Name: Dang Khoa Le
Student ID: 103844421
Major: Software Engineering

Individual Assessment

Project Aim: Distinction (D)

Background:

Your consulting company has been hired to develop a new Engineering Solution for a Client. This solution must contain the ability to function within an online, Internet of Things Environment. Your manager has assigned you the task of developing the prototype solution for the Internet component of the Project.

The Client has advised that they have already deployed a IoT infrastructure for existing projects, and would like this solution to leverage this installation. Your design must use the existing IoT infrastructure to broker communications between components

IoT Infrastructure

The design is running the MQTT (Mosquitto) Message Broker on: rule28.i4t.swin.edu.au. This design utilises a pre-made infrastructure, which uses an account that has been created (for this unit). Both of the username and password for this account is my student ID (103844421).

Design Title: IoT-based Elderly Health Monitoring System

Introduction:

The Internet of Things (IoT) has revolutionized the way we collect, analyse, and utilise data in various aspects of our lives. One such application is the IoT-based Elderly Health Monitoring System (IEHMS), a comprehensive solution designed to monitor and ensure the health, safety, and overall wellbeing of elderly individuals living independently or in care

Figure. IoT

facilities. This system comprises a set of sensor devices, a user interface, and a secure MQTT (Message Queuing Telemetry Transport) broker that facilitates communication between devices and users.

Key Components:

1. Subscription:

All subscription (topics) is direct via port 1883 (mqtt) with the broker rule rule28.i4t.swin.edu.au (IP: 136.186.230.28) provided by the unit resources and maintain proper subscription, allowing receiving messages, also granting private accessibility for authorized users at specific topics. A username and password is prerequisite to allow credential accessibility, using student ID (103844421).

    a. Private topics:

The idea of distinguishing private and public topics in this design architecture is mainly due to the presence of the "wake_up_time" and "heart_rate" topics, which are confidential health data of the elderly user, and should be kept privately from strangers and/or attacker from exploiting it. In private topics, the base-topic level will be the site or the username (103844421), which have different sub-topics level 1 and level 2 associate to it. While '#'

syntax wildcard executes all sensor data from all of the sub-topic levels to the base-topic, '+' syntax executes all of the sub-topic level 2 corresponded to any sub-topic level 1. In order to access any sensor data values, clients and users will have to subscribe to the base-topic, which is the credential username, while also have to be appropriately match the credential username, password and IP address location, this implementation enhance the cybersecurity and privacy of user data from attackers.

b. Public topics:

The base-topic level will be the site named as "public", which have different sub-topics level 1 and level 2 associate to it. By similar to the public topics subscription, the '#' syntax wildcard executes all sensor data from all of the sub-topic levels to the base-topic, while '+' syntax executes all of the sub-topic level 2 corresponded to any sub-topic level 1. In order to access any sensor data values, all users will have to subscribe to this "public" base-topic, this public topics briefly allows anyone to access the current environmental data of the surrounding to the elderly living place, and not being restricted by the credential username (103844421).

2. Sensor Devices (Device 1, Device 2, Device 3)

These sensor devices are able to generate and publish either public and private topics messages simultaneously.

◊ Device 1: Monitors indoor and outdoor temperatures, providing essential information to ensure the elderly's comfort and safety.

Terminal console command to use Device 1 (python file): python /path/to/Device1.py

◊ Device 2: Measures humidity levels, aiding in controlling indoor climate conditions for health and wellbeing.

Terminal console command to use Device 2 (python file): python /path/to/Device2.py

◊ Device 3: Monitors air quality (ASI - Air Quality Index) and pollution rates, crucial for respiratory health and a cleaner living environment.

Terminal console command to use Device 3 (python file): python /path/to/Device3.py

