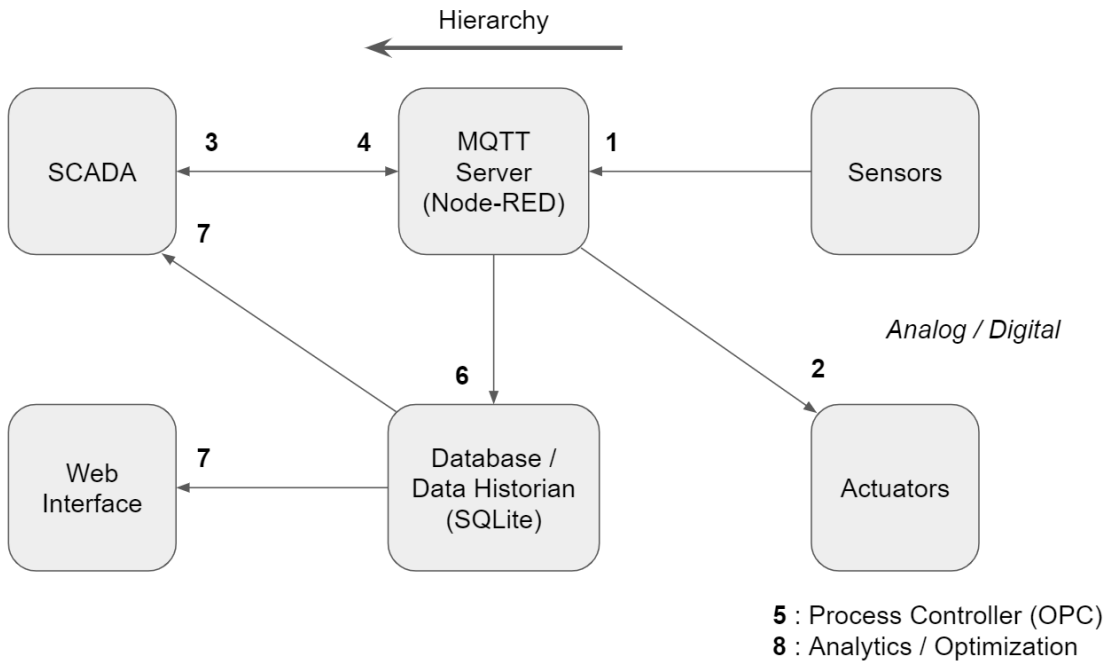
CO326 Project Report

Group E : Occupation and Access Control

Group Members

Team Leader	Bandara S M P C	E/17/027
Task 1	Rathnayake R L D A S	E/17/284
Task 2	Morais K W G A N D	E/17/212
Task 3	Shalha A M F	E/17/327
Task 4	Rishard M I	E/17/005
Task 5	Weerasinghe S P D D S	E/17/379
Task 6	Perera K S D	E/17/246
Task 7	Gunathilaka R M S M	E/17/100
Task 8	Nawarathna K G I S	E/17/219

Graphical representation of each task



Content

Introduction

Occupancy and Access control is an essential part of modern commercial buildings and premises. No matter which type of commercial building you work (hospitals, hotels, institutes, etc.) human interactions and security have to be managed properly. In this project we have proposed a way to automate this process.

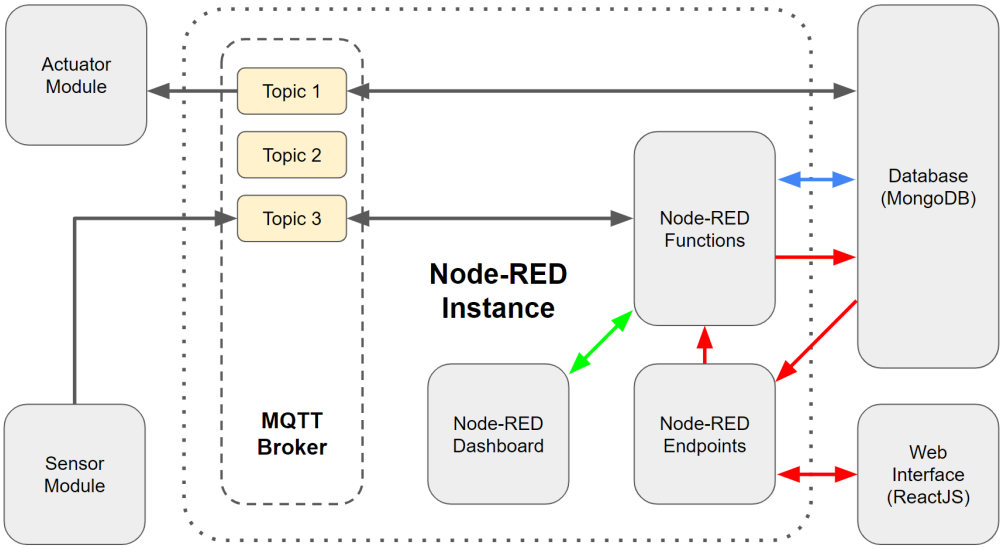
These kinds of buildings are usually very crowded and busy. These cause an inefficient working/learning environment for the people. The current pandemic situation puts more pressure on this matter and people are encouraged to reduce human interactions.

When it comes to Access control, we have to consider both authentication and authorization of people. And authentication doesn't mean exploiting the privacy of the people, so we have to keep track of the people while protecting their identity. Authorisation is a main security aspect, when it comes to the safety of the company property and the people.

Main Features of the System

1. Count the number of people in each room.
2. Number of people who pass each corridor.
3. Authentication and Authorization of people for rooms.
4. Identify the people who entered the room.
5. Daily reports on attendance.
6. Provide Occupancy details for HVAC and Safety groups.

High Level Architecture

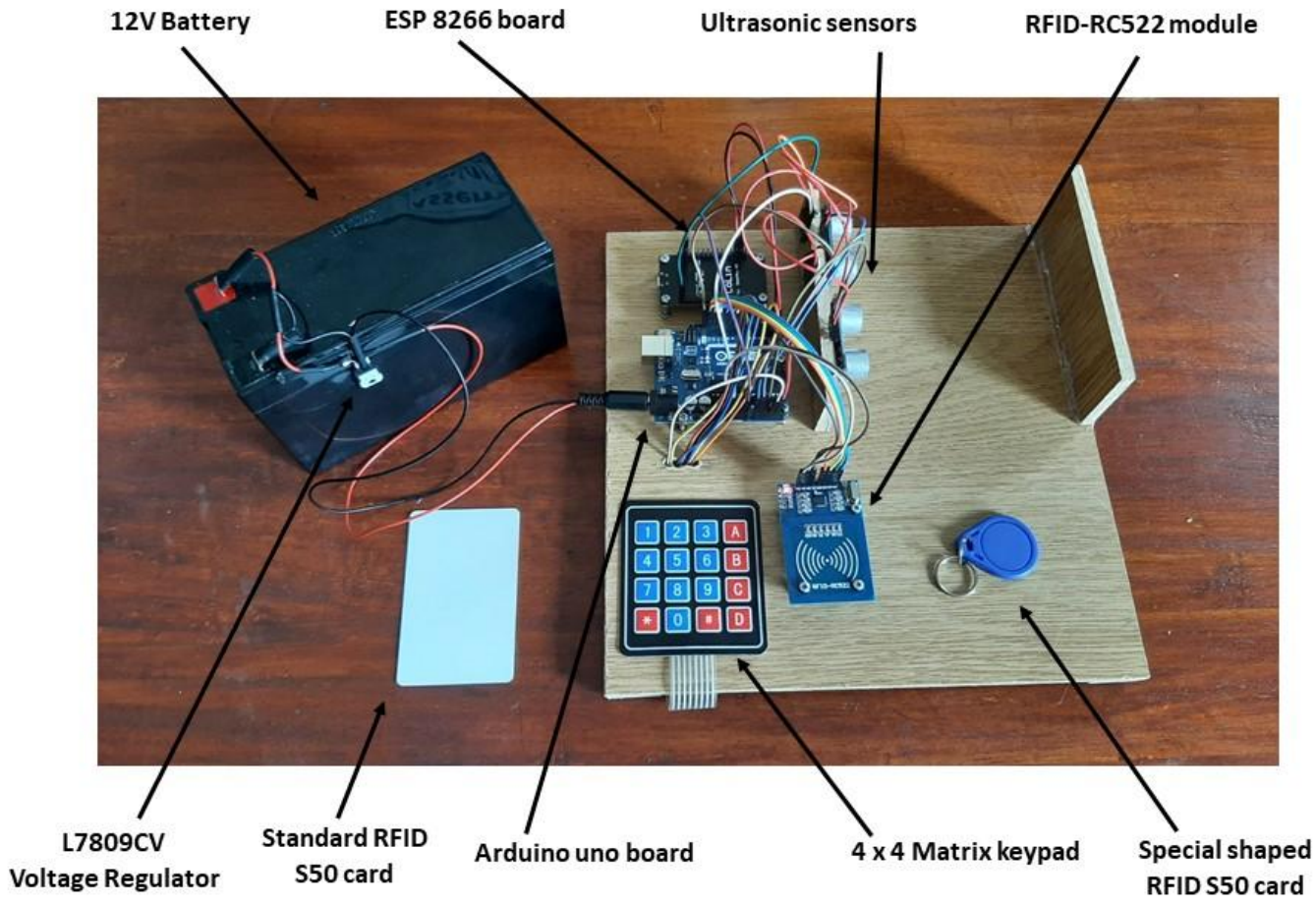


Sensors to read data and send to MQTT Server

Introduction

In the Occupancy and Access control system mainly three sensors have used to sense the data. They are Ultrasonic sensor, RFID sensor and the 4x4 matrix keypad.
In the Occupancy control part, to count the number of people who crossed the door and to detect the direction, two Ultrasonic sensors were used.
In the Access control part, for Authorization and Authentication, the RFID sensor and Keypad were used to get the data.

Hardware Implementation



Core hardware options

1. Arduino uno board(input voltage - 7 to 12v) with ESP8266 board (input voltage - 4.5 to 10v)
   Pros:
   · Can supply 5v input to the Ultrasonic sensors
   · Arduino uno board is capable of input voltages between 7-12v
   . Can tolerate high voltages than esp8266 and esp32
   Cons:
   · Should use external module for the Wi-Fi connection

2. Arduino uno board(input voltage - 7 to 12v) with ESP8266 Wi-Fi module
   Pros:
   · Can supply 5v input to the Ultrasonic sensors
   Cons:
   · Need USB-UART to program the Wi-Fi module

3. Esp32 board (input voltage - 2.2 to 3.6v)
   Pros:
   · Has the Wi-Fi facility
   Cons:
   · Should use external power supply for the Ultrasonic sensors

Therefore, all three options have pros and cons.

By considering the above **pros and cons**, and the **availability of the components** we choose the **1st option** for the hardware implementation.

Ultrasonic sensors

- Ultrasonic sensor 1 and Ultrasonic sensor 2 always measure the distance (The default distance in the prototype is 11.5 cm) one after another. It takes 26ms for one round. After each round it calculates the difference of the distances measured by two sensors. (difference = dIst_sensor2 – dist_sensor1)
- The values for the difference smaller than 1 cm are ignored as the tolerance interval.
- If the difference is greater than 1; it means that a person has crossed the Ultrasonic sensor 1 first. That means the person has entered the room. Then it will publish 1 to the relevant topic.
- If the difference is smaller than -1; it means that a person has crossed the Ultrasonic sensor 2 first. That means the person has left the room. Then it will publish -1 to the relevant topic.
- After detecting a difference value greater than 1 or smaller than -1, it will take a delay of two seconds to avoid crossing the other sensor by the same person. (Average value to cross the two sensors by a person is 2S)

RFID-RC522 module

- This is used for the authentication process.
- Before a person enters a room, it checks if the person has access to the relevant room. There is a list of RFIDs which have the access to the relevant room, stored in the database.
- To get access to a room, the person's RFID should have access to that room.
- When a person scans an RFID card, It reads by the RFID-RC522 module and then it will be published to the relevant

4 x 4 matrix keypad

- This is used for the Authorization process.
- Before a person enters a room, he or she should enter the password relevant to the room to get the access.
- To enter a password first '*' should be pressed. Then can enter the password and finally should press '#' to submit.
- Then the entered string will be published to the relevant topic using MQTT.

ESP 8266 board

- This is used for the Wi-Fi connection. The total data processing is done by the Arduino uno board. After processing the sensor data, the final result is sent to the ESP 8266 board using serial communication.
- When a result comes from the Arduino, in the ESP 8266, first it checks from which sensor that data came from.
- There are three distinct topics for each room to publish the sensor data.

  They are,

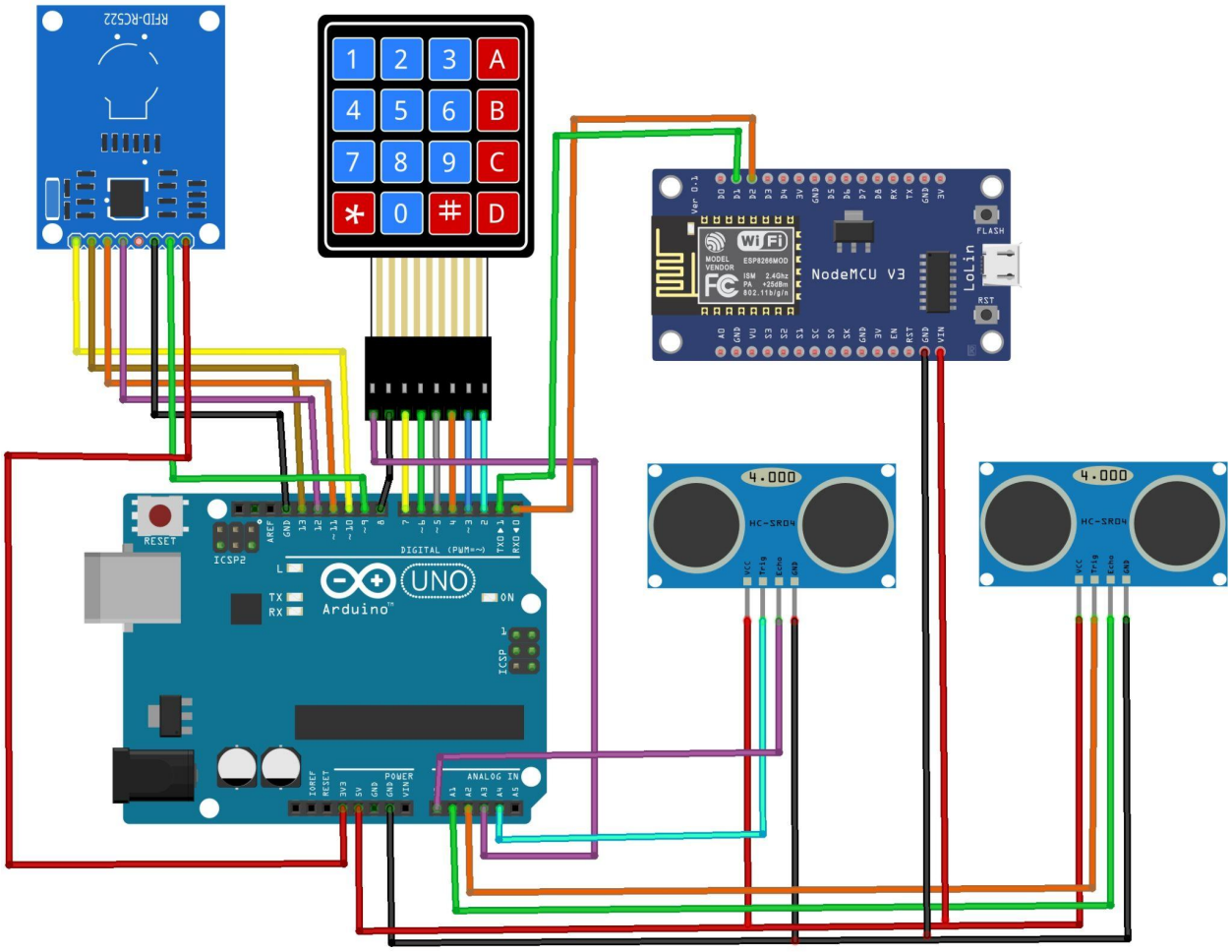
  326project/smartbuilding/occupancy/<floorno>/<roomno>/ultrasonic
  326project/smartbuilding/occupancy/<floorno>/<roomno>/rfid
  326project/smartbuilding/occupancy/<floorno>/<roomno>/keypad

  Then it selects the relevant topic and publishes the data to that topic using MQTT.

12V Battery and L7809CV Voltage Regulator

- This is used to power the System. The power is supplied from the battery to the Arduino through the barrel port. The Arduino is capable of input voltage between 7v to 12v. But the best input voltage is 9v. Therefore, L7809CV Voltage regulator is used to reduce the 12v to 9v. Then the 9v is supplied to the Arduino.
- The Ultrasonic sensors, RFID-RC522 and ESP8266 board are powered by the 5v and 3.3v pins.

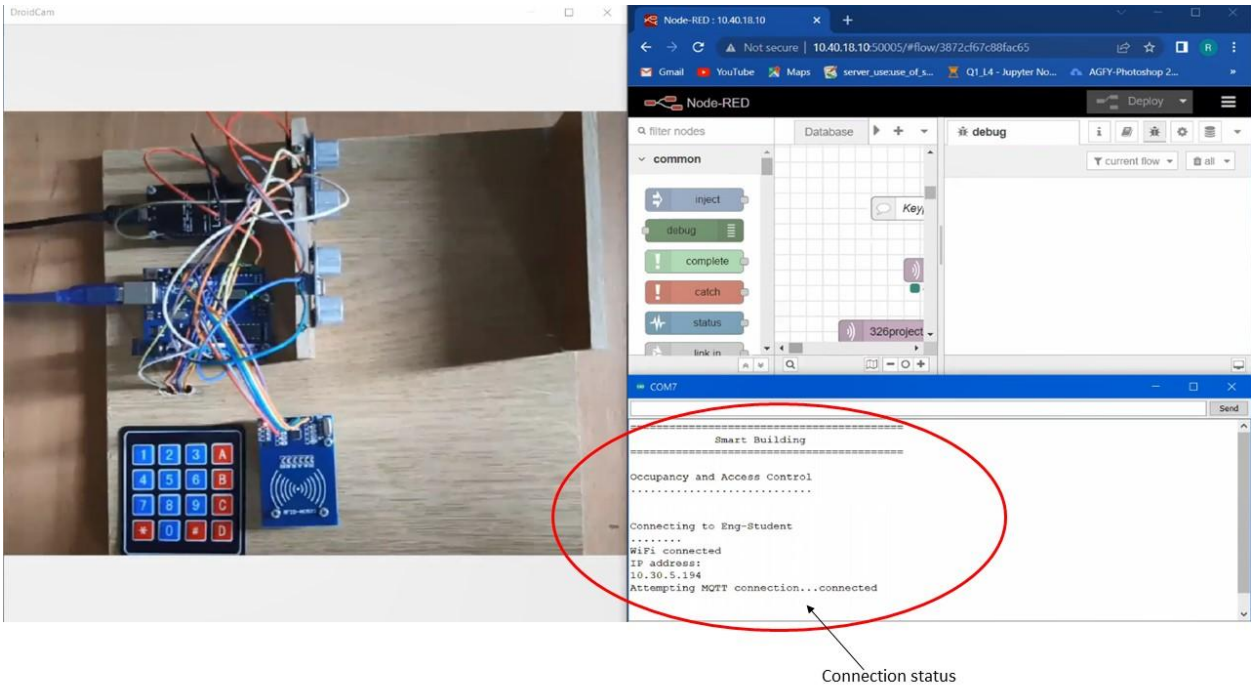
Circuit Diagram



Test and Results

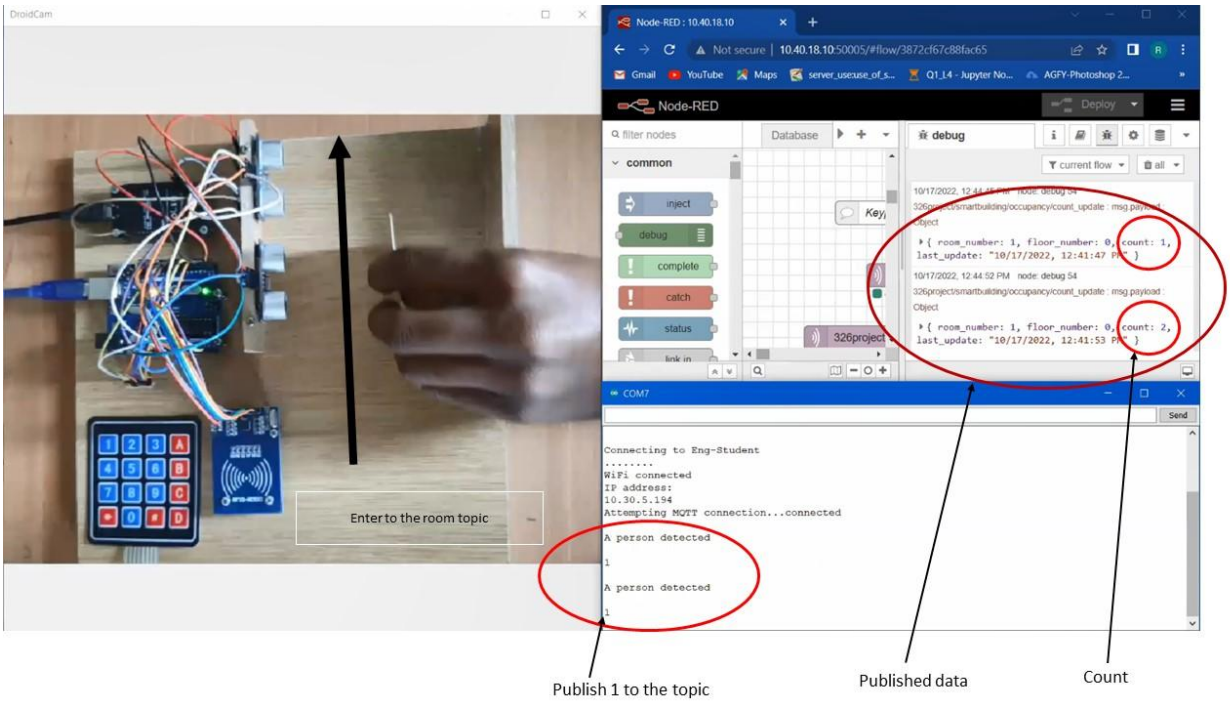
Connecting to the Wi-Fi

- When connected to the Wi-Fi it will show the message in the serial monitor.

