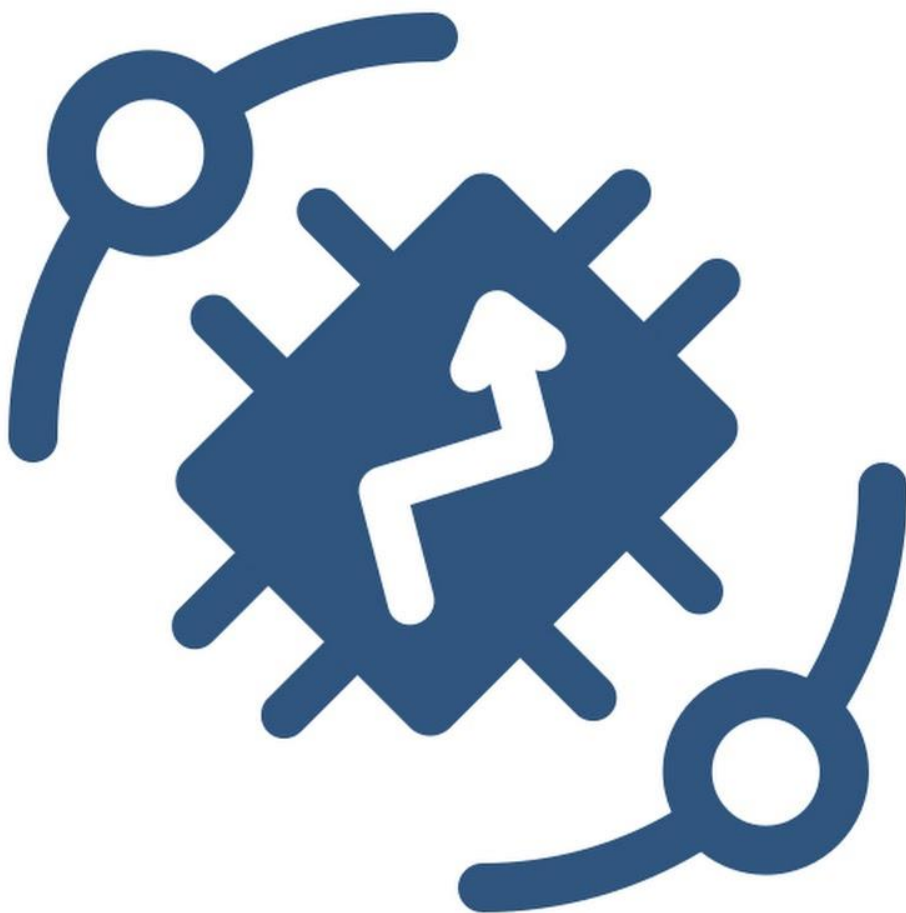


Juha Mustonen

Puhdastilojen olosuhteiden seurantajärjestelmän käyttöliittymä



Insinööri (AMK)

Tieto- ja viestintäteknikka

Kevät 2021



**KAMK • University
of Applied Sciences**

Tiivistelmä

Tekijä: Juha Mustonen

Työn nimi: Puhdastilojen olosuhteiden seurantajärjestelmän käyttöliittymä

Tutkintonimike: Insinööri (AMK), tieto- ja viestintätekniikka, peliala

Asiasanat: Esineiden internet, pilvipalvelut, thingsboard, käyttöliittymä

Tämän opinnäytetyön tarkoituksena oli rakentaa Oulun yliopiston Mittaustekniikan yksikön puhdastilojen olosuhteita valvova järjestelmä. Sen tuli rakentua kolmesta osasta: sensoreista, pilvipalvelimesta ja tietokannasta. Henkilökohtainen osuus painottui käyttöliittymän suunnitteluun. Käyttöliittymän tuli visualisoida mitattavat arvot ja ilmoittaa mahdollisista poikkeamista. Sensoreita oli kolme, ja ne mittasivat kuutta eri arvoa: lämpöä, kosteutta, paine-eroa, tvoc:tä, pienhiukkas- ja hiilidioksidipitoisuutta. Palvelin oli Oulun yliopiston itse ylläpitämä ja sen pilvipalveluna toimi PostgreSQL-tietokannan pohjalle rakennettu Thingsboard.

Työ toteutettiin omatoimisella prototyyppien kehittämisellä palautteen mukaan, kunnes saatiin kelvollinen lopputulos. Thingsboardin tullessa yhä tutummaksi pystyttiin toteuttamaan yhä paremmin yksikön tarkoituksiin sopiva käyttöliittymä. Työ onnistui yleisesti ottaen hyvin ja minimivaatimukset saavutettiin. Valmistettu käyttöliittymä esittelee viimeisimpien olosuhteiden keskiarvot ja muuttaa niiden väriä ihannearvoista poiketessa. Lisäksi arvoa painamalla pääsee sen aikasarjaan. Kuitenkin liittymässä esiintyy pieniä asetteluongelmia joidenkin ikkunakokojen kanssa.

Abstract

Author: Juha Mustonen

Title of the Publication: UI of Condition Monitoring System for Clean Room

Degree Title: Bachelor of Engineering, Information- and Communications Technology, Gaming

Keywords: internet of things, cloud service, thingsboard, user interface

The aim of this thesis was to develop a Condition Monitoring System for Oulu University's Measurement technology research unit's Clean Room in Laboratory 13. It was meant to be built on three components: the sensors, a cloud server and a database. The personal part of this project focused on the User Interface. The interface was to visualize measurement data and inform of any deviations. There was to be three sensors, which measured five elements: temperature, humidity, pressure, TVOC, small particle- and carbon dioxide concentration. Oulu University provided the server, which utilized Thingsboard as a cloud service, built on PostgreSQL database.

The work was done by developing prototypes in accordance to feedback, until a sufficient result was achieved. As Thingsboard became more familiar to work with it became easier to develop a user interface the unit needed. Overall, the work was successful and minimal requirements were met. The prepared interface shows the latest conditional values' averages and changes their colors when they deviate from ideal values. By pressing the value the user can read its time series. However, the interface does have some positioning issues with certain window sizes.

Sisällys

1	Johdanto	1
2	Teoriataustan esittely.....	2
2.1	Esineiden internet.....	2
2.2	Pilvipalvelut.....	2
2.3	Thingsboard	4
3	Thingsboardin rakenne.....	5
3.1	Rule Chains.....	5
3.2	Devices	5
3.3	Dashboards	5
4	Tavoitteet	6
5	Kehitysvaiheet	7
5.1	1. Käyttöliittymäprototyyppi	7
5.2	2. Käyttöliittymäprototyyppi	9
5.3	3. Käyttöliittymäprototyyppi	11
5.4	4. Käyttöliittymäprototyyppi	13
5.5	5. Käyttöliittymäprototyyppi	14
5.6	6. Käyttöliittymäprototyyppi	15
5.7	7. Käyttöliittymäprototyyppi	16
6	Viimeistely versio	18
6.1	Sääntöketju	18
6.2	Laiterekisteri	19
6.3	Käyttöliittymä.....	19
7	Yhteenveto	22
	Lähteet.....	23

Liitteet

Lyhenteet ja termit

Cloud	<i>Pilvi</i> , langaton palvelu joka lähettää, vastaanottaa ja käsittelee dataa siihen liitettyjen laitteiden kanssa
CSS	<i>Cascading Style Sheets</i> , verkko-ohjelmointikieli, jolla määritellään verkkosivujen ulkoasu
HTML	<i>Hypertext Markup Language</i> , verkko-ohjelmointikieli, jolla luodaan verkkosivujen perusrakenne
IoT	<i>Internet of Things</i> , systeemit laitteiden ja pilven välillä
JavaScript	Verkko-ohjelmointikieli, jolla verkkosivut saadaan suorittamaan monimutkaisia toimintoja
JSON	<i>JavaScript Object Notation</i> , tiedon välittämiseen optimoitu tiedostomuoto
TVOC	<i>Total Volatile Organic Compound</i> , haihtuvien orgaanisten yhdisteiden kokonaispitoisuus
UI	<i>User Interface</i> , käyttöliittymä

1 Johdanto

Työharjoitteluni oli Oulun yliopiston Mittaustekniikan yksikössä. Se halusi tutustua alati kasvavaan IoT (*Internet of Things*) -alaan, joten harjoittelu keskittyi IoT-laitteiden teoriaan ja palvelujen testaamiseen. Erityisesti Google Cloudiin sekä Thingsboardiin.

Google Cloudilla onnistuttiin luomaan kokonaisvaltaisten IoT-laiteratkaisujen prototyyppijä. IoT-laite lähetti tietoa Googlen pilvipalvelimelle, joka tallensi sen tietokantaansa. Tietokanta oli yhdistetty Google Data Studio kuvaajapalveluun, joka upotettiin omatekoiselle verkkosivulle, jota päätelaite voi lukea. Onnistuttiin myös saamaan pilven analysoimaan vastaanottamaansa tietoa ja lähettämään vastauksen IoT-laitteelle.

Thingsboardilla onnistuttiin lähettämään tietoa IoT-laitteelta kannettavalle luodulle Thingsboard-palvelimelle. Thingsboardilla on oma käyttöjärjestelmäpalvelunsa, jolla onnistuttiin tietoa visualisoimaan. Palvelin saatiin myös analysoimaan tietoa ja lähettämään vastauksia laitteelle.

Osana IoT-palvelujen käyttöönottoa yksikkö halusi seurantajärjestelmät puhdastiloilleen, joten se tarjosi tilaisuuden tehdä tästä opinnäytetyö. Järjestelmän tuli rakentua neljästä osasta: kolmesta sensorista, pilvestä, käyttöliittymästä ja tietokannasta. Sensorit mittaavat kuutta osa-aluetta: lämpöä, kosteutta, paine-eroa, TVOC:tä (*Total Volatile Organic Compound*), pienhiukkas- ja hiilidioksidipitoisuutta. Yliopiston palvelimelle asetettu Postgres-tietokanta toimii niin Thingsboardin alustana kuin vastaanotetun datan tietokantana. Thingsboard toimii pilvenä ja ylläpitää sisäisiä käyttöliittymätoimintoja. Opinnäytetyön erityisenä painopisteenä oli käyttöliittymäsuunnittelu.

