

Joni Pääkkö

TELEAMP-VAHVISTIMEN KÄYTTÖLIITTYMÄ

TELEAMP-VAHVISTIMEN KÄYTTÖLIITTYMÄ

Joni Pääkkö
Opinnäytetyö
Kevät 2015
Tietotekniikan koulutusohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan koulutusohjelma, ohjelmistokehitys

Tekijä: Joni Pääkkö
Opinnäytetyön nimi: TeleAmp-vahvistimen käyttöliittymä
Työn ohjaaja: Kari Jyrkkä
Työn tilaaja: TeleSilmukka Oy
Työn valmistumislukukausi ja -vuosi: Kevät 2015 Sivumäärä: 28 + 3 liitettä

Työn aiheena oli tehdä uusi web-pohjainen käyttöliittymä olemassa olevan Windows-pohjaisen käyttöliittymän korvaajaksi. Tavoitteet työssä oli saada toimiva kokonaisuus, jossa loppukäyttäjä ohjaa käyttöliittymällä koordinaattoria.

Työ toteutettiin pääsääntöisesti PHP- ja HTML-kielillä. Apuna työssä oli käytössä valmis web front end -kirjasto josta löytyi valmiit muotoilut ulkoasun kehittämiseen. PHP:llä toteutettiin käyttöliittymän looginen ja toiminnallinen osuus. Kokonaisuuteen sisällytettiin myös MySQL-relaatiotietokanta tiedon tallettamista varten. Projektin läpiviennissä käytettiin mahdollisimman tehokkaasti vapaan lähdekoodin ohjelmistoja.

Projektissa ei saavutettu aivan kaikkea vaatimuslistan ominaisuuksia.

Asiasanat: HTML, PHP, web front end, ZigBee, käyttöliittymät, nettisivut

ABSTRACT

Oulu University of Applied Sciences
Information Technology, Software Development

Author: Joni Pääkkö

Title of thesis: TeleAmp amplifier user interface

Supervisor: Kari Jyrkkä

Customer: TeleSilmukka Oy

Term and year when the thesis was submitted: Spring 2015 Pages: 28 + 3 appendices

The subject of the project was develop new web based user interface to replace old Windows OS software user interface. Aim of the work was to get a functional entity where end-user interacts user interface to control coordinator.

Project was mainly developed using PHP and HTML language. There was also used a web front end package with prewritten cascading style sheets. User interface logical and functional part was written in PHP. MySQL database was included in whole entity to save data. Open source was effectively used during the project

All the requirements were not achieved from a list of features. This was due among many things like the challenge of the project and the lack of know-how.

Keywords: HTML, PHP, web front end, ZigBee, user interfaces, webpages

ALKUSANAT

Tämän insinööriyön mahdollisti TeleSilmukka Oy. Työn ohjaavana opettaja on toiminut Kari Jyrkkä Oulun ammattikorkeakoulusta ja työn valvojana on toiminut TeleSilmukka Oy:ltä Mikko Haho ja Ville Kivelä.

Haluan kiittää tilaajaa ja tilaajan yhteys henkilöä Ville Kivelää, opinnäytetyön ohjaajaa Kari Jyrkkää ja muita henkilöitä, jotka ovat auttaneet prosessin aikana.

Oulussa 3.6.2015

Joni Pääkkö

SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
ALKUSANAT	5
SISÄLLYS	6
SANASTO	7
1 JOHDANTO	9
2 JÄRJESTELMÄARKKITEHTUURI	10
2.1 Digi ConnectPort X4/IA	11
2.2 Raspberry Pi	12
2.3 MySQL tietokanta	12
3 TOIMINNALLINEN MÄÄRITTELY JA SUUNNITTELU	13
3.1 Vaatimukset	13
3.2 Käyttöliittymä	13
3.3 Käyttöliittymän ja ConnecPortin välinen yhteys	14
4 JÄRJESTELMÄARKKITEHTUURI	15
5 KÄYTETYT OHJELMISTOT	17
5.1 HTML	17
5.2 CSS	17
5.3 PHP	18
5.4 HTML-Kickstart	18
5.5 MySQL	19
5.6 Vagrant	19
6 TOTEUTUKSEN TULOS	21
7 YHTEENVETO	25
LÄHTEET	26
LIITTEET	VIRHE. KIRJANMERKKIÄ EI OLE MÄÄRITETTY.
LIITE 1. Vaatimusmäärittelyt	
LIITE 2. Esimerkki yhteyden muodostamisesta tietokantaan	
LIITE 3. Datat hakeminen ja tulostaminen tietokannasta	

SANASTO

Apache	Avoimeen lähdekoodiin perustuva HTTP-palvelinohjelmisto.
API	Application programming interface
Bower	Ohjelmistopakettinhallinta järjestelmä.
Client	Asiakasohjelma.
Composer	Sovellustason pakettinhallintaohjelma PHP-kielelle.
CSS	Cascading Style Sheets, internetsivujen tyyliarkkitiedosto.
cURL	Olio-ohjelmointikieli vuorovaikutteiseen web-sovelluksien luomiseen.
Digi ESP	Integroitu kehitysympäristö Digin tuotteille.
GD	Avoimen lähdekoodin kuvanmuokkaustyökalu joka toimii Linux-palvelimessa.
Git	Hajautettu versiohallintaohjelmisto.
Grunt	JavaScriptin tehtävien suorittamiseen tarvittava ohjelmisto.
Gulp	Node.js-pohjainen tehtävien suorittamiseen tarvittava ohjelmisto.
HTML	Hypertext Markup Language, internetsivujen ohjelmointikieli.
IDE	Integrated Development Environment, ohjelmointiympäristö.
Imagick	Grafikan käsittelyyn tarkoitettu ohjelma, jota suoritetaan komentoriviltä.

Laravel Installer	Asennusohjelmisto Laravel-nimiselle web back end -sovel- lukselle.
Mcrypt	Tiedostojen salausohjelmisto.
Memcached	Hajautettu välimuistiohjelma.
MySQL	Relaatiotietokanta
NPM	Paketin hallintaohjelma JavaScriptille.
PHP	PHP: Hypertext Preprocessor, palvelinpuolella käytetty oh- jelmointikieli.
RPi	Raspberry Pi, yhden mikropiirin tietokone.
Ruby	Dynaamisesti tyyppittävä oliopohjainen ohjelmointikieli.
Screen	Ohjelma jota käytetään multipleksaamaan useita virtuaali- konsoleja yhteen konsoliin.
TCP	Transmission Contron Protocol, tietoliikenneprotokolla.
Telnet	Yhteysprotokolla pääteyhteyksiin Internetin ylitse.
vim	Avoimeen lähdekoodiin perustuva tekstieditori.
Web Front End	Osa-alue verkkosivujen kehittämisessä jonka loppukäyt- tjä näkee ja jota loppukäyttjä käyttää.
Yeoman	Avoimen lähdekoodin kehitystyökalu web-sovelluksille.