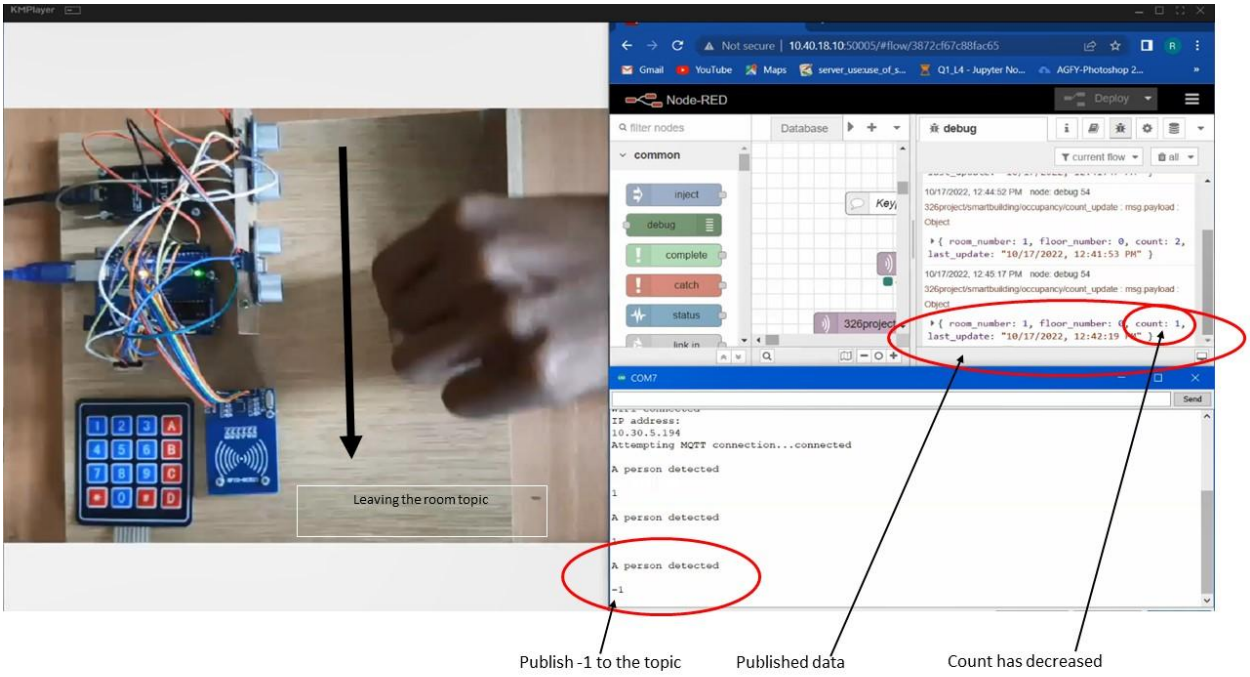


Testing the Ultrasonic sensors

- When a person enters a room it will be published to the relevant topic, and the count will be increased.

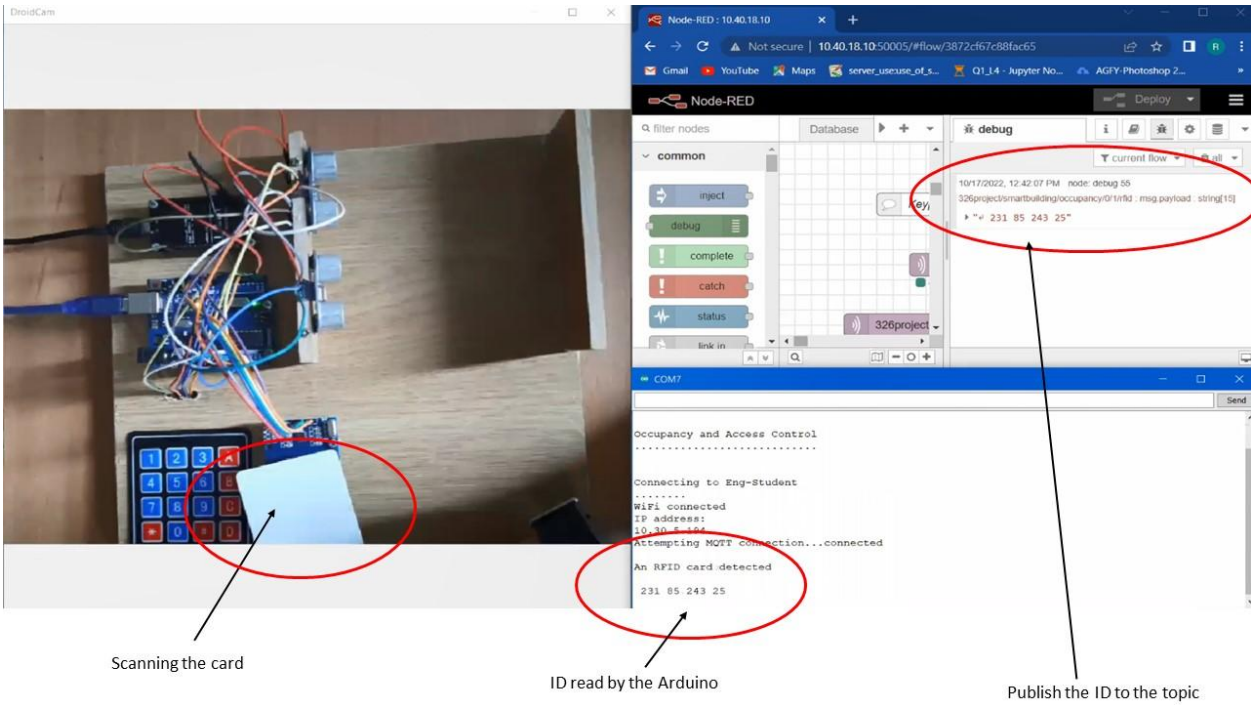


- When a person leaves a room it will be published to the relevant topic, and the count will be decreased.

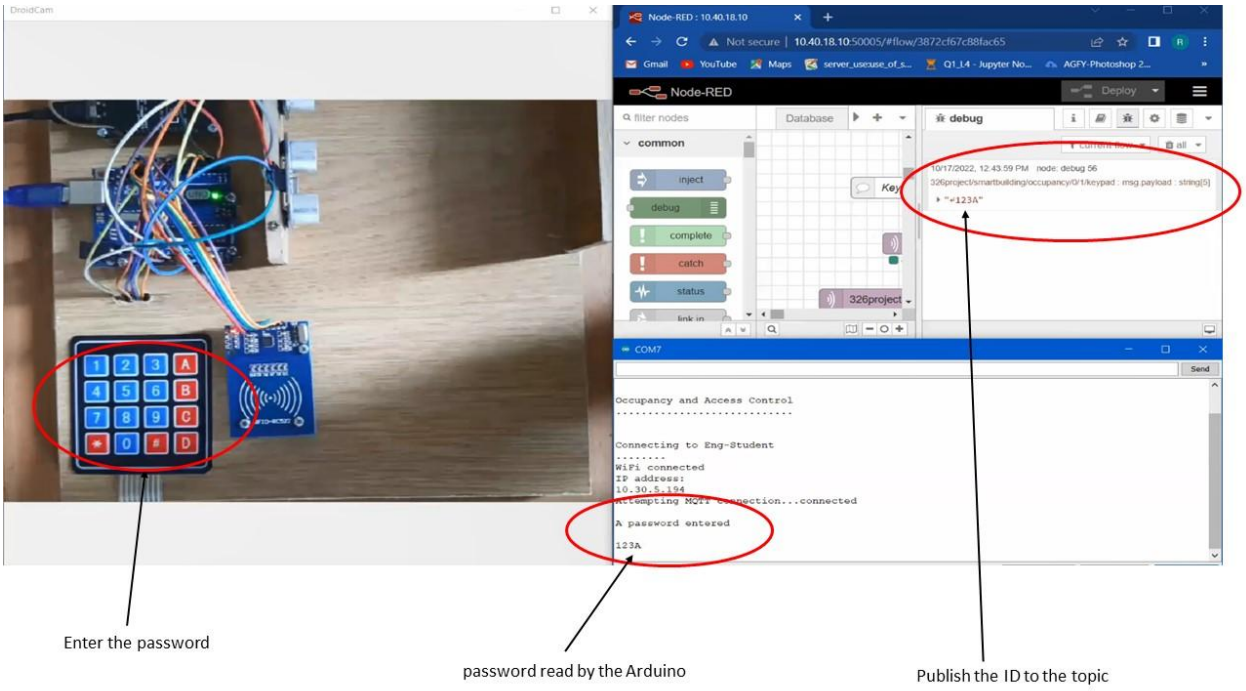


Testing RFID Sensor

- When a person scans an RFID card from the RFID sensor, It is read by the Arduino and Published to the relevant topic.



- When a person enters a password, It is read by the Arduino and Published to the relevant topic.



Discussion

When considering the whole Sensor system, using this it is possible to achieve many useful requirements, such as,

- Count the number of people in a room
- Number of people who pass the corridor
- Identify the people who entered to the room
- Authentication and Authorization of people
- Record the Number of people who pass each corridor.
- Daily reports on attendance.
- Provide Occupancy details for HVAC and Safety groups.

And also, this does not consume much power. Therefore, this system is very efficient and very useful for many places, where the above mentioned things are required.

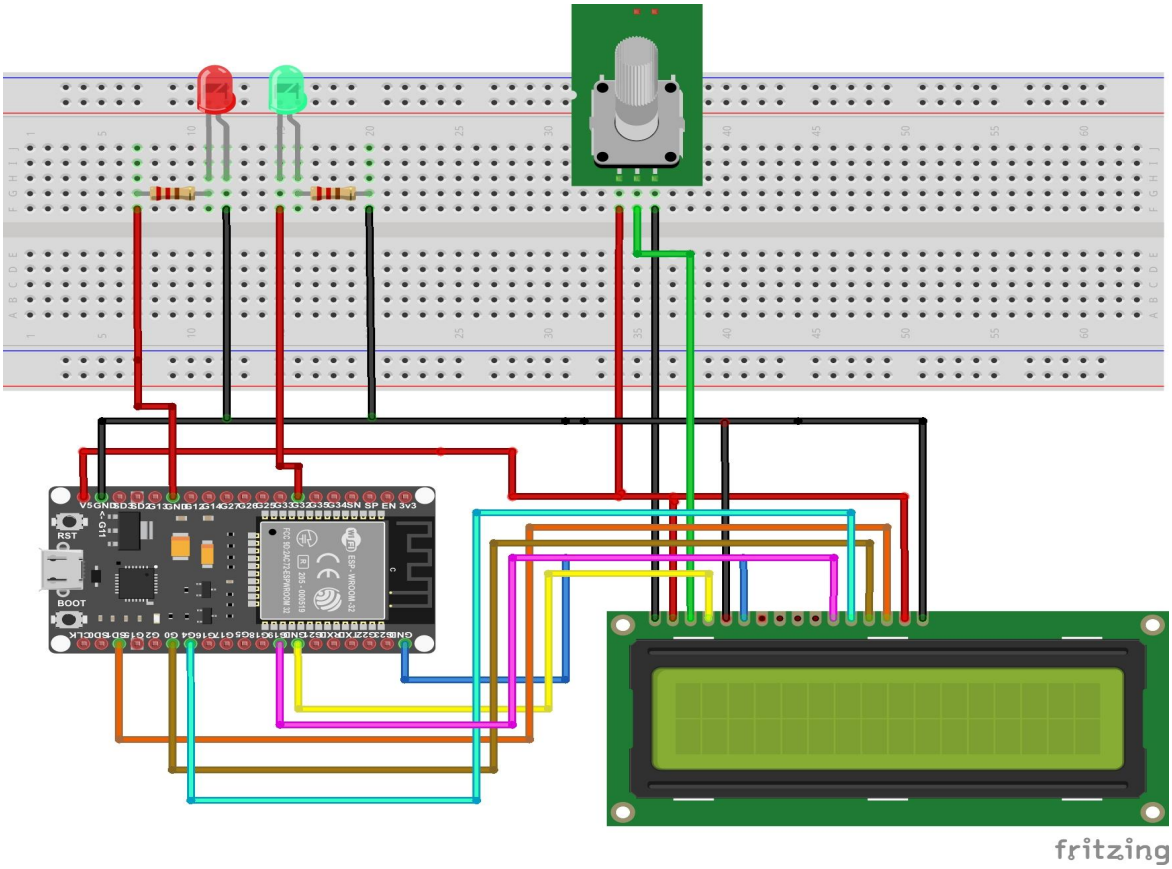
Things that could not implement

- In this system when entering a room or leaving from a room, people should go one after the other. Because if two or more people cross the sensors at the same time, the sensors will take it as one person. This issue should be developed by future developments.
- When considering the keypad, there can be used only 0 to 9 integers, ‘A’,’B’,’C’,’D’,’#’ and ‘*’, for passwords. In future implementation it should be added more characters and symbols into the keypad.

Read data from MQTT Server and take decisions and control actuators

- Occupancy and access control is done according to the information given by processed sensor details.
- After sending sensor details to the main controller(ESP32) for data processing, the main controller decides what to do with the actuators.
- When someone enters the room, he has to authenticate using an RFID card reader or keypad. After authentication is done, if he is a valid person, Then the electromagnetic door lock should be opened and the LED display shows a welcome message and relevant information about the user. If he is not a valid person, then the error message will be displayed on the LED panel.

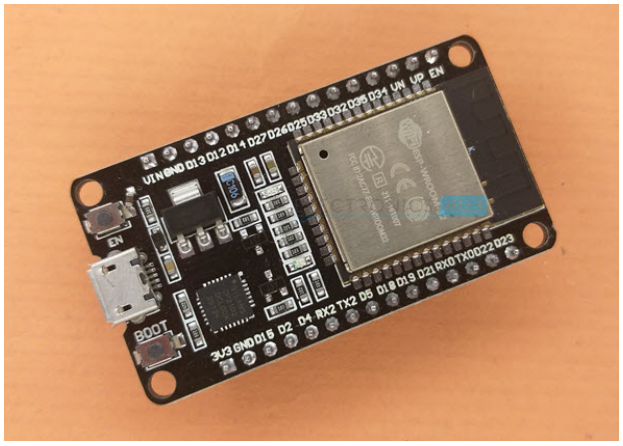
Circuit Diagram



Here LED bulbs are used as replacement of solenoid door locks.

Used Hardware Components

1.ESP32



There are 34 programmable GPIOs and Inbuilt wifi. The ESP32 board operates between 2.2V to 3.6V. But we supply 5V from the Micro-USB port. It is programmed with an Arduino IDE.

2.LCD Display

The relevant messages are shown here. The maximum current draw is about 200mA.
Operating Voltage : 4.7V to 5.3V

3. LED Bulbs - They are used to show output of solenoid door lock
4. Potentiometer - It is used to adjust the contrast of the LCD display
5. Resistors - for the protection of LED bulbs
6. Jumper wires

MQTT Communication

The following topics are subscribed to control signals to the primary actuators of the system of a room in the ground floor.

`326project/smartbuilding/occupancy/0/1/lcd`

When the room limit is exceeded, this topic returns ‘true’ boolean value.

`326project/smartbuilding/occupancy/0/1/ledred`

When an authorized person enters the room, this topic returns ‘true’ boolean value.
When an unauthorized person enters the room, this topic returns ‘false’ boolean value.

Controlling Actuators

1.When an authorized person enters the room.
    The LCD panel displays the message “Authorized, Access Granted”.
    The Green LED is turned ON.

2. When unauthorized person enters to the room
    The LCD panel displays the message “Unauthorized, Access Denied”.
    The Red LED is turned ON.

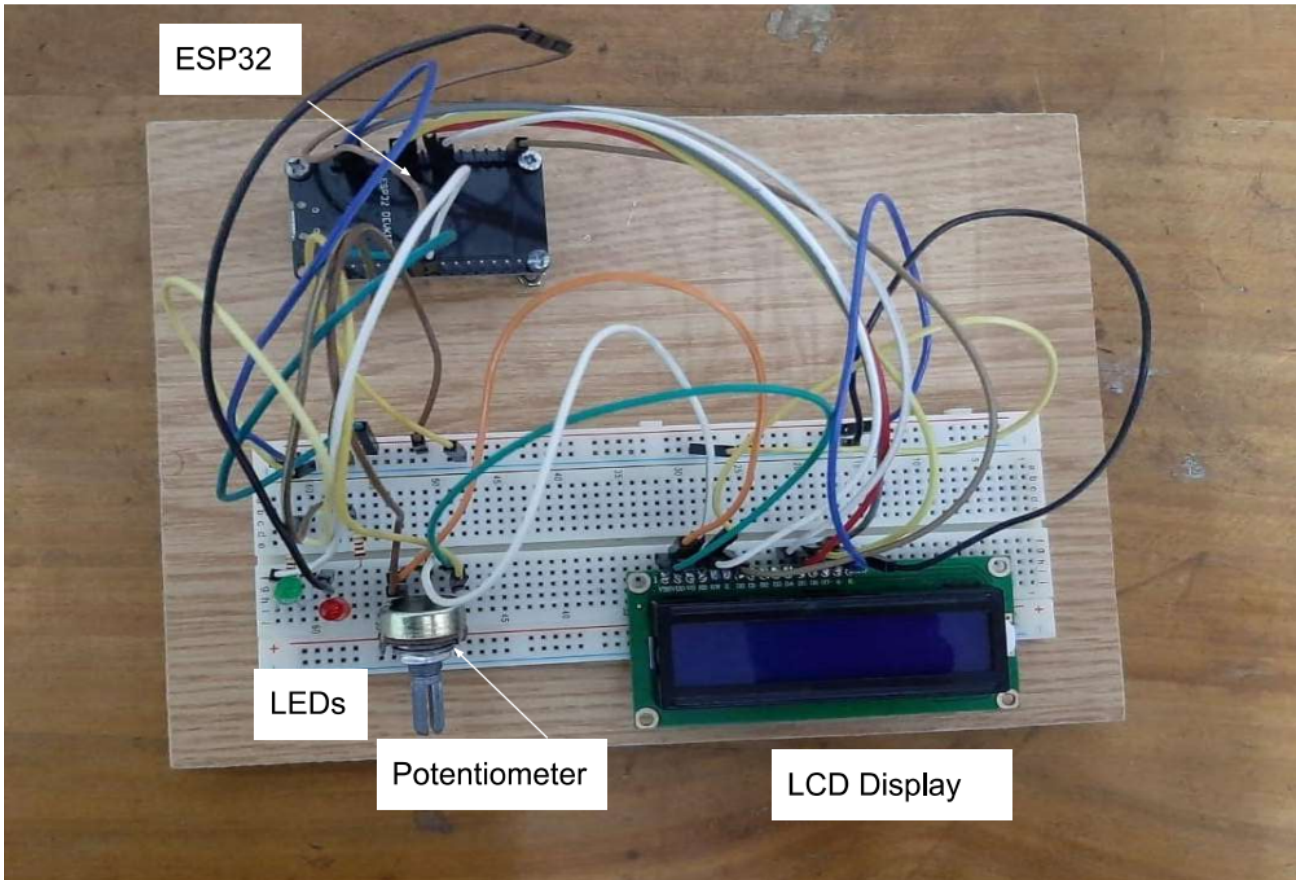
3. When the room limit is exceeded.
    The LCD panel displays the message “Crowded, Limit Exceeded”.
    The Red LED is turned ON.

4. When the room limit is not exceeded.
    The LCD panel displays the message “Not crowded”.
    The Green LED is turned ON.

Future Implementation

Here we are using LEDs to show up the functionality of solenoid door lock. In future implementation we hope to replace LEDs with Solenoid door locks. Then we have to think of power specifications of solenoid door lock as well.

Actual Implementation



Read data from MQTT Server display status of the system on SCADA

Introduction

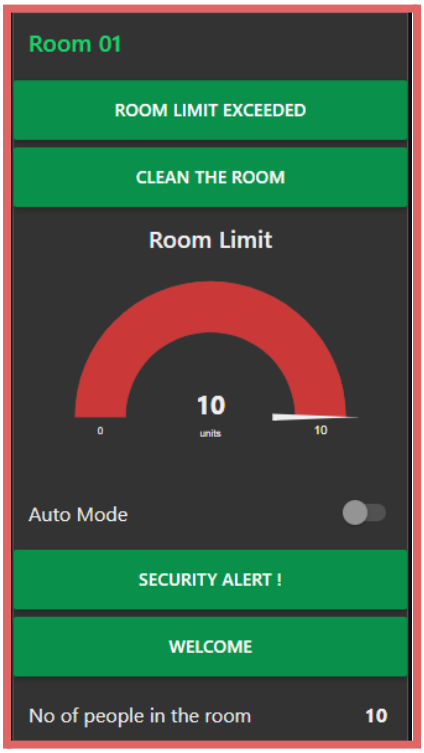
Under this section , the main purpose is to receive data from the sensors and then this raw data is sent to necessary end points to be processed and finally results are sent to the SCADA system in order to be displayed so that the user can make necessary decisions.
Here the main three data that are received to the MQTT broker are

1. Data from the Ultrasonic sensor
2. Data from the RFID
3. Data from the Keypad

Here the data sent by the ultrasonic sensor to the MQTT broker. Then the MQTT broker sends it to the database. By processing the data in the database the count is sent to the MQTT broker. Then this count value is sent to the flow control in order to be further processed and based on the conditions given in flow control necessary results are displayed on the dashboard.