2 Teoriataustan esittely

Opinnäytetyön keskeisimmät aihealueet ovat esineiden internet, pilvipalvelut ja käyttöjärjestelmät.

2.1 Esineiden internet

Esineiden internetillä tarkoitetaan yleisesti systeemejä, joissa tekniset laitteet ovat yhteydessä verkkoon. (Waher 2015, 19) Näin ne voivat esimerkiksi lähettää sensorien lukemaa tietoa analysoitavaksi ja visualisoitavaksi tai vastaanottaa etäohjauskomentoja. (Uppa 2017, 8)

Yleisesti ottaen IoT-laitteet pyrkivät mahdollisimman vähäiseen virrankulutukseen, koska ne toimivat usein paristoilla verkkovirran sijaan. (Uppa, 2017,15) Parhaimmillaan yksi paristo takaa vuosien toimintakyvyn. (Mata, 2015, 27) Tietoa voidaan välittää esim. HTTP- (Waher 2015, 60), CoAP- (Waher 2015, 60) tai MQTT-protokollilla (Waher 2015, 138).

IoT-laitteet voivat lähettää ja vastaanottaa tietoa esim. JSON-muodossa. JSON-tiedostomuoto on kehitetty nimenomaan tiedonvälitykseen. Se muodostuu aaltosulkeiden sisään suljetuista yhdestä tai useammasta nimi-arvo-parista. Parin osat erotetaan kaksoispisteillä, kokonaiset parit pilkuilla. (Uppa 2017, 27)

2.2 Pilvipalvelut

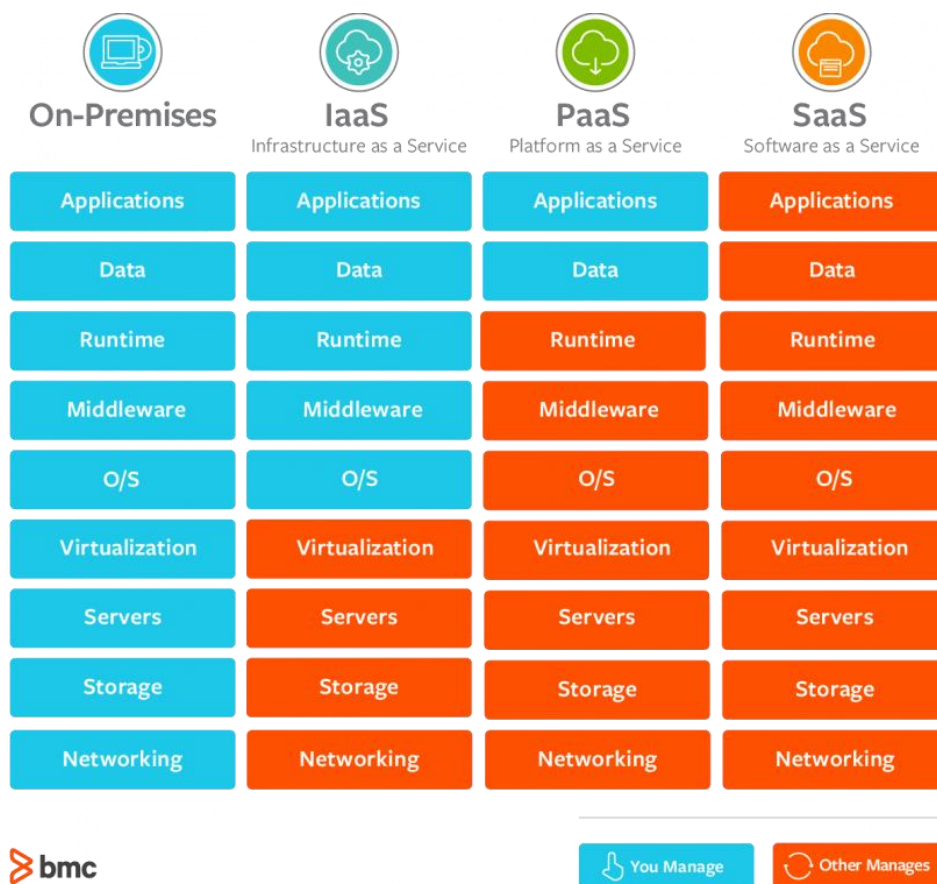
Pilvipalvelu on verkkopalvelimelle tehty palvelu, jolle käyttäjät voivat lähettää dataa esim. varastoitavaksi tai analysoitavaksi.

Pilvipalveluja voidaan jakaa kolmeen ryhmään pilven ylläpitäjän mukaan. Ensimmäisenä ovat **Public Clouds** eli *julkiset pilvet*, joita ylläpitää julkinen palveluntarjoaja. Toisena ovat **Private Clouds** eli *yksityispilvet*, joita käyttäjä itse ylläpitää. Kolmantena ovat **Hybrid Clouds** eli *hybridipilvet*, jotka liittävät yhteen kaksi ensimmäistä tyyppiä. (ZDNet, 2020)

Julkisten pilvipalvelujen käytöllä on monia etuja. Käyttäjän ei tarvitse ostaa, ylläpitää tai päivittää omaa laitteistoa, vaan hän voi vuokrata sen pilvipalveluntarjoajalta. Maksun voi hoitaa käytön

mukaan. Resurssien tarpeen kasvaessa on myös helppoa skaalata omaa toimintaa. Julkiset pilvipalvelut ovatkin hyödyllisiä pienille ja keskisuurille yrityksille. (Microsoft Azure, 2020)

Toinen tapa jaotella pilvipalveluja on palvelun sisällön mukaan. **Infrastructure as a Service (IaaS)** eli *Infrastruktuuri palveluna* tarjoaa koko infrastruktuurin käyttäjälle. **Platform as a Service (PaaS)** eli *alusta palveluna* tarjoaa rajapinnan ohjelmistokehitykselle. **Software as a Service (SaaS)** eli *ohjelmisto palveluna* tarjoaa valmiita ohjelmistoja. (Kuva 1) (BMC Blogs, 2020)



Kuva 1. Eri pilvipalvelutyypin vertailua: <https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-what-is-the-difference-and-how-to-choose/>

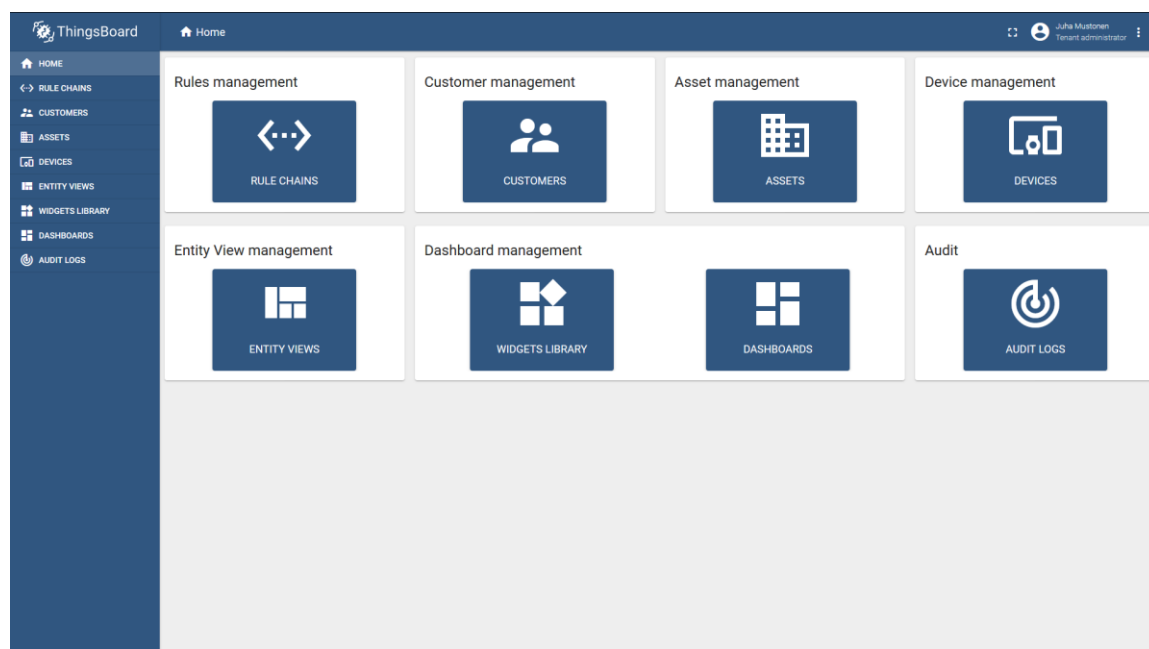
2.3 Thingsboard

Thingsboard on avoimen lähdekoodin IoT-palvelu. (Thingsboard, 2020) Sen pilvi on yksityinen ja asennetaan omalle laitteistolle, mutta se tarjoaa myös internetissä toimivan demoversion.

(Thingsboard Login, 2020) Thingsboard painottaa laitehallintaan sekä datan käsittelyyn ja visualisointiin. Siitä on sekä ilmainen yhteisöversio että maksullinen ammattilaisversio. (Thingsboard Pricing, 2020)

Thingsboardilla on kolmitasoiset käyttöoikeudet. Ylimpänä on **Administrator** eli *ylläpitäjä*, joka luo ja hallinnoi vuokralaisia. Keskimmäisenä on **Tenant** eli *vuokralainen*, joka voi olla esimerkiksi yritys tai organisaatio. (Kuva 2) Tämä luo laiterekisterit, käyttöliittymät ja säätää sääntökettjut. Lisäksi hän luo asiakasrekisterit ja valtuuttaa näille laitteet ja käyttöliittymät. **Clientilla** eli *asiakkaalla* on pääsy itselleen valtuutettuihin rekistereihin ja käyttöliittymiin. (Uppa, 2017, 28)

IoT-laitteet voidaan yhdistää joko MQTT-, HTTP- tai CoAP-protokollilla. HTTP:lle ja MQTT:lle on salausalgoritmit. Käyttäjä voi luoda laiterekistereitä ja resurssirekistereitä, joihin laitteet voidaan liittää.



