ET0731: IoT Security Project Proposal

Logo

Description automatically generated

**School of Electrical and Electronic Engineering**

Diploma in Computer Engineering

**Project Title: Free19 (Group B)**

Cars parked in a parking garage

Description automatically generated with low confidence

**Academic Year:**  2021/22, Semester 2

**Class:**  DCPE/FT/3A/24

**Submitted by (Student ID):**

Tan Kai Sheng (P1946936)

Lim Yuan Hong (P1946910)

**(Jan 2022)**

**Table of Contents**

[**1 – About** 2](#_Toc94091116)

[**2 – Objectives** 2](#_Toc94091117)

[**3 – Device Features** 2](#_Toc94091118)

[**4 – Security Features** 3](#_Toc94091119)

[**5 – System Diagram** 4](#_Toc94091120)

[**6 – Flowchart** 5](#_Toc94091121)

[**7 – Project Plan & Gantt Chart** 7](#_Toc94091122)

# **1 – About**

Singapore introduced circuit breakers and restrictions in response to the COVID-19 outbreak in attempt to prevent the disease from spreading further. To prevent the COVID-19 disease from spreading, we decided to do research into how to prevent infectious diseases from spreading. After conducting study, it was discovered that one of the best and easiest methods to prevent the disease from spreading is to ensure that everyone has a constant temperature, as this is one of the indicators that can indicate whether a person is infected with the COVID-19 virus. As a result, we're introducing the "Free19."

Free19 is a car parking system that can measure the temperature of visitors at the gantry before allowing them to enter the carpark. Furthermore, it can provide real-time temperature of the recent guests, number of vacancies in the carpark, gantry gate’s on/off state through visualizations via a dashboard. If a guest's temperature is high (>37°C), the system will notify us and provide the temperature to us via email and telegram.

Free19 includes a telegram bot that assists only authorized people in opening the gantry gate for guests when they are unable to access the carpark due to gate malfunction or other issues. Another function is an alert function that alarms the carpark manually, to alert guests that there is an intruder inside the carpark.

The information acquired about the temperature of the guests is used to keep the carpark system safe for them.

Group project’s wiki: <https://sites.google.com/view/et0731iots-free19/home>

# **2 – Objectives**

Our aims with Free19 are as follows:

* To measure the temperature of guests at the gantry before allowing them to enter the carpark for prevention of COVID-19 spreading.
* To monitor the temperature of recent guests, number of vacancies in the carpark, gantry gate’s state in real-time through Thingspeak and Thingsboard to provide further safety for guests.
* To know whether there’s a need for medical assistance for the guest via email and telegram.
* Help authorized people to open the gantry gate for guests through a telegram bot when needed.

# **3 – Device Features**

The following are the features of Free19 :

* **Automated Carpark Gantry:** The LM35 temperature sensor, which is located on the carpark gantry, measures the guest's temperature before permitting them to enter the carpark. The servo motor (gantry gate = closed) turns to 180°C if the guest's temperature is above or equals to 37°C, a red LED illuminates and buzzer rings to signify an unsuccessful admittance. If the temperature of the guest is normal (36-37°C), the servo motor (gantry gate = open) will turn to 0°C, and a green LED will illuminate to signify successful admittance.
* **Vacancies in Carpark:** The HC-SR04 ultrasonic/distance sensor, which is located at the back of the parking lot, measures the length of a carpark lot. If distance is more or equals to 4.8m (reduced to 48cm for implementation), the total amount of vacancies increases by 1. If distance is lesser than 4.8m (reduced to 48cm for implementation), the total amount of vacancies decreases by 1. The total number of vacancies in the carpark is 200.
* **Email Alert (AWS IoT & SNS):** If the LM35 temperature sensor reads more or equals to 37°C, we'll get an email by using AWS IoT and SNS to alert us that the guest has high fever and may need medical help.
* **Dashboard (Thingspeak/Thingsboard):** The dashboard includes visualizations such as the number of vacancies in the carpark to give guests vacancy information, a line graph of guests’ temperature and gantry gate’s state to monitor and help keep the carpark safe from intruders. The current temperature of the latest guest will be displayed, to help indicate whether the guest needs medical assistance.
* **Telegram Bot:** When guests are unable to access the carpark due to gate malfunction or other issues, only authorized people can open the gantry gate for them through the telegram bot. Another feature is an alert system that manually alarms the carpark to warn guests that there is an intruder inside the carpark. Another function is that it helps to alert us whether there’s a need for medical assistance for the guest.

