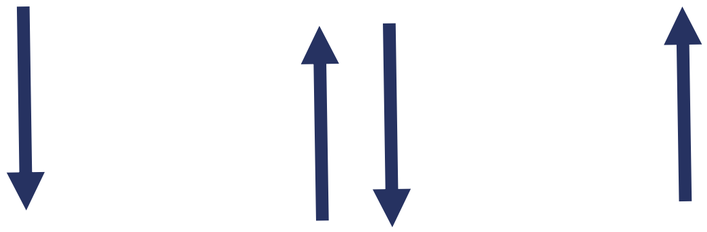
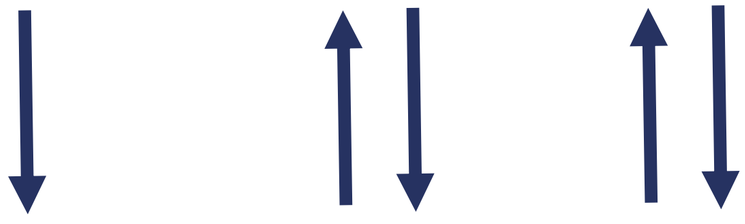
Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp





classROOMprojectH3@gmail.com

Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

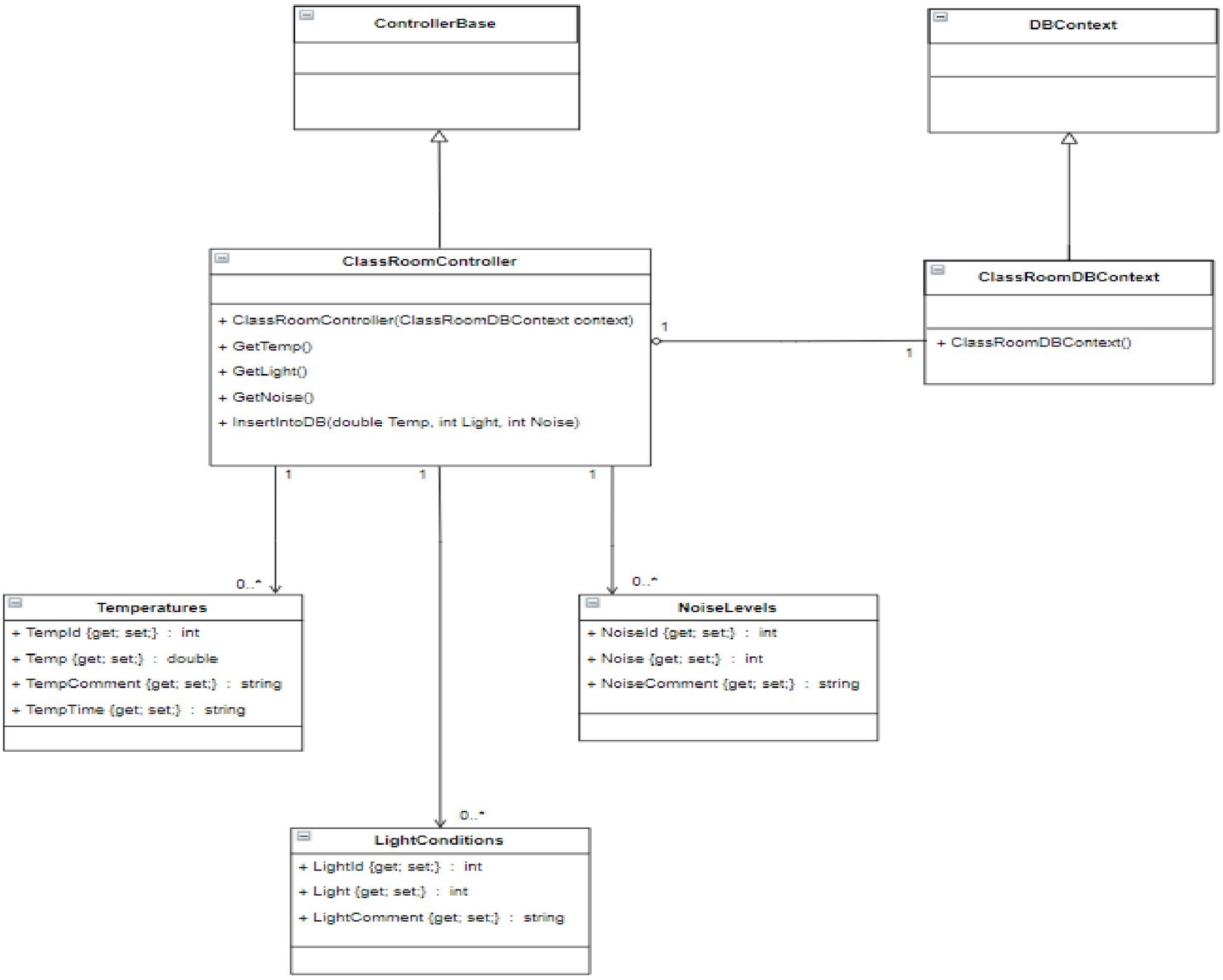
Indholdsfortegnelse:

Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

Figur 1 illustrerer projektets Klassediagram for ClassRoomAPI:

Dette Api modtager data fra Arduino via httppost forespørgslens InsertIntoDB metode som videresendes til databasen Class-RoomDB og samme Api henter fra databasen via de tre httpget forespørgsler for at sende data til Angular. Alle private attributter ligger gemt i get / set.

