



Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

EERIK LEHTONEN

3D-tulostimen materiaalien seuranta RFID-tageilla

SÄHKÖ-JA AUTOMAATIOTEKNIIKAN KOULUTUSOH-
JELMA
2017

Tekijä(t) Lehtonen, Eerik	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Kuukausi Vuosi 26.5.2020
	Sivumäärä 31	Julkaisun kieli Suomi
Julkaisun nimi 3D-tulostimen materiaalien seuranta RFID-tageilla		
Tutkinto-ohjelma Sähkö- ja automaatiotekniikka		
<p>Tiivistelmä</p> <p>Opinnäytetyön tavoitteena oli toteuttaa 3D-tulostimen materiaalien seuranta RFID:tä hyödyntäen. Työ tehtiin Satakunnan ammattikorkeakoulun käyttöön.</p> <p>Projektissa toteutettiin PHP:lla verkkosivut ja Pythonilla RFID-tagien lukuun käytettävä ohjelma. Lisäksi projektissa hankittiin koululle vaaka tämän projektin tarpeisiin ja muuhun käyttöön. Tässä projektissa sillä punnitaan filamenttirullia. Verkkosivuilta piti pystyä hallitsemaan ja seuraamaan varastotietoja filamenteista.</p>		
Asiasanat RFID, Python, ohjelmointi, PHP, seuranta		

Author(s) Lehtonen, Eerik	Type of Publication Bachelor's thesis	Date Month Year 26.5. 2020
	Number of pages 31	Language of publication: Finnish
Title of publication 3D printer material tracking with RFID tags		
Degree program Electrical and automation engineering		
<p>Abstract</p> <p>The object of this thesis was to make a 3D printer material tracking solution for the Satakunta University of Applied Sciences.</p> <p>The output of this thesis was a website created with PHP and a tag reader program created with Python. A scale was also bought for the school for this thesis and for other use. In this thesis we used it for measuring of the 3D printing filament spools. From the website it was possible to manage and track the inventory of filaments.</p>		
<p>Key words</p> <p>RFID, Python, programming, PHP, tracking</p>		

SISÄLLYS

1 JOHDANTO.....	6
2 PROJEKTIN TOIMEKSIANTO.....	7
2.1 Työn kuvaus.....	7
2.2 Työn rajausta ja riskit.....	7
3 PROJEKTIN LÄHTÖKOHDAT.....	9
3.1 Laitteet.....	9
3.1.1 RFID	9
3.2 Ohjelmointikielet	10
3.2.1 Python.....	10
3.2.2 PHP.....	11
3.3 Tietokannat	12
3.3.1 MySQL	12
3.4 Sovellukset.....	12
3.4.1 Speedway Connect	13
3.4.2 Visual Studio Code.....	13
4 PROJEKTIN ALOITUS	15
4.1 Projektin kuvaus.....	15
4.2 Projektin määrittely	15
4.3 Projektissa käytettävät välineet.....	15
5 PROJEKTIN SUUNNITTELU	16
5.1 Teoreettinen suunnittelu	16
5.2 RFID-laitteisto	16
5.3 Vaaka.....	18
6 PROJEKTIN TOTEUTTAMINEN JA TULOS.....	20
6.1 Palvelin.....	20
6.2 RFID-rajapinta	20
6.3 Tietokannan toteutus	21
6.4 Verkkosivujen toteutus.....	23
6.4.1 Yhteyden muodostaminen	23
6.4.2 Tiedon haku.....	24
6.4.3 Tiedon lisääminen	24
6.4.4 Tietojen listaaminen	24
6.4.5 Tietojen muokkaaminen	25
6.4.6 Tietojen poistaminen ja deaktivoiminen.....	25
6.4.7 Kulutuksen seuraamista varten historia.....	26

6.4.8 Etusivu ja valikko.....	26
6.5 Projektin lopputulos	27
7 PROJEKTIN ARVIOINTI JA PÄÄTTÄMINEN	28
7.1 Arviointi projektin vaiheista ja resurssien hallinnasta	28
7.2 Aikataulu	28
7.3 Projektin tuloksen hyödyntäminen ja arviointi.....	28
LÄHTEET	

1 JOHDANTO

Projektissa oli tarkoituksena tehdä RFID:n avulla 3D-tulostusmateriaalin seuranta SAMK:ille. Ideana oli saada inventaario materiaalista ja sen määrästä. Tämän oli tarkoitus onnistua RFID-lukijalla ja vaa'alla, joista luetaan tiedot ja listataan ne tietokantaan. Projekti tehtiin helpottamaan inventaarion seuranta ja materiaalin kulutusta.

Tehtäväni projektissa oli määritellä käytettävät rajapinnat, ohjelmointikielet ja vaaka. Näiden määrittämisen jälkeen piti vielä hankkia vaaka ja toteuttaa ohjelmoinnilla valituilla rajapinnoilla sovellus.

2 PROJEKTIN TOIMEKSIANTO

2.1 Työn kuvaus

Saamani toimeksianto oli tehdä 3D-tulostus materiaalille RFID:llä seuranta. 3D-tulostus materiaalina toimii filamentti eli muovilanka. Filamentteja löytyy useita eri laatuja ja ne voivat olla eri värisiä, vahvuisia tai materiaaleja (3DJAKE [www-sivut](#) 2020). Toiminnaksi kuvattiin, että tullaan filamenttirullan kanssa toimipisteelle ja asetetaan se vaa'an päälle. Tämän jälkeen verkkosivuilta löytyy kaikki toiminnallisuus. RFID-lukijalta saadaan EPC ja vaa'alta saadaan painotieto. Näin saataisiin seurattua materiaalin tilannetta varastossa ja kulutusta. Näistä tiedoista nähtäisiin myös, koska tarvitaan varastoon lisäyksiä. Paino on ensisijainen mittaustyökalu tässä projektissa. Materiaalista piti ylläpitää tietoja ja näitä oli esimerkiksi kokonaispaino, rullan paino, materiaali, halkaisija ja väri. Käytettäväksi välineiksi sain koululta RFID-lukijan, antennin ja tehtäväkseni jäi vaa'an hankkiminen.

2.2 Työn rajausta ja riskit

Työn rajausta pohdittiin sen mukaan mitkä asiat olisivat järkeviä toteuttaa aikataulun mukaisesti ja olivat täysin välttämättömiä toiminnalle. Projektin aloitusta varten rajattiin toimintaan välttämättömäksi tietokanta ja verkkosivut, joista pystyi hallitsemaan olemassa olevia tietoja. Tätä varten pitää olla ensin joku tapa saada tietoa laitteilta tietokantaan. Myös kirjautuminen oli pyydetty osa projektia, mutta ei pakollinen, jos aikataulu ei siihen riitä. Lisäksi sivujen ulkoasu rajattiin heti alussa projektin ulkopuolelle.