Kuva 2. Thingsboardin vuokralaisnäkyvän etusivu demo.thingsboard.io/home-sivulla esitettynä

Thingsboardilla on monipuoliset sisäänrakennetut toiminnot käyttöliittymien kehittämiseen. Käyttäjä voi luoda käyttöliittymäpohjia ja valita niihin valmiita pienoishjelmia mm. reaaliaikaisen kuvaajien, viimeisimpien arvojen ja hälytysten kuvaamiseen tai laitteiden etäohjaukseen. Lisäksi käyttäjä voi luoda omia pienoishjelmia. (Thingsboard, 2020)

3 Thingsboardin rakenne

3.1 Rule Chains

Rule Chains eli sääntöketjut käsittelevät kaikki Thingsboardille tulevat viestit. Ne muodostuvat nodeista. Yksi ketjuista on juuriketju, jonka läpi kaikki viestit menevät. Siitä se voidaan halutessa johtaa muihin ketjuihin.

Nodit paitsi tallentavat viestien tiedot voivat myös analysoida ja muokata niitä tai lähettää ne ulkoisille tahoille. (Thingsboard Rule Engine, 2020)

3.2 Devices

Devices on laiterekisteri. Se näyttää yksittäisten laitteiden tiedot, kuten viimeisimmät viestit, attribuutit, ja hälytyslokin. Jokaiselle laitteelle on määritettävä ainakin nimi ja tyyppi, jotka voidaan päättää vapaasti.

Thingsboardilla on kaksi tunnistetapaa, jolla viestin luotettavuus varmistetaan: **access token** eli *poletti* ja **X.509 Certificate** eli *X.509-varmenne*. Tunniste liittää viestin metadataan kyseisen laitteen nimen, jonka perusteella sääntöketju tietää, minkä laitteen viestin se käsittelee. (Thingsboard Devices, 2020)

3.3 Dashboards

Dashboards on käyttöliittymärekisteri. Käyttöliittymät esittelevät laitetietoa muokattavilla diagrammeilla, joko viimeisimpinä arvoina tai aikasarjoina. Käyttöliittymät liitetään laiterekistereihin **aliaksien** avulla. Alias voi vastata yhtä tai useampaa laitetta. Käyttöliittymät voidaan määrittää asiakastason käyttäjien nähtäviksi, tai ne voidaan tehdä julkisiksi, jolloin kuka tahansa linkin saanut näkee käyttöliittymän. (Thingsboard Dashboards, 2020)

4 Tavoitteet

Projektin tavoitteena oli luoda automaattinen olosuhteiden valvontajärjestelmä Laboratorio 13:n puhdastiloille. Järjestelmän tuli visualisoida olosuhdearvoja käyttöliittymällä ja tehdä hälytyksiä. Mitattavia arvoja oli kuusi: lämpötila, kosteus, paine-ero, TVOC, pienhiukkas- ja hiilidioksidipitoisuus. Järjestelmän kuului rakentua kolmesta osasta: sensoreista, pilvestä ja käyttöliittymästä. Pilvenä käytettiin yksikön omalle palvelimelle asennettua Thingsboardin IoT-alustaa, jolla oli sisäinen palvelunsa käyttöliittymän luomiselle ja käytölle.

Käyttöliittymän tuli ilmaista sensorien lähettämää mittausdataa kahdella tavalla: viimeisimpien arvojen keskiarvoina ja aikasarjoina. Lisäksi käyttäjällä piti olla mahdollisuus tarkastella aiempia mittausarvoja. Olennaista oli että käyttäjä pystyy havaitsemaan mahdolliset standardiarvopoikkeamat ja niiden vakavuus välittömästi.

Järjestelmän piti myös lähettää hälytyksiä edellä mainituista standardiarvopoikkeamista, mutta tämä ei ole olennaista käyttöliittymäsuunnittelun kannalta.

5 Kehitysvaiheet

Käyttöliittymän kehittämistä aloittaessa oli Thingsboard jo asennettu yliopiston palvelimelle. Koska sensoreita ei ollut vielä asennettu, laitettiin mikropiiri lähettämään satunnaisia arvoja, jotta pystyttäisiin testaamaan järjestelmän toimivuutta reaaliaikaisesti. Puutteellisten tietojen ja erinäisten väärinkäsitysten johdosta ei alussa ollut täydellistä kuvaa sensorien määrästä ja lähettämistä arvoista.

Käyttöliittymän suunnittelussa oli varsin vapaat kädet. Käytännössä työskentely oli lähinnä prototyyppien itsenäistä suunnittelua ja kehittämistä. Valmiit prototyypit esiteltiin toimeksiantajalle, joka ehdotti muutoksia. Tätä prosessia jatkettiin kunnes oltiin tyytyväisiä.

5.1 1. Käyttöliittymäprototyyppi

Käyttöliittymätoimintoja testatessa syntyi 1. prototyyppi. (Kuva 3)



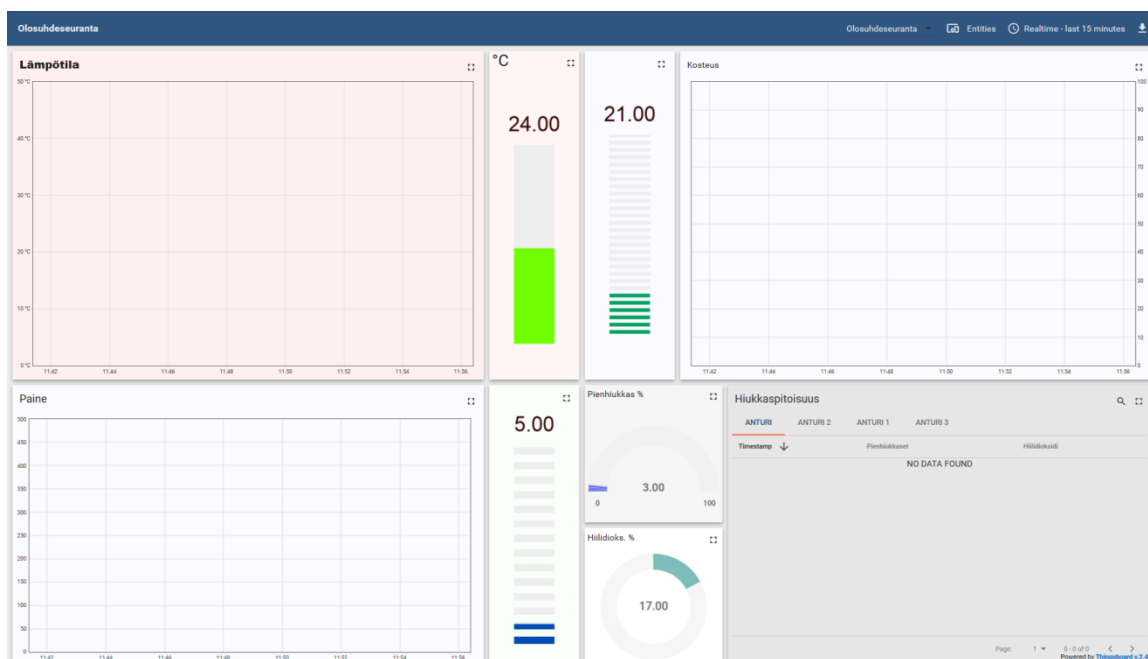
Kuva 3. 1. prototyypin kaava

1. Prototyypissä oli Thingsboardin valmis laitevalikko. (Kuva 4)

Name	Active
Sensor 1	false
Sensor 2	false
Sensor 3	false

Kuva 4. 1. prototyypin päävalikko

Käyttäjän valitessa laitteen se näytti kyseisen laitteen kaikkien mitattavien arvojen aikasarjat ja viimeisimmät arvot. (Kuva 5)

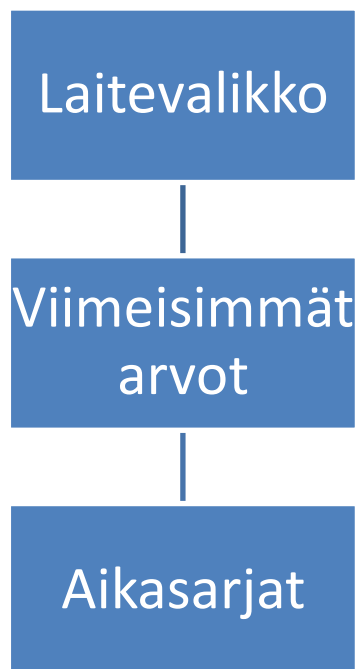


Kuva 5. 1. prototyypin aikasarjanäyttö. Mikropiiri ei ollut kuvan ottamishetkellä kytketty, joten vain viimeisimmät arvot näkyvät

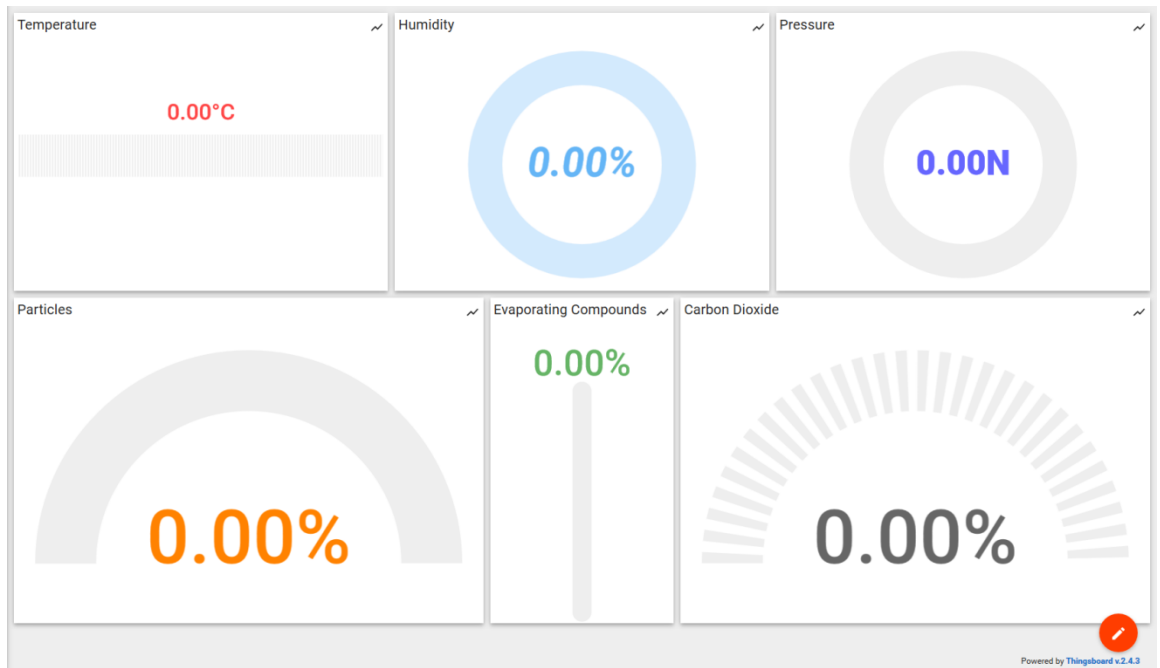
Johtopäätöksenä tämä oli kuitenkin liian tiivis ja vaikealukuinen.

