing visualizations.
* Data Management: The ThingSpeak API stores and processes real-time sensor data for integration into the AR application.

## Comparison with Existing Solutions and Literature Review

To highlight the benefits of the proposed method and its inspiration from earlier research and solutions examined in GP1, a thorough comparison with current solutions is essential. In order to address issues with smart building management and sustainable energy education, the suggested project combines IoT, AR, and Unity.

GP1 examined a number of current options, including implementations based on the Raspberry Pi, Adafruit IO, and Blynk. The scalability, functionality, and appropriateness for real-time applications of these systems were assessed. The components used for GP2 were influenced by the knowledge acquired.

* IoT Platforms: Adafruit IO and Blynk offered excellent illustrations of IoT integration platforms. They were not as well adapted, nevertheless, for user-focused instructional applications and real-time AR integration. Because of its compatibility with Unity and Vuforia, ease of usage of the API, and secure data management, ThingSpeak was selected for GP2, making it the superior option for real-time visualization.
* Hardware Selections: ESP32 was compared with Raspberry Pi and Arduino Uno in GP1. Despite its computational capacity, the Raspberry Pi was expensive and resource-intensive. Due to its absence of Wi-Fi, the Arduino Uno was not appropriate for Internet of Things applications. Because of its low cost, integrated Wi-Fi, and energy economy, the ESP32 turned out to be the ideal choice.
* Augmented Reality Systems: A number of AR platforms, such as Google ARCore and Microsoft HoloLens, were examined. Even while these systems had sophisticated features, they were not as accessible or affordable for educational usage as Vuforia and Unity, which strike a compromise between usability, scalability, and price.

## Unique Advantages of the Proposed System

The proposed solution improves upon existing solutions by combining the strengths of IoT and AR in a single framework. It achieves:

* Real-time data acquisition and visualization.
* Immersive AR experiences for better user engagement.
* Cost-effectiveness through the use of ESP32 and ThingSpeak.
* A scalable and modular design for future expansions.

The system leverages learnings from GP1 by addressing the limitations of alternative solutions and building a unique, user-focused system.

## Proposed Design

The Proposed Design is a culmination of innovative solutions aimed at addressing complex engineering challenges associated with integrating IoT, AR technologies, and sustainable practices in smart buildings. The design showcases the systematic efforts of the project team to develop a reliable, user-friendly, and efficient system that meets the requirements of modern energy management and educational applications.

### Problem Analysis and Engineering Challenges

The proposed design overcomes major engineering issues identified during the analytical process. Among these obstacles was the seamless integration of AR and IoT technology to offer precise, real-time environmental data, which is critical for developing compelling and useful feedback loops. The team also confronted the challenge of user engagement, which necessitated the creation of an intuitive interface capable of efficiently conveying information to a wide range of users, including students and instructors. Given resource management limits, it became critical to balance the prices and scalability of hardware and software solutions. These problems were critical in determining the design requirements, and they guided the team's approach throughout development.

### Solution Requirements

The proposed system was developed to meet the following requirements:

* Real-time Monitoring: Collect environmental data using IoT sensors (ESP32 and DHT11) and make it available via AR applications.
* Scalability: Allow for future extensions, such as adding sensors or integrating with other AR/IoT systems.
* User-Focused Design: Create an engaging interface that is simple to use, interactive, and visually informative.
* Cost Efficiency: Ensure that the hardware and software components stay within budget without sacrificing quality.
* Data security: Use secure API keys and cloud-based systems to protect data during transmission and storage.

### Design Criteria

The system was designed with sustainability, user engagement, and efficient performance as core goals. Accuracy was achieved through the careful calibration of the DHT11 sensor, while responsiveness was ensured by leveraging Unity and Vuforia's real-time augmented reality (AR) target processing capabilities. The integration of IoT devices with the ThingSpeak platform met the criteria for reliable data transfer and storage.

The design also addressed key constraints such as cost-effectiveness, ease of use, and resource availability. The ESP32 microcontroller was chosen for its affordability and suitability for IoT applications, making it an ideal choice for this project. The modular design supports future scalability, allowing for easy integration of additional components or features as needed.

The system captures real-time temperature and humidity data using the DHT11 sensor and ESP32, storing and visualizing this data through the Unity platform and Vuforia AR integration. Data is uploaded to ThingSpeak to enable dynamic interaction and analysis. The user-friendly interface minimizes the need for technical expertise, making the system accessible to a wide range of users.

It is also highly cost-effective, particularly for deployment in educational institutions, and prioritizes data privacy and security through the use of secure API keys and encrypted communication. Additionally, the immersive AR interface provided by Vuforia and Unity enhances user engagement, creating an interactive learning environment. Designed to integrate seamlessly with existing systems, the solution reduces deployment complexity while maintaining a focus on performance and scalability.

### Constraints

The project's design had to work within certain limits, which altered the team's decision-making process. One key barrier was the requirement to use low-cost technology while maintaining data quality and AR capability. This resulted in the choice of ESP32 and DHT11 as the fundamental IoT components. Furthermore, the software needed to work effectively with IoT platforms like ThingSpeak, which put restrictions on data formats and API usage. Another problem was assuring scalability while minimizing power usage, especially given the possibility for future growth. Finally, the design had to be user-friendly in order to support educators and students without having sophisticated technological skills.

### Solution Overview

The final planned design includes:

* IoT Sensors: ESP32 microcontroller and DHT11 sensor for temperature and humidity measurement.
* AR technology includes Unity and Vuforia for image identification and real-time data visualization.
* Cloud Platform is ThingSpeak for safe data storage and processing.
* Interactive Applications: C# scripts that display real-time sensor data in an engaging augmented reality interface.

This complete strategy creates a mix between creativity and pragmatism, tackling hard technical challenges while being user-centric. The modular architecture enables future scaling and system adaptation. The team's dedication to rigorous problem-solving and intuitive design ensures that the project effectively accomplishes its goals, making it functional and impactful.

## Detailed High-Level Specifications

To ensure clarity and functionality, the project design is broken down into interconnected blocks, each representing a vital component of the system. Here is a full breakdown of each block and its specific role:

### IoT Sensor Block

This block includes the ESP32 microcontroller and the DHT11 sensor, which together form the basis for real-time environmental data collecting. The ESP32 powers the system, reads sensor data, and wirelessly transmits it. The DHT11 sensor monitors temperature and humidity with great accuracy and sends the data to the ESP32 for processing. The IoT Sensor Block provides constant monitoring of environmental variables, making it an essential component of the system's performance.

A screenshot of a computer

Description automatically generated

Figure 2: ESP32 Microcontroller with DHT11 Temperature Sensor.

Figure 3: ThingSpeak IoT PlatformFigure 4: ESP32 Microcontroller with DHT11 Temperature Sensor.

### Prototype Inputs and Outputs

The prototype is intended to offer a real-time environmental monitoring and visualization solution by combining cloud platforms, IoT sensors, and augmented reality technologies. The system's inputs and outputs are well-defined, allowing hardware and software components to interact with one another seamlessly.

*Prototype Inputs*

The inputs of the system originate from the IoT sensors and user interactions. These are the primary data sources driving the system functionality:

*Environmental Data:*

* The DHT11 sensor collects real-time temperature and humidity readings from the environment.
* These measurements are digitally transmitted to the ESP32 microcontroller for further processing.

*Predefined Image Targets:*

* Specific target images (e.g., building images) are created and registered in the Vuforia database.
* When an ARCamera detects these images, it triggers the overlay of AR content.