1 JOHDANTO

Opinnäytetyönä luotiin käyttöliittymä TeleAmp-vahvistimille. Tavoitteena oli luoda responsiivinen käyttöliittymä.

Työn tilasi TeleSilmukka Oy. Yritys on keskittynyt äänentoiston ja induktiosilmukoiden kehittämiseen. Yrityksen tavoitteena on kehittää langatonta vahvistinyksikköä, joka kaiuttimeen liitettynä vastaanottaa ja toistaa FM-radiosignaalia. Laitetta voidaan käyttää äänentoistoon esimerkiksi ulkoilmatapahtumissa, kampuksilla, kasarmeilla tai luonnononnettomuuksien varoitusjärjestelmissä.

Vahvistinyksikkö saa käyttöjännitteen aurinkopaneelilla ladattavasta akusta, mikä tekee siitä ekologisen. Vahvistimella on korkea hyötysuhde eikä se tarvitse langallista tehonsyöttöä. Yksikkö sisältää mikrokontrollerin, jolla ohjataan vahvistimiin asetettavia parametreja. Ohjaussignaalit lähetetään mikrokontrollerille XBee-radiopiirin kautta.

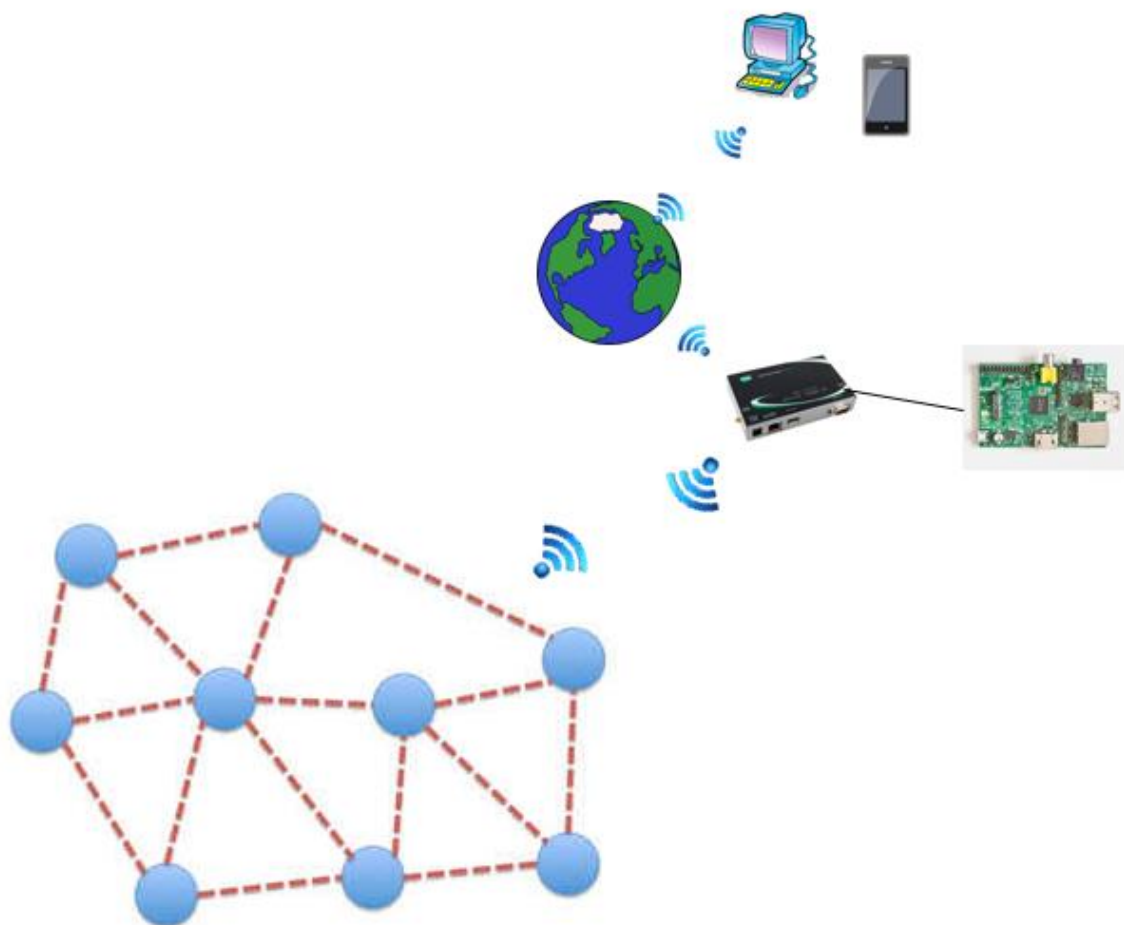
Opinnäytetyön alussa vahvistinyksiköllä oli Windows-käyttöjärjestelmälle toteutettu käyttöliittymä. Tarkoitus oli luoda alustariippumaton web-käyttöliittymä, jolla vahvistinyksikköä voitaisiin operoida. Käyttöliittymän toimintaympäristö ja koodaus tapahtuisi HTML- ja PHP koodilla.

Työn tavoitteena oli tutkia ja asentaa toimiva Python-palvelin Digi ConnectPort X4 IA -reitittimelle ja todentaa, että web-palvelin oli toteutettavissa laitteella. Tällä välttyttäisiin käyttämästä esim. Raspberry Pi -mikrotietokonetta. Projektissa tutkittiin myös MySQL-tietokannan tarpeellisuutta. Tavoitteena oli saada toteutettua erittäin tärkeät ominaisuudet ja sen jälkeen optionaaliset. Ominaisuudet löytyvät liitteestä 1.

2 JÄRJESTELMÄARKKITEHTUURI

Yrityksellä oli tiedonsiirtoa varten olemassa Digi ConnectPort X4 IA -reititin, johon alkuperäisen suunnitelman mukaan oli tarkoitus asentaa Pythonilla toimiva palvelin toiminnollisuuksineen. Koska ConnectPortin Python-versio on 2.4, päädyttiin käyttämään palvelimena Raspberry Pi -mikrotietokonetta. ConnectPortin Python-versiota ei voi nostaa laiteohjelmistopäivityksellä (1).

Projektin kokonaisvaltainen idea selittyy kuvan 1 avulla.



