

Shuangliang Kolu

# Kehokartta-nettisivun suunnittelu ja toteutus

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikka

Insinöörityö

10.4.2015

|   |   |
|---|---|
| Tekijä<br>Otsikko   | Shuangliang Kolu<br>Kehokartta-nettisivun suunnittelu ja toteutus |
| Sivumäärä<br>Aika   | 30 sivua<br>10.4.2015   |
| Tutkinto  | Insinööri (AMK)   |
| Koulutusohjelma   | Tietotekniikka  |
| Suuntautumisvaihtoehto  | Ohjelmistotekniikka   |
| Ohjaaja(t)  | Lehtori Ilpo Kuivanen<br>Projektipäällikkö Teemu Hammaren         |
| <p>Insinööriyön aiheena on Kehokartta-nettisivun suunnittelu ja toteutus. Kehokartta on kehitysvaiheessa oleva terveysseurantapalvelu, jonka avulla käyttäjät voivat seurata omaan terveyteensä liittyviä asioita. Kehokartta-palvelu tarjoaa terveysseuranaan lisäksi monia erilaisia laboratoriomittauksia. Terveystilanne ja laboratorion mittaustulokset esitellään käyttäjille diagrammien avulla.</p> <p>Projektin tavoitteena oli luoda nettisivu palvelun käyttäjille, laboratoriolle ja sivuston ylläpitäjälle. Sivustot ovat responsiiviset, mobiilystävälliset, ja ne tarjoavat hyvän käyttöliittymän käyttäjille. Projektissa on tarkoitus käyttää sivuston runkona Bootstrap-kirjastoa, joka tarjoaa responsiivisen pohjarakenteen työpöytään ja mobiiliin.</p> <p>Projektissa käytettävät ohjelmointikielet ovat HTML, HTML5, CSS, PHP, SQL ja JavaScript. Projektin kehitystyökaluna on WampServer. Kehityspalvelin on Apache, joka käyttää PHP-versiota 5.5 ja MySQL-versiota 5.6. Työssä käytetään seuraavia kirjastoja: Bootstrap v3.3.2:ta runkona, Flot 0.8.3 -diagrammia piirtämiseen, html5shiv:tä ja Respond HTML5:tä simuloimiseen vanhoissa Internet Explorer -selaimessa.</p> <p>Projekti loppui onnistuneesti. Se täytti kaikki asetetut vaatimukset ja tarjoaa hyvän rakenteen jatkokehitykselle.</p> |   |
| Avainsanat  | Nettisivu, HTML, JavaScript, PHP, SQL                             |

|   |   |
|---|---|
| Author(s)<br>Title  | Shuangliang Kolu<br>Kehokartta webpages design and implementation |
| Number of Pages<br>Date   | 30 pages<br>10 April 2015   |
| Degree  | Bachelor of Engineering   |
| Degree Programme  | Information Technology  |
| Specialisation option   | Software Engineering  |
| Instructor(s)   | Ilpo Kuivanen, Senior Lecturer<br>Teemu Hammaren, Project Manager |
| <p>The topic of this thesis is the design and implementation of the Kehokartta website. Kehokartta is a health monitoring service in the development phase, which offers a many health related variables that the service users can track by themselves. The Kehokartta service also offers many different laboratory analyses. The health condition and the laboratory results are displayed to the service users in forms of diagrams.</p> <p>The aim of the project was to create a website for the users, laboratory and webmaster. The sites are responsive, mobile friendly and offer a good user interface to the service users. The websites use the Bootstrap library as framework. Bootstrap libraries offer a responsive framework to desktop and mobile devices.</p> <p>The programming languages used during the project were the following: HTML, HTML5, CSS, PHP, SQL and JavaScript. The project development tool was WampServer. The development platform was Apache server us-ing PHP 5.5 and MySQL 5.6. The project also used the following libraries: Bootstrap v.3.3.2 as framework, Flot 0.8.3 to draw the diagrams, html5shiv and Respond to simulate HTML5 in older Internet Explorer browsers.</p> <p>The project was completed successfully. It met all requirements and offers a very good framework for future development.</p> |   |
| Keywords  | Webpage, HTML, JavaScript, PHP, SQL                               |

# Sisällys

## Lyhenteet

|       |                            |    |
|-------|----------------------------|----|
| 1     | Johdanto                   | 1  |
| 2     | Suunnittelu                | 2  |
| 2.1   | Pohjarakenne               | 2  |
| 2.2   | Käyttäjän sivusto          | 3  |
| 2.3   | Laboratorion sivusto       | 5  |
| 2.4   | Hallinta sivusto           | 6  |
| 2.5   | Tietokanta                 | 6  |
| 2.5.1 | Tunnukset                  | 7  |
| 2.5.2 | Asetukset                  | 7  |
| 2.5.3 | Arvot                      | 8  |
| 2.5.4 | Mittaustulokset            | 9  |
| 3     | Toteutus                   | 10 |
| 3.1   | Pohjarakenne               | 11 |
| 3.1.1 | Sivuston pohja             | 11 |
| 3.1.2 | JavaScript                 | 13 |
| 3.1.3 | SQL-kyselyt                | 16 |
| 3.2   | Käyttäjän sivusto          | 17 |
| 3.2.1 | Tilanne-sivu               | 17 |
| 3.2.2 | Arvon-sivut                | 19 |
| 3.2.3 | Asetus-sivu                | 21 |
| 3.3   | Laboratorion sivusto       | 22 |
| 3.3.1 | Käyttäjälista-sivu         | 22 |
| 3.3.2 | Tulos-sivu                 | 24 |
| 3.4   | Hallinta-sivu              | 26 |
| 4     | Jatkokehitys ja yhteenveto | 28 |
|       | Lähteet                    | 30 |

## Lyhenteet

|       |  |
|-------|--|
| WWW   | World Wide Web. Internet-verkossa toimiva hajautettu hypertextijärjestelmä.                |
| HTML  | Hypertext Markup Language. Standardoitu kuvauskieli.                                       |
| HTML5 | Hypertext Markup Language version 5. Uusi HTML-versio, joka tarjoaa enemmän ominaisuuksia. |
| CSS   | Cascading Style Sheets. WWW-dokumenteille kehitetty tyylikieli.                            |
| PHP   | Hypertext Preprocessor. Palvelinpuolen ohjelmointikieli.                                   |
| PDO   | PHP Data Object. SQL-kyselyn suoritukseen tarkoitettu laajennus PHP-kielille.              |
| SQL   | Structured Query Language. IBM:n kehittämä standardoitu kyselykieli tietokannoille.        |

## 1 Johdanto

Insinööriyön aiheena on Kehokartta-nettisivun suunnittelu ja toteutus. Kehokartta on kehitysvaiheessa oleva terveysseurantapalvelu, jonka avulla käyttäjät voivat seurata omaan terveyteensä liittyviä asioita. Projektin tavoitteena on luoda nettisivu palvelun käyttäjille, laboratoriolle ja sivuston ylläpitäjälle. Sivustot ovat responsiiviset, mobiilistävälliset ja tarjoavat hyvän käyttöliittymän käyttäjille. Projektissa on tarkoitus käyttää sivuston runkona Bootstrap-kirjastoa, joka tarjoaa responsiivisen pohjarakenteen työpöytään ja mobiiliin. Projektin tilaajana on Metropolia Ammattikorkeakoulun Arabianrannan toimipiste käyttömestari Teemu Hammaren.

Monet älypuhelimet ja älykellot tarjoavat monia ohjelmia terveyteen liittyvien asioiden seuraamiseen, mutta monet näistä ohjelmista seuraavat vain muutamaa muuttujaa tai eivät tarjoa monimutkaisia muuttujia. Esimerkiksi monella saattaa olla verensokerimittari tai verenpainemittari kotona ja hän haluaa kirjata nämä mittaukset ylös. Kehokartta-palvelun on tarkoitus tarjota sivusto, jolla on suuri määrä mitattavia muuttujia. Kehokartta tarjoaa myös monia erilaisia laboratoriomittauksia. Mittaustulokset esitellään käyttäjille diagrammien avulla. Laboratorion mittauksen varten käyttäjät lähettävät tarvittavat näytteet laboratorioon, jossa mittaustulos lisätään käyttäjien tietokantaan.

Projektissa käytettävät ohjelmointikieliset ovat HTML, HTML5, CSS, PHP, SQL ja JavaScript. Projektin kehitystyökaluna on WampServer. Kehityspalvelin käyttää PHP-versiota 5.5 ja MySQL-versiota 5.6. Työssä käytetään seuraavia kirjastoja: Bootstrap v3.3.2 -runkoa, Flot 0.8.3 -diagrammia piirtämiseen, html5shiv HTML5:tä ja Respond HTML5:tä simuloimiseen version 9 tai aiemmissa Internet Explorer -selaimissa. Bootstrapin tarjoamat elementit tukevat Internet Explorer -selaimet, joten sivusto tukee helposti enemmän selaimia käyttämällä html5shiv- ja Respond-kirjastoja.

Sivuston käyttöliittymäkielenä on englanti, joten työssä monet sivut, tiedostot, termit ja muuttujien nimet ovat englanniksi. Työssä tiedostot käsitellään aina päätteen kanssa, esimerkiksi index.php:tä. Suurin osa työn koodeista on esimerkkikoodeja tietoturvasuussyistä. Useat toteutukset ovat työssä melko toistuvia, joten toteutusta käsitellään tarkasti vain kerran.

Työ on jaoteltu ja otsikoitu kolmeen eri osaan: suunnitteluun, toteutukseen ja yhteenvehtoon. Suunnittelussa on sivuston ja tietokannan vaatimukset. Toteutuksessa on sivus-

ton rakenne-toteutus ja esimerkkikoodeja. Yhdenvedossa on sivuston tilanne projektin loputtua ja sivuston jatkokehityksestä.