*API Configuration:*

* ThingSpeak API keys are predefined as inputs in the code, allowing secure transmission and retrieval of data between the ESP32 and the cloud platform.

*Prototype Outputs*

The outputs represent the system’s ability to deliver actionable information and an engaging user experience:

*Cloud Data Visualization:*

* Real-time temperature and humidity data is uploaded to ThingSpeak.
* The platform processes and visualizes the data in graphs and charts for user analysis.

*AR Visualization:*

* Unity and Vuforia use the detected image target to display real-time sensor data, such as temperature and humidity, as AR text overlays.
* This creates an immersive experience where users can view dynamic data in their physical environment.

*User Feedback:*

* Through the AR interface, users receive immediate feedback on environmental conditions, enabling quick understanding and decision-making.

*Data Flow Description*

The following steps detail how inputs and outputs are managed within the system:

*Sensor Input:*

* The DHT11 sensor detects temperature and humidity in its surroundings and sends the data as a digital signal to the ESP32 microcontroller.

Data Transmission:

* The ESP32 processes the sensor readings and transmits them to the ThingSpeak cloud platform using HTTP POST requests via a Wi-Fi connection.

*Data Storage and Processing:*

* ThingSpeak receives the data, stores it in a dedicated channel, and generates real-time visualizations.

*AR Display:*

* Unity retrieves data from ThingSpeak using HTTP GET requests and API keys.
* The ARCamera recognizes the predefined target image using Vuforia datasets and displays the retrieved data as an AR overlay.

### Cloud Platform Block.

The Cloud Platform Block uses ThingSpeak to store, process, and show acquired data. The ESP32 communicates sensor values to ThingSpeak using secure HTTP requests that are

authenticated with API keys. ThingSpeak divides the data into channels and generates real-time graphical visuals for easier analysis. This block is critical to data management and security, allowing for seamless connection with other system components.

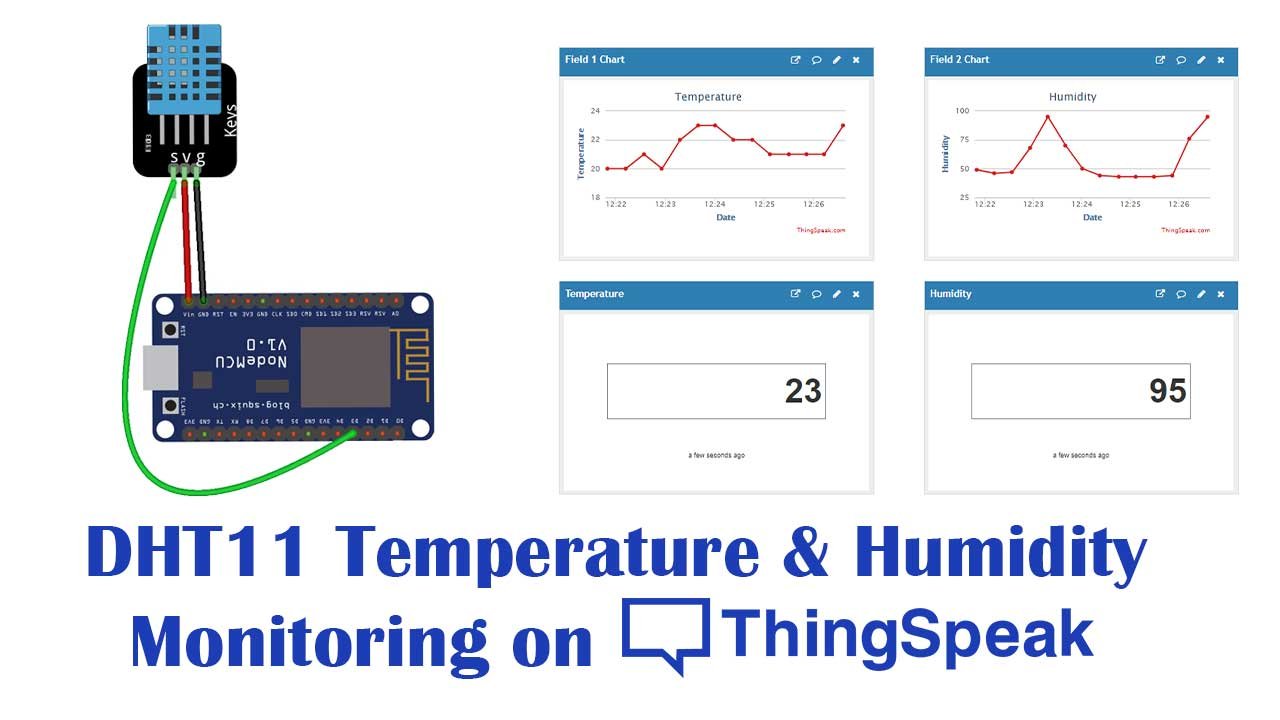


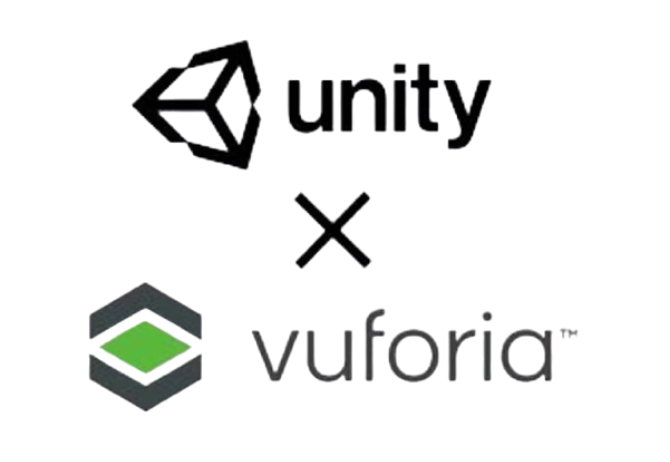
Figure 5: ThingSpeak IoT Platform

Figure 6: AR System (Unity & Vuforia).Figure 7: ThingSpeak IoT Platform

### AR Technology Block

This block is powered by Unity and Vuforia, which manage the AR functionality. Vuforia does picture identification using preloaded datasets, while Unity handles the 3D rendering and user interface. The AR Technology Block superimposes real-time data from ThingSpeak onto particular visual targets, giving consumers an interesting and immersive experience. This block guarantees that the AR application generates correct and responsive visuals.

Figure 8: AR System (Unity & Vuforia).



### Interactive Application Block.

The Interactive Application Block's core is built around custom C# scripts written in Unity. These scripts handle the interactions between the AR interface and the backend data. For example, they use ThingSpeak to receive sensor information and dynamically update the AR display. This block also manages user interactions, ensuring that the system remains intuitive and usable.

