   
**Technological University Dublin**   
**Blanchardstown Campus**

Bachelor of Engineering in Computer Engineering

H3024 Project 2

*Project Title*

Remote machine monitoring and tracking device

*Author*

Erikas Valiukevicius

B00141111

TU807

*Supervisor*

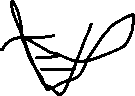
Mark Deegan

*Date*

18/2/2024

# Declaration

The material contained in this assignment is the author’s original work, except where work quoted is duly acknowledged in the text. No aspect of this assignment has been previously submitted in any other unit or course.



Signed: Date: 18/2/2024

# Abstract

# Acknowledgements

I would like to thank Mark Deegan, Ivan smyth and Fergus maughan for helping me in the development of this project and assisting in my learning throughout my time in their care.

I would also like to thank all my other lecturers whose classes aided me in the understanding and development of the necessary skills to be able to perform at my best for this project.

# Table of Contents

Declaration ii

Abstract iii

Acknowledgements iv

Table of Contents v

Abbreviations, Symbols, Acronyms vi

List of Figure Captions viii

List of Tables ix

Product Specifications 1

Executive Summary 2

Chapter 1 Introduction 3

1.1 The Aim: 3

1.2 Motivation: 4

1.3 Layout of Report: 4

Chapter 2 Background Research 5

Chapter 3 Project Design 6

3.1 **Theory and reasoning for the choice.** 6

3.2 **Hardware design** 6

3.3 Software Design 10

3.4 Software Flow Diagram 11

3.6 Discussion 11

Chapter 4 Project Construction and Testing 13

4.1 Construction 13

4.2 Testing Procedure 13

4.3 Software Testing 13

4.4 Fault Tree 13

4.5 Results 13

Chapter 5 Safety and Ethical Considerations of the Project. 14

5.1 Health and Safety Issues: 14

5.2 Ethical Considerations: 14

Chapter 6 Conclusions and Recommendations 16

References 17

Appendix 18

Timetable 19

# Abbreviations, Symbols, Acronyms

1. **ESP-32**: Espressif Systems' ESP32, a low-power system-on-chip microcontroller widely used in IoT applications.
2. **IoT**: Internet of Things, a network of interconnected devices that communicate and exchange data.
3. **LoRa**: Long Range, a wireless communication technology for long-range, low-power IoT applications.
4. **NB-IoT**: Narrowband Internet of Things, a cellular communication standard optimized for IoT devices.
5. **Wi-Fi**: Wireless Fidelity, a wireless networking technology based on IEEE 802.11 standards.
6. **Bluetooth**: A wireless technology standard for short-range communication between devices.
7. **GPS**: Global Positioning System, a satellite-based navigation system providing location and time information.
8. **MQTT**: Message Queuing Telemetry Transport, a lightweight messaging protocol for IoT applications.
9. **Azure**: Microsoft Azure, a cloud computing platform for building, deploying, and managing applications and services.
10. **RoHS**: Restriction of Hazardous Substances, a directive restricting the use of certain hazardous materials in electrical and electronic equipment.
11. **CE**: Conformité Européenne, a certification mark indicating conformity with health, safety, and environmental protection standards for products sold within the European Economic Area.
12. **UL**: Underwriters Laboratories, a safety certification company providing safety-related certification, validation, testing, inspection, and auditing services.
13. **GDPR**: General Data Protection Regulation, a European Union regulation on data protection and privacy for all individuals within the EU and the European Economic Area.
14. **ISO**: International Organization for Standardization, an international standard-setting body responsible for developing and publishing standards.
15. **HIPAA**: Health Insurance Portability and Accountability Act, a US legislation ensuring data privacy and security provisions for safeguarding medical information.
16. **CCPA**: California Consumer Privacy Act, a state statute intended to enhance privacy rights and consumer protection for residents of California, United States.
17. **ISO 9001**: International Organization for Standardization's standard for quality management systems, demonstrating a commitment to meeting customer requirements and continuous improvement.
18. **ISO 14001**: International Organization for Standardization's standard for environmental management systems, focusing on minimizing environmental impact and improving resource efficiency.
19. **GD**: Global Distribution, a supplier of electronic components and technologies.
20. **RS**: RS Components, a global supplier of electronic components and solutions.
21. **Farnell**: Farnell Ireland, a distributor of electronic components and related products.
22. **SEEED STUDIO**: A manufacturer and supplier of electronic hardware and open-source hardware platforms.
23. **ADAFRUIT INDUSTRIES**: A company specializing in open-source hardware, including microcontroller boards, sensors, and DIY electronics kits.