Similarly RFID and Keypad data given by the sensors are processed using flow controls and then the necessary results are displayed on the dashboard.

Design choices



1)Displays data read from ultrasonic sensor

Here the raw data from the ultrasonic sensor are 1 and -1 . These are sent to the database inorder to get the count. Then this count is published to a certain topic. Therefore using this data published , we send it to the flow controls for the necessary decisions . And based on the decisions the results are displayed on the NodeRed dashboard.

Here the number of people in a room is displayed in a gauge indicator and when the room limit is exceeded it shows its maximum of 10 people.

Below that the dashboard displays the actual count even though the room limit is exceeded.

Also when the room limit is exceeded an alert notification will pop up on the dashboard.

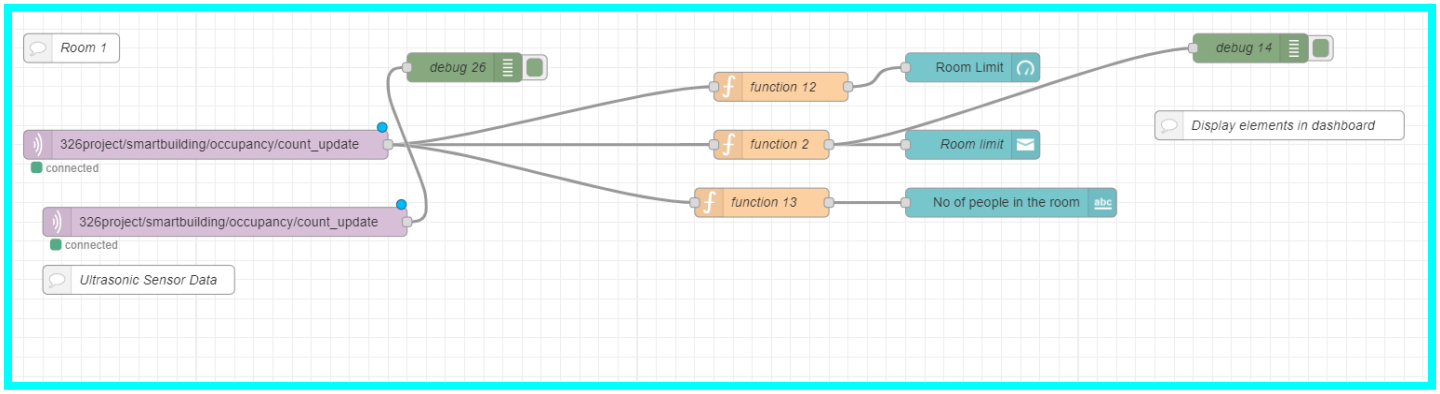
2)Displays data read from RFID sensor

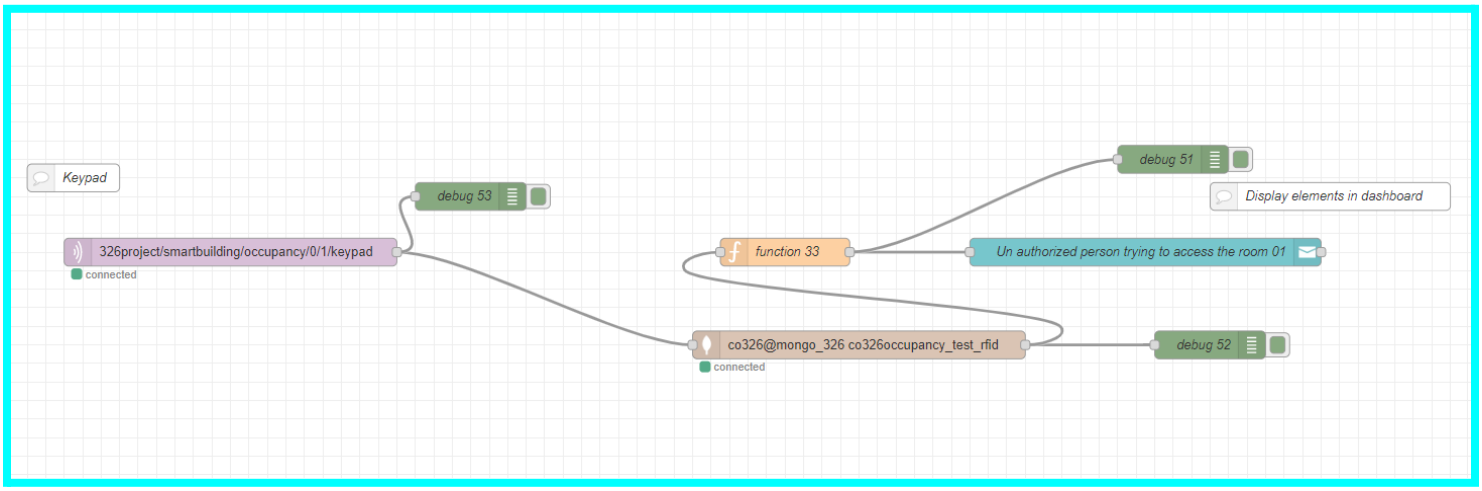
The Data(RFID) published to the topic is read and after confirming whether it is an authorized id with the database and based on few flow control conditions an alert notification will pop up on the dashboard.

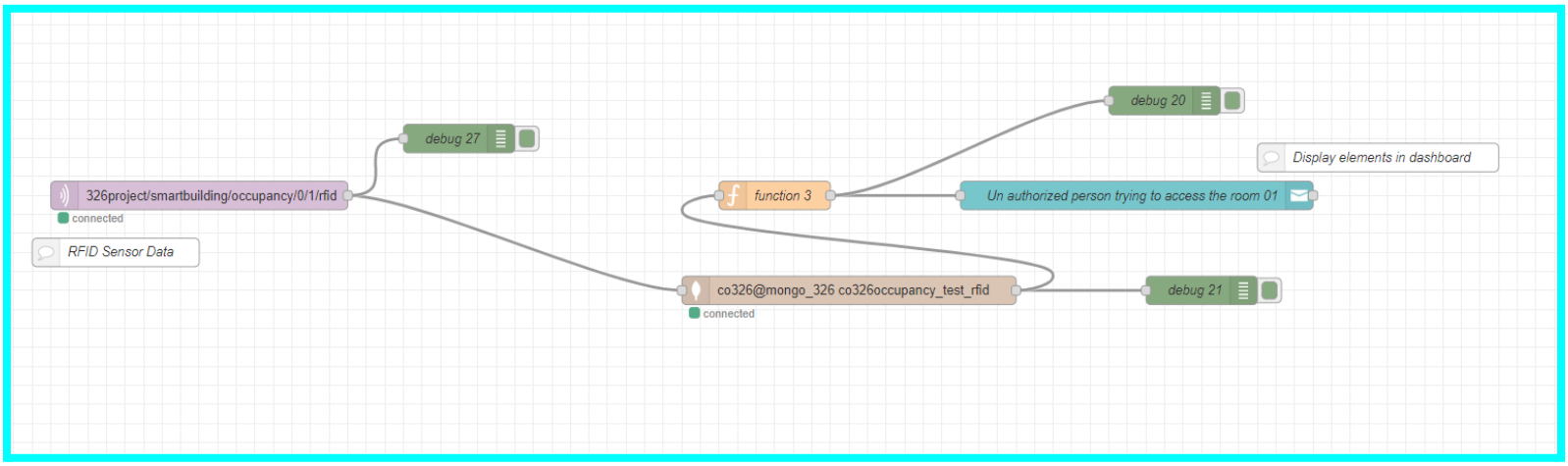
3)Displays data read from Keypad

The Data published to the topic is read and after confirming whether it is an authorized id with the database an alert notification will pop up on the dashboard.

Implementation

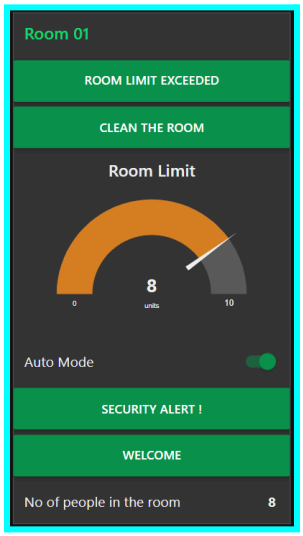


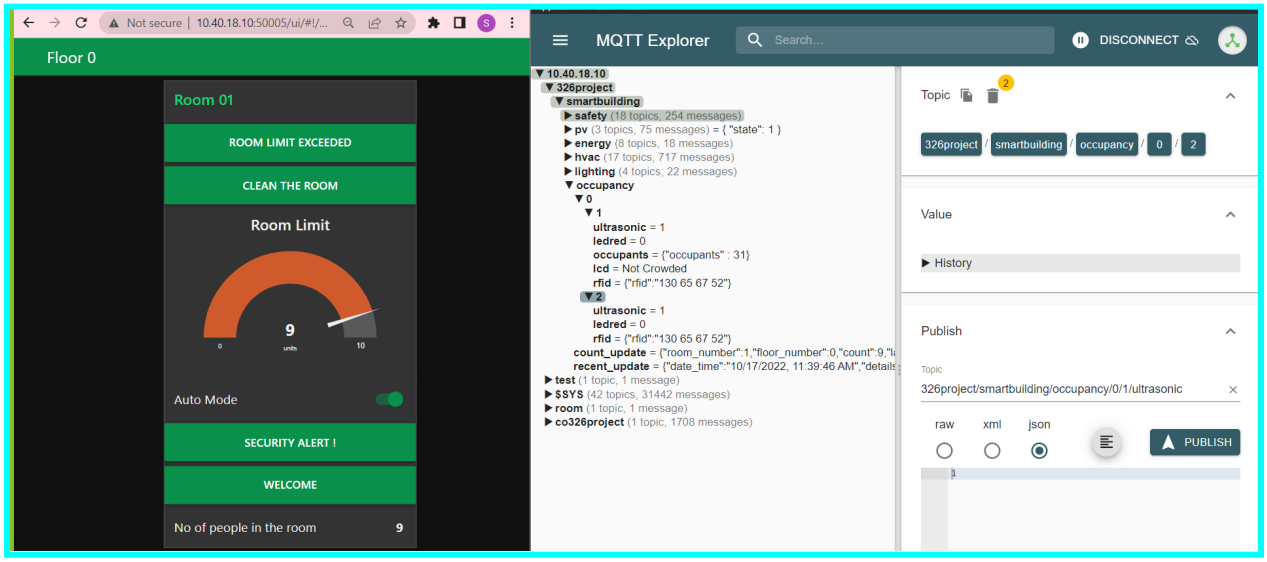




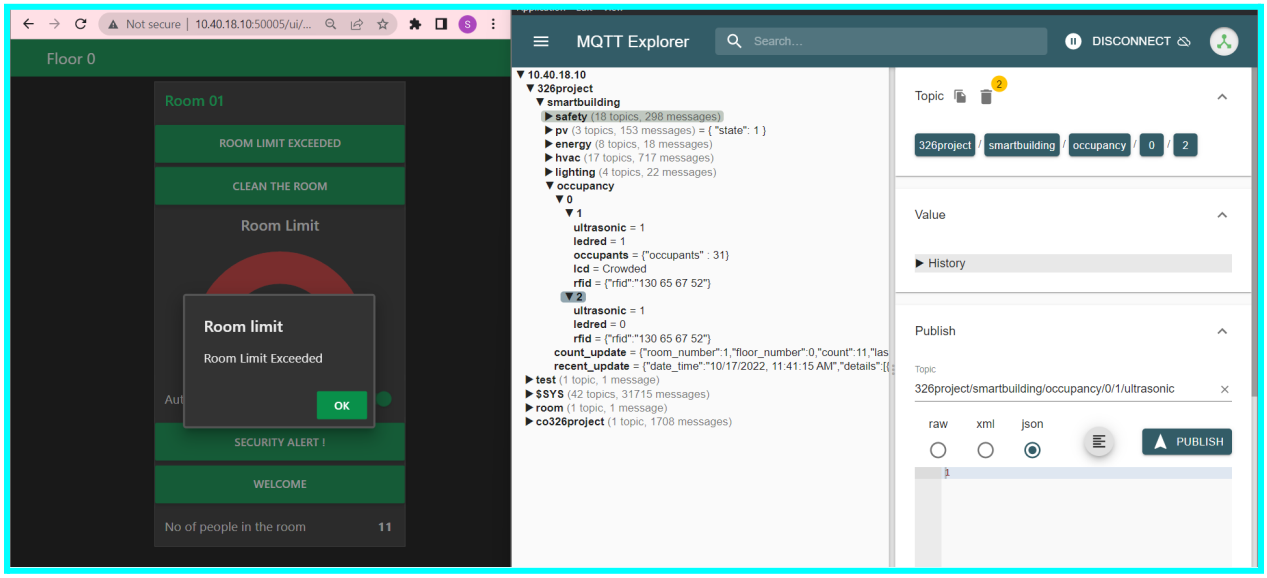
Tests and results

1. Shows the no of people in the dashboard when data is published to ultrasonic sensor topic

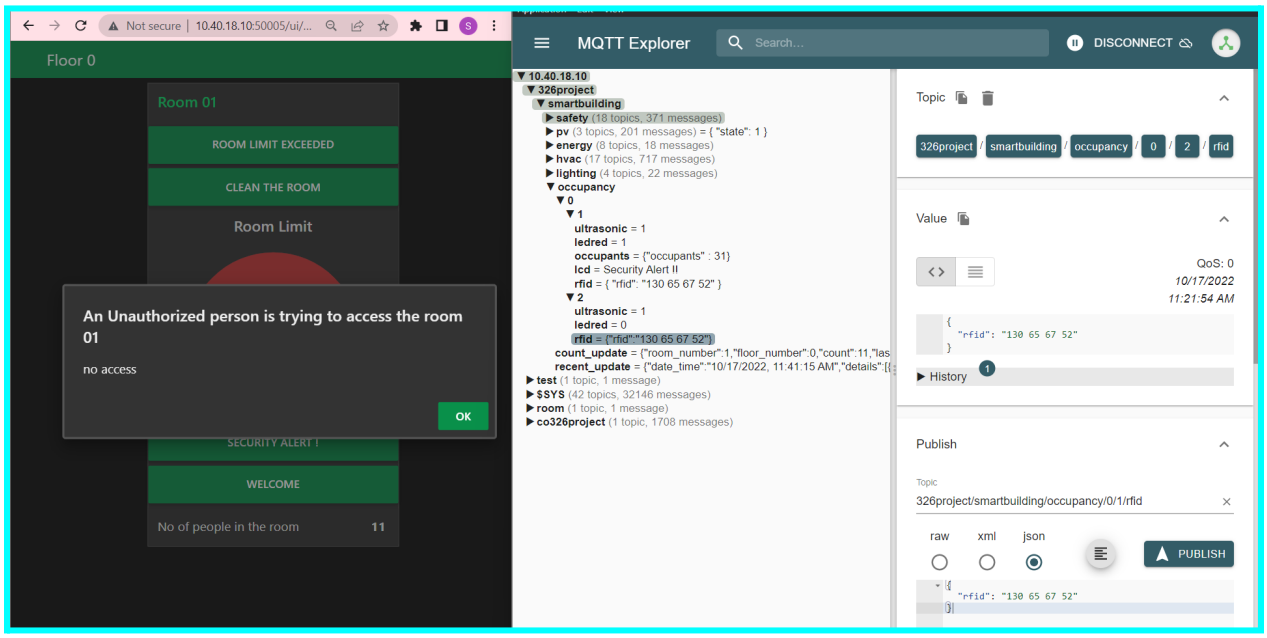




2.When room limit is exceeded a notification pops up on the screen



2.When the wrong RFID is published to topic an alert notification pops up.

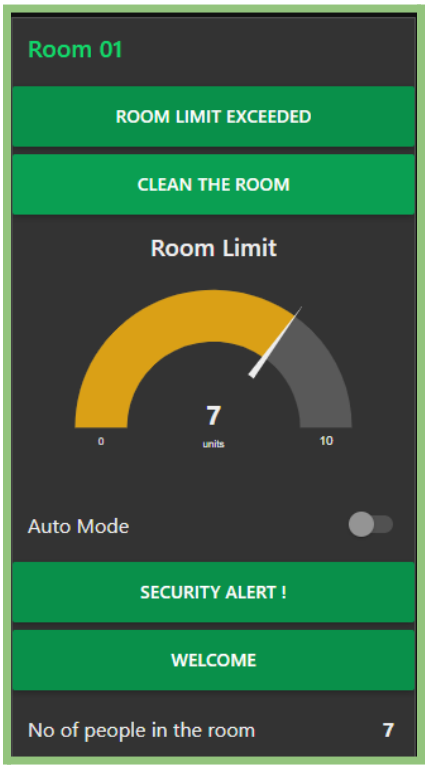


Inputs from the SCADA should send to the MQTT server

Introduction

Under this section , the basic functionality is to transfer data efficiently to the MQTT server from SCADA. As the SCADA , we have implemented the NodeRed dashboard.That is , based on the data displayed to the user , the user can make decisions and input data in order to handle the actuators remotely. And also we have given another option called the auto mode , this eliminates the need of a user to input data. Here based on the read data from the sensors , the decisions are automatically taken and control signals are sent to the actuators.

Design Choices



In the NodeRed dashboard we have given two options to the user.

1. AUTO MODE
2. MANUAL MODE

Manual Mode : Here based on the read data from the sensors the user will have to make decisions on his own and input necessary control signals to handle the actuators.

Auto mode : Here based on the read data from the sensor the system will automatically make necessary decisions and send control signals to the actuators.This mode eliminates the need of a user to be inspecting the SCADA 24/7.

Inorder to check the room limit we have used a gauge indicator.Under normal instances the count of the room is constantly displayed at the bottom part of the dashboard “ No of people in the room : " .This constantly updates when a person enters or leaves the room. The full scale deflection of the gauge is set to 10 people. That is if the room count exceeds 10 , then the gauge will show a full scale deflection.

Alongside it when the room count exceeds a notification is set to pop up informing that the room count has exceeded.

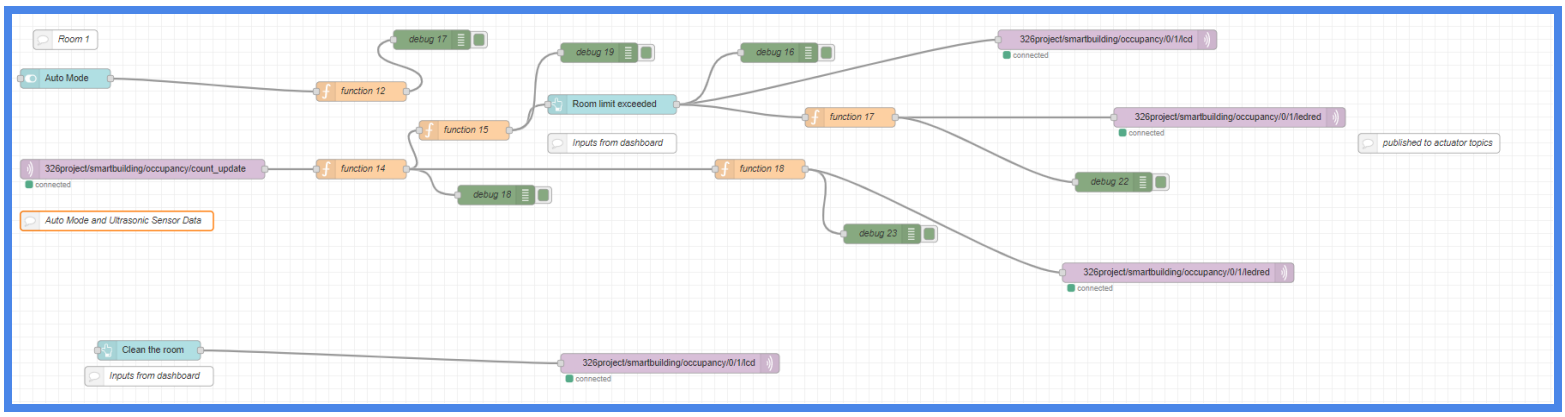
Under the manual mode , if the user detects that the room count has exceeded from the SCADA then we have given the user an option to click on a button “ ROOM LIMIT EXCEEDED “ . As a result of this input signal , a message is displayed on the LCD screen that we have used at the entrance of the room notifying that the room is over crowded and not to enter until it is free again.