RFID-lukijan ja vaa'an rajapinnat olivat TCP-pohjaiset, joten niitä tulisi olla suhteellisen helppo käsitellä. Palvelimeen on SSH-yhteys.

Työllä oli useita riskejä. Vaaka piti tilata ja sen toimitusaikaa ei ollut tiedossa. Jos jotain pitäisi tehdä etänä, toimisivatko yhteydet. Lisäksi Covid-19 tilanne vaikutti aikatauluihin ja työn kulkuun. Koulut olivat kiinni ja työtä ei päässyt testaamaan

lopullisessa sijoituspaikassa. Käyttönoton testaus jää tekemättä tästä syystä ja toimivuutta paikan päällä ei voitu varmistaa.

3 PROJEKTIN LÄHTÖKOHDAT

3.1 Laitteet

Projektissa oli alussa käytössä RFID-lukija ja sen antenni. Lisäksi projektiin tuli myöhemmin mukaan vaaka. RFID-lukija ja antenni annettiin koulun puolelta valmiiksi valittuna. Siinä oli kaikki riittävät ominaisuudet tähän projektiin. Vaaka hankittiin myöhemmin annettujen vaatimusten perusteella.

3.1.1 RFID

RFID (Radio Frequency Identification) on radiotaajuuksilla toimivan tekniikan nimitys. RFID teknologialla voidaan siirtää tietoa automaattisesti radiotaajuuksilla etänä. Tiedon siirtämiseen tarvitaan RFID-tunniste ja RFID-lukija. RFID-tunnisteilla tai tageilla on sisällään siru, antenni ja vähän muistia. Jokaisessa tunnisteessa voi olla yksilöllinen tunnistenumero. Näihin voidaan lisätä tietoja tuotteesta, valmistusprosesseista, ajasta, paikasta, yms. Etäisyyden mukaan tunniste tai tagi luetaan eri taajuuksialueilla. Kuvassa 1 esitelty miten RFID toimii.



Kuva 1. Miten RFID toimii? (Toptunniste [www-sivut](http://www.sivut) 2020).

Tunnisteen lukeminen tapahtuu joko RFID-lukijalaitteella tai mobiililaitteella. Lukija voi olla käsi­käyttöinen tai kiinteästi asennettu lukupisteeseen asennettu automaattinen lukija, kuten sisääntuloporttiin. Jos lukijassa ei ole integroitua antennia sellainen pitää

asentaa. Lisäksi RFID-järjestelmät vaativat usein, jonkun taustajärjestelmän missä saatua tietoa käsitellään (Toptunniste [www-sivut 2020](#)). Tägeja on passiivia ja aktiivisia. Aktiivisella on parempi kantama, mutta hinta on korkeampi, koska tagi sisältää akun. Passiivisella on lyhyempi kantama, mutta ovat halvempia (atlasRFIDstore [www-sivut 2020](#)). RFID tagin tunniste on nimeltään EPC (Electronic Product Code). Sillä voidaan antaa jokaiselle tagille oma uniikki tunniste (EPC-RFID INFO [www-sivut 2020](#)).

RFID tageja käytetään esimerkiksi eläinten seurantaan ja tuotantolaitoksissa (Margaret Rouse 2019). Yksi suuri RFID teknologian käyttäjä on armeija. Aseiden sijainteja voidaan seurata etänä tai sotilaiden liikkeitä. Lisäksi teollisuudessa tageja käytetään paljon. Esimerkiksi inventaarion seuraaminen tageilla on yleistä. Varaston saldon tietystä tuotteesta voidaan tarkistaa helposti lukijoilla (Shahid Mansuri 2019).

3.2 Ohjelmointikielet

Tässä projektissa käytettiin ohjelmointikielenä PHP:ta ja Pythonia, koska Pythonilla oli nopea tehdä RFID-lukijaa varten rajapinta ja PHP:lla tietokannan käyttäminen ja käyttäjän syötteen käsittely onnistui hyvin. Lisäksi osasin jo molemmista kielistä perusteet niin työskentely oli sujuvaa näillä kielillä, ja ne täyttivät kaikki vaadittavat toiminnallisuudet tähän projektiin.

3.2.1 Python

Python on interaktiivinen olio-ohjelmointi kieli. Pythonissa on moduulit, virheen poiminta, dynaaminen kirjoitus, korkean tason dynaamiset muuttujat ja luokat. Python on helposti toiselle alustalle siirrettävä (Python Software Foundation [www-sivut 2020](#)). Pythonin etu moneen kieleen verrattuna on, että sen voi ajaa suoraan. Sitä ei tarvitse erikseen alkaa kääntämään, kuten osia kielistä. Lisäksi Python muistuttaa ulkoasultaan englannin kieltä, mikä tekee siitä helposti kirjoitettavan. Pythonin uusin versio on 3 (Wc3schools [www-sivut 2020](#)) Python on myös parhaimmaksi arvosteltu ohjelmointikieli vuonna 2019 kolmatta vuotta putkeen IEEE:n listoilla (Full Stack Python [www-sivut. 2020](#)). Pythonia käytetään esimerkiksi Wikipediassa, Googlessa, Yahoo!:ssa, CERN:issä ja NASA:ssa. Pythonilla tehdään esimerkiksi käyttöjärjestelmän tärkeitä

toimintoja, verkkosivuja ja joskus myös pelejä. Python-koodia on käytössä joka puolella. Esimerkiksi YouTubessa, Instagramissa ja Dropboxissa. Dropboxin koodista suurin osa on Pythonia (Code institute www-sivut 2020). Kuvassa 2 on vielä lyhyt esimerkki Pythonilla tehdystä ohjelmasta. Ohjelma tulostaa tekstin Hello, World!

```
print("Hello, World!")
```

Kuva 2. Esimerkki Pythonista.

3.2.2 PHP

PHP on vuonna 1994 tehty kieli. PHP on muuttunut monta kertaa vuosien varrella version muutosten välissä (The PHP Group www-sivut 2020). Uusin PHP:n versio on 7.4.5 (The PHP Group www-sivut 2020). Kuvassa 3 lyhyt esimerkki yksinkertaisesta PHP-ohjelmasta, joka tulostaa tekstiä.

```
<!DOCTYPE html>
<html>
<body>

<?php
echo "My first PHP script!";
?>

</body>
</html>
```

Kuva 3. Esimerkki PHP:sta.

PHP lyhenne sanoista PHP: Hypertext Preprocessor. PHP on HTML sisäistetty ohjelmointikieli, jonka syntaksissa on otettu paljon lainauksia C, Java ja Perl kielistä (The PHP Group www-sivut. 2020).