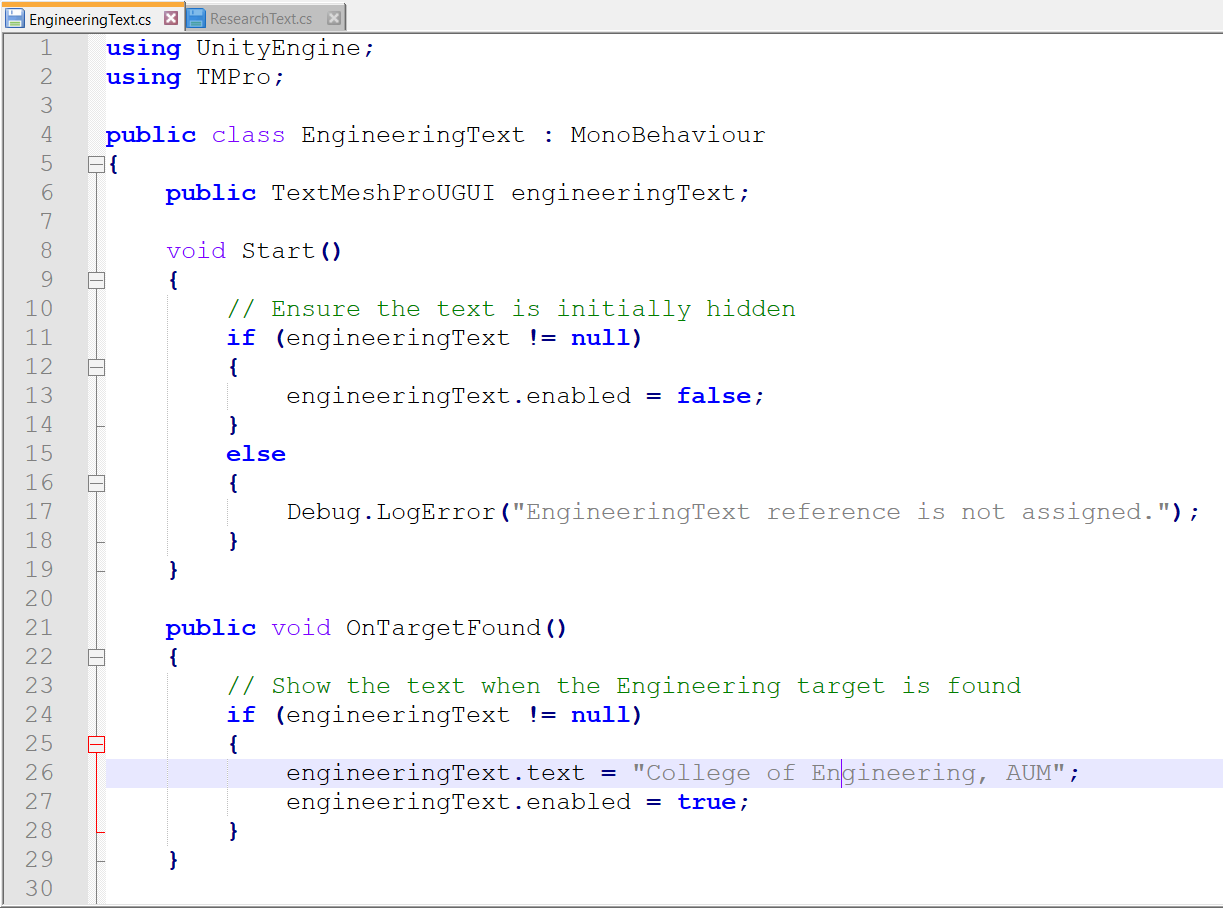
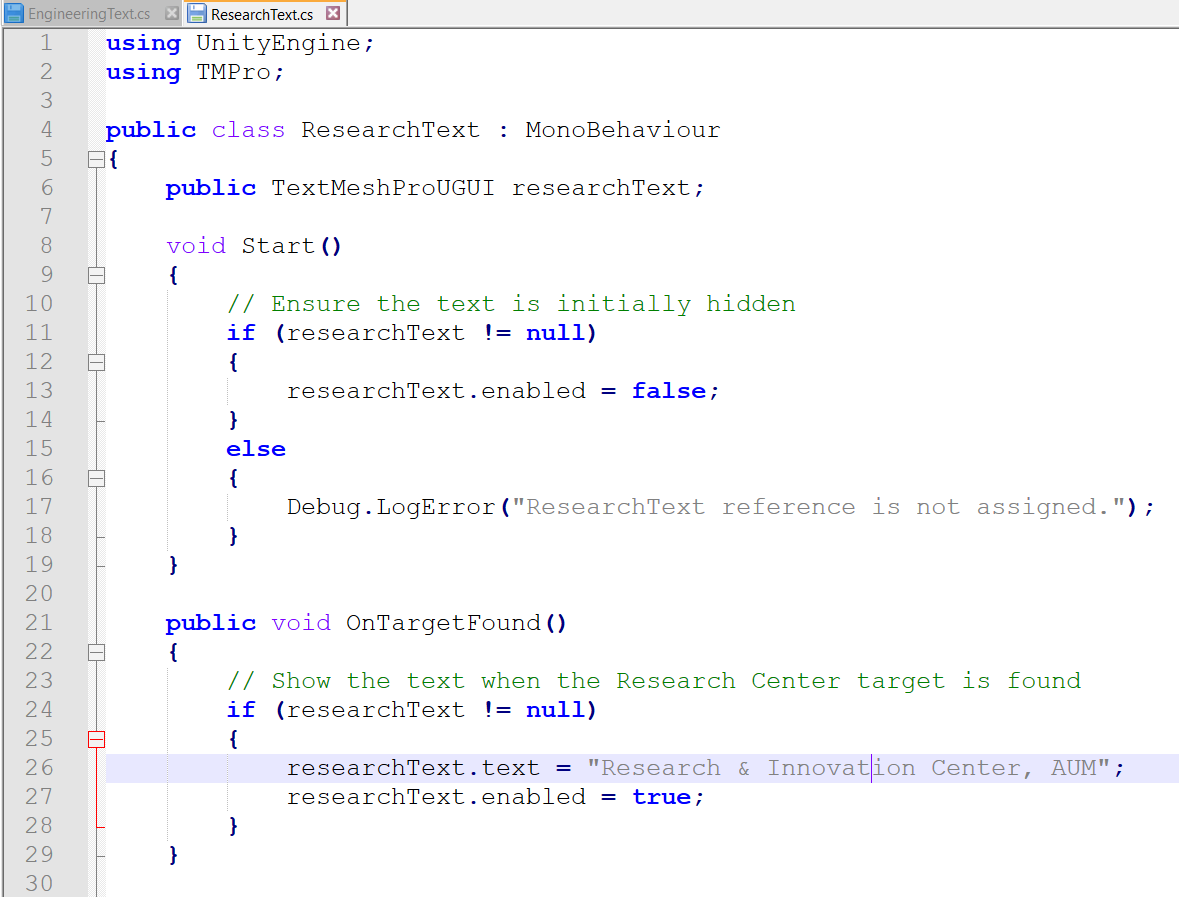


Figure 9: C# Code to Display The Text in Unity.

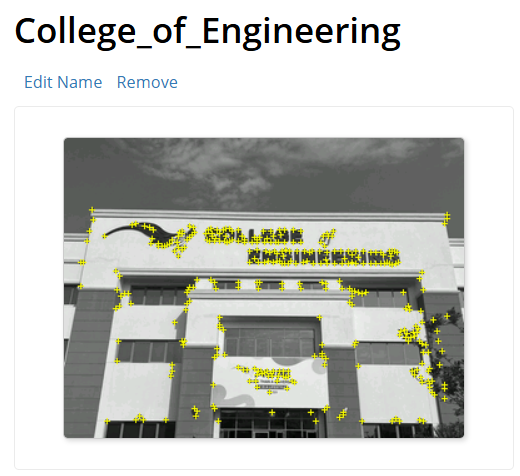
### Integration and Communication Block

The Integration and Communication Block guarantees that the system's components function together seamlessly. It facilitates communication among the IoT Sensor Block, Cloud Platform Block, and AR Technology Block. This comprises Wi-Fi data transmission, API request coordination, and synchronization of AR content changes with real-time sensor data. This block guarantees that the system operates seamlessly and reliably.

