

Juha Larmia

Web-sovellus mehiläispesien keräämiseen ja tarkasteluun

vaakadatan

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikka

Insinöörityö

6.5.2013

Tekijä(t) Otsikko Sivumäärä Aika	Juha Larmia Web-sovellus mehiläispesien vaakadatan keräämiseen ja tarkasteluun 47 sivua + 2 liitettä 6.5.2013
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Ohjelmistotekniikka
Ohjaaja(t)	Lehtori Vesa Ollikainen Toiminnanohjaaja Heikki Vartiainen
Opinnäytetyön tarkoituksena oli kehittää web-sovellus mehiläispesien vaakadatan keräämiseen ja tarkasteluun. Tällaisesta web-sovelluksesta olisi paljon hyötyä mehiläistarhaajille eri puolilta Suomea. Web-sovelluksen tulisi olla helppokäyttöinen mehiläistarhaajille sekä mahdollisimman helposti ylläpidettävissä myös tietotekniikkaa vähemmän osaavan toimesta. Työn lopputuloksena saatiin Suomen Mehiläishoitajain Liitolle käyttövalmis vaakapesäsovellus. Opinnäytetyöraportissa käydään yksityiskohtaisesti läpi tätä opinnäytetyönä syntynyttä vaakapesäsovellusta sovelluskehittäjän näkökulmasta. Opinnäytetyöraportissa tutustutaan ensin vaakapesäsovelluksessa käytettyihin tekniikoihin HTML, JavaScript, CSS, PHP ja MySQL. Raportissa käydään tämän jälkeen läpi sovelluksen ja tietokannan rakenne, perehdytään sovelluksessa käytettyyn Highcharts-kirjastoon, selvitetään kuntakartasta muodostunutta haastetta sekä tutustutaan automaattivaakapesiä varten suunniteltuun tietoa vastaanottavaan lisäosaan. Raportissa tutkitaan myös vaakapesäsovellukseen liittyviä tietoturvariskejä (SQL-injektio, Cross Site Scripting yms.) sekä selvitetään keinoja tehdä sovelluksesta turvallisempi. Lopuksi raportissa pohditaan työn lopputulosta suhteessa asetettuihin tavoitteisiin. Lisäksi ehdotetaan jatkokehitysideoita, joilla sovellusta voisi vielä parantaa.	
Avainsanat	mehiläispesä, PHP, JavaScript, MySQL, Highcharts

Author(s) Title	Juha Larmia Web application for recording and observing beehive weights
Number of Pages Date	47 pages + 2 appendices 6 May 2013
Degree	Bachelor of Engineering
Degree Programme	Information and Communications Technology
Specialisation option	Software Engineering
Instructor(s)	Vesa Ollikainen, Senior Lecturer Heikki Vartiainen, Executive Director
The purpose of this bachelor's thesis was to create a web application for recording and observing beehive weights. This kind of web application would be very beneficial for beekeepers in Finland. The targets for the web application are user friendliness for the regular users and easy maintainability for the admin users who are not that familiar with IT. The outcome of this bachelor's thesis was a web application for the designed purpose ready for use by Finnish Beekeepers' Association. The thesis report contains detailed description of the web application from software developer's point of view. This thesis report will first introduce the different web technologies used in the web application HTML, JavaScript, CSS, PHP and MySQL. Next, the structures of the application and database are reviewed followed by the study of Highcharts library used for creating graphs. Moreover, the challenge of drawing a map of the municipalities of Finland is explored, as well as the solution for data receiving from automated beehive scales is described. The report then studies possible security vulnerabilities of the web application (SQL injection, Cross Site Scripting etc.) and proposes ways to improve the security. Finally, the end result is examined and compared to the targets for the application. In addition, ideas for further development of the application are proposed.	
Keywords	beehive, PHP, JavaScript, MySQL, Highcharts

