A picture containing text

Description automatically generated

|  |  |
| --- | --- |
| Module code and name: | Internet of Things – 6COSC014C |
| Assessment: | Portfolio |
| Student ID: | 00012670 |
| Submission deadline: | 12.01.2023 |
| Wordcount |  |
| Results date and type of feedback |  |
| Result (Pass or Fail) |  |
| **Feedback** | |
|  | |

**Introduction to case**

Research laboratories play a key role in promoting knowledge and innovation in various scientific disciplines. By conducting cutting-edge experiments and processing confidential materials, these dynamic environments are at the forefront of revolutionary discoveries. The nature of their work requires not only precision and perfection, but also the availability of a reliable security infrastructure to protect valuable assets and maintain the integrity of ongoing research.

Within the research laboratory industry, offers a variety of activities ranging from molecular biology and pharmaceutical research to materials science and beyond. Laboratories often specialize in certain fields, conducting experiments that contribute to scientific progress and technological advances.

The different nature of research activities in laboratories generates specific needs. These can include secure storage of sensitive data, controlled access to experimental facilities, and protection from potential threats to both physical and digital assets. As laboratories increasingly rely on advanced technology, there is a growing demand for comprehensive security solutions.

Certain problems arise in the daily work of research laboratories. There is a problem of avoiding unauthorized persons, concerns about the security of valuable data and the need for a quick and effective response in emergency situations. Incidents such as theft, accidental exposure, or interference with experiments highlight the practical obstacles that laboratories face. These problems highlight the need for a solution to ensure security.  
  
**Proposed solution**

**Proposed solution**: analyse identified problems and propose rationalised solution. How exactly your solution will help? How effective the implementation will be? What part of life will get improvement? What resources will be economised by proposed solution? You are bound to budget but try to be creative and resourceful. Propose perfect solution, so as if you had resources. Later you can downsize it or find compromise.

Emergency Response: Quick detection and alerting for fire and unauthorized access.

**Things used**

For alarm system project, the following components were used:

Boards:

* Arduino Uno board
* ESP8266

Sensors:

* Ultrasonic Sensor
* Motion Sensor
* Flame Sensor

Actuators:

* Active Buzzer
* LCD 1602 Module (Display)
* Red and blue LEDs

Inputs:

* Membrane Switch Module
* Button (Small)

The Arduino Uno board serves as the central processing unit for the alarm system project. It is handling inputs, controlling actuators, and managing the overall the system logic. It processes data acquired from different sensors, including ultrasonic, motion, and flame sensors.

The ultrasonic sensor measures the distance of objects within its range, providing the system with spatial awareness. Meanwhile, the motion sensor detects any movement in its vicinity, serving as a key component for intrusion detection. Both ultrasonic sensors and motion sensors play a crucial role in intrusion detection, increasing the overall functionality of the security system. Additionally, the flame sensor provides an extra layer of security by signalling the presence of fire or high temperatures.

The active buzzer serves as an audible alarm when certain events are triggered. Specifically, when motion or presence of a flame is detected, the active buzzer and corresponding red LED are activated. To deactivate buzzer and LED, someone should enter the correct password.

Password entry is facilitated by using a membrane switch. As the password is entered, the corresponding numbers are displayed on an LCD screen. In addition, the LCD display provides real-time feedback, indicating whether the entered password is correct or incorrect.

After successfully entering the correct password, the buzzer ceases its sound, and the red LED turns off, signalling the termination of the alarm state. Simultaneously, the blue LED lights up, visually indicating that the alarm system has turned off.

It's important to note that in the case of the flame detection, entering the correct password does not lead to a return to a safe state. The system will continue to issue alarms regardless of whether the person enters the correct or incorrect password. This behaviour is designed to prioritize immediate attention and action when a fire is detected, providing increased awareness and rapid response to potential fire hazards. In this case, the visual indication with the blue LED will be disabled due to the alarm status caused by the flame detection sensor.

To reset the correct password for returning system to alarm state, a person can press a small button. This action activates the buzzer and red LED once again.

**Communication**

For the Arduino Uno and the LCD display communication, it was chosen I2C.  
I2C was selected due to its simplicity and efficiency, ensuring uninterrupted connection of multiple devices on the single bus.

For the ESP8266, Wi-Fi serves as the primary means of communication with the remote server, establishing a secure HTTPS connection for secure data exchange. Additionally, Serial communication is employed to exchange data between the Arduino Uno and the ESP8266.

**Circuit prototype**

Note: In the Tinkercad simulation, a flame sensor is not available, so it was substituted with a photoresistor in the schematic. However, in the actual project implementation, a flame sensor is used.