PHP tukee olio-ohjelmointia ja funktionaalista ohjelmointia (PHP: The "Right" Way 2016, 3).

Marraskuussa 2018 PHP:tä käytettiin 80 % verkkosivuista palvelinpuolen ohjelmointikielenä. PHP:llä tehdään yleensä dynaamisia verkkosivuja. Dynaamiset www-sivut

yleensä ottavat vastaan käyttäjältä tietoa ja sen perusteella tekevät jotain. PHP:lla esimerkiksi käsitellään käyttäjän syötettä tai asetetaan keksit. Sitä voidaan käyttää tekemään myös komentoriviohjelmia. PHP on myös palvelinkieli. Kun käyt sivulla, jossa käytetään PHP:ta selaimesi ei suorita PHP:ta vaan palvelin tekee sen ja näet HTML:län mitä PHP tuotti (freeCodeCamp [www-sivut](#) 2020).

3.3 Tietokannat

3.3.1 MySQL

MySQL on yleisin avoimen lähdekoodin SQL (SQL Structured Query Languagea) relaatiotietokanta. Sen omistaa Oracle. MySQL on yksi tarjolla olevista tietokanta ratkaisuista (Oracle Corporation [www-sivut](#). 2020).

Tiedot talletetaan tauluihin, jotka on jaettu pienempiin palasiin, joita sanotaan kentiksi. SQL:stä tuli standardi ANSI-standardi vuonna 1968 ja ISO-standardi vuonna 1987. Vaikka SQL onkin standardisoitu, jokaisella valmistajalla on omat eronsa toteutuksessa. Samat peruskomennot löytyvät silti kaikista. MySQL on Relational Database Management System (RDBMS). RDBMS on pohjana kaikille moderneille tietokannoille. Tiedot talletetaan objekteihin ja tauluihin (W3schools [www-sivut](#) 2020). Kuvassa 4 esimerkki SQL-lauseesta.

```
SELECT * FROM Customers;
```

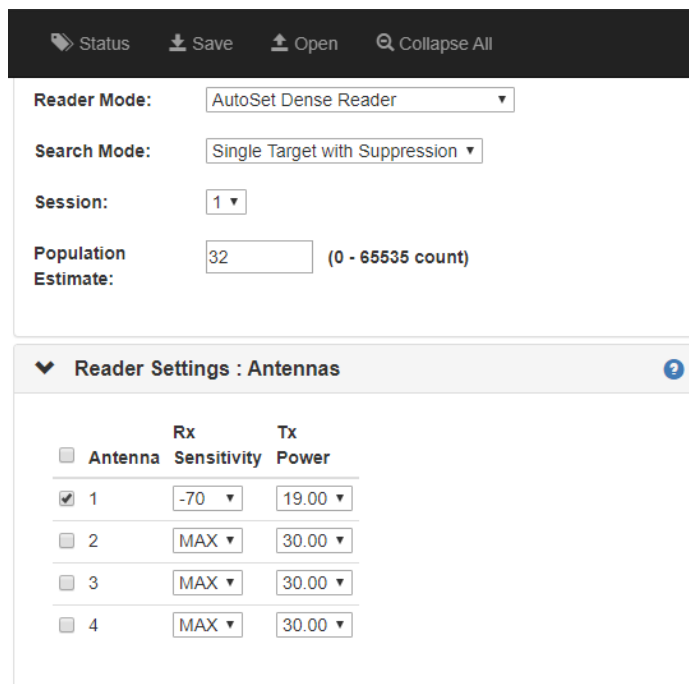
Kuva 4. Esimerkki SQL lauseesta.

3.4 Sovellukset

Tässä projektissa käytin Speedway Connectia RFID-lukijan asetusten hallintaan. Sieltä saatiin helposti vaihdettua lukijan asetukset projektiin sopiviksi ja asetettua lukijan TCP-serveri tilaan Python-ohjelmointia varten. Koodin kirjoittamiseen Visual Studio Codea sen helpon laajennettavuuden ja hyvien etäyhteyksien hallinnan takia. Näin kirjoitettua koodia suoraan etänä ja testattua nopeasti.

3.4.1 Speedway Connect

RFID-lukijassa oli käytössä Speedway Connect ohjelmisto, jonka avulla lukijasta saitehtyä TCP-serverin RFID-tagin tietojen hakua varten. Speedway Connectin avulla lukijaa saa hallittua etänä ja sen graafinen käyttöliittymä on helppokäyttöinen, vaikka RFID-teknologia ei olisi entuudestaan tuttua (IMPINJ Support www-sivut 2020). Ku-
vassa 5 esimerkki Speedway Connectin käyttöliittymästä.



The screenshot displays the Speedway Connect software interface. At the top, there is a dark header bar with icons and labels for 'Status', 'Save', 'Open', and 'Collapse All'. Below this, the main configuration area includes several settings:

- Reader Mode:** A dropdown menu set to 'AutoSet Dense Reader'.
- Search Mode:** A dropdown menu set to 'Single Target with Suppression'.
- Session:** A dropdown menu set to '1'.
- Population Estimate:** A text input field containing '32', with a range indicator '(0 - 65535 count)' to its right.

Below these settings is a section titled 'Reader Settings : Antennas' with a blue help icon. It contains a table with four columns: 'Antenna', 'Rx Sensitivity', and 'Tx Power'.

Antenna	Rx Sensitivity	Tx Power
<input checked="" type="checkbox"/> 1	-70	19.00
<input type="checkbox"/> 2	MAX	30.00
<input type="checkbox"/> 3	MAX	30.00
<input type="checkbox"/> 4	MAX	30.00