KUVA 1. Laitteistokuva

Loppukäyttäjä hallinnoi tuotetta tietokoneelta tai tabletti- tai mobiililaitteelta. ConnectPort toimii reitittimenä, johon voidaan ottaa yhteyttä joko langattomasti mobiiliverkon 3G-yhteydellä tai jos laite on kytketty langattomaan verkkoon. Yh-

teydenotto on mahdollista myös langallisesti lähiverkossa. RPi on liitetty ConnectPortiin ja RPi:llä suoritetaan web-palvelinohjelmaa ja kaikki käyttöliittymään liittyvä ohjelmisto sijaitsee RPi:ssä. Käyttöliittymäohjelmisto käyttää ConnectPortin ZigBee-verkkoa kommunikoidakseen langattomasti ZigBee-verkossa olevien laitteiden kanssa.

2.1 Digi ConnectPort X4/IA

ConnectPort X4 IA on ZigBee to Ethernet & Cellular -reititin, jolla tiedonsiirto onnistuu langattomasti radiomoduuleille. Malli tukee matkapuhelinverkkoa, joka takaa laitteen liitettävyyden Internetiin tilanteissa, joissa ei ole käytössä kiinteää verkkoa. (2.) Laite on yhteensopiva Python-koodin kanssa ja näin ollen antaa valmiuden toimia myös palvelimena kevyellä Python-pohjaisella koodilla. Laitteen tukema Python-versio on 2.4 ja suurin osa Python-web-palvelinohjelmistoista toimivat versiolla 2.7 (3). Tästä syystä palvelinidea hylättiin tämän laitteen kohdalla.



KUVA 2. Digi ConnectPort X4/IA (4.)

2.2 Raspberry Pi

Raspberry Pi, lyhennettynä RPi, on uuden sukupolven mikrotietokone, jonka pinta-ala ei ole juuri pankkikorttia isompi. RPi:n suoritusnopeus on 700 megahertsiä ja muistia laitteessa on 256 megatavua. Mikrotietokoneelle on ladattavissa ilmaisia käyttöjärjestelmiä laitteen kotisivuilta. (5.)



KUVA 3. Raspberry Pi (6)

2.3 MySQL-tietokanta

Osana työtä selvitettiin MySQL-tietokannan tarpeellisuutta projektiin. Tiedon keruu on tämän projektin yhteydessä keskeinen asia. Projektikonaisuus tuottaa tietoa, jota haetaan ja lähetetään edestakaisin, jolloin on tärkeää tallentaa tietoa tietokantaan muistiin. Jotta syntyvää ja käytettävää tietoa olisi mahdollista tallentaa kohtuullisilla kustannuksilla, tarvitaan tietokantoja.

Selvitystöiden päätyttyä tultiin siihen tulokseen, että tietokanta on olennainen osa projektia, mihin pystyy helposti tallentamaan dataa ja sieltä voi helposti hakea dataa sekä tulostaa se käyttäjälle.

3 TOIMINNALLINEN MÄÄRITTELY JA SUUNNITTELU

Projektin alkaessa käytiin tilaajan kanssa keskustelu, jossa selvitettiin millaisia toiminnollisuuksia ja vaatimuksia laitteen toimintaan tulee sisällyttää.

3.1 Vaatimukset

Tuotteen vaatimuslista on liitteessä 1. Kaikkia tuotteen vaatimuksia ei ole ehditty tekemään projektin haasteellisuuden vuoksi, mutta olennaisimmat on tehty.

Sovellukselle asetettiin vaatimuksia, joista enemmistö luokiteltiin erittäin tärkeiksi ominaisuuksiksi. Vaatimuksissa tärkeimpänä oli nettisivun responsiivisuus eli skaalautuvuus tietokoneen näytöltä mobiililaitteelle. Sivuston täytyi olla myös käyttäjäystävällinen käyttäessä mobiili- ja tabletilaitetta.

Vahvistimelle lähetettäviä tärkeimpiä vaatimuksia olivat sen neljäportainen vahvistusteho, radiokanavan valinta, laitteen mykistäminen, vahvistimen nimeäminen, laitteiden ryhmittäminen ja hallinta ryhmäkokonaisuutena, virransäästötila ja liikkumistila. Optionaalisena asetuksena annettiin radiopiirin lähtötason säätö ja MySQL-tietokannan toiminta.

Käyttöliittymänäkymän vaatimukset olivat saada akun jännitetaso näkyvin ja GPS-sijainti. Optionaalisena oli saada GPS-sijaintinäkymä offline-kartalla.

3.2 Käyttöliittymä

Käyttöliittymästä ei ollut valmista suunnitelmaa, vaan minulle annettiin valtuudet luoda siitä omannäköinen. Kysymyksen jälkeen käyttöliittymästä alkoi hahmottua visuaalinen kuva. Hyvänä apuna käytettiin Windows-pohjaisen sovelluksen ulkonäköä. Windows-sovelluksen ulkoasu auttoi ymmärtämään, mitä kaikkea käyttöliittymässä on löydettävä ja mitä kaikkea sillä voidaan ohjata.

3.3 Käyttöliittymän ja ConnectPortin välinen yhteys

Käyttöliittymän ja ConnectPortin välinen liikennöinti tullaan toteuttamaan myöhemmin TCP-verkkoliikenteenä. Tämä osa-alue jäi toteuttamatta opinnäytetyöhön ajanpuutteen vuoksi.