## 2 Suunnittelu

Projektin tavoitteena oli luoda responsiivinen sivusto, jonka voi ottaa käyttöön ja tarjota hyvä pohjarakenne jatkokehitykselle. Sivusto koostuu kolmesta osasta: käyttäjän, laboratorion sekä hallinnan sivustosta. Näillä kaikilla on käytössä sama pohjarakenne, jonka päälle sivustot on rakennettu. Tarkemmat vaatimukset ja suunnittelut on jaettu viiteen osaan. Ne ovat pohjarakenne, käyttäjän sivusto, laboratorion sivusto, hallintasisivusto ja tietokanta.

### 2.1 Pohjarakenne

Sivustot on erotettu toisistaan palvelimessa. Juurikansiossa on käyttäjän sivusto ja kaikki kirjasot. Laboratorio ja hallintasisivustot ovat kansioina juurikansion alla. Pohjarakenneideana on luoda kaikille sivustolle samanlainen pohjarakenne, johon voi helposti lisätä uusia sivuja kehitysvaiheessa. Jokaisella sivustolla on oma päätiedosto `index.php`. Palvelin lataa `index.php:n` ensimmäisenä, kun käyttäjät tulevat sivustoille.

Sivusto käyttää `index.php`-runkoa. `Index.php` itse ei näytä HTML-sisältöä, mutta sen avulla näytetään oikea sivu. Se näyttää oikean sivun PHP:n `include`-funktion avulla. Oikein näytetty sivu tunnistetaan PHP:n sessio-muuttujan avulla. Jokaisella sivustolla on oma sessio-muuttuja, joka säilyttää käyttäjien senhetkisen sijainnin kyseisessä sivustossa. Sen sijaan että käyttäjiä ohjataan uudelle sivulle, sivusto muuttaa HTML-sisällön käyttäen JavaScriptiä. Kun käyttäjät painavat linkkiä tai lähettävät lomakkeen, JavaScript suorittaa kyseisen toiminnan ilman sivuston uudelleenlataamista. Tämän menetelmän avulla sivusto lataa uudelleen vain tarvittavat osat koko sivun sijaan.

Sivun rakenne on jaettu moniin eri PHP-tiedostoihin, jotka kokonaisuudessa muodostavat näytettävän sivu. Tämän toteutustyylin avulla monet PHP-tiedostot ovat uudelleen käytettävissä helposti. JavaScriptin ja tämän toteutustyylin avulla sivusto voi helposti uusia yhden tiedoston sisällön lataamatta koko sivua uudelleen. Kuvan 1 esimerkki esittää sivun, joka on rakennettu kolmesta eri tiedostosta. Sivun pohjana on `index.php`,

ylempi alue on nav.php ja alempi alue on daily.php. Nav.php on navigointipalkki, ja se on jokaisen sivun yläpuolella ja näyttää kyseisen sivuston pääsivut.

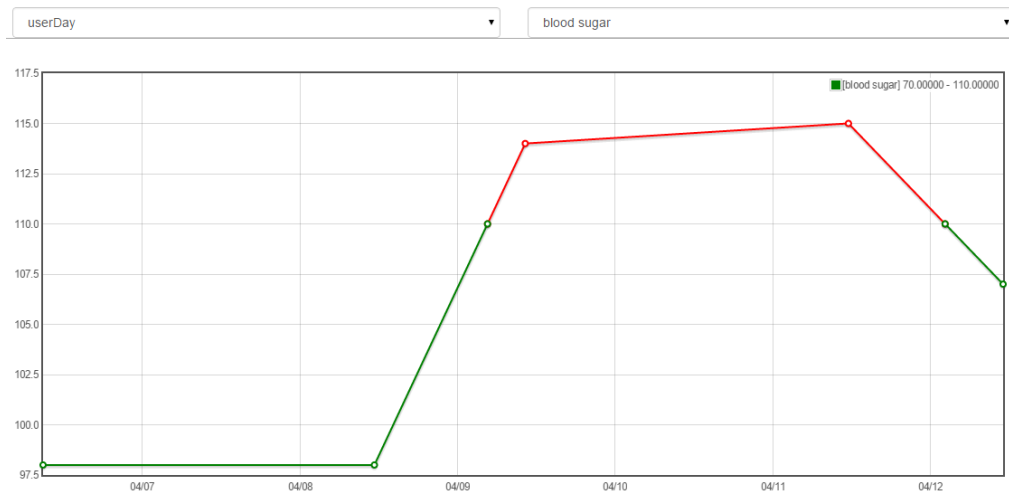
Kuva 1. Esimerkkisivu.

## 2.2 Käyttäjän sivusto

Käyttäjän sivusto on tarkoitettu Kehokartta-palvelun käyttäjille käytettäväksi. Käyttäjän sivusto tarjoaa käyttäjille monia työkaluja, joiden avulla käyttäjät voivat tarkistaa oman terveytensä. Käyttäjä voi ilmoittaa arvoja päivittäin, viikottain tai kuukausittain. Käyttäjän sivustossa on tarkoitus olla seuraavia pääsivuja navigointipalkissa kotisivu home, tilanne status, päivä daily, viikko weekly, kuukausi monthly ja asetus setting. Kotisivu on alustavasti tarkoitettu palvelun ilmoituksia varten. Kehityksessä se oli jätetty tyhjäksi.

Tilanne-sivussa käyttäjät voivat seurata heidän terveystilaansa diagrammien avulla. Kuva 2 esittää sivun rakenteen navigointipalkin alla. Sivun ylävasemmalla on pudotusvalikko arvojen kategorioista. Sitten kun käyttäjä on valinnut kategorian, esille tulee pudotusvalikko arvoista, jotka kuuluvat kyseiseen kategoriaan. Valitun arvon avulla sivusto luo käyttäjille diagrammin valitun arvon mittaustuloksesta käyttäen Flot-kirjastoa. Diagrammin x-akseli on mittauksen päivämäärä ja y-akselin on mittauksen tulos. Kun mittaus ylittää arvon terveellisen tulosrajan, viivan väri muuttuu punaiseksi.





Kuva 2. Verensokerin diagrammi.

Asetus-sivulla käyttäjät voivat asettaa palvelun asetukset. Alustavasti asetukset ovat arvojen aktivoiminen, niiden toistojaksojen asettaminen, sähköpostin ja salasanan muuttaminen. Arvot voivat esimerkiksi toistua 3 päivän välein, 2 viikon välein ja niin edelleen. Aktivoitujen ja ei-aktivoitujen arvojen ulkoasut ovat erilaisia. Kuva 3 esittää aktivoitujen ja ei-aktivoitujen arvojen ulkoasut. Kuvassa 3 vasemmalla on askelmäärä, joka ei ole aktivoitu ja oikealla kalorimäärällä, joka on aktivoitu yhden päivän toistojaksolla.

Daily status iteration.

Kuva 3. Päivän arvot asetussivussa.

Seurantasivut ovat päivä daily, viikko weekly ja kuukausi monthly. Käyttäjät voivat lisätä ja muokata heidän seuraamansa arvojen mittaustulokset näillä sivuilla. Sivut näyttävät kaikki aktivoitujen arvojen kyseisessä sivussa. Sivut näyttävät myös, milloin seuraava lisäyspäivämäärä on kuvan 4 mukainen. Nämä sivut näyttävät myös vanhoja mittaustuloksia. Arvon, jonka mittaushistoria näyttää, valitaan pudotusvalikon avulla. Kun arvo on valittu, esille tulee mittaustuloslista. Mittaustuloslistan avulla käyttäjät voivat muokata vanhoja mittaustuloksia.

Kuva 4. Arvon syöttökenttä päiväsivussa.

Kuvasta 4 näkee esimerkin päivä-sivun syöttökentästä. Vasemmalla on poltettu kalorit, jonka lisäyspäivämäärä on ylitetty. Oikealla on sykemittaus, jonka lisäyspäivämäärää ei ole ylitetty. Käyttäjät voivat silti enable-napin avulla lisätä uuden mittaustuloksen, vaikka sen lisäyspäivämäärää ei ole ylitetty.

### 2.3 Laboratorion sivusto

Käyttäjät voivat Kehokartta-palvelun avulla mitata monia arvoja, jotka vaativat laitteistoa ja kemikaaleja. Kehokartta-palvelun laboratorio suorittaa monia näistä mittauksista. Palvelun käyttäjät lähettävät näytteet laboratorioon, jossa näytteen arvot mitataan ja mittaustuloksia merkitään palvelun käyttäjille. Laboratorion sivuston avulla laboratorion käyttäjät voivat kirjata näytteen mittaustulokset palvelun käyttäjille. Laboratorion sivustolla on seuraavat pääsivut navigointipalkissa: käyttäjät user, mittaustulos result ja asetus setting. Asetussivu on alustavasti laboratorion käyttäjien sähköpostin ja salasanan muuttamiseen. Jatkokehityksessä saattaa tulla lisää ominaisuuksia asetussivuun, joten sen on hyvä olla erillinen sivu.

Palvelun tietokannassa on paljon henkilökohtaisia tietoja. Yksityisyyden suojaamiseksi laboratorio käsittelee käyttäjiä heidän numerokoodiaan käyttäen. Numerokoodit generoidaan käyttäjille rekisteröintivaiheessa.

Käyttäjät-sivulla on lista kaikista palvelun käyttäjistä. Laboratorion käyttäjät voivat valita tämän listan avulla palvelun käyttäjiä. Kun käyttäjä on valittu, sivusto ohjautuu suoraan mittaustulossivulle. Käyttäjä-sivun käyttäjälista näyttää vain 10 käyttäjää kerrallaan estääkseen liian korkean sivun rakenteen. Käyttäjälistaa navigoidaan sen oman valitsinpalkin avulla. Valitsinpalkin avulla laboratorion käyttäjät voivat valita, ketkä kymmenestä käyttäjistä näytetään. Esimerkiksi 1 - 10, 11 - 20, 21 - 30 ja niin edelleen.