5.2 2. Käyttöliittymäprototyyppi

Seuraavaksi ideoitiin käyttöliittymä, jossa ylimpänä tasona olisi myös 1. Prototyypin mukainen laitevalikko. (Kuva 4) Käyttäjän valitessa laitteen avautuisi kyseisen laitteen jokaista viimeisintä arvoa näyttävä valikko. Jokaisen viimeisimmän arvon lohkon nurkassa olisi kuvake, joka avaisi kyseisen arvon aikasarjan. (Kuva 6, Kuva 7) Toivottiin myös että JavaScriptillä pystyttäisiin linkkaamaan koko lohko, jos se vain ehdittäisiin tehdä.



Kuva 6. 2. prototyypin kaava

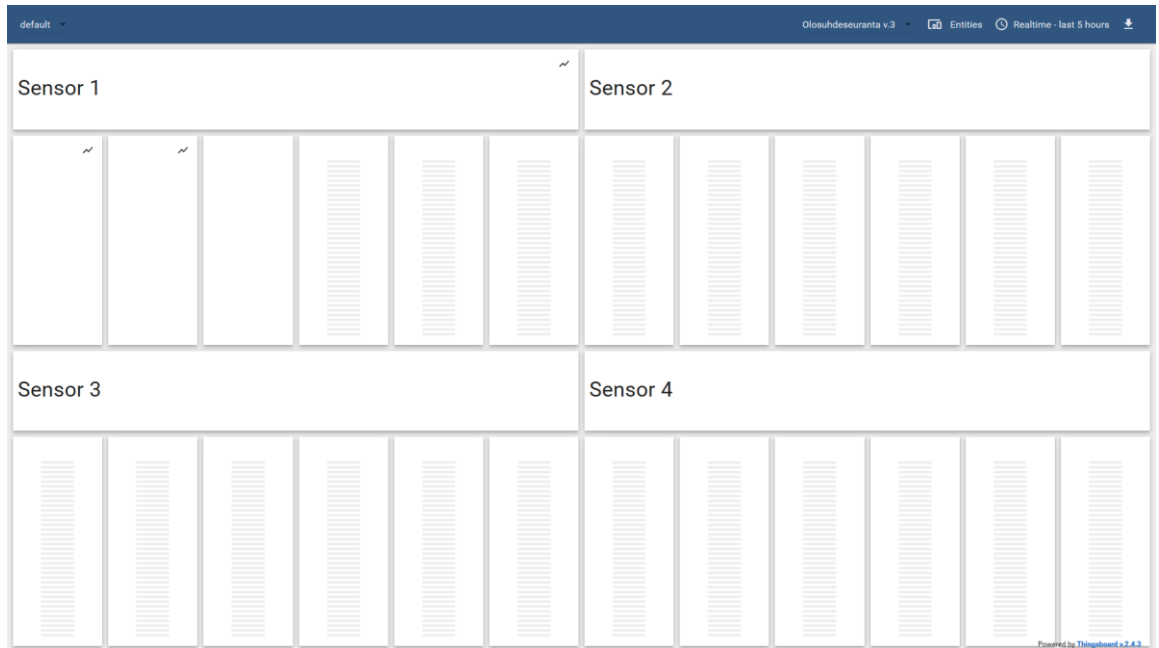


Kuva 7. 2. prototyypin viimeisimpien arvojen näyttö. Tehty Thingsboardin internet-demolla

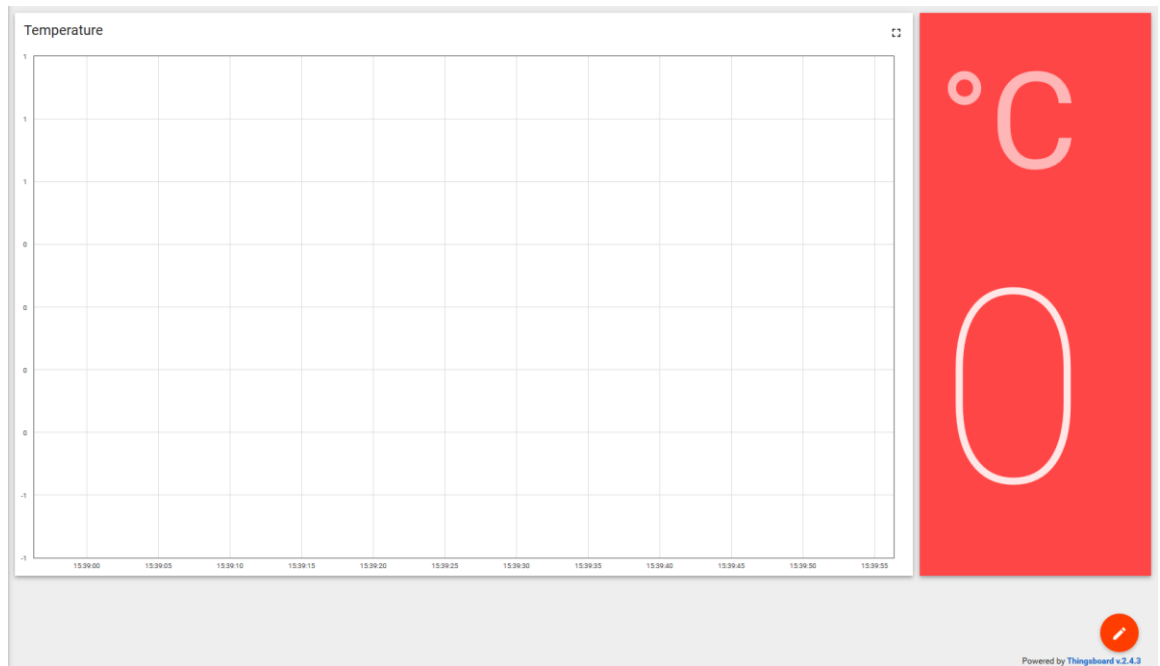
Idea esitettiin toimeksiantajalle, mutta tämä halusi käyttöliittymän josta näkisi jokaisen huoneen ja sensorin tilan yhdeltä ruudulta. Lisäksi haluttiin että palkit visualisoisivat useamman viimeisimmän arvon keskiarvoja pelkän viimeisimmän arvon sijaan.

5.3 3. Käyttöliittymäprototyyppi

Dashboardsilla ei ollut valmista pienoishjelmaa, joka voisi sekä esitellä useampia arvoja, että vaihtaisi väriä niiden mukaan. Lopulta päädyttiin luomaan 3. prototyyppi, jolla olisi kaikkien sensorien arvot vierekkäisinä palkkeina. (Kuva 8) Palkkien yläkulman kuvakkeita painamalla pääsisi aikasarjaan. (Kuva 9)



Kuva 8. 3. prototyypin viimeisimpien arvojen näyttö. Bugin takia kolme ensimmäistä palkkia eivät näy



Kuva 9. 3. prototyypin aikasarja

Tämän ongelma oli että vierekkäiset nimipalkit menivät päällekkäin näytön kavetessa, esim. mobiililaitteella. Ehdotettiin että luotaisiin jokaiselle sensorille yksi palkki joka muuttaisi väriä jos yksikin arvo poikkeaa hyväksyttävien ulkopuolelle, tai että luotaisiin mobiilille erillinen käyttöliittymä.

Toimeksiantaja ei halunnut useampaa käyttöliittymää, joten piti pystyä sovittamaan sama käyttöliittymä joka näytölle. Tässä vaiheessa esiteltiin sensorit (Liite 1. Olosuhdeanturin mittausdata) ja ihannearvot. (Liite 2. Raja-arvot) Jokaiselle arvolle on kolme tasoa, hyvä (*vihreä*), välttävä (*keltainen*) ja huono (*punainen*).

3. Prototyyppi oli tehty sillä oletuksella että kaikilla arvoilla olisi optimaalinen keskiarvo. Sen sijaan osalla arvoista oli nolla-arvo optimaalinen, mitä matalampi sen parempi. Näin ollen osan palkeista tulisi myös olla nollassa, mikä saattaisi aiheuttaa käyttäjällä hämmennystä siitä, onko kaikki kunnossa. Päätettiin siirtyä takaisin aiempaan ideaan erilaisista viimeisimpien arvojen näyttäjistä.

5.4 4. Käyttöliittymäprototyyppi

4. prototyyppiä varten esiteltiin järjestelmän sensorit (Liite 1) ja raja-arvot (Liite 2). Se seurasi pääosin 2. prototyypin ideaa. Jokainen arvo näkyisi päävalikosta (Kuva 10) ja sen yläkulman kuvaketta painamalla pääsisi kyseisen arvon aikasarjaan. Sensorien nimet näkyisivät yläkulmasta.



Kuva 10. 4. prototyypin päävalikko

Prototyyppiä esitellessä toimeksiantajalle kävi ilmi, että sensorit oli esiteltäessä ymmärretty väärin. Oli saatu käsitys, että jokaiselle huoneelle olisi oma sensori, joka mittaisi vain omia arvojaan. Sen sijaan jokaiselle huoneelle oli yksi jokaista sensoria, ja kaikista huoneista mitattiin kaikkia arvoja. Näin ollen 4. prototyyppi olisi tarvinnut kolme kertaa enemmän arvonnäyttäjiä kuin kuvassa, eivätkä ne olisi millään mahtuneet.