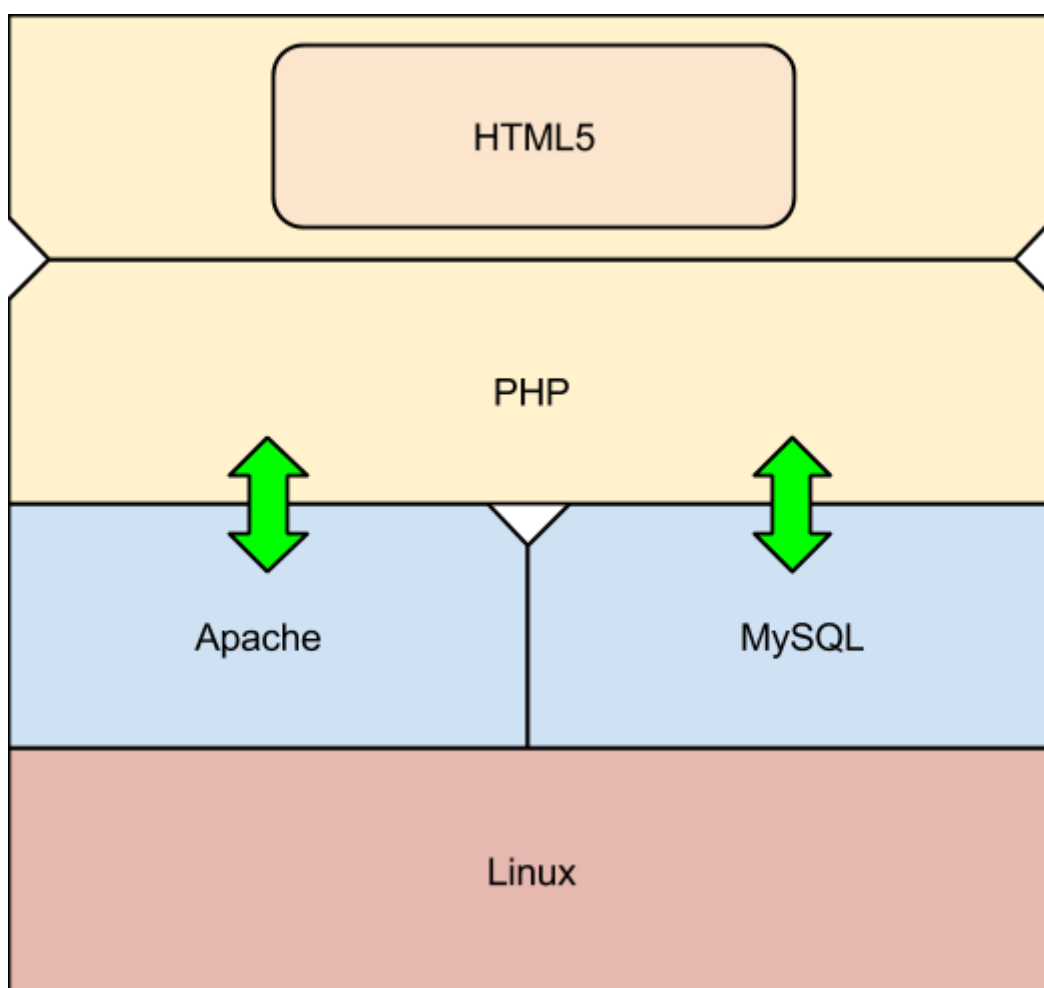
Teoriassa yhteys muodostetaan siten, että ConnectPortille kirjoitetaan Python-koodi (7), joka kuuntelee ennalta määrättyä TCP-porttia, esimerkiksi portti numero 8888. Koodi kirjoitetaan Digi ESP IDE:llä, jolla ohjelma kirjoitetaan suoraan ConnectPortiin ja se käynnistetään automaattisesti latauksen jälkeen. Digi ESP:ssä avautuu Telnet-yhteys, jota käytetään testausvaiheessa.

Vastaavasti PHP-koodiin kirjoitetaan TCP-client-ohjelma, joka ottaa yhteyden ConnectPortin IP-osoitteeseen ja lähettää datan. PHP:lle löytyy ohje, miten sokettiyhteys luodaan (8).

Teoriassa ideana oli lähettää PHP-koodilla lyhyt viesti, esimerkiksi *Hello World from PHP!* ja ConnectPort olisi vastannut tähän omassa konsolissaan *Greeting from 192.168.10.1*, joka olisi ollut ConnectPortin käyttämä IP-osoite. Tällä todetaisiin yhteyden toimivuus, josta olisi lähdetty kehittämään verkkoliikenne protokollaa projektin tarvitsemalle tasolle.

4 JÄRJESTELMÄARKKITEHTUURI

Alustana toimii vapaan jakelun käyttöliittymä Linux, johon asennettiin www-palvelin Apache ja tietokantapalvelin MySQL. Kuvasta 4 voi havainnollistaa järjestelmän rakenteen. PHP toimii palvelinpuolen skriptinä, millä voidaan kontrolloida sitä, mitä tietokantaan kirjoitetaan ja mitä sieltä haetaan. HTML-koodi on upotettuna PHP-sivuille.



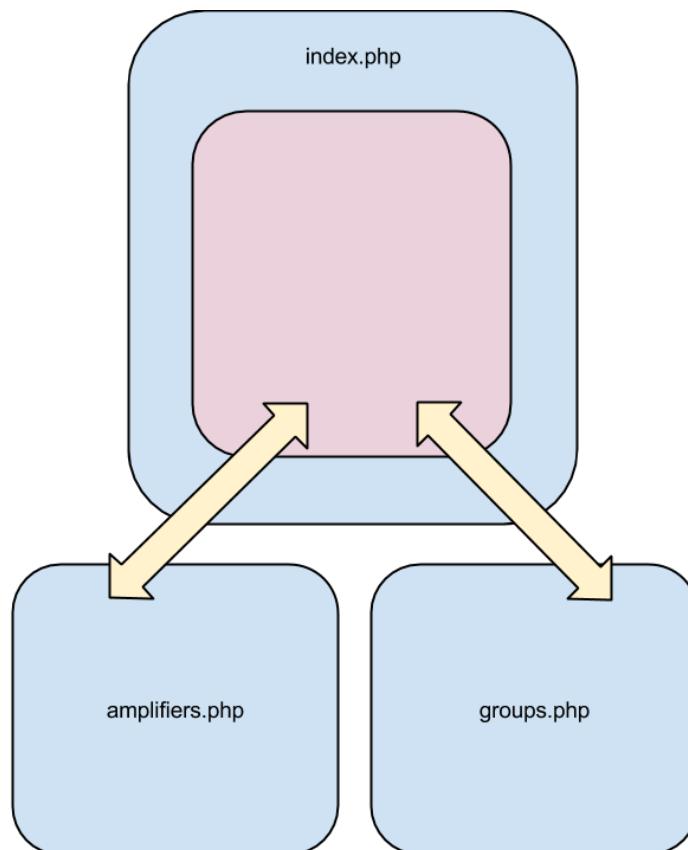
KUVA 4. Nettisivun rakenne

Kuvassa 5 näkyy käyttöliittymän perusrunko, joka muodostui *index.php*-sivun sisään, jossa sivun painikkeille määriteltiin *include*-komento. Include sisältää toisen näytettävän sivun nimen, esim. *include 'amplifiers.php'*; hakee välilehden sisällön *amplifiers.php*-tiedostosta. Tällä saavutettiin se etu, että jokaisen sivun

muotoilua ei tarvinnut luoda uudelleen, vaan muotoilu saadaan *index.php*-sivun mukaan.

Välilehden PHP-sivujen ensimmäiset ohjelmakoodit ovat MySQL-kyselyjä, joilla haetaan dataa tietokannasta. Tämä on tärkeä suorittaa ensimmäisenä siksi, että saadaan data valmiiksi odottamaan tulostamista.

Ennen MySQL-kyselyjä on otettava yhteys tietokantaan liitteen 2 esimerkin mukaisella tavalla. MySQL-kyselyjen jälkeen rakentuu itse sivu apuna käyttäen HTML-Kickstartin valmiita tyyliä. Liitteessä 3 on esimerkki datan noutamisesta tietokannasta ja sen tulostamisesta näytölle.