# List of Figure Captions

[An automated list similar to the table of contents]

# List of Tables

[An automated list similar to the table of contents]

# Product Specifications

(Consult with you supervisor as to the type of list of specification (if any) needed here.)

# Executive Summary

(This section summarizes the entire report in such a way that the reader can gain all the salient information contained in the main report without having to read the entire report)

# Chapter 1 Introduction

# 1.1 The Aim:

The primary objective of this project is to design, develop, and implement a remote machine monitoring and tracking device. This device aims to mitigate issues related to machinery mistreatment, abuse, and breakdowns by providing real-time monitoring and logging capabilities. By accurately tracking machinery data, such as temperature, pressure, speed, impact, and location, the device will enable businesses to proactively address maintenance needs, optimize machine usage, and reduce overall operational costs.

The following unique targets guide the venture:

Hardware Integration: Integrate the ESP32-S3 microcontroller with ancillary hardware components, the use of verbal exchange protocols like I2C and SPI.

Wireless Connectivity: Implement wireless networking using the integrated 2.4 GHz Wi-Fi and Bluetooth modules of the ESP32-S3. Establish verbal exchange with a personal hotspot to enable faraway device management.

MQTT Configuration: Configure the MQTT protocol at the ESP32-S3 to facilitate lightweight messaging and connectivity to other additives within the device. Utilize the Mosquitto message broker for efficient communication.

Cloud Services: Deploy a virtual gadget on Microsoft Azure running the Ubuntu Linux OS. Set up and configure Docker packing containers for crucial services which includes NodeRED, Mosquitto, InfluxDB, Grafana, and Portainer to create a scalable and modular backend infrastructure.

Docker and Containerization: Use Docker and Docker Compose for efficient containerization, allowing the deployment and control of numerous offerings inside isolated bins.

NodeRED Implementation: Develop and deploy NodeRED flows to facilitate the implementation of flow-based programming, allowing for seamless data flow between nodes and enhancing system flexibility.

System Testing: Conduct thorough testing of the entire system, including individual hardware components, communication protocols, cloud services, and visualization tools. Ensure the reliability, stability, and security of the system.

# 1.2 Motivation:

The motivation behind this project is multifaceted.

Firstly, businesses across various industries face significant challenges related to machinery maintenance, with issues such as mistreatment, abuse, and unexpected breakdowns leading to costly repairs and downtime.

By implementing a remote monitoring and tracking solution, businesses can gain better insights into machinery health, identify potential issues early on, and take proactive measures to prevent breakdowns and minimize downtime.

Moreover, there is a growing emphasis on environmental sustainability and reducing carbon emissions.

Machinery breakdowns not only incur financial costs but also contribute to increased resource consumption and environmental impact.

By implementing an effective monitoring system, businesses can optimize machinery usage, reduce the need for repairs and replacements, and ultimately minimize their carbon footprint.

# 1.3 Layout of Report:

This report is structured to provide a comprehensive overview of the entire project lifecycle. It begins with an introduction outlining the project objectives and motivation. Following this, the report delves into background research, exploring relevant technologies, platforms, and concepts essential for understanding the project scope.

Subsequently, the report details the project design phase, including hardware and software design considerations, construction procedures, and testing methodologies. Additionally, safety and ethical considerations are addressed to ensure the project adheres to relevant standards and regulations, safeguarding both human safety and data privacy.

Towards the conclusion, the report presents key findings, conclusions, and recommendations based on the project outcomes. It provides insights into the efficacy of the developed solution, its potential impact on businesses and the environment, and areas for future research and improvement.

Overall, this report serves as a comprehensive documentation of the project, offering valuable insights into the development of a remote machine monitoring and tracking device and its implications for various stakeholders.