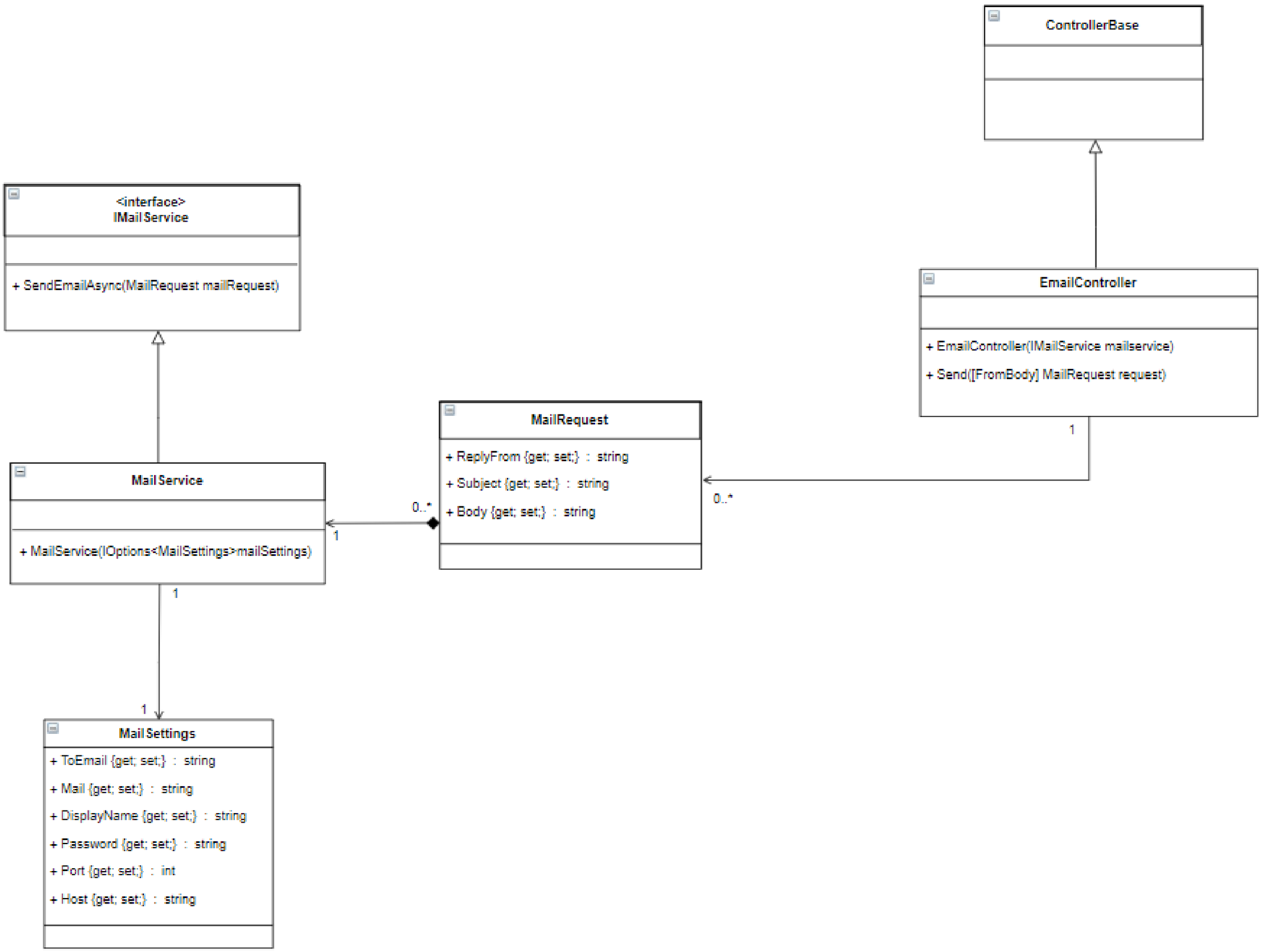


Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

Figur 2 illustrerer projektets Klassediagram for CRMailAPI:

Dette Api modtager data fra Angular´s kontaktformular via httppost forespørgslens Send metode som sender en email til projektets mailadresse classROOMprojectH3@gmail.com med info om brugers input. Alle private attributter ligger gemt i get / set.

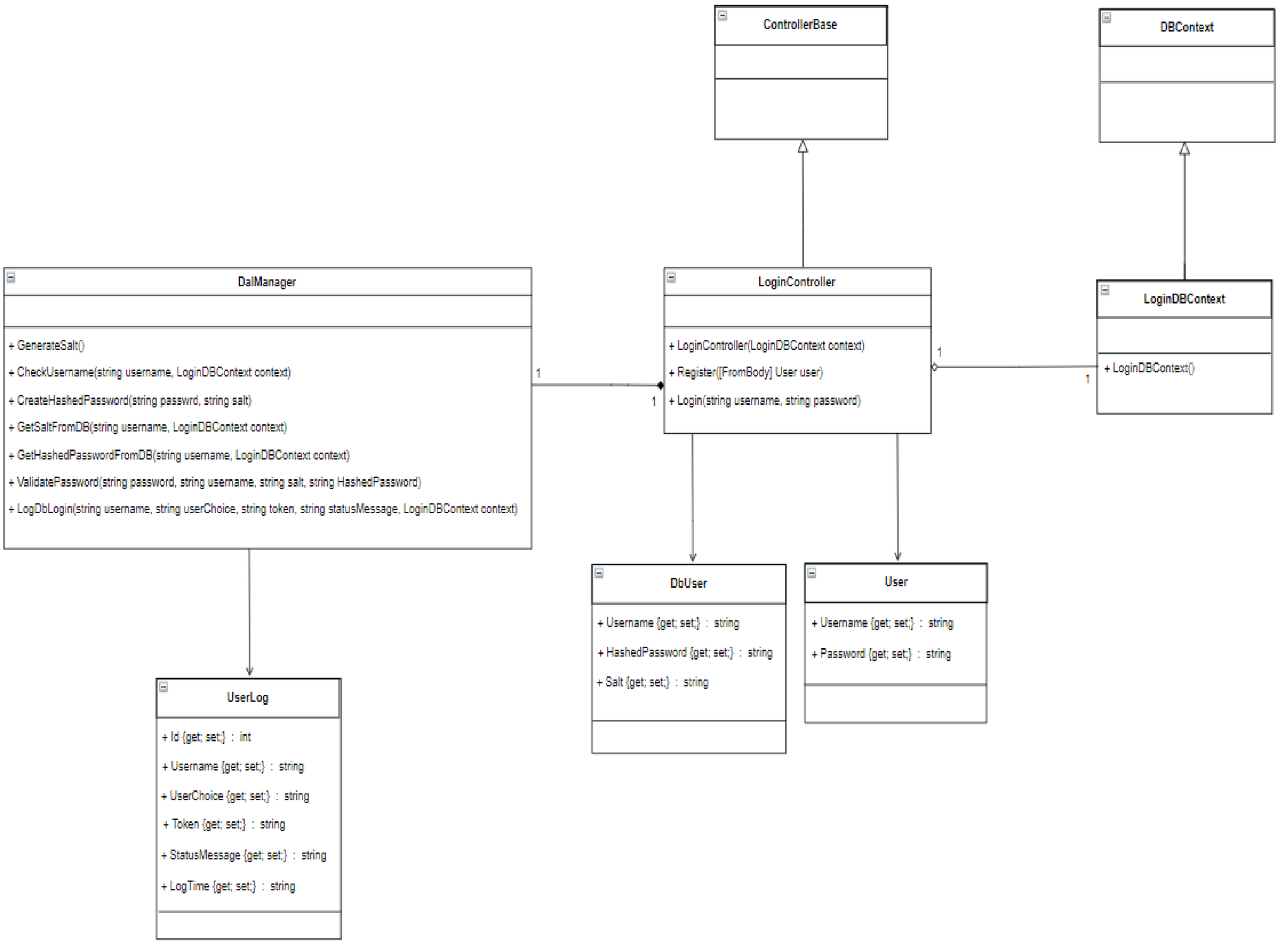


Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

Figur 3 illustrerer projektets Klassediagram for CRLogAPI:

Dette Api modtager data fra Angular´s loginformular via httppost forespørgslens Register metode som sender data videre til database tabellen Users. Log ind foregår via httpget forespørgslens Login metode, som først tjekker på brugerens input, herefter validerer om brugernavnet findes i databasen, validerer passwordet som hash hvis det gør og returnerer en Json webtoken hvis begge er rigtige. Alle private attributter ligger gemt i get / set.