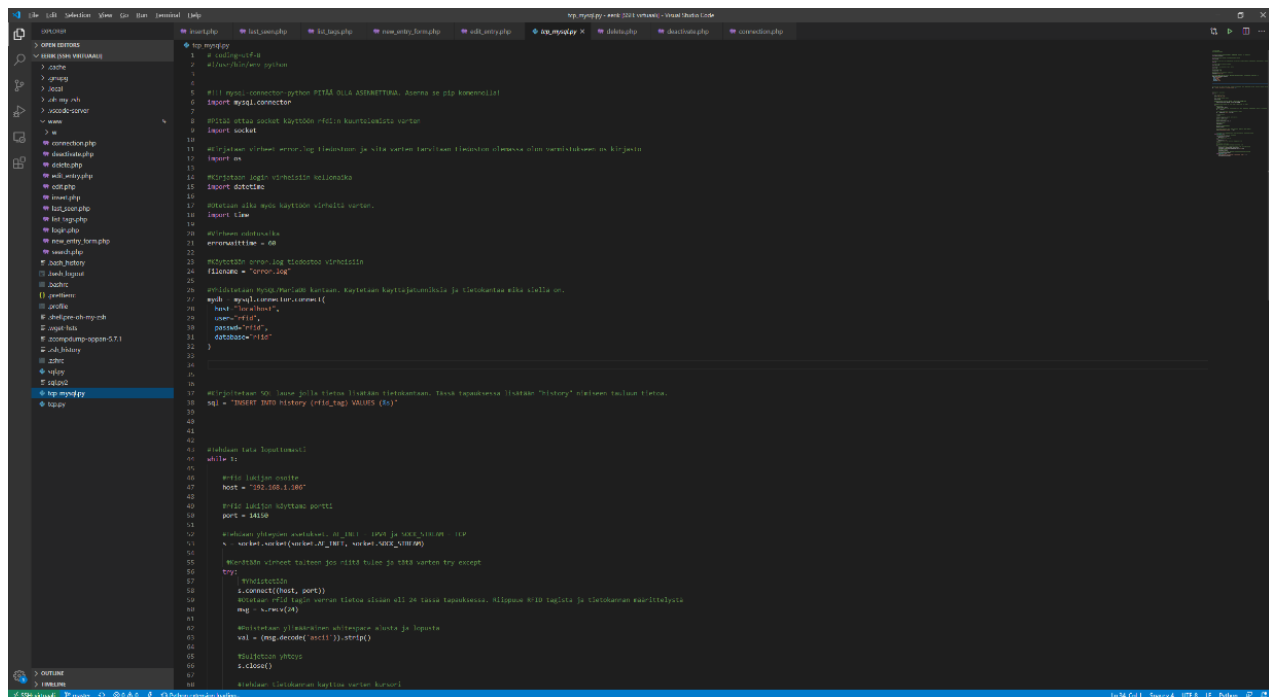
Kuva 5. Speedway Connect käyttöliittymä.

3.4.2 Visual Studio Code

Visual Studio Code on Microsoftin ilmainen koodieditori. VS Code on myös avointa lähdekoodia ja se on jaettu Githubissa (Visual Studio Code www-sivut 2020)

VS Code sisältää kehitystyökaluja, kuten koodintäyttöä ja debug työkaluja (Visual Studio www-sivut 2020).

Erona Microsoftin toiseen tuotteeseen Visual Studioon on se, että Visual Studio on kokonainen IDE (integrated development enviroment) ja sisältää enemmän ominaisuuksia mitä käytetään ohjelmointia tehdessä mukaanlukien laajemmat debug työkalut (Sloan 2018). Kuva 6 sisältää esimerkin Visual Studio Coden käyttöliittymästä.



Kuva 6. Visual Studio Code.

Työtä varten valittiin käytettäväksi editoriksi Visual Studio Coden sen helpon laajennettavuuden takia. Sain siihen kaikki käyttämäni kielet tuetuiksi ja liitettyä sen suoraan etäyhteydellä palvelimelle missä työskentelin. Näin saatiin koodista helposti luettavaa ja nopeasti testattavaa.

4 PROJEKTIN ALOITUS

4.1 Projektin kuvaus

Projektissa oli tilattu RFID-lukijalle toteutettu sovellus, josta voidaan seurata 3D-tu-
lostus materiaalin varastoa. Varastolle määriteltiin tietyt ominaisuudet mitä sen pitäisi
pystyä toteuttamaan. Kieliä ei ollut määritelty, mutta toiveissa oli esitetty PHP ja Pyt-
hon. Käyttöliittymän ulkoasu ei kuulunut projektiin. Se tehdään myöhemmin erillisenä
projektina.

4.2 Projektin määrittely

Toiminallisuutta määriteltiin, niin että tietokannassa piti olla RFID-tagin EPC, paino,
rullan paino, väri, materiaali ja aktiivinen tila ja niiden lisäksi muutama lisäkenttä
myöhempiä mahdollisia toiminnallisuuksia varten. Sivulla pitää pystyä lisäämään
vaa'alla olevan RFID-tagin perusteella tuotteen paino. Tuotetta lisätessä täytetään ken-
tät ja päivittäessä vaihdetaan vain paino uuteen painoon. RFID-tagin kaikkia tietoja
piti pystyä päivittämään myöhemminkin, jos joudutaan vaihtamaan esimerkiksi käy-
tössä oleva rulla toiseen eri painoiseen rullaan.

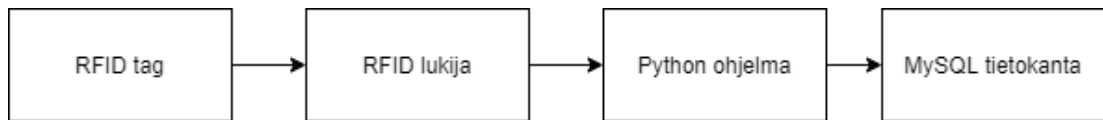
4.3 Projektissa käytettävät välineet

Projektia varten oli koululta RFID-lukija ja antenni. Työtä varten piti myös löytää so-
piva vaaka ja sellainen tilattiin. Vaa'aksi tuli lopulta Kern KFP-V20. Siihen päädyttiin
hinnan ja ominaisuuksiensa puolesta. Vaa'an päälle piti mahtua 3D-materiaalin rullat
ja kestää niiden paino. Lisäksi vaa'an piti olla tarpeeksi tarkka muuhunkin käyttöön
kuten kappalelaskurina toimimiseen ja tämä ominaisuus kyseisestä vaa'asta löytyy,
mutta tärkein ominaisuus omalle projektilleni oli etäyhteys. Valittu vaaka tarjosi hyvän
sarjadata/ethernet-muuntimen ja sitä kautta vaa'alta sai tiedon etänä.

5 PROJEKTIN SUUNNITTELU

5.1 Teoreettinen suunnittelu

Projektissa piti siis suunnitella verkkosivut, tietokanta ja Python-ohjelma. Ensiksi suunniteltiin tietokannan skeeman mitä lähdettiin toteuttamaan. Sen jälkeen tehtiin Pythonilla testiohjelma, joka osasi kuunnella TCP-porttia ja sitä kautta tulostaa EPC-koodin, minkä antenni on viimeksi nähnyt. Kuva 7 esittää ohjelman teoreettista toiminnallisuutta.



Kuva 7. RFID-lukijalta tiedon siirto tietokantaan.

Tämän jälkeen oli aika lisätä tietokantaan tiedon tallentaminen. Kun saatiin tämän toimimaan, lähdettiin toteuttamaan verkkosivun toteutusta.