KUVA 5. Käyttöliittymän sivun rakenne

5 KÄYTETYT OHJELMISTOT

Työn ohjelmointikielenä käytettiin nettisivuille tyypillistä HTML-kieltä ja sen tyyliohjeistamiseen valmiita tyyliarkkeja eli CSS-tiedostoja. Sivun luomisessa auttoi valmis web front end -kehitysalusta, jossa tyyli- ja ohjeet tyylien käyttöön on valmiina.

5.1 HTML

HTML on lyhenne sanoista Hypertext Markup Language. HTML on avoimesti standardoitu kieli, jolla voidaan kuvata hyperlinkkejä. Kielellä voidaan merkitä tekstin rakennetta eli sitä, mikä on otsikko, alaotsikko, leipätekstiä jne.

HTML5-kielellä tarkoitetaan HTML-spesifikaation viimeisintä versiota. Nykyisellään HTML-kieli on elävä standardi. Elävällä standardilla tarkoitetaan sitä, että sinne lisätään ominaisuuksia, jotka vakiintuvat ajan myötä ja jo olemassa olevia ominaisuuksia tarkennetaan (9.). Elävä standardi on luettavissa WhatWG:n nettisivuilta (10).

5.2 CSS

CSS on lyhenne sanoista Cascading Style Sheets eli porrastetut tyyliarkit. CSS-tiedostolla voidaan ennalta määrittää www-dokumenteille tyyliohjeita (11). Tyyliarkkeja käytetään kuvaamaan ja muotoilemaan nettisivuja ja käyttöliittymiä, mitkä on kirjoitettu HTML- ja XHTML-kielellä.

Esimerkki CSS:n perussyntaksi valitsimesta, ominaisuudesta ja arvosta. Esim: `tbl_bg { background-color: black; }`, jossa `tbl_bg` on luokan nimi, kun tätä halutaan käyttää, sitä kutsutaan komennolla `class="tbl_bg"`. Aaltosulkeiden sisällä oleva ominaisuus `background-color` on HTML-kielessä taustaväriin syntaksi. Värisyntaksille annetaan arvoksi musta.

Edellä oleva esimerkki koodissa käytettynä `class="tbl_bg"` muuttaa sivun taustaväriin mustaksi.

CSS-tiedostossa voi olla useita tyylimääräyksiä peräkkäin eroteltuna aaltosuluilla. Aaltosulkujen sisällä voi olla eri ominaisuuksia ja arvoja eroteltuna toisistaan puolipisteellä. (11.)

5.3 PHP

PHP:n alkuperäinen lyhenne on sanoista Personal Home Page tools. Myöhemmin nimi on muuttunut PHP: Hypertext Preprocessor. PHP on laajalti käytössä oleva palvelinpuolen ohjelmointikieli, joka on kehitetty erityisesti web-koodaukseen. (12.)

HTML-sivut on tyypillisesti sijoitettu PHP-tiedoston sisälle. PHP-koodi erotellaan aloitus- ja lopetusmerkinnöillä `<?php .. ?>`. (13.)

PHP:n keksi vuonna 1994 tanskalais-grönlantilainen Rasmus Lerdorf, joka kirjoitti pienen kokoelman C-kielisiä CGI-skriptejä nimellä Personal Homepage Tools. Lerdorf julkaisi PHP:n 8.6.1995 nimellä PHP/FI (Personal Homepage Tools / Forms Interpreter). PHP:n viimeisin vakaa versio on 5.6.7, julkaistu 19.3.2015. (13.)

Esimerkki koodia *helloworld.php*

```
<HTML>
<HEAD>
<TITLE>PHP HELLOWORLD</TITLE>
<BODY>
<?PHP ECHO "<P>HELLO WORLD!</P>; ?>
</BODY>
</HTML>
```

5.4 HTML-Kickstart

HTML-Kickstart on kevyt web front end -koonpano, joka sisältää HTML-, CSS- ja jQuery-tiedostoja, jotka auttavat luomaan joustavan ja responsiivisen nettisivun hetkessä. Käyttämällä valmista web front end -pakettia säästetään koodin kirjoittamisessa useita tunteja, koska ei tarvinnut kirjoittaa omia CSS-tiedostoja (14.). Työn nopeuttamiseksi HTML-Kickstartin kotisivuilla löytyvästä ohjeesta näkee, miten erilaiset komponentit toimivat ja miten niitä käytetään (15).

5.5 MySQL

MySQL on relaatiotietokantaohjelmisto, johon on erittäin helppo tallettaa tietoja. MySQL:n kehitti vuonna 1995 Michael Widenius yhdessä ystävänsä David Axmarkin kanssa. MySQL on laajalti käytössä oleva ohjelmisto, jonka tunnetuimpia käyttäjiä, asiakkaita on mm. Facebook, Google ja Wikipedia. MySQL on saanut nimensä löytäjänsä Michael Wideniuksen tyttären Myn mukaan. Loppuosa, SQL, tulee sanoista Structured Query Language. (16.)

MySQL on hyvin suosittu relaatiotietokanta web-palveluiden tietokantana. Tietokannan päälle rakennettava ohjelmalogiikka tehdään usein PHP-, Python- tai Perl-ohjelmointikielellä. Myös muilla kielillä on mahdollista tehdä MySQL-ohjelmalogiikkaa, koska myös valmis rajapinta löytyy mm. C:lle, C++:lle, C#:lle, Javalle ja Rubyille. Tässä projektissa käytettiin PHP-ohjelmointikieltä. MySQL-tietokannan sisällön, taulujen ja datan hallintaan on olemassa muutamia eri vaihtoehtoja, joista yksi on komentorivi ja toinen on graafinen käyttöliittymä. Projektissa käytettiin jo entuudestaan tuttua MySQL-Workbench graafista ohjelmistoa tietokannan sisällön hallintaan. (17.)

Tietokanta koostuu *tauluista (table)* ja taulujen välisistä *yhteyksistä (relation)*. Taulut koostuvat *tietueista (record)* ja *kentistä (field)*. Kenttä on tietokannan pienin osa, sillä jokaista tietokantaan tallennettavaa tietoa varten tarvitaan kenttä ja kaikki kentät yhdessä muodostavat tietueen.

MySQL-Workbench on visuaalinen tietokannan suunnittelutyökalu, joka yhdistää SQL:n kehittämisen, hallinnan, tietokannan suunnittelun, luomisen ja ylläpidon yhdeksi kehitysympäristöksi MySQL-tietokanta järjestelmään.(18.)