### Vuforia Target Image and Feature Extraction

The project utilizes Vuforia, a leading AR platform integrated into Unity, for object tracking and image recognition. The platform uses feature extraction to recognize and track objects in real-time, analyzing target images and identifying unique patterns, shapes, and textures. These reference points are stored in a Vuforia database, which is used during runtime to compare the live camera feed with stored features. Target images are carefully selected and optimized for high recognition accuracy.

Figure 10: Vuforia database image recognition



Each image is uploaded to the Vuforia Target Manager, which generates a dataset of feature points. This dataset is then imported into the Unity Editor, where the ARCamera in Unity overlays the virtual content. The feature extraction process ensures the AR system can reliably detect and track target images in different lighting conditions, angles, and distances. The integration of Vuforia with Unity and IoT data from ThingSpeak enhances the project's ability to provide interactive and informative AR content, ensuring its accuracy and reliability.

## Detailed Low-Level Specifications

This section delves further into the low-level specifications for each design component, concentrating on their unique features, settings, and performance requirements.

### IoT Sensor

ESP32 Microcontroller:

The device features a 32-bit dual-core processor with a clock speed of up to 240 MHz, 520 KB SRAM, and 4 MB Flash storage. It offers built-in Wi-Fi and Bluetooth for seamless communication with ThingSpeak and the AR application and operates at 3.3V for low-power applications.

Figure 11: ESP32 Microcontroller.

DHT11 Temperature / Humidity Sensor:

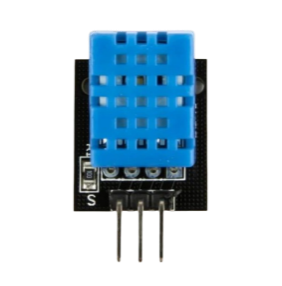
The DHT11 sensor measures temperature (from 0˚C to 50˚C) with a ±2°C and humidity accuracy and a ±5% accuracy, outputs a digital signal compatible with ESP32 GPIO pins, and performs reliably under standard environmental conditions.

Figure 12: DHT11 Temperature Sensor

A screenshot of a computer

Description automatically generatedA key aspect of the system is the smooth interaction between Unity, the ThingSpeak cloud server, and the sensor (DHT11 coupled to ESP32). This section describes the connections between these elements, the data that moves between them, and how they work together to accomplish the project's goals.

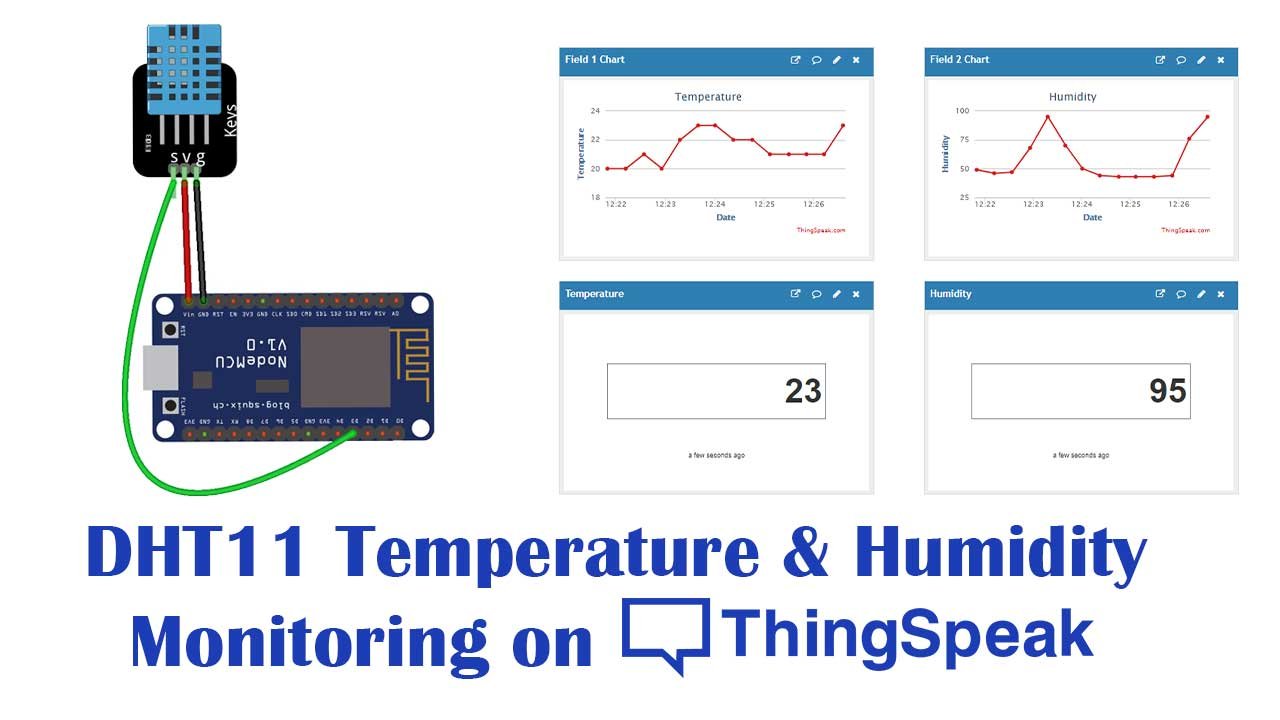
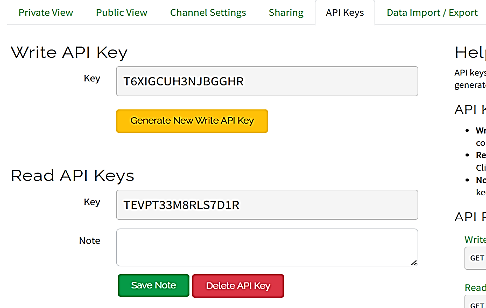
Figure 13: ESP32 Microcontroller with DHT11 Temperature Sensor.

### Cloud Platform

ThingSpeak:

The data transmission protocol is HTTP and MQTT, with secure uploads and real-time retrieval via API keys. It can process data updates in 15 seconds, supports up to 8 fields per channel, and has a scalable free tier that can handle 3 million messages per year.

Figure 14: ThinkSpeak displays data received through API key



### AR Technology

Unity:

Unity features a real-time 3D rendering engine, efficient scene management, and cross-platform compatibility for iOS, Android, and desktop applications, allowing for efficient handling of multiple objects and animations.

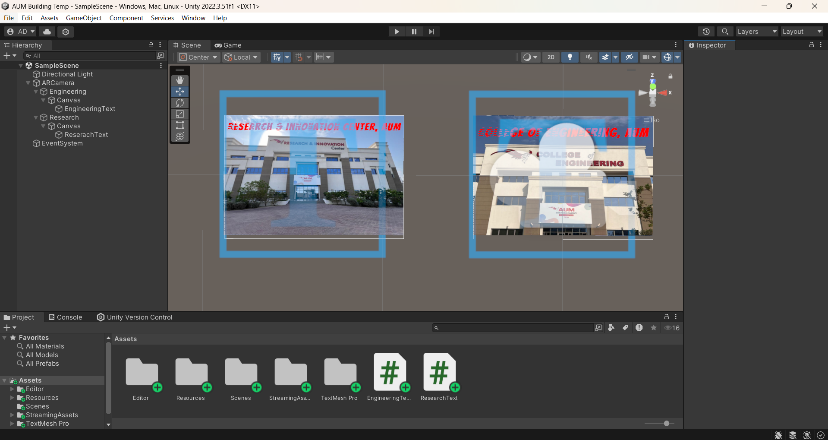


Figure 15: Unity Editor Interface Window.