Figur 4 illustrerer projektets UnitTest på CRLogAPI:

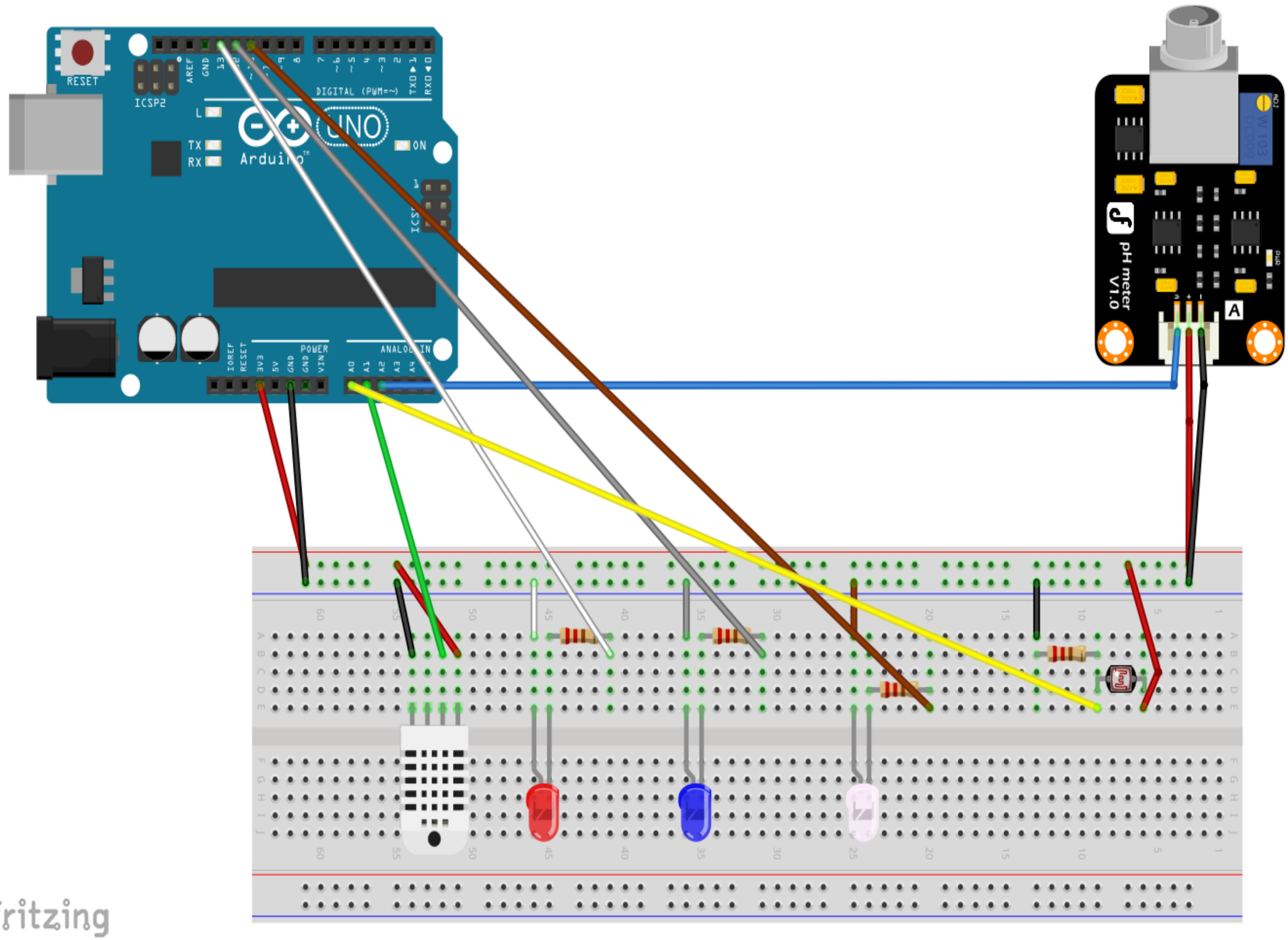
Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

Figur 5 illustrerer projektets Arduino design:

I vores Arduino opsætning har vi en temperaturmåler, en fotoresistor og en lyd sensor. Til hver opsætning er der en tilhørende LED-lampe der lyser hvis den målte værdi er højere eller lavere end en bestemt værdi. På den måde kan vi med lethed se, om f.eks. lyset er tændt eller om der bliver snakket i lokalet, altså om der befinder sig nogen.

Hver sensor har et output de sender data til - vi har brugt A0-A2 hvor lys er A0, temperatur er A1 og lyd er A2. LED-lamperne er hver forbundet til en digital port imellem 11-13, som bruges til at få lamperne til at lyse afhængig af værdien fra den tilhørende sensor.



Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

Figur 6 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 7 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
{
```

Figur 8 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

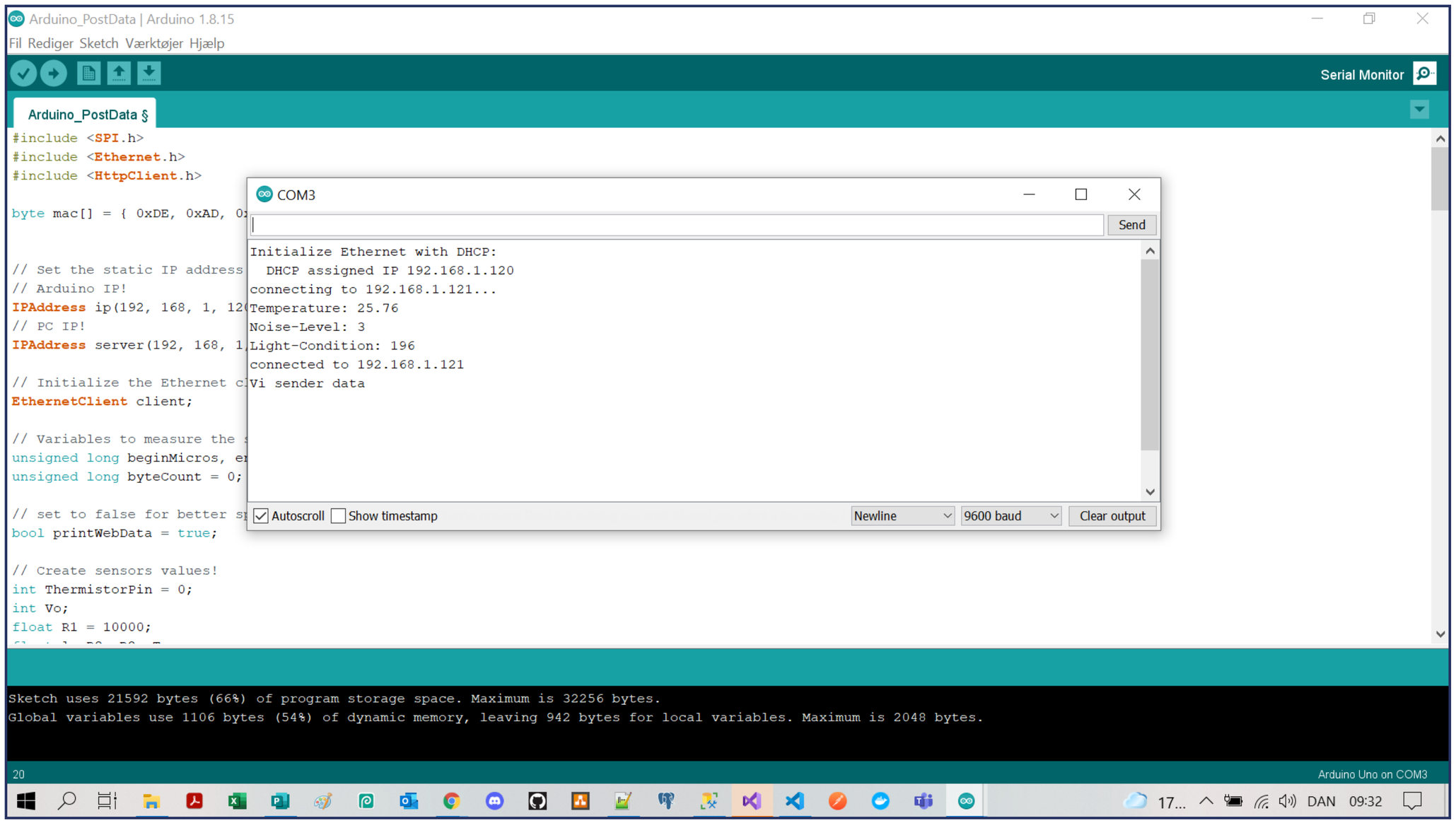
Figur 9 illustrerer projektets Arduino kode del 4 af 4:

```
  // 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 10 illustrerer Arduino til API (connecting & sending data til ClassRoomAPI):



Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

Figur 11 illustrerer projektets RDS:

De tre tabeller (Temperatures, LightConditions og NoiseLevels) har ingen relationer til hinanden, men alle data til de tre tabeller sendes fra Arduino via ClassRoomAPI. Til gengæld er der en relation mellem de to tabeller (Users og UserLogs), hvor Users UserName (unikke navne, som kun kan oprettes en version af) er PK og selv samme UserName i UserLogs er FK.

Temperatures

TempId	Temp	TempComment	TempTime

LightConditions

LightId	Light	LightComment

NoiseLevels

NoiseId	Noise	NoiseComment

Users

UserName	HashedPassword	Salt

UserLogs

Id	UserName	UserChoise	Token	StatusMessage	LogTime

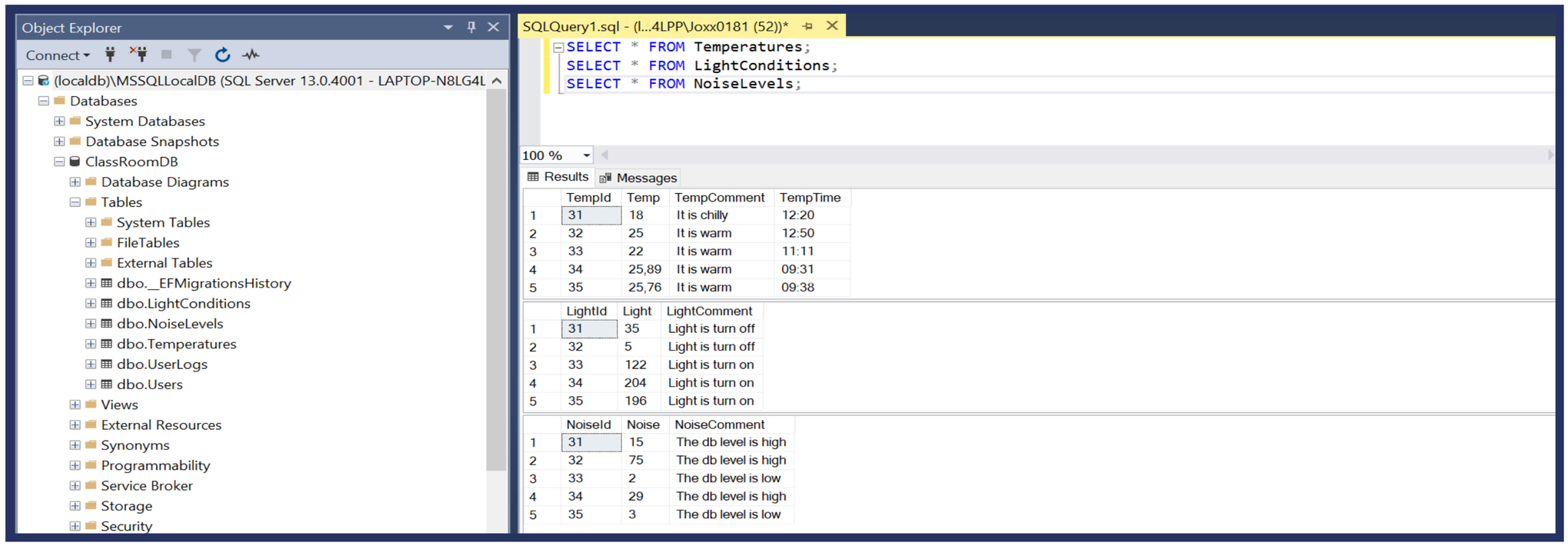
Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

Figur 12 illustrerer Databasen ClassRoomDB:

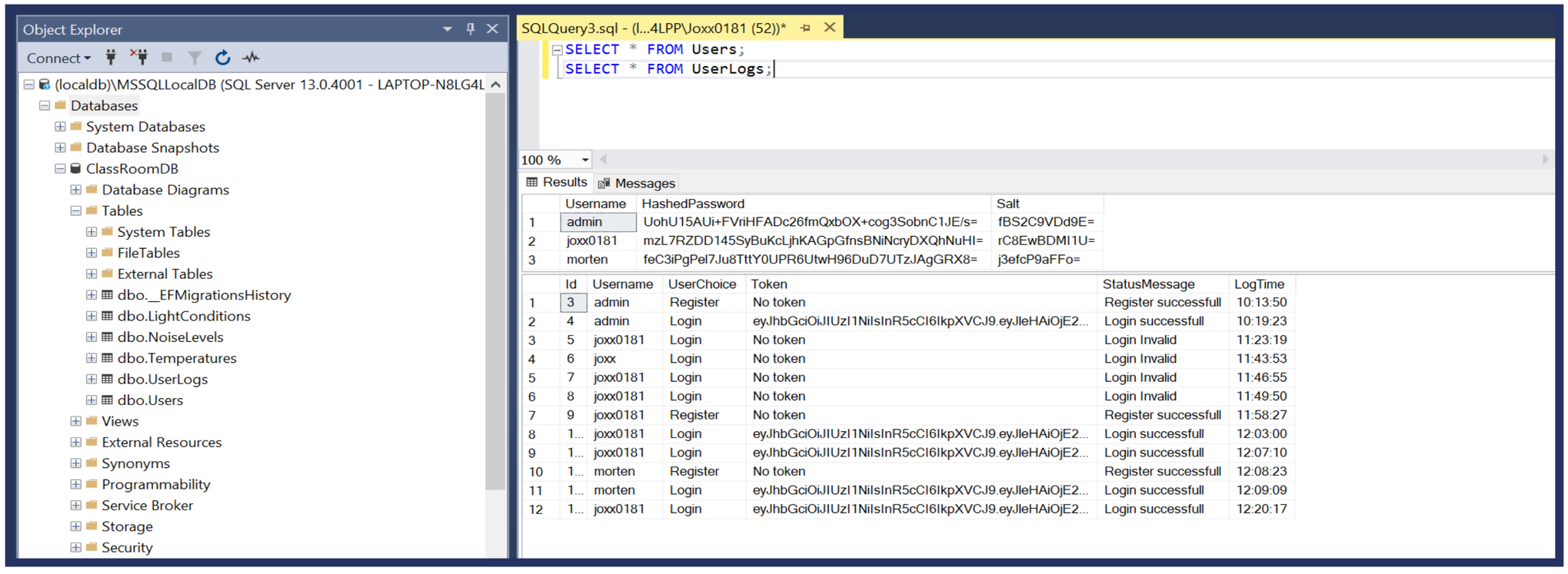
```
CREATE DATABASE ClassRoomDB;

GO
```

Ovenstående sql query er det eneste query, der er udført i databasen - Nedenstående SELECT forespørgsel viser de tre tabeller (Temperatures, LightConditions og NoiseLevels) som er oprettet med Entity Framework Core i ClassRoomAPI .