5.5 5. Käyttöliittymäprototyyppi

5. prototyyppiä suunniteltaessa otettiin huomioon aiemmin esitetyt vaatimukset, että päävalikosta näkyisi heti ovatko olosuhteet kunnossa. Päätettiin, että tämä parhaiten tämä ilmenisi luomalla Rule Chainissa jokaiselle huoneelle kustomoitu condition (*olotila*)-arvo, joka määritettäisiin ihannearvosta voimakkaimmin poikkeavan arvon mukaan. Jos yksikin arvo poikkeaisi ihannearvostaan, menisi huoneen condition-arvo keltaiseksi tai punaiseksi. (Kuva 11) Arvovalikko olisi samanmallinen kuin 4. prototyypin päävalikko. (Kuva 10)



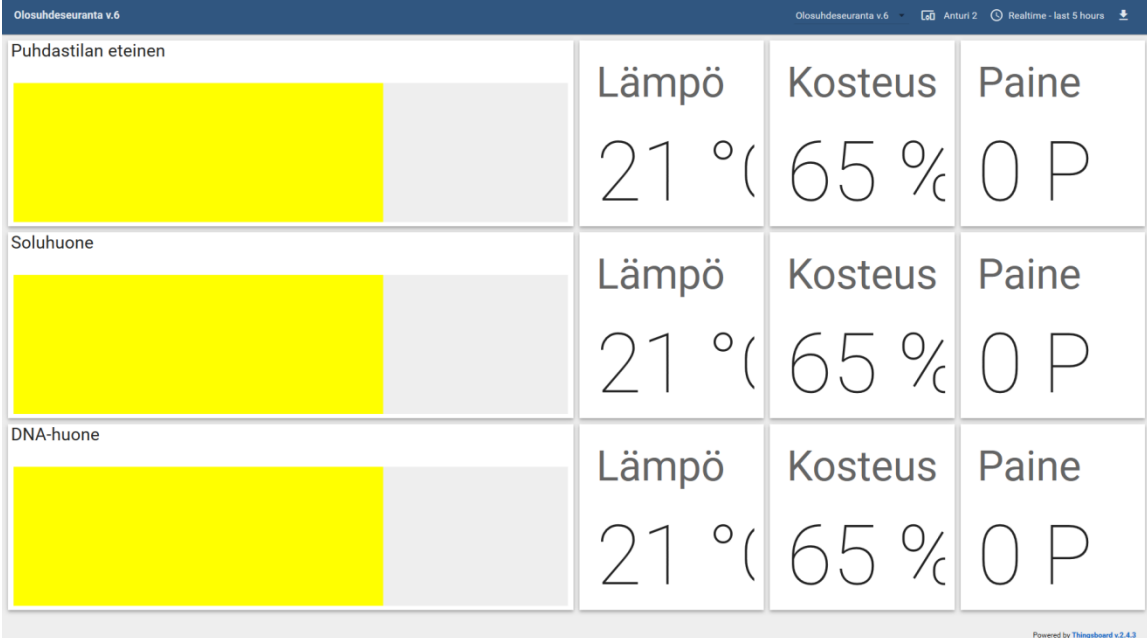
Kuva 11. 5. prototyypin päävalikko

Toimeksiantaja piti tätä kuitenkin liian yksinkertaisena. Hän halusi ainakin jokaisen huoneen lämmön, kosteuden ja paineen näkyvän päävalikossa. Lisäksi hän koki 4. Prototyyppiin pohjautuvan laitevalikon liian sekavana.

5.6 6. Käyttöliittymäprototyyppi

6. Prototyyppi näyttää huoneiden yleistilojen lisäksi jokaisen lämmön, kosteuden ja paineen.

(Kuva 12)



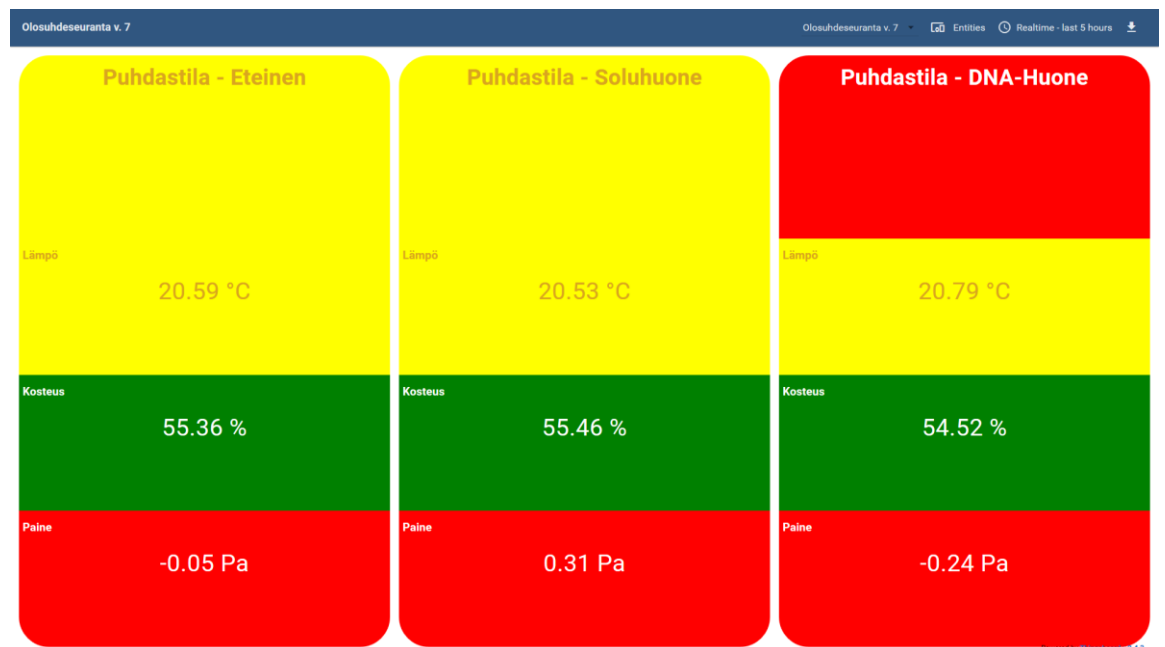
Room	Lämpö	Kosteus	Paine
Puhdastilan eteinen	21 °C	65 %	0 Pa
Soluhuone	21 °C	65 %	0 Pa
DNA-huone	21 °C	65 %	0 Pa

Kuva 12. 6. prototyypin päävalikko

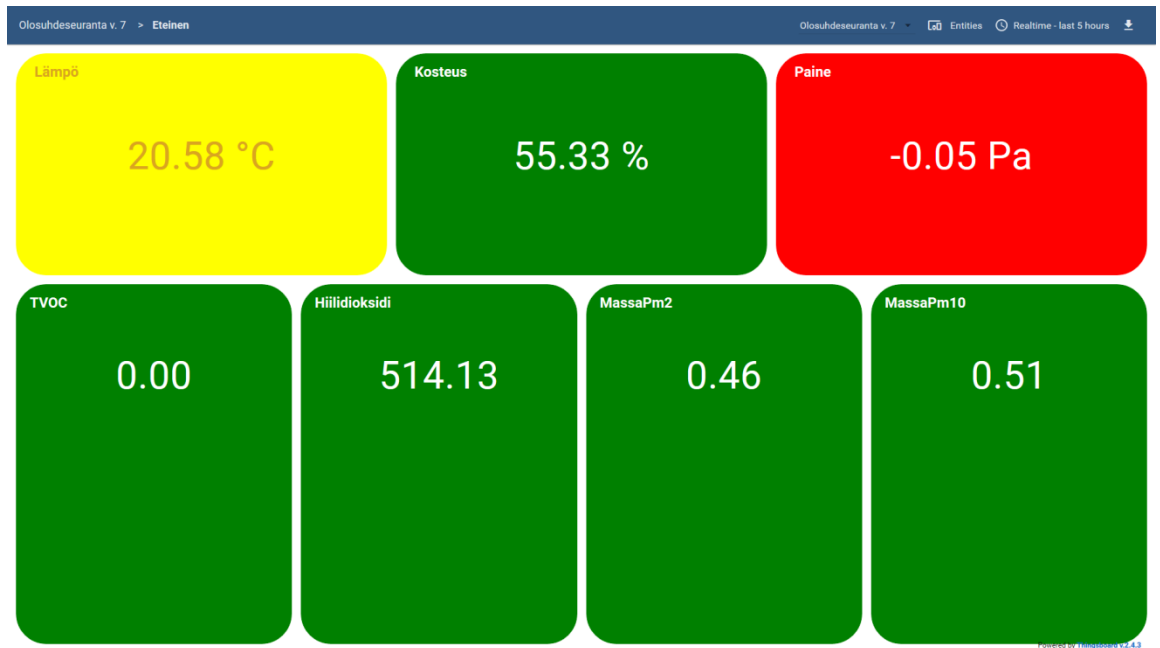
Ratkaisu täytti sinällään aiemmin esitetyt minimivaatimukset, mutta verkko-ohjelmointikielillä muokattavia pienoisohjelmia testattua tajuttiin, että niillä saisi paremmin toimeksiantajan alkuperäiseen ideaan sopivan käyttöliittymän. Niinpä tämä versio hylättiin.