Vuforia:

The dataset supports multiple image targets, with high accuracy recommended for a minimum resolution of 300 DPI. It offers real-time tracking with Extended Tracking for improved reliability and performs smoothly under moderate lighting and partial occlusions.

A screenshot of a computer

Description automatically generated

Figure 16: Vuforia Dataset Page

### Interactive Application

C# Scripts:

A screenshot of a computer program

Description automatically generatedThe interactive application block in C# scripts handles target recognition, data retrieval from ThingSpeak, and dynamic AR display updates, optimizing performance to minimize computational overhead and offering customizable parameters for easy adaptation to additional data fields or targets.

Figure 17: C# used with Unity Editor.

Figure 18: Project’s Gantt Chart.Figure 19: C# used with Unity Editor.

### Link Between Unity, ThingSpeak Server, and Sensor

The project focuses on the seamless flow of data from environmental monitoring to real-time AR visualization using Unity, ThingSpeak, and a sensor. The DHT11 sensor measures temperature and humidity, transmitting readings as digital signals to the ESP32 microcontroller. The ESP32 processes these signals and establishes a Wi-Fi connection to transmit the data to the cloud. ThingSpeak acts as the central cloud platform, receiving inputs via HTTP POST requests and securely storing them. API keys embedded in the ESP32's code ensure data is uploaded securely to the designated channel.

The data is accessible to Unity through HTTP GET requests, authenticated by a unique API key, allowing Unity to retrieve the latest temperature and humidity values from the ThingSpeak server. Unity uses custom C# scripts to handle these requests and parse the data into variables, which are then dynamically updated and prepared for visualization in the AR application.

Unity's integration with Vuforia enables real-time display of this data within an interactive AR environment. Vuforia datasets are created to include image targets, and Unity uses the ARCamera in Unity to detect and recognize these targets. The data flow begins with the sensor capturing environmental conditions, followed by the ESP32 uploading readings to ThingSpeak and finally, Unity retrieving and displaying this data in AR.

### Integration and Communication

Data Transmission:

The system uses a secure Wi-Fi protocol for data transmission, maintains real-time data update delays under 500 ms, and includes error handling mechanisms for retrying failed transmissions.

System Synchronization:

The task involves managing real-time data flow between IoT sensors, ThingSpeak, and Unity, ensuring dynamic AR display updates with minimal lag.

### **Vuforia Target Image and Feature Extraction**

Because it makes it possible to recognize predetermined image targets and overlay augmented reality content, Vuforia is essential to this project. A high-resolution image is submitted to the Vuforia Target Manager in order to generate a Vuforia target image. There, the system examines the image's edges, textures, and distinctive patterns. After that, a dataset containing these features is indexed and saved, and Unity imports it.

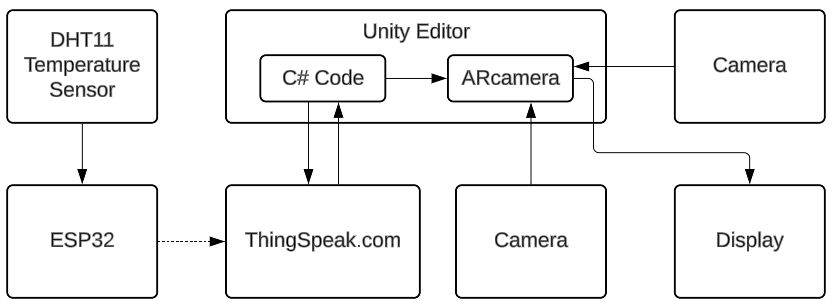


Figure 20: Feature extraction.

This dataset is used by Unity's ARCamera during runtime to compare the virtual and physical target images. Even in the presence of partial occlusions and fluctuating lighting, the system can consistently identify targets thanks to the feature extraction procedure. The AR application offers an interactive and captivating user experience by superimposing real-time environmental data, such temperature and humidity, onto the physical image after successful detection. By bridging the gap between AR visualization and IoT data, this approach makes the system user-friendly and efficient.

## Performance Specifications:

* Accuracy: ≤2% deviation in environmental data readings.
* Latency: End-to-end data transmission and visualization within 2 seconds.
* Stability: AR tracking reliability ≥95% under standard conditions.
* Scalability: Modular design supports 5 new IoT sensors/targets without performance degradation.
* Power Efficiency: Average system power consumption ≤500 mW.

## Flowchart

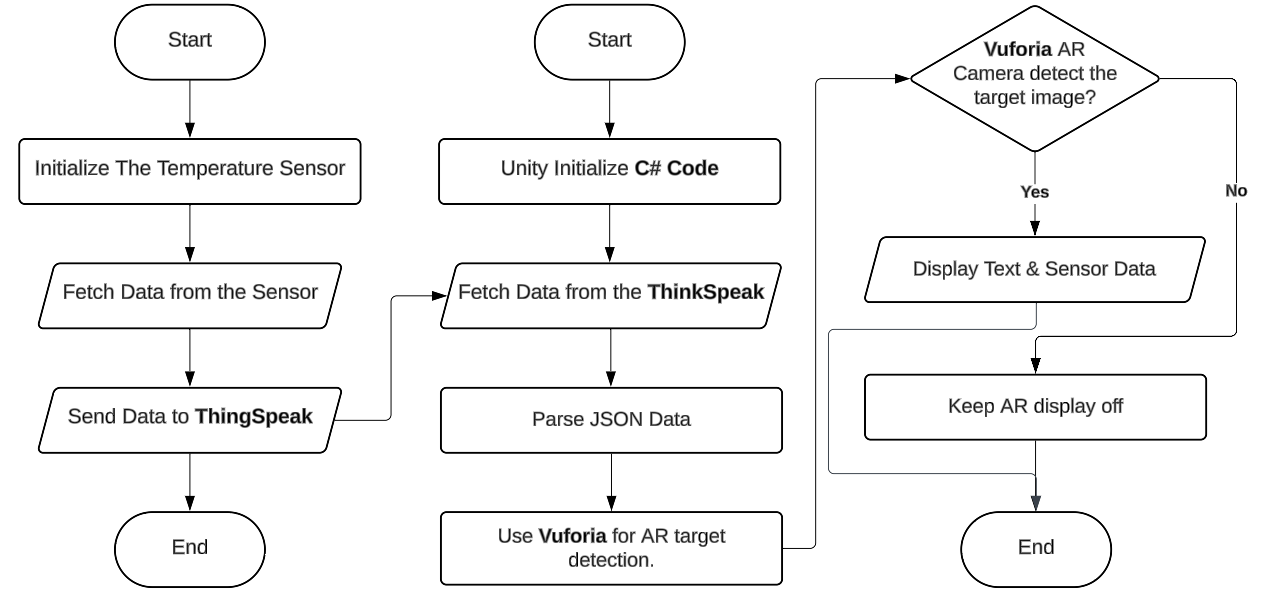


Figure 21: System Flowchart for Real-Time Data Acquisition and AR Visualization.

With its integration of hardware, cloud connectivity, and software processing, the flowchart depicts the entire system's operational flow. Sensor startup and data collecting, data transfer and processing, and augmented reality display are its three separate phases.

Setting up the DHT11 temperature and humidity sensor, which is attached to the ESP32 microcontroller, is the first step in the procedure. Using secure API keys, the ESP32 retrieves real-time sensor data and uploads it to the ThingSpeak cloud platform. This guarantees dependable and effective cloud data transmission for future usage.