Mittaustulossivussa laboratorion käyttäjät voivat lisätä ja muuttaa palvelun käyttäjien mittaustuloksia. Laboratorion käyttäjät näkevät pelkästään heidän lisäämänsä mittaustulokset valitun palvelun käyttäjän mittaushistoriassa. Tämän sivun avulla laboratorion käyttäjät voivat myös hakea palvelun käyttäjiä heidän numerokoodinsa avulla.

## 2.4 Hallintasivusto

Hallintasivusto on tarkoitettu palvelun ylläpitäjille. Hallintasivustossa voi lisätä ja muuttaa mitattavia arvoja, laboratorion käyttäjätunnuksia ja tarkistaa palvelun käyttäjien tunnukset. Hallintasivustolla on seuraavia pääsivuja: navigointipalkissa käyttäjät user, arvot variable, laboratorion käyttäjät labuser ja asetus setting. Asetussivu on alustavasti ylläpitäjien sähköpostin ja salasanan muuttamiseen. Jatkokehityksessä saattaa tulla lisää palveluun liittyviä asetuksia, joten se on parempi luoda erillisenä sivuna kehitysvaiheessa.

Käyttäjäsivussa on lista kaikista palvelun käyttäjistä. Käyttäjää tunnistetaan heidän numerokoodinsa avulla. Ylläpitäjä voi tämän sivun avulla generoida uuden salasanan käyttäjille. Kyseinen ominaisuus on tarkoitettu vain hätätilaa varten.

Arvon-sivussa on lista kaikista palvelua mitattavista arvoista. Jokaisella mitattavalla arvolla on nimi, mittaussyksikkö, kategoria, terveellinen minimi ja maksimitulokset. Kategoria on arvon alijärjestelmä. Käyttäjien itse mittaamien arvojen kategoriat ovat alijärjestelmän sijaan niiden toistojaksot. Toistojaksot ovat päivä, viikko ja kuukausi.

Laboratorion käyttäjäsivu listaa kaikki laboratorion palvelun käyttäjät. Ylläpitäjä luo tämän sivun avulla laboratorion käyttäjille uudet tunnukset. Laboratorion tunnukset tarvitsevat nimikkeen ja sähköpostin. Sivun generoi luonnin jälkeen väliaikaisen salasanan laboratorion käyttäjille ja lähettää sen heille sähköpostissa.

## 2.5 Tietokanta

Tietokannassa ovat kaikki palvelun tiedot. Säilytettävät tiedot ovat sivustojen käyttäjätunnuksien, käyttäjien ja laboratorion mittaustulokset ja arvojen tiedot. Tiedot on jaettu moniin eri tauluihin tietokannassa. Tietokannassa on paljon linkitystä taulujen välillä. Linkityksen avulla taulut pääsevät helposti käsiksi linkitetyn taulun tietoihin id-solun avulla. Linkitetty id annetaan SQL-kyselyssä SQL:n join-parametrin arvoksi. SQL:n join yhdistää kyselyn aikana taulut yhteen niiden linkitettyjen solujen avulla. SQL-kysely, jossa käytetään SQL joinia, tuo enemmän räsitystä yhden SQL-kyselyn suoritukseen, mutta säästää erittäin paljon tilaa tietokannassa, jos sitä käytetään oikein.

Tietokanta on jaoteltu kolmeen erilaiseen tauluun. Ne ovat tunnukset, asetukset, arvot ja mittaustulokset.

### 2.5.1 Tunnukset

Tunnustaulut on jaettu kolmeen eri tauluun tietokannassa. Taulut ovat palvelun käyttäjille nimeltä user, laboratorion käyttäjille nimeltä labuser ja palvelun ylläpitäjille nimeltä admin. Syynä on SQL-kyselyn helpottaminen, joka estäisi yksinäisen ison taulun muodostumisen tietokannassa. Käyttäjätunnuksen ja salasanan tarkistaminen on myös paljon helpompaa kirjautumisessa, jos jokaisella sivustolla on oma tunnustaulu. Eri sivustot tarkistavat vain omasta taulusta tunnukset ja salasanat. Näillä tauluilla on silti melko sama rakenne. Jokaisella taululla on seuraavia soluja.

- Id-solu on uniikki tunnistusavain. Linkitykset käyttävät tämän solun SQL joinin parametrin arvona.
- Username-solu on käyttäjätunnus. Palvelun käyttäjän taulussa tämän arvona on numerokoodi. Laboratorion ja ylläpitäjän taulussa tämän arvona on käyttäjätunnus.
- Email-solu on käyttäjien sähköpostiosoite.
- Password-solu on kryptattu salasana.
- Date-solu on tunnuksen päivämäärä.

Taulukko 1. Tunnuksien taulujen rakenne.

| Solun nimi | Solun data tyyppi                          |
|------------|--|
| id         | INT, PRIMARY KEY, NOT NULL, AUTO INCREMENT |
| username   | VARCHAR(100)                               |
| email      | VARCHAR(255)                               |
| password   | VARCHAR(255)                               |
| date       | INT  |

### 2.5.2 Asetukset

Asetus-taulu on nimeltä uservars. Tässä taulussa on palvelun käyttäjien päivä, viikko, kuukausiarvojen toistojaksot. Arvot on linkitetty user-tauluun ja variable-tauluun niiden

id-solun avulla. Sivusto tarkistaa tämän taulun avulla, mitä arvoja käyttäjät seuraavat, niiden toistovälit ja niiden seuraavan lisäyspäivämäärän.

- Id-solu on uniikki tunnistusavain. Linkitykset käyttävät tämän solun SQL joinin parametrin arvona.
- UserId-solu on linkitys palvelun käyttäjän tauluun.
- VarsId-solu on linkitys arvon tauluun.
- Iteration-solu on linkitetyn arvon toistoväli linkitetyllä käyttäjällä.
- Type-solu on linkitetyn arvon toistojaksoon tyyppi. Muuttujana on päivä, viikko tai kuukausi.

Taulukko 2. Asetustaulun rakenne

| Solun nimi | Solun data tyyppi                          |
|------------|--|
| id         | INT, PRIMARY KEY, NOT NULL, AUTO INCREMENT |
| userId     | INT  |
| varsId     | INT  |
| iteration  | INT  |
| type       | VARCHAR(255)                               |

### 2.5.3 Arvot

Arvot-taulu on nimeltään variable. Tässä taulussa ovat kaikki sivustossa mitattavien arvojen tiedot. Tämän taulun ja linkityksen avulla tietokanta pystyy säästämään erittäin paljon tilaa mittaustulostauluista. Sen sijaan, että jokaisella mittaustulos-taulun rivillä olisi arvon tiedot, ne säilytetään tässä taulussa. Tämän linkityksen avulla mittaustulos-taulu tarvitsee ainoastaan yhden solun linkitykseen. Monet kyselyt tarvitsevat myös ainoastaan arvojen tiedot, joten se on hyvä olla omana tauluna.

- Id-solu on uniikki tunnistusavain. Linkitykset käyttävät tämän solun SQLää joinin parametrin arvona.
- Name-solu on arvon nimi.
- Unit-solu on arvon yksikkö.

- Min-solu on alin tulos terveellisen rajalla.
- Max-solu on ylin tulos terveellisen rajalla.
- Category-solu on arvon kategoria. Käyttäjän itse mittaamissa arvoissa ne ovat userday, userweek, usermonth.

Taulukko 3. Arvon tauluun rakenne

| Solun nimi | Solun data tyyppi                          |
|------------|--|
| id         | INT, PRIMARY KEY, NOT NULL, AUTO INCREMENT |
| name       | VARCHAR(255)                               |
| unit       | VARCHAR(255)                               |
| min        | DECIMAL(10,5)                              |
| max        | DECIMAL(10,5)                              |
| category   | VARCHAR(255)                               |

#### 2.5.4 Mittaustulokset

Mittaustulokset on jaettu neljään eri tauluun. Taulut ovat toistojakson mukaan ja laboratorion mittauksista varten. Taulut ovat toistojakson päville nimeltä daily, toistojakson viikolle nimeltä weekly, toistojakson kuukaudelle nimeltä monthly ja laboratorion mittaukselle nimeltä labresult. Jaottelut vähentävät mittaustulosten määrää per taulu ja helpottavat sivuston SQL-kyselyiden toteutusta huomattavasti. Labresult-tilussa on yksi solu enemmän kuin muissa tauluissa. Se tarvitsee mittaustuloksen lisääjän tiedot. Laboratorion käyttäjät näkevät pelkästään omat mittaukset valitulle käyttäjille.

- Id-solu on uniikki tunnistusavain. Linkitykset käyttävät tämän solun SQL joinin parametrin arvoa.
- UserId-solu on linkitys palvelun käyttäjän tauluun.
- VarsId-solu on linkitys arvon tauluun.
- Value-solu on mittaustulos.

- Date-solu on mittauspäivämäärä.

Taulukko 4. Mittaustulostaulujen rakenne.

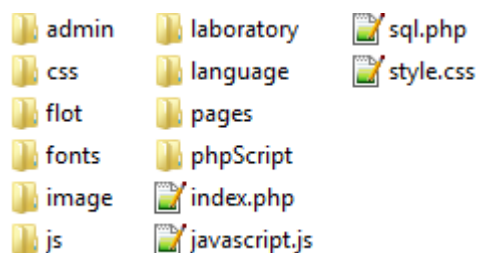
| Solun nimi | Solun data tyyppi                          |
|------------|--|
| id         | INT, PRIMARY KEY, NOT NULL, AUTO INCREMENT |
| userId     | int  |
| varsId     | int  |
| value      | decimal(10,5)                              |
| date       | int  |

Kaikilla neljällä tauluilla on taulukko 4:n rakenne. Labresult-tilulla on näiden lisäksi labuserid-solu. Labuserid-solu on linkitys laboratorion käyttäjän tauluun. Tämän solun avulla sivusto tunnistaa mittaustuloksen lisääjän. Laboratorion käyttäjät näkevät laboratorion sivuston tulossivussa pelkästään heidän lisäämänsä mittaustulokset valituille käyttäjille. Sivustossa suoritettava SQL-kysely osaa labuserid-solun avulla hakea oikeat mittaustulokset taulusta.