3. User Broker (User.py):

This broker allows elderly individuals to access real-time data from the sensor devices at the care facilities. The User interface also enable the elderly to keep track on their wake-up time, aiding them in setting a regular schedule; besides, also keep track on the elderly user's heart-rate (bpm) simultaneously, hence publishing these as private topic messages that only Client can access it. It is crucial for caregivers and the elderly users themselves to keep an eye on their wellbeing, at least daily.
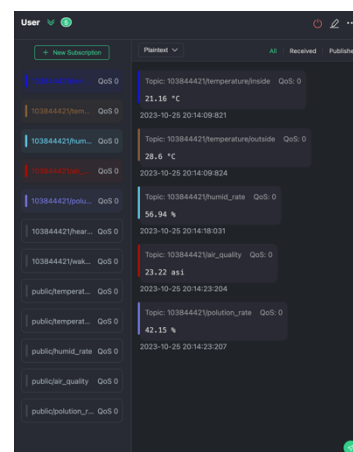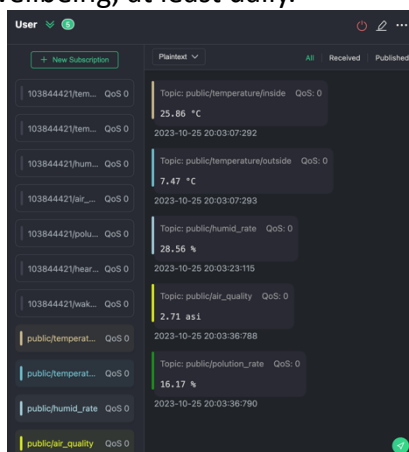
Terminal console command to check data on User broker:
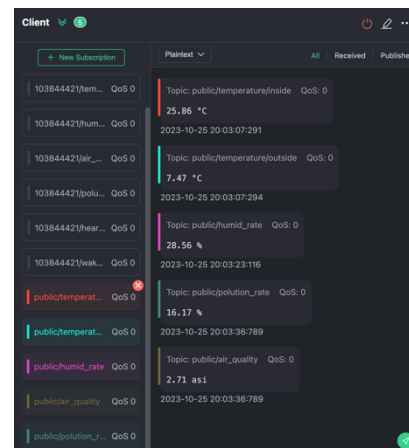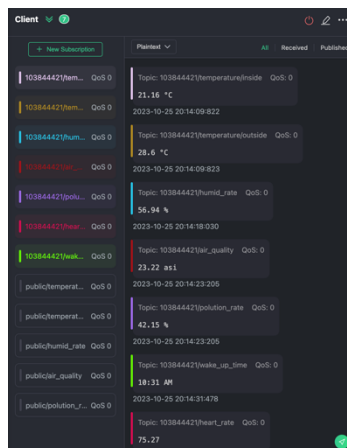python /path/to/User.py 103844421/<topic>
or public topics via:
python /path/to/User.py public/<topic>
Terminal console command to input time data on User broker: python /path/to/User.py

4. Client Broker (Client.py):

This broker allows caregivers, such as nurses or family members, to access and monitor real-time data from the sensor devices and ethe elderly user's data. The Client broker also provides the caregivers with insights into the elderly's heart health and daily routine (particularly their wake-up time), aiding them in setting a regular schedule by the privilege to access the two private topics.

Terminal console command for Client broker (python file):
python /path/to/Client.py 103844421/<topic>
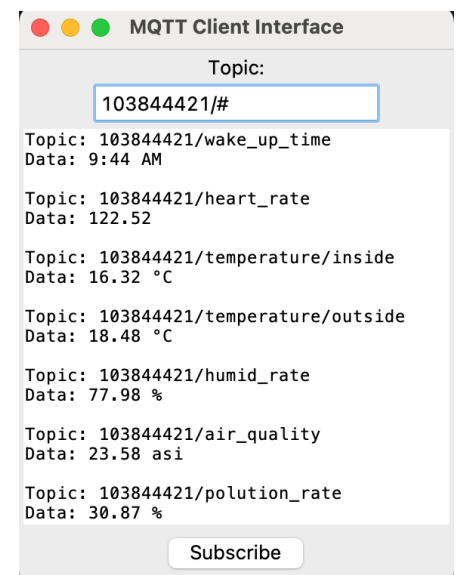or public topics via:
python /path/to/Client.py public/<topic>

5. Virtual Client Broker (Client_Interface.py):

By similar to the Client broker, this Client Interface broker will replace the Graphical MQTT Client with a virtual python-based interface that allows users to input their interested data-type (topics, either private or public), then output the relevant output data from that topic. This virtual broker is designated for both the elderly and the caregivers and act as a substitutional option while they cannot interact with the User and Client broker, allowing them to monitor and observe data in another way.

Terminal command virtual Client broker (python file): python /path/to/Client_Interface.py

Python terminal console script examples (from VS Code):
* Private topics examples:

* Public topics examples:

6. Cybersecurity Measurement:

The system ensures that data access is restricted to authorized users, preventing unauthorized external access to the private network for the elderly's safety and privacy. Cybersecurity component plays a pivotal role in this design as attackers can manipulates personal data/information from user/client to use as a threat against them.