Link to prototype: <https://www.tinkercad.com/things/aUBzHmd4OUq-stunning-lappi-turing>

**Calculation of the power, current, and voltage for the circuit.**

To calculate the power, current, and voltage for the components in circuit, it is needed to consider the specifications and connections of each component. Here's a breakdown of the calculations:  
All sensors, actuators and inputs operate at 5V.

Formula for calculation power: Power (P) = Voltage (V) \* Current (I)

Ultrasonic Sensor:

* The current is 15mA (0.015A)
* P = 5V \* 0.015A = 0.075W (or 75mW)

Motion Sensor:

* The current is 10mA (0.01A)
* P = 5V \* 0.01A = 0.05W (or 50mW)

Flame Sensor:

* The current is 20mA (0.02A)
* P = 5V \* 0.02A = 0.1W (or 100mW)

Active Buzzer:

* The current is 10mA (0.01A)
* P = 5V \* 0.01A = 0.05W (or 50mW)

LCD 1602 Module (Display):

* The current is 100mA (0.1A)
* P = 5V \* 0.1A = 0.5W (or 500mW)

Membrane Switch Module:

* The membrane switch module consumes negligible power.
* Power (P) = Voltage (V) \* Current (I)
* The current is negligible (0A)
* P = 5V \* 0A = 0W

The total power consumption:

Total Power (P) = 0.075W + 0.05W + 0.1W + 0.05W + 0.5W + 0W + 0W

= 0.775W (or 775mW)

The total current flowing through the circuit:

Total Current (I) = 0.015A + 0.01A + 0.02A + 0.01A + 0.1A + 0A + 0A

= 0.155A (or 155mA)

**Technology used**

In evaluating programming languages for alarm system project, two options were considered: Python and C++.

Python is widely recognized for its popularity in IoT development, offering extensive libraries and frameworks. It offers advantages such as ease of use, rapid prototyping capabilities, and compatibility with various platforms.

However, Arduino, using C++, was chosen as the programming language in this project. The main reason for choosing Arduino lies in my familiarity with the platform, acquired during my studies. Additionally, Arduino easy-to-use programming environment and its compatibility with a wide range of sensors and actuators. Arduino serves as a simple yet powerful platform for embedded systems, supported by well-documented libraries and a large and active community.

While considering monitoring interface options for the project, both mobile and web applications were evaluated. Ultimately, the decision was made in favour of the web app for its broader accessibility. The web-based interface ensures compatibility between different devices and operating systems.

For data storage, two potential options were considered such as Firebase and Google Sheets. Both Google Sheets and Firebase are cloud-based data storage solutions provided by Google. The decision was made to select Google Sheets because it is a familiar and user-friendly tool for storing and organizing data. Moreover, Google Sheets is completely free compared to Firebase, which incurs costs after a certain period, this also influenced decision-making.

The website for monitoring and controlling data was constructed using Google Apps Script. The primary reason for opting Google App Script in this project was its capability to effortlessly to fetch data from Google Sheets and present it on a website without requiring extensive web development knowledge. Furthermore, in Apps Script, the deployment of a web application is simplified.

**Obtained result**

**Important links**

Google Sheet link: <https://docs.google.com/spreadsheets/d/1P9u0xJFg3E7EYesyTlPyrsUoioiszEqL8neSN0WPhzk/edit#gid=0>

App Script link: <https://script.google.com/home/projects/1yh-6-f5RcCBK_NdWThKkujL5GRzlhBVpPIAKSK1FNlcipgk_UrQdb2Bx/edit>

Website link:

GitHub link:

**Reflection on results and future work**

To remotely control the system, I decided to change password through the website. However, there is a limitation in the password length – it can only be set to exactly 8 characters, not more or less. This restriction arises from the Arduino code where the password length is defined with precisely 8 characters. The code includes an if-else statement, ensuring that passwords with more or fewer characters won't be validated.

The second limitation is also associated with the password change function, specifically the inability to change the password using only integer numbers. When users input only numbers, Google Sheets stores them as integers. However, in the Arduino code, it was necessary to pass the password in the form of a string. While it’s technically possible to convert an integer to a string, it was decided to require users to enter a password with at least one letter or a character other than digits. This decision was made to enhance password reliability and security.

In the future, I would like to enhance the password change feature to use passwords with various lengths exceeding the current limit of 8-character. In addition, I would also like to improve the buzzer sound so that it sounds more like an alarm tone. Moreover, to improve the alarm system, it would be nice to implement a notification message. So that when a detection event occurs, a notification message is displayed immediately.

**Reference**