### 3 Toteutus

Palvelimella on kolme eri sivustoa. Ne ovat palvelun käyttäjille, laboratorion käyttäjille ja ylläpitäjille. Kaikki kolme sivustoa käyttävät samaa pohjarakennetta, toteutustyyliä, ulkoisia kirjastoja ja CSS-tiedostoa. Palvelinjuurikansiossa ovat kaikki kirjastot ja palvelun käyttäjän sivusto. Laboratorion ja ylläpitäjän sivustot ovat omissa alikansioissa. Kehitysvaiheessa kaikkia kirjastoja ei ylläpidetä paikallisessa palvelimessa. Ne on lisätty ulkoisen palvelimen kautta. Näin sivustot saavat käyttöön aina viimeisimmän version. Kehityksen loppuvaiheessa on silti hyvä ottaa vakaan versio kyseisestä kirjastosta palvelimelle.

Kuva 5 esittää palvelimen juurikansion kehitysvaiheessa. Admin-kansion sisällä on ylläpitäjän sivusto. Laboratory-kansion sisällä on laboratorion sivusto. Css, fonts ja js ovat Bootstrap-kirjaston kansiot. Flot on Flot-kirjaston kansio. style.css on sivuston yhteinen CSS-tyylitiedosto. Sql.php:n sisällä ovat käyttäjän sivuston SQL-kyselyt. Javascript.js:n sisällä on käyttäjän sivuston JavaScript-funktiot. Phpscript-kansion sisällä on kaikki käyttäjän sivuston PHP-funktion tiedostot.



Kuva 5. Palvelinjuurikansio

### 3.1 Pohjarakenne

Sivustoilla on yhteinen pohjarakenne, jota jokainen sivusto palvelimessa käyttää. Sivustojen pohjasivu `index.php` on luotu projektin alussa. Sen jälkeen pohjasivu kopioitiin jokaisen sivuston omiin kansioihin. Jokaisessa sivustossa on seuraavia kansioita ja tiedostoja omassa kansiossa. Kyseiset tiedostot ja kansiot ovat kuvan 5 tyyllisesti palvelimella.

- `Index.php` on tiedosto, jonka palvelin lataa ensimmäiseksi, kun käyttäjät tulevat sivustolle.
- `Javascript.js`-tiedostossa on kyseisen sivuston JavaScript-funktiot.
- `Sql.php`-tiedostossa on PHP-luokka, jonka sisällä ovat kaikki kyseisen sivuston SQL-kyselyt.
- `Pages`-kansiossa ovat kaikki kyseisen sivuston pääsivut.
- `Phpscript`-kansiossa ovat kaikki kyseisen sivuston PHP-funktiot.

Kaikki palvelun sivustot käyttävät alustavasti Bootstrapin tarjoamia tyylejä. Pientä muokkausta on tehty joihinkin HTML-elementtien kokoihin. Melko paljon sivuston elementeistä käyttää Bootstrap Grid. Bootstrap Gridiä on joukko CSS-tyyliluokkia HTML-elementille. Nämä luokat muuttavat HTML-elementtien ulkoasut selainresoluution mukaan.

#### 3.1.1 Sivuston pohja

`Index.php`-tiedoston rakenne on melko sama kaikissa sivustoissa. Ainoat erot ovat PHP-session muuttujien nimet, kirjaston polut ja `style.css`-tiedoston polut. `Index.php` sisältää suurin osiin HTML-koodeja, vaikka sen päätteessä on `php`. Ainoat PHP-koodit



Index.php tiedostossa ovat alussa suoritettava PHP-session aloitus ja body-elementissä suoritettava sijaintitarkastus. Index.php:n HTML:n sisältö koostuu kahdesta osasta: head-elementistä ja body-elementistä.

Head-elementtiin sisällä ovat kaikki HTML-metamäärittelyt. Ne ovat esimerkiksi sivuston nimi, kirjastojen sijainti ja style.css-tyylitiedoston sijainti. Head-elementin sisällä tarkistetaan näiden lisäksi selainversion Internet Explorer varalta, koska sivustossa käytetään HTML5-elementtejä eikä vanhoja Internet Explorer -versioita tueta HTML5:ssä. Kuva 6 esittää ehtolauseen, jota käytetään selaintarkistuksessa. Jos selain on vanha Internet Explorer, sivusto simuloi HTML5:n tarjoamaan toiminnot käyttämällä html5shiv:tä ja respond-kirjastoja. Kuvassa 6 käytetään HTML:n ehdollinen kommentti, jota Internet Explorerin versiot 5 – 9 käyttävät. Ensimmäinen rivi tarkistaa, onko selain Internet Explorer 9 tai alempi versio. Jos selain on versiota 9 vanhempi, se lataa html5shiv:in ja respond-kirjastot.

```
<!--[if lt IE 9]>  
<script src=“./html5shiv-printshiv.min.js”></script>  
<script src=“./respond.js”></script>  
<![endif]-->
```

Kuva 6. Head-elementtiin selaintarkistus.

Body-elementissä sivusto lataa näytettävän HTML-elementit käyttäen PHP:n session muuttujaa nimeltä currentpage. Currentpage-muuttujan avulla sivusto tietää, missä pääsivussa käyttäjä on. Pääsivut ovat sivuja, joihin päästään navigointipalkista. Kuva 7 esittää koodin, joka tarkistaa, ovatko käyttäjät kirjautuneet sivustolle, ja ohjaa käyttäjät oikeille sivulle käyttäen currentpage-muuttujaa. Ensiksi se tarkistaa, ovatko käyttäjät kirjautuneet sivustolle käyttäen PHP-session muuttujaa userdata. Jos eivät ole, niin sivusto näyttää kirjautumiselle tarkoitetun sivun nimeltä login.php. Userdata-muuttuja sisältää käyttäjä id:n ja numerokoodin. Sivusto tunnistaa myös käyttäjän kirjautuneena, jos tämän session arvon on asetettu.

```

if(!isset($_SESSION['userData'])){
    include "../pages/login.php";
}
else{
    include "../pages/" . $_SESSION["currentPage"] . ".php";
}

```

Kuva 7. Index.php body-elementin sisältö.

Currentpage-muuttujan arvon on sama kuin pääsivujen tiedoston nimet. Pääsivun tiedostot säilytetään palvelimessa pages-nimisen kansion sisällä. Tämän toteutustyyliin avulla sivu voi helposti ladata oikeat tiedostot pages-kansiosta. Currentpage-arvoa alustetaan ja asetetaan pääsivujen alussa, jos käyttäjä on kirjautunut sivustoon.

Login.php-nimisiä tiedostoja on kaksi kappaletta jokaisessa sivustossa. Ne sijaitsevat pages- ja phpscript-kansiossa. Pages-kansion sisällä oleva login.php on sivu, joka näytetään, jos käyttäjät eivät ole kirjautuneet sivustolle. Kirjautumisen aikana se suorittaa phpscript-kansiossa olevan login.php-tiedoston. Phpscript-kansiossa oleva login.php tarkistaa annetut tiedot ja asettaa userdata-muuttujan arvot, jos käyttäjän antamat tiedot ovat oikeat. Sen jälkeen käyttäjiä ohjataan kotisivulle, jossa asetetaan currentpage-muuttujan arvo. Kirjautumisessa käytetään PHP password\_verify-funktiota. Salasanan kryptaamisessa käytetään PHP:n hash- ja salt-funktioita. Kuva 8 esittää pages-kansiossa olevan login.php:n sisältöä. Jos käyttäjä ei ole kirjautunut sivustoon, index.php näyttää tämän sisällön body-elementin sisällä.

```

<div class="container">
  <form class="loginForm form-login" method="post" action="./phpScript/login.php">
    <h2>Log in to Kehokartta</h2>
    <label for="inputEmail" class="sr-only">Log in to Kehokartta</label>
    <input id="inputEmail" name="username" class="form-control" placeholder="username" required="" autofocus="">
    <label for="inputPassword" class="sr-only">password</label>
    <input type="password" id="inputPassword" name="password" class="form-control" placeholder="password" required=""><br />
    <button class="btn btn-lg btn-primary btn-block" type="submit">login</button>
  </form>
</div>

```

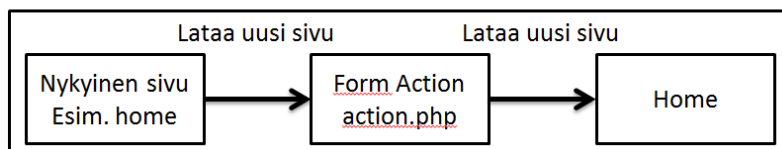
Kuva 8. Pages-kansion login.php-tiedoston sisältö.

### 3.1.2 JavaScript

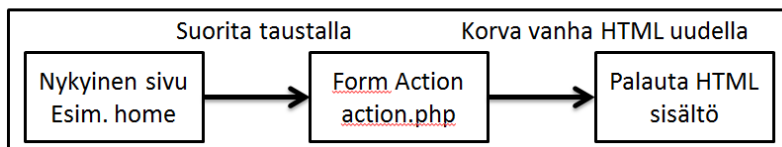
JavaScript.js-tiedoston sisällä on kaikki kyseisen sivuston JavaScript-funktiot. JavaScriptin avulla sivusto suorittaa monta HTML-elementtien muokkausta ilman sivun lataamista uudelleen. Kaikki JavaScript-funktiot sivustossa käyttävät melko samaa toteutusta HTML:n formin, buttonin ja linkkien käsittelyyn.

