

IoT seuranta

Case: Kahvinkeitin

Antti Pietiläinen

Opinnäytetyö

Toukokuu 2017

Tietotekniikan koulutusohjelma

Tietoverkkotekniikka

Tekijä(t) Pietiläinen Antti	Julkaisun laji Opinnäytetö. AMK Sivumäärä	Päivämäärä 05.2017 Julkaisun kieli Suomi Verkojulkaisulupa myönnetty* x
Työn nimi IoT seuranta Case: Kahvinkeitin		
Tutkinto-ohjelma Tieto- ja viestintäteknikka		
Työn ohjaaja(t) Rantonen Mika, Häkkinen Antti		
Toimeksiantaja(t) Huotari Jouni, JAMK		
Tiivistelmä <p>Työn taustana oli Jyväskylän ammattikorkeakoulun henkilökunnan tarve löytää tapa saada reaaliaikaista tietoa, jonka avulla voi selvittää, onko taukotilassa tarjolla juotavaa. Tehtäväksi muodostui luoda tapa seurata taukotilan Moccamasterin pannun tilaa, jonka avulla voitaisiin määrittellä, kuinka monelle hengelle juomaa riittää. Työn tavoitteena oli kehittää järjestelmä, joka kertoisi keittimen täyttäjälle, kuinka monelle hengelle juomien tulisi riittää. Tällä ehkäistäisiin liiallista tai liian vähäistä juoman valmistamista.</p> <p>Työn toteutustavaksi valittiin prototyypin kehittäminen, jota voidaan käyttää tiedon keräämiseen, käsittelyyn ja esittämiseen. Prototyyppi kerää tarvittavan tiedon esitettäväksi käyttäjille.</p> <p>Työn tuloksena luotu järjestelmä kykenee keräämään tiedon pannun tilasta ja hakemaan tiedon perusteella arvion siitä, kuinka täynnä keittimen pannu kyseisellä hetkellä on. Tämä tieto on käyttäjien saatavissa samoin kuin arvio pannun tilasta.</p> <p>Prototyypin toiminnan pohjalta voidaan todeta, että käyttäjillä on toteutuksen pohjalta parempi mahdollisuus saada tarvitsemansa tieto Moccamasterin pannun tilasta ja siitä, onko tarjolla juotavaa.</p>		
Avainsanat (asiasanat) Kvantunnistus, IBM Watson, IBM Bluemix, Raspberry Pi, Node-RED		
Muut tiedot		

Author(s) Pietiläinen Antti	Type of publication Bachelor's thesis Number of pages	Date 05.2017 Language of publication: Finnish Permission for web publication: x
Title of publication IoT Monitoring Case: Coffeemaker		
Degree programme Information and Communication Technologies		
Supervisor(s) Rantonen Mika, Häkkinen Antti		
Assigned by Huotari Jouni, JAMK		
Abstract <p>The study was based on the needs of the staff at JAMK University of Applied Sciences who wanted to find a way to get real-time information allowing staff members to find out whether there is anything to drink in the Moccamaster coffee pot in the staff lounge/kitchen. The task was to create a way to track the status of the device to determine how many people could get a cup of coffee. The aim of the thesis was to develop a system that would indicate the person filling the pots with coffee and water how many people would be there to have a cup of coffee, which would prevent excessive or too little coffee preparation.</p> <p>The development of a prototype was chosen as a way to implement the study, which can be used for data capture, processing and presentation. The prototype collects the information needed to be presented to the users.</p> <p>As a result of the work, the system created is able to collect information about the state of the Moccamaster and to obtain an estimate of how full the boiler pot is at that time. This information is available to users as well as an estimation of its status (full, almost empty, etc.).</p> <p>Based on the design of the prototype, it can be stated that users, on the basis of the implementation, have better access to the information they need about the information regarding the state of the Moccamaster and whether there is anything to drinking in the teachers' lounge/kitchen.</p>		
Keywords/tags (subjects) Image Recognition, IBM Watson, IBM Bluemix, Raspberry Pi, Node-RED		
Miscellaneous		

Sisältö

1	Työn lähtökohdat	5
1.1	Toimeksiantajan esittely.....	5
1.2	Tehtävänanto	5
1.3	Tavoitteet	6
2	Työn teoriapohja	7
2.1	Esineiden internet	7
2.2	Raspberry Pi.....	7
2.3	Raspbian	9
2.4	IBM Bluemix	9
2.5	IBM Watson.....	10
2.6	Node-RED	10
2.7	MQTT.....	12
3	Suunnitelma	13
4	Toteutus	18
4.1	Laitteiston käyttöönotto	18
4.2	Bluemixin käyttöönotto.....	25
4.3	Kuvan tunnistus.....	28
4.4	Kuvamateriaalin kerääminen	28
4.5	Tunnisteen luonti	29
4.6	Tunnistuksen toteutus.....	33
4.7	Kalenterin tiedon hakeminen.....	36
5	Pohdinta	39
5.1	Työn eteneminen ja haasteet.....	39
5.2	Jatkokehitys ideoita.....	41
5.3	Yhteenveto	42
	Lähteet.....	44
	Liitteet	46

Liite 1.....	46
Liite 2. Toinen liite	47

Kuviot

Kuvio 1. Raspberry Pi 3 Model B	8
Kuvio 2. Node-RED ohjelmointi	11
Kuvio 3. Suunnitelma verkko rakenteesta	13
Kuvio 4. Tunnistuksen palautus	14
Kuvio 5. Pannullinen teetä	15
Kuvio 6. Node-RED-vaihtoehtoja.....	17
Kuvio 7. Salasanan vaihto.....	18
Kuvio 8. Node-RED, asennus sekä käyttöosoitteet	19
Kuvio 9. Node-RED, konsoli käynnistyy	20
Kuvio 10. Node-RED, verkkosivu	20
Kuvio 11. Apache2, oletussivu.....	21
Kuvio 12. Sertifikaatti	22
Kuvio 13. Sertifikaatin konfigurointi Apacheen	23
Kuvio 14. Sivuston SSL-varmenne	24
Kuvio 15. Ääkkösten konfiguraatio.....	25
Kuvio 16. Applikaatiokatalogi.....	26
Kuvio 17. Palveluun yhdistäminen	26
Kuvio 18. Kuvan tunnistuksen yhdistäminen palveluun	27
Kuvio 19. Kvantunnistuksen asetukset	27
Kuvio 20. Tunnistussarja	29
Kuvio 21. Tunnistustapaukset	30
Kuvio 22. Tunnisteen tiedot	31
Kuvio 23. Käyttöön määritellyt tunnisteet.....	31
Kuvio 24. Tunnistuksen palautus	32
Kuvio 25. Kolmannes pannusta juotu	32
Kuvio 26. Kuvanottaminen ja tunnistaminen.....	33

Kuvio 27. Tunnistuksen lopputulos	34
Kuvio 28. Tunnistuksen aloitus.....	35
Kuvio 29. Ajoitus mahdollisuudet.	35
Kuvio 30. Jamk tunnuksilla tunnistautuminen	37
Kuvio 31. Tilanvaraus html	38

Lyhenteet

API	Application programming interface / Ohjelmointirajapinta
DNS	Domain Name System
GB	Gigatavu
GPIO	General-purpose input/output
IoT	Internet of Things / Esineiden internet
IP	Internet Protocol
IPv6	Internet Protocol version 6
JAMK	Jyväskylän Ammattikorkeakoulu
MB	Megatavu
MQTT	Message Queue Telemetry Transport
SD	Secure Digital
SSL	Secure Sockets Layer

1 Työn lähtökohdat

1.1 Toimeksiantajan esittely

Työn toimeksiantaja esittelee itseään verkkosivuillaan: ”Jyväskylän ammattikorkeakoulu on vetovoimainen ja kansainvälinen korkeakoulu. Yksiköitämme ovat ammatillinen opettajakorkeakoulu, hyvinvointiyksikkö, liiketoimintayksikkö sekä teknologiayksikkö. Toimipisteemme sijaitsevat eri puolilla Jyväskylää ja Saarijärven Tarvaalassa.” (Tutustu ja menesty. 2016)

Työ tehtiin Jyväskylän ammattikorkeakoulun teknologiayksikön, tekniikan alan tietojen ja viestintäteknologian alalle. Suuntautumisvaihtoehtoina JAMK:n tietojen ja viestintäteknologian opinnoissa ovat Kyberturvallisuus, mediatekniikka, ohjelmistotekniikka sekä tietoverkkotekniikka.

Jyväskylän ammattikorkeakoulu kuvailee Tietoverkkotekniikan opintoja omilla verkkosivuillaan seuraavasti:

” Internet on tuttu kaikille, mutta kuinka moni tuntee sen toiminnan syvällisesti ja osaa hallita sen liikennettä ja sisältöä? Koulutamme sinusta monipuolisen käytännön tekijän ja kansainvälisen osaajan työelämän tarpeisiin!” (Suuntautumisvaihtoehdot 2017.)

Tieto- ja viestintäteknikan opintojen laajuus on 240 opintopistettä eli noin 3-4 vuotta.

1.2 Tehtävänanto

Työn tehtävänä olisi luoda tapa, jolla voidaan hallita kahvinkeitintä, toimisto ympäristössä. Työllä olisi tarkoitus ehkäistä tilanteita, joissa juomaa ei ole keitetty riittävästi tai sitä on liikaa tarpeeseen nähden. Tämä oli tarkoitus toteuttaa monitoroimalla keitintä sekä hakemalla dataa mahdollisesta kulutuksesta eri lähteistä ja yhdistämällä nämä tiedot keskenään.

1.3 Tavoitteet

Työn tavoitteena oli luoda menetelmä, jolla saadaan ohjattua kahvinkeittoa. JAMK:n Dynamon kampuksella opettajien taukotilassa kahvinkeittäminen on jokapäiväinen tapahtuma, ja aina juotavaa ei riitä kaikille tai sitä on keitetty liikaa.

Kyseisen toteutuksen tuli ehkäistä liiallista sekä liian vähäistä keittämistä ennakoimalla opettajien ja muun henkilökunnan tarpeita. Tämä oli tarkoitus saada aikaan IoT (Internet of Things) -ratkaisulla, joka kerää tietoa henkilökunnasta vapaasti saatavilla olevasta materiaalista sekä heidän omien ilmoitustensa perusteella. Kerättyä tietoa oli tarkoitus käyttää ennustamaan, kuinka monta kupillista pitää keittää, että kaikki saavat kahvinsa haluamaansa aikaan. Tietoa kerättiin myös jo kahvin keittämässä olevan kahvin määrästä, jonka perusteella ennustettiin, kuinka monelle henkilölle jo keitetyt juomat tulisivat riittämään.

Kerättyjen tietojen perusteella osataan kertoa kahvinkeittovuorossa olevalle henkilölle, kuinka paljon juojia on tulossa ja kuinka paljon he mahdollisesti olisivat juomassa.

2 Työn teoriapohja

2.1 Esineiden internet

Esineiden internetistä tai IoT:sta puhutaan, kun jokapäiväisiä laitteita kytketään verkkoon. Ensimmäisiä esimerkkejä tästä olivat yleisesti käyttöön otetut, etäluettavat sähkömittarit. IoT saattaa tarkoittaa jääkaappeja, autoja tai vaikka mökkikameroita. Mikäli laitteella on IP-osoite (Internet Protocol), sen voidaan sanoa olevan osa esineiden intranetiä. IoT-ratkaisuja on alkanut kehittymään suuremmissa määrin IPv6:n (Internet Protocol version 6) myötä, sillä se mahdollistaa yksilölliset osoitteet jokaiselle laitteelle.

Esimerkkinä IoT:sta voidaan pitää sääasemaa, jonka keräämät säätiedot voidaan välittää käsittelyyn ja jakaa käyttäjille. Toisena esimerkkinä voidaan pitää henkilökohtaista laitetta, joka valvoo käyttäjän asuntoa. Laite kerää tietoja sähkönkulutuksesta, ovien avaamisesta, sekä kenen avaimella ovi on avattu tai mitkä laitteet ovat päällä. Nämä tiedot voidaan jakaa omistajalle, joka voi kännykällään sammuttaa lapsilta tietokoneet, jos koti tehtävät ovat tekemättä. Tiedoilla voidaan, myös hälyttää poliisit paikalle, jos asuntoon murtaudutaan.

IoT mahdollistaa tiedon keräämisen käyttäjän haluamasta paikasta ja haluamallaan tavalla, jonka jälkeen tietoa voidaan käyttää, käsitellä tai jakaa hetkessä ympäri maailmaa internetin ansiosta.

2.2 Raspberry Pi

Raspberry Pi on The Raspberry Pi Foundation -nimisen organisaation kehittämä luotokortin kokoinen minitietokone. Raspberry Pi kehitettiin alun perin koulutusikäyttöön. Organisaation tarkoitus oli tarjota halpa nykyaikainen tietokone, jolla voidaan opettaa, vaikka nuorille lapsille tietokoneen käyttöä. (About us 2017.)

Kuvio 1 on nähtävänä tällä hetkellä organisaation kehittynein luottokortin kokoinen tietokone Raspberry Pi 3.



Kuvio 1. Raspberry Pi 3 Model B

Raspberry Pi –tietokoneet ovat lyhyessä ajassa kehittyneet yksiytimisistä harrastelijakorteista moniytimisiksi, jopa teollisuuteen kelpaaviksi hallintalaitteiksi. Vaikka Piin päätarkoitus onkin pysynyt samana, ovat laitteen monipuoliset hallinta-ominaisuudet ja muokattavuus eri tehtäviin onnistuneet nostattamaan Raspberry Piin jopa teollisuussovelluksiin, kuten saksalainen KUNBUS GmbH on tehnyt muokatulla Revolution Pi-teollisuuskortillaan. The Raspberry Pi Foundation on myös julkaissut korttinsa kaaviot ja ilmoittanut aikeistaan tehdä projektista OpenSource-pohjainen. (Revolution Pi 2017.)

Raspberry Pi on vuosien saatossa julkaistu useita eri versioita. Kehityskaaren aikana kortille on myös kehitetty useita oheislaitteita kuten kameroita, näyttöjä ja muita varusteita. Vain osa näistä on itse kehittäjäorganisaation tuotteita. The Raspberry Pi Foundation-organisaation ja tuotteen omilla verkkosivuilla löytyy lista tämän hetki-

sistä ei-kolmannen osapuolen kehittämistä tuotteista. Raspberry Pi on noussut suurrehkoon suosioon kehittäjien keskuudessa, niin muutkin yritykset ovat tehnyt laitteelle sopivia tuotteita. (Products 2017.)