Figur 13 illustrerer Databasen ClassRoomDB:

Nedenstående SELECT forespørgsel viser de to tabeller (Users og UserLogs) som er oprettet med EF Core i CRLogAPI.

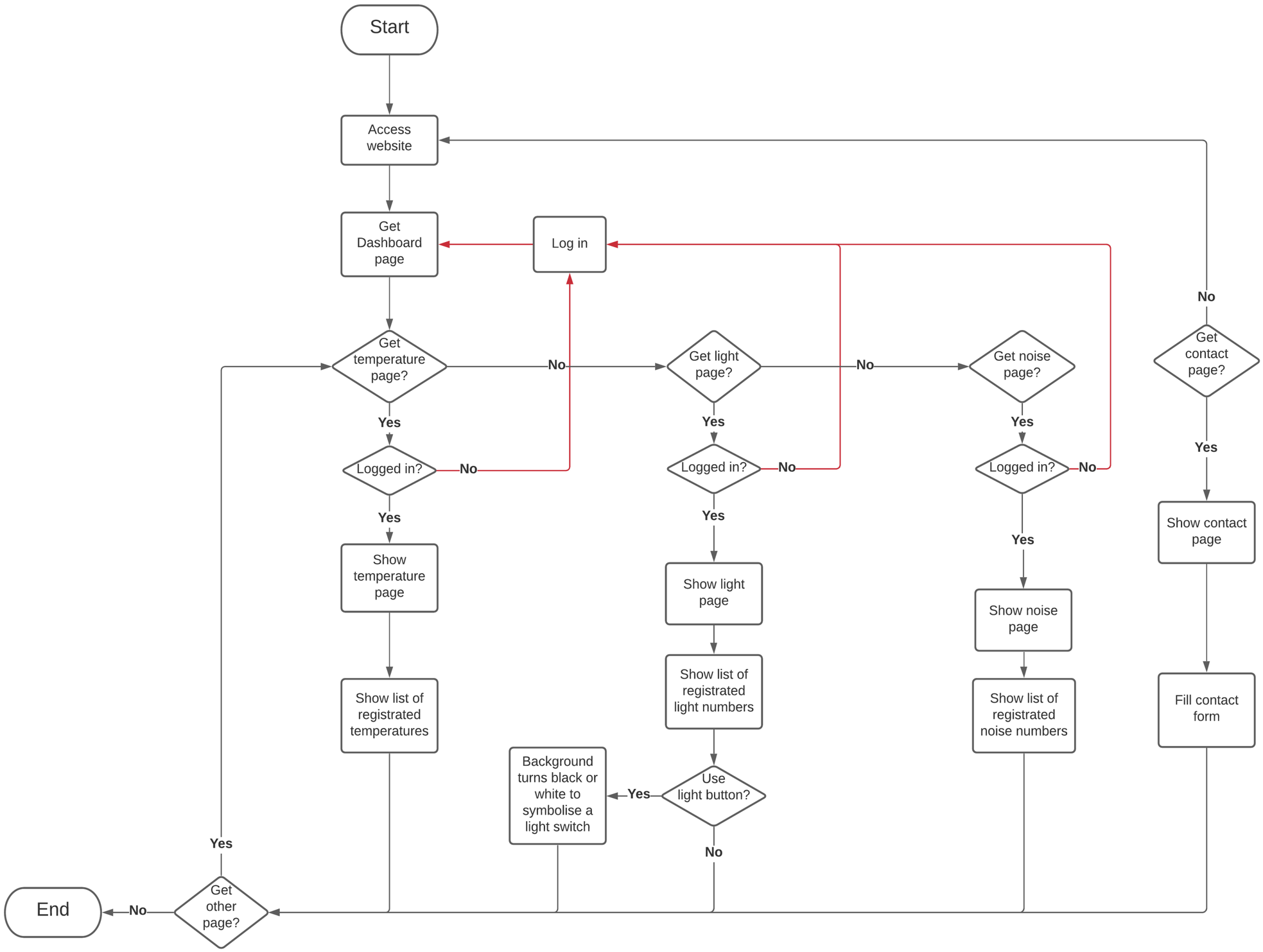


Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

Figur 14 illustrerer projektets Flowchart:

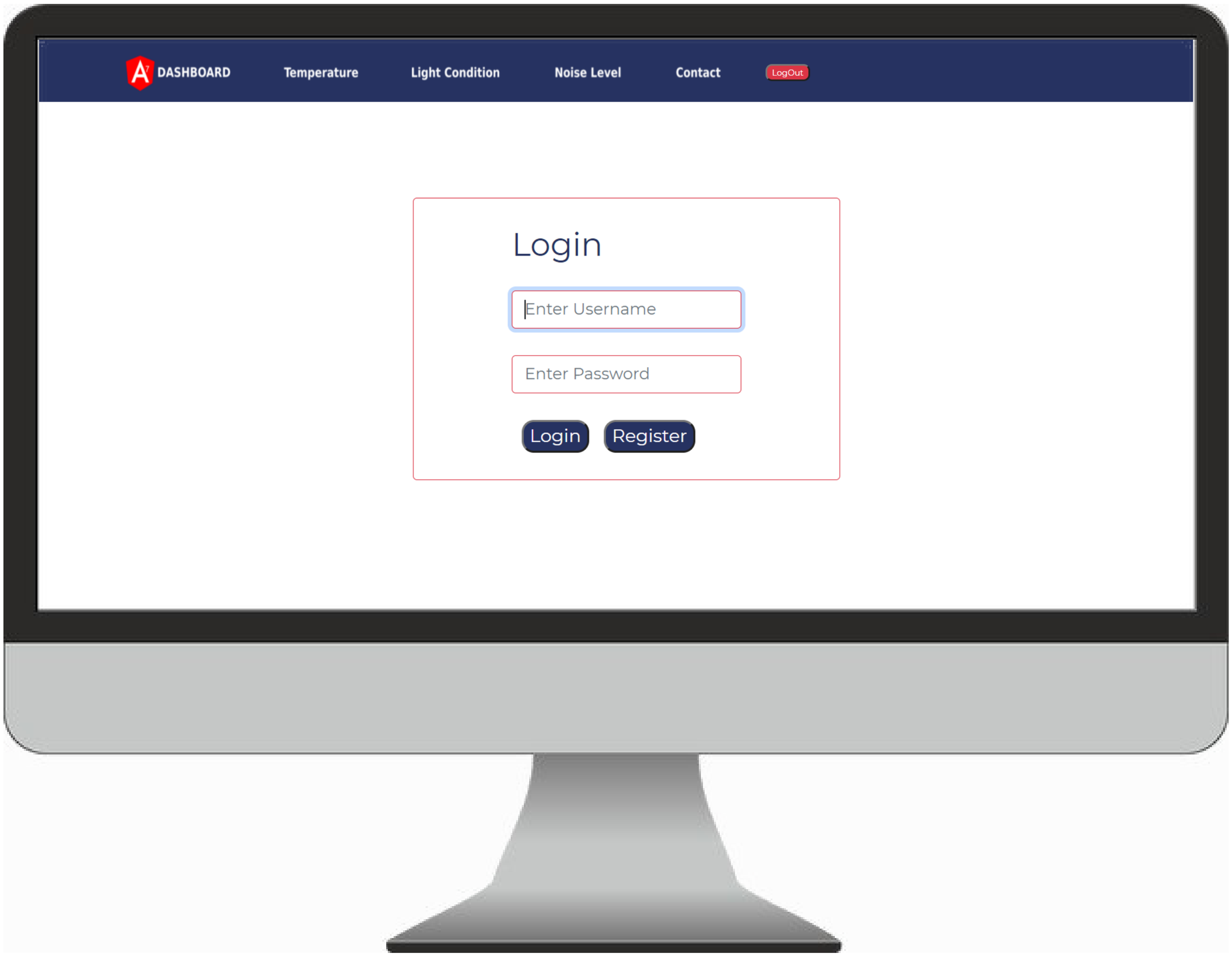
Når siden åbnes, er det første der ses forsiden. Herfra kan der navigeres til fire sider: temperatur, lys, lydniveau. For at kunne tilgå disse sider skal du være logget ind og have fået en Json Web Token i form af en cookie. Fra temperatur, lys og lydniveau-siderne kan der på hver side ses en tabel over alle rækker data der er registreret. Udover dette er der på lys siden en søgefunktion på tabellen der gør det muligt f.eks. at søge på bestemte lys niveauer eller kommentarer. Der er også en knap på lydsiden der spiller en lyd afhængig hvor højt lydniveauet er. Til sidst er der en kontaktformular hvor man kan sende en mail om eventuelle problemer eller andre ting.