# **4 – Security Features**

* **TLS Security:** AWS IoT message broker uses TLS version 1.2 to encrypt all communication. TLS is used to ensure the confidentiality of the application protocols. For MQTT, TLS encrypts the connection between the device and the broker. TLS client authentication is also used to identify devices.
* **AWS IoT Device Security:** Each device identifies itself and authenticates to the cloud using a X.509 certificate. You use policies to control what devices have access to.
* **API Security:** When sending data from the LM35 temperature sensor to Thingspeak, a writeAPI key is needed to prevent unauthorized devices from sending altered data.
* **Use strong and unique passwords for WIFI network:** Using common words or passwords that are easy to guess, such as ‘password’ or ‘123456’ should be avoided. Instead, we will consider using complex and unique password which is made up of capital letters, small letters, numbers and symbols.
* **Use of HTTPS:** HTTPS is used to securely send data to thingspeak instead of HTTP.
* **Encrypted emails:** Emails sent by AWS IoT to gmail is encrypted with the standard encryption, which is TLS.
* **Telegram Bot:** Only authorized users can use the bot and custom commands are created such that only authorized users know how to control the system.

# **5 – System Diagram**

Diagram

Description automatically generated

The LEDs, servo motor and buzzer will be connected as outputs to the NodeMCU, while the temperature sensor (LM35) and ultrasonic sensor (HC-SR04) will be attached as inputs. The temperature, gantry gate’s state and vacancies will then be sent to Thingsboard, ThingSpeak through HTTPS and AWS IoT via MQTT via the NodeMCU.

The condition of the guest will be determined using the temperature data given to AWS IoT, and if it is above 37°C, an email will be sent to alert us. The data supplied to ThingSpeak and Thingsboard will be sent to the dashboard for easy event monitoring.

The dashboard includes visualizations such as the number of vacancies in the carpark to give guests vacancy information, a line graph of guests’ temperature and gantry gate’s state to monitor and help keep the carpark safe from intruders. The current temperature of the latest guest will be displayed, to help indicate whether the guest needs medical assistance.

The LM35 temperature sensor, Red LED, Servo Motor will be connected to the telegram bot through the internet, to determine gantry gate status and controlling it, create alert function to alarm guests that an intruder is inside and whether there’s a need for medical assistance for the guest.

# **6 – Flowchart**

(ESP8266, LM35, LEDs, Servo Motor, Buzzer) – > Thingspeak & Thingsboard & AWS

Diagram

Description automatically generated

(ESP8266, HC-SR04) – > Thingspeak & Thingsboard

A picture containing text, sky, indoor, map

Description automatically generated

(ESP8266, Red LED, Buzzer, LM35 Temperature Sensor) – > Telegram Bot

A picture containing text, sky, map

Description automatically generated

# **7 – Project Plan & Gantt Chart**

Timeline

Description automatically generated

The Contributions is shared between the both of us for all tasks.

Week 11 & 12: **Ideation & Discussion**

This week we generated ideas and set on a single idea to use.

Week 13: **Submit Proposal**

This week we will we submit the proposal to the lecturer-in-charge.

Week 13 - 15: **Plan & Assemble Circuit**

Plan out the circuit and assemble the whole circuit with coding.

Week 15: **Create Dashboard & Secure Device**

Create Dashboard with Thingspeak and Thingsboard and secure the device with desired security features.

Week 16: **Secure Device, Testing & Debugging**

Secure device, do multiple testing and debug errors found.

Week 17: **Assembling Prototype**

Assemble everything together to prepare for demonstration next week.

Week 18: **Demonstration to Lecturer**