Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

Projekt Dokumentation

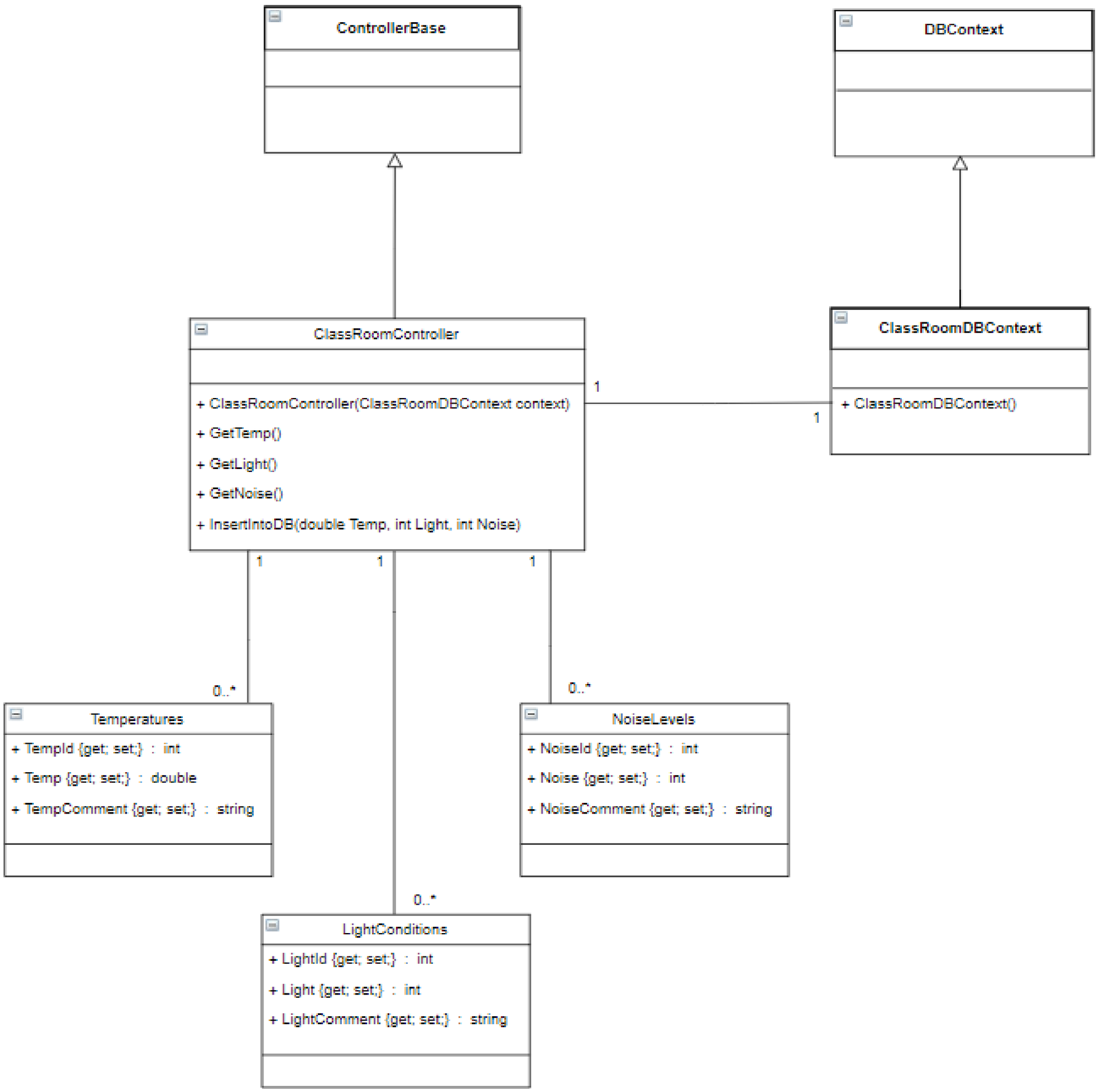
Udarbejdet af: Gruppe 1, 4108h3dakp

Indholdsfortegnelse:

Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

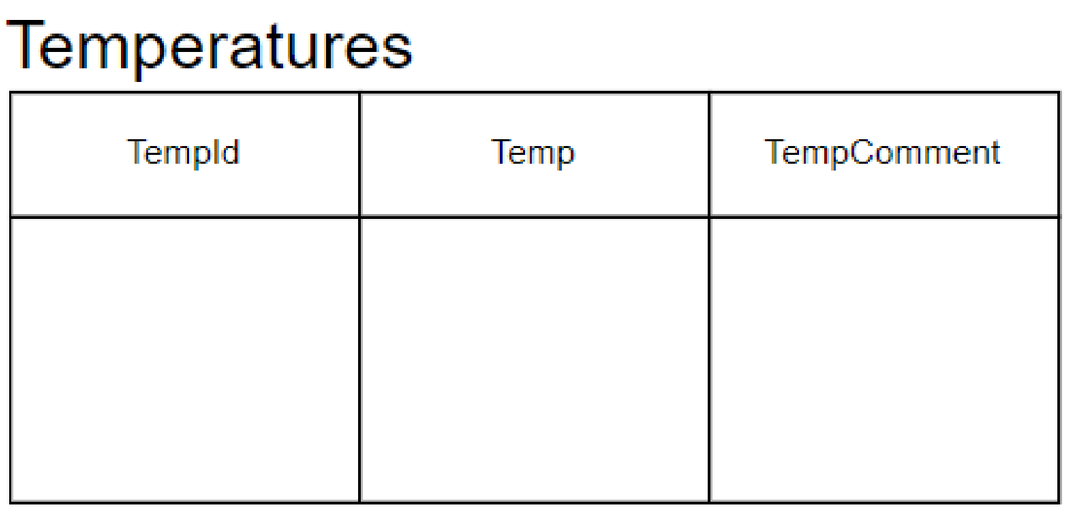
Figur 1 illustrerer projektets Klassediagram:



Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

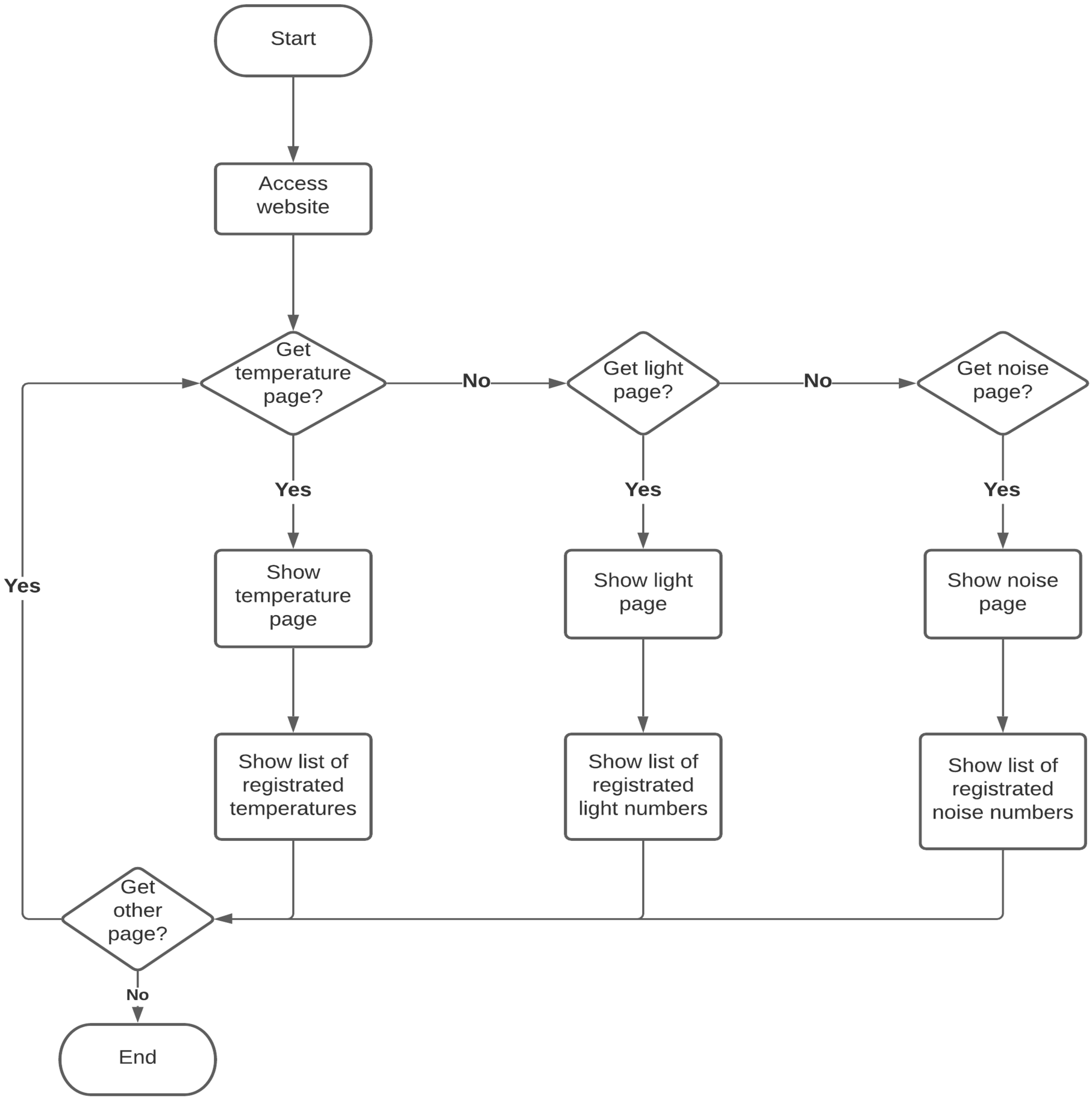
Figur 2 illustrerer projektets RDS:



Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

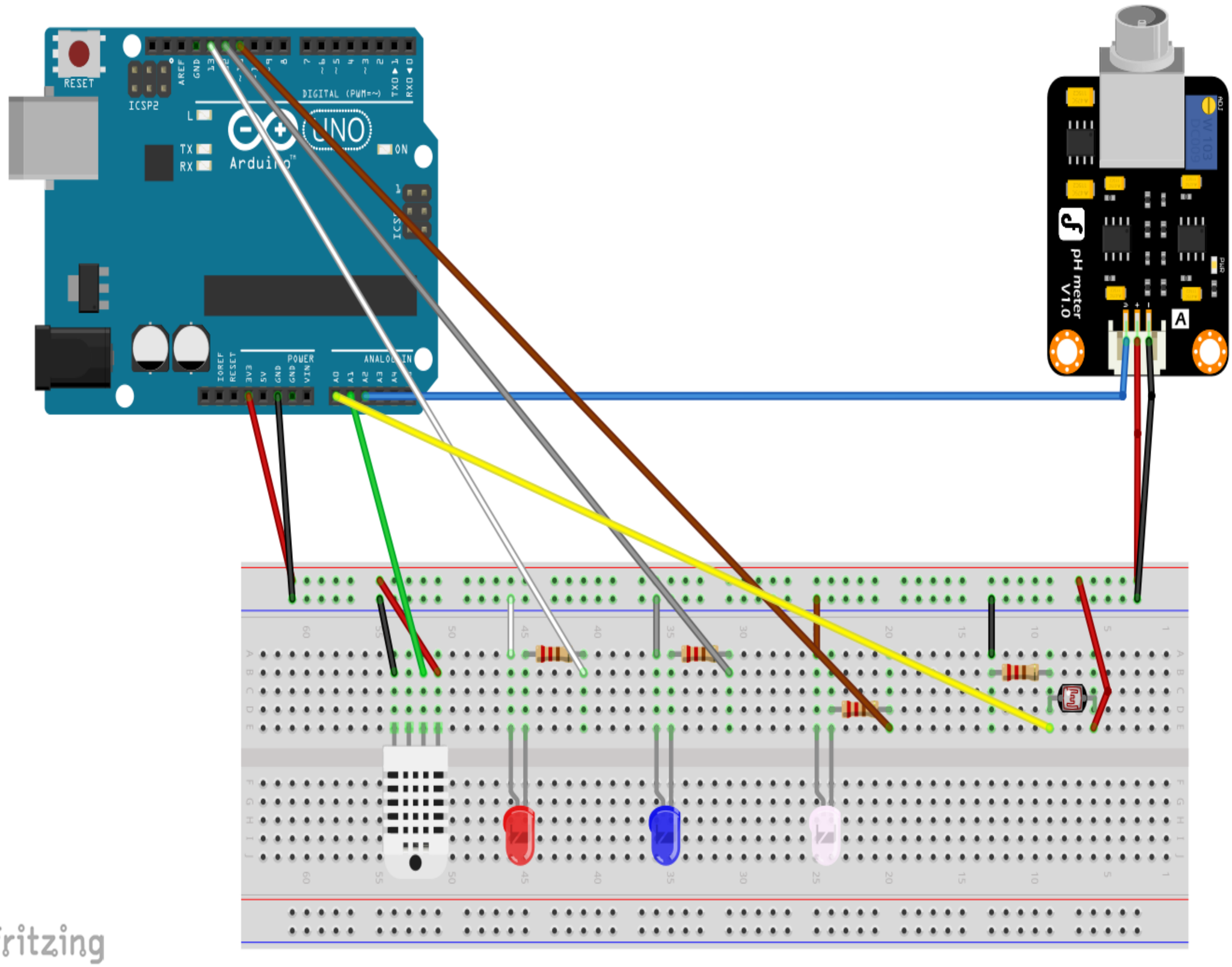
Figur 3 illustrerer projektets Flowchart:



Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

Figur 4 illustrerer projektets Arduino design:



Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

Figur 5 illustrerer projektets Arduino kode del 1 af 4:

```
#include <SPI.h>
#include <Ethernet.h>
#include <HttpClient.h>

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };


// Set the static IP address to use if the DHCP fails to assign!
// Arduino IP!
IPAddress ip(192, 168, 1, 120);
// PC IP!
IPAddress server(192, 168, 1, 121);

// Initialize the Ethernet client library!
EthernetClient client;

// Variables to measure the speed!
unsigned long beginMicros, endMicros;
unsigned long byteCount = 0;

// set to false for better speed measurement!
bool printWebData = true;

// Create sensors values!
int ThermistorPin = 0;
int Vo;
float R1 = 10000;
float logR2, R2, T;
float c1 = 1.009249522e-03, c2 = 2.378405444e-04, c3 = 2.019202697e-07;
int val1;
int val2;


void setup() {

 // You can use Ethernet.init(pin) to configure the CS pin!
 Ethernet.init(10);  // Most Arduino shields

   // initialize digital pin LED_BUILTIN as an output!
  pinMode(LED_BUILTIN, OUTPUT);

 // Open serial communications and wait for port to open!
 Serial.begin(9600);

 while (!Serial) {
   // wait for serial port to connect. Needed for native USB port only!
   ;
 }
```

Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

Figur 6 illustrerer projektets Arduino kode del 2 af 4:

```
// start the Ethernet connection!
 Serial.println("Initialize Ethernet with DHCP:");

 if (Ethernet.begin(mac) == 0) {
  Serial.println("Failed to configure Ethernet using DHCP");

  // Check for Ethernet hardware present!
  if (Ethernet.hardwareStatus() == EthernetNoHardware) {

   Serial.println("Ethernet shield was not found.  Sorry, can't run without hardware. :(");

   while (true) {

    // do nothing, no point running without Ethernet hardware!
    delay(1);
   }
  }

  if (Ethernet.linkStatus() == LinkOFF) {
   Serial.println("Ethernet cable is not connected.");
  }

  // try to congifure using IP address instead of DHCP!
  Ethernet.begin(mac, ip);
 } else {
  Serial.print("  DHCP assigned IP ");
  Serial.println(Ethernet.localIP());
 }
 // give the Ethernet shield a second to initialize!
 delay(1000);
 Serial.print("connecting to ");
 Serial.print(server);
 Serial.println("...");

 // Calculate temperature and statements!
 Vo = analogRead(A1);
 R2 = R1 * (1023.0 / (float)Vo - 1.0);
 logR2 = log(R2);
 T = (1.0 / (c1 + c2*logR2 + c3*logR2*logR2*logR2));
 T = T - 293,15;
 T = (T * 9.0)/ 5.0 + 32.0;
 T = (T - 32)/1.8;

  if ( T < 18 )
 {
  // turn the temperture LED off!
  digitalWrite(13, LOW);
 }
 else
 {
```

Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

Figur 7 illustrerer projektets Arduino kode del 3 af 4:

```
// turn the temperture LED on!
   digitalWrite(13, HIGH);
 }


  if ( val1 < 10 )
  {
   // turn the noise-level LED off!
   digitalWrite(12, LOW);
  }
  else
  {
   // turn the noise-level LED on!
   digitalWrite(12, HIGH);
  }

  if ( val2 < 50 )
  {
   // turn the light-condition LED off!
    digitalWrite(11, LOW);
  }
  else
  {
   // turn the light-condition LED on!
    digitalWrite(11, HIGH);
  }

  // Output in monitor!
  Serial.print("Temperature: ");
  Serial.println(T);
  delay(2000);

  val1 = analogRead(A2);
  Serial.print("Noise-Level: ");
  Serial.println(val1);
  delay(2000);

  val2 = analogRead(A0);
  Serial.print("Light-Condition: ");
  Serial.println(val2);
  delay(2000);

  String PostData ="Vi sender data";

  // if you get a connection, report back via serial!
  if (client.connect(server, 44889)) {
   Serial.print("connected to ");
   Serial.println(client.remoteIP());
```

Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

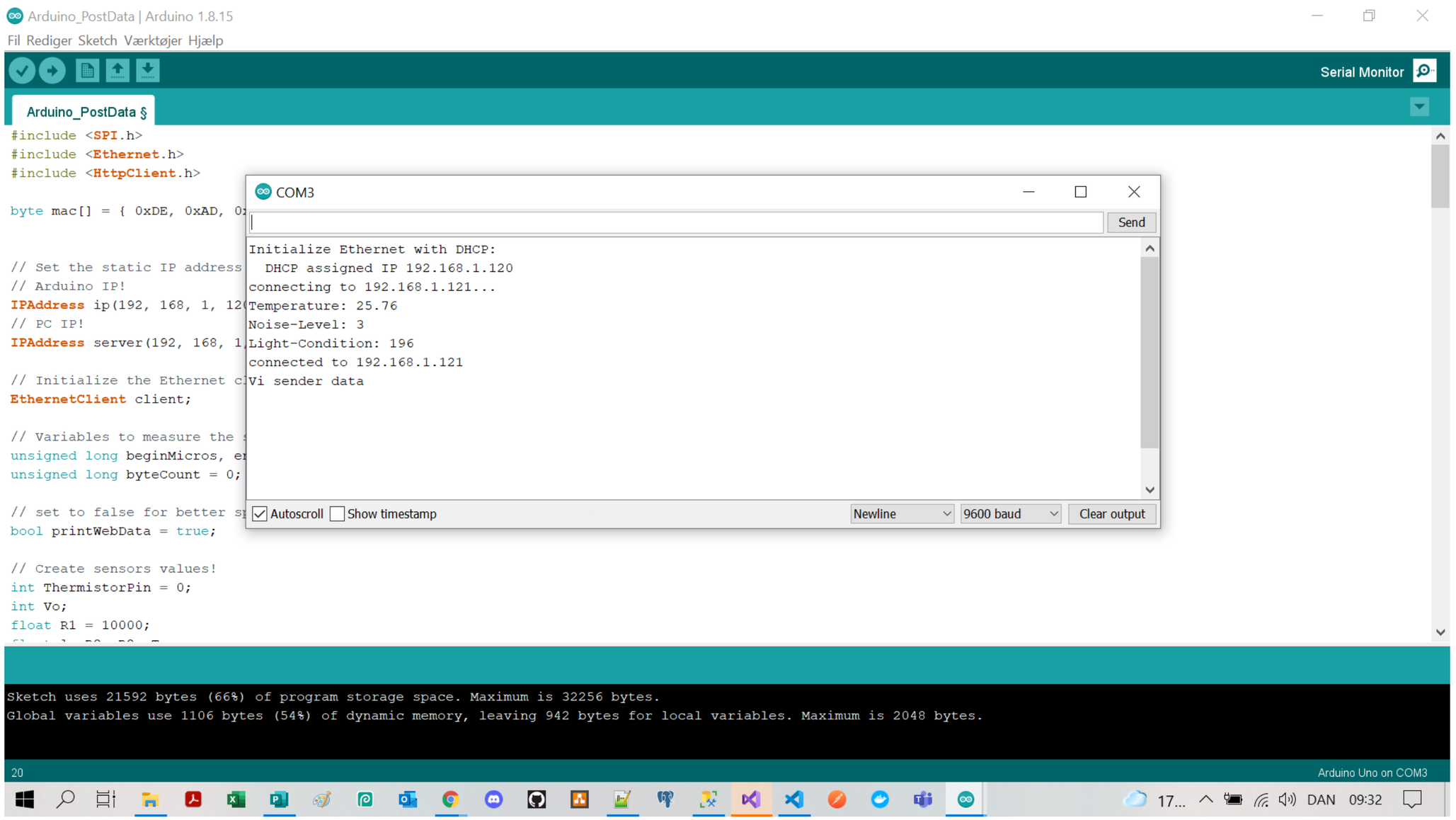
Figur 8 illustrerer projektets Arduino kode del 4 af 4:

```
  // Make a HTTP POST with values from temperature-, light- and noisesesensor request!
  client.println("POST /api/ClassRoom/AllPosts?Temp=" + String(T) + "&Light=" + String(val2) + "&Noise=" + String(val1) + " HTTP/1.1");
  client.println("HOST: 192.168.1.121:44889");
  client.println("User.Agent: Arduino/1.0");
  client.println("Connection: close");
  client.print("Content-length: ");
  client.println(PostData.length());
  client.println();
  Serial.println(PostData);
 } else {
  // if you didn't get a connection to the server!
  Serial.println("connection failed");
 }
 beginMicros = micros();
}
void loop() {
 // if there are incoming bytes available rom the server, read them and print them!
 int len = client.available();
 if (len > 0) {
  byte buffer[80];
  if (len > 80) len = 80;
  client.read(buffer, len);
  if (printWebData) {

   // show in the serial monitor (slows some boards)!
   Serial.write(buffer, len);
  }
  byteCount = byteCount + len;
 }

 // if the server's disconnected, stop the client!
 if (!client.connected()) {
  endMicros = micros();
  Serial.println();
  Serial.println("disconnecting.");
  client.stop();
  Serial.print("Received ");
  Serial.print(byteCount);
  Serial.print(" bytes in ");
  float seconds = (float)(endMicros - beginMicros) / 1000000.0;
  Serial.print(seconds, 4);
  float rate = (float)byteCount / seconds / 1000.0;
  Serial.print(", rate = ");
  Serial.print(rate);
  Serial.print(" kbytes/second");
  Serial.println();

  // do nothing forevermore!
  while (true) {
   delay(1);
  }
 }
}
```

Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

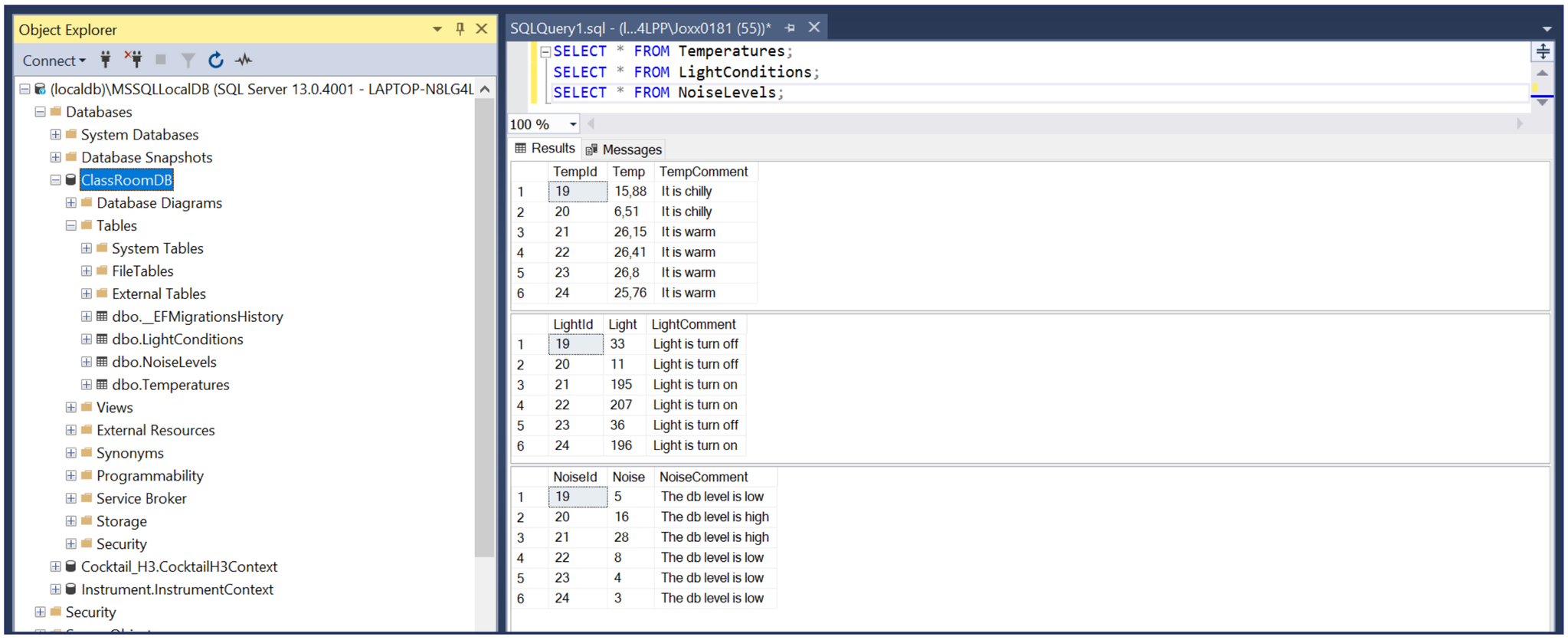
Figur 9 illustrerer Arduino til API (connecting & sending data):



