



# Smarthome

Turn on your lights, close your blinds, control  
the inside temperature and automate everything.



# Inhoudsopgave

<b>Briefing</b>	<b>6</b>
<b>Analyse</b>	<b>8</b>
Firebase	8
React webapp	8
Raspberry Pi	9
MQTT	9
RF 433Mhz	9
Nodes	9
<b>Planning</b>	<b>10</b>
Node & NodeMCU software	10
Raspberry Pi initiële setup & MQTT broker	10
Node.js hub	11
React app	11
<b>Inspiration</b>	<b>12</b>
<b>Sitemap</b>	<b>14</b>
<b>Wireflow</b>	<b>15</b>
<b>Styleguide</b>	<b>16</b>
<b>Visual designs</b>	<b>17</b>
<b>Code snippets</b>	<b>21</b>
<b>Wiring diagrams</b>	<b>26</b>
<b>Screenshots</b>	<b>27</b>
<b>User manual</b>	<b>31</b>
<b>Arduino / NodeMCU</b>	<b>33</b>
<b>Raspberry Pi</b>	<b>34</b>
Benodigdheden	34
Installatie	34

Configuratie	34
Smart home als service	35
Gebruik	35
<b>Webapp en iOS app</b>	<b>36</b>
Benodigdheden	36
Installatie	36
Configuratie	36
Deploy naar Firebase	36
Smart home iOS app	36



# Discover

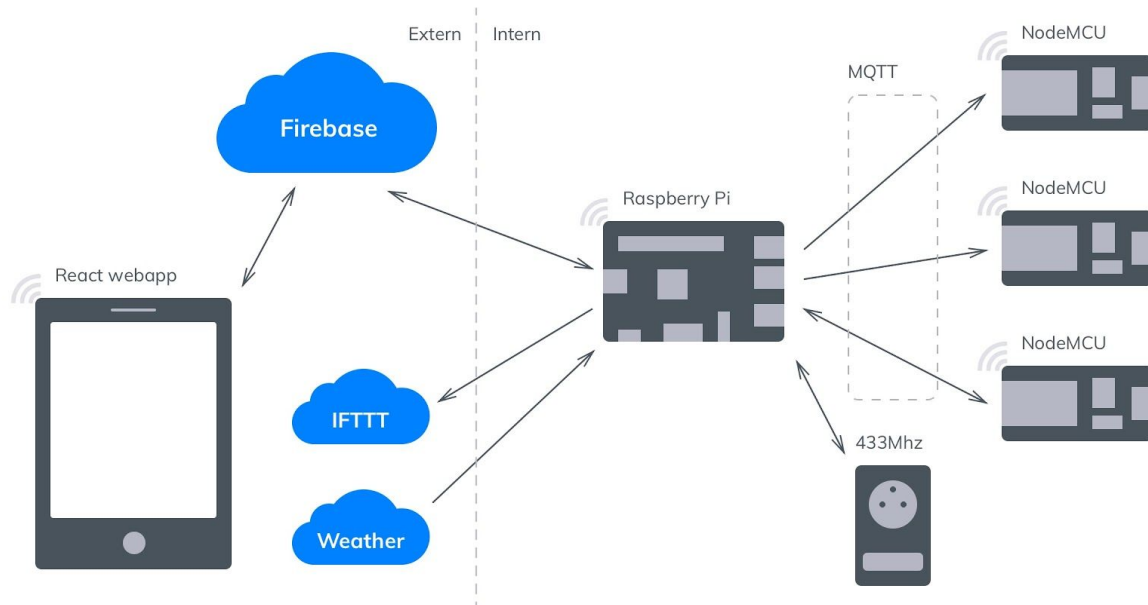
# Briefing

Hoe kan een softwaresysteem realtime communicatie realiseren tussen de gebruikers, software en hardware, en dit op een gebruiksvriendelijke en attractieve manier?

# Define



# Analyse



Het systeem bestaat uit vier essentiële delen, zijnde de React webapp, Firebase, Raspberry Pi hub en alle nodes. Bovenstaand schema illustreert de verhouding tussen deze delen. Let op de stippellijn die de externe delen en interne delen voorstellen. Alle onderdelen links van de stippellijn zijn externe delen die elk op een eigen netwerk zitten. De onderdelen rechts van de stippellijn zijn interne delen die allemaal op hetzelfde netwerk zitten.