Kaikki sivuston HTML form -datat lähetetään post-menetelmällä. Post ei säilytä väliaikattiedostoja selaimessa toisin kuin get, joten se on paljon parempi henkilökohtaisten ja tärkeiden tietojen käsittelyyn. Post action suoritetaan käyttäen JQuery-kirjaston ajax-funktiota. Tämän menetelmän avulla post action suoritetaan nykyisen sivun taustalla, ja käyttäjä voi jatkaa suoraan siitä, mihin hän jäi lähetettyään lomakkeen. Kuva 9 näyttää normaalin HTML form -suorituksen ja JQuery ajax-funktion erot. Normaalisissa form-suorituksessa selain lataa action-tiedoston, ja suorituksen jälkeen selain menee action-tiedoston ohjaamalle sivulle. JQuery taas suorittaa action-tiedoston nykyisen sivun taustalla ja korvaa vanhan HTML-elementtiin action-tiedoston palauttamilla.

## Normaali



## Ajax



Kuva 9. Normaali ja JQuery-kirjaston HTML form -suoritus.

```

jQuery(document).ready(function ($) {
  $(document).on('submit', '.loginForm', function(e) {
    e.preventDefault();
    $.ajax({
      url: $(this).attr('action'),
      type: $(this).attr('method'),
      data: $(this).serialize(),
      success: function(html) {
        $("div.container").replaceWith(html);
      }
    });
  });
});

```

Kuva 10. JavaScript-funktion HTML-formille.

Kuvan 10 JavaScript-funktiota suoritetaan, kun käyttäjä painaa kuvan 8 nappia. Funktio saa suoritettavan tiedoston nimien kuvan 8 action-muuttujan arvosta. Funktio korvaa suorituksen jälkeen container-elementin sisällön suoritettuaan tiedoston palauttamalla sisällöllä.

Kaikki sivuston napit eivät ole HTML formin sisällä. Sivustossa on melko paljon nappeja, jotka muokkaavat HTML-elementtien sisältöä ilman PHP:n funktiota. Esimerkiksi jos käyttäjä haluaa lisätä uutta mittauksia mutta seuraava lisäyspäivämäärä on toistojakson mukaan päivän päästä. Hän painaa kyseisen syöttökentän vieressä olevaa nappia, joka poistaa disabled-parametrin syöttökentästä. Kuva 11 esittää edelliseen esimerkin muutoksen HTML-elementissä. Monet sivuston esittämät arvot ovat HTML input-elementin sisällä. Jos kyseisen syöttökentän ei ole tarkoitus olla käytettävissä, sillä on disabled-parametrina. Disabled-parametri estää käyttäjän muuttamasta syöttökentän sisältöä.



Kuva 11. Esimerkki HTML-elementin muutoksesta.

Kuvan 11 toiminta toteutetaan JavaScriptin avulla. Napilla on HTML onclick-muuttuja, jossa sille määrätään JavaScript-funktio. HTML onclick-arvona on käytettävän JavaScript-funktion nimi ja sen tarvitsemat parametrit. JavaScript-funktio etsii oikean HTML-elementin ja poistaa sen disabled-parametrin.

```
<button type="button" class="btn btn-primary buttonWithInput" onclick="forceNewEntry(event)">enable</button>
function forceNewEntry(event) {
  event.preventDefault();
  $(event.target.parentNode).find('input').each(function() {
    $(this).prop("disabled", false);
  });
  $(event.target).replaceWith('<button type="button submit" class="btn btn-primary buttonWithInput" id="submit">submit</button>');
}
```

Kuva 12. Napin HTML-elementti ja sen JavaScript-funktio.

Kuva 12 esittää kuvan 11 enable-napin HTML-elementin ja sen JavaScript-funktion. JavaScript-funktio etsii HTML-elementin, poistaa sen disabled-parametrin ja korvaa painetun napin uudella submit-napilla.

Kaikki JavaScript-funktiot, jotka HTML-elementit käyttävät, tarvitsevat parametrina event. Event on HTML:n tapahtumien rajapinta ja sen avulla JavaScript voi muokata monia selaintoimintoja. Esimerkiksi event.preventDefault(); kuvassa 12 estää selaimen perustoiminnan.

Kaikki sivuston sisäiset HTML a-linkit toimivat samalla tavalla kuin napit ja formit. Kuva 13 esittää navigointipalkissa olevien linkkien rakennetta ja niiden JavaScript-funktioita. Niiden onclick-arvona on JavaScript-funktio, joka korvaa HTML-elementtien sisällöt. JavaScript-funktio hakee ajaxin avulla oikean sivun pages-kansiosta.

```
<a href="javascript:void(0);" onclick="pageGoTo(event, \'home\')">home</a>
function pageGoTo(event, page){
    event.preventDefault();
    $.ajax({
        url: "pages/" + page + ".php",
        type: "POST",
        data: "page=page",
        success: function(html) {
            $('div.container').replaceWith(html);
        }
    });
}
```

Kuva 13. Esimerkki linkkien HTML-elementistä ja JavaScript-funktiosta.

### 3.1.3 SQL-kyselyt

Jokaisella sivustolla on oma tiedosto SQL-kyselyitä varten. Niissä ovat kaikki kyseisen sivuston käyttämät SQL-kyselyt. SQL-kyselyn tiedosto on PHP-luokka, jonka phpscript-kansiossa olevat PHP-funktiot kutsuvat. Tarvittaviin kyselyihin päästään käsiksi kutsuamalla luokan funktiota sen luonnin ja alustuksen jälkeen. Luokka luo yhteyden tietokantaan alustuksessa.

Kuva 14 esittää yksinkertainen SQL-kyselyn tiedostorakenteen. Se on PHP-luokka nimeltä SQLConnector. Luokan alustuksessa luodaan yhteys tietokantaan. Luokalla on getUserById-niminen funktio, joka hakee käyttäjän taulusta käyttäjä annettun id:n avulla.

```
class SQLConnector{
    public function __construct(){
        try{
            $this->connection = new PDO("mysql:host=hostname;dbname=database", "username", "password");
        }catch (PDOException $error) {
            print "Error!: " . $error->getMessage() . "<br/>";
            die();
        }
    }
    public function getUserById($id){
        $sqlQuery = $this->connection->prepare('SELECT * FROM user WHERE id=?');
        $sqlQuery->execute(array($id));
        return $sqlQuery->fetchAll(PDO::FETCH_ASSOC);
    }
}
```

Kuva 14. Esimerkki SQL-kyselytiedostosta.

Kaikki sivuston SQL-kyselyn funktiot käyttävät PDO- ja strip\_tags-funktioita. Ne tarjoavat erittäin hyvän suojan SQL-injektion varalta. SQL-injektio on haitallinen hyökkäys sivustoa vastaan, joka käyttää SQL-kyselyitä syöttökentissä. PHP:n strip\_tags-funktio poistaa post-arvoista erikoismerkit, joita SQL-kyselyt käyttävät. PDO taas hyväksyy annetut arvot pelkästään muuttujina eikä osana SQL-kyselyitä.

### 3.2 Käyttäjän sivusto

Käyttäjän sivusto on tarkoitettu palvelun käyttäjille. Tämän sivuston avulla palvelun käyttäjät pääsevät käyttämään kaikki Kehokartan tarjoamat palvelut. Sivuston rakenne on jaettu kolmeen eri osaan. Ne ovat tilanne, arvot ja asetukset. Käyttäjän sivustossa on kuusi eri pääsivua: navigointipalkissa kotisivu nimeltä home, tilanne nimeltä status, päivän arvot nimeltä daily, viikon arvot nimeltä weekly, kuukauden arvot nimeltä monthly ja asetus nimeltä setting. Kotisivu ei ole käytössä kehityksen aikana, joten se on alustavasti tyhjä sivu.

Käyttäjän sivustossa on kehitysvaiheessa yksinkertainen rekisteröintisivu. Rekisteröintisivussa on alustavasti pelkästään sähköpostin ja salasanan syöttökentät. Sitten kun käyttäjä on syöttänyt sähköpostin ja salasanan, lähetetään hänelle sähköpostissa generoitu numerokoodi.

#### 3.2.1 Tilanne-sivu

Tilanne-sivussa ovat diagrammit palvelun käyttäjien mittaustuloksesta. Arvoja valitaan kahdesta pudotusvalikosta kuvan 2 esittämällä tavalla. Ensimmäiseksi käyttäjät valitsevat arvon kategorian, johon kuuluvat myös käyttäjien itse mittaamat arvot. Sen jälkeen käyttäjät valitsevat näytettävän arvon valitusta kategoriasta.

Kategorialista luodaan käyttäen SQL-kyselyä ja HTML select-elementtiä. Kuva 15 esittää kategorialistan HTML-rakennetta ja sen JavaScript-funktiota. Kategorialistan JavaScript-funktio määrittellään käyttäen HTML onchange -muuttujaa. Onchange-muuttuja suorittaa annetun JavaScript-funktion aina, kun käyttäjät valitsevat kategorian kategorialistasta. Kategorialistan arvoja luodaan käyttäen SQL-kyselyä. SQL-kysely hakee arvon-taulusta kaikki uniikit kategoriat käyttäen SQL:n group by -parametria. Kun SQL-kysely on suoritettu, se palauttaa taulun, jossa ovat kaikki kategoriat. Tämän

taulun ja PHP foreach -funktion avulla listataan kaikki kategoriat HTML select -elementin sisälle.

```

<div class="col-xs-12 col-sm-6 col-md-6 categoryList">
  <select class="form-control" id="sel1" onchange="selectCategory(event, value)">
    <option value="" style="display:none;" disabled selected>Variable category</option>
  <?php
    $category = $connection->getAllVariableCategory();
    foreach($category as $value){
      echo "<option>".$value['category']."</option>";
    }?>
  </select>
</div>
<div class="col-xs-12 col-sm-6 col-md-6 variableList"></div>

function selectCategory(event, value){
  event.preventDefault();
  $.ajax({
    url: "phpScript/selectCategory.php",
    type: "POST",
    data: "category="+value,
    success: function(html) {
      $('div.variableList').replaceWith(html);
    }
  });
}

```

Kuva 15. Kategorian listan HTML-rakenne ja sen JavaScript-funktio.