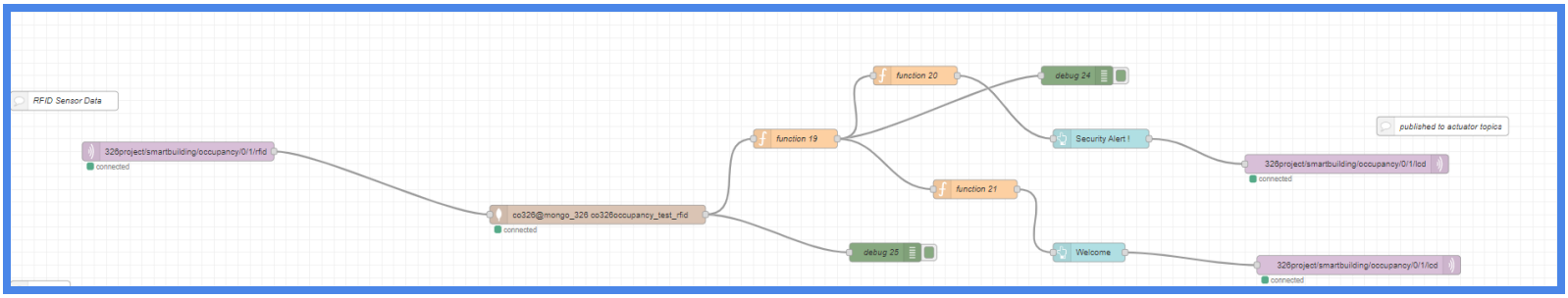
If there are security breaches (due to the RFID and Keypad access controls) the user will be notified with a pop up alert notifying a security breach.
Under the manual mode we have provided a button “ SECURITY ALERT “ for the user. Once this is pressed , a message is displayed on the LCD screen notifying unauthorized access and an alarm will go off. (To represent an alarm we have used a red LED)

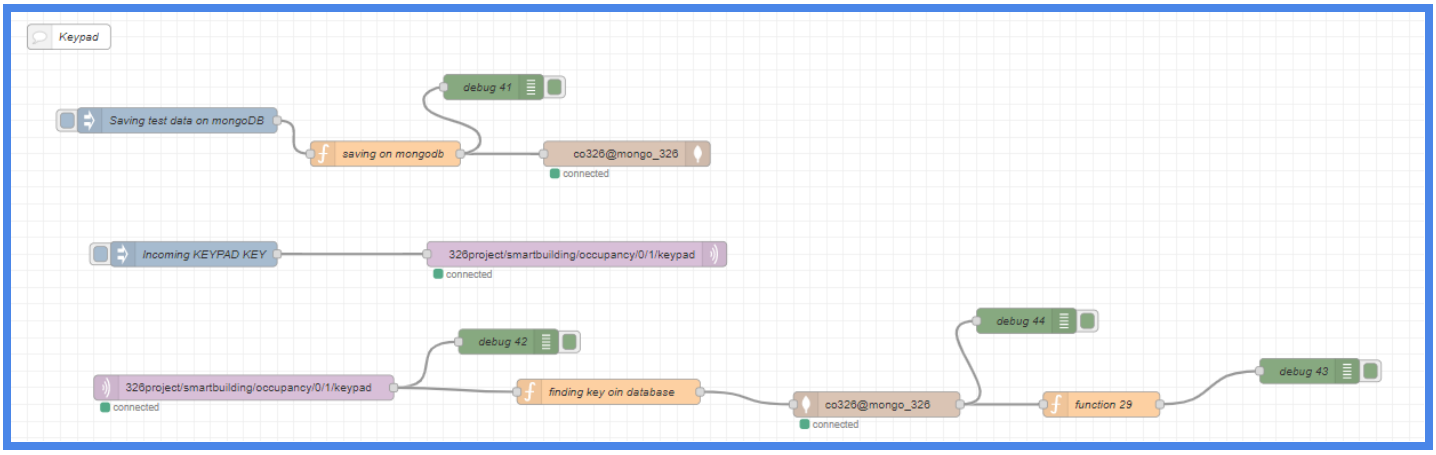
If an authorized access is detected by the User , then he is given an option to display a Welcome note on the LCD display using the “ WELCOME “ button under manual mode.

Another option is provided to the user to input signals to the cleaning staff and the automatic cleaning equipment to start cleaning the room when it is required using the “ CLEAN THE ROOM “ button in the manual mode.

Implementation

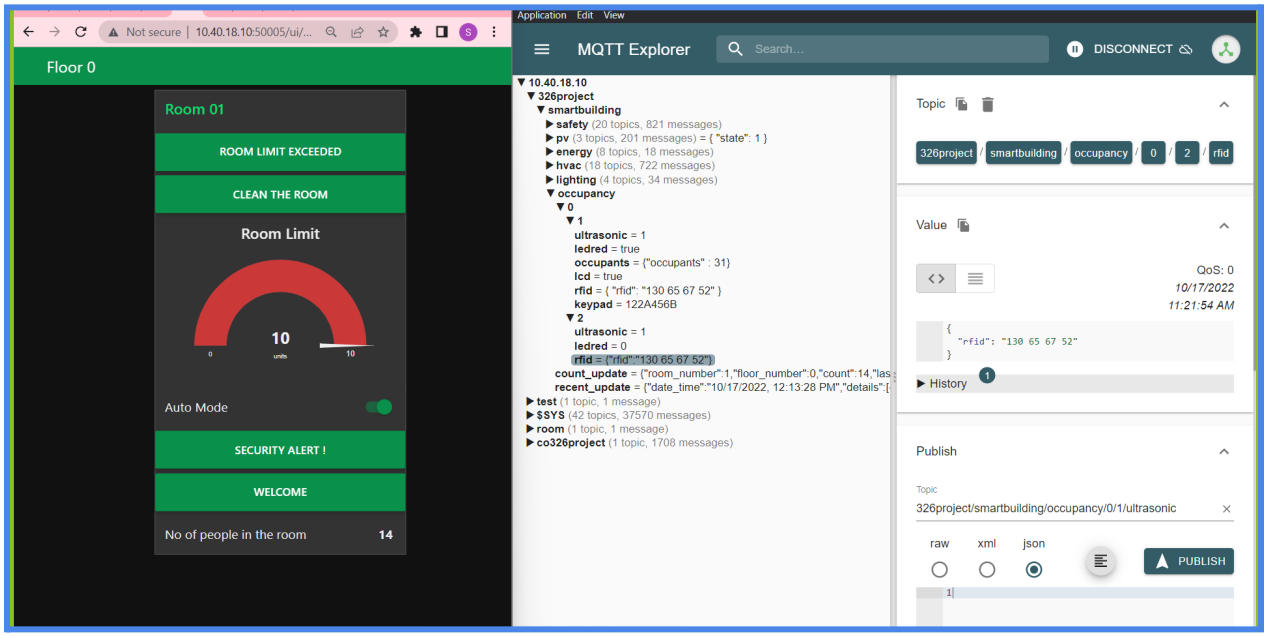

Room 1
Auto Mode
function 12
debug 17
debug 19
debug 16
Room limit exceeded
function 15
Inputs from dashboard
function 17
326project/smartbuilding/occupancy/0/1/lcd
connected
326project/smartbuilding/occupancy/0/1/ledred
connected
published to actuator topics
debug 22
326project/smartbuilding/occupancy/count_update
connected
function 14
debug 18
function 18
debug 23
Auto Mode and Ultrasonic Sensor Data
326project/smartbuilding/occupancy/0/1/ledred
connected
Clean the room
Inputs from dashboard
326project/smartbuilding/occupancy/0/1/lcd
connected


RFID Sensor Data
326project/smartbuilding/occupancy/0/1/rfid
connected
co326@mongo_326 co326occupancy_test_rfid
connected
function 19
function 20
debug 24
Security Alert !
published to actuator topics
326project/smartbuilding/occupancy/0/1/lcd
connected
function 21
debug 25
Welcome
326project/smartbuilding/occupancy/0/1/lcd
connected

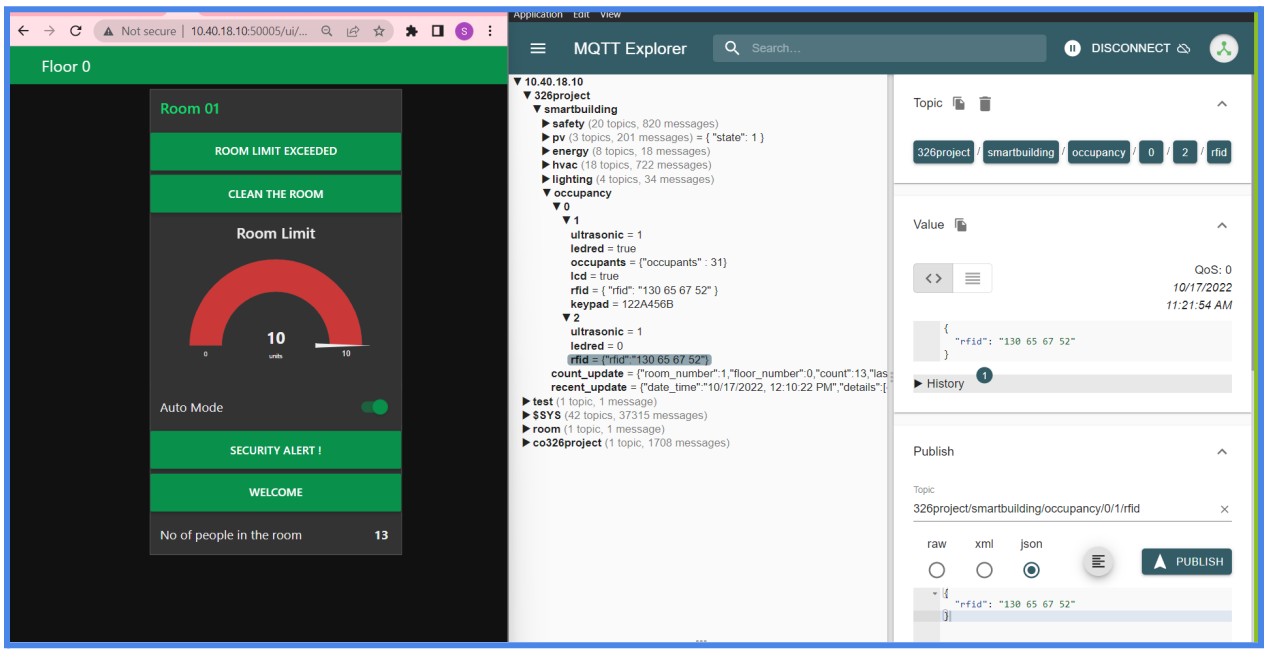

Keypad
Saving test data on mongoDB
debug 41
saving on mongodb
co326@mongo_326
connected
Incoming KEYPAD KEY
326project/smartbuilding/occupancy/0/1/keypad
connected
debug 42
326project/smartbuilding/occupancy/0/1/keypad
connected
finding key oin database
co326@mongo_326
connected
debug 44
function 29
debug 43

Tests and results

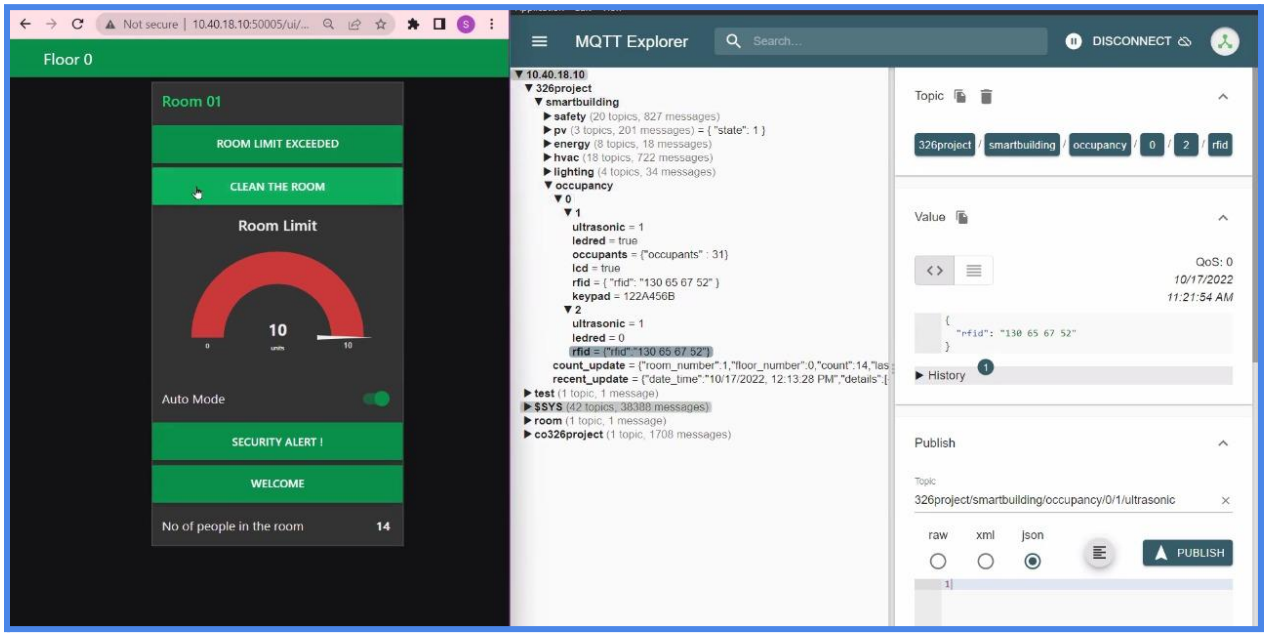
1.When room limit is exceeded that is published to lcd topic .



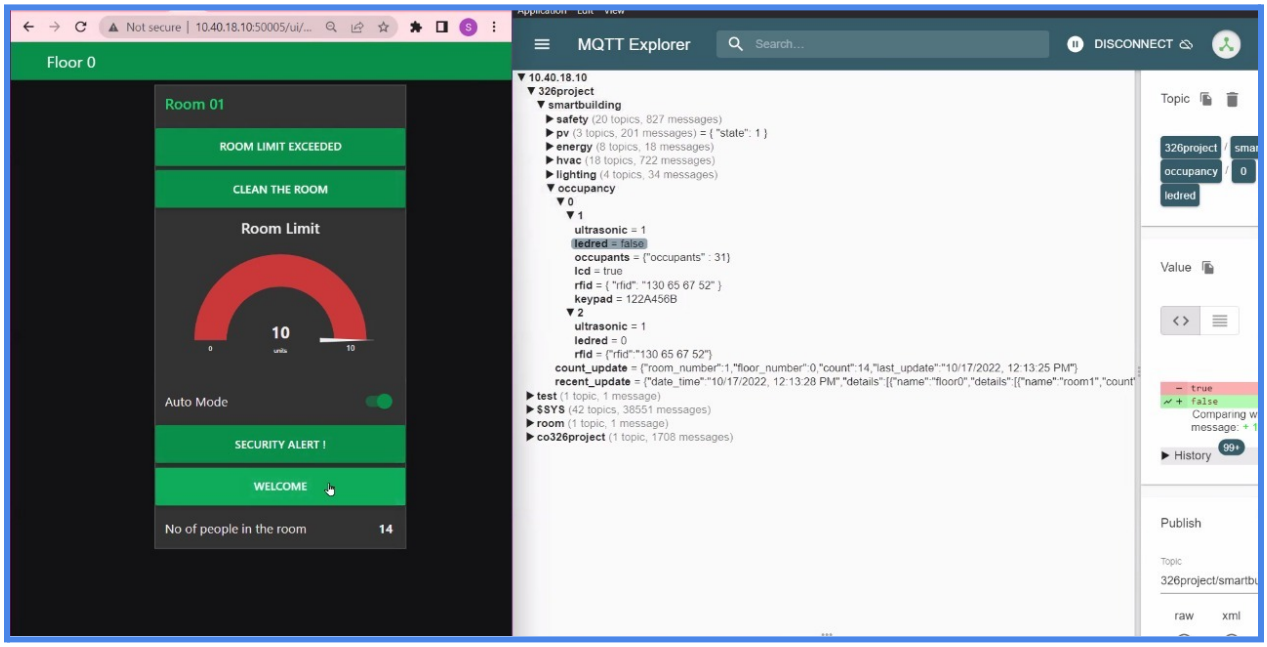
2.When wrong rfid is published to the sensor topic ledred topic is set to true



3.When the “CLEAN THE ROOM ” button is pressed, the lcd topic is set to true.



4.When “WELCOME” button is pressed “ledred” topic is set to false.



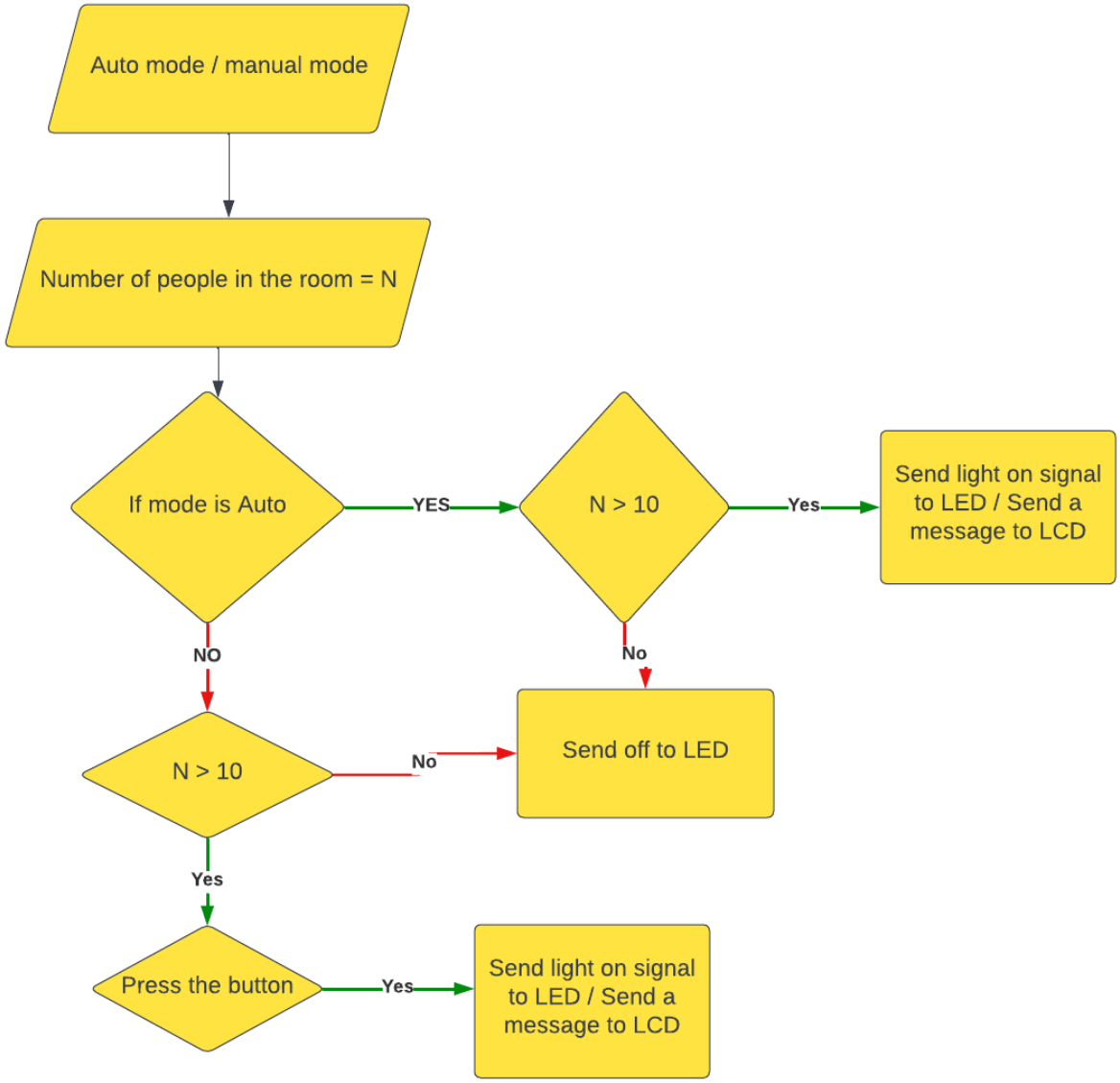
Process controller with operating and optimizing process and algorithms

1.Process control for occupancy threshold maintenance

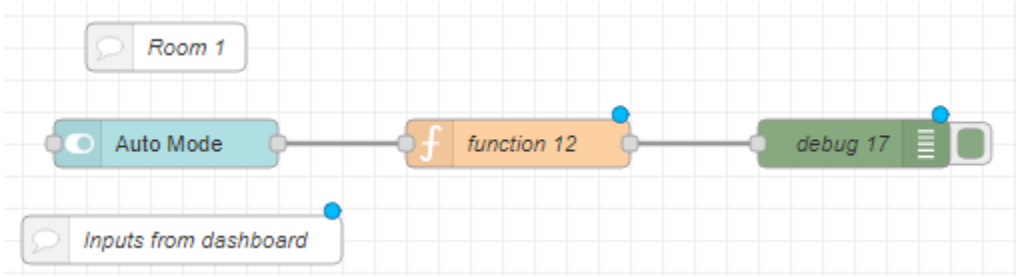
There are two modes in the Scada dashboard for control purposes. Auto and manual are those two modes.

When the number of persons in a room exceeds. In the auto mode a warning message will be sent to the LCD automatically. In the manual mode there is a button to send the warning message to the LCD. So a user who controls the scada will have to press it.

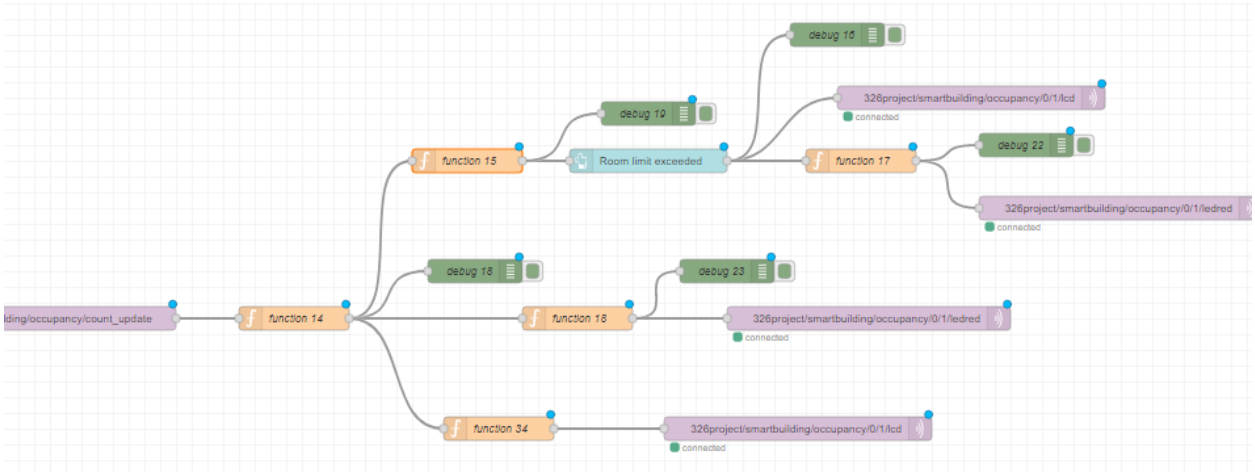
Functions are implemented in such a way that takes the input from the scada to check the mode. And then retrieve the updated count of people in the room by the database and check with the threshold value.