Raspberry Pi on hyvä esimerkki laajenevasta esineiden internetistä, koska Raspberry pysty keräämään sensoritietoja paikoista, joista tietoa ei ole ennen kerätty tai se on ollut hankalaa. Kerätyt tiedot kyetään myös toimittamaan internetin yli käyttäjälle.

2.3 Raspbian

Raspbian on The Raspberry Pi Foundationin oma Raspberry Pi käyttöön tarkoitettu debian-pohjainen Linux-käyttöjärjestelmä, jonka sai ladattua suoraan Raspberry Pin verkkosivuilta. Latauksen jälkeen asennettavan paketin koko on hiukan yli 4 GB, joka tarkoittaa, että Raspberry Pin ”kovalevynään” käyttämä muistikortti tulisi olla vähintään yhtä iso, jotta käyttöjärjestelmän asentaminen olisi mahdollista. SD-kortin pienimmäksi mahdolliseksi kooksi suositellaan 8 GB:n korttia organisaation omilla sivuilla, mutta kortin valinnassa kannattaa myös ottaa huomioon laitteen käyttötarkoitus ja kuinka paljon tietoa Raspberry Pille halutaan tallentaa. Sivustolta elinux.org löytyvän vertailun mukaan Raspberry Pi on mahdollista saada käyttämään jopa 512 GB:n SD-korttia. (SD Cards 2017.)

2.4 IBM Bluemix

Bluemix on IBM:n pilvipalvelualusta, jonka kautta IBM tarjoaa yrityksille verkkoratkaisuja kuvan tunnistuksesta big dataratkaisuihin. Bluemix:n avulla voidaan kehittää ratkaisuja, jotka hakevat, käsittelevät ja/tai tekevät työtehtäviä edellä haetun ja/tai käsitellyn tiedon pohjalta. Tämän tiedon avulla voidaan esimerkiksi säädellä omakotitalon lämmitystä tulevan sään mukaan tai sammuttaa ylikuumentunut etäohjattu laite tietoverkosta. Bluemix mahdollistaa useiden eri toimintojen käyttämisen ja tallentamisen pilvipohjaisesti. Tämä käytännössä tarkoittaa, että tietoihin on pääsy kaikkialta, jossa internetyhteys on mahdollista saavuttaa. (What is Bluemix 2017.)

2.5 IBM Watson

IBM Watson on IBM:n versio tekoälystä. Watson mahdollistaa useita koneellisia tiedon käsittelymenetelmiä. Watsonia on esitelty ja käytetty useissa eri tarkoituksissa kuten esimerkiksi yhdysvaltalaisen ”Jeopardy” televisiovisailun kilpailijana. Watson mahdollistaa myös kuvantunnistuksen, jonka avulla on mahdollista tunnistaa kuvassa esiintyvä objekti. Watson mahdollistaa myös sekä äänen muuntamisen tekstiksi että tekstin muunnon ääneksi. (Go beyond artificial intelligence with Watson 2017.)

IBM Watson kuvantunnistus toimii vertaamalla tunnistettavaa kuvaa aikaisemmin tunnistukselle opetettuihin kuviin. Watsonille voidaan opettaa käyttäjän toimesta niin sanottu ”Custom classifier” eli omatekoinen tunniste, jonka avulla käyttäjä voi luoda itselleen sopivan tunnistejoukon. Watsonin oman custom classifiers ohjeen mukaan oman tunnisteen voi luoda usealla tavalla kuten Curl, Node, Java tai Python (Guidelines for training classifiers 2017).

Kuvantunnistuksen opetus tapahtuu lähettämällä vähintään 10 kuvan zip-paketti, joka sisältää halutun koulutusjoukon. Tämän lisäksi tarvitaan toinen vähintään 10 kuvan zip-paketin, joka sisältää eri koulutusjoukon joko kielteisiä tai toisesta tunnistettavasta asiasta. Zip-paketin koko saa olla maksimissaan 100 MB ja koko opetustapauksen 250 MB IBM Watsonin ”Guidelines for training classifiers ” ohjeiden mukaan. IBM ohjeiden mukaan vanhoja jo tunnistukselle opetettuja tapauksia on mahdollista muokata ja poistaa käyttäen samoja menetelmiä (Visual Recognition 2017).

Ilman käyttäjän opettamia omia erillisiä tunnisteita saadaan tunnistukselta takaisin IBM:n omat vakiotunnisteet asioista. Käyttäjän itse luomat tunnisteet ovat sitä parempia mitä enemmän kuvia tunnisteeseen on lisätty.

2.6 Node-RED

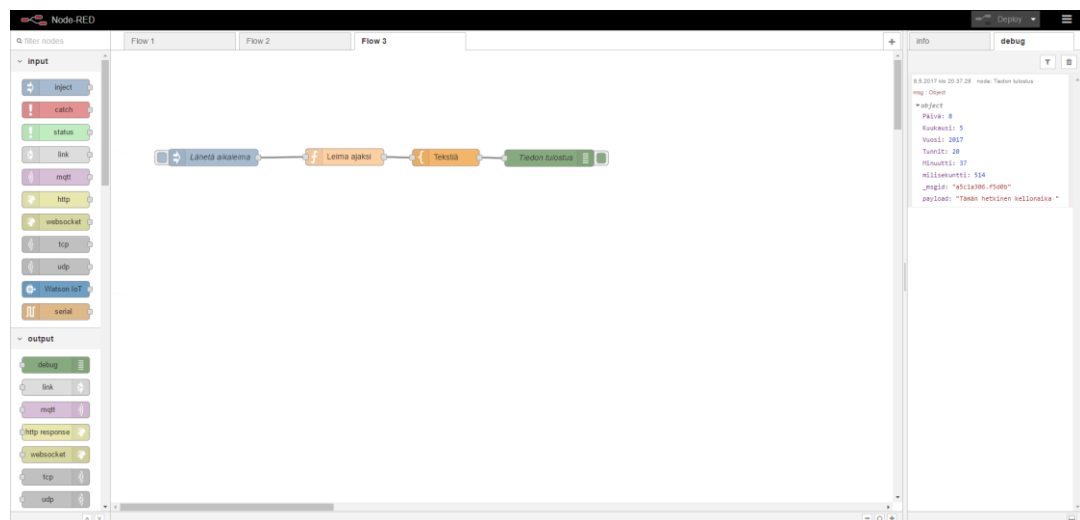
Node-REDin Node.js:n päälle rakennettu kevyt kehitysympäristö, jota voidaan käyttää useilla käyttöalustoilla, kuten BeagleBone Black, Arduino, Android ja Microsoft Azure. Node-RED toimii pilvipalveluna ja paikallisilla laitteilla (Node-RED 2017).

Node-RED:n avulla on helppo päästä käsiksi useisiin Raspberry Pi ominaisuuksiin. Node-RED tarjoaa helposti ymmärrettävän verkkosivupohjaisen käyttöliittymän, jonka avulla käyttäjä voi rakentaa Raspberry Pilleen, vaikka sähköpostiaan lukevan palvelimen, joka sytyttää ledin Raspberry Piin GPIO (General-Purpose input/output) pinneiltä, jos uusi posti on saapunut. Node-REDin avulla voidaan ohjata useita Raspberry Piin ominaisuuksia koodin suorittamisesta rautapuolen komentoihin.

Node-REDin kehittivät Nick O'Leary ja Dace Conway-Jones, jotka työskentelivät IBM:llä. Node-RED oli alunperin ”konsepti MQTT-aiheisiin liittyvien kartoitusten visualisointiin ja manipulointiin”. Projekti vaihtui open-source-pohjalle vuonna 2013 ja oli yhtenä JS Foundation perustus projekteista vuonna 2016.(History 2017.)

Nykyään Node-REDin avulla voidaan rakentaa monimutkaisia hallintaympäristöjä, jotka pystyvät tulkitsemaan sisään tulevaa dataa. Tämä data voidaan ohjata sen jälkeen eteenpäin esimerkiksi vaikka Bluetooth ohjatun robotin hallintaan.

Node-RED:ä käytetään asettamalla ohjelmistossa tarjolla olevia solmuja tai nodeja, joita voidaan linkittää toisiinsa. Kuvio 2 nähdään kuinka Node-RED:llä tehdään ohjelmisto, joka tulostaa päivämäärän ja kellonajan.



Kuvio 2. Node-RED ohjelmointi

Ohjelmoinnissa käytettävät solmut valitaan oikeassa laidassa olevasta listasta, jotka voidaan vetää ruudulle ja käyttää halutulla tavalla ohjelmoinnissa. Ruudulle vedetyt

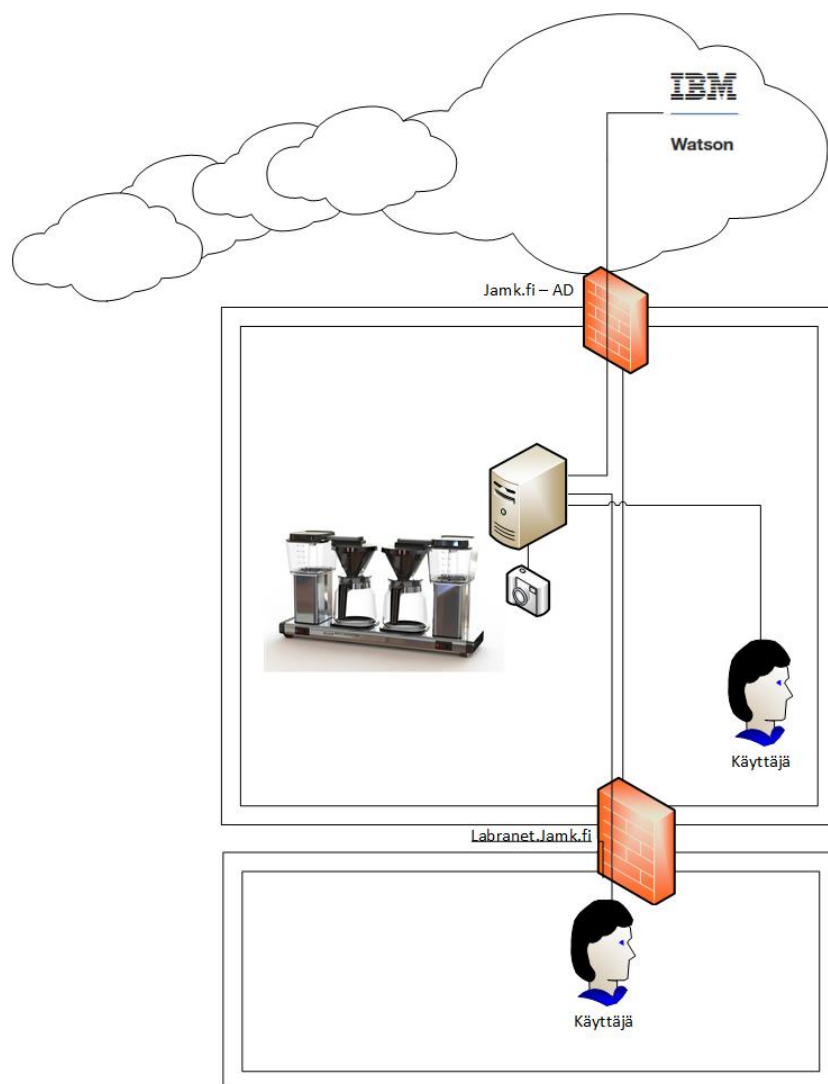
solmut voidaan yhdistä toisiinsa ja ohjelmoida toimimaan halutulla tavalla osana haluttua ohjelmaa.

2.7 MQTT

MQTT tai Message Queue Telemetry Transport on IoT laitteiden välinen keskustelu protokolla. Protokollan kehittivät Andy Stanford-Clark ja Arlen Nipper vuonna 1999. MQTT käyttää vakiona portteja 1883 ja 8883, joista porttia 8883 käytetään yleensä SSL (Secure Sockets Layer) salauksen kanssa. (Frequently Asked Questions 2017.)

3 Suunnitelma

Työn suunnittelu aloitettiin testaamalla IBM Watsonin kuvantunnistusmenetelmää. Kuvantunnistuksen testaamista pidettiin pääasiallisena prioriteettina sillä sitä pidettiin parhaana mahdollisena tapana monitoroida keittimen tilannetta. Samalla etsittiin parasta tapaa käyttämään IBM Bluemix pilvipalvelualustaa, jonka päällä Watsonia suunniteltiin käytettävän. Kuvio 3. nähdään kuinka tiedon siirtäminen IBM Bluemixin suuntaan suunniteltiin toteutettavan.



Kuvio 3. Suunnitelma verkko rakenteesta

Kun saatiin selville, kuinka Watsonia voidaan käyttää tunnistamaan kuvia, alettiin selvittämään, kuinka tunnistuksessa käyttökelpoinen kuva saadaan otettua, sekä kuinka kyseinen kuva saadaan tunnistukseen.

Kuvan ottamiseen käytettiin Raspberry Pi tietokonetta, joka tuntui sopivan tarkoitettuun tehtävään mainiosti. Raspberry Piin käyttöä tuki myös mahdollisuus käyttää kortille tarkoitettua kameramodulia, joka mahdollistaa 5-megapixelin kuvien ottamisen.

Seuraavaksi alettiin selvittää mikä olisi paras tapa saada otettua kuvantunnistuksen tarvitsemat kuvat keittimen tilasta. Julkisen tilan vuoksi kameran asettaminen tilaan, sekä kamerasta saatavan kuvan käyttäminen kuvan tunnistamiseen asettivat haasteita kameran sijoittelulle. Kuvantunnistuksen toimimisen vuoksi kamerasta saatavassa kuvassa ei pitänyt olla paljoo tunnistettavaa.