Arvolista toimii samalla tavalla kuin kategorialista. Kun käyttäjä on valinnut kategorian, kategorian onchange-muuttuja suorittaa sille annetun JavaScript-funktion. JavaScript-funktiossa on ajax-kysely, joka suorittaa PHP-funktion. PHP-funktio palauttaa HTML-elementin kaikki kyseisen kategorian kuuluvista arvoista kuvan 15 esittämällä tavalla. Arvolista onchange-muuttuja suorittaa JavaScript-funktion, joka tuo esille diagrammin käyttäen Flot-kirjastoa.

Kuva 16 esittää arvon mittaustulosten diagrammin koodit. Flot-diagrammin datat haetaan tietokannasta käyttäen ajax-kyselyä. Ajax-kysely suorittaa PHP-funktion, joka hakee annetun arvon id avulla mittaustuloksia tietokannasta. PHP-funktio palauttaa suorituksen jälkeen datat JavaScript-funktiolle. Datat on pakattu json-formaattiin. Kun JavaScript-funktio on saanut PHP-funktiolta datat, se suorittaa kuvan 16 funktion, joka luo diagrammin annetulla datalla. Kuvan 16 esittämä funktio saa parametrina data-arvon alin ja ylin terveellinen tulos sekä diagrammin option-parametrit. Diagrammin option-parametrit muokkaavat diagrammin ulkoasua.

```

<div class="col-xs-12 col-sm-12 col-md-12 flot-container">
  <div class="flot-placeholder"></div>
</div>

function plotdiagram(data, min, max, options){
  $.plot($(".flot-placeholder"), [{
    label: "["+name+" "+min+" - "+max,
    data: data1,
    color: "green",
    threshold:[
      {below: min,color:"red"},
      {below: max,color:"green"},
      {below: 99999,color:"red"}
    ]
  }], options);
}

```

Kuva 16. Flot-diagrammin luontifunktio.

Kuva 16 esittää myös, miten diagrammissa muutetaan viivaan värit arvojen alin ja ylin terveellisen tuloksen avulla. Väriin muuttamisessa käytetään Flot-kirjaston Threshold-lisäosaa. Sen avulla diagrammi listaa kaikki maksimi 99999 ja ylin terveellinen -arvon välillä olevat tulokset punaisena, ylin ja alin terveellinen -arvon välillä olevat tulokset vihreänä ja kaikki tulokset alin terveellinen -arvon alla punaisena.

### 3.2.2 Arvon-sivut

Arvon-sivut ovat päivä nimeltä daily, viikko nimeltä weekly ja kuukausi nimeltä monthly. Näillä sivuilla on yhteinen rakenne. Niiden ainoat erot ovat päivämäärän tarkistus ja mitä arvoja kyseiset sivut näyttävät. Arvon-sivut koostuvat mittaustulosten syöttökentistä ja mittaustulosten tuloslistasta.

Arvoissa näytetään kyseisen sivun aktivoituneet arvot. Näitä arvoja haetaan asetuksista ja arvon tietokantatauluista. Kuva 17 esittää SQL-kyselyä, joka hakee kyseiset taulut tietokannasta. SQL-kysely käyttää käyttäjän id-solua ja arvojen type-solua parametrina. Kysely hakee listan arvoista, jotka käyttäjät ovat aktivoineet arvon-sivuissa. Asetus-taulusta näkee, mitkä arvot käyttäjät ovat aktivoineet ja arvon-taulusta saa kyseisen arvon tiedot. SQL-kyselylle annetaan käyttäjän id:n lisäksi type parametrina. Typen avulla erotetaan päivä-, viikko- ja kuukausi-sivujen arvot toisistaan.

Kuvan 17 SQL-kysely liittää arvon -taulu asetus-tauluun käyttäen SQL:n left join -parametria. Se tuottaa taulu, joka listaa kaikki kyseisen käyttäjän asetus-taulun rivit type-muuttujan mukaan. Jokainen rivi kyseisessä taulussa on liitetty data-arvon taulusta. Tuloksena on taulu, jossa on soluja sekä asetus-taulusta että arvon-taulusta.



```

$sqlQuery = $this->connection->prepare('
SELECT
    usexvars.iteration, variable.name,
    variable.id, variable.unit
FROM usexvars
LEFT JOIN variable ON usexvars.varsid=variable.id
WHERE usexvars.userid=? AND usexvars.type=?');
$sqlQuery->execute(array($userId, $type));
return $sqlQuery->fetchAll(PDO::FETCH_ASSOC);

```

Kuva 17. SQL-kysely.

Kun ollaan saatu lista kaikista kyseiseen sivun arvoista, ne käydään läpi PHP:n foreach-operaatiolla. Jokaisessa arvossa tarkistetaan, mikä sen viimeisiin lisäyspäivämäärä on ja lasketaan sen seuraava lisäyspäivämäärä. Tarkistuksen avulla käyttäjille esitellään oikeat syöttökentät. Syöttökentät ovat joko pois käytöstä disabled-parametrilla tai käytettävissä. Sen tila riippuu seuraavasta lisäyspäivämäärästä. Tarkistus suoritetaan kuvan 18 esittämällä tavalla. Kuvassa 18 käytetään kahta eri PHP:n DateTime-muuttujan arvoa. Ne ovat tämänhetkinen aika nimeltä timenow ja seuraavan lisäys aika nimeltä nexttime. Nexttime-muuttujan arvoa saa lisäämällä SQL-taulusta iteration-solun arvon viimeisiin lisättyyn arvon päivämäärään. If-lause tarkistaa, onko timenow-muuttujan arvo isompi tai saman päivä / viikko / kuukausi kuin nexttime-muuttujan arvo. Jos se on isompi tai sama, niin seuraava lisäyspäivämäärä on nyt tai mennyt.

```

$timeNow = new DateTime(date("Y-m-d", $time));
$nextTime = new DateTime(date("Y-m-d", strtotime("+". $value['iteration']." days", $latest[0]['date'])));
if(
    $timeNow > $nextTime ||
    ($timeNow->format('Y') == $nextTime->format('Y') &&
     $timeNow->format('m') == $nextTime->format('m') &&
     $timeNow->format('d') == $nextTime->format('d'))
)
){

```

Kuva 18. Aikatarkistus PHP:n DateTime-muuttujan avulla.

Arvon-sivun lopussa on lista mittaustulosten historiasta. Käyttäjät voivat tämän listan avulla muuttaa vanhoja mittaustuloksia. Käyttäjät valitsevat näytettävän arvon pudotusvalikosta. Pudotusvalikossa ovat kaikki kyseiseen sivuun kuuluvista arvoista. Sitten, kun käyttäjä on valinnut arvon, esille tulee kyseisen arvon mittaustuloshistoria. Jokaisen mittaustulosmerkinnän vieressä on nappi, jota painamalla käyttäjä voi muuttaa kyseistä mittaustulosta. Pudotusvalikolla on samanlainen onchange kuin status-sivun pudotusvalikolla. Muutos syöttökenttä ja napin toiminnot on esitetty kuvassa 19.

Kuva 19 esittää historialistan ja change-napin JavaScript-funktion. Kyseinen funktio suorittaa PHP-funktion, joka palauttaa HTML-elementin, jossa on syöttökenttä ja nappi. JavaScript korvaa pudotusvalikon vieressä olevan tyhjän HTML-elementin sisällön PHP-funktion palauttamalla. Kuvassa 19 näkyy syöttökenttä ja sen nappi pudotusvalikon vieressä.

| id | variable name | date(y.m.d) | value | change                                |
|----|---------------|-------------|-------|---------------------------------------|
| 1  | steps         | 15.04.13    | 9785  | <input type="button" value="change"/> |
| 2  | steps         | 15.04.12    | 11111 | <input type="button" value="change"/> |

```
function changeResultButton(event, id, nro, name, value, date){
    event.preventDefault();
    $.ajax({
        url: "phpScript/changeResultButton.php",
        type: "POST",
        data: "edit=false&id="+id+"&nro="+nro+"&name="+name+"&value="+value+"&date="+date,
        success: function(html) {
            $('#div.changeResultContainer').replaceWith(html);
        }
    });
}
```

Kuva 19. Arvon historialista ja sen napin JavaScript-funktio.

### 3.2.3 Asetus-sivu

Asetus-sivussa käyttäjät voivat asettaa arvot aktiiviseksi, asettaa arvojen toistojaksot ja muuttaa omien tilien tiedot. Asetus-sivun arvoja haetaan arvon-taulusta ja asetus-taulusta kuvan 17 esittämällä tavalla. Left join -parametrin sijaan tämä SQL-kysely käyttää right join -parametria. Tämä kysely hakee kaikki arvot arvon-taulusta ja liittää jokaiselle arvon-taulun riville asetus-taulusta rivin linkitetyn id-solun avulla. Jos SQL-kysely ei löydä asetus-taulun rivejä joihinkin arvon id:ihin, niiden solut ovat tyhjiä SQL-kyselyn palauttamassa taulussa. SQL-kyselyn palauttama taulu käydään läpi käyttäen PHP:n foreachiä. Kyseinen arvo on aktiivinen, jos sen SQL-kyselyn palauttaman taulun rivissä on asetus-taulun soluja. Jokaisen arvon HTML-elementin sisältö riippuu siitä, onko se aktivoitunut vai ei, jos arvo ei ole aktivoitunut, niin sen syöttökentässä on disabled-parametri kuvan 3 mukaisesti.

Arvon aktivoiminen lisää uuden rivin asetus-tauluun tietokannassa. Aktivoimiseen tarkoitettu syöttökenttä ja nappi ovat HTML form -elementin sisällä. HTML formilla on kuvan 10 tyylinen JavaScript-funktio, joka suorittaa PHP-funktion, kun käyttäjät aktivoivat arvoa. PHP-funktio lisää uuden rivin asetus-tauluun. Lisättyään uuden rivin PHP-funktio

palauttaa HTML-elementin, joka näyttää kyseisen arvon aktivoituneena. Uusi HTML:n sisältö vaihdetaan vanhan tilalle käyttäen JavaScript `replacewith`ia kuvan 10 esittämällä tavalla.