Sisällys

Lyhenteet

1	Johdanto	1
2	Vaakapesäsovelluksen pääpiirteet	2
2.1	Mikä on vaakapesä?	2
2.2	Sovelluksen lähtökohdat ja päämäärät	2
2.3	Käytetyt tekniikat ja työkalut	3
2.3.1	HTML	4
2.3.2	JavaScript	5
2.3.3	CSS	7
2.3.4	PHP	8
2.3.5	MySQL	9
2.4	Sovelluksen rakenne	10
2.4.1	Käyttäjryhmät	13
2.4.2	Etusivu	15
2.4.3	Selaa vaakapesiä -sivu	17
2.4.4	Omat vaakapesät -sivu	18
2.4.5	Muokkaa vaakapesiä -sivu	18
2.4.6	Käyttäjät-sivu	19
2.4.7	Omat tiedot -sivu	20
2.4.8	Tietokannan luontiskriptit	20
2.5	Tietokannan rakenne	21
2.5.1	Kayttaja-taulu	22
2.5.2	Kunta-taulu	23
2.5.3	Pesa-taulu	24
2.5.4	Paino-taulu	24
2.5.5	Kasvi-taulu	26
2.5.6	Painokasvi-taulu	26
2.6	Highcharts-kirjasto kuvaajien piirtämiseen	27
2.7	Kuntakartta	30
2.8	Ohjelmallinen vaakapesädatan lähettäminen	33
2.9	Sovelluksen tietoturva	34
2.9.1	SQL-injektio	34
2.9.2	XSS (Cross Site Scripting)	38
2.9.3	Etätiedostojen käyttö	41

2.9.4	Tallennetut salasanat	42
3	Tulokset ja jatkokehitys	43
4	Yhteenveto	44
	Lähteet	46
	Liitteet	
	Liite 1. Vaakapesäseurannan käyttöohjeet	
	Liite 2. Ohjelmallinen vaakapesädatan lähettäminen	

Lyhenteet ja käsitteet

CSS	Cascading Style Sheets. Web-sivujen tyyliohje, jonka avulla voi muotoilla web-sivua monipuolisemmin kuin HTML:llä.
HTML	Hypertext Markup Language. Selainpäässä ajettava web-dokumentin kuvauskieli.
JavaScript	Selainpäässä ajettava komentosarjakieli, jolla voi luoda web-sivulle dynaamista sisältöä.
MD5	Message-Digest Algorithm 5. Tiivistealgoritmi, joka tuottaa mistä tahansa datasta 128-bittisen tiivisteen. Tämä tiiviste esitetään yleensä 32-merkkisenä heksakoodina, esim. "8c1788205b6abffb0c6a4a1f4b10395a".
MySQL	Relaatiotietokantaohjelmisto dynaamisen datan tallennukseen.
PHP	PHP: Hypertext Preprocessor. Palvelinpäässä ajettavaa web-sovelluskoodia.
Scrum	Ketterässä ohjelmistokehityksessä käytetty projektinhallinnan viitekehys.
Template	Web-sovelluskehityksessä käytetty näkymän mallipohja, joka haetaan tyypillisesti osaksi web-sivun näkymää.

1 Johdanto

Tämän opinnäytetyön tarkoituksena on kehittää web-sovellus tarhattujen mehiläispesien painon seurantaan. Mehiläispesien painonkehityksestä eli vaakadatasta on hyötyä mehiläistarhaajille hunajan määrän, satokauden alun ja lopun sekä kasvien kukinta-aikojen arvioimisessa. Näillä tiedoilla tarhaaja voi suunnitella tulevia mehiläispesiin tehtäviä hoitotoimenpiteitä paremmin. Lisäksi julkisesta vaakadatasta on myös vaa'attomille tarhaajille suurta hyötyä.

Tässä opinnäytetyöraportissa kuvataan yksityiskohtaisesti vaakapesäsovellusta sekä kerrotaan sovellukseen liittyvistä teknologioista. Vaakapesäsovellus sekä tämä opinnäytetyöraportti tehdään Suomen Mehiläishoitajain Liitolle.

Opinnäytetyön tavoitteena on saada valmiiksi helppokäyttöinen, turvallinen ja tyylikkään näköinen vaakapesäsovellus, jota eri puolella Suomea asuvat mehiläistarhaajat pystyvät käyttämään itsenäisesti ja jota pystyisi ylläpitämään tietotekniikkaa vähemmän osaavakin. Opinnäytetyöraportin tavoitteena on kuvata vaakapesäsovellusta yksityiskohtaisesti ja toimia tarvittaessa dokumentaationa toiselle ohjelmistotekniikka-alan osaajalle, mikäli vaakapesäsovellusta joskus ryhdytään jatkokehittämään.