5.7 7. Käyttöliittymäprototyyppi

7. prototyyppi on tehty verkko-ohjelmointikielillä (HTML/CSS/Javascript) muokattavilla pienoisohjelmilla. Tärkein ero aiempiin versioihin verrattuna on viimeisimpien arvojen taustavärien vaihtuminen arvon hyväksyttävyyden mukaan. Tämä on toteutettu HTML:n sisään luodulla Javascriptillä. Huoneita kuvaavat palkit ovat pystyssä, jotta rakenne pysyisi järkevänä myös mobiilinäytöllä. Palkkien ylimpänä ovat huoneiden yleistilat. (Kuva 13) Niitä painamalla pääsee arvovalikkoon, jossa näkyvät kyseisen huoneen kaikki viimeisimmät arvot. Pienhiukkasarvoista on esitelty vain MassaPM2 ja MassaPM10 tilan rajallisuuden vuoksi. (Kuva 14) Yksittäisistä arvoista pääsee aikasarjoihin, jotka on edelleen tehty Thingsboardin valmiilla pienoisohjelmilla. (Kuva 15)

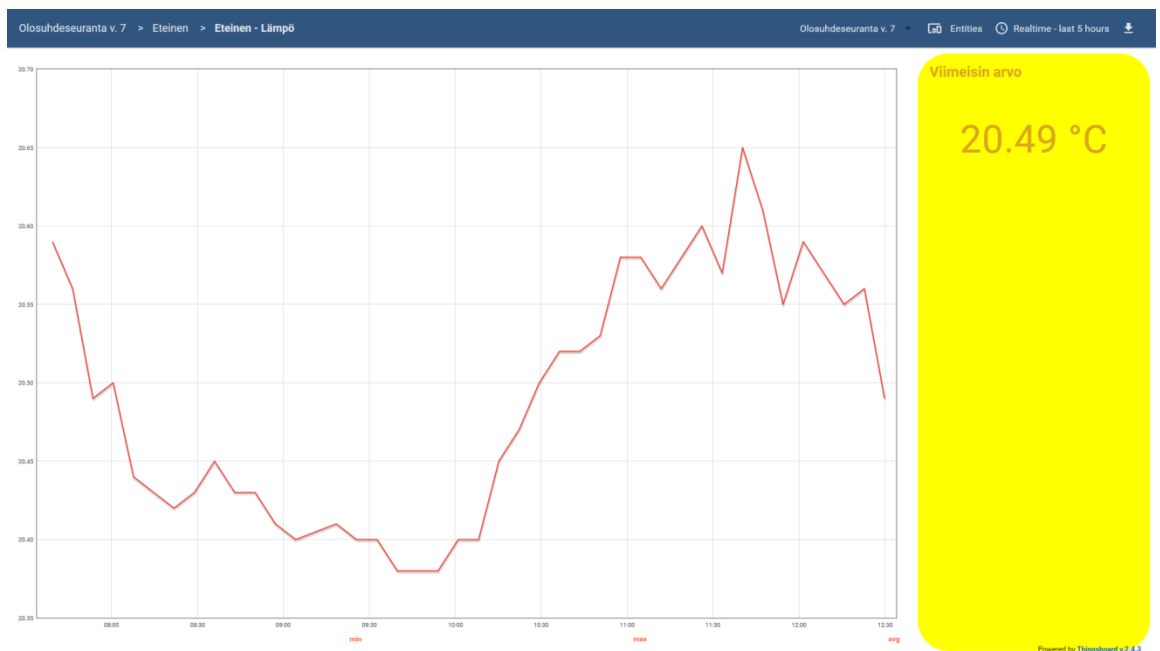


Kuva 13. 7. prototyypin laitevalikko



Kuva 14. 7. prototyypin arvovalikko

Aikasarjan vieressä on viimeisin arvo, lohko vaihtaa väriä sen optimaalisuuden mukaan. (Kuva 15)



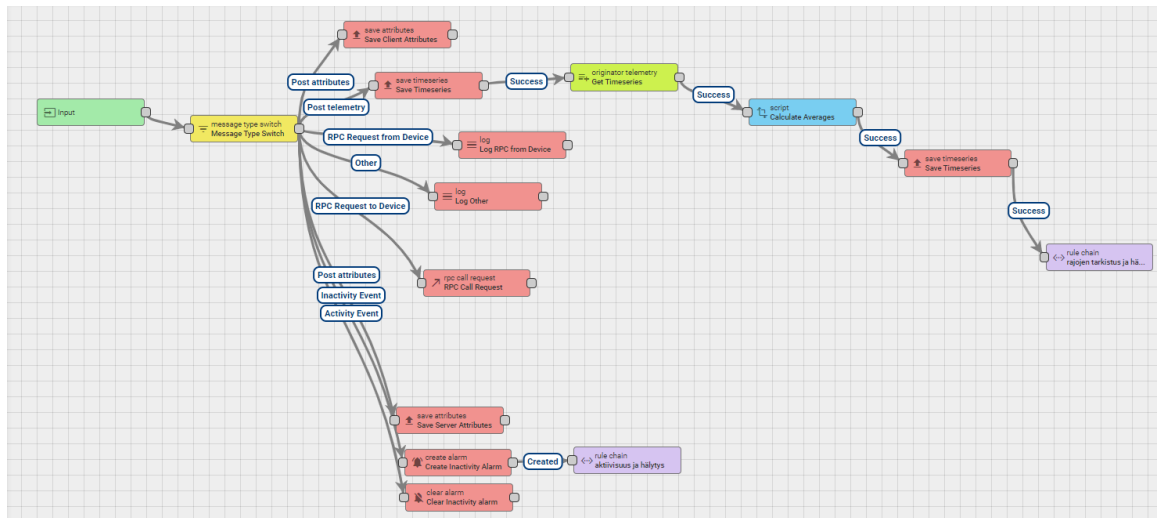
Kuva 15. 7. prototyypin aikasarja, eteisen lämpötila

Koska päävalikossa oli riittävästi tilaa, toimeksiantaja ehdotti että kaikki viimeisimmät arvot voitaisiin liittää päävalikkoon. Näin käyttöliittymästä tulisi kaksitasoinen ja huomattavasti yksinkertaisempi.

6 Viimeistely versio

6.1 Sääntöketju

Sääntöketju pohjautuu Thingsboardin valmiiksi annettuun juurisääntöketjuun. Työtoverien avustuksella onnistuttiin selvittämään, miten viimeisimmät arvot voitiin liittää viestin metadataan. Sääntöketju laitettiin ottamaan jokaisen viestin kymmenen viimeisintä arvoa ja muodostamaan niistä keskiarvon. Jälkeenpäin työtoveri lisäsi nodeja, joilla systeemi saadaan lähettämään hälytys ja sähköposti-ilmoitus arvojen muuttuessa. (Kuva 16)



Kuva 16. Viimeistellyn version sääntöketju

Alla esitellään sensorista saadun tiedon laiterekisteriin kytkemiseen ja analysoimiseen olennaiset nodit:

Input on viestin sisääntulo

message type switch jakaa viestin.

save timeseries tallentaa viestin laiterekisteriin. Tätä on käytettävä kahdesti koska originator telemetry toimii vain kun sen edessä on save timeseries-nodi

originator telemetry ottaa laiterekisteriltä halutun määrän viestejä, tässä tapauksessa kymmenen, ja liittää ne metadataan

script laskee viimeisimpien viestien arvojen keskiarvot ja liittää ne viestiin.

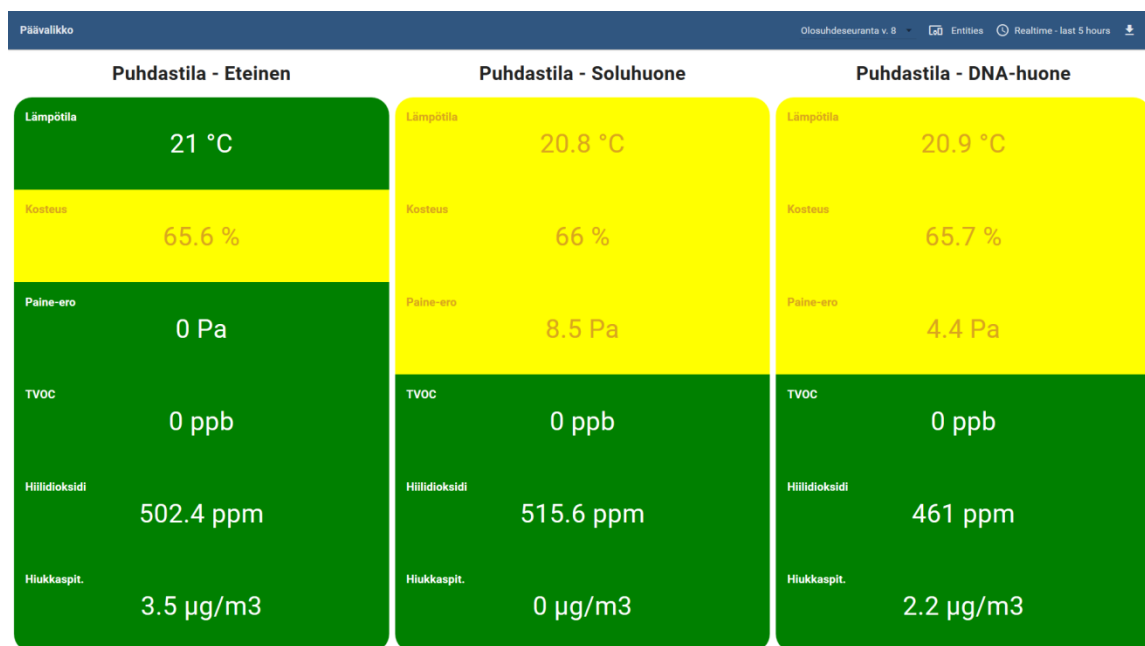
save timeseries tallentaa viestin keskiarvojen kanssa.

6.2 Laiterekisteri

Laiterekisterissä on nimike jokaiselle kolmelle anturille. Niiden nimet ovat Olosuhdeanturi 1, Olosuhdeanturi 2 ja Olosuhdeanturi 3. Tunnisteena käytetään polettia.