Figure. Cybersecurity

7. Data generator

This design currently using random numbered data generator to provide a fake data output (as there are no actual sensor being connected to the system). However, the design is eligible and capable of integrating with a real sensor and can be able to provide the same interaction experiences to users and clients.

Device1: inside temperature should be varied from 15 to 30 °C when outside temperature is more severe, ranging from -10 to 40 °C.
Device2: outside humidity rate should be varied from 20 to 80%, commonly found in most country in sub-tropical area.
Device3: outside air quality should be varied from 1 to 30 asi when the pollution rate is more severe, ranging from 10 to 50 %, mostly found in many countries including Australia.
User: The elderly wake up time commonly founded to be among 4.00 to 10.59 AM while their heart rate is found to be at healthy (or fair) condition within 40 (low) to 140 (high) bpm (beats per minute).

The random data generator is executed by a python code such as (from Device2.py):

```
fake_humidity = round(random.uniform(20, 80), 2)
```

8. Usage and Benefits:
    a. For Elderly Individuals:
◊ Elderly individuals have access to an easy-to-use interface that provides them with essential information about their living environment.
◊ They can check the indoor and outdoor temperatures, humidity levels, and air quality to ensure they are comfortable and safe.
◊ They can keep track on their heart beat at any time, which allows them to understand their physical wellbeing and adjust their diet and exercising routine properly.
◊ The system helps them maintain a consistent wake-up routine, promoting healthy living habits, which support the individual positively towards their physical and mental wellbeing.

    b. For Caregivers (nurses and/or family members):
◊ Caregivers can remotely monitor the health and wellbeing of the elderly, making it easier to provide prompt care or assistance.
◊ Access to temperature, humidity, and air quality data aids in creating an optimal living environment.
◊ Access to their clients' heart-rate to keep track on their client's physical health everywhere.

Figure. Health monitoring
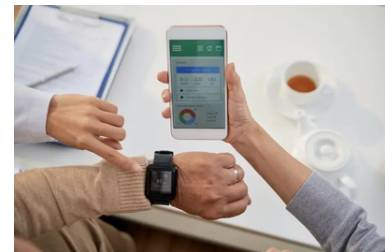
◊ Caregivers can track the elderly's daily schedule and intervene if necessary.

    c. For other unauthorised users:
Friends, family or relatives of the elderly user, who hasn't pre-subscribed to grant access (using the private topic 103844421) before, can also access the environmental condition of the place that the elderly lives, therefore, get to know them better. (Scenario: A French friend of my grandparent (Australian) couldn't know anything about the situation at my grandfather's place, he want to send a gift, but not sure if it is cold or hot there, this design allows him to know the place is very cold, so he send him hot tea instead).

9. Overall Benefits:
The IEHMS offers several advantages:
a. Improved Health: Ensures the elderly live in comfortable and healthy conditions. By understanding health factors, the elderly user can adapt with weather changing and assure better self-care and medical intervene.

Figure. The health and wellbeing

b. Remote Monitoring: Allows caregivers to keep an eye on their client wellbeing from distance, which also empower an individual caregiver to take care of more than one client simultaneously.
c. Data-driven Decisions: Data collected from the sensor devices enable evidence-based caring and medical decisions.
d. Enhanced Independence: Enables the elderly to live independently with added safety, privacy and security, by monitoring the presence of the internal and external impact.

10. Usage Instructions:
    a.   Elderly User Access (User.py and User broker):

To access the system, the elderly user can simply run the "User.py" script on their device. They will be able to check public topics such as temperature, humidity, and air quality information, along with their daily wake-up time and heart-rate to be updated at least daily.

    b.   Caregiver Access (Client.py and Client broker):

Caregivers can use the "Client.py" script with different sub-topics' data that have been pre-subscribed ("temperature/inside," "temperature/outside," "humid_rate", "heart_rate", etc) to access either public or private data.

The client script provides caregivers with real-time monitoring and control over the elderly's living conditions.

11. Conclusion:

The IoT-based Elderly Wellbeing Monitoring System (IEHMS) offers a comprehensive solution for ensuring the health and wellbeing of elderly individuals. It combines data from multiple sensor devices and provides easy access to users, including the elderly themselves and their caregivers. By monitoring crucial metrics and offering insights into daily routines, the system is a valuable tool for supporting independent living for the elderly while ensuring their safety and wellbeing.