## Firestore

Firestore wordt gebruikt om de staat bij te houden van elke node en sensor. Bij elke aanpassing worden de aanpassingen naar alle verbonden toestellen gestuurd, in dit geval de app en de Raspberry Pi hub.

## React webapp

De app leest alle data van nodes en sensoren in via Firestore en toont deze in een webinterface. Om de staat van een node of sensor aan te passen, wordt deze door de app aangepast in Firestore.

# Raspberry Pi

De Raspberry Pi zit in het intern netwerk binnenshuis en zorgt voor de afhandeling van alle aanpassingen in Firebase. Wanneer er een aanpassing gebeurt in Firebase, leest de hub in om welke node het gaat en stuurt vervolgens het correcte commando door via MQTT of RF 433Mhz naar de node. MQTT kan draaien op de Raspberry Pi of in the cloud.

## MQTT

MQTT is een realtime communicatie protocol waarbij geabonneerd kan worden op topics en op deze topics gepubliceerd kan worden. Wanneer elke node op een ander topic is geabonneerd, kan er tussen de bepaalde node en de hub gecommuniceerd worden zonder andere nodes lastig te vallen.

## RF 433Mhz

RF modules zijn heel populair door de lage kostprijs en lange levensduur. Deze komen vaak met een afstandsbediening en kunnen opnieuw geprogrammeerd worden. De zender en ontvanger maken gebruik van de 433Mhz frequentie om binaire codes te sturen en te ontvangen.

## Nodes

Nodes kunnen zowel inputs als outputs zijn. In dit project worden ESP8266 microcontrollers en RF 433Mhz ontvangers gebruikt. ESP8266 microcontrollers kunnen verbinding maken met een wifi-netwerk.

# Planning

Taak	Prioriteit	Tijdsbesteding
Nodes & NodeMCU software	1	10 uur
Raspberry Pi initiële setup & MQTT broker	4	4 uur
Node.js hub	2	24 uur
React webapp	3	25 uur
Product dossier	/	8 uur
Debugging & extra's	/	+ 30%

## Node & NodeMCU software

Taak	Prioriteit	Tijdsbesteding
MQTT connectie	1	1 uur
Firebase connectie	2	1 uur
Relais module	3	2 uur
Servomotor module	4	3 uur
Temperatuur module	5	2 uur
Button module	6	1 uur

## Raspberry Pi initiële setup & MQTT broker

Taak	Prioriteit	Tijdsbesteding
Raspbian OS installeren	1	1 uur
Raspberry Pi configureren	2	1 uur
MQTT broker installeren en configureren	3	1 uur
Node installeren	4	1 uur