Opinnäytetyössä käydään läpi ensin vaakapesäsovelluksessa käytettyjä web-tekniikoita sekä sovelluksen rakennetta ja tietokantaa. Tämän jälkeen tutustutaan kuvaajien piirtoon Highcharts-kirjaston avulla. Kuvaajien piirron jälkeen käsitellään erästä sovelluksen haastetta eli kuntakartan piirtämistä. Tämän jälkeen esitellään automaattivaakoja ja omia sovelluksia varten tehty lisäosa, joka vastaanottaa mittaustuloksia vaakapesäsovellukseen. Lopuksi tutkitaan vielä mahdollisia tietoturvariskejä vaakapesäsovelluksessa.

Opinnäytetyön liitteinä on kaksi ohjekirjaa, jotka tehdään vaakapesäsovelluksen käyttöä varten: vaakapesäsovelluksen käyttöohje sekä ohjelmallisen vaakadatan lähettämisen ohje. Nämä on valittu liitteiksi opinnäytetyöraporttiin siksi, että ne ovat tärkeä osa vaakapesäsovellusta.

2 Vaakapesäsovelluksen pääpiirteet

2.1 Mikä on vaakapesä?

Mehiläistarhaajien hoitamat mehiläispesät ovat useimmiten laatikoita, joiden sisällä on mehiläisten kennostoa. On olemassa sekä kiinteärakenteisia että irtorakenteisia mehiläispesäiä. Irtorakenteiset ovat selvästi yleisempiä niiden helppokäyttöisyyden ja muunneltavuuden takia. Irtorakenteiset mehiläispesät koostuvat laatikoista, jotka pinotaan päällekkäin. Laatikoita lisätään mehiläispesän kasvaessa, jotta mehiläisiltä ja ennen kaikkea hunajalta ei tila loppuisi kesken. Mehiläistarhaajien avuksi on kehitetty mehiläispesiin tarkoitettuja seurantavaakoja, joiden avulla mehiläispesän painoa voi mitata. Mehiläispesistä, joissa on tällainen vaaka, käytetään nimitystä vaakapesä. [1.]

Vaakapesän avulla saadaan tärkeää tietoa esimerkiksi satokauden alusta tai lopusta. Vaakapesän mittaustulosten pohjalta voi päätellä, milloin hunajan määrä on suurimmillaan eli milloin on sopivin aika kerätä hunajaa. Mehiläistarhaaja voi myös mittaustulosten perusteella arvioida, tarvitseeko mehiläispesälle tehdä jotain toimenpiteitä. Vielä enemmän vaakadatasta olisi hyötyä, jos mehiläistarhaajat, ja heistä erityisesti vaakapesättömät, voisivat seurata saman paikkakunnan tai saman seudun muiden mehiläispesien painonkehitystä sekä näihin tehtyjä toimenpiteitä. Mehiläistarhaajille suunniteltu vaakapesäsovellus helpottaa tätä seurantaa.

2.2 Sovelluksen lähtökohdat ja päämäärät

Suomen Mehiläishoitajain Liitto (SML) haki vuoden 2012 lopussa työharjoittelijaa/lopputyöntekijää vaakapesäprojektiinsa, ja minut valittiin tähän projektiin. SML:n tarkoituksena oli tehdä mehiläispesien painonkehityksen tallennukseen ja tarkasteluun web-pohjaisen sovelluksen. Vastaava sovellus oli jo olemassa MTT:n (Maa- ja elintarviketalouden tutkimuskeskus) kehittämänä ja se ehti olla yli 10 vuotta käytössä SML:n jäsenille. Erinäisistä syistä johtuen SML päätti kuitenkin ryhtyä kehittämään omaa web-sovellusta mehiläispesien vaakadatan keräämiseen. Ehdin muutaman päivän kokeilla MTT:n sovelluksen julkista eli kirjautumattomille käyttäjille näkyvää osaa, ennen kuin se laitettiin salasanan taakse.

Sovelluksen toteutusta suunnittelin SML:n toiminnanjohtajan Heikki Vartiaisen eli esimieheni opastuksella. Valmiissa sovelluksessa tulisi olla käyttäjiä eli näin ollen tarvitaan kirjautumismekanismi. Kirjautuneiden käyttäjien tulisi pystyä lisäämään järjestelmään vaakapesiä ja vaakapesiin niiden painolukemia. Lisäksi sovelluksessa tulisi olla julkinen puoli, jossa olisi mahdollista tarkastella sovellukseen kirjattuja painonkehityksiä sekä taulukkomuodossa että graafisesti. Minulla oli vapaat kädet valita käytettävät ohjelmointiteknologiat. Valitsin lopulta PHP:n, koska PHP oli minulle entuudestaan tuttu ja se on myös vaivaton ottaa käyttöön. Lisäksi PHP:n valintaa puolsi se seikka, että SML:n jäsenrekisteri oli tehty PHP:lla, joten vaakapesäsovelluksen pystyisi käyttöönottovaiheessa lisäämään jäsenrekisterin rinnalle samalle palvelimelle. Jäsenrekisteri käytti myös MySQL:ää, joten vaakapesäsovellukselle löytyisi näin valmis MySQL-palvelin, eikä ylimääräisiä palvelimia tarvitsisi vuokrata.