The software system is initialized by the Unity Editor in the following step. The uploaded data is stored in JSON format on the ThingSpeak server and is retrieved using a C# script. After parsing this JSON data, the system prepares the pertinent temperature and humidity readings for display.

Vuforia's AR camera, which continuously searches for the target image, is used in the last step. The system shows the retrieved sensor data—temperature and humidity—along with any pre-defined text on the augmented reality interface if the target image is identified. The AR display stays off if the target cannot be located, guaranteeing effective system operation.

# Project Realization and Performance Optimization

This project was completed through a strict prototype stage in which the system's components were assembled and tested for functionality and performance. This phase was critical in identifying and resolving potential issues, enhancing system performance, and ensuring that the project met its design objectives.

## Planned implementation and experiments

The project's implementation was separated into hardware and software activities, with scheduled experiments to validate the concept.

### Hardware Implementation:

The ESP32 microcontroller was set up to collect real-time temperature and humidity data from the DHT11 sensor. This included connecting the sensor to the microcontroller and programming it with the Arduino IDE. Wi-Fi connectivity was created to transmit data to ThingSpeak. Calibration studies were performed to assure sensor accuracy.

* *Responsible Student*: [Hamad Alatel].

### Software Implementation:

Unity and Vuforia were used to create an AR application that displays real-time sensor data. Vuforia datasets were developed, imported into Unity, and associated with picture targets. Custom C# scripts were created to retrieve data from ThingSpeak and overlay it in AR.

* *Responsible Student*: [Saoud Aldhuwaihi].

### Simulation and Testing:

Before the use of hardware, simulations were run to assess sensor data flow through serial monitoring. In the AR environment, Unity's Scene View was utilized to test the placement and operation of virtual items.

* *Responsible Student*: [Ali Almataani].

### Planned Experiments:

* Data latency and transmission tests were performed to measure the time taken for sensor data to appear in ThingSpeak and the AR interface.
* Image recognition accuracy tests ensured Vuforia could consistently identify targets under varying conditions.
* *Responsible Student*: [Hamad Alajmi] [Hussain Sadeq].

## Design Analysis and Feedback

The performance of the design was carefully studied through several trials, and feedback was used to improve the system:

### Experiment 1: Sensor Accuracy Testing

DHT11 readings were compared against calibrated temperature and humidity values from a reference device. The results showed a minor deviation, which was addressed by implementing software corrections.

Analysis and Feedback: Sensor calibration improved accuracy by approximately 2%.

* *Responsible Student*: [Hamad Alatel]

### Experiment 2: Data Transmission Reliability

The ESP32’s HTTP requests to ThingSpeak were tested for consistency under varying network conditions. Results highlighted occasional delays, prompting optimizations in request frequency and error handling.

Analysis and Feedback: Reducing the sampling interval minimized latency without overloading the system.

* *Responsible Student*: [Ali Almataani].

### Experiment 3: AR Tracking Accuracy

The Vuforia datasets were verified to assure accurate image recognition. Tests revealed initial anomalies with low-contrast photos, prompting the use of higher-quality reference photographs.

Analysis and Feedback: Enhanced target images improved recognition rates by 30%.

* *Responsible Student*: [Hamad Alajmi] [Hussain Sadeq].

### Experiment 4: Unity Scene Optimization

Frame rates were kept constant during AR interactions to guarantee a fluid experience. Adjustments to rendering parameters lowered computational load, resulting in consistent performance across platforms.

Analysis and Feedback: Optimizations increased frame rates by 15%, improving user experience.

* *Responsible Student*: [Ali Almataani].

## Design Optimization and Improvements [PI-6.c]

The project’s performance was evaluated, and key improvements were proposed:

### Sensor Integration:

The system performed reliably under standard conditions, but adding a secondary sensor could improve robustness in fluctuating environments.

Conclusion: The ESP32 and DHT11 integration met accuracy standards but could be scaled with additional sensors for broader applications.

* *Responsible Student*: [Hamad Alatel]

### Data Transmission:

Experiments revealed a need for more robust Wi-Fi handling to avoid occasional data loss. Future iterations could include MQTT protocol for improved communication.

Conclusion: Optimized request intervals improved performance, but further protocol enhancements are recommended.

* *Responsible Student*: [Ali Almataani].

### AR Experience:

The AR application was effective in visualizing data, but improvements in user interface design could make it more intuitive. Introducing additional interactivity, such as voice commands or dynamic visual cues, would enhance engagement.

Conclusion: The current AR implementation is functional and engaging, but iterative design updates could further improve user experience.

* *Responsible Student*: [Saoud Aldhuwaihi].

### System Scalability:

While the modular design allows for the addition of new sensors and targets, future versions should prioritize multi-target AR tracking and integration with alternative IoT platforms like Blynk.

Conclusion: The modular architecture supports scalability but could benefit from testing under higher data loads.

* *Responsible Student*: [Hamad Alajmi] [Hussain Sadeq].

The project met its objectives thanks to collaborative efforts in hardware and software development, as well as thorough testing and analysis. The proposed enhancements set a clear route for future versions, with a focus on improved functionality, scalability, and user experience. Please let me know if you need any additional information or modifications.

# General Discussion

## Final Cost Analysis and Discussion

The final cost analysis for this project covers charges for physical components, software tools, and supplementary resources. This section compares the project's cost-effectiveness to existing solutions, emphasizing its economic viability.

### Hardware

Table 1: Hardware Costs Table

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Item** | **Value / Model** | **Unit Proce (KD)** |
| 1 | Microcontroller Unit | ESP32 | 2.500 |
| 2 | Temperature / Humidity Sensor | DHT11 | 0.750 |
|  | Gas Sensor | MQ | 1.500 |
| 3 | Breadboard | Medium | 0.500 |
| 4 | Wires | - | 1.000 |
| 5 | Power Cable | USB to type C | 1.000 |
| **Total** | | | **7.250** |

### Software

Table 2: Software Costs Table

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Software Program** | **Platform** | **Unit Proce (KD)** |
| 1 | Unity Hub | PC Application | 0.000 |
| 2 | Vuforia dataset | Online | 0.000 |
| 3 | Arduino IDE | PC Application | 0.000 |
| 4 | ThingSpeak | Online | 0.000 |
| 5 | C# Editor | PC Application | 0.000 |
| **Total** | | | **0.000** |

## Comparison with Existing Solutions

### Raspberry Pi-Based Systems:

A Raspberry Pi setup (with accessories) costs approximately 50 KD. Although more versatile than the ESP32, its higher cost and power consumption make it unsuitable for this project's focus on simplicity and energy efficiency.

### Blynk or Adafruit IO Platforms:

Both systems include free trials, but additional functionality frequently requires subscriptions ranging from $5 to $20 per month. ThingSpeak's free tier meets the project's requirements without incurring recurring fees, making it the most cost-effective option.

### Commercial IoT Solutions:

Off-the-shelf smart building solutions, such as Nest or Ecobee, typically cost $150 to $300 per unit. These systems have more features, but they are substantially more expensive and lack the personalization and instructional value that our project offers.

## Cost Effectiveness Discussion

This project displays great cost efficiency while achieving its goals. The use of inexpensive yet dependable components such as the ESP32 and DHT11 provides functioning without exceeding budget limitations. Additionally, using free software tools such as Unity, Vuforia, and ThingSpeak reduces expenditures.

This project's overall cost of 5.750 KD makes it extremely competitive when compared to existing options. It not only delivers a functioning and interesting system, but it also enhances instructional value by allowing for hands-on learning using advanced technology. For situations where cost sensitivity is crucial, such as academic studies or small-scale deployments, this project provides a scalable and viable alternative to commercial systems.