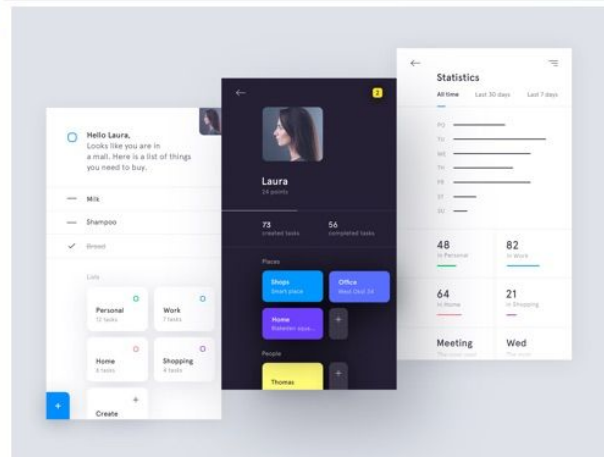
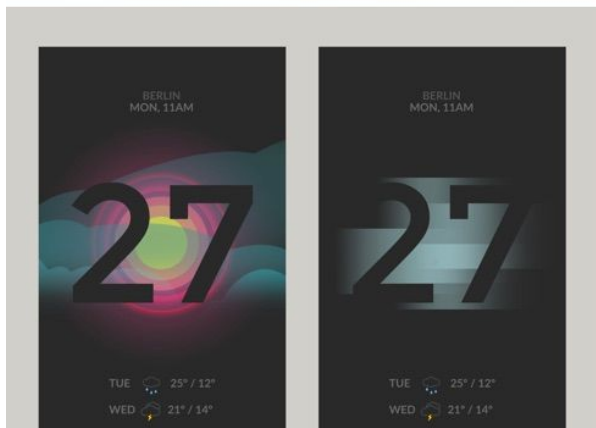
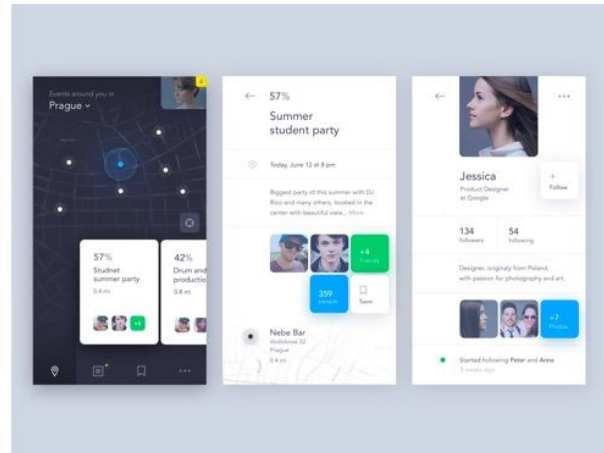
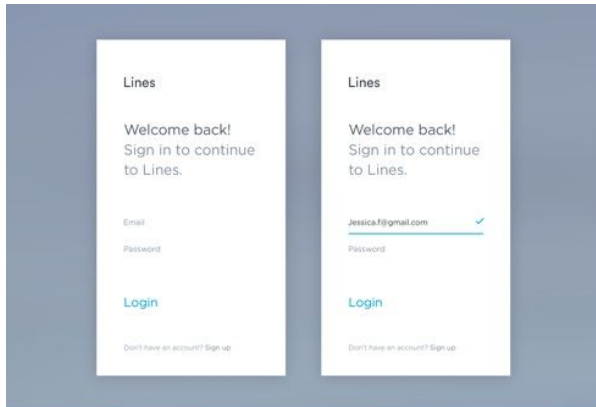
## Node.js hub

Taak	Prioriteit	Tijdsbesteding
Firebase connectie	1	2 uur
MQTT connectie	2	1 uur
Devices en sensoren registreren in Firebase	3	2 uur
Aanpassingen detecteren en afhandelen	4	4 uur
Thermostaat	5	4 uur
Automatisaties checken en afhandelen	6	6 uur
IFTTT integreren	7	3 uur
Service aanmaken	8	2 uur

## React app

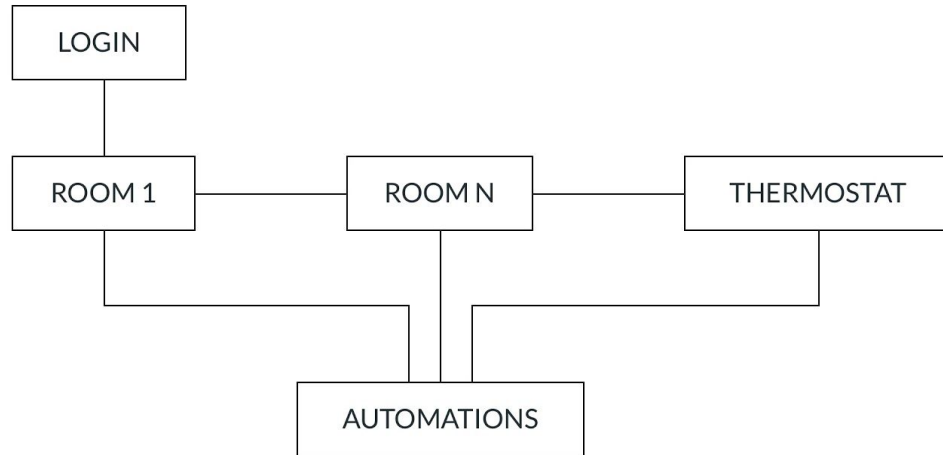
Taak	Prioriteit	Tijdsbesteding
React, Flow & Styled Components installeren	1	2 uur
Firebase setup en configureren	2	2 uur
Pagina systeem en header	3	3 uur
Output devices tonen en besturen	4	4 uur
Sensoren tonen	5	2 uur
Integratie met OpenWeatherMap	6	3 uur
Firebase authenticatie integreren	7	3 uur
Thermostaat component	8	2 uur
Automatisaties aanmaken en beheren	9	4 uur