5.6 Vagrant

Vagrant on ohjelmisto, jolla voidaan luoda virtuaalinen kehitysympäristö apuna käyttäen esimerkiksi VirtualBoxia tai VMwarea (19). Projektissa käytettiin virtuaalista kehitysympäristöä ja näin ollen mahdollistettiin offline-työnskentely tietokoneella. Projektissa ei tarvittu fyysistä palvelinta, jossa testata verkon ylitse ohjelmiston toimivuutta, vaan palvelin virtualisoitiin tietokoneelle. Samalla kun ohjelmistoa kehitettiin, pystyttiin testaamaan uudet toiminnollisuudet ja tekemään

niihin tarvittavat muutokset. Ennen kaikkea Vagrantin sandbox tarjoaa sen, että voidaan luoda kohdeympäristöä vastaava kehitysympäristö.

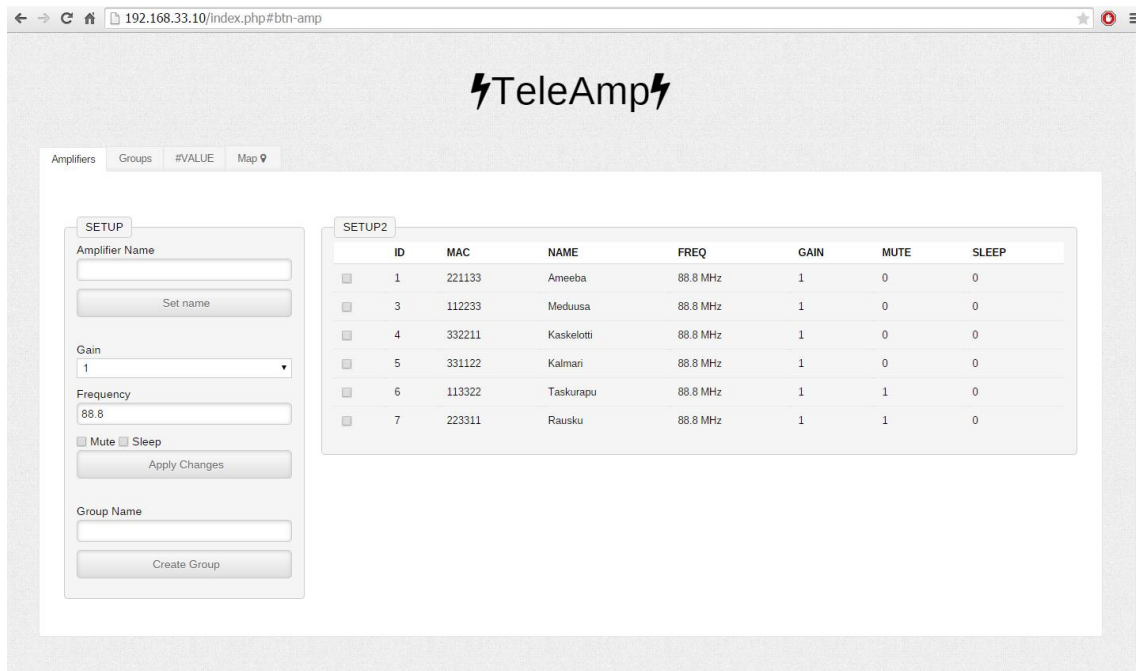
Virtuaalisoidun kehitysympäristön etuna on se, että sen voi sekoittaa ja tuhota täysin menettämättä mitään tärkeää ja jo olemassa olevaa fyysistä konfiguraatiota. Pilalle menneen virtuaalisoidun kehitysympäristön voi poistaa ja luoda aina uuden ja näin saada taas puhtaan alustan, mihin lähteä kehittämään.

Scotch Box antaa valmiin Linux serverikonfiguraation, missä on esiasennettuna tarvittavat työkalut ja ohjelmistot.

Scotch Boxin sisältämiin palvelinohjelmistoihin kuuluu valmiiksi Apache, vim, MySQL, PHP 5.5, Ruby, Git, Screen, Composer, Laravel Installer, cURL, GD ja Imagick, Mcrypt ja Memcached. Frontend ominaisuuskiin kuuluu valmiiksi NPM, Grunt, Bower, Yeoman ja Gulp.(20.)

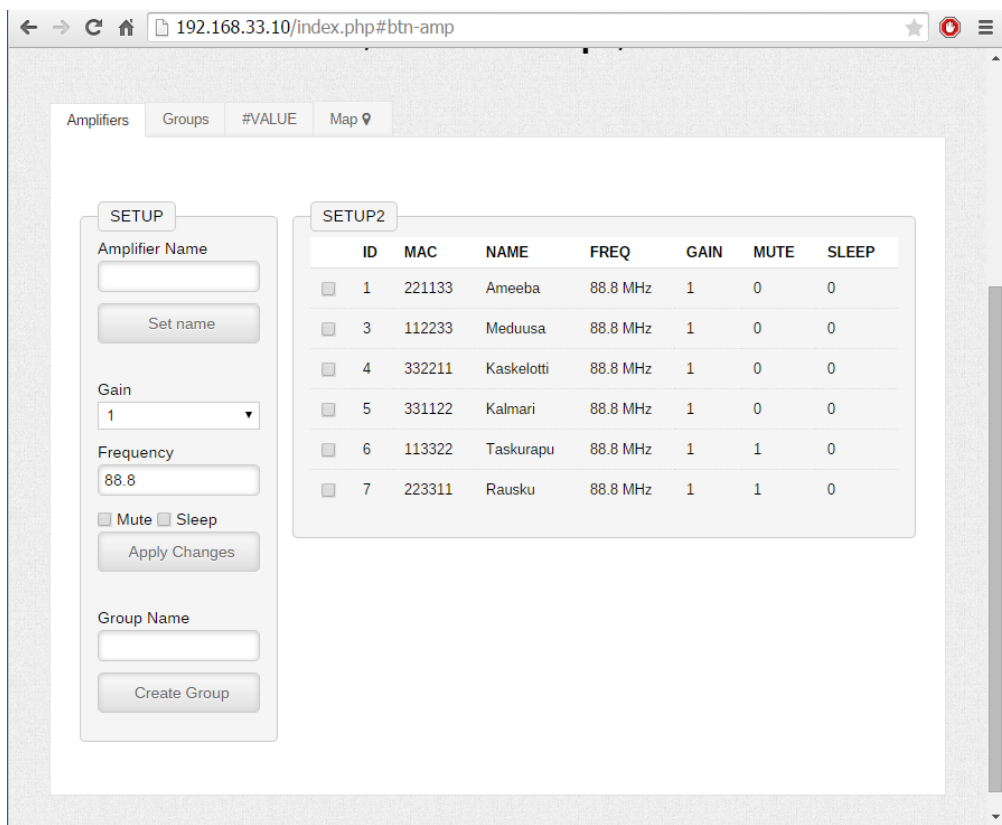
6 TOTEUTUKSEN TULOS

Käyttöliittymästä suunniteltiin kokonaan responsiivinen. Se mukautuu käytettävän laitteen näytön koon mukaan ja säilyttää käytettävyytensä. Kuvassa 6 käyttöliittymä on tietokoneen näytöllä.