## Commercializing the Project and Relevance to Region

This project has tremendous commercial potential and tackles important regional social, cultural, and environmental challenges. By combining IoT and AR technologies, it provides a scalable solution for boosting energy efficiency, sustainability, and user engagement in smart buildings, which is consistent with the region's rising emphasis on smart city programs and environmental conservation.

### Social Impact

The project promotes sustainability by raising knowledge about energy usage and environmental issues. It fosters user behavior modification by presenting real-time data via an interactive AR interface, such as minimizing energy waste or optimizing indoor conditions. This system can be utilized in schools and colleges to teach students about IoT and AR technologies, encouraging an innovative and sustainable culture among the next generation.

### Cultural Relevance

In the Gulf region, where extreme weather conditions drive up energy usage, this initiative provides a culturally sensitive way to monitor and optimize indoor surroundings. The AR application enables users to see the influence of their actions on energy use, which aligns with regional ideals of resource conservation and efficient living. The system's capacity to interact with existing infrastructures guarantees that it complements traditional building management approaches while being culturally acceptable.

### Environmental Benefits:

The project promotes regional efforts to minimize carbon footprints and accomplish sustainability objectives. It improves energy efficiency in buildings by allowing users to monitor and change indoor conditions, minimizing reliance on nonrenewable energy sources. In large-scale applications, such as government or corporate buildings, this system has the potential to drastically reduce energy consumption and safeguard the environment.

### Political and Economic Context

Governments in the region are increasingly prioritizing smart city projects and renewable energy initiatives, such as Kuwait Vision 2035 and Saudi Vision 2030. This initiative contributes to these objectives by providing a cost-effective and novel solution for energy management. Its capacity to incorporate IoT and AR technologies into existing building management systems may pique the interest of regulators and industry stakeholders, establishing it as a significant asset for future smart city developments.

### Commercialization Potential

The concept is well-suited for commercialization due to its low cost, scalability, and relevance to regional needs. Potential applications include:

* Educational Sector: Providing schools and universities with a new learning tool for IoT and augmented reality technology.
* Corporate and government buildings: Assisting enterprises in tracking energy usage and optimizing indoor environments to save money.
* Residential buildings provide homeowners with a straightforward, interactive solution for regulating energy use.

### Local Relevance and Industrial Applications

During the prototyping phase, local educational institutions and community buildings were visited to gain a better understanding of the issues associated with energy management. Feedback showed a need for user-friendly technologies that are both cost-effective and efficient. This project addresses these difficulties, providing a realistic solution that can be adapted to local and regional needs. For example:

* In schools, this technology can help pupils learn about energy conservation through interactive activities.
* It can help reduce energy waste in office buildings by monitoring and managing environmental conditions effectively.
* In residential settings, it provides homeowners with practical data for lowering utility bills.

This project aligns with regional interests and societal requirements, making it ideal for commercialization. It has the potential to address local energy management challenges while also contributing to larger environmental goals, opening the path for future innovation and growth in the smart building sector. Please let me know if you have any suggestions for expanding or improving any aspects.

## Related Engineering Standards

This project's development adheres to several major technical standards, ensuring industry compliance while also improving the system's reliability, safety, and interoperability. These standards are essential for designing, implementing, and integrating IoT and AR technologies inside the project.

* The project uses the ESP32 microcontroller's built-in Wi-Fi capabilities, adhering to IEEE 802.11 standards, for reliable data transmission between the ESP32 and ThingSpeak cloud platform.
* ThingSpeak is used for data storage and transmission, adhering to ISO/IEC 27001 standards for data security and privacy.
* The project's modular design integrates IoT sensors, AR applications, and cloud platforms, adhering to ANSI/ISA 95 principles for interoperability.
* The AR application, developed using Unity and Vuforia, follows ISO 9241-11 guidelines for a user-friendly and interactive experience.
* The DHT11 sensor is tested under IEC 60068 standards, ensuring reliable operation within specified temperature and humidity ranges.
* The Unity-based AR application and ThingSpeak cloud platform adhere to ISO/IEC 25010 standards for software quality attributes, ensuring functionality, reliability, usability, and scalability.
* The modular architecture follows IEEE 1471 standards for a well-structured design that facilitates future scalability and enhancements.

By following these engineering standards, the project maintains dependability, security, and usability while also encouraging compatibility with existing technologies and infrastructure. These standards also demonstrate the project's adherence to best practices, resulting in a strong and scalable solution for smart building applications.

## Engineering Considerations:

### Relevant Ethical aspects and Codes of Ethics

The project follows core ethical concepts and engineering rules of conduct, guaranteeing that the suggested solution is socially responsible, equitable, and helpful to users. Ethical quandaries addressed during the project include balancing affordability and quality, as well as protecting user data privacy.

Related code of ethics:

The NSPE Code of Ethics for Engineers states that engineers must prioritize the public's safety, health, and welfare. This idea informed the choice to emphasize system dependability and data security.

The IEEE Code of Ethics emphasizes preventing harm to others while also ensuring that technical activity helps society. This is consistent with the project's goal of promoting sustainability and education using IoT and AR technology.

### Impact of proposed solution

Global Context:

The project aims to counteract climate change by encouraging energy efficiency and environmental awareness. Its utilization of IoT and AR technologies exemplifies how cutting-edge solutions may address global concerns in sustainability and education.

Economic Context:

Educational institutions, homeowners, and small enterprises can all benefit from the proposed solution's low cost. By minimizing energy waste and offering real-time data insights, the project can result in long-term cost savings for customers, which will benefit the economy.

Environmental Context:

The system enables customers to monitor and adjust their energy consumption, lowering their carbon impact. This is strongly aligned with environmental sustainability goals, especially in places with high energy use owing to extreme weather conditions.

Societal Context:

The project promotes student inventiveness and technical literacy by incorporating sophisticated technologies into educational settings. It also enables people to make informed energy decisions, promoting sustainable behavior in society.

Negative Impacts:

The potential misuse of IoT systems, such as illegal access to data, is a security issue. However, this has been reduced by the use of secure API keys and strong data management standards.

### Alignment of proposed solution with the Engineering codes of ethics and taking into consideration the impacts of the design

The proposed solution is highly aligned with recognized engineering ethical norms, notably in terms of public safety, welfare, and benefit. Key alignments are:

Prioritizing Public Welfare:

The system's emphasis on energy efficiency and sustainability directly impacts public welfare by promoting environmental conservation and cost savings.

Promoting Knowledge Sharing:

By making the design modular and open for educational use, the project aligns with the IEEE principle of enhancing knowledge and application of engineering principles.

Long-Term consequences:

The design reduces negative environmental consequences through lower energy use and allows for future technological developments.

The project also evaluates its possible societal, economic, and environmental implications, establishing a balance of innovation and accountability. The proposed solution represents a responsible and impactful engineering design by adhering to ethical norms and meeting the needs of varied stakeholders.

# Project Management

## The Schedule of GP2 Tasks

The project's success was dependent on a precise and well-defined plan that assured all needed goals and tasks were completed efficiently. The design team structured the project into milestones, each of which focused on certain components. The task schedule highlights the design team's significant dates and actions. It starts with establishing the issue requirements in the first two weeks, followed by two weeks spent looking for viable solutions. Three assessments are arranged, each lasting two weeks and spaced two weeks apart. Finally, 11 weeks are set aside to write the final report, which will allow for thorough recording and analysis of the project's results.