# Inspiration



# Design

# Sitemap



# Wireflow

## Login

Welcome back  
Lorem ipsum dolor sit amet,  
consectetur adipiscing elit.

username

password

login

## Automations

Automations

When temperature is more  
then 25 then trigger  
temperature event.

Delete automation

Add automation

topic

value

trigger

## Lights

Shaun's room Living room

Room name ☐

Room name ☐

Blinds name ☐

Room name ☐

## Blinds

Living room Thermostat

Blinds name ☐

Room name ☐

Blinds name ☐

## Thermostat

Thermostat Lights

9 Weather in Ghent

21° 45%

Thermostat



# Styleguide

Header



Lights

Thermostat >

Forms

username

password

sign in

login with Google

Weather

9°

Weather in Ghent

22 km/h 91%

Automations

When humidity is less then 30  
then trigger humidity event.

Last fired at 26/12/2017 11:47

Delete this automation >

Controls

Bureaulamp



Outside spot



Bathroom



Garage door



Sensors



22°



43%



Heading 1 2.6rem regular Muli

Heading 2 1.8rem regular Muli

Lorem ipsum dolor sit amet, consectetur  
adipiscing elit, sed do eiusmod tempor  
incididunt ut labore et dolore magna  
aliqua.

1.6rem/2.6rem regular Muli

# Visual designs

Login

Welcome back!

With Smart home you can turn on your lights, close your blinds and control the inside temperature.