5.2 RFID-laitteisto

Sain koululta valmiiksi valitut RFID-laitteet työtä varten. Käytössä oli Speedway Revolution lukija. RFID-lukijassa oli kaikki tarvittavat ominaisuudet. Siinä oli hyvät hallinta ominaisuudet etänä ja sen sai tulostamaan viimeksi nähdyn tagin EPC:n helposti etänä lukemista varten. Kuva 8 on RFID-lukijasta, joka oli projektissa käytössä.



Kuva 8. RFID-lukija.

Testauksessa käytössä oleva antenni ei todennäköisesti tule käyttöön työn valmistuttua. Antenni on epäkäytännöllisen iso ja turhan tehokas projektin suunniteltuun käyttöön. Lopullisessa asennuksessa ideana oli käyttää pienempää lyhyen matkan antennia. Kuva 9 on antennista joka projektissa käytössä.



Kuva 9. RFID-lukijan antenni.

5.3 Vaaka

Työtä varten piti koulun hankkia vaaka. Projektia varten selvitettiin vaihtoehtoja ja Kern KFP-V20 täytti vaikkei vaaditut ehdot. Näyttönä on Kern KFS-TM. Siinä riittää tarkkuus 3D-materiaalin punnitsemiseen ja sen päälle mahtuu rullat hyvin. Laite sisältää myös ominaisuuksia kappaleiden laskemiseen ja muuhun mahdollisesti koulussa tarvittavaan. Kuva 10 on vaa'asta, joka hankittiin projektia varten.



Kuva 10. Vaaka.

Vaaka piti saada myös etäkäyttöön. Sitä varten piti hankkia sarjadata/ethernet-muunnin. Tästä sai tehtyä TCP-serverin samoin, kuin RFID-lukijasta. Joten samalla

periaatteella kuin RFID-lukijan kanssa oli toimittu, pystyi tekemään Python-ohjelman joka kuuntelee TCP-porttia. Kuva 11 on sarjadata/ethernet-muuntimesta.



Kuva 11. Vaa'an serial portista ethernet muunnin.

6 PROJEKTIN TOTEUTTAMINEN JA TULOS

6.1 Palvelin

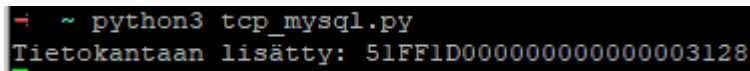
Koululta saatiin käyttöön Debian Linux virtuaalipalvelimen työn testaamista varten. Debian avoimen lähdekoodin käyttöjärjestelmä, joka käyttää Linux- tai FreeBSD-ydintä (Debian [www-sivut 2020](#)) Debian on viidenneksi suosituin Linux-käyttöjärjestelmä kuuden kuukauden aikana (Distrowatch [www-sivut 2020](#)). Debian on suosittu palvelinkäytössä, koska se on avointa lähdekoodia, vakaa, turvallinen ja helposti päivitettävä käyttöjärjestelmä (Pontikis [www-sivut 2020](#)).

Palvelin on verkkosivuja ja tietokantaa varten. Python ohjelman voi ajaa joko palvelimella tai jossain toisessa tietokoneessa, mutta tiedot tallennetaan palvelimelle.

6.2 RFID-rajapinta

Valitsin Pythonin RFID-lukijan kanssa kommunikointiin ja vaa'an kanssa kommunikointiin, koska se on helposti luettavaa ja ylläpidettävää koodia. Lisäksi osasin siihen jo perusteet valmiiksi (The Hello World Program [www-sivut. 2020](#)). Omaan työtäni varten tarvitsin socket ja tietokanta ohjelmointia (Python Software Foundation [www-sivut. 2020](#)). Sen toteuttaminen Pythonilla oli helppoa. Pythonin omista dokumenteista löytyivät hyvät ohjeet perusohjelman tekemiseen. Lisäksi minulla oli hallussa jo Python-kielen perusteet entuudestaan.

Ensiksi tein Pythonilla socket client komentoriviohjelman. Tällä kuuntelin RFID-lukijan tietoa TCP:llä. Uuden tiedon saavuttua tulostetaan se näkyviin. Seuraavaksi lisäsin tietokantaan kirjoittamisen sen jälkeen, kun olin suunnitellut tietokannat. Nyt tagi, joka on nähty viimeksi, kirjataan ylös kantaan. Lopuksi tein virheen kirjauksen tiedostoon, jos tietokantaan ei pystytty kirjoittamaan. Kuva 12 esittää python ohjelman toiminnallisuutta.



```
~ python3 tcp_mysql.py
Tietokantaan lisätty: 51FF1D00000000000000003128
```

Kuva 12. Python-ohjelman tulostus näkymä.

6.3 Tietokannan toteutus

MySQL on yleisessä käytössä. Sitä käyttävät muun muassa Facebook ja Twitter (MySQL www-sivut. 2019). MySQL on ollut monien yritysten käytössä jo vuosia ja sitä on helppo ylläpitää (Hostinger www-sivut 2020).

Näillä perusteilla päädyin valitsemaan MySQL tietokannan.

Ensiksi suunnittelin tietokannan Pythonia varten. Tämä oli hyvin yksinkertainen toteutus millä kirjattiin vain viimeksi nähty tieto ja viimeksi nähtyä tietoa käytetään eri vaiheissa projektia. Tästä saadaan historia tarvittaessa, mitkä tagit ovat nähty ja mihin kellonaikaan jatkokehitystä varten. Kuva 13 esittää history tietokannan taulun.

history	
PK	<u>id (INT)</u>
	rfid_tag (VARCHAR 25)
	timestamp (TIMESTAMP)

Kuva 13. Tagi historia taulu.

Tämän jälkeen piti alkaa suunnitella tietokantoja PHP:tä varten. Tarvitsin tietokannan mihin viimeksi nähty tagi ja sen tiedot lisätään. Tietokannassa on id, joka on juokseva luku. Tällä varmistetaan, että tietokannassa jokaisella tietueella on yksilöllinen tunniste. Sen jälkeen on rfid_tag, joka on uniikki. Kahta samaa tagia ei tietokantaan ei saa laittaa. Näin estetään virheet. Tietokannasta löytyy sen lisäksi painoille paikat ja onko tietue aktiivinen vai ei. Ja määrittelyssä oli muutama ylimääräistä jatkokehitystä varten, joten niitä löytyy kolme kappaletta. Kuva 14 esittää tags tietokannan taulun.

tags	
PK	<u>id (INT)</u>
UNIQUE	rfid_tag (VARCHAR 25)
	weight (INT)
	spool_weight (INT)
	color (VARCHAR)
	firstseen (TIMESTAMP)
	active (INT)
	material(VARCHAR 45)
	extra1(VARCHAR)
	extra2(VARCHAR)
	extra3(VARCHAR)

Kuva 14. Tags taulu.