# Chapter 2 Background Research

2.1 **MIT App Developer**: The MIT App Inventor is a web application integrated development environment originally provided by Google, and now maintained by the Massachusetts Institute of Technology (MIT). It allows users to create software applications for the Android operating system using a visual drag-and-drop interface.

2.2 **Node-RED**: Node-RED is a flow-based development tool for visual programming developed by IBM. It is built on Node.js and provides a browser-based editor that makes it easy to wire together devices, APIs, and online services.

2.3 **MQTT (Message Queuing Telemetry Transport)**: MQTT is a lightweight messaging protocol ideal for IoT applications. It is designed for high-latency or unreliable networks and follows a publish-subscribe messaging pattern, making it efficient for communication between devices and servers.

2.4 **Azure**: Microsoft Azure is a cloud computing platform that offers a wide range of services, including computing, analytics, storage, and networking. Azure provides robust infrastructure and platform services to support IoT solutions, including data storage, analytics, and machine learning capabilities.

2.5 **ESP32**: The ESP32 is a low-power system-on-chip microcontroller with integrated Wi-Fi and Bluetooth capabilities. It is widely used in IoT applications due to its versatility, low cost, and power efficiency.

2.6 **Adafruit Ultimate GPS v3**: The Adafruit Ultimate GPS v3 module is a compact GPS module based on the MTK3339 chipset. It provides accurate positioning information with support for multiple satellite constellations, including GPS, GLONASS, and Galileo.

2.7 **LoRa (Long Range)**: LoRa is a long-range, low-power wireless communication technology designed for IoT applications. It operates in unlicensed radio bands and can transmit data over long distances with low power consumption, making it suitable for applications such as remote monitoring and asset tracking.

2.8 **NB-IoT (Narrowband Internet of Things)**: NB-IoT is a cellular communication standard optimized for IoT devices. It offers low-power, wide-area connectivity with extended coverage and is well-suited for applications that require long battery life and reliable connectivity in challenging environments.

# Chapter 3 Project Design

## 3.1 **Theory and reasoning for the choice.**

The project aims to address the issues of machinery mistreatment, abuse, and breakdowns by providing a comprehensive tracking and monitoring solution. The choice of technologies is based on their suitability for achieving this goal. MIT App Inventor and Node-RED were selected for their ease of use in developing mobile and IoT applications, respectively. MQTT was chosen as the messaging protocol for its lightweight and efficient communication over unreliable networks. Azure IoT Hub offers cloud-based data storage and management capabilities, making it suitable for storing sensor data. The ESP32 microcontroller was selected for its versatility and built-in Wi-Fi and Bluetooth connectivity. The Adafruit Ultimate GPS module provides accurate positioning data for tracking machinery locations.

## 3.2 **Hardware design**

1. **ESP32 Microcontroller:**
   * The ESP32 microcontroller serves as the central processing unit of the monitoring device, responsible for interfacing with sensors, establishing communication with the MQTT broker, and managing data transmission.
   * It offers built-in Wi-Fi capabilities, making it suitable for connecting to local networks and communicating with cloud services.
2. **Sensors:**
   * Temperature Sensors: Utilized for monitoring machine temperature levels, detecting overheating or abnormalities.
   * Pressure Sensors: Measure pressure levels in hydraulic systems or pneumatic systems, providing insights into machine performance.
   * Vibration Sensors: Detect mechanical vibrations and shocks, indicating potential machinery malfunctions or impacts.
   * Adafruit Ultimate GPS Module: Provides accurate location tracking and geolocation data, enabling real-time machine tracking and positioning.
3. **Power Supply:**
   * Power source considerations depend on the deployment environment and power requirements of the monitoring device.
   * Options include rechargeable batteries, solar panels for remote deployments, or direct power connections for stationary installations.
4. **Communication Interfaces:**

A wireless network is a computer network that uses wireless data connections between network nodes. Wireless networking allows homes, telecommunications networks and business installations to avoid the costly process of introducing cables into a building, or as a connection between various equipment locations.

Wireless networking for this project was our personal hotspot for the setup, as the esp32-s3 already has built in 2.4 GHz Wi-Fi and Bluetooth module installed

### I2C

I2C ( inter integrated Circuit) is a synchronous, multi-master, multi-slave communication protocol.