Käyttäjän sähköposti- ja salasananakentät toimivat käyttäen sivustoa kuvan 10 esittämällä funktiolla. HTML form suorittaa action-muuttujan PHP-funktion tiedoston käyttäen JavaScript `ajax`ia. Kun PHP-funktio on suoritettu, JavaScript korvaa nykyisen HTML-sisällön PHP-funktion palauttamalla sisällön.

### 3.3 Laboratorion sivusto

Laboratorion sivusto on tarkoitettu laboratorion käyttäjille. Tämän sivuston avulla laboratorionkäyttäjät voivat lisätä uutta mittaustulosta palvelun käyttäjille. Laboratorion sivustossa on kolme pääsivua navigointipalkissa. Ne ovat käyttäjälista nimeltä `user`, tulos nimeltä `result` ja asetukset nimeltä `setting`.

Asetus-sivussa on pelkästään laboratorion käyttäjien tilinhallinta. Laboratorion käyttäjät voivat tämän sivun avulla muuttua heidän sähköpostinsa ja salasansa. Asetus-sivun rakenne on sama kuin käyttäjän sivuston asetussivu ilman arvojen aktivoinnin tarkoitettua osiota. SQL-kyselyt kohdistuvat laboratorion käyttäjien tauluun nimeltä `labuser`.

#### 3.3.1 Käyttäjälista-sivu

Käyttäjälista-sivussa on listattu kaikki palvelun käyttäjät kuvan 20 tyyliin. Tämä sivu koostuu kahdesta osasta, käyttäjelistasta ja sen valitsinpalkista. Kuvassa 20 näkyy käyttäjälista ja sen alapuolella oleva valitsinpalkki. Käyttäjälista näyttää vain 10 käyttäjää kerrallaan, joten se tarvitsee valitsinpalkin, jonka avulla laboratorion käyttäjät voivat valita esimerkiksi 1 - 10, 11 - 20, 21 - 30 ja niin edelleen käyttäjistä.

| id | username   | select user |
|----|------------|-------------|
| 1  | testUser1  | Select User |
| 2  | testUser2  | Select User |
| 3  | testUser3  | Select User |
| 25 | testUser23 | Select User |
| 26 | testUser4  | Select User |
| 27 | testUser5  | Select User |
| 28 | testUser6  | Select User |
| 29 | testUser7  | Select User |
| 30 | testUser8  | Select User |
| 31 | testUser9  | Select User |

| 1 | 2 | 3 |

Kuva 20. Laboratorion sivuston käyttäjälistasivu.

Käyttäjälstatiedot haetaan tietokannasta käyttäen SQL:n limit-parametria kyselyssä. Limit tarvitsee kaksi muuttujaa. Ensimmäinen muuttuja on, mistä haku aloitetaan. Toinen muuttuja on, kuinka monta riviä haetaan ensimmäisestä muuttujasta. SQL-kyselyn palauttaman taulun avulla sivu luo käyttäjälisan HTML table-elementin sisälle. Jokaisessa HTML table -rivissä on myös nappi, joka vie laboratorion käyttäjän tulossivulle ja näyttää valitun palvelun käyttäjän tiedot. Napin JavaScript-funktio suorittaa kaksi eri toimintoa. Ensinnäkin se luo session-arvon valitulle käyttäjälle. Session-arvossa on valittuun käyttäjään uniikki id. Sen jälkeen se käyttää kuvan 13 JavaScript-funktion ja vie laboratorion käyttäjän tulossivulle.

Kuva 21 esittää HTML table -rakennetta. Div elementin alussa on Bootstrap responsiivinen CSS -tyyli, joka on tarkoitettu HTML tablelle. Taulun otsikkosolut ovat thead-elementin sisällä ja sisältösolut ovat tbody-elementin sisällä. Jokainen PHP foreach -toisto luo uuden rivin sisältötaulussa.

```

<div class="table-responsive"><table class="table">
  <thead><tr>
    <th style="text-align: center;">id</th><th style="text-align: center;">username</th>
    <th style="text-align: center;">date</th><th style="text-align: center;">select user</th>
  </tr></thead>
  <tbody class="resultListBody" style="text-align: center;">
    <!-- PHP foreach loop -->
  </tbody>
</table></div>

```

Kuva 21. Käyttäjälistan HTML-rakenne.

Valitsinpalkilla joka kymmenes käyttäjästä voi valita, mitkä näytetään käyttäjälistassa. Se muuttaa ensimmäisen arvon, joka annetaan SQL limit -parametrille. Jokaisella valitsinpalkin napilla on oma onclick JavaScript -funktio. Kyseinen JavaScript-funktio suorittaa ajax-kyselyn, joka antaa PHP-funktiolle näytettävän aloitusnumeron. Esimerkiksi aloitusnumero on 1, jos näytetään 1 – 10 käyttäjistä, aloitusnumero on 2, jos näytetään 20 – 30 käyttäjistä ja niin edelleen. PHP-funktio suorittaa SQL-kyselyn annetulla aloitusnumerolla. Kun SQL-kysely on suoritettu, PHP-funktio palauttaa uuden käyttäjälistan. PHP-funktion sisältö on sama kuin peruskäyttäjälistan luonti eri SQL-kyselyn taululla.

### 3.3.2 Tulossivu

Tulossivun sisältö riippuu kahdesta asiasta. Onko PHP session selecteduser alustettu ja tullaanko sivulle HTML formin kautta? Jos molemmat ovat tyhjiä, niin sivu näyttää pelkästään käyttäjähakuosion. Molemmat tarkistetaan sivun alussa käyttäen PHP isset -funktioita.

Kuva 22 esittää tulossivun, jossa selecteduser-arvo on alustettu ja asetettu. Jos se ei ole alustettu, niin punaisella merkattu alue ei olisi näkyvässä. Punaisen alueen yläpuolella on käyttäjähaku tarkoitettuun syöttökenttään. Punaisen alueen vasemmalla on mittauksetulosten lisäyskenttä ja mittauksetulosten muutoskenttä. Punaisen alueen oikealla on mittauksetulosten historialista.

Search user

username

---

User: testUser1

New laboratory result

blood sugar cycle  fasting glucose

mg/dL

Change "12" data

99

| id | variable name | value | date(y.m.d-H:i) | change                                |
|----|---------------|-------|-----------------|---------------------------------------|
| 13 | calcium       | 9     | 15.02.16-13:28  | <input type="button" value="change"/> |
| 12 | calcium       | 99    | 15.02.16-13:28  | <input type="button" value="change"/> |
| 11 | calcium       | 13123 | 15.02.16-13:28  | <input type="button" value="change"/> |
| 10 | magnesium     | 111   | 15.02.13-17:38  | <input type="button" value="change"/> |
| 9  | calcium       | 112   | 15.02.13-17:00  | <input type="button" value="change"/> |
| 8  | magnesium     | 113   | 15.02.13-16:59  | <input type="button" value="change"/> |
| 6  | sodium        | 114   | 15.02.13-16:41  | <input type="button" value="change"/> |
| 5  | sodium        | 15    | 15.02.13-16:41  | <input type="button" value="change"/> |
| 4  | sodium        | 13    | 15.02.13-16:41  | <input type="button" value="change"/> |
| 3  | potassium     | 15    | 15.02.13-16:41  | <input type="button" value="change"/> |

| 1 | 2 |

Kuva 22. Tulossivu.

Hakukentän avulla laboratorion käyttäjät voivat helposti hakea palvelun käyttäjiä heidän numerokoodinsa avulla. Hakukenttä ja nappi ovat HTML formin sisällä. HTML form suorittaa tämän sivun PHP-tiedoston uudestaan antaen palvelun käyttäjän numerokoodin post-arvona.

Sivun koodissa hakukentän jälkeen on tarkistusvaihe. Ensimmäiseksi tarkistetaan, tullaanko sivulle HTML formin kautta käyttäen PHP isset -funktioita. PHP isset -funktio tarkistaa postin antaman käyttäjäkoodin arvon. Isset-funktio tarkistaa, onko arvoa alustettu ja palauttaa arvon tosi tai epätosi. Jos isset-funktio palauttaa arvon epätosi, niin sivu ei näytä mitään sisältöä. Jos isset-funktio palauttaa arvon tosi, niin sivu hakee annetun käyttäjäkoodin tietokannasta ja asettaa selecteduser-arvon haetuilla tiedoilla. Session tarkistus suoritetaan post tarkistuksen jälkeen. Session tarkistus toimii samalla tavalla kuin post-tarkistus. Uusi näytettävä näkymä koostuu kolmesta eri osasta, jotka näkyvät kuvassa 22. Ne ovat laboratoriomittaustuloslisäys, mittaustuloslista ja mittaustulosmuokkauskenttä.

Mittaustuloslisäysosan avulla laboratorion käyttäjät voivat lisätä uuden mittaustuloksen valitulle käyttäjälle. Lisättävä arvo valitaan kategorian ja arvon pudotusvalikon avulla. Pudotusvalikot toimivat samalla tavalla kuin palvelun käyttäjän sivuston statussivun pudotusvalikot. Sitten kun arvo on valittu, JavaScript tuo esille syöttökentän, jonka avulla lisätään uusi mittaustulos käyttäjälle. Lopullinen mittaustuloslisäysnäkyvä on kuvan 22 mukainen. Syöttökenttä ja napit ovat HTML form -elementtien sisällä. HTML formilla on kuvan 10 tyylinen JavaScript-funktio, joka suorittaa annetun action-tiedoston.