SML:n päämääränä vaakapesäprojektissa oli luoda helppokäyttöinen ja nopea web-sovellus vaakapesädatan keräämiseen ja tarkasteluun siten, että sovellus säilyisi käyttökelpoisena ainakin seuraavat kymmenen vuotta. Vaakapesäsovelluksen julkisen osan eli vaakapesätietojen selauksen on tarkoitus olla avoinna kaikille mehiläistarhaajille. Vaakapesien ja mittaustulosten lisäys on mahdollista vain käyttäjille, joilla on tunnukset järjestelmään. Tunnukset aiotaan alustavasti antaa vain heille, jotka ovat SML:n tai jonkin sen alayhdistyksen jäsenenä ja joilla on käytössään mehiläispesävaaka. MTT:n kehittämä vaakapesäsovellus oli toimiva ja selkeä, joten päätin ottaa siitä mallia.

2.3 Käytetyt tekniikat ja työkalut

Vaakapesäsovelluksessa hyödynsin PHP:ta, HTML:ää, CSS:ää, JavaScriptiä ja MySQL:ää. Editointityökaluna käytin Notepad++:aa sen keveyden ja helppokäyttöisyyden takia. Tietokantojen tutkimiseen ja muokkaukseen sovelluksen kehitysvaiheessa käytin komentorivillä ajettavaa mysql-nimistä työkalua sekä internet-selaimella ajettavaa phpMyAdmin-sovellusta. Kehitysvaiheen alkupuolella testasin sovellusta Metropolian opiskelijoille tarkoitettulla kotisivupalvelimella. Metropolian MySQL-palvelin toimi sovelluksen tietokantapalvelimena tässä vaiheessa. Loppuvaiheessa siirryin käyttämään TNetin palvelimia, joita tulisi käyttää myös sovelluksen julkaisun jälkeen. Sovelluksen testaukseen käytin internetselaimia

Google Chrome, Mozilla Firefox sekä Internet Explorer. Koodien siirtoon koneelta web-palvelimelle käytin WinSCP-tiedonsiirto-ohjelmaa.

Projektinhallinnassa hyödynsin ketteristä ohjelmistokehitysmenetelmistä tuttua Scrum-ohjelmistokehitystä. Scrumissa työskennellään ohjelmiston toiminnallisuuksia ja ominaisuuksia lisäävästi. Scrum koostuu monista lyhyehköistä kehitysjaksoista eli sprinteistä. Yksi sprintti on pituudeltaan yleensä 1-4 viikkoa. Jokaisen sprintin päätteeksi ohjelmistosta pitäisi olla valmiina toimiva ja mahdollisesti julkaisukelpoinen versio. Vaakapesäprojektissa tämä toimi käytännössä niin, että pidimme esimieheni kanssa seurantalavereja noin viikon välein. Palavereissa kävimme läpi, mitä olin saanut aikaan eli edellisen sprintin aikaansaannokset, sekä suunnittelimme tavoitteet seuraavaan sprinttiin. Vastasin yksin ohjelmiston suunnittelu- ja kehitystyöstä. [2.]

Tässä luvussa esittelen lyhyesti käyttämäni web-sovelluskehitystekniikat (HTML, JavaScript, CSS, PHP, MySQL) ja esittelen niiden käyttötapoja ja -tarkoituksia. Lisäksi jokaisessa alaluvussa olen käsitellyt kyseisen tekniikan merkitystä ja roolia vaakapesäsovelluksessa.

2.3.1 HTML

CERN:n työntekijä Tim Berners-Lee kehitti vuonna 1989 tiedon jakoon HTML:n (HyperText Markup Language). Vuonna 1990 Berners-Lee sai valmiiksi ensimmäisen prototyypin internet-selaimesta. [3.]