sign in

Room 1

+

Shaun's room Living room >

Desk lamp

☒

Night light

☐

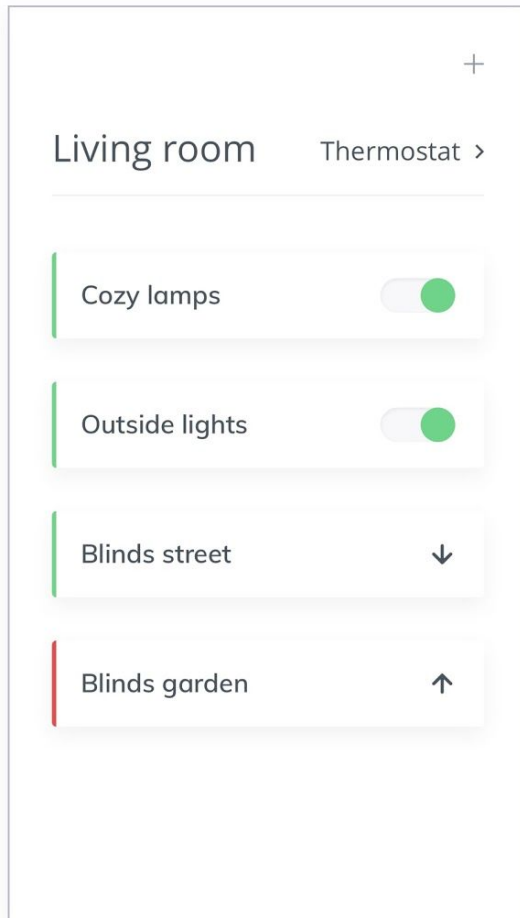
Outlet TV

☒

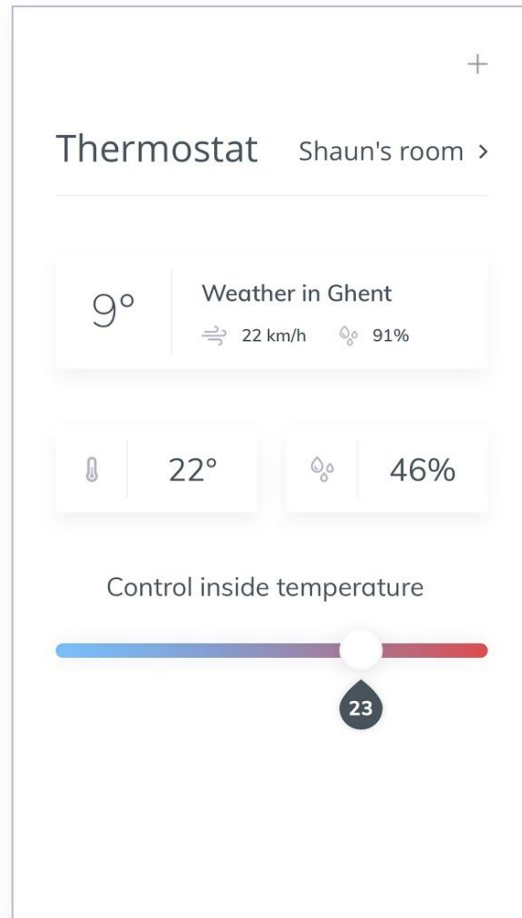
Blinds


↑

Room 2



Thermostat





## Automations

---

When temperature is more then 25 then trigger temperature event.

Last fired at 28/12/2017 16:14

---

**Delete this automation >**

When humidity is less then 30 then trigger humidity event.

Last fired at 26/12/2017 11:47

---

**Delete this automation >**

### Add automation

---

when sensor

is more than
is less than
equals

value

then tigger event

add automation

# Develop

# Code snippets

```
/*
 * Watch for changes in Firebase database, get data
 * and call function with changed device as parameter
 */
deviceRef.on("child_changed", snapshot => {
  handleDevice(snapshot.val());
});

/*
 * When snapshot contains multiple child nodes,
 * convert to array so you can map every child node
 */
const snapshotToArray = snapshot => {
  let returnArr = [];
  snapshot.forEach(childSnapshot => {
    let item = childSnapshot.val();
    item.key = childSnapshot.key;
    returnArr.push(item);
  });
  return returnArr;
};

/**
 * Get device platform, send signal to MQTT or RF433 and log action
 */
const handleDevice = device => {
  if (device.platform === "mqtt") {
    client.publish(device.config.topic.toString(),
device.state.toString());
  } else if (device.platform === "rf433") {
    let rfCode =
      device.state === 1 ? device.config.on_code :
device.config.off_code;
    executePythonScript("rfSend", rfCode);
  }

  writeLog("Device: " + device.name + " changed with payload: " +
device.state);
};
```

```

/**
 * Check if automation is triggered less than one hour ago
 * and send POST request to maker.ifttt.com
 */
const triggerIftttEvent = (automation, value) => {
  const date = new Date();
  const hourAgo = date.setHours(date.getHours() - 1);
  if (hourAgo >= automation.lastfired || !automation.lastfired) {
    // Set postdata
    let postData = querystring.stringify({
      value1: value
    });

    // Set request options
    let options = {
      hostname: "maker.ifttt.com",
      path:
`/trigger/${automation.then.event}/with/key/${config.ifttt.apikey}`,
      method: "POST",
      headers: {
        "Content-Type": "application/x-www-form-urlencoded",
        "Content-Length": postData.length
      }
    };

    // Create request
    const req = https.request(options, res => {
      res.setEncoding("utf8");
      res.on("data", chunk => {});
      res.on("end", () => {});
    });

    // Log error
    req.on("error", e => {
      writeLog("!!!ERROR!!! Ifttt: " + e.message);
    });

    // Send request
    req.write(postData);
    req.end();
  }
};

```

```

/**
 * Handle automations form when the submit button is pushed.
 * Get all data from state, check data, create new key in database
 * and set all data to new key if data is valid.
 * Show an error if not all data is present.
 */
handleForm = () => {
  const {
    sensor,
    condition,
    value,
    event,
    platform,
    devicestate
  } = this.state;

  if (sensor && condition && value && event && platform) {
    const postkey = this.props.automationRef.push().key;
    this.props.automationRef.child(postkey).set({
      key: postkey,
      if: {
        [condition]: value,
        id: sensor
      },
      then: {
        event: event,
        platform: platform,
        state: devicestate
      }
    });

    this.setState({
      sensor: "",
      condition: "morethan",
      value: "",
      event: "",
      platform: "default",
      devicestate: "default"
    });
  } else {
    this.setState({ error: true });
  }
};