Tarvitsin myös tietokannat viimeksi nähdyn painon ja kulutuksen seuraamiseen. Viimeksi nähtyjä painoja varten tein samanlaisen kannan, kuin RFID-tagien seuraamista varten olin tehnyt. Kuva 15 esittää weight_history taulun.

weight_history	
PK	<u>id (INT)</u>
	weight (INT 11)
	timestamp (TIMESTAMP)

Kuva 15. Paino historia tietokanta.

Materiaalin kulutusta varten tein tietokannan material_usage, johon lisättiin triggerillä automaattisesti tietoja, kun tags tauluun tehtiin päivityksiä. Kantaan kirjataan RFID-tagin, uusi paino, uusi rullan paino, vanha rullan paino, vanha paino ja aikaleima. Ideana oli, että tätä tietoa voidaan käyttää jatkokehityksessä myös. Kuva 16 esittää material_usage taulun.

material_usage	
PK	<u>id</u> (INT)
	rfid_tag (VARCHAR 25)
	weight (INT)
	spool_weight (INT)
	old_spool_weight (INT)
	old_weight (INT)
	timestamp (TIMESTAMP)

Kuva 16. Materiaalin kulutuksen seuranta tietokanta.

6.4 Verkkosivujen toteutus

Verkkosivujen toteutuksessa päädyin valitsemaan PHP:n sen yleisyyden takia. Esimerkiksi WordPress toimii PHP:lla. PHP:ta on käytössä kaikkialla internetissä. Se on helppokäyttöistä myös ja html tiedostot on helppo muuttaa PHP tiedostoiksi vaihtamalla tiedoston nimen vain .php:ksi (David 2018). PHP:ssa on myös hyvä tuki tietokantoihin (Angela Bradley 2019). PHP on serveri puolen kieli ja sitä ei saa sekoittaa HTML:llään, joka näkyy käyttäjille sivuilla (The PHP Group www-sivut. 2020).

6.4.1 Yhteyden muodostaminen

Ensin tein PHP-tiedoston missä muodostetaan tietokantayhteys. Jos yhteyden muodostaminen tietokantaan onnistui, tulostettiin tieto onnistumisesta. Virheen sattuessa tulostetaan virheilmoitus ja tietoa virheestä. Kuvassa 17 nähdään onnistuneen yhteyden muodostamisen ilmoitus ja kuvassa 18 nähdään esimerkki virheilmoituksesta.

Connected successfully

Kuva 17. Onnistunut yhteyden muodostaminen.

Connection failed: SQLSTATE[HY000] [1045] Access denied for user 'rfid'@'localhost' (using password: YES)

Kuva 18. Esimerkki virheestä.

6.4.2 Tiedon haku

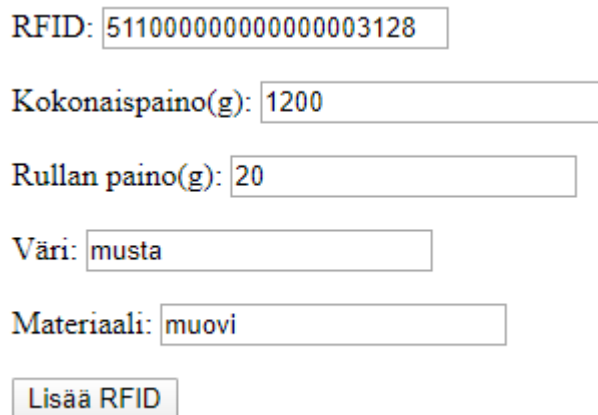
Seuraavaksi oli vuorossa tiedon hakua kannasta. Ideana oli pystyä lisäämään tai muokkaamaan viimeiseksi nähtyä tagin tietoa. Tein PHP-tiedoston, joka hakee viimeisen RFID-tagin ja tulostaa sen. Kuva 19 on esimerkki toiminnallisuudesta.

Viimeksi nähty RFID: 511000000000000003128

Kuva 19. Esimerkki viimeksi nähdystä RFID-tagista.

6.4.3 Tiedon lisääminen

Tämän jälkeen oli vuorossa tehdä tietojen lisäämistä varten oma PHP-tiedosto. Tässä käytettiin viimeksi nähtyä tagia ja ideana on, että paino saadaan myös vaa'alta tähän automaattisesti. Muut tiedot pitää täyttää käsin. Jos tiedon lisääminen onnistui siitä, ilmoitetaan käyttäjälle tai virheen sattuessa tulostetaan virhe käyttäjälle. Kuva 20 on esimerkki tiedon lisäämisestä.



RFID: 511000000000000003128

Kokonaispaino(g): 1200

Rullan paino(g): 20

Väri: musta

Materiaali: muovi

Lisää RFID

Kuva 20. Uuden RFID-tagin lisääminen tietokantaan.

6.4.4 Tietojen listaaminen

Nyt kun uusia tietoja pystyi lisäämään tein seuraavaksi tavan katsoa tietokannassa olevia tietoja. Tätä varten tulostetaan tags taulun tiedot näkyviin ja lasketaan jäljellä oleva materiaali vähentämällä kokonaispainosta rullan paino. Lisäksi tulostetaan kolme nappia tiedon muokkaamista varten. Muokkaa avaa tiedon muokkauksen, deaktivoi

asettaa RFID-tagin tiedot epäaktiiviseksi ja piilottaa ne, kun taas poista tuhoaa tiedot kokonaan. Kuva 21 on esimerkki tietojen listaamisesta.

RFID tag	Kokonaispaino(g)	Rullan paino(g)	Materiaalia jäljellä(g)	Väri	Materiaali	
511000000000000003128	5	3	2	musta	pahvi	<input type="button" value="Muokkaa"/> <input type="button" value="Deaktivoi"/> <input type="button" value="Poista"/>

Kuva 21. Lista tietokannan sisällöstä.

6.4.5 Tietojen muokkaaminen

Muokkaa nappi avaa sivun, joka muistuttaa hyvin paljon uuden tagin lisäämistä. Napin painamisen jälkeen käyttäjälle tulostetaan tieto onnistumisesta tai virheilmoitus ja virheen tiedot. Kuva 22 on esimerkki tiedon muokkaamisesta.

RFID:

Paino(g):

Rullan paino(g):

Väri:

Materiaali:

Kuva 22. Muokkaa tietoja.