Tunnistus toimi parhaiten, kun tunnistettava kuva on otettu läheltä tunnistettavaa kohdetta ja kuvien erot ovat selvästi erotettavissa toisistaan. Kuvantunnistuksen käyttöä selvitettyä opetettiin kuvantunnistusta tunnistamaan eroja pannullisessa teetä. Kuvio 4. nähdään, kuinka tunnistus käsittelee kuvaa täydestä pannusta.

```
1491211189334
{"custom_classes": 4, "images": [{"classifiers": [{"classes": [{"class": "T??ysi", "score": 0.987038}], "classifier_id": "Teet_1117810619", "name": "Teet??"}, {"classes": [{"class": "coffee maker", "score": 0.699, "type_hierarchy": "/appliance/coffee maker"}, {"class": "appliance", "score": 0.826}, {"class": "blender", "score": 0.592, "type_hierarchy": "/utensil/blender"}, {"class": "utensil", "score": 0.592}, {"class": "kitchen appliance", "score": 0.53}, {"class": "reddish brown color", "score": 0.865}, {"class": "maroon color", "score": 0.749}], "classifier_id": "default", "name": "default"}], "image": "IMG_20161225_171856.jpg"}], "images_processed": 1}
% Total % Received % Xferd Average Speed Time Time Time Current Dload Upload Total Spent Left Speed
0 0 0 0 0 0 0 --:--:-- --:--:-- --:--:--
0 0 165k 0 0 0 0 0 --:--:-- --:--:-- --:--:--
0 100 165k 0 0 100 165k 0 67584 0:00:02 0:00:02 --:--:--
67598 100 167k 100 1877 100 165k 710 64191 0:00:02 0:00:02 --:--:--
64193 100 167k 100 1877 100 165k 710 64184 0:00:02 0:00:02 --:--:--
64168
```

Kuvio 4. Tunnistuksen palautus

Kuvio 5 on kuvan tunnistukselle lähetetty kuva täydestä pannullisesta teetä.



Kuvio 5. Pannullinen teetä

Kuvio 5. Pannullinen teetä voidaan nähdä, että vaikka kuva on otettu suhteellisen läheltä eikä siinä näy kuin keitin, tunnistus palautti silti asioita kuten sekoitin ja ruokailuväline. Kaukaa otetusta kuvassa niin, että samalla saataisiin kuva puolesta keittiöstä ei olisi mahdollisuutta enää saada selville, mikä on tarkoitus tunnistaa.

Näiden löydösten takia suunnitelmaksi muodostui sijoittaa työssä käytettävä Raspberry Pi, sekä kameramoduli keittimen taakse, jotta kuvassa näkyisi mahdollisimman vähän ylimääräistä.

Kameran kiinnittäminen tiettyyn pisteeseen niin, että se saa otettua tunnistukselle sopivia kuvia todettiin olevan ongelmallista, mikäli keittimen liikehdintää ei saada minimoitua. Julkisessa tilassa sijaitseva keitin on henkilökunnan käytössä ja tästä johtuen keitin saattaa liikkua. Tämä aiheuttaisi ongelman kuvan tunnistukselle kameran

ollessa irrallaan keittimestä, jonka vuoksi Raspberry Pi ja kameramoduli suunniteltiin kiinnitettävän jollain menetelmällä suoraan keittimeen. Tällä pyrittiin ehkäisemään ongelmia kuvan tunnistuksessa.

Tässä vaiheessa alettiin myös suunnitella, kuinka henkilökuntakalenterista saataisiin tietoja järjestelmän käyttöön. Tämän perusteella oli tarkoitus selvittää, koska henkilökunnan jäsenet ovat työpaikalla ja saattaisivat olla juomassa kahvia. Henkilökuntakalenterin tiedot oli tarkoitus saada Jyväskylän ammattikorkeakoulun Microsoft 365-verkkokalenterista, jonne opettajat merkitsevät omat aikataulunsa. Tarkoituksena olisi saada aikaan järjestelmä, joka pystyy kertomaan keittäjälle, kuinka monta henkilöä paikalla mahdollisesti on ja mihin aikaan he olisivat pitämässä taukonsa.

Suunniteltu IBM Bluemixin käyttäminen järjestelmän hallinnassa sekä ylläpidossa johtaa Node-REDin käyttämiseen, joka on IBM Bluemixin IoT-peruskäyttöliittymä. Node-RED mahdollistaa useiden eri toimintojen käyttämisen järjestelmän toiminnassa. Kuvio 6 nähdään listaus suurimmasta osasta IBM Bluemixin Node-RED-ominaisuuksista, joita järjestelmässä voidaan käyttää.



Kuvio 6. Node-RED-vaihtoehtoja

Kuvio 6. Node-RED-vaihtoehtoja esitetyillä ominaisuuksilla Node-RED todettiin pystyvän luomaan tarvittavat rakenteet, jotka mahdollistavat järjestelmän toiminnan sekä kuvan tunnistamisen.

4 Toteutus

Työtä lähdettiin toteuttamaan hajautetulla menetelmällä niin, että osat voidaan lopuksi yhdistää luomaan toimiva järjestelmä. Tämä tarkoitti käytännössä jokaisen työssä käytettävän osa-alueen toiminnan varmistamista omana yksittäisenä osanaan ennen muiden työssä käytettyjen osien yhdistämistä toteutukseen. Tämän seurauksena työ kehittyi modulaarisesti, ja jokaista työn osaa voitiin työstää erillään muista.

4.1 Laitteiston käyttöönotto

Työn toteutus aloitettiin tekemällä Raspberry Pin käyttöönottoimenpiteet, joita tarvitaan valmiiksi asennetun paketin käyttöönotossa. Vaihtamalla laitteen vakiosalasanana turvataan, että kukaan ulkopuolinen ei pääse verkkolaitteelle käsiksi. Raspbian-käyttöjärjestelmän salasanan vaihtaminen onnistuu komennolla

```
$ passwd
```

Salasanan vaihtamiseksi käyttäjän tulee tietää laitteen salasana, jota ollaan vaihtamassa uuteen. Kuvio 7 nähdään salasanan vaihtoprosessi.

```
pi@raspberrypi:~ $ passwd
Changing password for pi.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Kuvio 7. Salasanan vaihto

Seuraavana suoritetaan järjestelmän päivittäminen, että laitteelle saadaan uusimmat versiot käytettävistä ohjelmista. Järjestelmän päivitys suoritetaan komennoilla

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

Järjestelmän päivittämisen jälkeen voidaan alkaa asentamaan työssä tarvittavia ohjelmia kuten Node-RED, jota käytetään toteutuspuolen tiedonkäsittelyyn.

Node-RED-ohjelmiston sivuilta saadulla asennus komennolla:

```
$ bash <(curl -sL https://raw.githubusercontent.com/node-RED/raspbian-deb-package/master/resources/update-nodejs-and-nodeRED)
```

Tämä käynnistää Node-RED asennuksen, jonka jälkeen Node-REDiä voidaan käyttää osoitteessa localhost:1880 tai laitteen IP osoitteessa portissa 1880. (ks. Kuvio 8).

```
Running for user pi at /home/pi

This can take 20-30 minutes on a Pi 1 - please wait.

  Stop Node-RED                ✓
  Remove old version of Node-RED ✓
  Remove old version of node.js -
  Update node.js LTS           ✓   Node v6.10.2   Npm 3.10.10
  Clean npm cache              ✓
  Install Node-RED core        ✓   0.16.2
  Install extra nodes          ✓
  Install serialport node     ✓
  Npm rebuild existing nodes  ✓
  Add menu shortcut           ✓
  Update systemd script       ✓
  Update update script        ✓

Any errors will be logged to /var/log/nodered-install.log

All done.
You can now start Node-RED with the command node-red-start
or using the icon under Menu / Programming / Node-RED
Then point your browser to localhost:1880 or http://{your_pi_ip-address}:1880

Started Wed 5 Apr 09:05:17 UTC 2017 - Finished Wed 5 Apr 09:08:52 UTC 2017
```

Kuvio 8. Node-RED, asennus sekä käyttöosoitteet

Node-REDin toiminnan varmistamiseksi laitteen uudelleen käynnistymisen jälkeen voidaan suorittaa komento:

```
$sudo systemctl enable nodeRED.service
```

Komento muuntaa Node-REDin palveluksi, joka käynnistyy aina koneen käynnistytessä.

Asennuksen jälkeen Node-REDin pitäisi olla käyttövalmiina. Node-REDin käynnistys tapahtuu Kuvio 8 nähdyllä komennolla:

```
$ node-RED-start
```

Komennon jälkeen Node-RED on käynnistynyt terminaali-ikkunaan, johon komento ajettiin (ks. Kuvio 9).

```

pi@raspberrypi:~ $ node-red-start

Start Node-RED

Once Node-RED has started, point a browser at http://91.154.164.174:1880
On Pi Node-RED works better with the Firefox browser

Use node-red-stop to stop Node-RED
Use node-red-start to start Node-RED again
Use node-red-log to view the recent log output
Use sudo systemctl enable nodered.service to autostart Node-RED at every boot
Use sudo systemctl disable nodered.service to disable autostart on boot

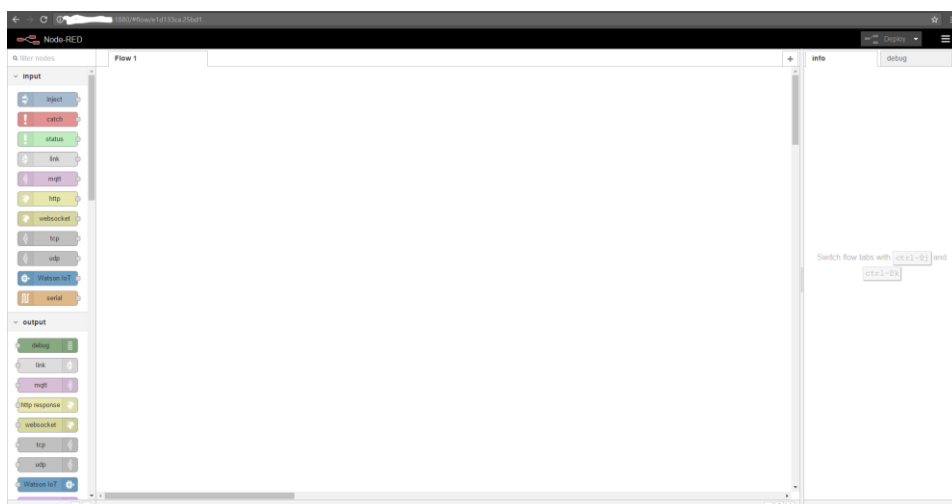
To find more nodes and example flows - go to http://flows.nodered.org

Started Node-RED graphical event wiring tool..
5 Apr 09:35:48 - [info]
Welcome to Node-RED
=====
5 Apr 09:35:48 - [info] Node-RED version: v0.16.2
5 Apr 09:35:48 - [info] Node.js version: v6.10.2
5 Apr 09:35:48 - [info] Linux 4.4.21-v7+ arm LE
5 Apr 09:35:49 - [info] Loading palette nodes
pi : TTY=unknown ; PWD=/home/pi ; USER=root ; COMMAND=/usr/bin/python -u /usr/lib/n
rdware/nrgpio.py info
pam_unix(sudo:session): session opened for user root by (uid=0)
pam_unix(sudo:session): session closed for user root
5 Apr 09:35:53 - [info] Settings file : /home/pi/.node-red/settings.js
5 Apr 09:35:53 - [info] User directory : /home/pi/.node-red
5 Apr 09:35:53 - [info] Flows file : /home/pi/.node-red/flows_raspberrypi.json
5 Apr 09:35:53 - [info] Creating new flow file
5 Apr 09:35:53 - [info] Server now running at http://127.0.0.1:1880/
5 Apr 09:35:53 - [info] Starting flows
5 Apr 09:35:53 - [info] Started flows

```

Kuvio 9. Node-RED, konsoli käynnistyy

Ohjelmiston käynnistymisen jälkeen Node-RED on käyttövalmis ja vastaa laitteen verkkoportissa 1880. Node-RED avautuu verkkoselaimeen laitteen IP-osoitteen ja portin takaa kuten. (ks. Kuvio 10.)



Kuvio 10. Node-RED, verkkosivu

Asennuksessa noudatettiin tuotteen sivuilla jaetun asennus dokumentin manuaali asennusta (Running on Raspberry Pi 2017).

Node-REDin toimintaan saannin jälkeen asennettiin Raspberry Pin päälle Apache2-ohjelmisto, joka mahdollistaa verkkosivujen julkaisemisen laitteella. Node-RED antaa mahdollisuuden käyttää verkkosivuja ohjelmiston hallintaan, mutta ei mahdollista omien sivujen julkaisua. Apache2 paikkaa edellä mainitun raon ja mahdollistaa itse tehtyjen verkkosivujen julkaisun tietoverkossa. Apache2:n asennus saatiin suoritettua komennolla

```
# apt-get install apache2
```

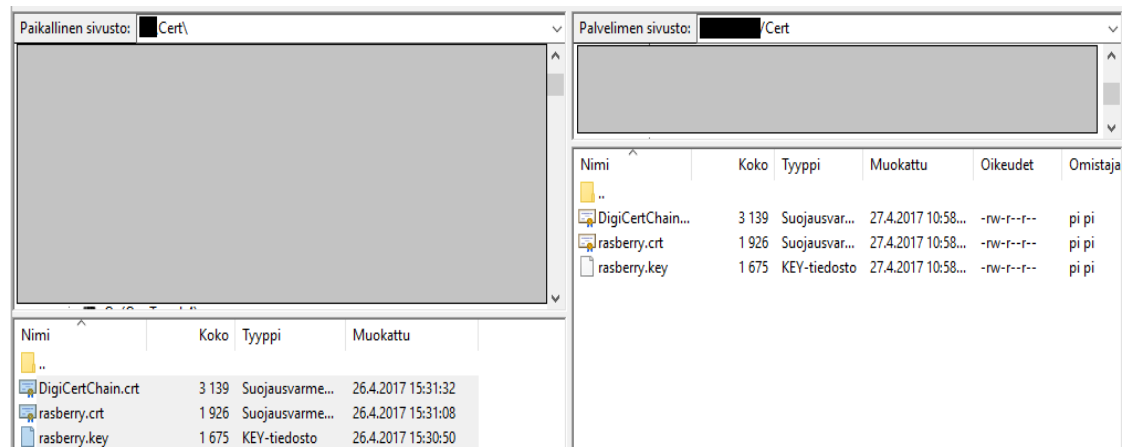
Asennuksen jälkeen Apache2 oli asennettu ja laitteen IP osoitteeseen yhdistettäessä sieltä saatiin vastaamaan ohjelmiston oletussivu (ks. Kuvio 11).



Kuvio 11. Apache2, oletussivu

Apache2-asennuksen jälkeen haluttiin lisätä mahdollisuus käyttää sivuja turvallisemmin käyttäen SSL-salattua liikennöintiä palvelun kanssa keskusteltaessa. Salaus mahdollistaa tiedon välittämisen, kuten salasanojen, palvelun kanssa keskusteltaessa ilman, että kyseistä tietoa voidaan napata selvänä tekstinä verkkoliikenteestä. Salausta pidettiin tärkeänä mahdollisen Office365 kalenteri liitettävyyden kannalta, sekä muun salassa pidettävän tiedon turvaamiseksi.

Salauksen mahdollistamiseksi pyydettiin Jyväskylän ammattikorkeakoulun tietohallinnolta sertifikaatti laitteelle, jonka olisi tarkoitus taata laitteen turvallisuus. Kuvio 12. Sertifikaatti nähdään laitteelle siirretyt tietohallinnolta saadut sertifikaatti tiedostot.



Kuvio 12. Sertifikaatti

Tiedostojen siirron jälkeen Apache2:lle annettiin komento, jolla salaus saadaan aktivoitua sekä luodaan salaukseen liittyvät tiedostot.