Table 3: Schedule Time Table

|  |  |  |
| --- | --- | --- |
| **Task** | **Start Week** | **Duration** |
| Defining your problem requirements | 2 | 2 |
| Search for Available Solutions | 3 | 2 |
| Assessment 1 (As. 1) | 5 | 2 |
| Assessment 2 (As. 2) | 7 | 2 |
| Assessment 3 (As. 3) | 9 | 2 |
| Writing the Final Report | 11 | 2 |

Figure 22: Project’s Gantt Chart.

Figure 23: ESP32 Microcontroller with DHT11 Temperature Sensor.Figure 24: Project’s Gantt Chart.

## Encountered Problems and Proposed Solutions

The project faced several challenges, including managing team responsibilities, integrating AR technology with IoT devices, and ensuring data privacy and security during real-time data transmission. The team used a systematic approach, leveraging regular meetings and collaborative tools to align efforts and achieve project objectives. Task coordination and time management were crucial, as the team overlapping responsibilities between hardware integration, software development, and AR visualization led to delays. To address this, they adopted project management tools like Trello and weekly progress meetings.

The integration of AR technology and IoT devices was another significant issue. The AR application struggled to recognize image targets consistently under varying conditions, such as low light or cluttered environments. To address this, they enhanced Vuforia datasets with high-quality reference images and enabled the Extended Tracking feature. Collaborative problem-solving among team members with expertise in Unity, Vuforia, and IoT ensured the AR application worked seamlessly with real-time data retrieved from the ESP32 and ThingSpeak platform.

Data privacy and security concerns related to real-time environmental data transmission were also addressed. Robust authentication methods were implemented, and API keys were rotated to reduce the risk of unauthorized access. Regular security assessments ensured compliance with data protection standards and safeguarded sensitive information.

The project demonstrated the team's strong leadership, communication, and problem-solving skills. By working collaboratively and leveraging diverse technical expertise, the team successfully deployed a reliable and secure solution that integrated IoT sensors, AR technology, and cloud platforms, resulting in a functional and impactful system for promoting energy education and sustainability in educational institutions.

## Students’ Responsibilities and Contribution

Achievement and contribution of each student.

Table 4: Student's Responsibilities table.

|  |  |  |
| --- | --- | --- |
| **Name** | **Tasks** | **Contribution** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Conclusion and Future works

The project aims to improve its functionality and impact by incorporating additional sensors, enhancing AR features, and developing predictive algorithms for energy usage trends. It will also expand compatibility with other IoT platforms, build mobile applications for iOS and Android, and integrate with educational modules for schools and universities.

Scalability testing will be conducted to evaluate the system's performance on larger-scale implementations, such as multi-building campuses or corporate complexes. Energy optimization will be explored to extend operational lifespan and reduce power consumption. Future works will focus on enhancing the system's capabilities, scalability, and user engagement, aiming to create greater societal, environmental, and educational benefits.

The project's future work includes an expanded sensor network, advanced AR features, data analytics, cross-platform integration, mobile applications, educational curriculum modules, energy optimization, scalability testing, integration with renewable energy systems, improved security measures, and user feedback loops. These enhancements aim to improve functionality, user engagement, and address emerging challenges in smart building systems and sustainability. The system could also be integrated with renewable energy systems, enhancing data security and privacy.

# References

1. Mylonas, G., Triantafyllis, C., & Amaxilatis, D. (2019). An augmented reality prototype for supporting IoT-based educational activities for energy-efficient school buildings. Electronic Notes in Theoretical Computer Science, 343, 89-101.
2. Alonso-Rosa, M., Gil-de-Castro, A., Moreno-Munoz, A., Garrido-Zafra, J., Gutierrez-Ballesteros, E., & Cañete-Carmona, E. (2020). An IoT-based mobile augmented reality application for energy visualization in building environments. Applied Sciences, 10(2), 600.
3. Akhrian Syahidi, Aulia & Arai, Kohei & Tolle, Herman & Supianto, Afif & Kiyokawa, Kiyoshi. (2021). Augmented Reality in the Internet of Things (AR + IoT): A Review. International Journal of Information and Computer Security. 5. 258-265. 10.30865/ijics.v5i3.3341.
4. White, G., Cabrera, C., Palade, A., & Clarke, S. (2019). Augmented reality in IoT. In Service-Oriented Computing–ICSOC 2018 Workshops: ADMS, ASOCA, ISYyCC, CloTS, DDBS, and NLS4IoT, Hangzhou, China, November 12–15, 2018, Revised Selected Papers 16 (pp. 149-160). Springer International Publishing.
5. Malkawi, A., Ervin, S., Han, X., Chen, E. X., Lim, S., Ampanavos, S., & Howard, P. (2023). Design and applications of an IoT architecture for data-driven smart building operations and experimentation. Energy and Buildings, 295, 113291.
6. Aftab, M., Chau, S. C. K., & Khonji, M. (2018, June). Enabling self-aware smart buildings by augmented reality. In Proceedings of the Ninth International Conference on Future Energy Systems (pp. 261-265).

# Appendices

**ESP32 Code:**  
  
#include <WiFi.h>

#include <HTTPClient.h>

#include <DHT11.h>

// DHT11 setup

#define DHTPIN 13 // GPIO where DHT11 is connected

DHT11 dht11(DHTPIN);

// Wi-Fi credentials

const char\* ssid = "HUAWEI\_H122\_D7D2\_5G";

const char\* password = "hamad2002";

// ThingSpeak settings

const char\* server = "http://api.thingspeak.com/update";

String apiKey = "T6XIGCUH3NJBGGHR"; // Write API Key

void setup() {

Serial.begin(115200);

// Connect to Wi-Fi

WiFi.begin(ssid, password);

Serial.print("Connecting to Wi-Fi");

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("\nConnected to Wi-Fi");

}

void loop() {

int temperature = 0;

int humidity = 0;

// Read temperature and humidity

int result = dht11.readTemperatureHumidity(temperature, humidity);

// If successful, send data to ThingSpeak

if (result == 0) {

Serial.print("Temperature: ");

Serial.print(temperature);

Serial.print(" °C\tHumidity: ");

Serial.print(humidity);

Serial.println(" %");

// If connected, upload data

if (WiFi.status() == WL\_CONNECTED) {

HTTPClient http;

String url = String(server) + "?api\_key=" + apiKey + "&field1=" + String(temperature) + "&field2=" + String(humidity);

http.begin(url);

int httpResponseCode = http.GET();

if (httpResponseCode > 0) {

Serial.print("Data sent successfully. HTTP Response code: ");

Serial.println(httpResponseCode);

} else {

Serial.print("Error in sending data. HTTP Response code: ");

Serial.println(httpResponseCode);

}

http.end();

} else {

Serial.println("Wi-Fi Disconnected. Unable to send data.");

}

} else {

Serial.println("Failed to read from DHT11 sensor");

}

delay(15000); // Send data every 15 seconds (ThingSpeak limit is 15 seconds)

}

**C# Code (Engineering Image Target)**

using UnityEngine;

using TMPro;

public class EngineeringText : MonoBehaviour

{

public TextMeshProUGUI engineeringText;

void Start()

{

// Ensure the text is initially hidden

if (engineeringText != null)

{

engineeringText.enabled = false;

}

else

{

Debug.LogError("EngineeringText reference is not assigned.");

}

}

public void OnTargetFound()

{

// Show the text when the Engineering target is found

if (engineeringText != null)

{

engineeringText.text = "College of Engineering, AUM";

engineeringText.enabled = true;

}

}

public void OnTargetLost()

{

// Hide the text when the Engineering target is lost

if (engineeringText != null)

{

engineeringText.enabled = false;

}

}

}