Functions used / debugging



function12- receiving the signal from the button on scada and set the global variable to on/off



When count is increased or decreased that message will received to the count_update topic

Debugging is used to test the incoming and outgoing data types.

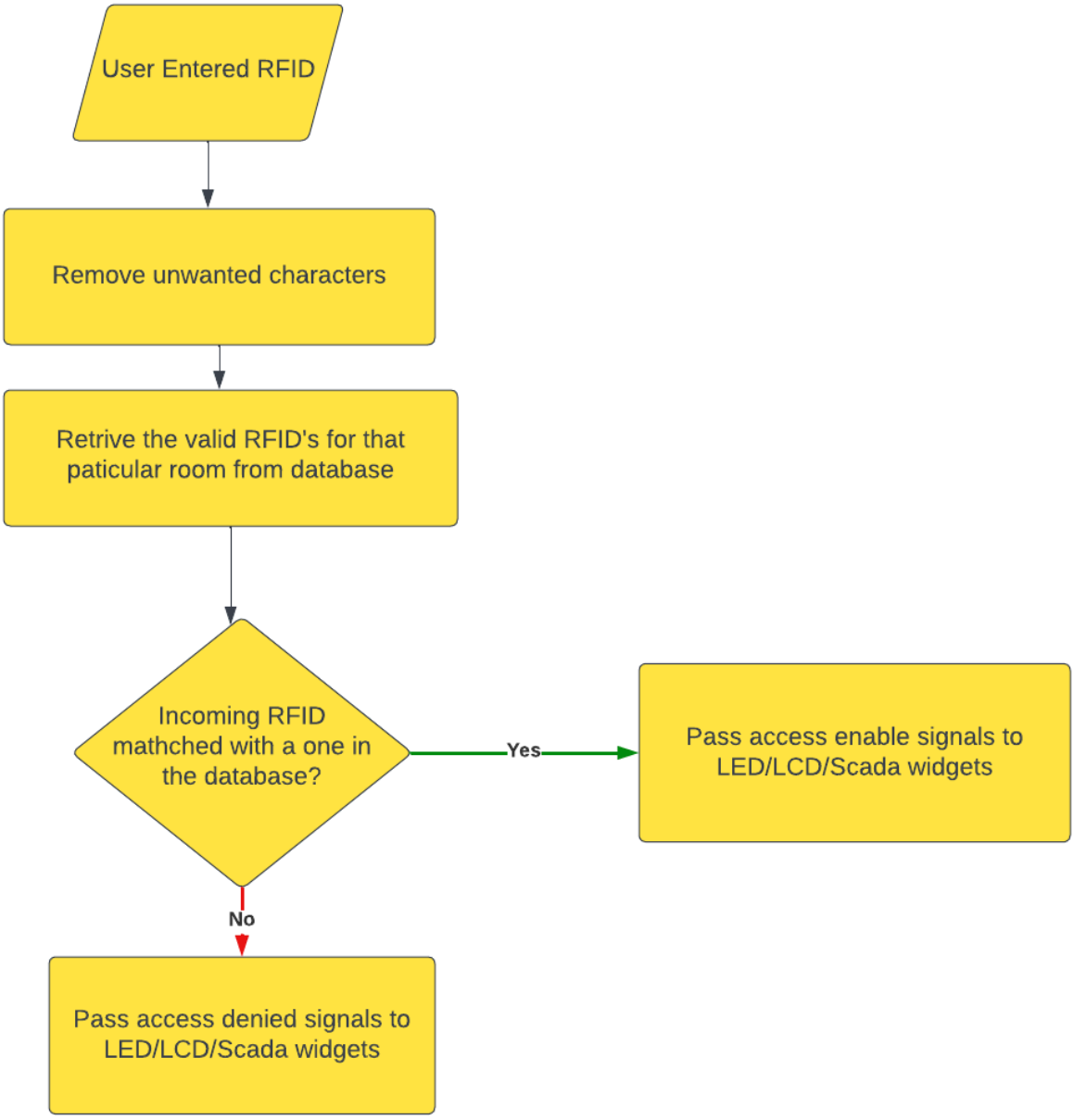
Function	Purpose
function14	receiving the people count from the database, and if it is more than the threshold make the global variable ON or otherwise OFF. Save the room numbers and floor number to use check occupancy in separate
function15	when in auto mode and limit exceeds pass a message to trigger a button on scada
function17	to send 1 to lcd
Function18	when room limit is not exceed send ‘false’ to LED
function34	when room limit is not exceed send ‘false’ to LCD

Debugging is used to test the incoming and outgoing data types.

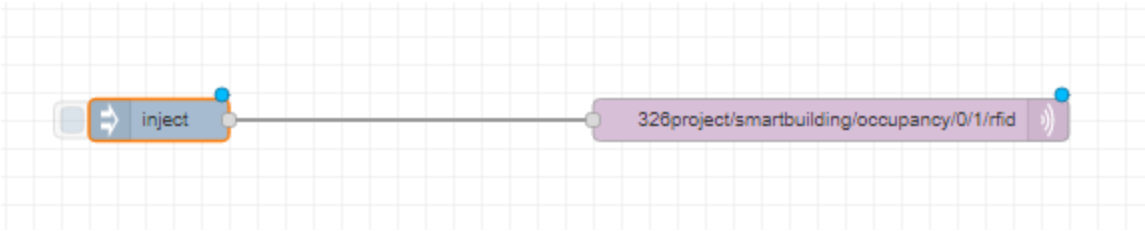
2. Authentication

There are two security levels built according to the devices used.

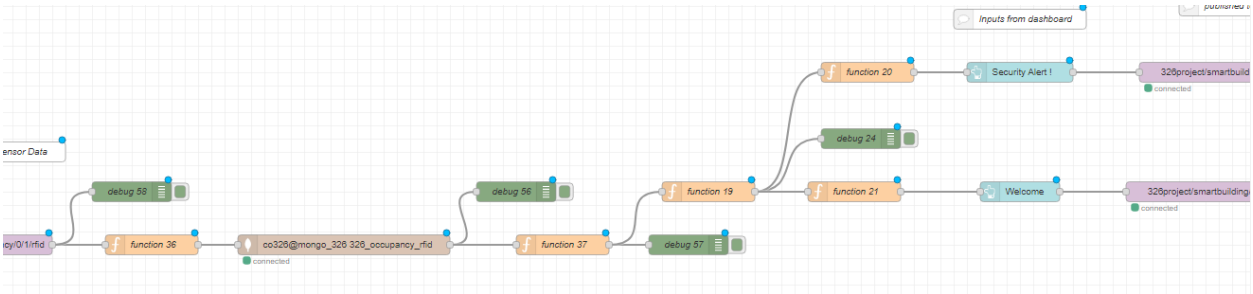
a. When using RFID there is a one unique ID for every person.
These ID's are saved in the mongoDB database. When user tries to access with his RFID it is matched with the ID's saved on the
database. According to the outcome of that access signals are passed to the actuators. RFID's are used to get a higher security level.

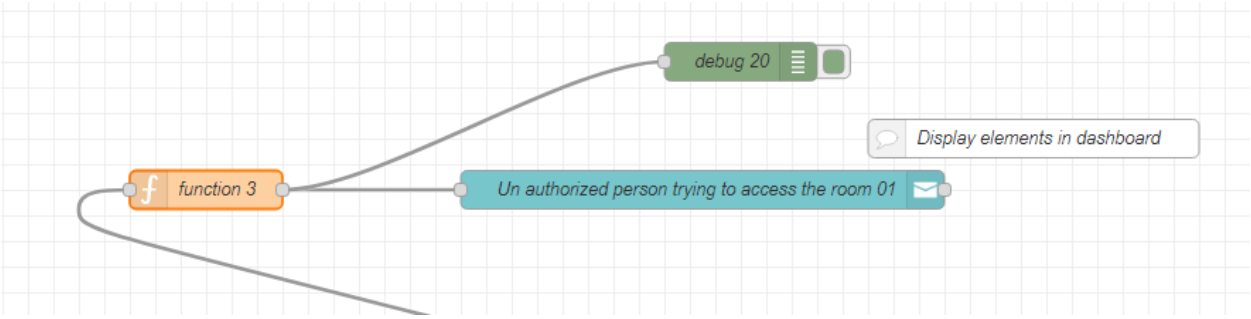


Functions and debugging



Used to publish test data to a topic

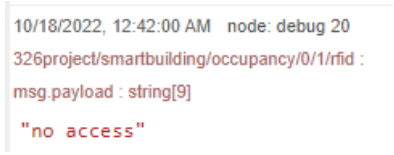




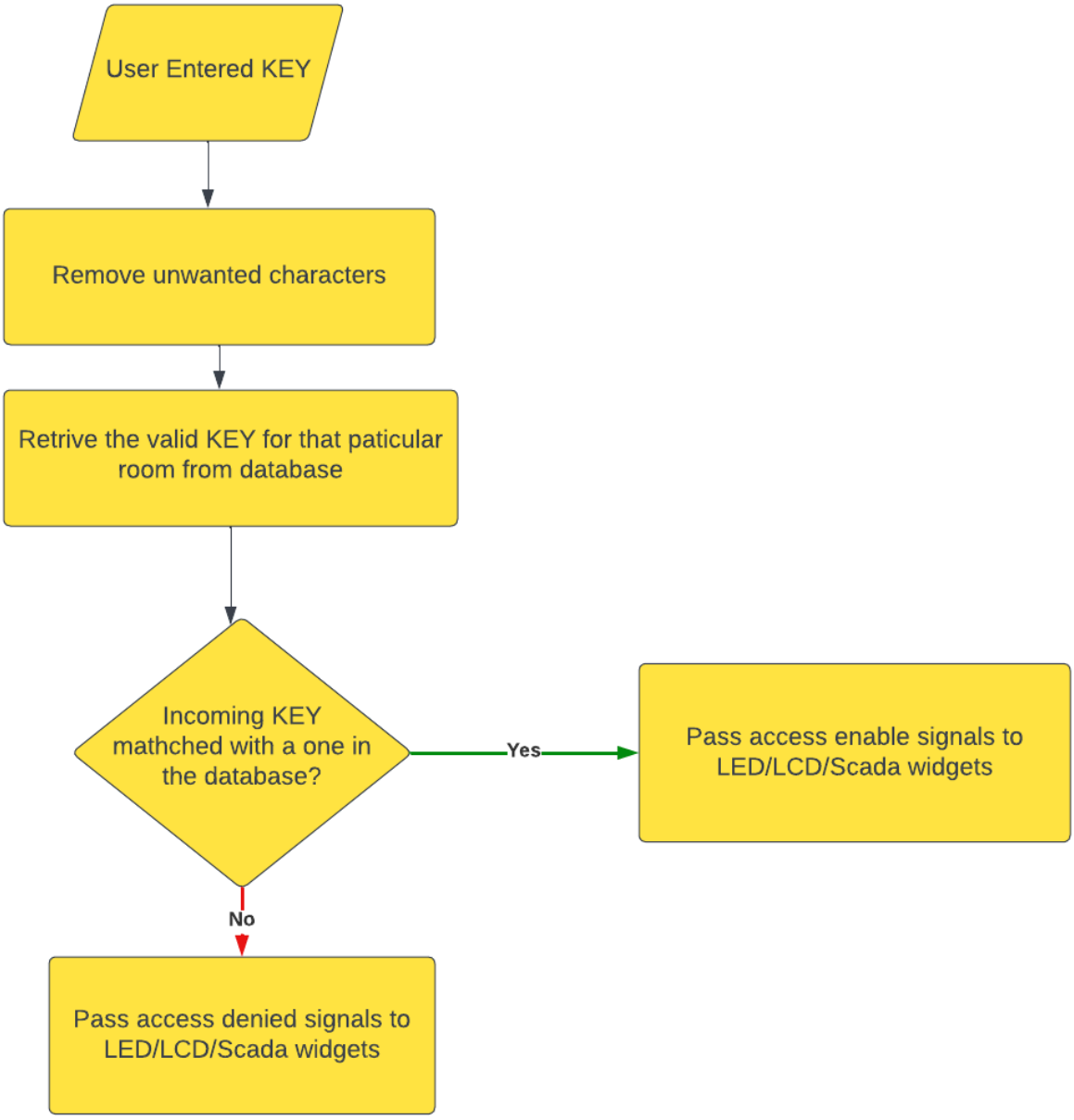
To display messages on SCADA

function	Purpose	Debugging
		Debug 58 Used to identify the incoming string format from the RFID scanner

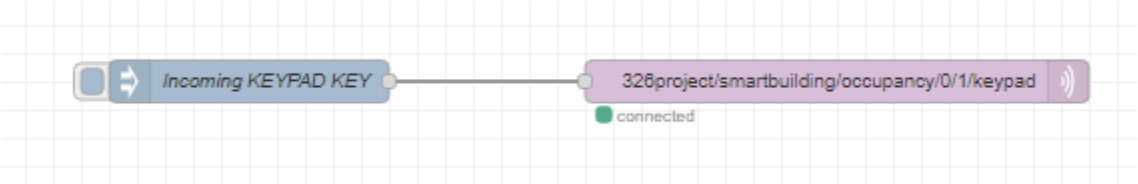
		10/18/2022, 12:42:00 AM node: debug 27 326project/smartbuilding/occupancy/0/1/rfid : msg.payload : string[15] "* 231 85 243 11"
Function 36	get the incoming RIID and process it and save it on a global variable get a list of RFID's valid for a given room	Debug 56 10/17/2022, 11:48:23 PM node: debug 56 326project/smartbuilding/occupancy/0/1/rfid : msg.payload : array[1] ▾ array[1] ▾ 0: object _id: "634d1d500502e71145aee68b" floor_number: 0 room_number: 1 ▾ rfid: array[2] 0: " 231 85 243 25" 1: " 250 187 91 89" keypad: "123A" Receives RFID's for the Relevant room
Function 37	get the RFID list Match the incoming RFID with list of RFID's	Debug 57 For a invalid RFID 10/17/2022, 11:48:23 PM node: debug 57 326project/smartbuilding/occupancy/0/1/rfid : msg.payload : number 0 For a valid RFID 10/17/2022, 11:57:59 PM node: debug 57 326project/smartbuilding/occupancy/0/1/rfid : msg.payload : number 1
Function 19	change the global enter state according to the given access (0/ 1)	
Function 20	return a triggering message for unauth access	

Function 21	return a triggering message for auth access	
Function 3	According to the entering state pass “access” / “no access “ messages to the Scada	Debug 20 For unauth access 

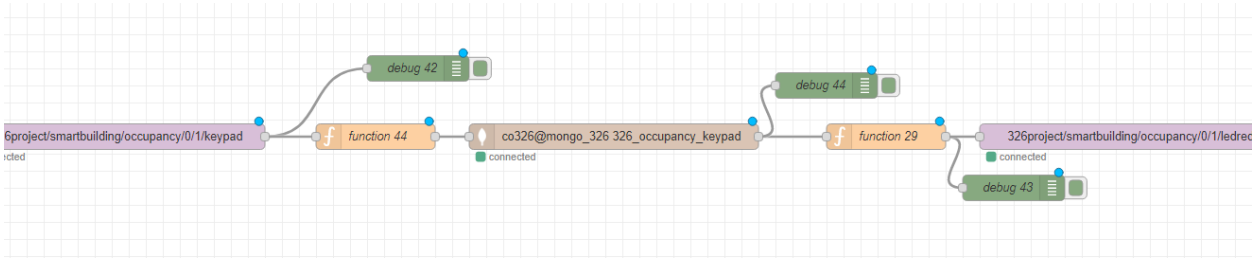
b.In the lower security level keypad is used. For one certain keypad there is one key. This key is saved securely on the database to retrieve again in case of a server failure. People who needs to enter the room must know the KEY for the keypad. When they Enter the KEY it will be matched with the saved key on the database and then access signals will be published to the actuator topics.

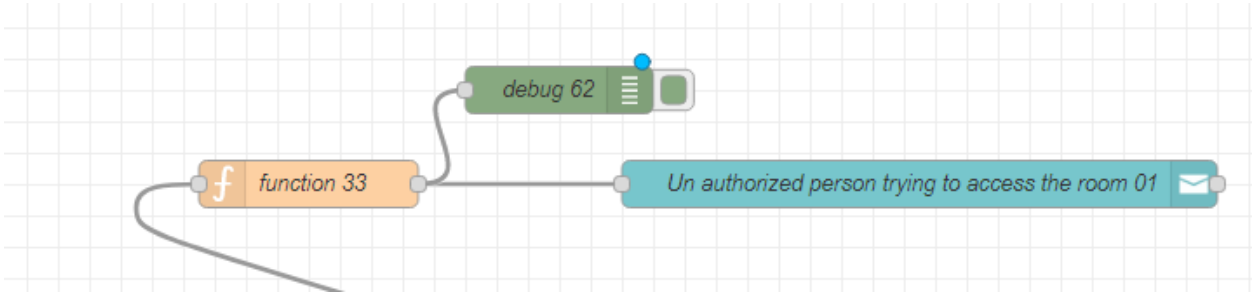


Functions and Debugging



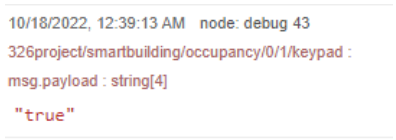
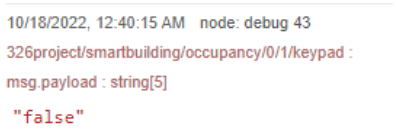
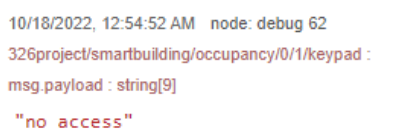
Used to send test KEY's





To send messages to the SCADA

Functions	Purpose	Debugging
		Debug 42 Used to identify the incoming string format from the keypad 10/18/2022, 12:48:53 AM node: debug 42 326project/smartbuilding/occupancy/0/1/keypad : msg.payload : string[5] "*456C"
Function 44	receive the incoming key and process it find the relevent key for the room from the database	Debug 44 10/18/2022, 12:35:00 AM node: debug 44 326project/smartbuilding/occupancy/0/1/keypad : msg.payload : array[1] ▾ array[1] ▾ 0: object _id: "634d91a90502e71145c61f53" floor_number: 0 room_number: 1 keypad: "456B" Receving the relevant KEY For the relevant room

Function 29	check the incoming key with the key on the database	Debug 43 For the valid key  For a invalid key 
Function 33	According to the entering state pass “access” / “no access “ messages to the Scada	Debug 62 For unauth access 

MQTT Data, Commands and events should be stored in the database

Introduction

Under this section two main things are done. There are

1. Data retrieve and modify the format
2. Modify the data and store them into the database

In the occupancy and access control section has Ultrasonic sensors, keypads and a RFID for each room in each floor. The ultrasonic sensors are used to identify whether a person get in to the room or get out from the room. RFID and keypad are used to authenticate the users. Each room has one unique password. That value should be entered by using the keypad and each room has authorized RFIDs. These values are stored under the separate collections in the mongoDB database. The value under each collection will be used by the other groups such as **Energy**, **Lighting** and **Control** groups. Therefore under this section stored data should be notified to the other people who are interested in some events.

Data flow

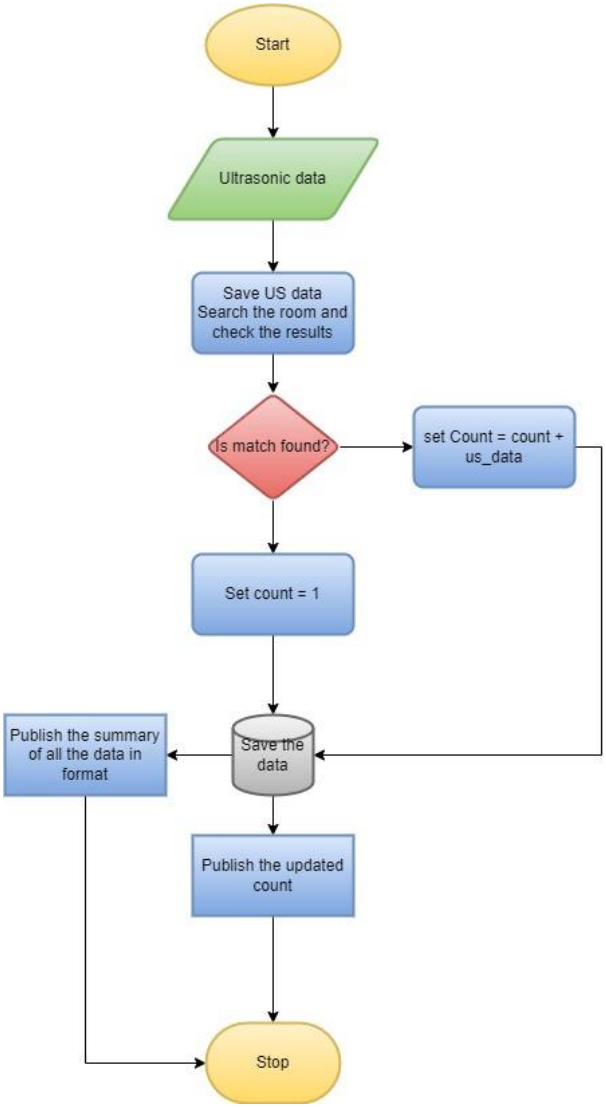
- *Update and notify the count of a room*

  In the smartbulding , each room in the each floor is identified as an one specific unit. When considering the room (If the room has occupancy and access control unit) it has an ultrasonic sensor system to identify the ***get in into the room*** and ***get out from the room*** events. When one of the event that mentioned above happens an integer value is published to the topic mentioned below.