```
# a2enmod ssl
```

Salaukseen liittyvien tiedostojen luonnin jälkeen saatiin sertifikaattiin liittyvät tiedostot yhdistettyä Apacheen. Kuvio 13. Sertifikaatin konfigurointi Apacheen nähdään Apachen asetukset, joissa sertifikaatti on liitetty ohjelmistoon.

```

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf

# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on

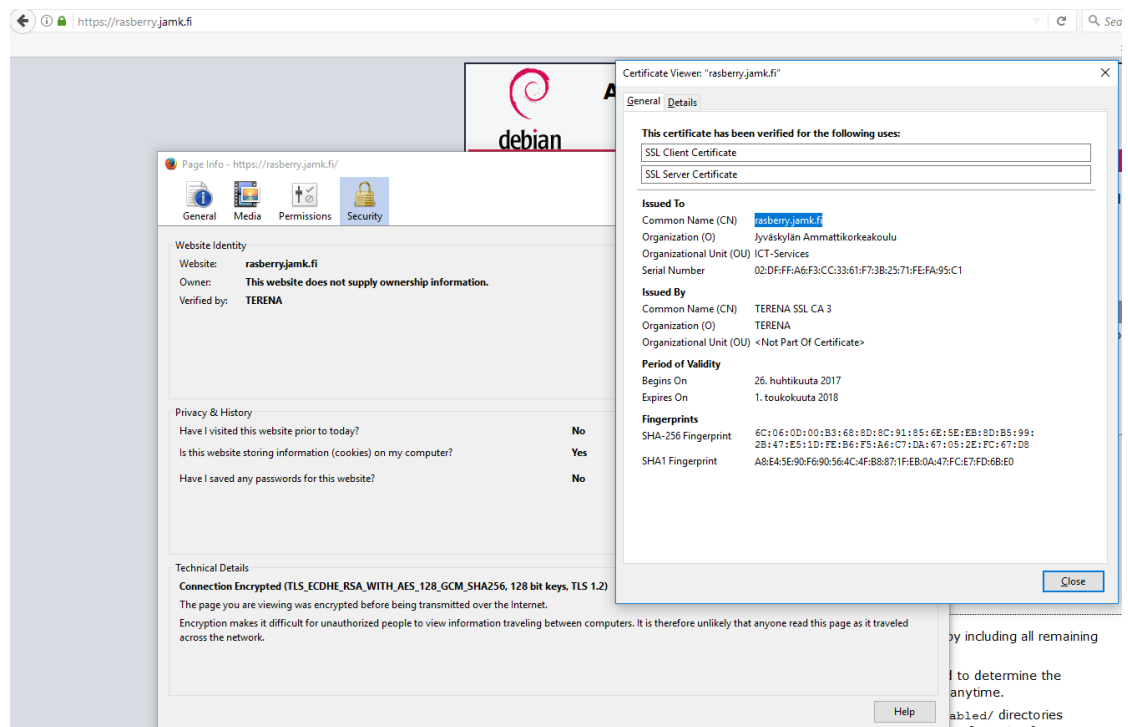
# A self-signed (snakeoil) certificate can be created by installing
# the ssl-cert package. See
# /usr/share/doc/apache2/README.Debian.gz for more info.
# If both key and certificate are stored in the same file, only the
# SSLCertificateFile directive is needed.
SSLCertificateFile      /Cert/raspberry.crt
SSLCertificateKeyFile   /Cert/raspberry.key

# Server Certificate Chain:
# Point SSLCertificateChainFile at a file containing the
# concatenation of PEM encoded CA certificates which form the
# certificate chain for the server certificate. Alternatively
# the referenced file can be the same as SSLCertificateFile
# when the CA certificates are directly appended to the server
# certificate for convinience.
SSLCertificateChainFile /Cert/DigiCertChain.crt

```

Kuvio 13. Sertifikaatin konfigurointi Apacheen

Sertifikaattiin liittyvien asetusten viimeistelyn ja Apache2n uudelleen käynnistyksen jälkeen SSL varmenteen toiminta voitiin varmistaa (ks: Kuvio 14.).



Kuvio 14. Sivuston SSL-varmenne

Apache2:n ja SSL varmenteen asennukseen löydettiin ohjeet Digitalocena.com sivustolta (How To Create a SSL Certificate on Apache for Ubuntu 14.04). Ääkkös ominaisuuden lisääminen Apacheen hoidettiin Ubuntu-fi.org foorumeilta Apachen asennus ohjeiden joukosta (Apache + PHP + MySQL asentaminen 2017). Kuvio 15. Ääkkösten konfiguraatio esittää apache2.conf tiedostoon tehtyä muokkausta, joka mahdollistaa ääkkösten toiminnan Apachessa.

```

GNU nano 2.2.6                               File: /etc/apache2/apache2.conf
</FilesMatch>

#
# The following directives define some format nicknames for use with
# a CustomLog directive.
#
# These deviate from the Common Log Format definitions in that they use %O
# (the actual bytes sent including headers) instead of %b (the size of the
# requested file), because the latter makes it impossible to detect partial
# requests.
#
# Note that the use of %{X-Forwarded-For}i instead of %h is not recommended.
# Use mod_remoteip instead.
#
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" vhost_combined
LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %O" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