```



```

/**
 * Handle slider drag and check closest step
 */
handleDrag = e => {
  let slider = this.slider.getBoundingClientRect();
  let mouseX = e.clientX - slider.left;

  if (mouseX > 0 && mouseX < slider.width) {
    let { step, knobX } = this.getClosest(
      mouseX,
      slider.width,
      this.props.steps
    );
    this.setState({ knobX: knobX, step: step });
    this.props.changeStep(step);
  }
};

getClosest = (mouseX, sliderWidth, steps) => {
  let stepWidth = sliderWidth / steps;
  let modulo = (mouseX / stepWidth) % steps;

  let initialStep = Math.floor(modulo);
  let aboveHalf = (modulo - initialStep) * 10 >= 5 ? true : false;

  let step = aboveHalf ? initialStep + 1 : initialStep;
  let knobX = step * stepWidth;

  return { step, knobX };
};

```

```

/**
 * Read out DHT sensor and send to Firebase in Arduino IDE
 */
void loop() {
    // put your main code here, to run repeatedly:
    if (!client.connected()) {
        reconnect();
    }

    float h = dht.readHumidity();
    float t = dht.readTemperature();

    if (isnan(h) || isnan(t)) {
        Serial.println("Failed to read from DHT sensor!");
    } else {
        Firebase.setFloat("sensors/temperature/value", t);
        Firebase.setFloat("sensors/humidity/value", h);
        delay(5000);
    }

    client.loop();
}

/**
 * Load URL in WKWebView component in Swift
 */
@IBOutlet weak var webView: WKWebView!

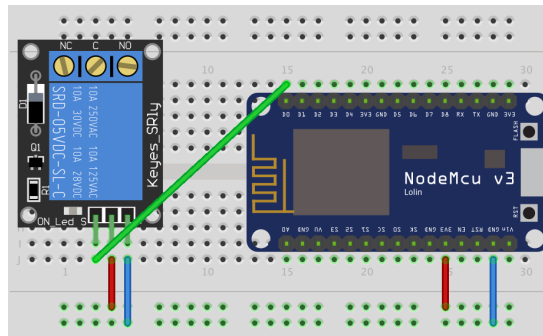
override func viewDidLoad() {
    super.viewDidLoad()

    let url = URL(string: "https://smarthome-9d4f6.firebaseio.com/")
    let request = URLRequest(url: url!)

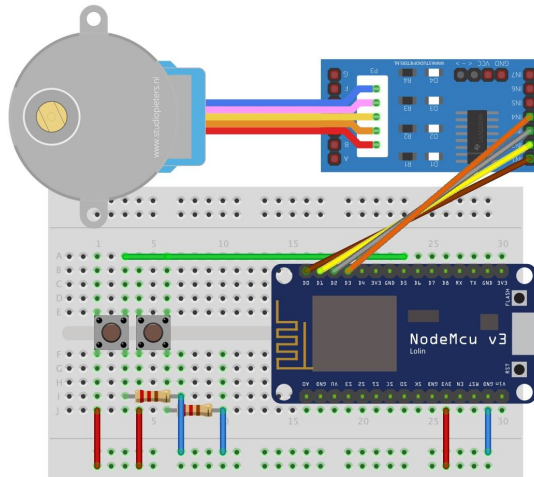
    webView.load(request)
    webView.scrollView.bounces = false
}

```

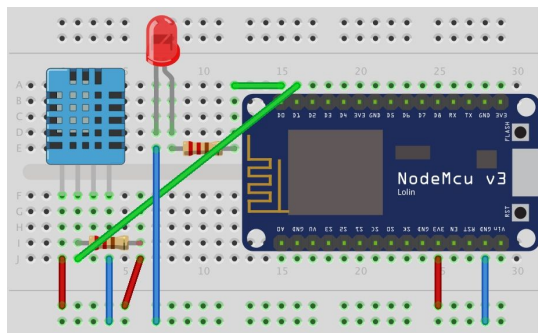
# Wiring diagrams



fritzing



fritzing



fritzing

# Screenshots

Login

## Smart home

With Smart Home you can turn on your lights, close your blinds and control the inside temperature.