HTML 2.0 julkaistiin vuonna 1995. Ei kestänyt pitkään siitä, kun jo loppuvuodesta 1997 julkaistiin HTML:stä versio 4.0. Tällä hetkellä lähes jokaisen selaimen tukema HTML 4.01 on ylivoimaisesti suosituin HTML-versio web-kehityksessä, sillä siitä on olemassa valmis standardi. [4.]

Modernista HTML5:stä tuli ensimmäinen versio ulos vuonna 2008. HTML5 mahdollistaa muun muassa videopätkien toiston www-sivulla (HTML 4.01:ssä tarvitaan Flashia) sekä piirtoalueiden käytön. HTML5:n standardi on edelleen luonnosvaiheessa. [4.]

HTML on www-sivujen kuvauskieli, joka on avoimesti standardoitu. HTML:n avulla voidaan www-sivuille lisätä kuvia, hyperlinkkejä ja lomakkeita. Lisäksi HTML:n avulla

tekstiä ja sivun rakennetta voi muotoilla (tarkempaan muotoiluun CSS sopii tosin paremmin). [5.]

HTML-koodi aloitetaan `<html>`-tagilla ja lopetetaan `</html>`-tagilla. Näiden tagien sisällä on tyypillisesti jaoteltuna `<head>`-osio ja `<body>`-osio. `<head>`-osion sisälle tulevat HTML-sivun otsikko, HTML-dokumentin metatiedot sekä mahdolliset CSS-tyylimäärykset (tai erillisten CSS-tiedostojen lisäys). Itse HTML-sivun sisältö laitetaan puolestaan `<body>`-osion sisälle.

W3C (World Wide Web Consortium) tarjoaa helppokäyttöisen HTML-validaattorin. Validaattori tarkistaa, että koodissa ei ole virheitä tai määritysten vastaisuuksia. Jos huomautettavaa löytyy, validaattori kertoo, mitä on vialla sekä miltä riviltä vika löytyy. HTML-dokumentteja voi validoida osoitteessa <http://validator.w3.org/>.

HTML on web-ohjelmoinnin ydin: lähes kaiken web-sovelluskehityksen ulkokuorena on käyttäjälle näytettävä HTML-sivu. Vaakapesäsovelluksessakaan tämä ei ole poikkeus. HTML:llä saadaan aikaan vaakapesäsovelluksessa sivun rakenne, linkit, lomakkeet, taulukot sekä kuvien näyttö. Myös PHP:lla aikaansaattava dynaaminen data tulostetaan HTML-koodina sivulle.

2.3.2 JavaScript

JavaScript-kielen kehitti alun perin Netscape, ja se julkaistiin vuonna 1995. JavaScriptin nimi oli aluksi LiveScript, mutta se vaihdettiin JavaScriptiksi markkinointisyistä. JavaScriptiä ei tule sekoittaa Javaan: sillä ei ole juurikaan tekemistä Java-kielen kanssa. JavaScript on täysin asiakaspäässä (selaimella) ajettavaa ohjelmakoodia. [6.]

JavaScriptiä voi käyttää kolmella eri tavalla: tapahtumamääritteissä (event attributes), erillisen JavaScript-tiedoston ajamalla tai HTML-koodiin upotettuna `<script>`-tagien sisälle. Tapahtumamääritteet ovat HTML-kieleen lisättyjä toiminnallisuuksia, joilla esimerkiksi käyttäjän hiiren napsautuksiin ja liikutteluihin tietyssä elementissä voidaan reagoida. Esimerkki tällaisesta tapahtumamääritteestä: [7.]

```
<a onmouseover="javascript:alert('Testi')">Testi</a>
```

Äskeisen esimerkin mukainen ohjelmakoodi avaa popup-ikkunan "Testi"-viestillä, kun käyttäjä siirtää hiiren osoittimen Testi-tekstin päälle. Toinen tapa käyttää JavaScriptiä on syöttää sitä HTML-dokumenttiin <script>-tagien sisälle. Seuraavana on esimerkki tästä tavasta. Esimerkki tulostaa selaimeen tekstin "Hei!". [7.]