# Include of directories ignores editors' and dpkg's backup files,
# see README.Debian for details.

# Include generic snippets of statements
IncludeOptional conf-enabled/*.conf

# Include the virtual host configurations:
IncludeOptional sites-enabled/*.conf

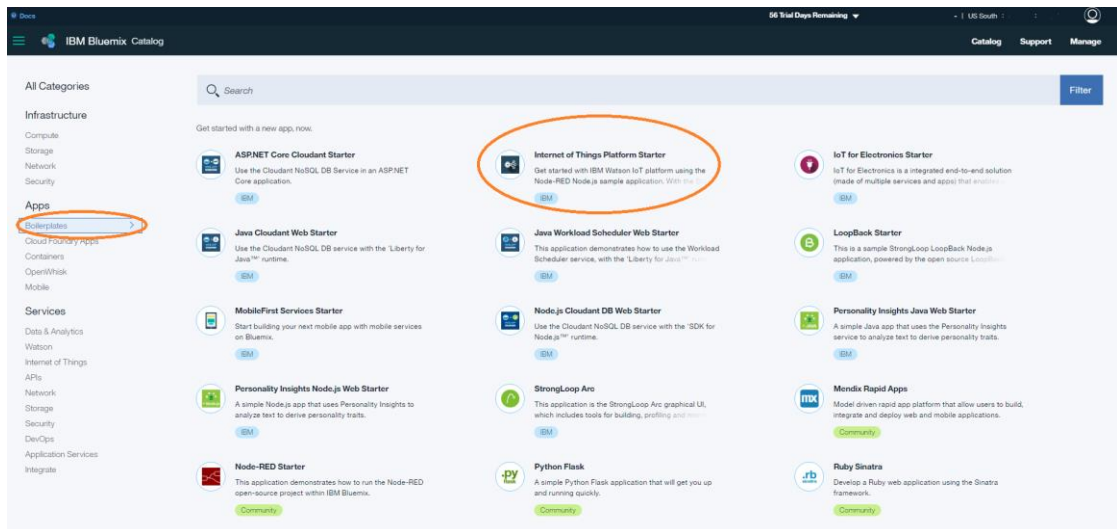
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
AddDefaultCharset      ISO-8859-1

```

Kuvio 15. Ääkkösten configuraatio

4.2 Bluemixin käyttöönotto

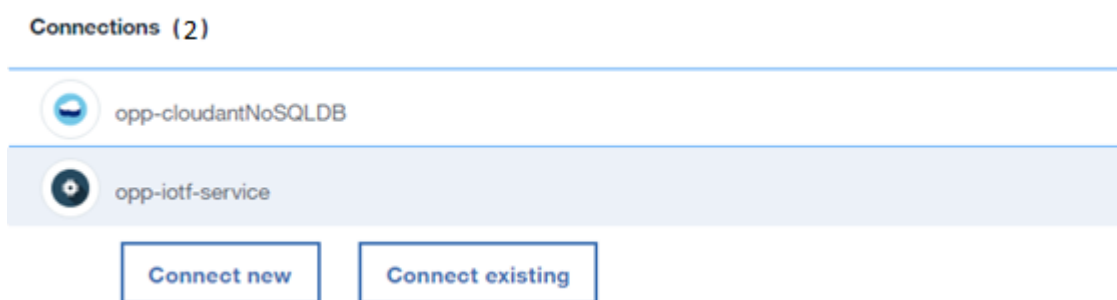
IBM Bluemix käyttämiseksi käyttäjän on luotava itselleen tunnukset. Bluemix kirjautumissivulta ” <https://console.ng.bluemix.net/>” ohjataan luomaan tunnukset, joiden avulla käyttäjä voi aloittaa Bluemixin 30 päivän testikäytön. Onnistuneen rekisteröinnin jälkeen käyttäjän tulee luoda itselleen applikaatio. Kuvio 16 nähdään katalogi, josta käyttäjä voi valita applikaatiotyyppin, joista valita.



Kuvio 16. Applikaatiokatalogi

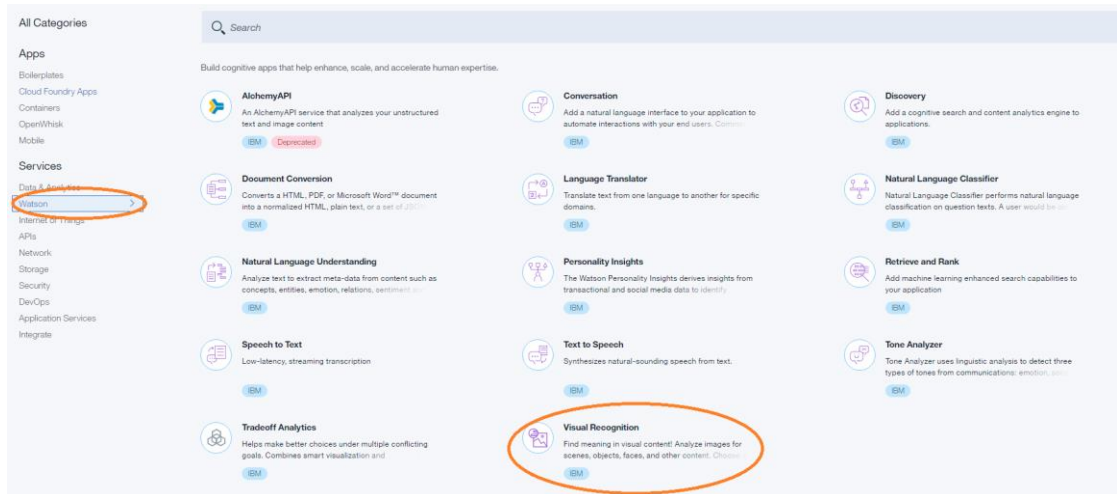
Kuvio 16. Applikaatiokatalogi oranssilla rengastettu "Boilerplates" ohjaa käyttäjän valikkoon, josta voidaan valita sopiva aloituspaketti. Työtä varten toimivin paketti on oranssin rengastuksen sisällä oleva "Internet of Things Platform Starter", joka valittiin työn tekemiseen.

Seuraavaksi applikaatiolle annettiin nimet ja valittiin muut valinnat, jonka jälkeen IoT applikaatio voidaan luoda. Luonnin jälkeen applikaatioon voidaan liittää kuvantunnistus. Kuvio 17 esittämästä napista käyttäjä voi luoda yhteyden Bluemix palveluihin.



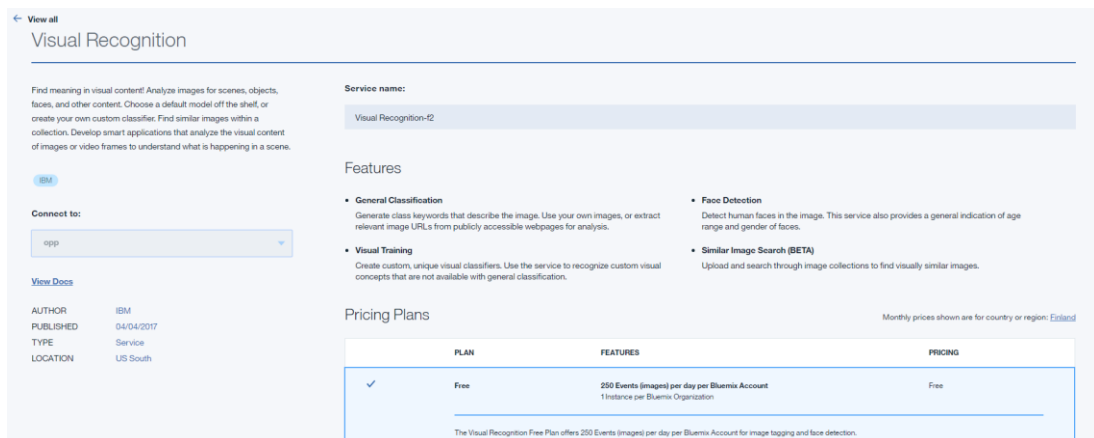
Kuvio 17. Palveluun yhdistäminen

Yhteyden luominen "connect new" napin takaa, josta avautuu ryhmä valintoja. Kuvio 18 kuvaa valintoja, jotka voidaan yhdistää aikaisemmin luotuun applikaatioon.



Kuvio 18. Kuvan tunnistuksen yhdistäminen palveluun

Kuvion rengastukset kuvaavat reittiä minkä kautta työssä käynnistettävä kuvantunnistus löytyy. Kuvantunnistuksen valinnan jälkeen aukeaa valintaikkuna, josta voidaan määritellä asetuksia jotka vaikuttavat tunnistukseen. Kuvio 19 näyttää ilmaisen kuvantunnistuksen asetukset sekä rajoitteet.



Kuvio 19. Kuvantunnistuksen asetukset

Ilmainen kuvantunnistus mahdollistaa kuvion mukaan 250 tunnistuskertaa päivässä. 250 tunnistusta päivässä tarkoittaa noin 4 tunnin ja 10 minuutin edestä tunnistusai-
 kaa, jos tunnistus ajetaan kerran minuutissa. Tämän rajoituksen huomioon ottaen
 tunnistusta tullaan ajamaan rajoitetusti noin puolitoista tuntia kerrallaan. Tarkem-
 masta ajoituksesta päätettiin päättää myöhemmin. (Bluemix kuvat 2017.)

4.3 Kuvan tunnistus

Kuvan tunnistuksen kytkeminen Bluemix palveluun mahdollistaa oman tunnisteiden luonnin sekä kuvan tunnistuksen käyttämisen Bluemix palvelussa. Watson mahdollistaa tunnistuksen käyttämisen myös Bluemix ympäristön ulkopuolelta oikeilla tunnistautumistiedoilla. Palvelun käyttäminen Bluemix palvelimen ulkopuolelta mahdollistaa sen, että kuvantunnistus ei ole yksinomaan lukittu Bluemix alle vaan sitä voidaan käyttää myös muissa ympäristöissä.

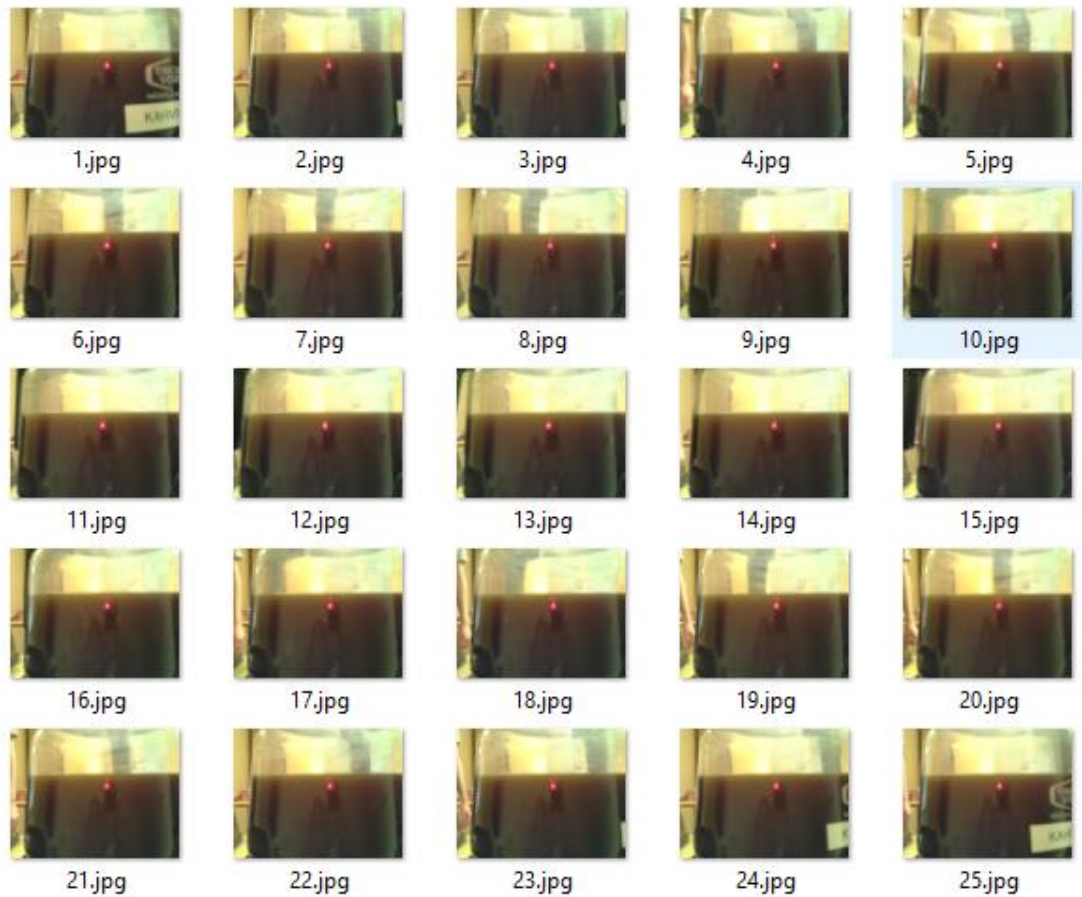
4.4 Kuvamateriaalin kerääminen

Kuvantunnistuksen käyttämiseksi tietyssä tilanteessa täytyy sen osata verrata tunnistettavaa kuvaa etukäteen opetettuun materiaaliin. Kuvamateriaalin tulisi olla tunnistettavissa samaksi aiheeksi ja samalla riittävän erilaista, että tunnistus osaa ymmärtää tilanteen mahdollisesti muuttuvan.

Tämän työn kohdalla materiaalina on kahvinkeitinpannussa oleva kahvin määrä, jonka pohjalta voidaan opettaa kuvantunnistus ymmärtämään kuinka paljon pannussa on jäljellä. Tämän kuvamateriaalin keräämiseksi pannusta tulee ottaa kuvia.

Toimivan tunnisteiden luomiseksi jokaisesta tunnistettavasta tilanteesta tulee ottaa 10 kuvaa, jotka tullaan syöttämään Watsonille. 10 kuvaa on minimimäärä, jonka Watson tarvitsee tunnisteiden luomiseksi niin, että toimiva tunniste voidaan luoda materiaalista. Kuvamateriaalin keräämisen yhteydessä päätettiin kumminkin ottaa ylimääräistä materiaalia varmuuden vuoksi ja tilannekohtaisia kuvia otettiin 25.

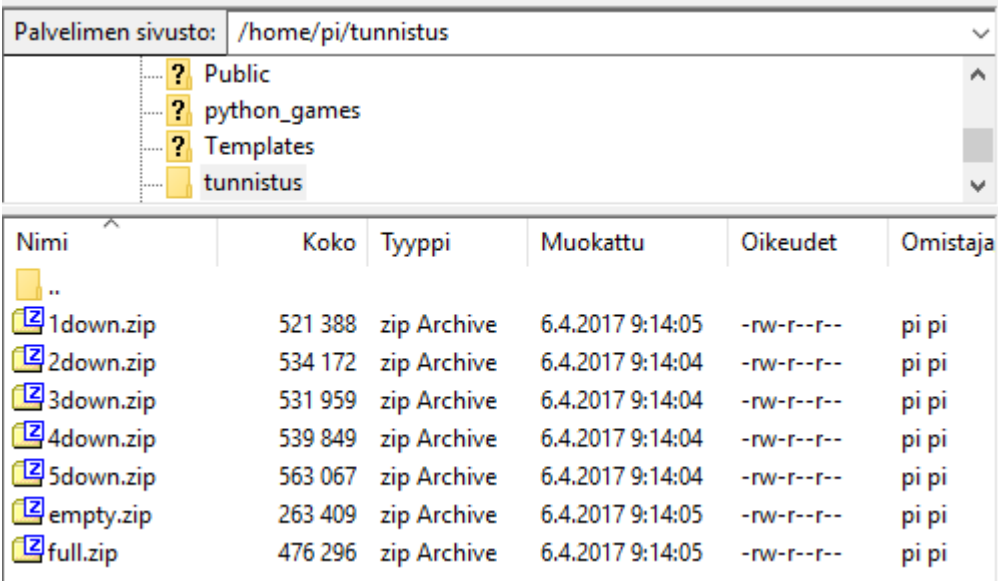
Kuvio 20 osoittaa kuinka samalta otetut kuvat näyttävät, mutta kuten kuvista voidaan huomata, niin pannussa olevat merkinnät ja kahvan sijainti muuttuvat. Kuvasarjan pohjalta uskotaan olevan mahdollista luoda toimiva tunnistustilanne.



Kuvio 20. Tunnistussarja

4.5 Tunnisteen luonti

Kuvamateriaalin keräämisen jälkeen voitiin siirtyä luoman tunniste, joka kattaa kaikki tilanteet. Työn suunnittelun ja alustavan tutkinnan pohjalta huomattiin, että yksinkertaisin tapa luoda toimiva tunniste on käyttää laitteen terminaalikkunassa toimivaa *Curl* menetelmää. Käytännössä tämä tarkoittaa, että kaikki tunnisteen luontiin tarkoitettu materiaali tulee siirtää laitteelle, jonka terminaalista tunnistetta ollaan luomassa. Tässä tapauksessa päätettiin käyttää työssä käytettävää Raspberry Piä. Kuvio 21 esittää kuinka tiedostot siirrettiin laitteelle omaan kansioon pakatuissa zip tiedostoissa.



Nimi	Koko	Tyyppi	Muokattu	Oikeudet	Omistaja
..					
1down.zip	521 388	zip Archive	6.4.2017 9:14:05	-rw-r--r--	pi pi
2down.zip	534 172	zip Archive	6.4.2017 9:14:04	-rw-r--r--	pi pi
3down.zip	531 959	zip Archive	6.4.2017 9:14:04	-rw-r--r--	pi pi
4down.zip	539 849	zip Archive	6.4.2017 9:14:04	-rw-r--r--	pi pi
5down.zip	563 067	zip Archive	6.4.2017 9:14:04	-rw-r--r--	pi pi
empty.zip	263 409	zip Archive	6.4.2017 9:14:05	-rw-r--r--	pi pi
full.zip	476 296	zip Archive	6.4.2017 9:14:05	-rw-r--r--	pi pi

Kuvio 21. Tunnistustapaukset

Materiaalin siirron jälkeen voidaan luoda komento, jolla paketit voidaan opettaa tunnistukselle sekä liittää niille nimet, joita voidaan käyttää tunnisteiden yhteydessä. Tunnisteiden luonnissa tarvitaan niin sanottu positiivinen ja negatiivinen tai useita positiivisia tapahtumia. IBM:n tarjoama ohje asiaan sivustollaan:

need at least 2 _positive_examples fields, (or 1 _positive_examples and 1 negative_examples field) to train a classifier. (Create a classifier 2017)

Tämä tarkoittaa, että jokaisen tapauksen perässä pitää olla merkintä tapauksen tyylistä. Näiden tietojen mukaan tunnisteiden opettamiseksi Watsonille terminaaliin syötettäväksi komennoksi muodostuu:

```

$ curl -X POST -F "Full_positive_examples=@full.zip" -F "Al-
most_Full_positive_examples=@1down.zip" -F
"Third_left_positive_examples=@2down.zip" -F
"Less_than_half_positive_examples=@3down.zip" -F "Al-
most_empty_positive_examples=@4down.zip" -F
"Last_change_positive_examples=@5down.zip" -F
"Empty_positive_examples=@empty.zip" -F "name=Pannussa" https://gateway-
a.watsonplatform.net/visual-recognition/api/v3/classifiers?api_key={<API poistet-
tu>}&version=2016-05-20"

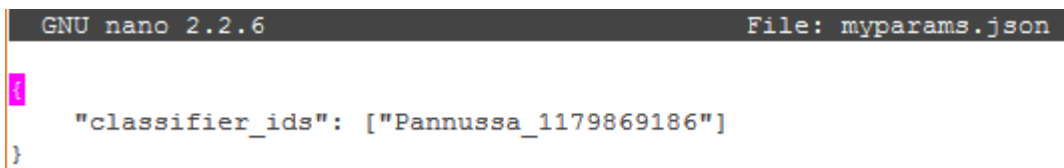
```

Pienen odotuksen jälkeen terminaali palautti viestin, joka kertoi tunnisteeseen liittyvät tiedot. Kuvio 22 näyttää viestissä esitetyt tiedot.

```
{
  "classifier_id": "Pannussa_1179869186",
  "name": "Pannussa",
  "owner": "da69bf94-9c2b-4bdf-bdd2-e98556188c96",
  "status": "training",
  "created": "2017-04-06T06:38:17.564Z",
  "classes": [
    {"class": "Full"},
    {"class": "Empty"},
    {"class": "Almost_Full"},
    {"class": "Almost_empty"},
    {"class": "Third_left"},
    {"class": "Less_than_half"},
    {"class": "Last_change"}
  ]
}
```

Kuvio 22. Tunnisteen tiedot

Tunnisteen luonnin jälkeen Watson käsittelee tunnistetta, jonka jälkeen sitä voidaan alkaa käyttämään. Tunnisteen käyttäminen terminaalissa vaatii parametri tiedoston, joka sisältää tunnistuksen käyttämät määrittymät. Kuvio 23 esittää laitteelle määritettyä config tiedostoa.



```
GNU nano 2.2.6 File: myparams.json
{
  "classifier_ids": ["Pannussa_1179869186"]
}
```

Kuvio 23. Käyttöön määritellyt tunnisteet

Määrittysten lisäämisen jälkeen tunnistusta voidaan käyttää nyt terminaalien kautta komennolla.

```
$ curl -X POST -F "images_file=@/home/pi/tunnistus/kuppi2/6.jpg" -F "parameters=@/home/pi/tunnistus/myparams.json" "https://gateway-
a.watsonplatform.net/visual-
recognition/api/v3/classify?api_key={2c0443359a6bdadd1f6cd787bc03855b3ad0d57e}&ver-
sion=2016-05-20"
```

Komennon suorittamisen jälkeen laite saa palautuksena viestin siitä, miten Watson on tunnistanut kuvan. Kuvio 24 nähdään kuvan tunnistukselta tulleet tiedot tunnistuksessa olleesta kuvasta.