Overall Criteria:

MQTT is a lightweight and efficient messaging protocol designed for scenarios where low bandwidth, high latency, or unreliable networks are common. It's based on the publish-subscribe pattern and is widely used on the Internet of Things (IoT) for machine-to-machine (M2M) communication. MQTT works by allowing devices or clients to publish messages to a broker and subscribe to receive specific messages from the broker. This makes it an excellent choice for real-time data transmission, especially in remote and resource-constrained environments.

Subscribing to MQTT topics is a fundamental part of how clients, whether they are devices or users, access the data. The structure of MQTT allows a hierarchy/ranking construction where higher-level topics can encompass lower-level topics. In this system (IEHMS), the following examples illustrate how topic-subscription works:

◊   Subscribing to the base-topic: For accessing general data associated with the account, a client/user subscribes to the username (for private subscription) and "public" (for public subscription). For example, to access all private data with username "103844421," the client subscribes to "103844421/#", while they can access "public/#" for all public data.

◊   Subscribing to subtopics: To access specific types of data, clients can subscribe to lower-level topics. For instance, to access indoor temperature data, the client/user subscribes to "103844421/temperature/inside" or "public/temperature/inside".

◊   Wildcard Subscription: The "#" and "+" wildcards enable users to subscribe to multiple topics at once. The "#" wildcard allows subscribing to all subtopics, while the "+" wildcard subscribes to all subtopics under a specific level. For example,

"103844421/temperature/+" would subscribe to both "inside" and "outside" temperature data, and "103844421/ #" subscribing to all topics of the base.

This design has excellently implemented a mqtt infrastructure (using paho-mosquitto, tk and random) that allows users and clients to access the elderly health status and external environment (via port 1883). The design concept strived to provide a subscription interface that empower their users to monitor and keep track on sensor data value, via subscription to the private/public base-topic, therefore enable confidential accessibility that restrict by a firewall and credential username and password.

Notice: As this project design strive to enhance the security power by encrypting a unique username and password, and redirecting a physical restriction to a private network (IP address), all users and clients, when using the IEHMS system have to use a VPN application to access with a valid authentication. The IP address is 136.186.230.28 (from the broker-rule) and the system can't be used elsewhere, which subscription cannot work (aren't allowed).

Pass Level Criteria:

◊   Two client devices (Client.py and User.py) have been implemented that generate data and post messages to a private topic on the MQTT server.

◊   Two client devices (Client.py and User.py) have been implemented that subscribes to an appropriate topic and print received messages to the screen.

◊   All clients (Device1.py, Device2.py, Device3.py, User.py, Client.py) subscribe to the public channel and print all public messages along with their sub-topic information.

◊   4 public topics and 6 private topics have been assigned, while "wake_up_time" and "heart_rate" can only be access via the private topic channel.

◊   I also have provided source code for the applications (all Python files) and instructions on how to run them.

Credit Level Criteria:

◊   More than one of the device applications (Device1.py, Device2.py, Device3.py, User.py) generates messages and subscribes to a topic.

◊   All device applications (Device1.py, Device2.py, Device3.py, User.py) subscribe to the public topic (103844421).

◊   The MQTT broker (User.py and Client.py) runs within the organization and blocks off-site access. A Cybersecurity report detailing the security issues is needed for this part.

Distinction Level Criteria:

◊   At least two devices must subscribe to different topics, responding to different requests or posts. Device1.py publishes temperature data, Device2.py publishes humidity data,

Device3.py publishes air quality and pollution rate data; these are different topics (sub-topics level 1 and 2).

◊ At least one device must generate messages to more than one topic. Device1.py generates data for "temperature/inside" and "temperature/outside" (differential by sub-topics), Device3.py publishes air quality and pollution rate data (this is 2 different base-topics), and User.py publishes wake-up time and heart-rate data (this is 2 different base-topics).

◊ Replacing the Graphical MQTT Client (Client broker) with a Python Client (Client_Interface) to act as a simple user-interface to monitor the system and generate or post messages. A Python-based user interface that can interact with the MQTT system has been implemented. This is a substantial extension to the Pass and Credit level, using additional code and libraries (installed 'tkinter' to create the user interface with paho-mqtt for MQTT communication). This interface will create a user-interaction window that seek for user's interested data-type input (subscribes to a topic) and therefore, output the corresponding output data.

Application:

From this assessment architecture, I want to provide a cyber-restricted design that support the elderlies wellbeing and maintain their healthy lifestyle routine, while also allows nurseries and caregivers to reduce their workloads, enable them to take care of more patients and clients simultaneously.