6.3 Käyttöliittymä

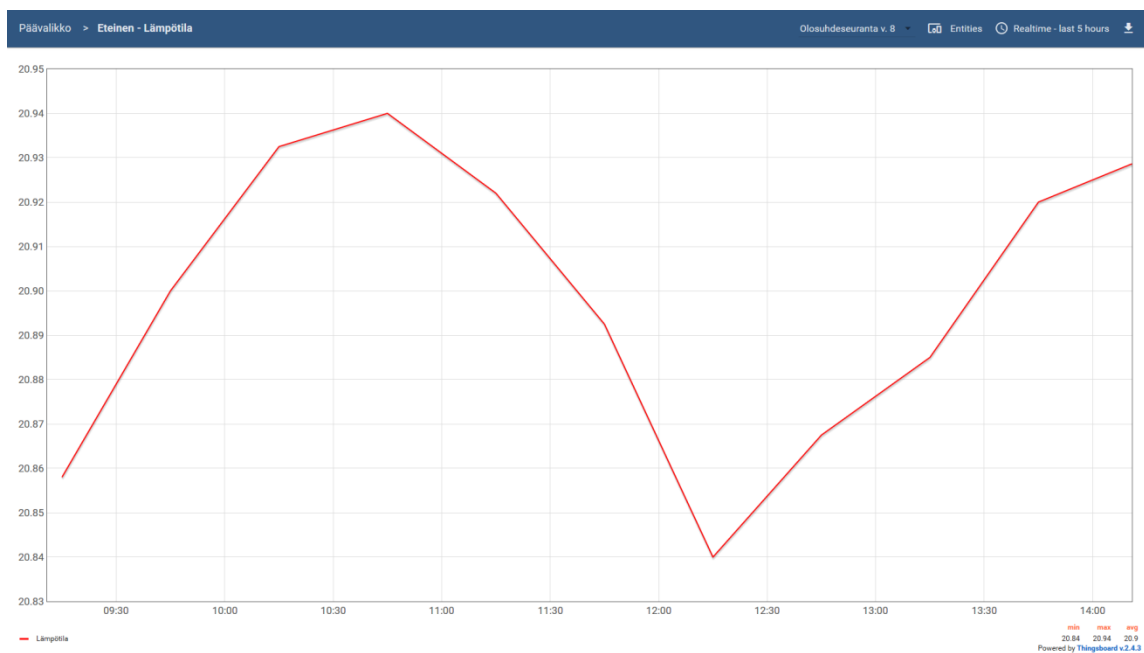
Viimeistellyn version päävalikko näyttää kaikkien kolmen huoneen mitattavien arvojen keskiarvot yhden desimaalin tarkkuudella. Arvoa painamalla pääsee suoraan sen aikasarjakuvaajaan. Arvot on pyöristetty pienoisohjelmassa käytetyllä JavaScriptillä. MassaPM2 on jäänyt hiukkaspitoisuuden kuvauksesta pois tilan rajallisuuden takia. Sen sijaan hiukkaspitoisuus kuvastaa vain MassaPM10:n keskiarvoa. (Kuva 17)



Kuva 17. Viimeistellyn version päävalikko. Arvot muuttavat väriään ihannearvoista poikkeamisen mukaan

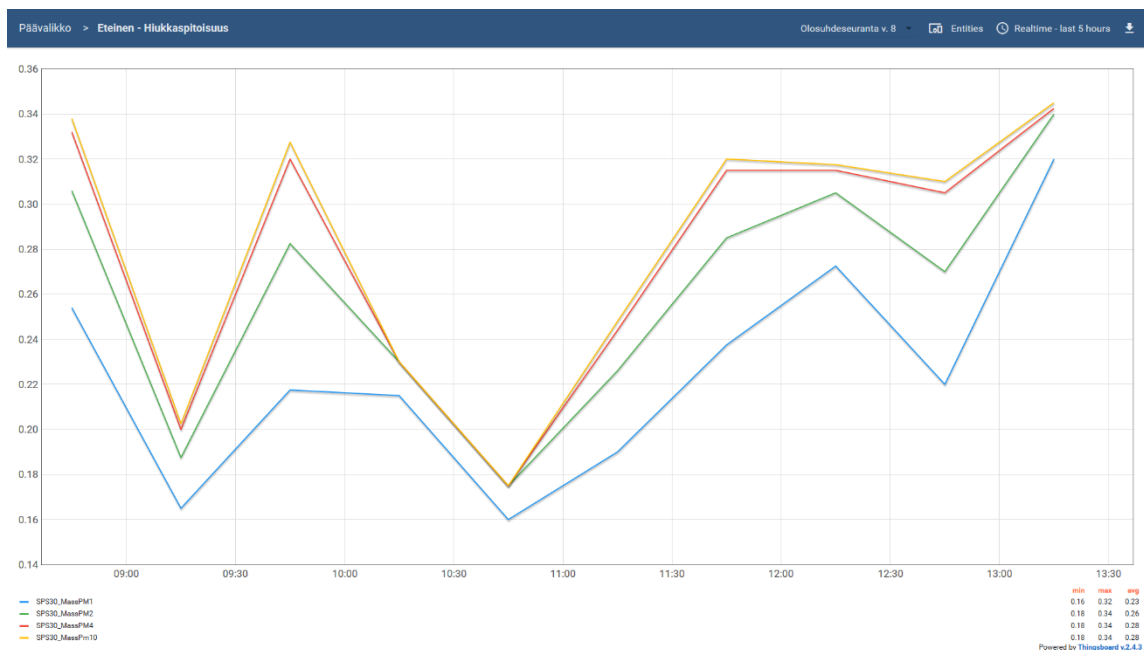
Aikasarja näyttää paitsi aikasarja-arvon halutulta ajalta myös sen pienimmän ja suurimman arvon ja keskiarvon. Kannattaa huomata, että koska päävalikon keskiarvot on muodostettu erikseen

sääntöketjussa, ne eivät vastaa yksittäisen aikasarjan esittämiä keskiarvoja, mikäli nämä on las-
kettu muuten kuin kymmenen viimeisimmän keskiarvon perusteella. (Kuva 18)



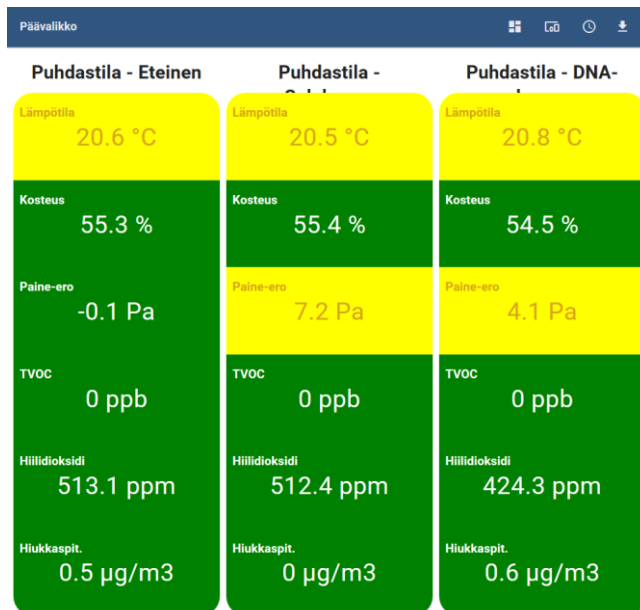
Kuva 18. Viimeistellyn version aikasarja, eteisen lämpötila

Siinä missä muut aikasarjat näyttävät vain yhden arvon, hiukkaspitoisuuden aikasarja näyttää
kaikki hiukkaspitoisuudet (MassaPM1, MassaPM2, MassaPM4, MassaPM10). (Kuva 19)



Kuva 19. Viimeistellyn version aikasarja, eteisen hiukkaspitoisuus

Ikävä kyllä aikaa ei ollut CSS:n toiminnan hiomiseen ikkunakoon vaihdellessa, joten huoneiden nimet menevät joidenkin ikkunakokojen kanssa peittoon. (Kuva 20)



Kuva 20. Viimeistellyn version päävalikko kuvakoon ollessa 1006 × 958, huomaa toisen ja kolmannen huoneen otsikko

Kuitenkin ikkunakoko, jossa tämä bugi ilmenee, on varsin rajattu, joten tämä tuskin merkitsee paljoa järjestelmän käytettävyyden kannalta.

Lisäksi olisi kenties ollut parempi jos arvot oltaisiin eritelty toisistaan selkeämmin esim. esittämällä ne eri fonteilla ja tyyleillä, kuten 2. prototyypissä. (Kuva 7) Aikasarjojen yhteyteen olisi ehkä myös voinut laittaa viimeisimmän arvon ja tilan näyttävän pienoisohjelman, kuten 7. prototyypissä. Tätä tuli kuitenkin pidettyä tarpeettomana ja tilaavievänä.

7 Yhteenveto

Tärkein asia mikä opinnäytetyötä tehdessä tuli opittua, oli ettei vaikeinta ole aina se, miten jonkin voi tehdä, vaan että onko ratkaisu paras mahdollinen. Kommunikointi ja tiedon varmistaminen ovat ensiarvoisen tärkeitä väärinkäsitysten välttämiseksi. Esimerkiksi 4. prototyypin väärinkäsitys oltaisiin voitu näin välttää. Muistiinpanojen pitäminen ajan tasalla on myös olennaista. Mikäli kustomoitavia pienoisohjelmia olisi testattu sinnikkäämmin projektin aiemmassa vaiheessa, olisi ne voitu ottaa nopeammin käyttöön. Tällöin aikaa olisi jäänyt enemmän käyttöliittymän hiomiseen.

Thingsboardin Dashboards tarjosi mainiot työkalut käyttöliittymän kehittämiseen, mutta vaati runsaasti omaa pohdintaa mieltä, millainen lopputulos olisi paras mahdollinen käytettävyyden kannalta. Kuitenkin se voisi painostaa tietyillä tavoilla enemmän muokattavuuteen. Käyttöliittymän eri ikkunoiden linkkaaminen toisiinsa projektin vaatimalla tavalla vaati linkin luomisen ylänurkkaan aiempien projektien tavoin, sitten sen avaamisen ja tallentamisen ulkoisesti.

Käyttöliittymäsuunnitteluun liittyvää taustamateriaalia ei löytynyt paljoa, joten optimaalisen UI:n periaatteita jouduttiin miettimään paljolti itse. Lopputulos lienee kuitenkin riittävän toimiva. CSS-kielen käyttäytyminen on muutenkin toisinaan ennalta-arvaamatonta, kuten viimeistellyn version ikkunakoon vaihtaminen osoitti. (Kuva 20) Kuten kaikessa ohjelmoinnissa, verkko-ohjelmoinnissa on hyvä varata aikaa kehitykseen ja testaukseen.