```
"custom_classes": 7,  
"images": [  
  {  
    "classifiers": [  
      {  
        "classes": [  
          {  
            "class": "Third_left",  
            "score": 0.998162  
          }  
        ],  
        "classifier_id": "Pannussa_1179869186",  
        "name": "Pannussa"  
      }  
    ],  
    "image": "6.jpg"  
  }  
],  
"images_processed": 1
```

Kuvio 24. Tunnistuksen palautus

Terminaalin kautta suoritettu tarkastus tehtiin materiaalin keruuvaiheessa otetusta kuvasta. Kuvio 25 on sama kuva, joka ajettiin tunnistuksen läpi.

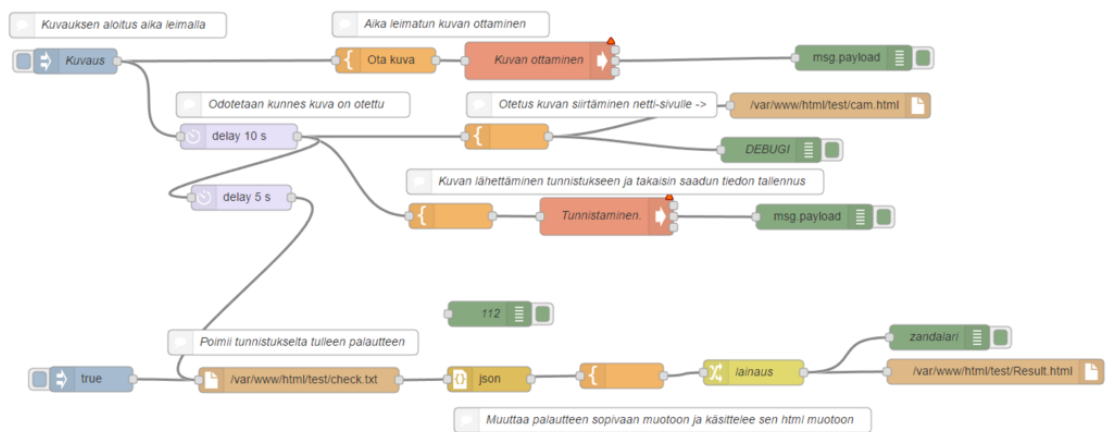


Kuvio 25. Kolmannes pannusta juotu

Tunnisteen luonnissa käytettiin apuna IBM Watsonin ohjeita, joista saatiin koodi esimerkkejä (Visual Recognition: API Reference 2017).

4.6 Tunnistuksen toteutus

Tunnistuksessa jatkettiin Raspberry Pin kameran käyttämistä materiaalin keräämisessä. Kameran ohjausta varten pystytettiin Raspberry Piin päällä pyörivälle Node-RED ympäristölle hallintamenetelmä, jonka avulla tunnistuksen tarvitsema kuvamateriaali voitiin ottaa. Node-RED hallintapaneelin Kuvio 26 voidaan nähdä, kuinka tämä on toteutettu.

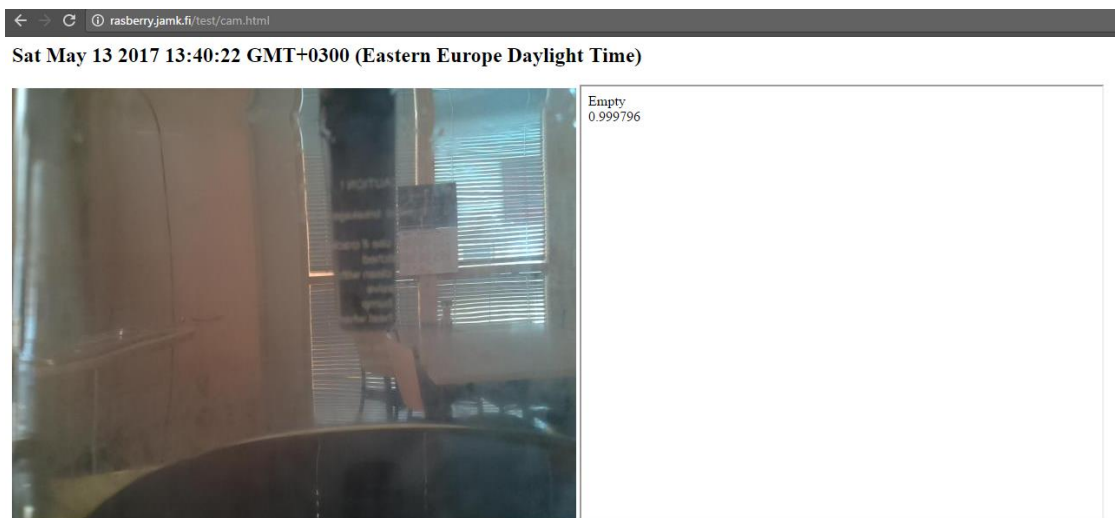


Kuvio 26. Kuvanottaminen ja tunnistaminen

Kuvio 26. Kuvanottaminen ja tunnistaminen nähdään, kuinka materiaalin keräys voidaan hoitaa nappia painamalla ja sen jälkeen linkittää eteenpäin. Node-RED mahdollistaa aikaisemmasta "laatikosta" saadun tiedon siirtämisen seuraavan käsittelypisteeseen. Kuvio 26. Kuvanottaminen ja tunnistaminen näyttää myös, kuinka alkuperäinen aikaleima käsitellään jokaisella prosessin asteella.

Käsittelyssä käytetään Node-REDin "template"-ominaisuutta, jonka avulla aikaisemmasta pisteestä saapuva data saadaan muokattua mukaan halutunlaiseen rakenteeseen. Aikaisemmasta pisteestä tuleva tieto voidaan muokata mukaan komentoon, joka suoritetaan Raspberry Pillä kuvan punaisissa laatikoissa.

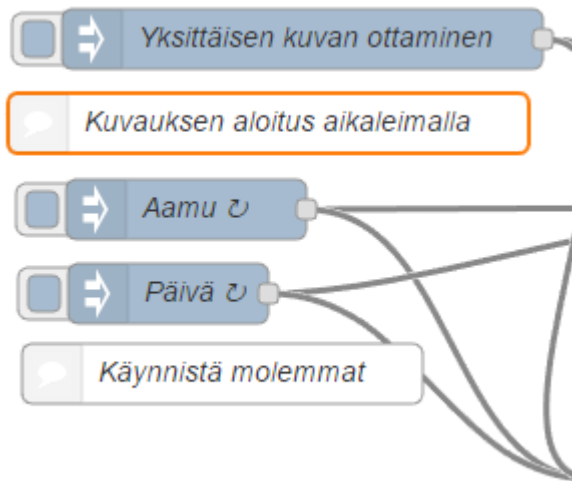
Kuvio 26. Kuvanottaminen ja tunnistaminen nähdyn rakenteen pohjalta voidaan julkaista verkkosivu, jossa nähdään otettu kuva materiaali sekä IBM Watsonin palauttama tunnistus kuvasta. Kuvio 27 esittelee mitä internetissä julkaistaan, kun prosessi aloitetaan painamalla nappia.



Kuvio 27. Tunnistuksen lopputulos

Node-RED prosessin ajo tuottaa laitteen nettisivulle kuvan mukaisen ilmoituksen keittimen tilasta. Sivun tarjoaa käyttäjälle kuvan keittimen tilasta sekä Watsonin kuvan tunnistuksen tarjoaman näkemyksen pannun tilasta. Seuraavana kohteena oli saada tunnistukselta tuleva teksti muokattua paremmin ymmärrettävään muotoon. Järjestelmä asetettiin keräämään tiedot tunnistukselle yleisimpiin aikoihin, jolloin henkilökunta on yleensä käyttämässä keitintä lisäämällä muokattuja kuvaus nappeja Node-RED toteutukseen.

Kuvio 28 voidaan nähdä Node-RED lisätyt ajoituksen mahdollista solmut, jotka käynnistävät tunnistuksen ennalta määritettyyn ajankohtaan.



Kuvio 28. Tunnistuksen aloitus

Node-RED mahdollistaa tehtävän aloituksen suoraan ilman tarvetta erillisen funktion kirjoittamiselle (ks Kuvio 29.).

The image shows two side-by-side screenshots of the Node-RED 'Repeat' node configuration interface. Both screenshots show the 'Payload' set to 'timestamp' and the 'Repeat' mode set to 'interval between times' with a frequency of 1 minute. The left screenshot is for the 'Aamu' (Morning) configuration, with the time range set to 'between 10:00 and 12:00'. The right screenshot is for the 'Päivä' (Day) configuration, with the time range set to 'between 13:00 and 15:00'. Both configurations have the days of the week checked from Monday to Friday, with Saturday and Sunday unchecked. A note at the bottom of each configuration states: 'Note: "interval between times" and "at a specific time" will use cron. See info box for details.'

Kuvio 29. Ajoitus mahdollisuudet.

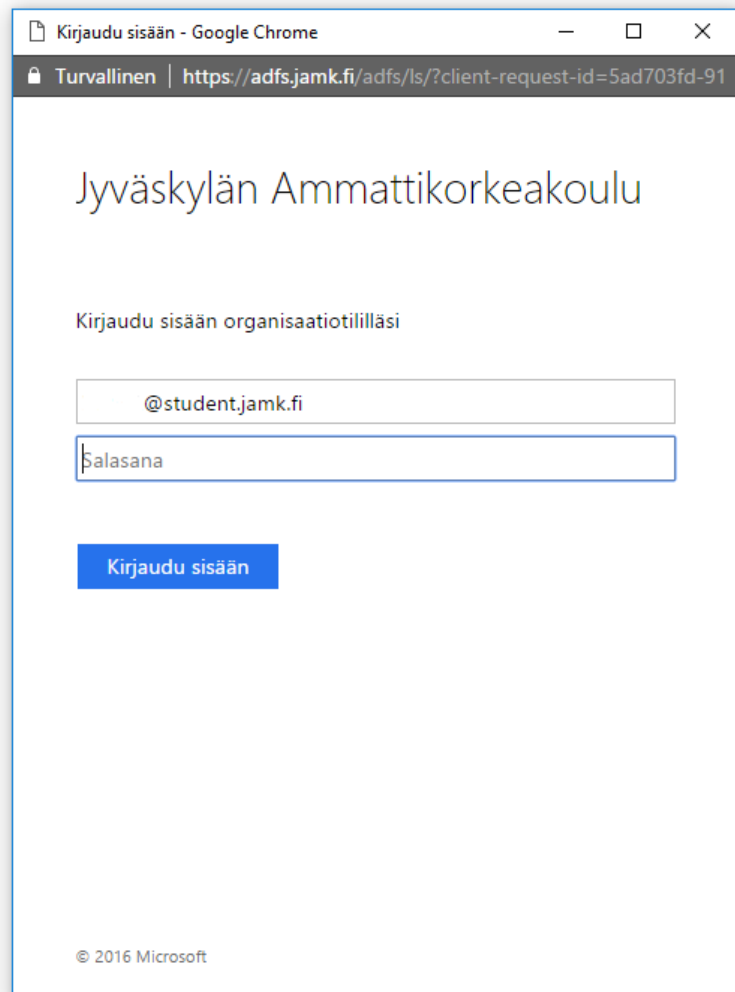
Toteutus on tarkoituksella rajoitettu 4 tunnin tunnistus aikaan päivässä. Watsonin ilmainen kuvan tunnistus mahdollistaa vain 250 kuvan tarkistamisen päivässä, jolloin 4 tunnin tarkistus ajalla tunnistus ajetaan 240 kertaa.

4.7 Kalenterin tiedon hakeminen

Henkilökunnan kalenteritietoa pyrittiin hakemaan usealla tavalla. Opettajien kalenteritietoa voidaan hakea Microsoft Office365 kalenterista, joka on lukittu organisaation sisäiseen käyttöön ja salasana suojattu. Toisena vaihtoehtona kalenteritiedolle olisi Jyväskylän ammattikorkeakoulun tilanvarausjärjestelmä, josta olisi nähtävissä minä aikoina opettajat olisivat koululla, mutta kyseisessä kalenterissa ei näy muita henkilökunnan jäseniä.

Kalenteritiedon integrointi projektiin järkevällä tavalla osoittautui hankalaksi. Tilanvarauksen kautta hankittu kalenteritieto osoittautui suhteellisen helpoksi niin, että käyttäjä huomaa helposti onko haetulla henkilöllä opetusta tietyinä ajankohtana. Tämä tieto ei kuitenkaan takaa ajankohtaisuutta. Office365 kautta haetun tiedon vahvuutena olisi sen ajankohtaisuus. Microsoftin API mahdollista Office365 tunnukset tunnistautumisen kolmannen osapuolen järjestelmään. Onnistuneen kirjautumisen saavuttaminen mahdollistaisi tiedon hakemisen Office365 kalenterista. Hakua ei kuitenkaan saatu toimimaan, sillä ohjeistusta tiedon hakuun Microsoftin API:n kautta ei ollut saatavilla. Kuvio 30. Jamk tunnuksilla tunnistautuminen esittää miltä Office365 Kirjautumis-ikkuna näyttää.

Microsoft Tunnistautuminen



Kuvio 30. Jamk tunnuksilla tunnistautuminen

Pelkällä kirjautumismahdollisuudella Microsoftin rajapintaan ei saada aikaan haluttua toimintoa, jolla kalenteri tieto saataisiin haettua Microsoftin tietokannasta.

Tilanvarauksen kautta saatava tieto on valmiiksi käsitelty tiettyyn muotoon, jonka uudelleen käsittely tehtävänantoa vastaavaan muotoon olisi käytännössä mahdotonta. Kuvio 31 esittää esimerkin siitä missä muodossa tilanvaraus tieto saadaan haettua.

Ohjelma (ryhmä) IT1451 / Viikko 13					
	Maanantai 27.03.2017	Tiistai 28.03.2017	Keskiviikko 29.03.2017	Torstai 30.03.2017	Perjantai 31.03.2017
8:00	08:00 - 11:15 IT35310.7KOV4 QoS-verkkojen suunnittelu ja P2_D331 Jarmo	08:00 - 09:30 LIIKKAUVA (Ei METS) P2_D430 Vesa Pentti	08:00 - 11:15 IT50120.7KOV3 Tietoturvan hallinta P2_D435 Rantonen Mika	08:00 - 14:00 IT50120.7KOV3 Tietoturvan hallinta P2_D435 Rantonen Mika	08:00 - 11:15 IT35310.7KOV4 QoS-verkkojen suunnittelu ja P2_D331 Jarmo
9:00	08:00 - 11:15 IT35310.7KOV4 QoS-verkkojen suunnittelu ja P2_D331 Jarmo	09:45 - 12:15 IT35310.7KOV4 QoS-verkkojen suunnittelu ja P2_D331 Jarmo			08:00 - 11:15 IT40400.7KOV3 Tietoturvan toteutus P2_D435 Saan Henriik
10:00	08:00 - 11:15 IT40400.7KOV3 Tietoturvan toteutus P2_D435 Saan Henriik				
11:00					
12:00	12:15 - 15:45 IT35310.7KOV4 QoS-verkkojen suunnittelu ja P2_D331 Jarmo		12:15 - 15:00 IT50120.7KOV3 Tietoturvan hallinta P2_D435 Rantonen Mika	12:15 - 15:45 IT35310.7KOV4 QoS-verkkojen suunnittelu ja P2_D331 Jarmo	12:15 - 15:45 IT40400.7KOV3 Tietoturvan toteutus P2_D435 Saan Henriik
13:00	12:15 - 15:45 IT35310.7KOV4 QoS-verkkojen suunnittelu ja P2_D331 Jarmo	13:15 - 16:45 IT35100.7KOV4 Reaktiivikuvaukset P2_D318 Antero			12:15 - 15:45 IT40400.7KOV3 Tietoturvan toteutus P2_D435 Saan Henriik
14:00	12:15 - 15:45 IT40400.7KOV3 Tietoturvan toteutus P2_D435 Saan Henriik			15:00 - 18:00 IT55500.7KOV3 Päätyöväkät P2_D327 Jarmo	
15:00					
16:00					
17:00					
18:00					
19:00					