6.4.6 Tietojen poistaminen ja deaktivoiminen

Deaktivoi ja poista napit vievät käyttäjän sivulle, jossa varmistetaan vielä, haluaako käyttäjä tehdä kyseisen toimenpiteen vai ei. Näin estetään vahingossa tietojen hävittäminen. Deaktivoiminen oli active tiedon muuttaminen tietokannassa ykkösestä nol-laksi. Tageja, joiden active tilan tieto on nolla ei listata, mutta tiedot ovat yhä olemassa. Poistaminen poistaa kyseisen tagin tiedot kokonaan. Kuva 23 on esimerkki tiedon deaktivoimisesta ja varmistuksesta.

Oletko varma, että haluat deaktivoida RFID tagin: 511000000000000003128?

Deaktivoi

Kuva 23. Varmistus sivu.

6.4.7 Kulutuksen seuraamista varten historia

Viimeiseksi lisäsin vielä sivun, josta nähdään materiaalin kulutus. Sivun listaa tagin, käytetyn materiaalin määrän, paljonko materiaalia on vielä jäljellä ja aikaleiman tapahtumasta. Tällä sivulla käytetään material_usage taulua. Sieltä haetaan vanhat painot ja niistä vähennetään uusi paino ja näin saadaan kulutus laskettua. Tarvittaessa tähän sivuun voisi lisätä muitakin ominaisuuksia. Kuva 24 on esimerkki historia näkymästä.

RFID tag	Käytetyn materiaalin määrä(g)	Materiaalia jäljellä(g)	Timestamp
511000000000000003129 0	7		2020-05-20 11:29:04
511000000000000003129 0	7		2020-05-20 11:29:28
511000000000000003129 0	7		2020-05-20 11:29:37
511000000000000003122 0	7		2020-05-20 11:39:36
511000000000000003122 0	7		2020-05-20 11:44:07
511000000000000003122 0	7		2020-05-20 11:52:39
511000000000000003122 0	7		2020-05-20 11:52:52
511000000000000003122 0	7		2020-05-20 11:56:01
511000000000000003122 0	7		2020-05-20 11:56:27
511000000000000003129 0	7		2020-05-20 12:07:42
511000000000000003129 3	4		2020-05-20 13:04:39

Kuva 24. Historia sivu.

6.4.8 Etusivu ja valikko

Lopuksi tein etusivun ja valikon. Valikossa on vain listattuna kaikki käytettävät sivut, mitä käyttäjälle on tarjolla. Etusivulla näytetään viimeksi nähty tagi ja viimeksi rekisteröity paino. Tiedot haetaan tags_history ja weight_history tauluista. Niistä otetaan uusin lisätty tieto. Jos tagi on jo olemassa kannassa, näytetään paljonko materiaalia, on jäljellä. Päivitä tiedot nappi päivittää viimeksi nähdyn tagin painoksi viimeksi nähdyn painon. Kuva 25 näyttää etusivun ja valmiin kokonaisuuden.

[Etusivu](#) | [Tagi lista](#) | [Historia](#)

Viimeksi nähty RFID: 511000000000000003129

Viimeksi rekisteröity paino: 10g

Materiaalia jäljellä: 4g

Päivitä tiedot

Kuva 25. Etusivu ja valikko, jos tieto oli jo olemassa.

Uuden tagin lisääminen vie aikaisemmin tehdylle sivulle ja täyttää siihen valmiiksi painon ja tagin tiedot. Kuva 26 on esimerkki uuden tiedon lisäämisestä.

Viimeksi nähty RFID: 511000000000000003130

Viimeksi rekisteröity paino: 10g

Tagia ei löytynyt tietokannasta.

Lisää uusi tagi

Kuva 2. Uuden tagin lisääminen.

6.5 Projektin lopputulos

Projektin tuloksena oli verkkosivut, tietokanta ja Python-ohjelma. Valitettavasti vaa'an saapumisen myöhästymisen takia Python-ohjelman muokkaaminen painotietojen vastaanottamiseen jäi tekemättä. Pohja sille on jo olemassa, mutta viimeinen hienosäätö toimivuudesta puuttuu. Tämä tarkoittaa sitä, että automaattinen painon hakeminen vaa'alta ei tekemässäni projektissa toimi. Projektin tilaajalle sain toimitettua verkkosivut, josta tageja pystyy lisäämään, muokkaamaan ja poistamaan, kuten haluttiin. Sivujen ulkoasu on karu, mutta kuten projektissa aluksi määriteltiin, lopullinen ulkoasu on oma projektinsa myöhemmin.

7 PROJEKTIN ARVIOINTI JA PÄÄTTÄMINEN

7.1 Arviointi projektin vaiheista ja resurssien hallinnasta

Projekti oli selvästi jaettu eri vaiheisiin. Ensin oli määrittely. Määrittelyssä käytin mielestäni liikaa aikaa ja palasin määrittelemään asioita uudestaan myöhemmin, kun tajusin että asioita voisi tehdä paremmin toisella tavalla. Sen jälkeen oli toteutus, joka alkoi RFID-lukijaan perehtymisellä. Siinä meni oma aikansa ja kun sain käsityksen lukijan toiminnasta, pääsin aloittamaan seuraavan vaiheen eli ohjelmoinnin. Ohjelmointi vaihe sujui mielestäni kaikista parhaiten. Ei tullut vastaan montaa ongelmaa, jonka ratkaisemiseen meni aikaa turhan paljon. Mielestäni projektin vaiheet menivät ihan hyvin, mutta resurssien hallinnassa olisin voinut olla parempi ja käyttää aikaa heti aluksi enemmän suunnitteluun.

7.2 Aikataulu

Projektilla ei ollut mitään kiinteitä aikatauluja ja sellaista olisi ollut vaikea määritellä ohjelmistoprojektiin. Tavoitteena oli saada projekti valmiiksi ennen kesää. Tässä aikataulussa onnistuttiin pysymään, mutta sen takia vaa'an toiminnallisuus jäi vajaaksi valmiissa projektissa. Muuten projektin eteneminen sujui suhteellisen nopeasti ja tilaajan kanssa käydyissä keskusteluissa asiaa käsitelimme. Näitä oli aluksi harvemmin ja lopuksi viikoittain.

7.3 Projektin tuloksen hyödyntäminen ja arviointi

Projektin tuloksen hyödyntämistä on vaikea arvioida, koska en päässyt asentamaan ja koittamaan projektia tilaajan tiloissa lopuksi covid-19 tilanteen vuoksi. Kotona tehdyt testit osoittivat ohjelman niiltä osin, kun se oli valmis toimivaksi.

Mielestäni pääsin aika hyvin tavoitteisiin. Myöhemmin ajateltuna muutama asiaa olisi voinut tehdä toisella tavalla, mutta lopputuloksen kannalta niillä ei ollut merkitystä. Esimerkiksi sen sijaan, että tein tyhjästä kaiken ensi kerralla olisin valinnut, jonkun sovelluskehiksen minkä päälle olisin tehnyt oman toteutukseni. Näin olisin säästänyt

aikaa jonkin verran. Toinen mikä tuli myöhemmin mieleen oli, että ehkä olio-ohjelmoinnilla PHP:stä olisi saanut myös helpommin käytettävän ja laajennettavan.