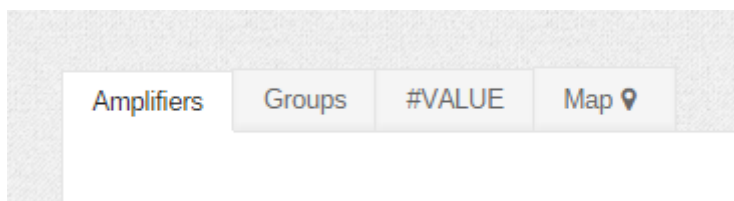
KUVA 6. Käyttöliittymä

Kuvassa 7 näkyy, miten sivu mukautuu, kun sivua pienennetään, mutta sivun tyyliasu säilyttää muodon.



KUVA 7. Käyttöliittymä skaalattuna pienempään

Index.php-tiedostoon luotiin välilehtipalkki, kuten kuvassa 8. Välilehtien avulla käyttöliittymän selaaminen nopeutuu, eikä sivuja tarvitse aina ladata uudestaan, kun vaihdetaan toiselle välilehdelle.



KUVA 8. Välilehdet

Välilehdet toteutettiin kuvan 9 mukaisella ohjelmakoodilla. Koodissa määritellään painikkeelle id, class ja style. Id:ksi yleensä laitetaan jokin selkeä nimi mahdollisesti lyhennettynä, jotta nopeasti katsomalla tiedetään, mihin kohtaan käyttöliittymässä kyseinen koodin osa vaikuttaa. Class ja style arvot ovat suo-

raan HTML-Kickstartin omia tyylimuotoiluja, joihin löytyy ohjeet HTML-Kickstartin nettisivuilta. Välilehden tiedot haetaan omasta PHP-tiedostosta nopeasti *include*-komennolla.

```
<!-- TAB CONTENT FOR AMPLIFIERS -->
<div id="btn-amp" class="tab-content clearfix" style="display: block;">
  <?php
    include 'amps.php';
  ?>
</div>

<!-- TAB CONTENT FOR GROUPS -->
<div id="btn-groups" class="tab-content cleafix" style="display: block;">

  <?php
    include 'group.php';
  ?>
</div>
```

KUVA 9. Kuvakaappaus ohjelmakoodista, *include*.

Haasteellisin osuus käyttöliittymän toiminnollisuudessa oli painikkeiden logiikan miettiminen ja niiden ongelmakohtien poistaminen. Kuvassa 10 näkyy ohjelmakoodi yhdestä napista. Joka kerta, kun painikkeeseen tehtiin uusi ominaisuus, testattiin se monta kertaa, ennen kuin se toimi haluamalla tavalla.

```
54 #SQL Set Name
55 if (isset($_POST['btnSetName'])) {
56     $labelName = $_POST['txtName'];
57     $select = $_POST['valinta'];
58     $sqlName = "UPDATE device SET name = " . $labelName . " WHERE iddevice = ";
59     if ($labelName != "") {
60         foreach ($select as $key) {
61             $sqlName .= $key . " ";
62         }
63         echo $sqlName;
64         $conn->query($sqlName);
65     }
66     else {
67         echo "No name";
68     }
69     header("Location: " . $_SERVER['PHP_SELF'] . "#btn-amp");
70
71 }
```

KUVA 10, Kuvakaappaus ohjelmakoodista

Koodissa jokaisen napin toiminnollisuus aloitettiin *if (isset)*-ehdolla, joka poisti käyttöliittymästä virhetekstiä. Lauseke *isset (variable)* tarkistaa *variable*-muuttujasta ehdon, että siinä on dataa eikä se ole tyhjä. Ohjelmakoodissa oleviin tekstinsyöttökenttiin tehtiin myös virheentarkistus, jotta ohjelmaa ei suoriteta, jos käyttäjä jättää kirjoittamatta tekstiä. Lauseke *if(\$labelName != "")* tarkistaa sen, että muuttuja *\$labelName* on muuta kuin tyhjä. Operaattori *!=* tarkistaa ehtolausekkeesta, että muuttuja *\$labelName* ei vastaa ehdossa olevaa tyhjää arvoa. Toiminnollisuuden loppuun asetettiin *header*-rivi, joka tyhjentää osoitekentän tiedot ja asettaa olemassa olevan sivun polun uusiksi selaimen osoitekenttään.

7 YHTEENVETO

Työn tarkoitus oli luoda toimiva käyttöliittymäkokonaisuus TeleAmp-vahvistinlaitteelle. Työ rakennettiin kokonaisuudessa HTML-kielellä apuna käyttäen HTML-Kickstart web front end -kehitysympäristöä. Aikaa kului paljon tietoa etsiessä ja kokeillessa uusia vastaantulevia esimerkkejä soveltaen niitä projektiin.

Digi ConnectPort X4 IA toi myös haastetta projektin läpivientiin. Alun perin ConnectPortin piti toimia verkkosivua ylläpitävänä laitteena, mutta laitteen vanha Python-versio haittasi tilannetta. Laitteen tuettu Python-versio on versio 2.4 ja yleisimmät Pythonilla toimivat serverit vaativat version 2.7. Verkkosivun ylläpitoon valittiin laitteeksi Raspberry Pi -mikrotietokone, jossa laitteen fyysinen koko sekä virrankulutus on pieni. Nämä spesifikaatiot sopivat hyvin projektiin.

Digi ConnectPortille ei onnistuttu asentamaan Python-web-palvelinta, vaan web-palvelin toimii Raspberry Pi -mikrotietokoneella. Projektissa huomattiin tietokannan olevan oleellinen ja tärkeä paikka tallettaa helposti tietoa ja tämän takia relaatiotietokanta otettiin tuotteen kehitykseen mukaan. Suurin osa erittäin tärkeistä vaatimuksista toteutettiin, kuten neljäportainen vahvistus, radiokanavan asetus, mykistäminen, vahvistimen nimeäminen, ryhmien luominen ja laitteiden hallinta ryhmänä, virransäästötila, manuaalinen viiveen asettaminen, radiopiirin lähtötason säätäminen ja toimivuus ilman internet yhteyttä eli niin sanottu offline toimivuus, jolloin käyttäjän täytyy toimia saman langattoman tukiaseman läheisyydessä kuin ConnectPort ja web-palvelin.