```

Elements Console Sources Network Timeline Profiles Application Security Audits


| Ohjelma (ryhmä) IT1451 / Viikko 13 |                                                                                       |                                                                                       |                                                                                       |                                                                                       |                                                                                       |
|------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
|                                    | Maanantai<br>27.03.2017                                                               | Tiistai<br>28.03.2017                                                                 | Keskiviikko<br>29.03.2017                                                             | Torstai<br>30.03.2017                                                                 | Perjantai<br>31.03.2017                                                               |
| 8:00                               | 08:00 - 11:15<br>IT35310.7KOV4<br>QoS-verkkojen<br>suunnittelu ja<br>P2_D331<br>Jarmo | 08:00 - 09:30<br>LIIKKAUVA (Ei<br>METS)<br>P2_D430<br>Vesa Pentti                     | 08:00 - 11:15<br>IT50120.7KOV3<br>Tietoturvan<br>hallinta<br>P2_D435<br>Rantonen Mika | 08:00 - 14:00<br>IT50120.7KOV3<br>Tietoturvan<br>hallinta<br>P2_D435<br>Rantonen Mika | 08:00 - 11:15<br>IT35310.7KOV4<br>QoS-verkkojen<br>suunnittelu ja<br>P2_D331<br>Jarmo |
| 9:00                               | 08:00 - 11:15<br>IT35310.7KOV4<br>QoS-verkkojen<br>suunnittelu ja<br>P2_D331<br>Jarmo | 09:45 - 12:15<br>IT35310.7KOV4<br>QoS-verkkojen<br>suunnittelu ja<br>P2_D331<br>Jarmo |                                                                                       |                                                                                       | 08:00 - 11:15<br>IT40400.7KOV3<br>Tietoturvan<br>toteutus<br>P2_D435<br>Saan Henriik  |
| 10:00                              | 08:00 - 11:15<br>IT40400.7KOV3<br>Tietoturvan<br>toteutus<br>P2_D435<br>Saan Henriik  |                                                                                       |                                                                                       |                                                                                       |                                                                                       |
| 11:00                              |                                                                                       |                                                                                       |                                                                                       |                                                                                       |                                                                                       |
| 12:00                              | 12:15 - 15:45<br>IT35310.7KOV4<br>QoS-verkkojen<br>suunnittelu ja<br>P2_D331<br>Jarmo |                                                                                       | 12:15 - 15:00<br>IT50120.7KOV3<br>Tietoturvan<br>hallinta<br>P2_D435<br>Rantonen Mika | 12:15 - 15:45<br>IT35310.7KOV4<br>QoS-verkkojen<br>suunnittelu ja<br>P2_D331<br>Jarmo | 12:15 - 15:45<br>IT40400.7KOV3<br>Tietoturvan<br>toteutus<br>P2_D435<br>Saan Henriik  |
| 13:00                              | 12:15 - 15:45<br>IT35310.7KOV4<br>QoS-verkkojen<br>suunnittelu ja<br>P2_D331<br>Jarmo | 13:15 - 16:45<br>IT35100.7KOV4<br>Reaktiivikuvaukset<br>P2_D318<br>Antero             |                                                                                       |                                                                                       | 12:15 - 15:45<br>IT40400.7KOV3<br>Tietoturvan<br>toteutus<br>P2_D435<br>Saan Henriik  |
| 14:00                              | 12:15 - 15:45<br>IT40400.7KOV3<br>Tietoturvan<br>toteutus<br>P2_D435<br>Saan Henriik  |                                                                                       |                                                                                       | 15:00 - 18:00<br>IT55500.7KOV3<br>Päätyöväkät<br>P2_D327<br>Jarmo                     |                                                                                       |
| 15:00                              |                                                                                       |                                                                                       |                                                                                       |                                                                                       |                                                                                       |
| 16:00                              |                                                                                       |                                                                                       |                                                                                       |                                                                                       |                                                                                       |
| 17:00                              |                                                                                       |                                                                                       |                                                                                       |                                                                                       |                                                                                       |
| 18:00                              |                                                                                       |                                                                                       |                                                                                       |                                                                                       |                                                                                       |
| 19:00                              |                                                                                       |                                                                                       |                                                                                       |                                                                                       |                                                                                       |



```

<table border="1" style="border-collapse: collapse; width: 100%; text-align: center;">
| Ohjelma (ryhmä) IT1451 / Viikko 13 | | | | | |
|---|---|---|---|---|---|
| | Maanantai 27.03.2017 | Tiistai 28.03.2017 | Keskiviikko 29.03.2017 | Torstai 30.03.2017 | Perjantai 31.03.2017 |
| 8:00 | 08:00 - 11:15 IT35310.7KOV4 QoS-verkkojen suunnittelu ja P2_D331 Jarmo | 08:00 - 09:30 LIIKKAUVA (Ei METS) P2_D430 Vesa Pentti | 08:00 - 11:15 IT50120.7KOV3 Tietoturvan hallinta P2_D435 Rantonen Mika | 08:00 - 14:00 IT50120.7KOV3 Tietoturvan hallinta P2_D435 Rantonen Mika | 08:00 - 11:15 IT35310.7KOV4 QoS-verkkojen suunnittelu ja P2_D331 Jarmo |
| 9:00 | 08:00 - 11:15 IT35310.7KOV4 QoS-verkkojen suunnittelu ja P2_D331 Jarmo | 09:45 - 12:15 IT35310.7KOV4 QoS-verkkojen suunnittelu ja P2_D331 Jarmo | | | 08:00 - 11:15 IT40400.7KOV3 Tietoturvan toteutus P2_D435 Saan Henriik |
| 10:00 | 08:00 - 11:15 IT40400.7KOV3 Tietoturvan toteutus P2_D435 Saan Henriik | | | | |
| 11:00 | | | | | |
| 12:00 | 12:15 - 15:45 IT35310.7KOV4 QoS-verkkojen suunnittelu ja P2_D331 Jarmo | | 12:15 - 15:00 IT50120.7KOV3 Tietoturvan hallinta P2_D435 Rantonen Mika | 12:15 - 15:45 IT35310.7KOV4 QoS-verkkojen suunnittelu ja P2_D331 Jarmo | 12:15 - 15:45 IT40400.7KOV3 Tietoturvan toteutus P2_D435 Saan Henriik |
| 13:00 | 12:15 - 15:45 IT35310.7KOV4 QoS-verkkojen suunnittelu ja P2_D331 Jarmo | 13:15 - 16:45 IT35100.7KOV4 Reaktiivikuvaukset P2_D318 Antero | | | 12:15 - 15:45 IT40400.7KOV3 Tietoturvan toteutus P2_D435 Saan Henriik |
| 14:00 | 12:15 - 15:45 IT40400.7KOV3 Tietoturvan toteutus P2_D435 Saan Henriik | | | 15:00 - 18:00 IT55500.7KOV3 Päätyöväkät P2_D327 Jarmo | |
| 15:00 | | | | | |
| 16:00 | | | | | |
| 17:00 | | | | | |
| 18:00 | | | | | |
| 19:00 | | | | | |