Mitään näin laajaa ohjelmistoprojektia en ollut ennen tehnyt. Opein tässä projektissa paljon uutta ohjelmoinnista virheiden kautta mitä matkalla tuli tehtyä ja uusien asioiden tekemisen yhteydessä. En ollut ennen ollut socket ohjelmointia tehnyt ja nyt sekin tuli tutuksi Pythonilla. PHP olin vain käyttänyt vähän aikaisemmin ja en oikeastaan sen kanssa ollut tietokantoihin koskenut, mutta tämän projektin myötä ymmärrän nyt PHP:n tietokantaominaisuuksista. Tietokannat sen sijaan olivat valmiiksi tuttuja, ja niistä en voi sanoa oppineeni mitään uutta. Lähinnä projekti toimi muistin virkistämiseksi niiden kanssa.

Lopuksi voisin todeta, että pääsin mielestäni lähes kaikkiin tavoitteisiin mitä projektissa oli. Vaa'an myöhäisen saapumisen takia Python-ohjelma jäi siihen kesken ja se oli isoin epäonnistuminen projektissa. Projekti muuten suhteellisen nopeasti, mutta kirjoitusprosessissani on edellään parannettavaa. Projektin rajaus oli onnistunut mielestäni.

Jatkokehitykseksi projektiin näkisin kirjautumisen, olio-ohjelmoinnin käyttämisen ja Python-ohjelman viimeistelyn vaa'alle. Lisää jatkokehitysideoita varmasti olisi tullut, jos projektia olisi päässyt testaamaan paremmin paikan päällä.

LÄHTEET

3DJAKE www-sivut. 2020. Viitattu 24.5.2020. <https://www.3djake.fi/filamentit>

Angela Bradley 2019. 'Why use PHP?'. Thoughtco. Viitattu 12.5.2020
<https://www.thoughtco.com/why-use-php-2694006>

atlasRFIDstore www-sivut 2020. Viitattu 6.5.2020. <https://www.atlasrfid-store.com/rfid-beginners-guide/>

Code institute www-sivut. 2020. Viitattu 19.5.2020. <https://codeinstitute.net/blog/what-is-python-used-for/>

David 2018. 'Why use PHP in 2019?'. Thoughtful Code. 12.6.2018. Viitattu 12.5.2020 <https://www.thoughtfulcode.com/why-use-php/>

Debian www-sivut. 2020. Viitattu 19.5.2020. <https://www.debian.org/intro/about>

Distrowatch www-sivut. 2020. Viitattu 19.5.2020. <https://distrowatch.com/>

EPC-RFID INFO www-sivut. 2020. Viitattu 22.5.2020 <https://www.epc-rfid.info/>

freeCodeCamp www-sivut. 2020. Viitattu 19.5.2020. <https://www.freecodecamp.org/news/the-best-php-examples/>

Full Stack Python www-sivut. 2020. Viitattu 12.5.2020. <https://www.fullstackpython.com/why-use-python.html>

Hostinger www-sivut. 2020. Viitattu 15.2.2020. <https://www.hostinger.com/tutorials/what-is-mysql>

IMPINJ Support www-sivut. 2020. Viitattu 24.5.2020. <https://support.impinj.com/hc/en-us/articles/202755278-Speedway-Connect>

Margaret Rouse 2019. RFID (radio frequency identification). Viitattu 10.5.2020.
<https://internetofthingsagenda.techtarget.com/definition/RFID-radio-frequency-identification>

MongoDB, Inc www-sivut. 2020. Viitattu 12.5.2020. <https://www.mongodb.com/nosql-explained>

MySQL www-sivut. 2019. Viitattu 12.5.2020. <https://www.mysql.com/news-and-events/web-seminars/top-reasons-to-use-mysql-2019/>

MySQL www-sivut. 2020. Viitattu 11.5.2020. <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>

PHP: The "Right" Way. 2016. Sturgeon, P & Lockhart J. Viitattu 6.5.2020.
<https://leanpub.com/phptherightway>

Pontikis www-sivut. 2020. Viitattu 19.5.2020. <https://www.pontikis.net/blog/five-reasons-to-use-debian-as-a-server>

Python Software Foundation www-sivut. Viitattu 12.5.2020. <https://docs.python.org/3/howto/sockets.html>

Python Software Foundation www-sivut. Viitattu 12.5.2020. python.org/doc/esays/blurb/

Python Software Foundation www-sivut. Viitattu 21.5.2020. <https://docs.python.org/3/faq/general.html>

Shahid Mansuri 2019. 'Complete Guide to RFID: Benefits, Applications, and Challenges' Dzone. 28.2.2019. Viitattu 19.5.2020 <https://dzone.com/articles/complete-guide-to-rfid-its-application-to-supply-c>

Sloan 2018. 'What's the difference between visual studio and vs code?'. DEV. 11.9.2018. Viitattu 11.5.2020 <https://dev.to/sloan/whats-the-difference-between-visual-studio-and-vs-code-424>

The Hello World Program www-sivut. 2020. Viitattu 12.5.2020. <https://thehello-worldprogram.com/python/why-python-should-be-the-first-programming-language-you-learn/>

The PHP Group www-sivut. 2020. Viitattu 10.5.2020. <https://www.php.net/manual/en/history.php>

The PHP Group www-sivut. 2020. Viitattu 10.5.2020. <https://www.php.net/>

The PHP Group www-sivut. 2020. Viitattu 12.5.2020 <https://www.php.net/manual/en/intro-whatcando.php>

The PHP Group www-sivut. 2020. Viitattu 6.5.2020 <https://www.php.net/manual/en/faq.general.php>

Toptunniste www-sivut. 2020. Viitattu 21.5.2020. <https://toptunniste.fi/rfid-nfc-tekologia/#pofo-1590083505-1692933557-1>

Visual Studio Code www-sivut. 2020. Viitattu 11.5.2020. <https://code.visualstudio.com/docs/supporting/faq>

Visual Studio Code www-sivut. 2020. Viitattu 11.5.2020. <https://code.visualstudio.com/docs/editor/whyvscode>

W3schools www-sivut. 2020. Viitattu 11.5.2020. https://www.w3schools.com/sql/sql_intro.asp

W3schools www-sivut. 2020. Viitattu 12.5.2020. <https://www.w3schools.com/php/default.asp>

W3schools www-sivut. 2020. Viitattu 21.5.2020. https://www.w3schools.com/python/python_intro.asp