Mittaustuloslista näyttää kaikki laboratorion käyttäjän lisäämät mittaustulokset valitulle käyttäjälle. Mittaustuloslista toimii samalla tavalla kuin käyttäjälistasivu. Tuloslista SQL-kysely hakee labresult-taulusta oikeat mittaustulokset laboratorion käyttäjän ja valitun käyttäjän id:n avulla. Mittaustuloslistassa oleva change-nappi tuo esille mittaustulosmuokkauskentän. Tulosmuokkauskenttä on yksinkertainen HTML form -elementti syöttökentällä ja napilla. Se käyttää kuvan 10 tyyllisen JavaScript-funktion.

### 3.4 Hallintasivu

Hallintasivun avulla sivuston ylläpitäjä voi lisätä uuden mitattavan arvon, muuttaa arvojen tiedot, generoida uuden salasanan palvelun käyttäjille ja hallita laboratorion käyttäjien tunnukset. Hallintasivustossa on neljä pääsivua navigointipalkissa. Ne ovat käyttäjälista, laboratorion käyttäjät, arvot ja asetukset. Asetus-sivun rakenne ja toiminto on kehitysvaiheessa sama kuin laboratorion sivuston asetussivu.

Käyttäjälistasivun avulla ylläpitäjä voi generoida ja lähettää uudet salasanat palvelun käyttäjille. Tämä toiminta on tarkoitettu pelkästään hätätapausta varten. Sivun rakenne on sama kuin laboratorion user-sivun rakenne. Ainoana erona on käyttäjälistasivun olevan napin onclick JavaScript-funktio.

Laboratorion käyttäjäsivun avulla ylläpitäjä voi luoda uuden laboratoriotunnuksen ja muuttaa olemassa olevia tunnuksia. Tämä sivu koostuu kolmesta osasta. Ne ovat laboratorion käyttäjän lisäysalue, laboratorion käyttäjälista ja laboratorion käyttäjälistan valitsinpalkki. Kuva 23 esittää tämän sivun rakennetta. Yläpuolella on laboratorion käyttäjän lisäysalue, keskellä käyttäjälista ja alhaalla käyttäjälistan valitsinpalkki.

Uuden laboratorion käyttäjän lisäysalue koostuu käyttäjätunnussyöttökentästä, sähköpostin syöttökentästä ja submit-nappista. Ne ovat kaikki HTML form -elementin sisällä. HTML formilla on kuvan 10 tyylinen JavaScript-funktio. Kun ylläpitäjä luo uuden tunnuksen sivu suorittaa PHP-funktion. PHP-funktio generoi uudelle laboratorion käyttäjälle salasanan ja lähettää tiedotteen tunnuksen luonnista annetulle sähköpostille.

Add new laboratory user

username  email

---

Laboratory users

|          |                                       |                   |                                       |
|----------|---------------------------------------|-------------------|---------------------------------------|
| labUser1 | <input type="button" value="change"/> | lab@user.com      | <input type="button" value="change"/> |
| labUser2 | <input type="button" value="change"/> | lab@user.com      | <input type="button" value="change"/> |
| lab      | <input type="button" value="change"/> | lab@metropolia.fi | <input type="button" value="change"/> |
| labtest  | <input type="button" value="change"/> | asdf@asdf.com     | <input type="button" value="change"/> |
| labtest1 | <input type="button" value="change"/> | asdf@asdf.com     | <input type="button" value="change"/> |
| labtest2 | <input type="button" value="change"/> | asdf@asdf.com     | <input type="button" value="change"/> |

| 1 |

Kuva 23. Laboratorion käyttäjän sivu.

Käyttäjälista toimii samalla tavalla kuin laboratorion sivuston käyttäjälistasivu. Se listaa kymmenen käyttäjää kerralla. HTML tablen sijaan foreach-toistossa luodaan kaksi HTML form -elementtiä kuvan 23 tyyliä. Vasemmalla olevalla formilla muutetaan kyseisen laboratorion käyttäjän käyttäjätunnusta. Oikealla olevalla formilla muutetaan kyseisen laboratorion käyttäjän sähköposti. Molemmilla formilla on kuvan 10 tyylinen JavaScript-funktio. Syöttökentät on oletuksena poistettu käytöstä. Kun ylläpitäjä haluaa muuttaa laboratorion käyttäjien tiedot, hän painaa kyseisen syöttökentän vieressä olevaa change-nappia. Change-napilla on kuvan 12 tyylinen JavaScript-funktio, joka poistaa viereisestä syöttökentästä disabled-parametrin.

Arvon-sivun avulla ylläpitäjä voi lisätä uudet mitattavat arvot palveluun ja muuttaa vanhojen arvojen tiedot. Sivussa on uuden arvon syöttökenttä ja pudotusvalikko kaikkien arvojen kategoriasta. Kun käyttäjä valitsee kategorian pudotusvalikosta, esille tulee lista kaikista kyseisen kategorian arvoista.

Kuva 24 esittää arvon-sivun näkymää, jossa kategorian pudotusvalikossa on kategoria valittuna. Sivun ylhäällä on uuden arvon syöttökenttä, keskellä on kategorian pudotusvalikko ja alhaalla lista kaikista valitun kategorian arvoista. Uuden arvon syöttökenttä ja muutoskentät ovat HTML form -elementin sisällä. Formilla on kuvan 10 tyylinen JavaScript-funktio. Arvojen muutoskentät aktivoidaan painamalla niiden vieressä olevaa change-nappia, joka poistaa syöttökentän disabled-parametrin. Kategorian pudotusvalikon onchange-muuttujan JavaScript-funktio suorittaa PHP-funktion. PHP-funktio ha-



kee kaikki kyseisen kategorian arvot tietokannasta ja palauttaa HTML-elementin, jossa ovat kaikki kyseisen kategorian arvot. HTML-elementti luodaan käyttäen PHP foreachia. Jokaisessa foreach-toistossa luodaan HTML form-elementti, joka sisältää kaikki arvon tiedot syöttökentän sisällä.

Add new variable

|                                       |                         |                         |
|---------------------------------------|-------------------------|-------------------------|
| Variable name                         | Variable good range min | Variable good range max |
| Variable unit                         | Variable Category       |                         |
| <input type="submit" value="submit"/> |                         |                         |

Variables

electrolytes

|           |                                       |              |      |                                       |
|-----------|---------------------------------------|--------------|------|---------------------------------------|
| calcium   | <input type="button" value="change"/> | 8.6          | 10.5 | <input type="button" value="change"/> |
| mg/dL     | <input type="button" value="change"/> | electrolytes |      | <input type="button" value="change"/> |
| magnesium | <input type="button" value="change"/> | 1.8          | 2.4  | <input type="button" value="change"/> |
| mg/dL     | <input type="button" value="change"/> | electrolytes |      | <input type="button" value="change"/> |
| potassium | <input type="button" value="change"/> | 3.5          | 5.1  | <input type="button" value="change"/> |
| mmol/L    | <input type="button" value="change"/> | electrolytes |      | <input type="button" value="change"/> |
| sodium    | <input type="button" value="change"/> | 136          | 145  | <input type="button" value="change"/> |
| mmol/L    | <input type="button" value="change"/> | electrolytes |      | <input type="button" value="change"/> |

Kuva 24. Arvon-sivu.

## 4 Jatkokehitys ja yhteenveto

Projektin tavoitteena oli luoda Kehokartta-palvelulle toiminnallinen sivusto, jossa palvelun käyttäjät voivat seurata monia terveyteen liittyviä arvoja. Projektissa luodaan myös laboratorion ja hallintasivuston palvelu tarjoajalle. Projektin lopputulos täytti kaikki tarvittavat vaatimukset ja sisältää monia lisäyksiä, jotka eivät olleet projektin suunnitelmassa alkuvaiheessa.

Kehitetyt sivustot tarjoavat erittäin hyvän pohjan jatkokehitykselle. Monet asiat sivustossa voi muokata erittäin helposti rikkomatta mitään muuta. JavaScript-funktiot ja PHP-funktiot ovat melko itsenäistä eivätkä riipu mistään muusta funktiosta, joten näiden muokkaus on erittäin helppoa ja turvallista. CSS:n tyyliluokat ovat sivustolla yhteisiä. Näitä muokkaamalla saa helposti yhtenäisen näkymän kaikista kolmesta sivustosta.

Sivustolla on vielä monia asioita kehitettävänä. Ulkoasu on suurin näistä asioista. Tavoitteena oli luoda toimiva sivusto, joten ulkoasun suunnittelu ja toteutus jäivät vähäiseksi. Suunnitelmana oli projektin alussa saada suunnittelija, joka luo hyvän ulkoasu-teeman iPhone-sovellukselle ja nettisivulle, mutta se ei onnistunut tämän projektin aikana. Sivuston tuomaa raskautta palvelimelle voi myös vähentää. Monet funktiot voidaan yhdistää lopullisessa versiossa. SQL-kyselyitä voidaan myös optimoida paremmin hakemaalla pelkästään soluja, joita tarvitaan. Monet optimoinnit kannattaa suorittaa vasta julkaisun loppuvaiheessa, joten kehityksen aikana on keskitytty vain toimintojen toteuttamiseen.

## Lähteet

- 1 PHP Manual. 2015. Verkkodokumentti. the PHP Documentation Group. <php.net/manual>. Luettu 4.03.2015.
- 2 JavaScript. 2015. Verkkodokumentti. Mozilla Developer Network and individual contributors. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. Luettu 12.03.2015.
- 3 HTML5. 2015. Verkkodokumentti. Mozilla Developer Network and individual contributors. <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>. Luettu 12.03.2015.
- 4 JQuery API. 2015. Verkkodokumentti. The JQuery Foundation. <api.jquery.com>. Luettu 19.03.2015.
- 5 Bootstrap. 2015. Verkkodokumentti. Bootstrap contributors. <http://getbootstrap.com> Luettu 19.03.2015.
- 6 Flot. 2015. Verkkodokumentti. IOLA and Ole Leursen. <www.flotcharts.org>. Luettu 19.03.2015.