```


```

Kuvio 31. Tilanvaraus html

Kuvio 31. Tilanvaraus html esitetään ote Jyväskylän ammattikorkeakoulun tilanvaraukset ja työjärjestykset näkymästä (<https://amp.jamk.fi/asio/>)

Näiden tietojen perusteella jouduttiin valitettavasti toteamaan, että työn eteenpäin liikkumisen nimissä kalenteri datan käyttäminen työssä jätettäisiin integroimatta toteutukseen.

5 Pohdinta

5.1 Työn eteneminen ja haasteet

Työ lähti liikkeelle suhteellisen notkeasti IBM Bluemixin ja Watsonin opettelulla, joiden ominaisuuksien selvittämiseen pystyisi itsessään tekemään tutkimuksen. Raspberry Pin liittäminen Bluemix palveluihin olisi ollut mahdollista suorittaa useilla eri menetelmillä, joista MQTT:n valittiin käytettäväksi sen keveyden takia. Tiedon siirtäminen Raspberryltä Bluemixiin saatiinkin onnistumaan protokollan yli, jonka jälkeen yritettiin siirtää kuvamateriaalia. Kuvien siirrossa Bluemix:lle tunnistettavaksi törmätettiin ongelmiin, jotka estivät lähetetyn kuvan tunnistamisen. Ongelmaa tutkiessa ei onnistettu löytämään ratkaisua ongelmaan ja päädyttiin suorittamaan tarkastus paikallisesti Raspberry Piin päällä. Tämän johdosta Bluemixin käyttäminen työssä jäi huomattavan vähälle ja sitä käytettiin lähinnä Watson palvelun ylläpitoon.

Watson kuvantunnistus onnistuttiin integroimaan työhön helposti paikallisesti käytettynä IBM:n API rajapinnan kautta, tämä mahdollisti kuvantunnistuksen käyttämisen ja opettamisen suoraan laitteelta. Bluemix ei kumminkaan kyennyt luomaan tunnistetta jostain syystä, jota ei työn aikana onnistettu ratkaisemaan. Tämä prosessi onnistuttiin suorittamaan Raspberry Pi:ltä, vaikka Bluemix olisikin ollut luonnollinen paikka luoda tunnistus. Watsonin opettamiseksi otettiin sarja kuvia keittimestä, joiden avulla tehtävä hoidettiin. Kun tunnistus oli opetettu, niin voitiin tunnistuksen käyttäminen aloittaa.

Bluemixin kanssa törmättyjen ongelmien takia päätettiin asiakas näkymä rakentaa paikalliselle laitteelle, josta loppukäyttäjät voivat nähdä keittimestä otetun kuvan sekä kuvantunnistukselta tulevan palautteen. Tunnistukselta palautuva tieto ja kuva, joka oli lähetetty tunnistettavaksi, oli suhteellisen helppo liittää verkkosivustolle. Verkkosivujen aikaan saamiseksi laitteelle asennettiin Apache2. Tämä mahdollisti kaiken kerätyn tiedon julkaisun helposti samalta laitteelta.

Seuraavaksi yritettiin lisätä kalenteriominaisuutta työhön, joka osoittautui lähes mahdottomaksi ongelmaksi. Toimivan ratkaisun löytäminen kalenteritiedon hakemiselle ja käsittelylle jäi tilanvaraus järjestelmän kohdalla kiinni rahaan, joka olisi mahdollistanut sopivan haun ostamisen järjestelmän kehittäjältä. Office365 kohdalla on-

gelmaksiksi muodostui Microsoftin dokumentaatio heidän omaa tuotettaan kohtaan. Microsoft tallentaa Office365 kalenterin tiedot pilveensä, josta tiedon hakemiseksi sinun pitää olla kirjautuneena organisaatiosi tunnuksilla. Kirjautuminen verkkotunnuksilla ei vielä riitä vaan pitäisi myös pystyä hakemaan tietoa, joka osoittautui alueeksi josta dokumentaatiota ei ollut tarjolla.

Suurin ongelma työn teossa tuntui olevan juuri dokumentaation puute tai Bluemixin tapauksessa dokumentaation virheellisyys. Internet on suuri paikka, josta löytää apua jos osaa etsiä, mutta ohjelmiston kehittäjän omat dokumentaatiot ohjelmiston käytöstä ollessa puutteellisia ja/tai virheellisiä, osoittautuu sovelluksen kehittäminen kyseisen ohjelman tai rajapinnan päälle hankalaksi. Bluemix esimerkiksi kehittyi vielä niin nopeasti, että heidän omat ohjeensa eivät ole onnistuneet pysymään ajan tasalla.

Toiseksi ongelmia aiheuttaneeksi tekijäksi muodostui Jyväskylän ammattikorkeakoulun oma verkko. Lopulta JAMKin tietohallinnon kanssa käydyn neuvottelun pohjalta Raspberry saatiin kiinni verkkoon niin, että toteutuksen käyttäminen on mahdollista niin kehityksen kannalta kuin loppukäyttäjääkin ajatellen.

Työssä onnistuttiin löytämään tapoja, joilla kuvantunnistus tällaisessa tehtävässä on mahdollista suorittaa. Watsonin kuvantunnistuksen vahvuutena olisi sen muokattavuus erilaisiin tehtäviin, mutta sen heikkous tämän hetkessä toteutuksessa on Watsonin tarvitsema vertailujoukko kuvia, joiden pohjalta tunnistuskategoriat on opetettu sille. Koska kuvat kohdistuvat pannuun, saattavat pannun ohitse kulkevat ihmiset tai sen läpi näkyvät esineet häiritä kuvantunnistusta. Mikäli kuva otetaan hetkellä jolloin, pannun edessä seisovalla henkilöllä on musta paita niin tunnistus saattaa kertoa että, pannu on täynnä. Todellisuudessa pannu olisi luultavasti tyhjä, mutta koska tunnistus ei kykene erottamaan vaihteluita pannun edessä tapahtuvassa liikenteessä, niin Watson antaa virheellistä tietoa käyttäjälle. Ongelmasta eroon pääsemiseksi tunnistus pitäisi päivittää tositilanteen mukaisilla kuvilla virheiden minimoimiseksi. Kameran sijoittaminen niin että kuvattava kohde on staattista taustaa vasten eliminoisi tunnistusvirheet.

Alun perin suunnitelmissa ollut kalenteri tiedon hankinta Office365 järjestelmästä jouduttiin lopulta hautaamaan työn edistymisen nimissä. Tiedon saamiseksi julkiseen rajapintaan pyrittiin aluksi jakamaan Office365 kalenteri tiedot Google kalenteriin, jonka julkaiseminen internetissä on suhteellisen vaivatonta. Tämä ei kuitenkaan tuottanut tulosta, joten työssä siirryttiin eteenpäin. Microsoftin API rajapinta, joka mahdollistaa Office365 tunnuksilla kirjautumisen verkko palveluun jouduttiin kanssa hylkäämään. Tätä päätettiin kumminkin yrittää uudelleen myöhemmin kun JAMK tietohallinnolta saatiin pyydettyä Dns (Domain Name System)-nimi Raspberry:lle ja SSL sertifikaatti, joita vaadittiin rajapinnan toimimiseksi julkisessa verkossa. Kumpikaan näistä vaihtoehdoista eivät lopulta tuottaneet tulosta vaikka Office365 tunnuksilla kirjautuminen saatiinkin onnitumaan. Microsoftin ohjeistuksen puute kyseisen API:n käyttöön lopulta pakotti luopumaan hankkeesta.

JAMK Tilanvarausjärjestelmää koitettiin käyttää Office365:n kalenteritietojen puutteen paikkaamiseen, mutta JAMK tietohallinnon mukaan pääsy heidän tietokantaansa, josta tiedot haetaan verkko sivuille, ei ole mahdollista. Tietokanta haulla olisi voinut kertoa kuinka monta opettajaa pitäisi olla paikalla, sen perusteella onko heillä ollut opetusta kyseisenä päivänä ja/tai tietyllä aika välillä. Valitettavasti kalenteri tieto jouduttiin jättämään pois toteutuksesta.

5.2 Jatkokehitys ideoita

Mikäli tämän työn pohjalta haluttaisiin rakentaa prototyyppiä parempi toteutus niin jatkokehitys olisi suositeltavaa. Jatkokehitys kohteina olisivat selvästi kuvan tunnistuksen tarkkuuden parantaminen, jonka toteuttaminen olisi suhteellisen helppoa mikä vaatisi vai toteutuksen pyörimistä ja kuvien keräämistä. Kameran sijainnin vaihtaminen niin, että kuvassa esiintyy vain pannu voisi myös pienentää haitallista vaihtelua.

Toisena kehitys kohteena voisi olla käyttäjätiedon kerääminen siitä, kuinka moni käyttäjä haluaa juotavaa sekä kuinka paljon ja näiden tietojen välittäminen keittäjälle, jonka vuoro on täyttää keitin. Tätä varten toteutukseen voisi olla hyvä lisätä näyttö, jossa esitettäisiin halukkaiden juojien määrä sekä kuinka paljon he haluavat juota-

vaa. Tässä työssä ei lähdetty luomaan näyttöä, sillä pääasiallisena käyttöliittymänä päätettiin pitää verkkokäyttöliittymä, mutta keittäjää ajatellen näyttö, jossa näkyy kuinka paljon juotavaa tulisi olla, voisi helpottaa asiaa. Kosketus näyttöä voisi myös käyttää tiedon ilmoittamiseen.

Suuremman mittakaavan tai kaupallisen toteutuksen luomisen yhteydessä jatkokehitys pitäisi viedä myös itse keittimen automatisointiin, johon tässä työssä ei koskettu. Keitin voitaisiin automatisoida niin, että se annostelee itse ilmoittautuneiden juomien mukaan sopivan määrän ja keittää juomat valmiiksi tiettyyn kellon aikaan. Tästä löytyy jo kotitalouksille tarkoitettuja toteutuksia, jossa käyttäjä voi ilmoittaa keittimelle, että hän haluaa juoda aamulla tähän aikaan valitsemansa juoman.

Työssä käyttäjälle jaettavan tiedon esittämisessäkin olisi parantamisen varaa, mutta tehtävänä oli rakentaa prototyyppi, joten toteutus päätettiin jättää tilaan, jota ei ole hiottu kaupallisen tuotteen tasolle. Jatkoa ajatellen tiedon jakaminen jonkinlaisen graafisesti näyttävämmän käyttöliittymän yli voisi antaa paremman kuvan toteutuksesta.

5.3 Yhteenveto

Omasta mielestäni työ saatiin ajoittain etenemään mallikkaasti, mutta joidenkin osa-alueiden kohdalla ajaututtiin umpikujiin. Nämä söivät paljon aikaa ja mielenkiintoa työn tekemisestä. Tämä lopulta johti työn venymiseen yli arvioidun ajan, jossa työn oli suunniteltu valmistuvan. Työn lähteinä käytettiin lähinnä internetin kautta löytyviä julkaisuja, kuten valmistajien sivustot, foorumit ja how-to oppaat, sekä useita valmistajien manuaaleja. Suurin osa hyödyllisestä materiaalista saatiin käytettyjen ohjelmistojen ja laitteiden historiasta. Vaikka ohjelmistojen ohjeista saatujen tietojen pohjalta onnistuttiin löytämään riittävät tiedot järjestelmän pystyttämiseen, olisi valmistajien ohjeissa etenkin Bluemixillä ollut parantamista ohjeiden päivittämisessä.

Henkilökohtainen osaamiseni työssä käytettyihin ohjelmiin ja menetelmiin oli lähes olematonta, vaikka olinkin entuudestaan tutustunut Raspberry Piin ja Apache2 toimintaan, olivat Node-RED ja Watson minulle aivan uudet aihealueet. Node-RED:n kautta tehtävän ohjelmoinnin opettelussa voisi sujuvasti viettää aikaa yhden opin-

näytetyön verran, sillä tämän työn toteutuksessa käytettiin vain pientä osaa ohjelmistossa tarjolla olevista vaihtoehdoista. JavaScript koodikielen osaaminen olisi työn edetessä antanut paremmat mahdollisuudet luoda koodia. Ilman JavaScript syvällistä osaamista jouduttiin turvautumaan ohjelmiston tarjoamiin tiedonkäsittelysolmuihin. Työn lopputuloksena toteutettu prototyyppi vastaa mielestäni osittain tehtävänantoa. Tehtävänannosta poiketen prototyyppi ei pysty käsittelemään kalenteritietoa, jonka pohjalta oli tarkoitus hakea tieto käyttäjien määrästä. Tällä hetkellä prototyyppi pystyy kaappaamaan kuvamateriaalia keittimen tilasta ja kertomaan käyttäjälle IBM Watsonin kuvantunnistuksen näkemyksen asiasta. Kuvamateriaalin ja Watsonin tunnistuksen palautuksen pohjalta käyttäjä pystyy määrittelemään onko taukotilassa juotavaa tarjolla. Prototyyppi onnistuttiin kumminkin luomaan niin, että se vastaa IBM Watsonin kuvantunnistukselle asetettua tehtävänantoa.

Watsonin kuvantunnistuksen luotettavuus materiaalin tunnistamisessa on työn edetessä todettu olevan hyväksyttävällä tasolla. Watson palauttaa usein samat tunnistustiedot peräkkäin otetuista kuvista, joissa vaihtelua ei ole havaittavissa, vaikka Watsonin antama tunnistuksen varmuus prosentti, jonka Watson antaa käyttäjälle merkinä siitä kuinka varmasti kyseinen asia/esine on tunnistettu, voikin vaihdella joitain prosentteja.

Toteaisin toteutuksen olevan toimiva ja käyttökelpoinen, vaikka prototyypissä olisikin paljon parannettavaa. Työn aikana tuli opittua paljon uutta, josta jotain osia uskon käyttäväni jatkossakin IoT toteutuksissa.

Lähteet

About us. 2017. Raspberry Pi historiaa The Raspberry Pi Foundation verkkosivuilla. Viitattu 9.5.2017. <https://www.raspberrypi.org/about/>.

Apache + PHP + MySQL asentaminen. 2017. Apache2, PHP ja MySQL asennus ja konfigurointiohjeet Linuxille Ubuntu.fi Foorumella. Viitattu 7.5.2017. <https://forum.ubuntu-fi.org/index.php?topic=313.0>.

Bluemix kuvat. 2017. Bluemix verkko-ohjelmiston sivusto. Viitattu 5.4.2017. <https://console.ng.bluemix.net>.

Create a classifier. 2017. Ohjeet Watsonin tunnisteen 'classifier' luontiin IBM Watsonin verkkosivuilla. Viitattu 11.5.2017. https://www.ibm.com/watson/developercloud/visual-recognition/api/v3/#create_a_classifier.

Frequently Asked Questions. 2017. Yleistietoa MQTT:stä protokollan verkkosivuilta. Viitattu 16.5.2017 <http://mqtt.org/faq>.

Go beyond artificial intelligence with Watson. 2017. IBM Watson Esittely sivusto. Viitattu 2.4.2017. <https://www.ibm.com/watson/>.

Guidelines for training classifiers. 2017. IBM Watson ohjeistus tunnistesta heidän verkkosivuillaan. Viitattu 3.4.2017 <https://www.ibm.com/watson/developercloud/doc/visual-recognition/customizing.html>.

History. 2017. Node-RED Historiaa ohjelmiston verkkosivuilla 3.4.2017. Viitattu. <https://nodered.org/about/>.

How To Create a SSL Certificate on Apache for Ubuntu 14.04. 2017. Ohjeet Apache2:n ja SSL sertifiikaatin asennukseen. Viitattu 29.5.2017. <https://www.digitalocean.com/community/tutorials/how-to-create-a-ssl-certificate-on-apache-for-ubuntu-14-04>

Insinööri (AMK), tieto- ja viestintätekniikan tutkinto-ohjelma 2016, 240 op. 2017. Insinöörin tutkinnon opinto suunnitelma JAMK verkkosivuilla. Viitattu 4.4.2017. <https://opinto-oppaat.jamk.fi/fi/opinto-opas-amk/tutkinto-ohjelmat-ja-opintotarjonta/suomenkieliset-opsit/2016-2017/tieto--ja-viestintatekniikka/>.

Jyväskylän ammattikorkeakoulun tilavaraukset ja työjärjestykset - Classroom Booking System. 2017. JAMK tilanvaraus verkkosivu. Viitattu 11.5.2017. <https://amp.jamk.fi/asio/>.

MODEL B SCHEMATICS. 2017. Raspberry Pin suunnittelu piirrustusten julkaisu organisaation verkkosivuilla. Viitattu 4.4.2017. <https://www.raspberrypi.org/blog/model-b-schematics/>.

Node-RED. 2017. Perustietoja ohjelmistosta Node-REDin verkkosivuilla. Viitattu 3.4.2017. <https://nodered.org/>.

PRODUCTS. 2017. The Raspberry Pi Foundation listaus organisaation tuottamista tuotteista heidän verkkosivuillaan. Viitattu 4.4.2017.

<https://www.raspberrypi.org/products/>.

Raspberry Pi 3 Model B. 2017. Raspberry Pin laitetietojen julkaisu The Raspberry Pi Foundation organisaation verkkosivuilla. Viitattu 4.4.2017.

<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>.

RASPBIAN. 2017. Raspbian käyttöjärjestelmän lataussivu. Viitattu 4.4.2017.

<https://www.raspberrypi.org/downloads/raspbian/>.

Revolution Pi Logo. 2017. Revolution Pin Logo tuotteen verkkosivuilla. Viitattu 4.4.2017. <https://revolution.kunbus.com/revpi-core/>.

Running on Raspberry Pi. 2017. Node-RED asennus ohjeet Raspberry Pille. Viitattu 11.5.2017. <https://nodered.org/docs/hardware/raspberrypi>.

SD Cards. 2017. The Raspberry Pi Foundation ohjeita Raspberry Pin muistikortin valinnan helpottamiseksi organisaation verkkosivuilla. Viitattu 4.4.2017.

<https://www.raspberrypi.org/documentation/installation/sd-cards.md>.

Suuntautumisvaihtoehdot. 2017. Tieto- ja viestintätekniiikan suuntautumisvaihtoehdot Jyväskylän Ammattikorkeakoulun Verkkosivustolla. Viitattu 4.4.2017.

<https://www.jamk.fi/fi/Koulutus/tekniikan-ala/insinööri-tieto-ja-viestintätekniiikka/>.

Tutustu ja menesty. 2017. JAMK esittelysivusto. Viitattu 2.4.2017

<http://www.jamk.fi/fi/Tietoa-JAMKista/Tutustu-JAMKiin/>.

What is Bluemix. 2017. IBM Bluemix pilvipalvelun esittely sivusto. Viitattu 2.4.2017.

<https://www.ibm.com/cloud-computing/bluemix/what-is-bluemix?lnk=hm>.

Visual Recognition. 2017. IBM Watson kuvan tunnistuksen esittely sivusto. Viitattu 11.5.2017. <https://www.ibm.com/watson/developercloud/visual-recognition.html>.

Visual Recognition: API Reference. 2017. Ohjeet Watsonin API:n käyttöön IBM Watsonin verkkosivuolla. Viitattu 11.5.2017.

<https://www.ibm.com/watson/developercloud/visual-recognition/api/v3/>

Working / Non-working SD cards. 2017. Lista Raspberry Pillä toimivista muistikortteista eLinux.org verkkosivustolla. Viitattu 4.4.2017. http://elinux.org/RPi_SD_cards.

Liitteet

Liite 1.

Liite 2. Toinen liite