sign in

Room 1

+

Shaun's room 

Living room >

Desk lamp

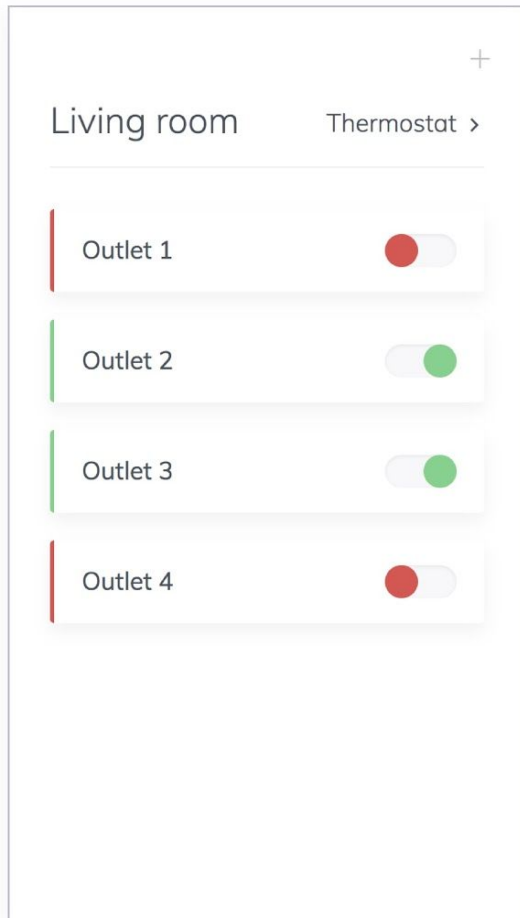
Night light

Outlet 1

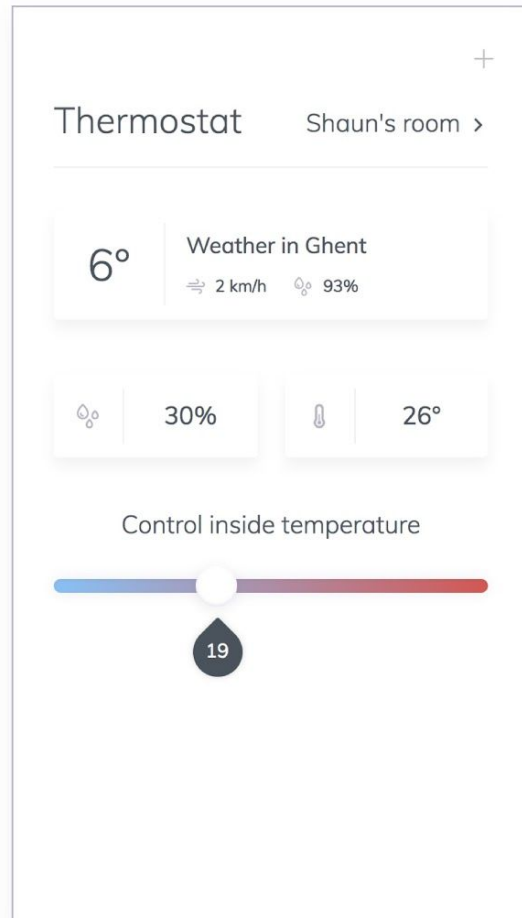
Outlet 2


Blinds

## Room 2



## Thermostat





## Automations

When time is equals 20:00 then turn shaun-bureaulamp off.

Never fired

Delete this automation

When temperature is more than 24 then turn shaun-blinds off.

Never fired

Delete this automation

when sensor

more than

value

then trigger

add automation

# Deliver

# User manual

Bedankt voor uw aankoop van een Smart home systeem, model X80085.

Wanneer je de app voor de eerste keer opent, wordt een gebruikersnaam en wachtwoord gevraagd. Deze gegevens worden aangemaakt tijdens de deployment van het systeem.