Lähteet

Dashboards | Thingsboard, viitattu 2.2.2020 <https://thingsboard.io/docs/user-guide/ui/dashboards/>

Devices | Thingsboard, viitattu 2.2.2020 <https://thingsboard.io/docs/user-guide/ui/devices/>

External Nodes | ThingsBoard. 2020. 17.2.2020
<https://thingsboard.io/docs/user-guide/rule-engine-2-0/external-nodes/>

Installing ThingsBoard on Windows. 2020. 17.2.2020
<https://thingsboard.io/docs/user-guide/install/windows/>

Mata, T. (2015) *Esineiden Internet standardit ja protokollat*. Opinnäytetyö. Tampereen Ammattikorkeakoulu. <https://www.theseus.fi/handle/10024/92798>

Pricing | ThingsBoard. 2020. 17.2.2020
<https://thingsboard.io/pricing/>

Getting Started with Rule Engine | Thingsboard, viitattu 18.12.2020 <https://thingsboard.io/docs/user-guide/rule-engine-2-0/re-getting-started/>

SaaS vs PaaS vs IaaS: What's The Difference & How To Choose – BMC Blogs. 2019. Viitattu 12.11.2020. <https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose/>

ThingsBoard - Open-source IoT Platform. 2020. Viitattu 7.2.2020 <https://thingsboard.io/>

Uppa, J. (2017) *Esineiden Internet ja IoT-alustat*. Opinnäytetyö. Tampereen Ammattikorkeakoulu. <https://www.theseus.fi/handle/10024/138552>

Waher, P. (2015) *Learning Internet of Things*. Birmingham: Packt Publishing

What is Cloud Computing? A Beginner's Guide | Microsoft Azure. 2020. Viitattu 7.2.2020
<https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/>

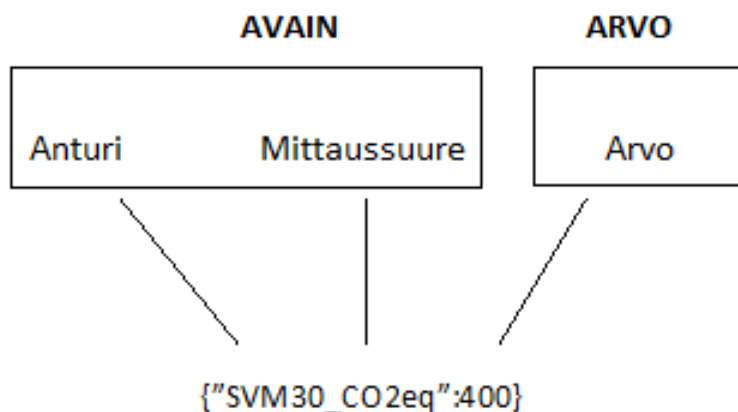
What is cloud computing? Everything you need to know about the cloud explained | ZDNet. 2020. Viitattu 10.11.2020 <https://www.zdnet.com/article/what-is-cloud-computing-everything-you-need-to-know-about-the-cloud/>

Olosuhdeanturilta lähetettävä data Thingsboard -palvelimelle

Sisällys

1. Lähetettävän anturidatan formaatti

Thingsboard -palvelimelle lähetettävä data sisältää JSON -formaatin mukaisen AVAIN-ARVO -yhdistelmän (esimerkki 1.), jossa AVAIN on tietotyyppiltään merkkijono (String) ja sisältää anturin nimen, sekä alaviivalla erotettuna anturin mittaaman suureen. ARVO on tietotyyppiltään joko kokonaisluku tai desimaaliluku ja ilmoittaa anturin mittaussuureen lukuarvon. Lukuarvojen yksiköt ovat listattuna erikseen tässä dokumentissa jokaisen anturin omassa spesifikaatiotaulukossa.



(Esimerkki 1. Anturidata JSON -formaattissa)

2. Käytettävät anturit:

- Sensirion SVM-30

https://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/9_Gas_Sensors/Datasheets/Sensirion_Gas_Sensors_SVM30_Datasheet.pdf

- Sensirion SDP-810

<http://www.farnell.com/datasheets/2611776.pdf>

- Sensirion SVM-30

https://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/9.6_Partiate_Matter/Datasheets/Sensirion_PM_Sensors_SPS30_Datasheet.pdf

2.1 SVM-30

Mittaussuureet:

- TVOC
- Etanoli
- H₂
- CO₂eq/Hiilidioksidiekvivalentti
- Lämpötila
- Ilman suhteellinen kosteus

2.1.1 Mittaussuureet, mittausvälit ja yksiköt

Mittaus suure	Mittausväli	Yksikkö/suhdeyksikkö
TVOC	0 - 60000	ppb
CO ₂ eq	400 - 60000	ppm
H ₂	0 - 1000	ppm
Etanoli	0 - 1000	ppm
Lämpötila	-20 - +85	°C
Suhteellinen kosteus	0 - 100	%RH

(Taulukko 1. SVM-30 mittaussuureet ja yksiköt [1.])

2.1.2 JSON -formaatin avainosat

JSON -avain	Selitys	JSON -arvon tietotyyppi
"SVM30_TVOC"	TVOC	Int
"SVM30_CO2eq"	Hiilidioksidiekvivalentti	Int
"SVM30_H2_signal"	H ₂ (Vety)	Int
"SVM30_Ethanol_signal"	Etanoli	Int
"SVM30_Temperature"	Lämpötila	Int
"SVM30_Relative_humidity"	Ilman suhteellinen kosteus	Int

(Taulukko 2. SVM-30 JSON -avainosat)

2.2 SPS-30

Mittaussuureet:

- Pienhiukkasten lukumääräpitoisuus
(hiukkaskokoluokille PM0.5, PM1.0, PM2.5, PM4, PM10)
- Pienhiukkasten massapitoisuus
(hiukkaskokoluokille PM1.0, PM2.5, PM4, PM10)

2.2.1 Mittaussuureet, mittausvälit, hiukkaskokoluokat ja yksiköt

Mittaus suure	Mittausväli	Yksikkö	Hiukkaskokoluokka	Hiukkaskoko	Yksikkö
Massapitoisuus	0 - 1000	µg/m ³	PM1.0	0.3 - 1.0	µm
			PM2.5	0.3 - 2.5	µm
			PM4.0	0.3 - 4.0	µm
			PM10.0	0.3 - 10.0	µm

Lukumääräpitoisuus	0 - 3000	1/cm ³	PM0.5	0.3 - 0.5	µm
			PM1.0	0.3 - 2.5	µm
			PM4	0.3 - 4.0	µm
			PM10	0.3 - 10.0	µm

(Taulukko 3. SPS30 mittaussuureet ja yksiköt [2.])

2.2.2 JSON -formaatin avainosat

JSON -avain	Selitys	JSON -arvon tietotyyppi
"SPS30_MassPM1"	PM1.0 massapitoisuus (µm/m ³)	Float
"SPS30_MassPM2"	PM2.5 massapitoisuus (µm/m ³)	Float
"SPS30_MassPM4"	PM4.0 massapitoisuus (µm/m ³)	Float
"SPS30_MassPM10"	PM10.0 massapitoisuus (µm/m ³)	Float
"SPS30_NumPM0"	PM0.5 lkm*(1/cm ³)	Float
"SPS30_NumPM1"	PM1.0 lkm*(1/cm ³)	Float
"SPS30_NumPM2"	PM2.5 lkm*(1/cm ³)	Float
"SPS30_NumPM4"	PM4.0 lkm*(1/cm ³)	Float
"SPS30_NumPM10"	PM10.0 lkm*(1/cm ³)	Float
"SPS30_NumPMPartSize"	"Typical Particle Size [µm]" [2.]	Float

(Taulukko 4. SPS-30 JSON -avainosat)

2.3 SDP-810

Mittaussuureet:

- Paine-ero
- Lämpötila

2.3.1 Mittaussuureet, mittausvälit ja yksiköt

Mittaussuure	Mittausväli	Yksikkö
Paine-ero	-500 - +500	Pa
Lämpötila	-40 - +85	°C

(Taulukko 5. SDP-810 mittaussuureet ja yksiköt [3.])

2.3.2 JSON -formaatin avainosat

JSON -avain	Selitys	JSON -arvon tietotyyppi
"SDP810_Differential_pressure"	Paine-ero	Float
"SDP810_Temperature"	Lämpötila	Int

(Taulukko 6. SDP-810 JSON-avainosat)

LÄHTEET

[1.] SENSIRION.COM / SVM-30:

https://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/9_Gas_Sensors/Datasheets/Sensirion_Gas_Sensors_SVM30_Datasheet.pdf

[2.] SENSIRION.COM / SPS-30:

[https://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/9.6 Particulate Matter/Datasheets/Sensirion PM Sensors SPS30 Datasheet.pdf](https://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/9.6_Partiate_Matter/Datasheets/Sensirion_PM_Sensors_SPS30_Datasheet.pdf)

[3.] SENSIRION.COM / SDP-810:

<http://www.farnell.com/datasheets/2611776.pdf>

Raja-arvot

Lämpötila:

- Hyvä (vihreä): -23°C
- Välttävä (keltainen) $20-21^{\circ}\text{C}$ tai $23-26^{\circ}\text{C}$
- Huono (pun): $<20^{\circ}\text{C}$ tai $>26^{\circ}\text{C}$

Kosteus:

- Hyvä (vihreä): 20-60%
- Välttävä (keltainen) 10-20% tai 60-70%
- Huono (pun): $<10\%$ tai $>70\%$

Hiilidioksidi:

- Hyvä (vihreä): <760 ppm
- Välttävä (keltainen) 760 ppm -960 ppm
- Huono (pun): >960 ppm

TVOC:

- Hyvä (vihreä): <100 ppb
- Välttävä (keltainen) 100 ppb- 600 ppb
- Huono (pun): >600 ppb

Hiukkaspitoisuus PM2.5

- Hyvä (vihreä): <12 ug/m³
- Välttävä (keltainen) 12 ug/m³-55,4 ug/m³
- Huono (pun): $>55,4$ ug/m³

Hiukkaspitoisuus PM10

- Hyvä (vihreä): <6 ug/m³
- Välttävä (keltainen) 6 ug/m³ - 15 ug/m³
- Huono (pun): >15 ug/m³

Paine-ero

- Hyvä (vihreä): -15 Pa $<$ paine-ero $<$ -10 Pa
- Välttävä (keltainen) -20 $<$ paine-ero $<$ -15 Pa tai -10 $<$ paine-ero $<$ -5 Pa
- Huono (pun): <-20 Pa tai $>$ -5 Pa