Projekt Dokumentation

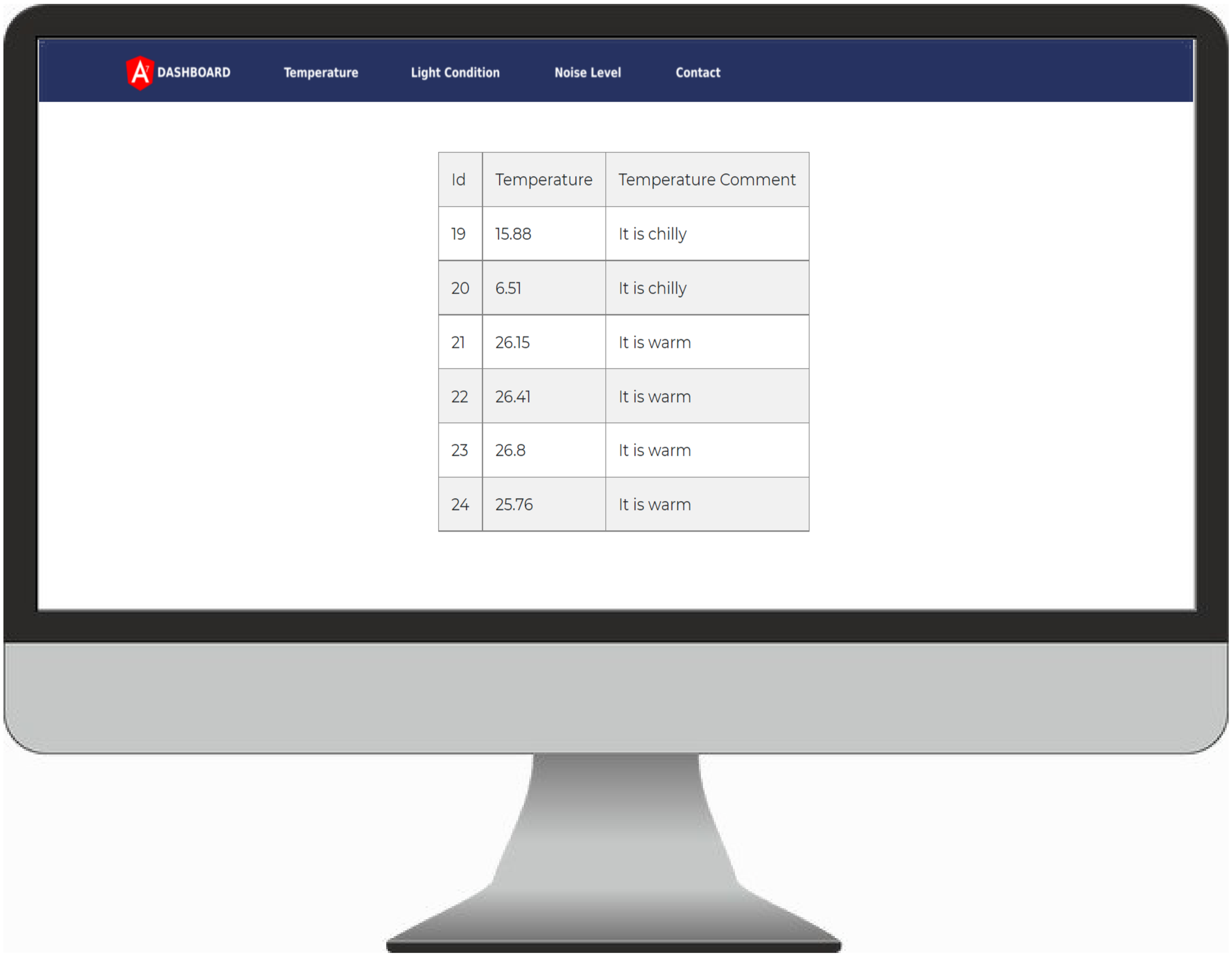
Udarbejdet af: Gruppe 1, 4108h3dakp

Figur 10 illustrerer Databasen ClassRoomDB:



Udarbejdet af: Gruppe 1, 4108h3dakp

Figur 11 illustrerer API til Angular (connecting & receive data):

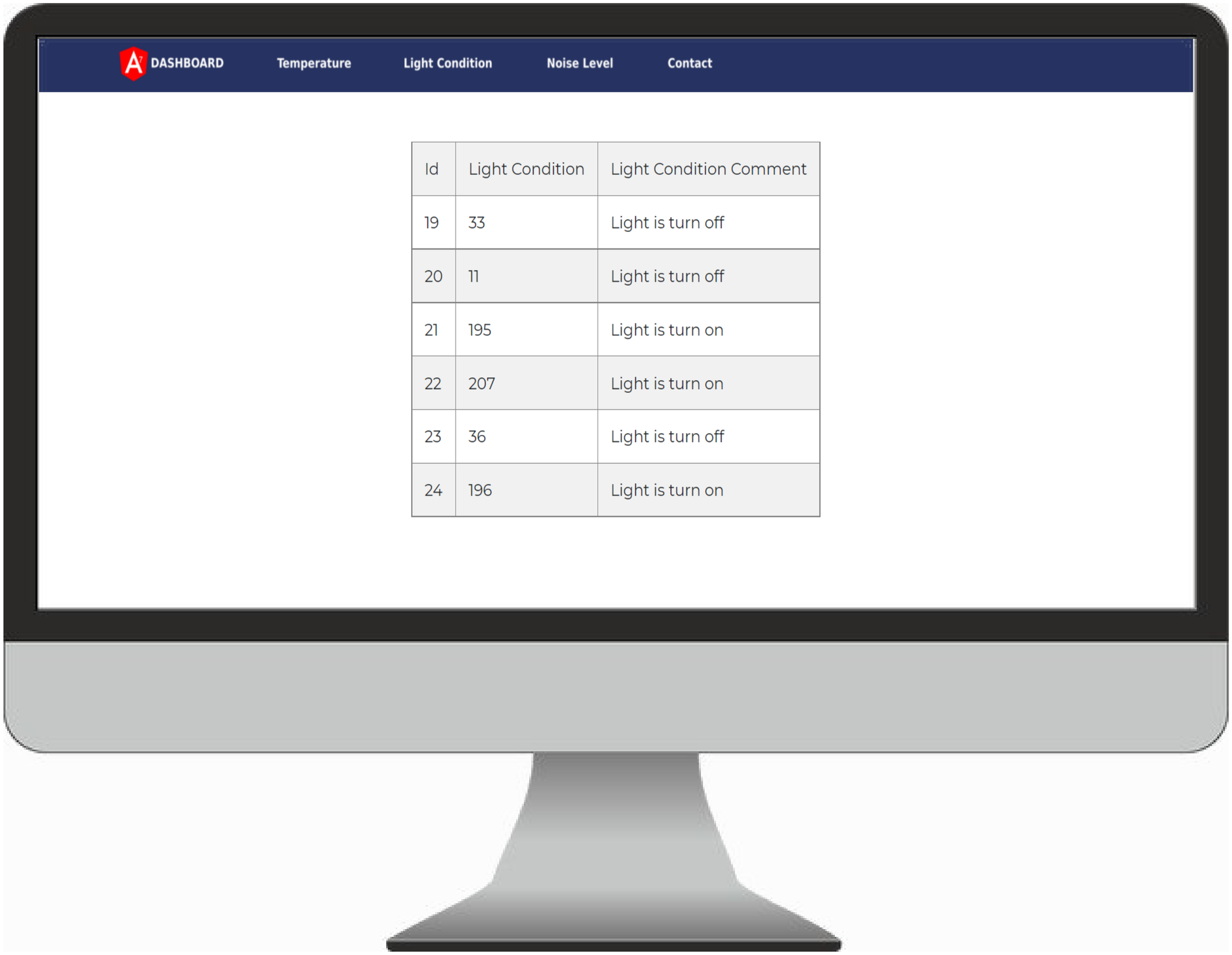


Id	Temperature	Temperature Comment
19	15.88	It is chilly
20	6.51	It is chilly
21	26.15	It is warm
22	26.41	It is warm
23	26.8	It is warm
24	25.76	It is warm

Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

Figur 12 illustrerer API til Angular (connecting & receive data):

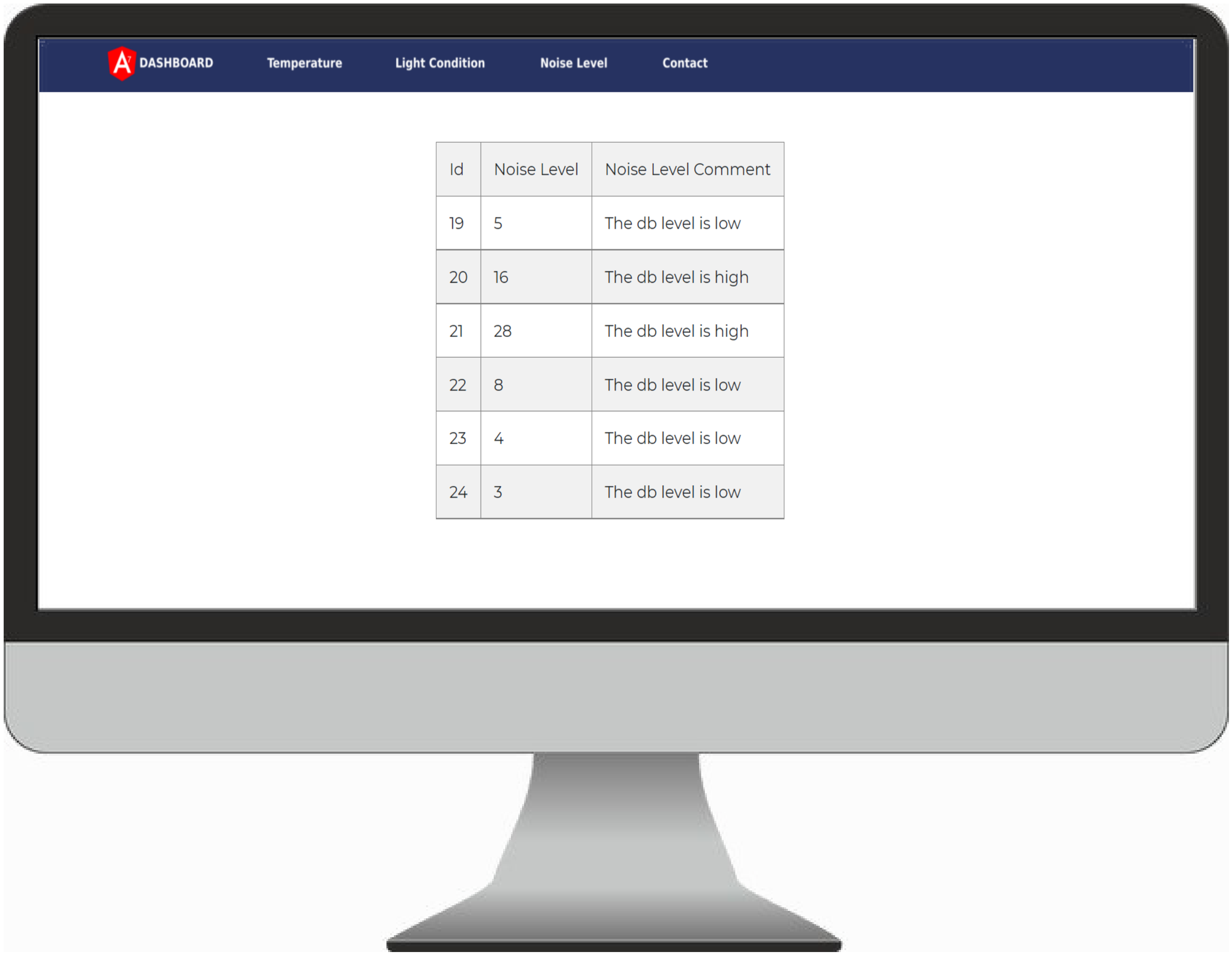


Id	Light Condition	Light Condition Comment
19	33	Light is turn off
20	11	Light is turn off
21	195	Light is turn on
22	207	Light is turn on
23	36	Light is turn off
24	196	Light is turn on

Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

Figur 13 illustrerer API til Angular (connecting & receive data):



Id	Noise Level	Noise Level Comment
19	5	The db level is low
20	16	The db level is high
21	28	The db level is high
22	8	The db level is low
23	4	The db level is low
24	3	The db level is low