You can connect :

multiple slaves to one master: for example, your ESP32 reads from a sensor using I2C and writes the sensor readings in an I2C OLED display.

multiple masters controlling the same slave: for example, two ESP32 boards writing data to the same I2C OLED display.

We used I2C over for example the serial port as serial ports are asynchronous.

Meaning?

This means that there is no clock data transmitted over the serial port, devices need a clock data to synchronise over to communicate correctly with each other.

I2C communication happens over 2 communication lines SDA and SCL which are both active low and as such require pull up resistors.

### SPI

Serial Peripheral Interface is a de facto standard for synchronous serial communication, used primarily in embedded systems for short distance wired communication between integrated circuits.

These devices are organized into a master and slave configuration, in which the master has control over the slaves and the slaves receive instruction from the master.

There are four signals required to implement SPI communication.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **SIGNAL** |  | **DESCRIPTION** | | MOSI |  | Data: Master Out — Slave In | | MISO |  | Data: Master In — Slave Out | | SCLK |  | Serial Clock | | CS |  | Chip Select | |

The clock signal is used to synch the data received and transmitted between devices.

### TCP/ IP

Transmission Control Protocol (TCP) is a communications standard that enables application programs and computing devices to exchange messages over a network. It is designed to send packets across the internet and ensure the successful delivery of data and messages over networks.

TCP organizes data so that it can be transmitted between a server and a client. It guarantees the integrity of the data being communicated over a network. Before it transmits data, TCP establishes a connection between a source and its destination, which it ensures remains live until communication begins. It then breaks large amounts of data into smaller packets, while ensuring data integrity is in place throughout the process.

### MQTT

MQTT is a standard messaging protocol for the Internet of Things (IoT).

It is designed as an extremely lightweight publish/subscribe messaging transport that is ideal for connecting remote devices with a small code footprint and minimal network bandwidth. MQTT today is used in a wide variety of industries, such as automotive, manufacturing, telecommunications, oil and gas, etc.

In this project the MQTT was called Mosquitto which is an open-source message broker. It is a very lightweight system for carrying messages.