  326project/smartbuilding/occupancy/<floorno>/<roomno>/ultrasonic
  (In here <floorno> is reperesnts the current flow and the roomno is the current room)

  Based on the event one integer value is published to the topic.
  - Get into the room (**1**)
  - Get out from the room (**-1**)

  Under ***this section*** that topics are subscribed and listen to them. Whenever these topics get an message , the relevant room count get updated based on the value.



In the database for the occupancy the following collection is used.

Collections

- 326_occupancy_room

This collection has a specific structure. Each room of a floor is represented by an one entry from this collection.

```
room_number: 1
floor_number: 0
count: 4
last_update: "10/19/2022, 8:46:40
PM"
```

Figure 1 : 326_occupancy_room collection

Consider the figure <number>. In this figure it is shown that the handling part of the room 1 of the floor 0. When the topic 326project/smartbuilding/occupancy/0/1/ultrasonic is get a message (1 or -1) that topic is listened. Then the using function node the data from the ultrasonic sensor is saved to the context in global scope. Then using the mongodb node check whether there is entry for {room_number : 1 , floor_number:0} . If the results is null it means there is no room such that.

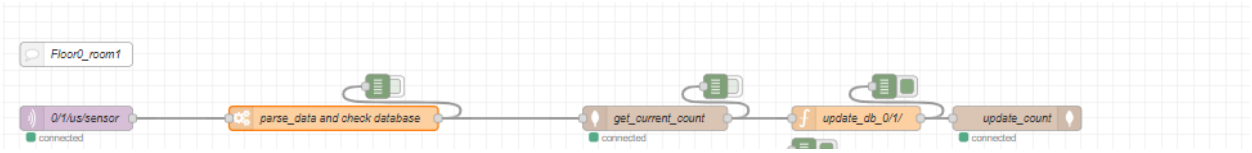


Figure 2 : Flow of the room 1 in floor 0

In the figure 2 when the message is arrived to the mqtt event that message is parsed by the function called “parse_data_and_check_database”. In that function it will save the data that comes from the sensor to global context called “`01_us_data`“. Then from that function serach the relevant room that matches with {room_number:1 , floor_number:0}.

```
msg.collection = "326_occupancy_room";
msg.topic = "01_ultrasonic_data";
msg.operation = "find";
msg.query = {"room_number":1,"floor_number":0}

global.set("01_us_data",msg.payload);


return msg;
```

Figure 3: function to search the room

Then that message is passed to the mogodb in node. Then it searches the relevant room from the collection. Results will be passed to the other end as an array. In the other end the results are parsed. It will check the results array from the mongodb does have atleast element. If there i no element that means relevant room does not have a collection. If it found one then room is exist aready. Now the count should update according to the above two scenarios.

1. If array length is zero ⇒ Create a new collection with given room and floor number with count field 1.
2. If found element ⇒ update the current count and update the found collection

```
let arr = msg.payload;
let sensor_data = global.get("01_us_data");


//Create the date and time objects
var targetTime = new Date();
var timeZoneFromDB = 5.45; //time zone value from database
//get the timezone offset from local time in minutes
var tzDifference = timeZoneFromDB * 60 + targetTime.getTimezoneOffset();
//convert the offset to milliseconds, add to targetTime, and make a new Date
var offsetTime = new Date(targetTime.getTime() + tzDifference * 60 * 1000);

let count;

if (arr.length == 0) {
    count = 1;
}
else {
    count = arr[0].count + sensor_data;
}

msg.payload = { "$set": { "room_number": 1, "floor_number": 0, "count": count, "last_update": offsetTime.toLocaleString() } }

msg.query = { "room_number": 1, "floor_number": 0 };
msg.topic = "db query";
return msg;
```

Figure 4: Update if found and insert new if not found

In here last MongoDB node is set as such a way when there is no collection for the room it will create a new one and update the count. If there is already a entry to the room the the count field will be update based on the mqtt event.

- Count =1 ⇒ If new collection is updated
- Count = count + (1 or -1)

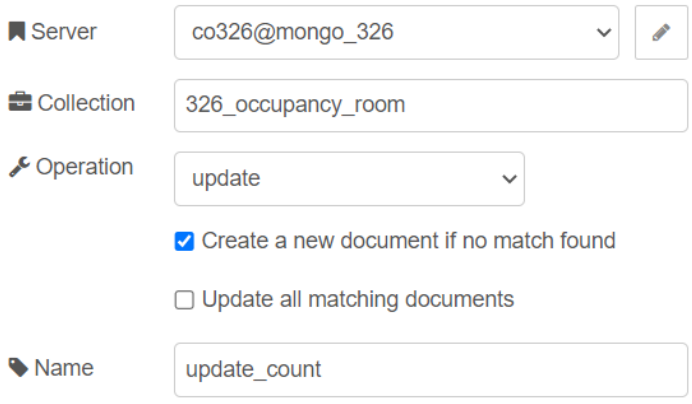


Figure 5: Configuration of MongoDB out evnet for a room

- **Publish the counts**

The count data of a room is important to other filed such as Energy , light and Control . Therefore, when there is some change occur in the database under the occupancy field the counts of the rooms are published to different topics.

This task is done under two separate way. When the change is occurred in the rooms count the current status of the rooms should be required to the **data analytics** part of our group. For that part data should be **preprocessed** before send to the analytic part. To that there is a separate flow in the database section.

Figure 6: Flow for the preprosses

Above flow is listen to the all all functions of the “update_db/floor_number/ room_number” functions. When that function is completed It stays 3s for DB processing the new data the retrieve the all rooms data. Then that data will be sent to the recent_update event.

```
let details = []
let arr = msg.payload;

// Hnadle uncaught errors
if (arr.length == 0) return msg;


//Create the date and time objects
var targetTime = new Date();
var timeZoneFromDB = 5.45; //time zone value from database
//get the timezone offset from local time in minutes
var tzDifference = timeZoneFromDB * 60 + targetTime.getTimezoneOffset();
//convert the offset to milliseconds, add to targetTime, and make a new Date
var offsetTime = new Date(targetTime.getTime() + tzDifference * 60 * 1000);

for (let floor = 0; floor < 4; floor++) {
    let floor_data = arr.filter(element => element.floor_number === floor);
    let inner_details = [];

    if (floor_data.length == 0) {
        for(let temp =1; temp <=2; temp++){
            inner_details.push({ "name": "room" + temp, "count": 0 });
        }
    }
    else {
```

```
        for (let j = 0; j < floor_data.length; j++) {
            inner_details.push({ "name": "room" + floor_data[j].room_number, "count": floor_data[j].count });
        }

    }

    details.push({ "name": "floor" + floor, "details": inner_details });

}

msg.payload = { "date_time": offsetTime.toLocaleString(), "details": details };

console.log(msg.payload);

return msg;
```

Figure 7 : Set the format of the data

The format of the data is given in the figure 8 below. This results are published to topic at the end and tha resuls will be used by the data analytics section of the occupancy.

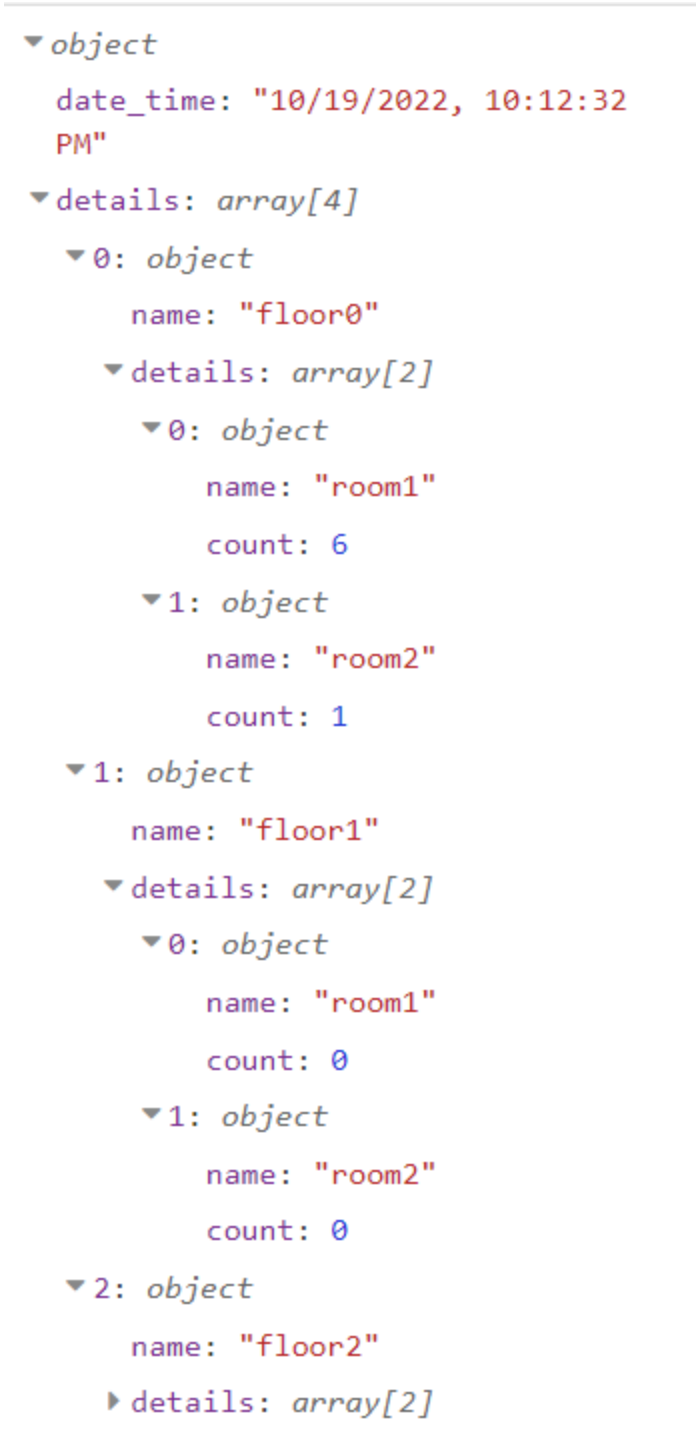


Figure 8: Format of the preprocessed data

The other event is when there is an change of a room occur that speicifc room data is published as a JSON data to the public through the 326project/smartbuilding/occupancy/count_update topic. In here data has the following structure.

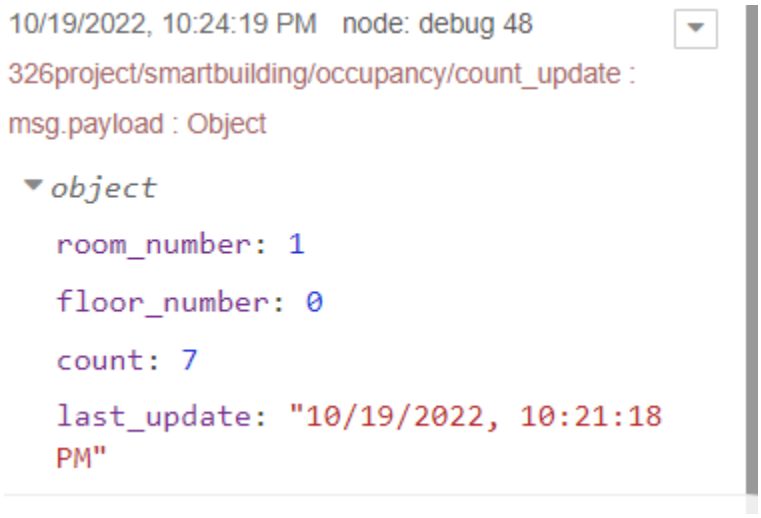


Figure 9 : Format of the updated message of the room

- Send data to the Energy section

  In here Energy section will oublsih the required count with the room number and floor number. Then according to the that room number and floor number data is earched from the collection. When there is match found that data with the count will be published to the topic 326project/smartbuilding/energy/<floornumber>/occupancy.

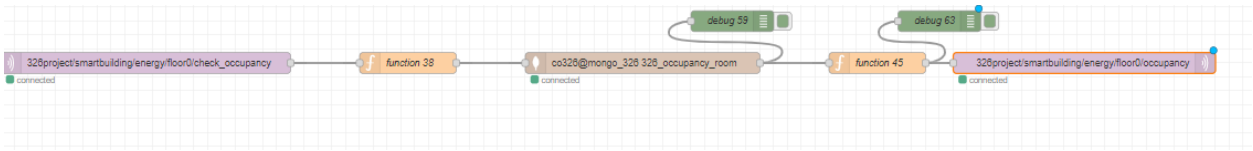


Figure 10: Data published to the Energy control team

Summary

In summary each data has unique flow to handle the ultrasonic sensor. (Figure 11)

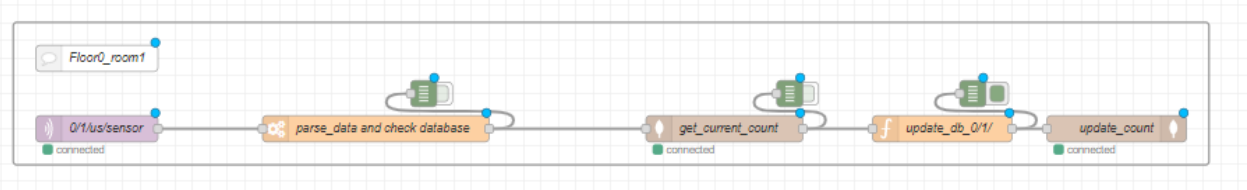


Figure 11: Floor 0 room1 flow

When ever there is update happens that data are published to the relevant topics.

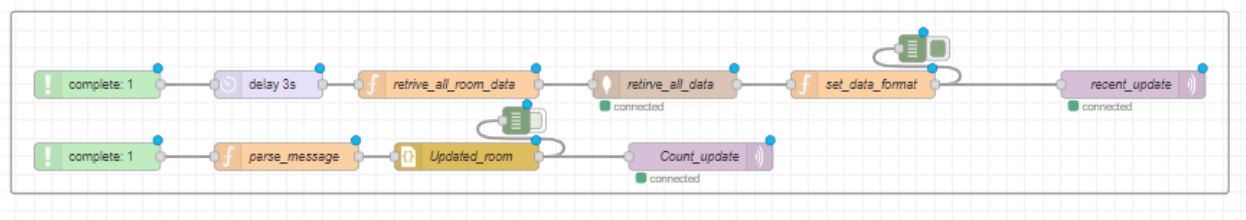


Figure 12 : Publish the change to the other sections

When the energy team needs a specific room details that data will bepublsihed as follows.

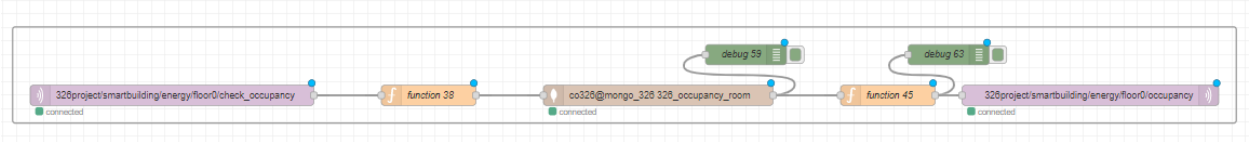


Figure 13 : Published data to the energy team

For each room has a separate flow like in the figure 11.

- RFID and Keypads collections

  Each room has a unique keypad value. That values are stored under the 326_occupancy_keypad collection.

  Considering the RFIDs each rooms have pre-identified list of RFIDs. That values are stored it the 326_occupancy_rfid collection.

  These value are already stored in the database when set the room setup.

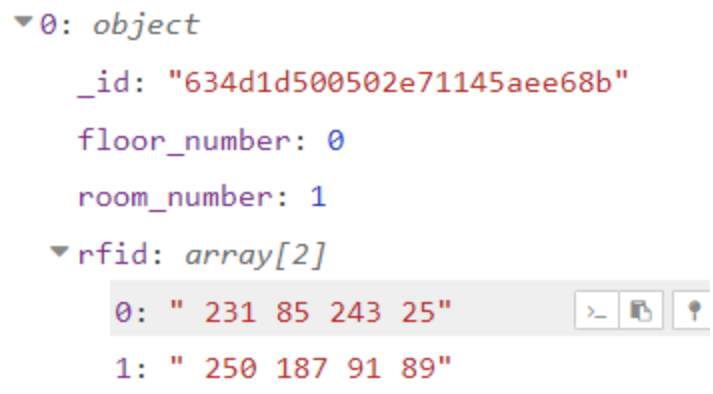


Figure 14 : Sample collection of the RFIDs

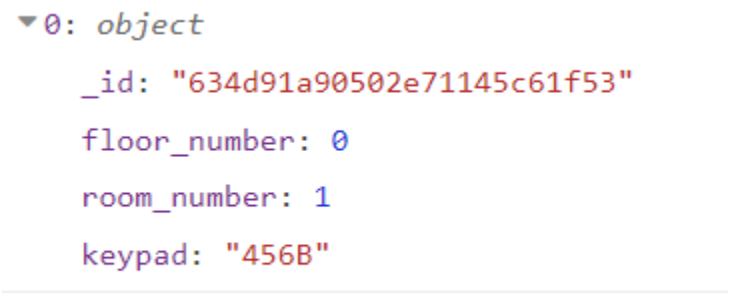


Figure 15: Sample collection of the Keypad

Web interface or SCADA pages should display the data in the database

Introduction

Under this section , the basic functionality is to implement a web interface to display the data in the database. There are 2 things that were focused on this section.

1) Implement the API to extract data from the database and to create endpoints.
2) Implement the web interface to to display the data

Basically there are 3 end points in the API. They are

- End point for room_details
- End point for past_data
- End point for statistical_data

In the web interface basically there are 3 pages. They are the Room details page, History page and the Statistical data page. In the Room details page, Current summary of the occupancy details will be shown. In the History page, past occupancy data will be shown as a graph. In the Statistical data page, statistical data like sum, average, minimum and maximum count will be shown for each floor.

Design Choices

ROOM DETAILS PAGE

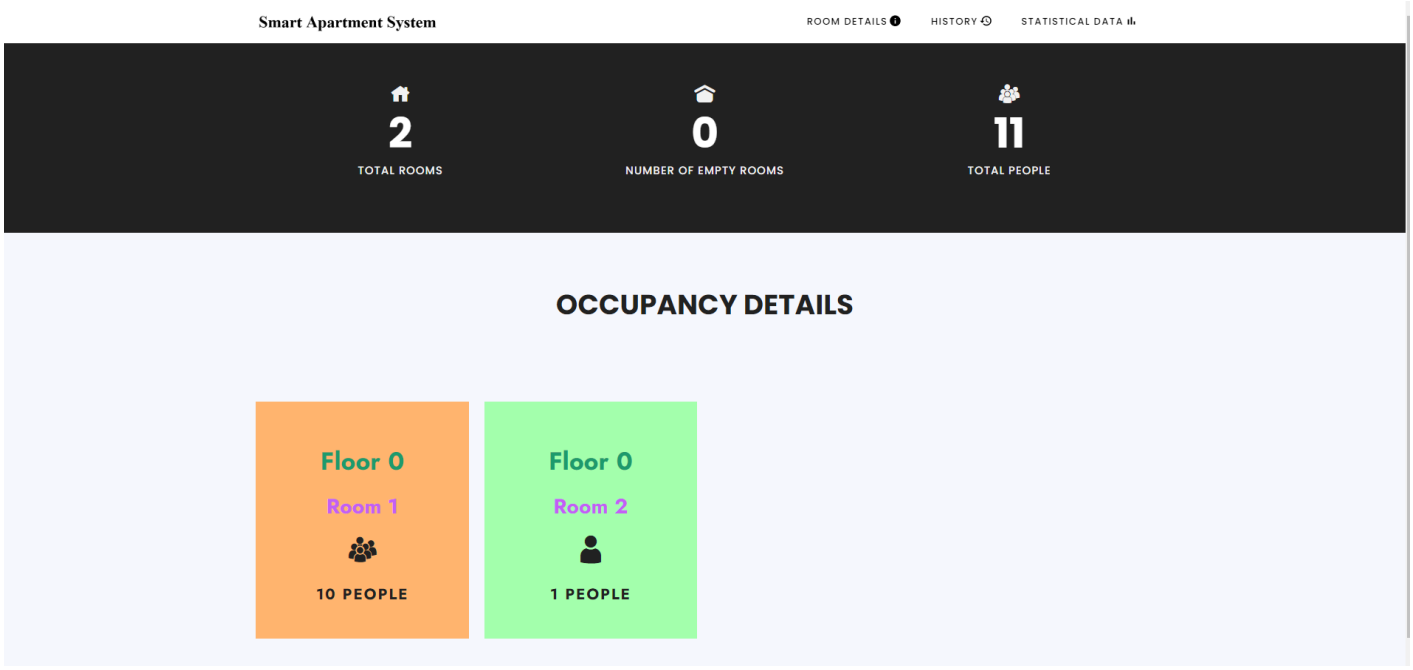


Figure 01: Room details page in the web interface

In the Room Details page total rooms, number of empty rooms and total people will be displayed as a summary of the apartment. Other than that details like Floor number, Room number, number of people of each room will be displayed as a card. Background color of the card will be changed according to the number of people in the room. Possible colors are green, yellow, orange and red.