Jatkoa ajatellen jäi monia asioita vielä kehittämisen varaan, kuten offline-kartat, viiveen laskukone, akun jännitetason näyttäminen ja datayhteyden luominen ConnectPortin ja RPi:n välille. Datayhteys on keskeisin asia ja tärkein kaikista, jotta tuote saadaan toimimaan, koska ilman datayhteyttä ei voida käyttöliittymällä ohjata yhtään laitetta.

Projektin myötä tietotaidon määrä web front end -kehityksessä on kasvanut huomattavan paljon ja tämä on ollut opettavaista. Projektiin käytettyä aikaa olin voinut kanavoida paremmin ja hyödyntää ajan tehokkaammin.

LÄHTEET

1. Release Notes PN 93000698_N3. 2015. Digi International Inc. Saatavilla: http://ftp1.digi.com/support/firmware/93000698_N3.txt. Hakupäivä: 26.5.2015.
2. Digi ConnectPort X4. 2015. Digi International Inc. Saatavissa: http://www.digi.com/pdf/ds_connectportx4.pdf. Hakupäivä 8.1.2015.
3. Web Frameworks for Python. 21.3.2015. Python Software Foundation. Saatavilla: <https://wiki.python.org/moin/WebFrameworks>. Hakupäivä 15.5.2015.
4. Kuva Digi ConnectPort X4. 2015. Digi International Inc. Saatavissa: <http://www.digi.com/products/wireless-routers-gateways/routing-gateways/connectportx4#overview>. Hakupäivä 6.4.2015.
5. Raspberry Pi. 2015. Raspberry Pi Foundation. Saatavissa: www.raspberrypi.org. 2.2.2015.
6. Kuva Raspberry Pi. 2015. Raspberry Pi Foundation. Saatavissa: <http://www.raspberrypi.org/forums/viewtopic.php?t=4751>. Hakupäivä 6.4.2015
7. Code a simple socket server in Python. 4.7.2013. BinaryTides, Silver Moon. Saatavilla: <http://www.binarytides.com/python-socket-server-code-example/>. Hakupäivä 4.5.2015.
8. Fsockopen. 2015. PHP Group. Saatavilla: <http://php.net/manual/en/function.fsockopen.php>. Hakupäivä 4.5.2015.
9. HTML. 9.4.2015. WhatWG. Saatavissa: <https://wiki.whatwg.org/wiki/FAQ>. Hakupäivä 11.5.2015.
10. HTML Living Standard. 2015. WhatWG. Saatavilla: <https://html.spec.whatwg.org/multipage/>. Hakupäivä 11.5.2015.
11. Cascadin Style Sheet. 2015. Wikipedia. Saatavissa: http://fi.wikipedia.org/wiki/Cascading_Style_Sheets. Hakupäivä 6.4.2015.
12. PHP 5 introduction, what is PHP. 2015. W3Schools. Saatavissa: http://www.w3schools.com/php/php_intro.asp. Hakupäivä 8.1.2015.
13. PHP. 2015. Wikipedia. Saatavilla: <http://fi.wikipedia.org/wiki/PHP>. Hakupäivä 8.5.2015.

14. HTML-Kickstart. 2014. StackOverflow. Saatavilla: <http://stackoverflow.com/tags/html5-kickstart/info>. Hakupäivä 6.4.2015.
15. HTML Kickstart. 2014. 99Lime. Saatavilla: <http://www.99lime.com/elements/>. Hakupäivä 6.4.2015.
16. MySQL. 2015. Wikipedia. Saatavissa: <http://en.wikipedia.org/wiki/MySQL>. Hakupäivä 20.5.2015
17. Top 10 MySQL GUI Tools. 2015. DatabaseJournal. Saatavilla: <http://www.databasejournal.com/features/mysql/article.php/3880961/Top-10-MySQL-GUI-Tools.htm>. Hakupäivä 26.5.2015.
18. MySQL-Workbench. 2015. Oracle Corporation. Saatavilla: <https://www.mysql.com/products/workbench/>. Hakupäivä 6.4.2015.
19. Vagrant. 2015. Wikipedia. Saatavilla: http://en.wikipedia.org/wiki/Vagrant_%28software%29. Hakupäivä 22.5..2015.
20. Cerminara, Nicholas, Scotch Box, 2015. Saatavilla: <https://box.scotch.io/>. Hakupäivä 6.4.2015.

Erittäin tärkeät ominaisuudet

- Tutki ja asenna Python serveri Digi ConnectPort X4 IA -laitteelle.
 - Jos ei toimi, tutki vaihtoehtoinen ratkaisu
- Tutki tietokannan tarpeellisuus

Vaatimukset käyttöliittymälle

- 4-portainen vahvistus
- Radiokanavan asetus
- Mykistys
- Vahvistimien nimeäminen
- Ryhmien luominen ja hallinnointi ryhmänä
- Virransäästötila
- Liikkumistila
- Akun jännitteentason näyttäminen
- Käyttöliittymäsivun päivitys
- Käynnistyessä etsii kaikki ZigBee radiot kantoalueelta
- Toimivuus ilman internetliittymää, ns. offline toimivuus.

Tärkeät ominaisuudet (optionaalinen)

- Sijainti kartalla GPS-tiedon perusteella
- Viivelaskelma GPS-tiedon perusteella tai manuaalinen asetus

Keskitason ominaisuudet (optionaalinen)

- Radiopiirin lähtötason säätäminen

```
<?php

$servername = "localhost";

$username = "u_name";

$password = "PassWd";

$dbname = "test_tbl";

// Create connection

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect_error) {

    die("Connection failed: " . $conn->connect_error);

}

?>
```

```
<?php
include 'connection.php' //Valmis komponentti MySQL yhteyden avaamiseen

// Luodaan MySQL kyselylauseke datan hakemiseen
$sql = "SELECT * FROM test_tbl";
// Suoritetaan haku ja sijoitetaan tulos muuttujaan
$result = $mysqli->query($sql) // $mysqli haetaan connection.php tiedostosta

?>

<div class="col_9 column">
  <fieldset>
    <legend>Amplifiers</legend>
    <table class="striped">
      <thead><tr>
        <th>&nbsp;</th>
        <th>ID</th>
        <th>VALUE</th>
        <th>GROUP ID</th>
      </tr></thead>
      <tbody>

<?php while($row = $result->fetch_array(MYSQLI_ASSOC)) {
// Luodaan while-loop joka kirjoittaa oikean määrän alla olevaa koodia.
?>
  <tr>
    <td><?php echo $row['id']; ?></td>
    <td><?php echo $row['value']; ?></td>
    <td><?php echo $row['group_id']; ?></td>
  </tr>

<?php }
// suljetaan while loop
?>

  </tbody>
</table>
</fieldset>
</div>
```