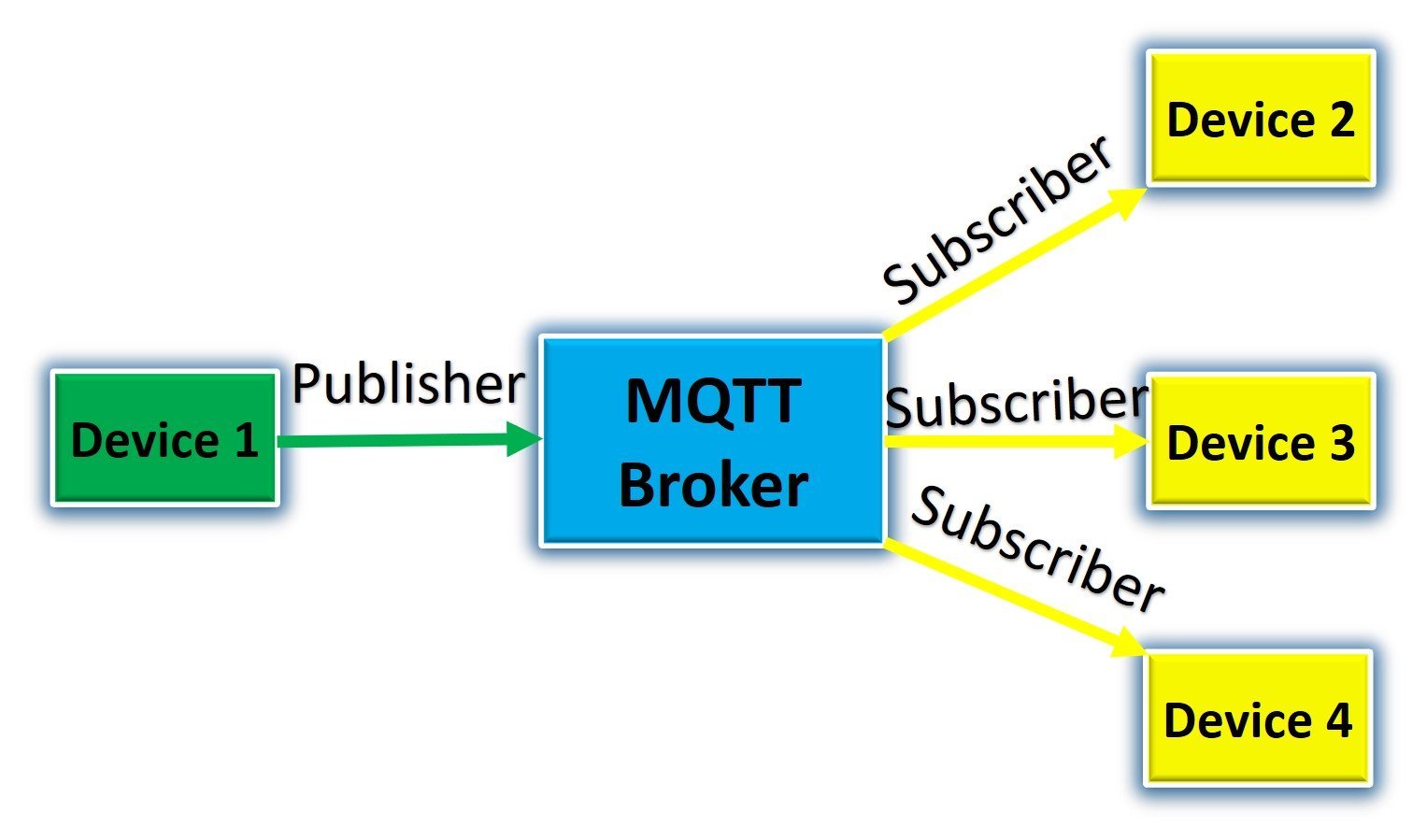


Figure 3 mqtt broker

### SSH

The Secure Shell (SSH) protocol is a method for securely sending commands to a computer over an unsecured network. SSH uses cryptography to authenticate and encrypt connections between devices. SSH also allows for tunnelling, or port forwarding, which is when data packets can cross networks that they would not otherwise be able to cross.

A screenshot of a computer program

Description automatically generated

Figure 4 ssh client

It's a bit like a secret passage for computer information. People often use SSH to control faraway computers, manage tech stuff, or share files without worrying about anyone spying.

Remote encrypted connections: SSH sets up a connection between a user's device and a faraway machine, often a server. It uses encryption to scramble the data that traverses the connection. An intercepting party would only find something like static — random data that means nothing unless it is decrypted. (SSH uses encryption methods that make decryption prohibitively difficult for outsiders.)

The ability to tunnel: In networking, tunneling is a method for moving packets across a network using a protocol or path they would not ordinarily be able to use. Tunneling works by wrapping data packets\* with additional information — called headers — to change their destination. SSH tunnels use a technique called port forwarding to send packets from one machine to another.

1. **Enclosure:**
   * The hardware enclosure provides protection for electronic components against environmental factors such as moisture, dust, and physical damage.
   * Enclosures should be designed to accommodate all components securely and allow for easy access to sensors and connectors for maintenance and troubleshooting.
2. **Environmental Considerations:**
   * Temperature, humidity, and exposure to elements may impact the performance and longevity of hardware components.
   * Selecting components rated for industrial or outdoor use ensures reliability and resilience in harsh environments.

A computer screen shot of a computer

Description automatically generated

## 3.3 Software Design

**Backend Infrastructure:**

**Cloud Database:** Utilizing Azure IoT Hub as the cloud-based database solution for storing machinery data. The database schema is designed to accommodate various data types, including sensor readings, machine identifiers, timestamps, and geographical coordinates.

**MQTT Broker:** Implementing an MQTT broker to facilitate communication between the ESP32 controller and the cloud database. MQTT is chosen for its lightweight, publish-subscribe messaging protocol, which is well-suited for IoT applications.

Data Processing: Developing backend scripts and services to preprocess incoming sensor data, perform data validation, and apply business logic rules. This includes aggregating sensor readings, detecting anomalies, and generating alerts or notifications based on predefined thresholds.

**Frontend User Interface:**

**Mobile App:** Designing a mobile application using MIT App Inventor to provide users with an intuitive interface for accessing machinery data. The app allows users to view real-time sensor readings, track machine locations on a map, set up custom alerts, and generate reports.