Projekt Dokumentation

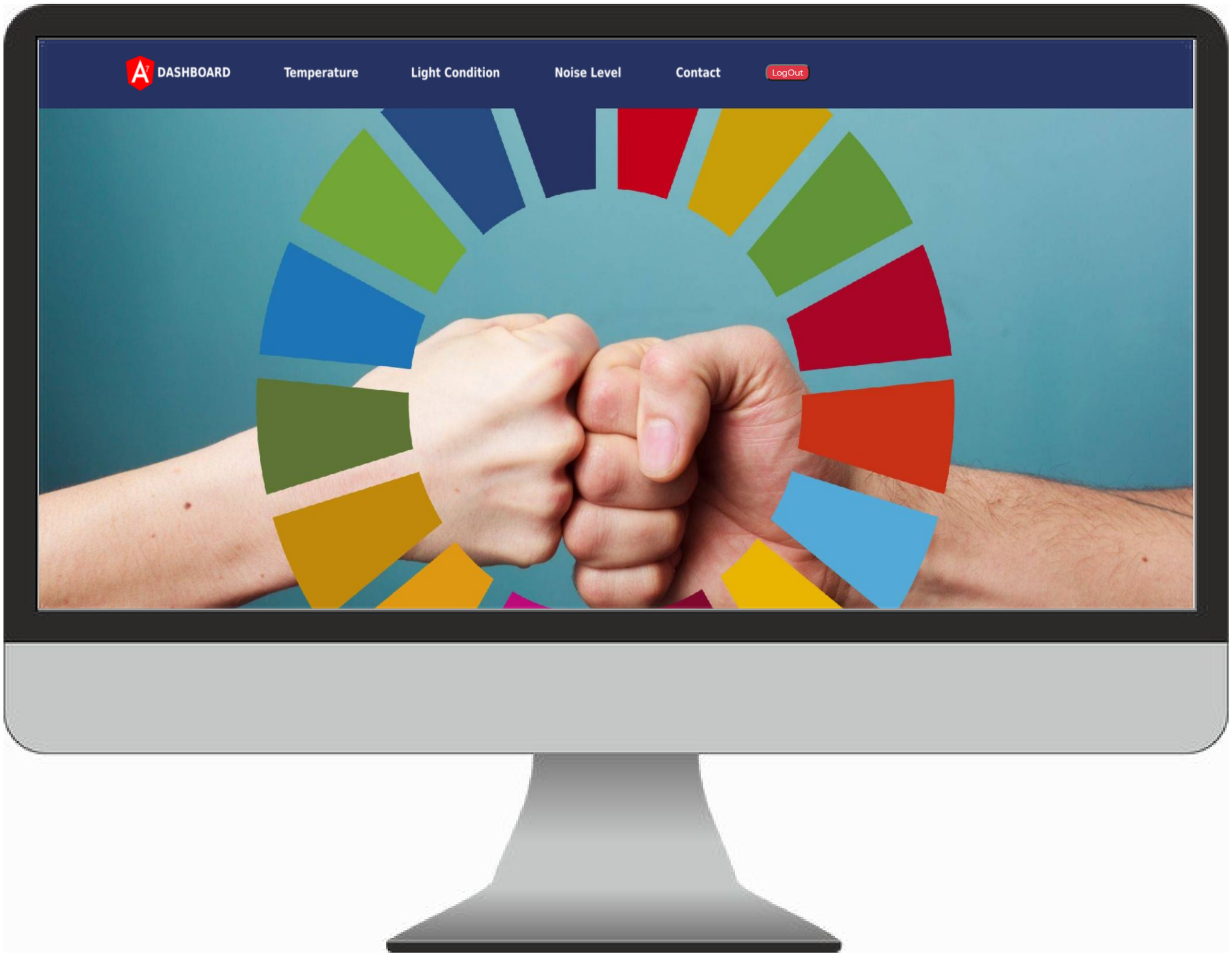
Udarbejdet af: Gruppe 1, 4108h3dakp

Figur 15 illustrerer Angular Registrering og login side (connecting med CRLogAPI):