HISTORY PAGE

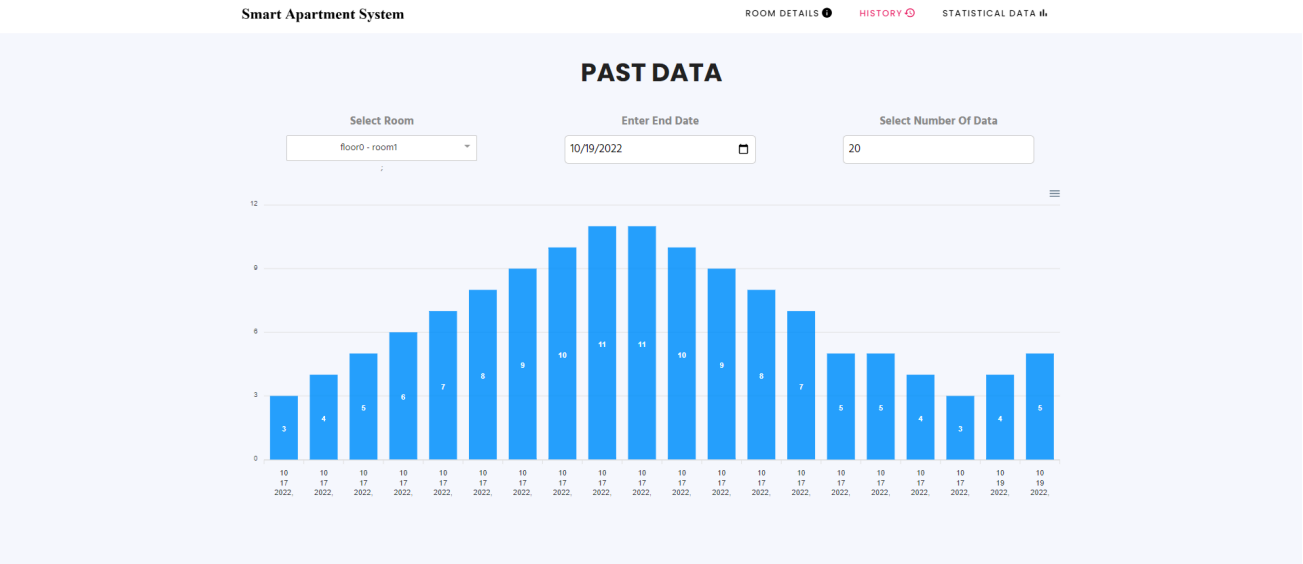


Figure 02: History page in the web interface

In the history page, number of people in each room at different times can be seen as a bar chart. Y-axis is the number of people in the selected room and x-axis is the date and time.

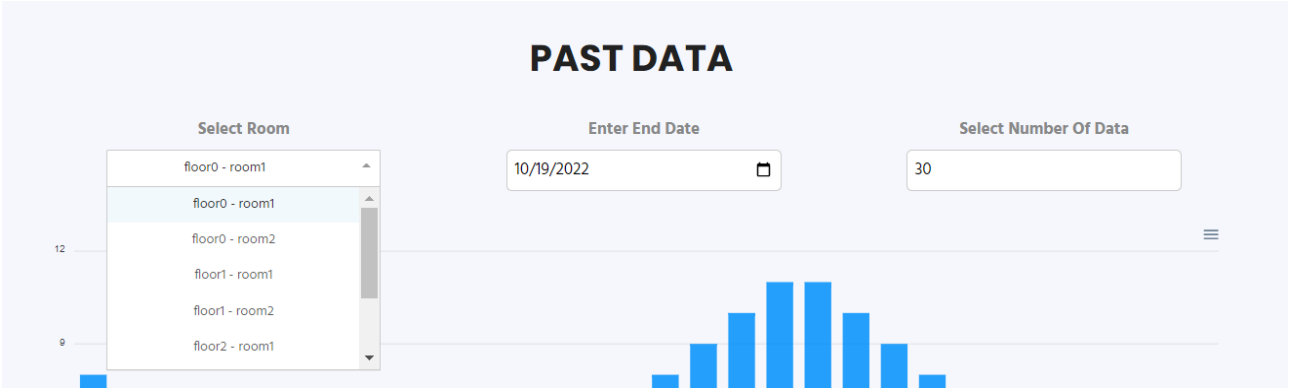
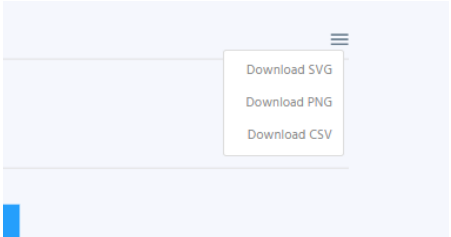


Figure 03: History page focusing the input options

User can select the room, date range and the number of bars in the chart. Under select room option, user can select a specific room. All the rooms will be shown in the dropdown menu. User can select the end date so that, bar chart will only show up to that date. And also user can select number of bars from 10 to 50.



	A	B
1	category	Number of people
2	10 17 2022	8
3	10 17 2022	7
4	10 17 2022	6
5	10 17 2022	5
6	10 17 2022	4

Figure 04: History page chart download options

Figure 05: Downloaded chart as a csv file

User can download bar chart as a png or svg file by the option shown in the web interface. The data shown in the graph can be download as a CSV file.

STATISTICAL DATA PAGE

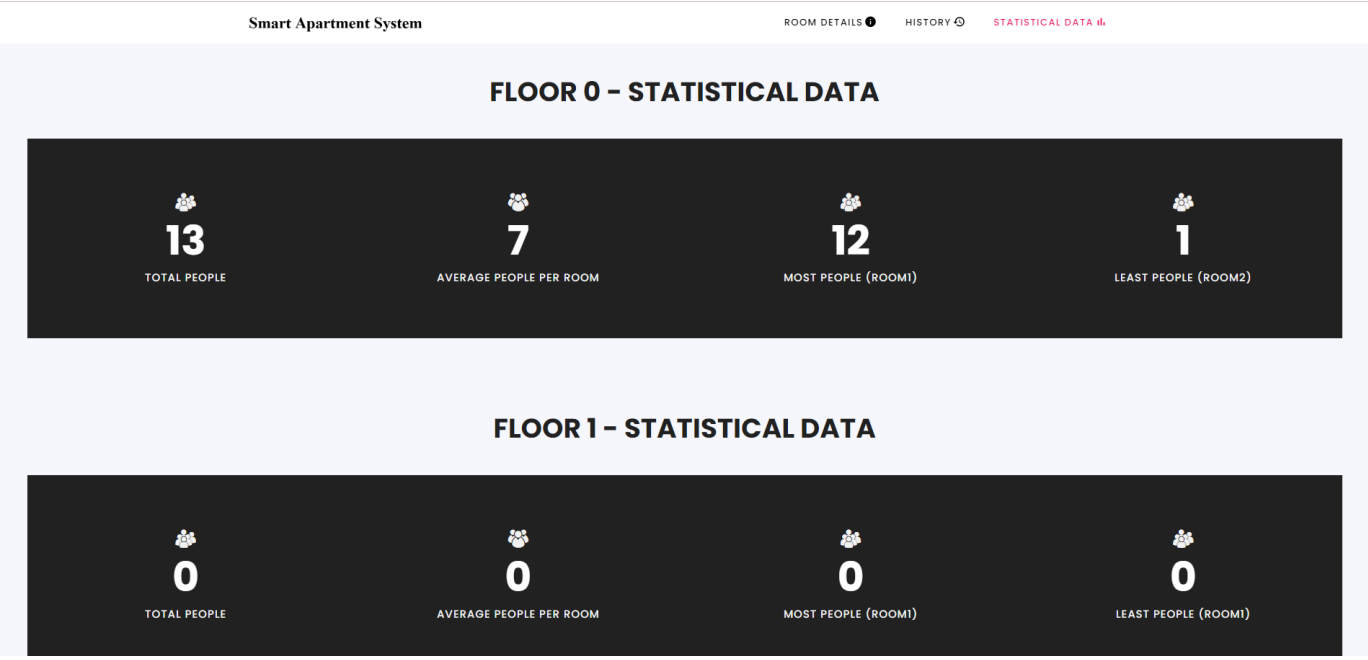


Figure 06: Statistical data page

Figure 07: Statistical data page

In the statistical data page, statistical data for each floor will be displayed. As statistical data total people in the floor, average number of people, most and least people in each floor will be shown.

Implementation

Basically I have implemented the API and the web interface. API is implemented inside NodeRed as a flow diagram.

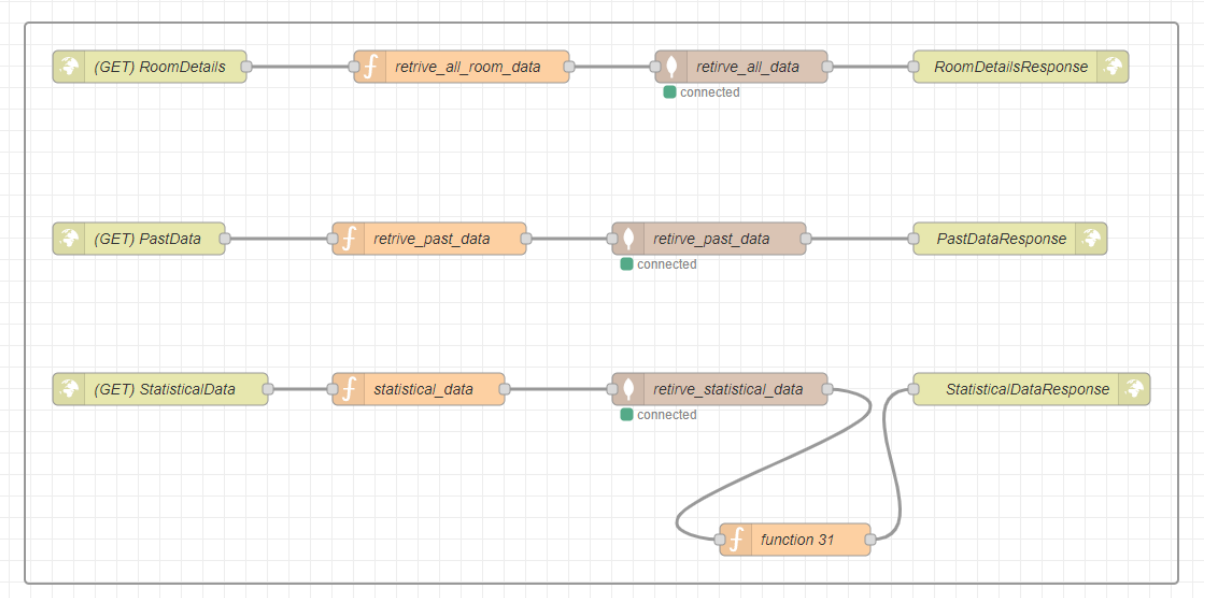


Figure 08: NodeRed API flow diagram

RoomDetails, PastData, StatisticalData are HTTP end points. There are connected to functions and those functions are connected to mongoDB queries. The result of the mongoDB queries are passed as HTTP responses.

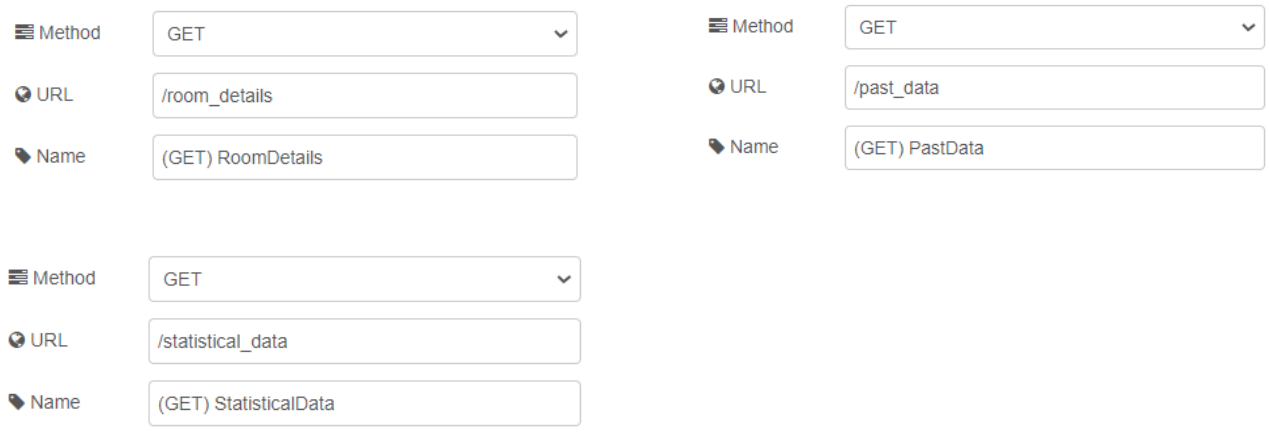


Figure 09: NodeRed API Endpoints implementation

Figure 10: MongoDB query to retrieve room data

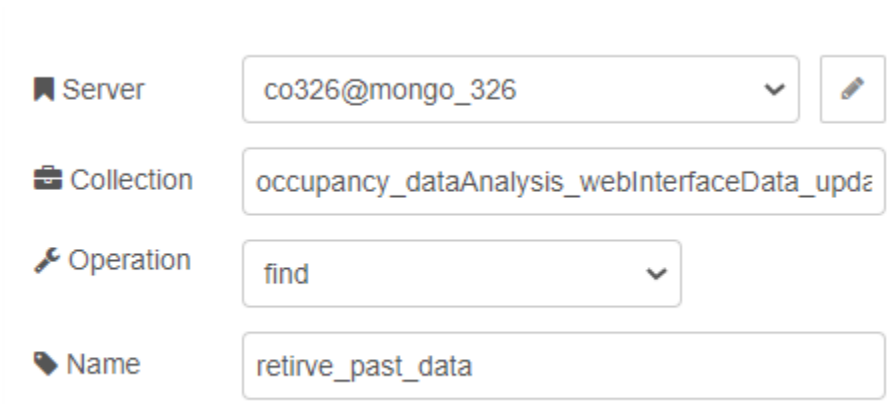


Figure 11: MongoDB query to retrieve past data

Figure 12: MongoDB query to retrieve statistical data

Web interface is implemented using ReactJS. When user load the web page or if he reload the web application it will send request to RoomDetails end point and and it will display the Room Details web page.

Figure 13: Web application menu bar

When user click any menu option (ROOM DETAILS, HISTORY OR STATISTICAL DATA) web application will internally send request to relevant endpoint to obtain the relevant data. That data will be processed and displayed in the web application.

Tests and results

The implemented API is tested using Postman. I have sent request to each end point in the NodeRed and obtain response from Postman to make sure API is working properly.