**Dashboard Interface:** Implementing a web-based dashboard using Node-RED for monitoring and visualization purposes. The dashboard provides a comprehensive overview of machinery status, including sensor graphs, machine health indicators, and event logs. Users can customize the dashboard layout and subscribe to specific machine alerts.

**Integration and Interoperability:**

**ESP32 Firmware:** Developing firmware for the ESP32 microcontroller to collect sensor data from various sensors, such as temperature probes, pressure sensors, vibration sensors, and the Adafruit Ultimate GPS module. The firmware communicates with the MQTT broker using the MQTT protocol for seamless integration with the backend infrastructure.

## 3.4 Software Flow Diagram

3.5 Alternate Design

An alternate design could involve using different microcontrollers, communication protocols, or cloud platforms based on project requirements and constraints. For example, alternative microcontrollers such as Arduino boards or Raspberry Pi could be considered, and different cloud platforms like AWS IoT or Google Cloud IoT could be evaluated.

## 3.6 Discussion

The design choices made in this project aim to address the specific needs of tracking and monitoring machinery in various industrial settings. However, there are several considerations and potential challenges to be discussed:

**Technology Suitability:** The chosen technologies, including MIT App Inventor, Node-RED, MQTT, Azure IoT Hub, ESP32, and Adafruit Ultimate GPS module, were selected based on their compatibility, ease of integration, and functionality. While these technologies offer robust features individually, their seamless integration into a cohesive system may pose challenges during implementation.

**Scalability:** One of the key considerations is the scalability of the system to accommodate a large number of machines and sensors. As the number of monitored machines increases, the system must be capable of handling the additional data influx efficiently. This requires careful design of the cloud database and optimization of data transmission protocols to ensure scalability without compromising performance.

**Data Security and Privacy:** The project involves collecting and transmitting sensitive machinery data, which raises concerns about data security and privacy. It is imperative to implement robust encryption mechanisms to safeguard data during transmission and storage. Additionally, access control measures must be implemented to restrict unauthorized access to the system and ensure compliance with data protection regulations.

**Reliability and Fault Tolerance:** The reliability of the system is crucial for real-time monitoring and tracking of machinery conditions. To ensure reliability, fault-tolerant mechanisms should be implemented to handle communication failures, sensor malfunctions, and other unforeseen events. This may involve implementing redundancy in data transmission paths, sensor calibration routines, and error detection and correction mechanisms.

**User Interface and Experience:** The usability of the mobile app and dashboard interface plays a significant role in user adoption and satisfaction. The user interface should be intuitive, visually appealing, and responsive to provide users with easy access to relevant machinery data. User feedback and usability testing will be essential to refine the user interface and improve the overall user experience.

**Future Enhancements:** While the current project scope focuses on basic machinery tracking and monitoring capabilities, there is potential for future enhancements and expansions. This may include incorporating machine learning algorithms for predictive maintenance, integrating additional sensor types for more comprehensive monitoring, and exploring advanced data visualization techniques for deeper insights into machinery performance.

A close-up of several white squares

Description automatically generated

# Chapter 4 Project Construction and Testing

## 4.1 Construction

## 4.2 Testing Procedure

## 4.3 Software Testing

## 4.4 Fault Tree

## 4.5 Results

# Chapter 5 Safety and Ethical Considerations of the Project.

Ensuring the safety and ethical integrity of the remote machine monitoring and tracking device is paramount to its successful deployment and operation. This chapter addresses health and safety concerns related to both hardware and software aspects of the project, along with ethical considerations in the production of multiple units of the device.

# 5.1 Health and Safety Issues:

**Hardware:**

1. **Electrical Safety:** Proper insulation and grounding of electrical components are crucial to prevent electrical hazards such as shocks or short circuits. All components should adhere to relevant safety standards, such as CE or UL certifications, to ensure compliance with electrical safety regulations.
2. **Mechanical Safety:** The hardware enclosure should be designed to minimize the risk of physical injury or damage to users and machinery. Rounded edges, secure mounting, and tamper-resistant design features enhance safety during installation and operation.
3. **Environmental Safety:** Components selected for the hardware design should be rated for the intended environmental conditions, including temperature, humidity, and exposure to dust or moisture. Compliance with environmental protection directives, such as RoHS (Restriction of Hazardous Substances), ensures the use of environmentally friendly materials and processes.