Udarbejdet af: Gruppe 1, 4108h3dakp

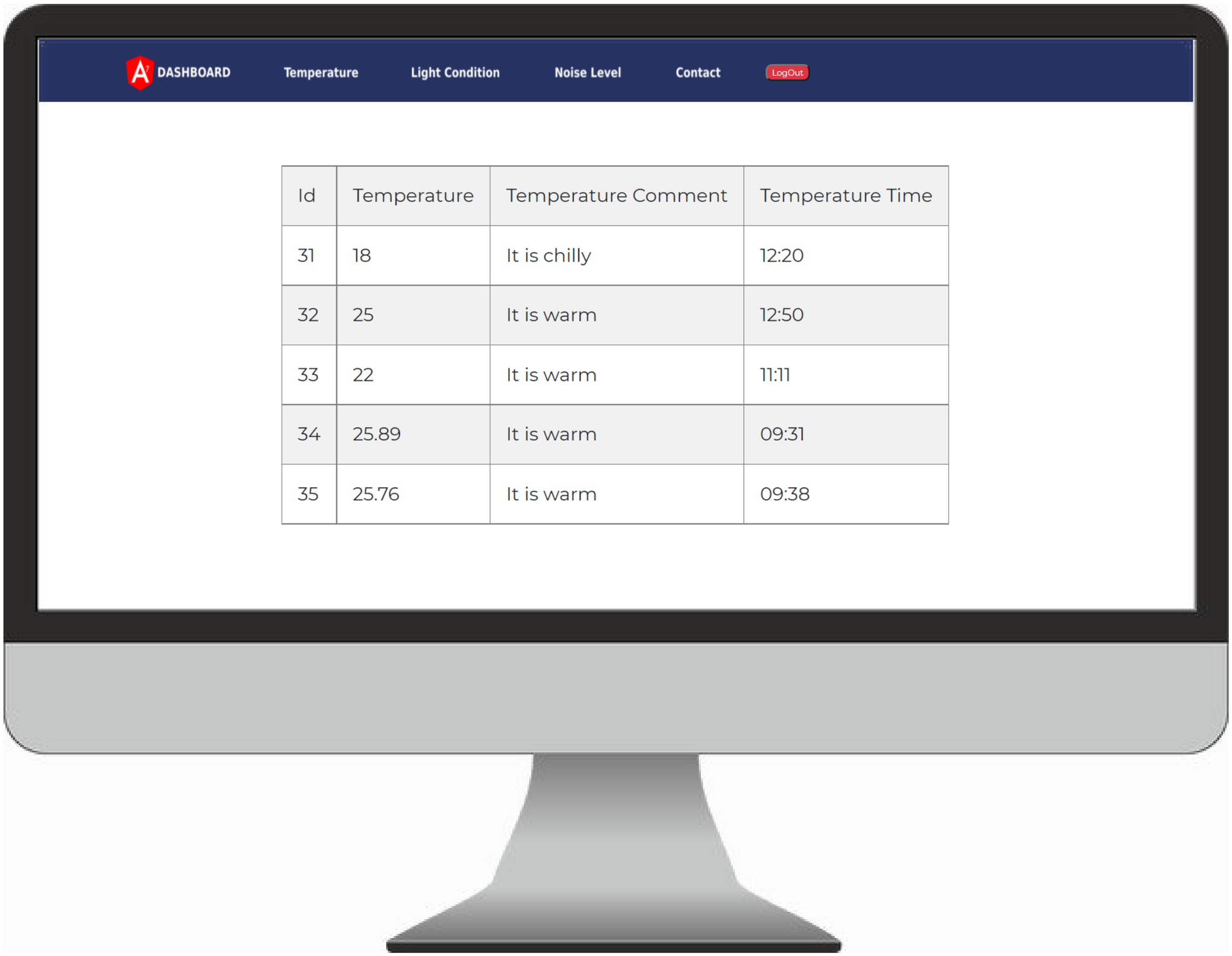
Figur 16 illustrerer omdirigering til Angular forside (this.router.navigate(['/home']);)…..):



Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

Figur 17 illustrerer Angular Temperatur data (connecting & receive data fra database via ClassRoomAPI):

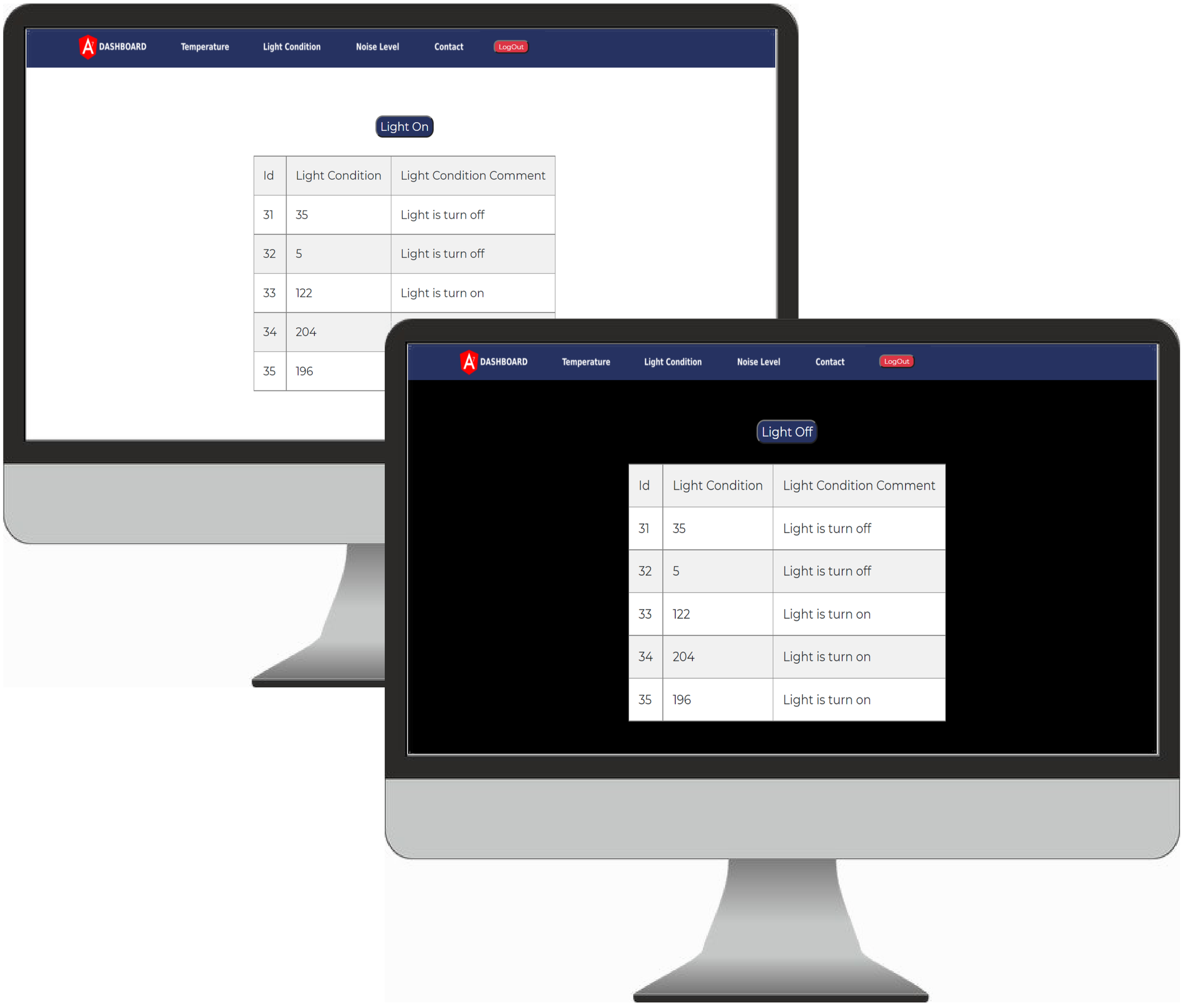


Id	Temperature	Temperature Comment	Temperature Time
31	18	It is chilly	12:20
32	25	It is warm	12:50
33	22	It is warm	11:11
34	25.89	It is warm	09:31
35	25.76	It is warm	09:38

Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

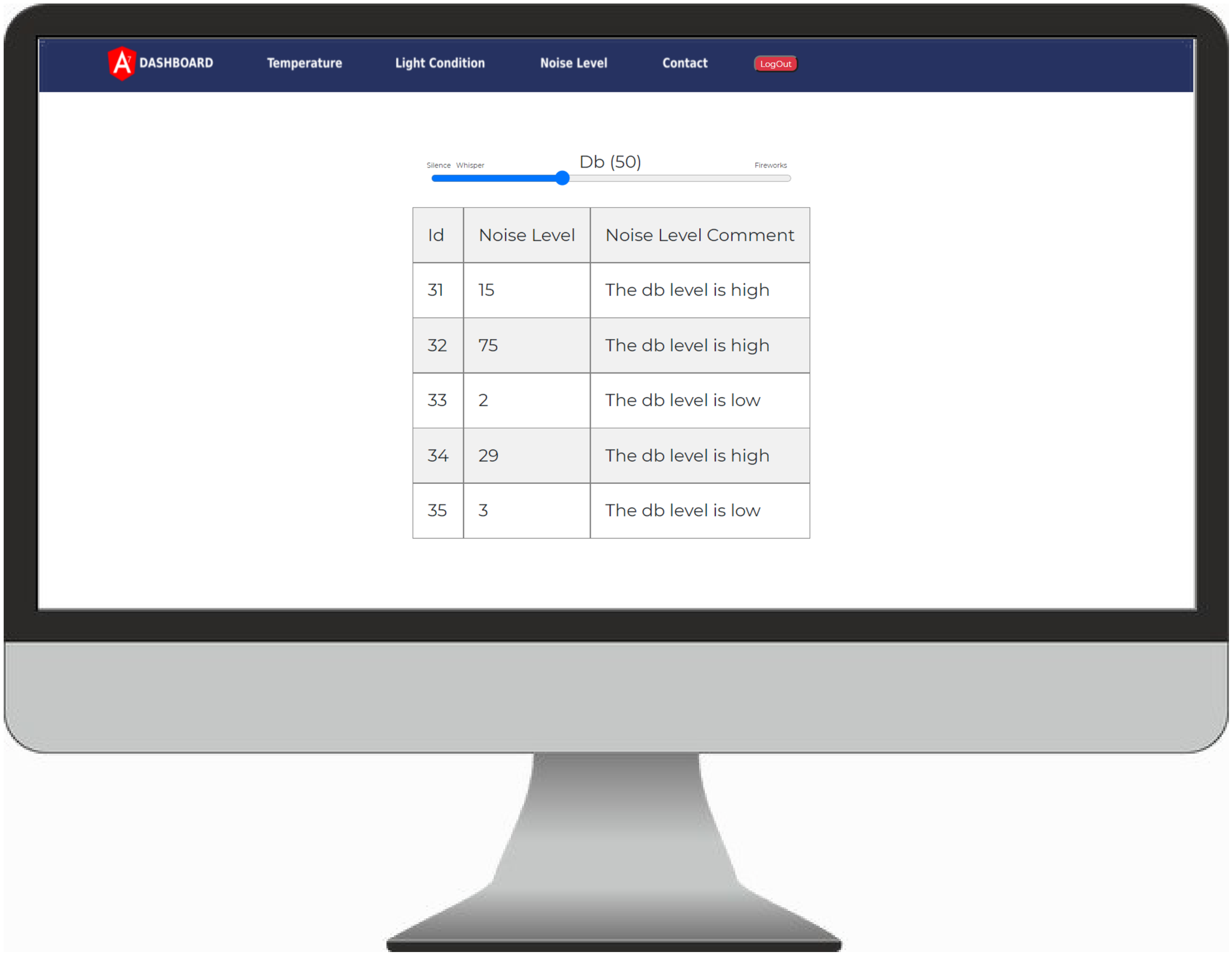
Figur 19 illustrerer Angul8r Lys data med en tænd/sluk lys funktion (connecting & receive data fra database via ClassRoomAPI):



Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

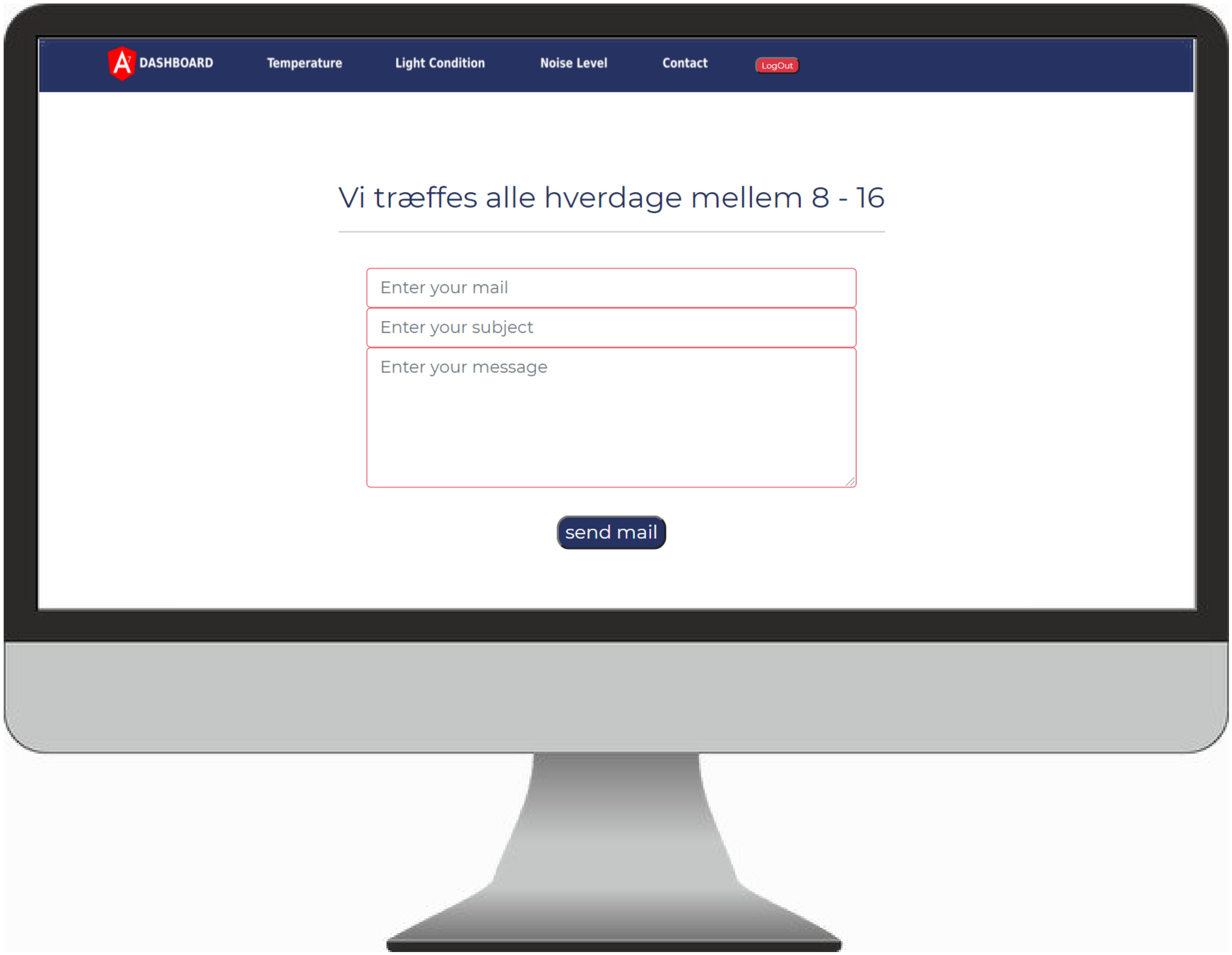
Figur 19 illustrerer Angular Lyd data med decibel skala (connecting & receive data fra database via ClassRoomAPI):



Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

Figur 20 illustrerer Angular Kontaktformular (connecting med CRMailAPI):



Projekt Dokumentation

Udarbejdet af: Gruppe 1, 4108h3dakp

Figur 21 illustrerer omdirigering til tak-for-din-mail side (this.router.navigate(['/contact-thanks']);)