Als je succesvol bent ingelogd en je hebt bij de deployment enkele nodes en rooms toegevoegd, dan zie je deze nu in de eerste ruimte de toegevoegde devices. Deze devices kan je in- en uitschakelen door op de toggle te duwen rechts van de naam.

Bovenaan links in de header staat de naam van het huidige tabblad, rechts ervan staat de naam van de volgende tab, en erboven staat een “plus”-icoon om automatisaties toe te voegen.

Wanneer je alle ruimtes hebt overlopen verschijnt het volgende tabblad “Thermostat” met daarop het weer buiten, de temperatuur en vochtigheidsgraad binnen en een regelaar voor de thermostaat.

De thermostaat gaat aan wanneer de temperatuur lager is dan de ingestelde temperatuur, en gaat uit wanneer de temperatuur 3 graden Celsius hoger is dan de ingestelde temperatuur. Wanneer de temperatuur stijgt door zon en niet door de verwarming, blijft de thermostaat uit.

Automatisaties zijn automatische events die worden getriggerd als een sensor voldoet aan een bepaalde voorwaarde. Standaard wordt elke toegevoegde sensor ondersteund, en bijkomend is er een tijd-sensor waarmee acties kunnen worden getriggerd vanaf een bepaald tijdstip. Het is mogelijk om devices aan te sturen met automatisaties, maar ook andere online diensten en notificaties met de externe service [ifttt.com](https://ifttt.com). Maak hiervoor een webhook action aan en geef de gekozen event naam in bij het aanmaken van de automatisatie.



# Deploy

# Arduino / NodeMCU

In de map /nodes staan alle voorgeprogrammeerde nodes. Open het gewenste programma met de Arduino IDE, vul alle configuratievelden in en upload het programma naar de microcontroller.

Elk programma is voorzien van een automatische wifi setup, waarmee je via een webinterface de netwerkinstellingen kan ingeven. Bekijk ook de diagrammen voor de correcte aansluitingen.

# Raspberry Pi

## Benodigdheden

- Raspberry Pi met Raspbian Stretch
- Git, Node en NPM
- Mosquitto broker

## Installatie

- Open terminal
- Installeer Smart home met onderstaande commando's

```
$ cd /home/pi
$ git clone https://github.com/gdmgent-1718-wot/smarthome
$ cd smarthome/server && npm install
```
- Installeer de nodige libraries voor RF433 communicatie

```
$ sudo apt-get install python3-pip
$ sudo pip3 install rpi-rf
```

## Configuratie

Kopieer `config/config.json.example` naar `config/config.json`. Vul alle velden in met de correcte data en voeg devices en sensoren toe.

# Smart home als service

Om Smart home te starten samen met de Raspberry pi, moet je het systeem instellen als systeem service. Voer hiervoor onderstaande commando's uit.

```
$ sudo cp smarthome.service /lib/systemd/system/smarthome.service
```

```
$ sudo chmod u+rwX /lib/systemd/system/smarthome.service
```

```
$ sudo systemctl daemon-reload
```

```
$ sudo systemctl enable smarthome
```

## Gebruik

- Start Smart home met `$ sudo service smarthome start`
- Stop Smart home met `$ sudo service smarthome stop`

# Webapp en iOS app

## Benodigdheden

- Git, Node en NPM
- Firebase CLI
- XCode

## Installatie

- Open terminal
- Installeer Smart home met onderstaande commando's

```
$ git clone https://github.com/shaunjanssens/smarthome Smarthome
$ cd Smarthome && npm install
```

## Configuratie

Kopieer `src/config/config.json.example` naar `src/config/config.json`. Vul alle velden in met de correcte data.

## Deploy naar Firebase

- Bouw alle bronbestanden en deploy naar Firebase met onderstaande commando

```
$ npm run deploy
```

## Smart home iOS app

In de map `/ios-app` staat een kant-en-klare iOS app waarbij enkel de url van de webapp moet worden aangepast. Open het project in XCode en pas de url aan in het bestand `ViewController.swift`. Bouw de bronbestanden en debug op je eigen toestel.