**Software:**

1. **Data Privacy and Security:** Implementation of encryption protocols and secure data transmission mechanisms safeguards sensitive information collected by the monitoring device. Adherence to data privacy regulations, such as GDPR (General Data Protection Regulation), protects user privacy and prevents unauthorized access to confidential data.
2. **System Reliability:** Robust software design and rigorous testing procedures minimize the risk of system failures or malfunctions, reducing the potential for operational disruptions or safety hazards.
3. **User Training and Documentation:** Clear instructions and user manuals educate users on safe handling and operation of the monitoring device, mitigating the risk of user errors or accidents.

# 5.2 Ethical Considerations:

1. **Fair Labor Practices:** Ensuring fair wages, working conditions, and labor rights for individuals involved in the production and assembly of the monitoring device promotes ethical manufacturing practices and supports the well-being of workers.
2. **Environmental Sustainability:** Adoption of eco-friendly materials, energy-efficient components, and recyclable packaging minimizes the environmental impact of device production and disposal. Compliance with international standards, such as ISO 14001 for environmental management systems, demonstrates a commitment to sustainability.
3. **Transparency and Accountability:** Providing transparent information about the device's capabilities, limitations, and potential risks fosters trust and accountability among stakeholders, including end-users, manufacturers, and regulatory authorities.

**Adherence to Relevant Standards:**

1. **Electrical Safety Standards:** Compliance with standards such as IEC 61010 ensures the safety of electrical equipment and users.
2. **Data Privacy Regulations:** Adherence to regulations such as HIPAA (Health Insurance Portability and Accountability Act) and CCPA (California Consumer Privacy Act) protects sensitive user data and ensures legal compliance.
3. **Product Quality Standards:** Certification under ISO 9001 for quality management systems demonstrates adherence to international quality standards and commitment to product excellence.

# Chapter 6 Conclusions and Recommendations

[You should draw conclusions from the work you have done on your project over the year and any recommendations for improvements etc if work were continuing on it.]

# References

1. Espressif Systems. (n.d.). ESP32 Technical Reference Manual. Retrieved from <https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf>
2. Adafruit Industries. (n.d.). Ultimate GPS v3. Retrieved from <https://learn.adafruit.com/adafruit-ultimate-gps/downloads>
3. MIT App Inventor. (n.d.). About MIT App Inventor. Retrieved from <https://appinventor.mit.edu/explore/library>
4. Node-RED. (n.d.). About Node-RED. Retrieved from <https://nodered.org/about/>
5. MQTT. (n.d.). MQTT Version 3.1.1. Retrieved from <http://www.steves-internet-guide.com/mqtt/>
6. Microsoft Azure. (n.d.). Azure Documentation. Retrieved from <https://docs.microsoft.com/en-us/azure/>
7. LoRa Alliance. (n.d.). What is LoRaWAN? Retrieved from <https://lora-alliance.org/about-lorawan>
8. IEEE Standards Association. (n.d.). IEEE Standards for Industrial Automation. Retrieved from <https://ieeexplore.ieee.org/browse/standards/reading-room/page>
9. Occupational Safety and Health Administration (OSHA). (n.d.). Machine Guarding. Retrieved from <https://www.osha.gov/machine-guarding>
10. Espressif Systems. (n.d.). ESP32 Technical Reference Manual. Retrieved from <https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf>

# Appendix

Specification sheets for components, software code, AutoCAD drawing, Etc

**NO PADDING**

Padding is where you add in data sheets for components which are standard and do not need to be included such as a common transistor type. It can also be the inclusion of 20 pages of a data sheet when only 4 of them are relevant.

Quantity does not necessarily mean quality.

**Circuit Diagrams from CadStar**

* Include proper block diagrams and circuit diagrams. **NO SCREEN DUMPS FROM ECAD**.
* Add in the printed diagrams in your report at the end or print to a PDF file and include it in the soft copy of your report.

# Timetable

Include here the timetable you used for Project 1 and your timetable for Project 2 in tabular form (from Excel).