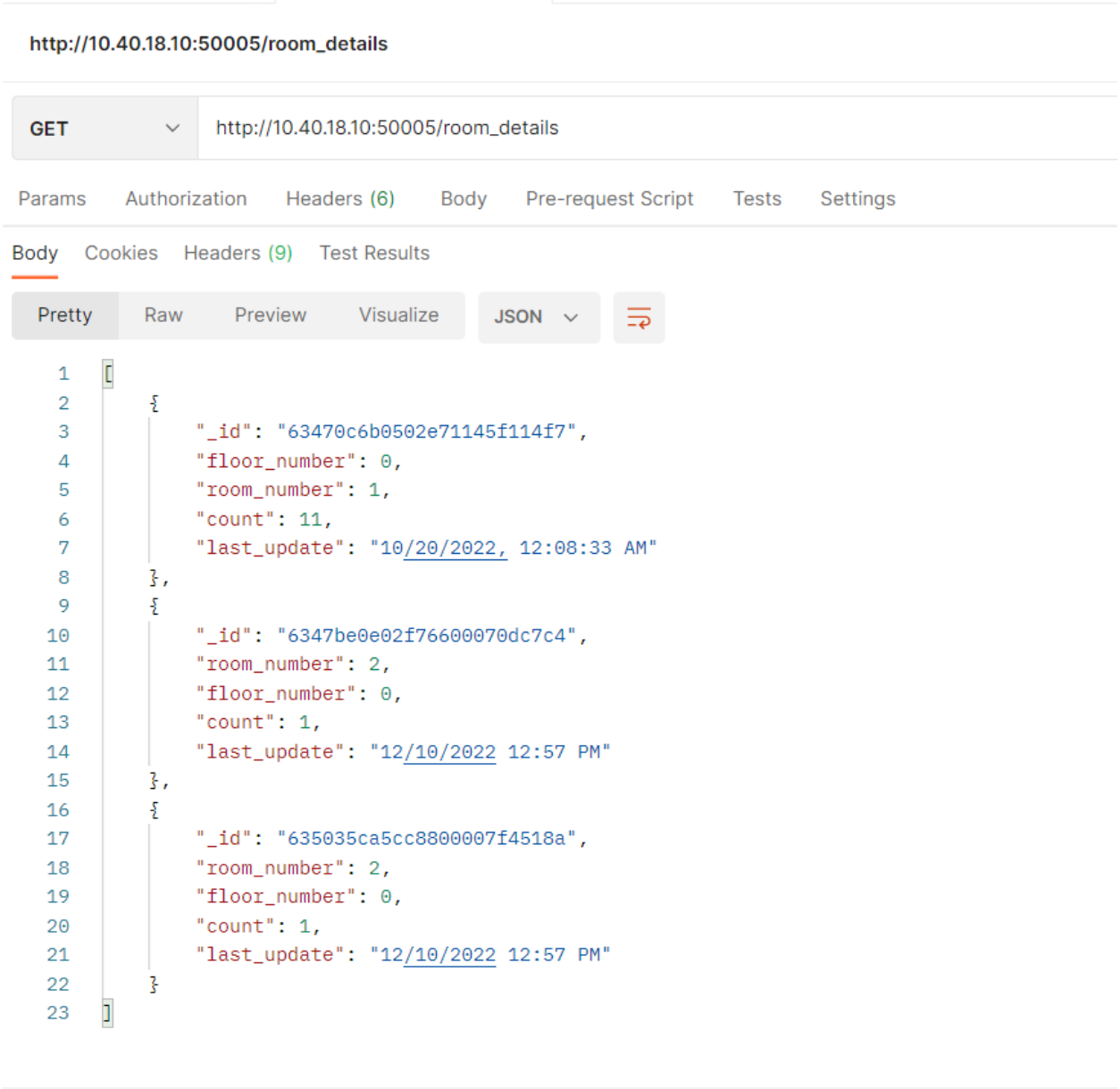


Figure 14: Testing room_details endpoint using Postman

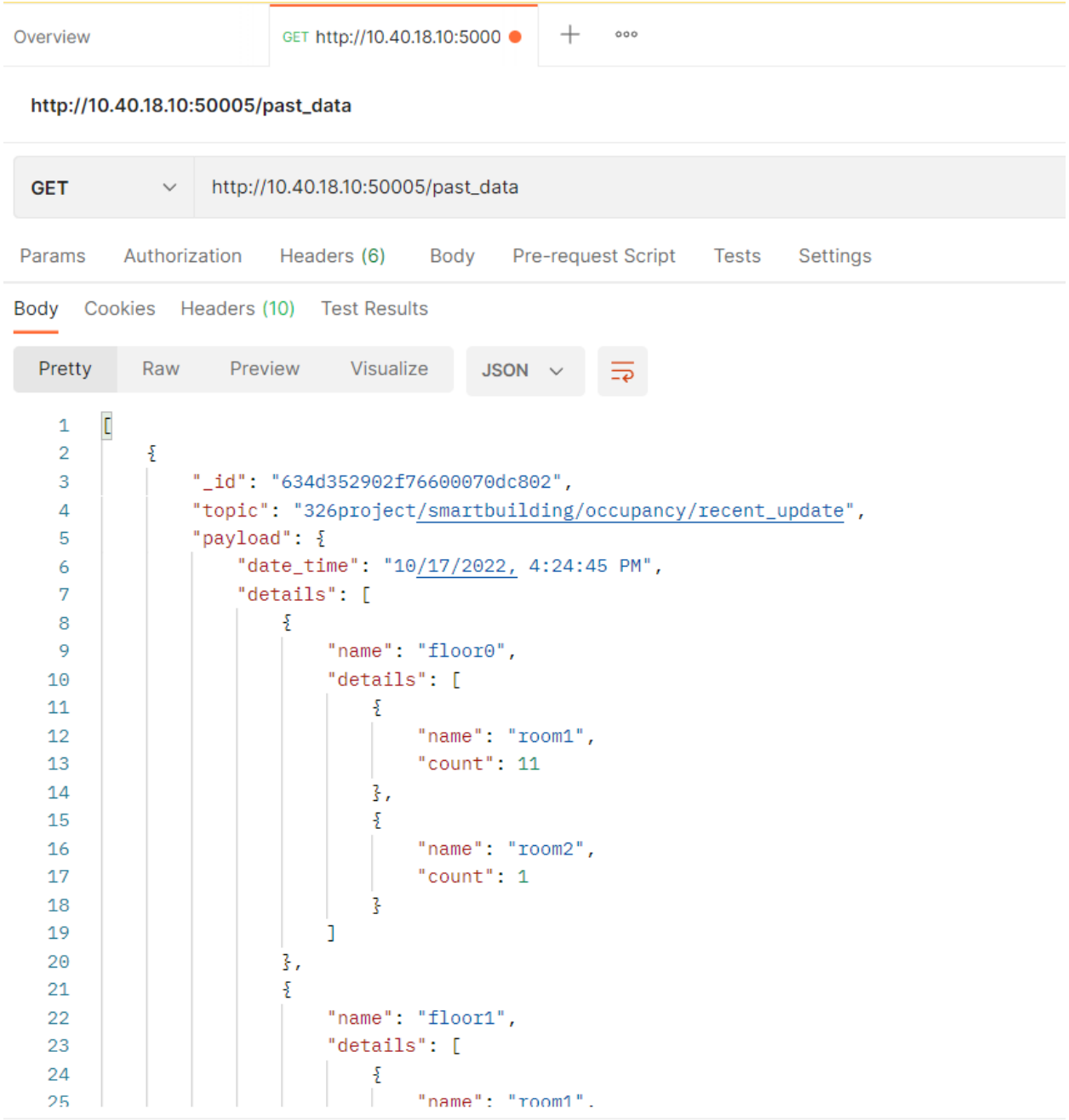


Figure 15: Testing past_data endpoint using Postman

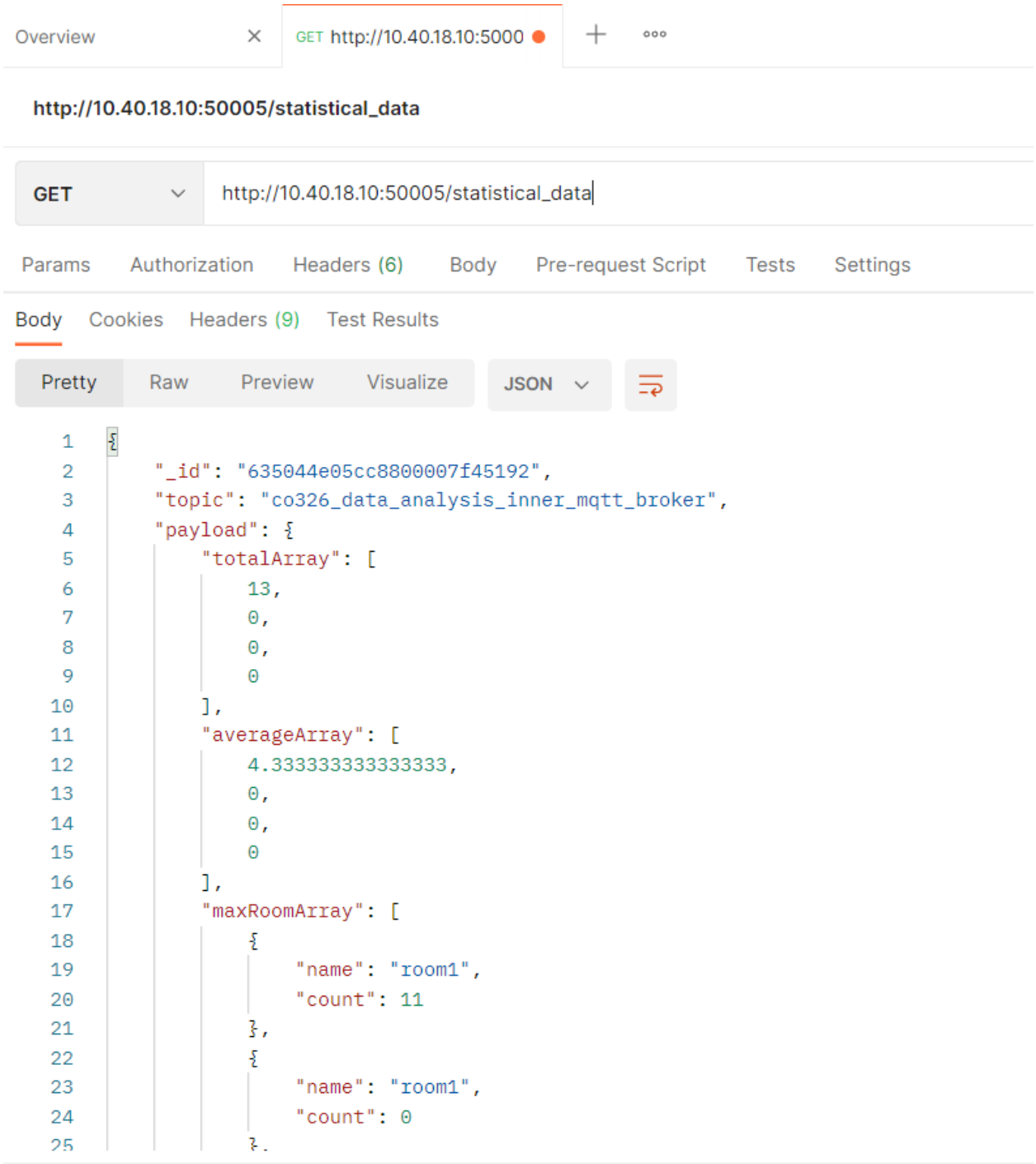


Figure 16: Testing statistical_data endpoint using Postman

Data analytics: Prediction, Optimization,Correlation

Main Features

Here data analysis and optimization parts were considered. When it comes to the predictions, since it is related to Machine learning aspects and those things are out of this course, prediction aspect is not much covered. But also the research related to that part was also done.

There were two main aspects that were focused here.

1. **Creating a dataset of the past/historical data records of the system**.

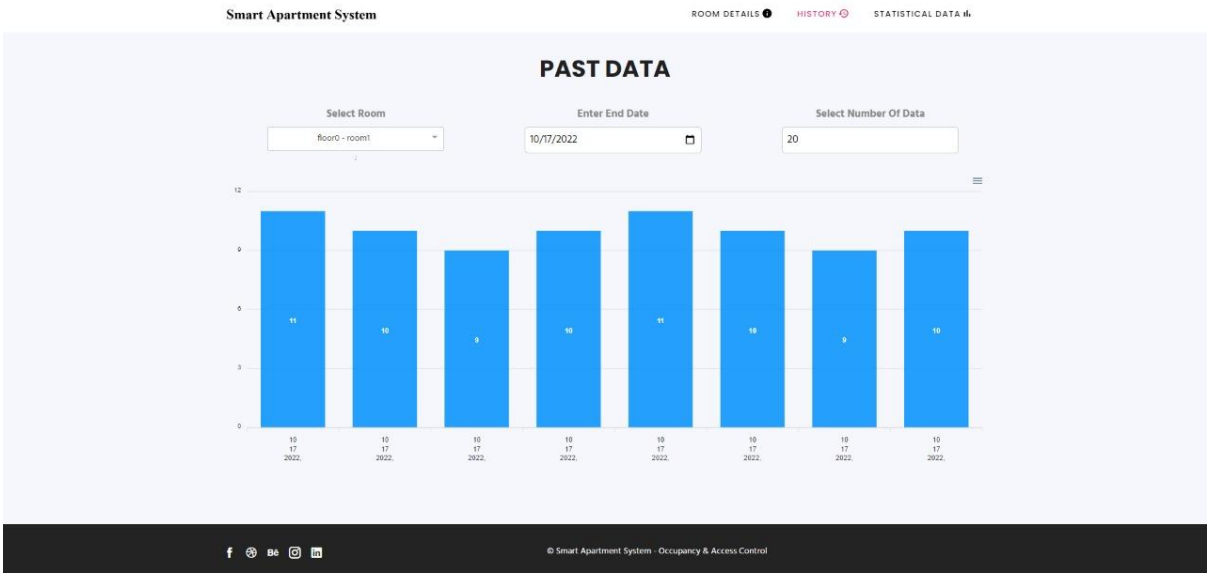
   - Here the number of people related in each room in each floor was recorded at a given time.

   - At each time the instance below the object was recorded.

```
{
    "date_time": "2022/10/22 22:10:15",
    "details": [
        {
            "name": "floor1111",
            "details": [
                {
                    "name": "room1",
                    "count": 5
                },
                {
                    "name": "room2",
                    "count": 5
                }
            ]
        },
        {
            "name": "floor2",
            "details": [
                {
                    "name": "room1",
                    "count": 8
                },
```

```
                    {
                        "name": "room2",
                        "count": 3
                    }
                ]
            }
        ]
}
```


- At each instance , this object was recorded with the time. Using this we can track the number of people in each room on each floor. Then after saving these data objects in the database, those data was used for displaying the past/history data in the web interface.



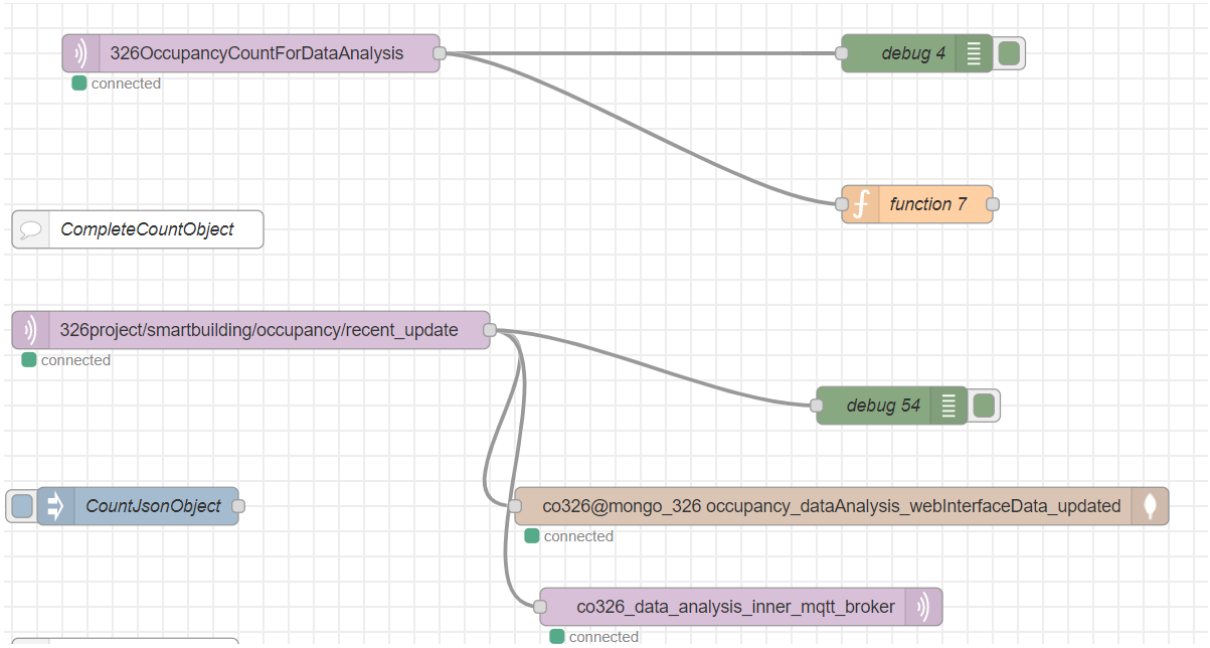
- This graph shows the number of people counted in each room on each floor at different time intervals. Also the room and the floor can be changed using the drop down menu in the website.

- This data is very useful to identify the hours that are congested due to the high number of people in the room. Also using this dataset, we can train a machine learning model to identify the most congested hours of a room and lowest congested hours of a room.

- For when the room is most congested, we can redirect those people to a lowest congested room using the machine learning model and AI algorithms. This can be considered as a huge optimization of the system.

- All those are possible due to the creation of a dataset.

- The implementation related to this feature was implemented using MQTT and MongoDB Database Instance on the node red server. Below is the implementation flow of this feature.



-

-First of all , the instantaneous data published to a topic called “326project/smartbuilding/occupancy/recent_update” and that topic was subscribed. Then those data will be send to a database collection called, “occupancy_dataAnalysis_webInterfaceData_updated”.

-Also those data is also published to another MQTT topic for data analysis feature(it is explained later). Then that data will be fetched from the database and sent to API endpoint for displaying the information on the web interface.

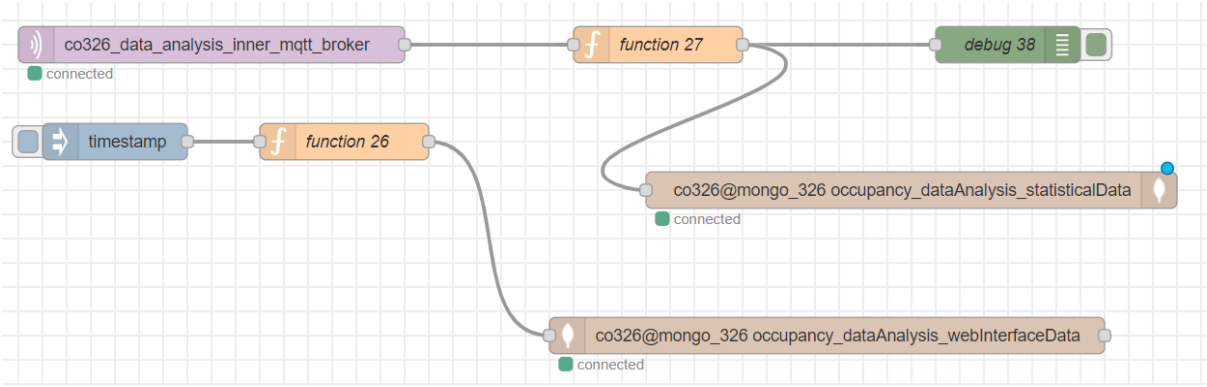
2. Data Analysis of People in each Room

- Here data analysis of the people count was done. There following aspects were calculated.

    1. Total number of people in the whole floor at the moment
    2. Average number of people in a room given a floor at the moment
    3. Room which has the maximum number of people in a floor at the moment.
    4. Room which has the minimum number of people in a floor at the moment.

- All of these were calculated using the instantaneous data that was published to the MQTT topic “co326_data_analysis_inner_mqtt_broker”.

- Following is the data flow related to this feature.



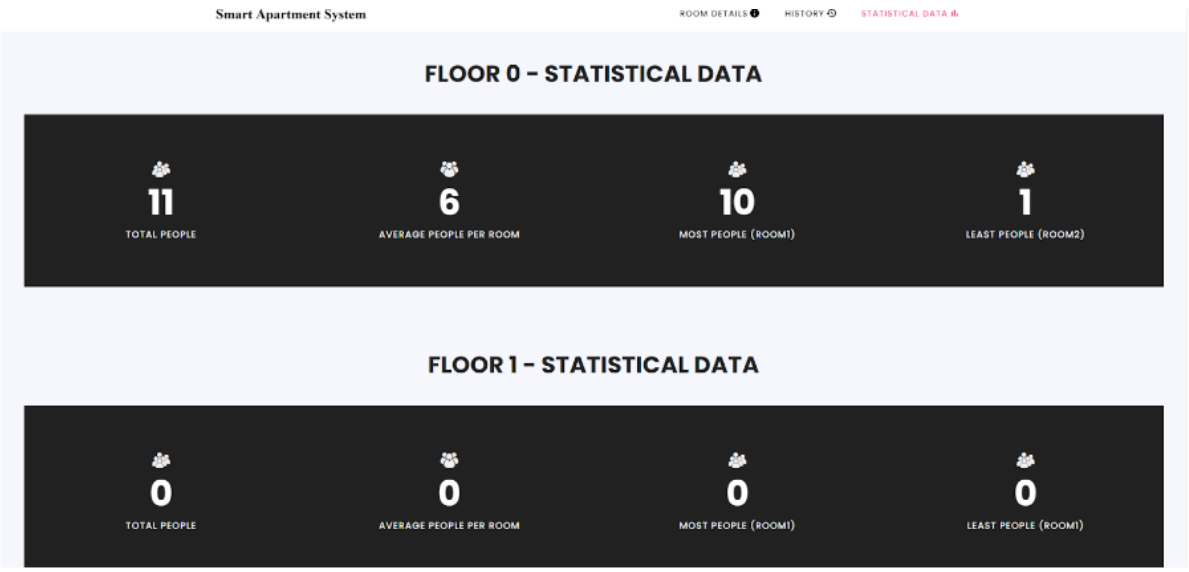
  - Here first of all the instantaneous data was subscribed from the mqtt topic “co326_data_analysis_inner_mqtt_broker”. Then those data will be analysed and calculated necessary parameters using the function widget it is connected to.

  - The data analysis part was done using the below code segment.

```
const totalArray = [];
const averageArray = [];
const maxRoomArray = [];
const minRoomArray = [];

let floor;
let rooms;
let room;
let totalCountOnFloor;
let maxIndex = 0;
let minIndex = 0;

let lastObj = msg.payload.details;

//calculate the average count of each room
for(let i=0;i<lastObj.length;i++){

    floor = lastObj[i].details;

    totalCountOnFloor = 0;
    maxIndex = 0;
    minIndex = 0;
```

```
    for(let j=0;j<floor.length;j++){

        room = floor[j];
        totalCountOnFloor += room.count;
        if(room.count>floor[maxIndex].count){
            maxIndex = j;
        }
        if (room.count < floor[minIndex].count) {
            minIndex = j;
        }

    }

    totalArray.push(totalCountOnFloor);
    averageArray.push(totalCountOnFloor/floor.length);
    maxRoomArray.push(floor[maxIndex]);
    minRoomArray.push(floor[minIndex]);

}

msg.payload = {
    "totalArray":totalArray,
    "averageArray" : averageArray,
    "maxRoomArray" : maxRoomArray,
    "minRoomArray" : minRoomArray
    }

return msg;
```

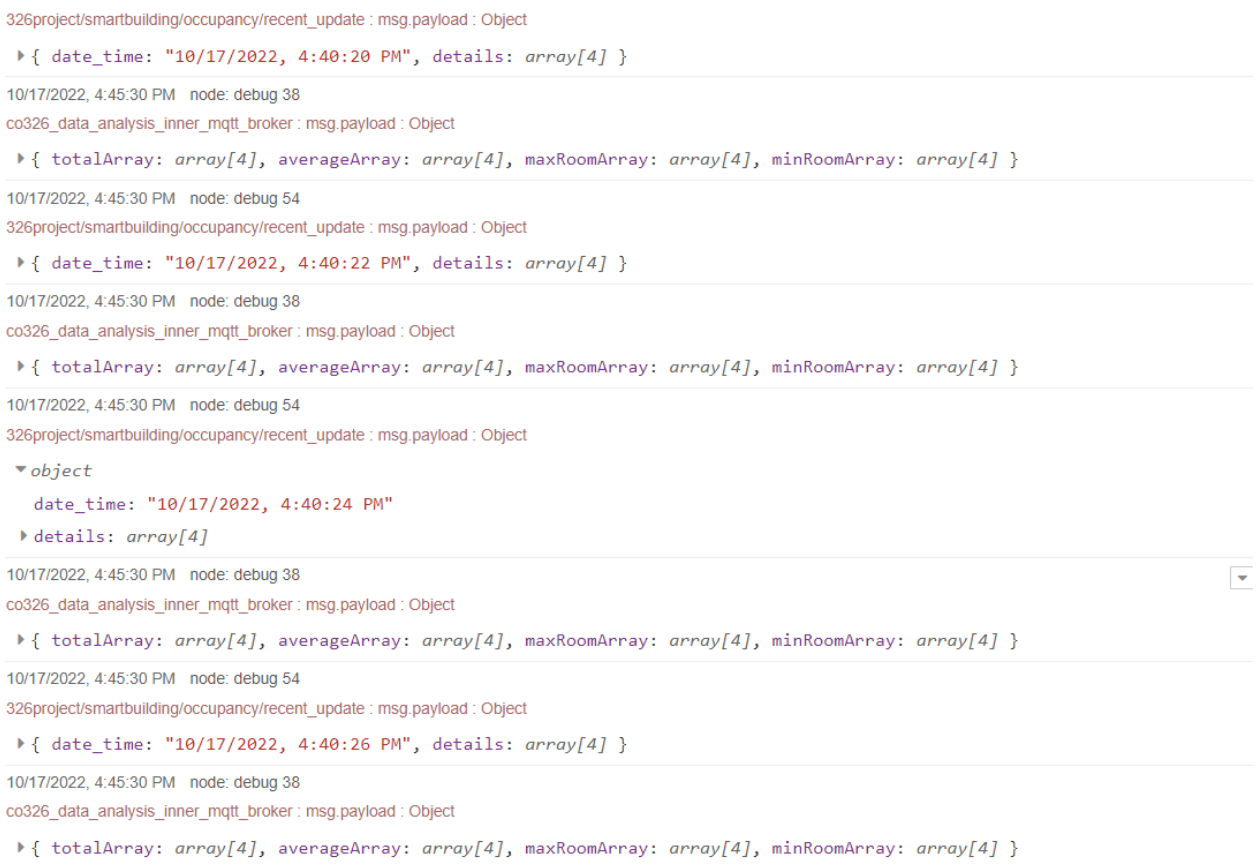
- Using this function we can get the above parameters discussed above. Then those data will be stored in the database, and those stored data will be fetched from an api endpoint and will be used to show details on the web interface. Below shown , that data is displayed at the web interface.



- This data will be used to optimize high people count in a room. For example : We can detect the maximum people room in the floor as well as minimum people count room given a time instance. Then we can move people from high congested rooms to low congested rooms.

Testing

- Testing was done using the Debug widget in the Node red instance.
- Using that tool, it was able to resolve all the bugs which were related to end to end communication from the data analysis MQTT topic to the web interface.
- Below are some of the debugging results.



Future Improvements

- There can be probable two future implementations related to data analysis and prediction aspects.

1. Machine Learning Model

- We can implement a machine learning model to identify the usual congestion pattern in different rooms and we can use different solutions to reduce that congestion.

- For example : We can identify the most congested room at a given time instance and also the least congested room at that time. Then we can redirect the people from the most congested room to the least one. These all can be automated with the help of the ML algorithm.

- To train this ML model , we can use the data set we have created as the first feature of this section

2. Implementing a CNN(Convolution Neural Network)

- This is useful for identifying a person who is entering the room.

- This is in addition to the RFID authentication method which is already implemented.

- The goal of this is to increase the security of the system by adding an additional layer of security to the system.

- When an unauthorized person is trying to enter a room, that is identified by the CNN model and then that will be informed to the security guards of the bundling and also that is shown in the web interface as well as Scada interface.