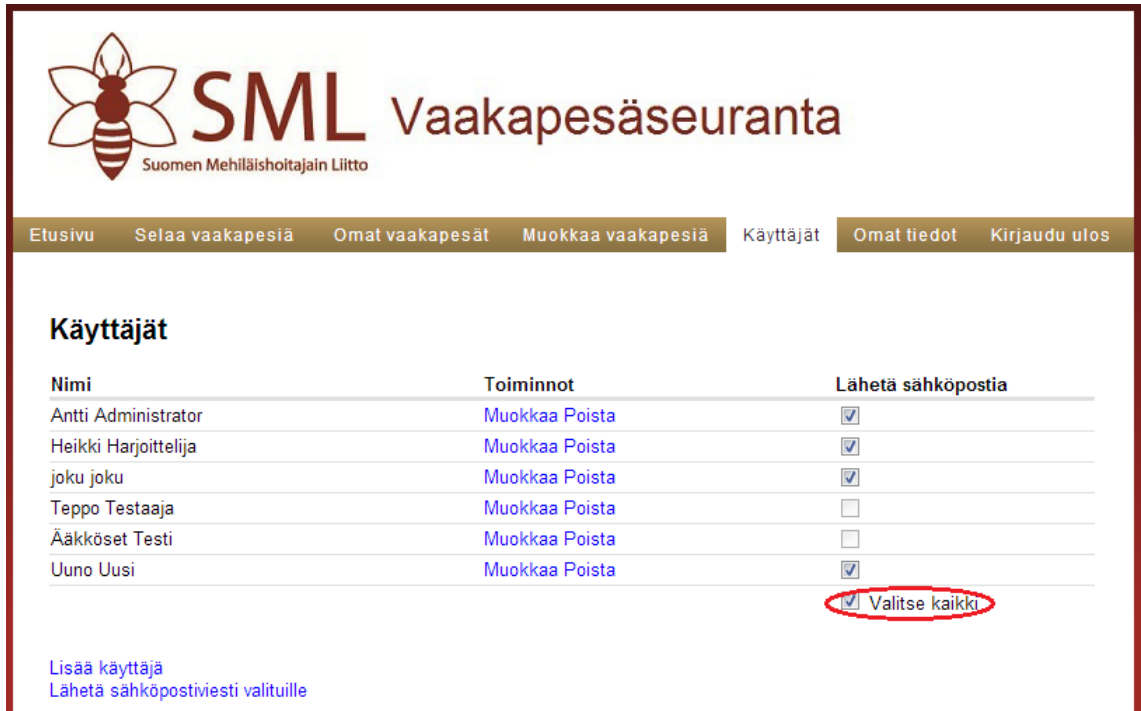
```
<script type="text/javascript">  
document.write("Hei!");  
</script>
```

Erillinen JavaScript-tiedosto voidaan ajaa HTML-dokumentissa seuraavanlaisella rivillä:

```
<script type="text/javascript" src="scriptit.js"></script>
```

Esimerkissä ajetaan scriptit.js-niminen JavaScript-tiedosto samasta hakemistosta kuin HTML-dokumentti. "Script src" eli lähdeosoite skriptille voi sijaita myös ulkoisella palvelimella. Tällöin annetaan halutun skriptin tarkka www-osoite. Tietoturvasyistä kannattaa kuitenkin käyttää vain paikallisia skriptejä (lisätietoa tästä tietoturvariskistä on luvussa 2.9.).

JavaScript on ollut vaakapesäsovelluksessa suuressa roolissa Highcharts-kuvaajanpiirtotyökalussa. En ole kuvaajanpiirtotyökalua itse tehnyt, mutta sen käytössä ja muokkauksessa oli eduksi JavaScript-osaamiseni. Lisäksi käytin JavaScriptiä sovelluksessa muun muassa "Lisätietoja pesästä" -popup-ikkunan avaamiseen sekä Käyttäjät-sivulla olevan "Valitse kaikki" -toiminnallisuuden toteuttamisessa (kuva 1).



SML Vaakapesäseuranta
Suomen Mehiläishoitajain Liitto

Etusivu Selaa vaakapesiä Omat vaakapesät Muokkaa vaakapesiä Käyttäjät Omat tiedot Kirjaudu ulos

Käyttäjät

Nimi	Toiminnot	Lähetä sähköpostia
Antti Administrator	Muokkaa Poista	☒
Heikki Harjoittelija	Muokkaa Poista	☒
joku joku	Muokkaa Poista	☒
Teppo Testaaja	Muokkaa Poista	☐
Ääkköset Testi	Muokkaa Poista	☐
Uuno Uusi	Muokkaa Poista	☒
		☒ Valitse kaikki

[Lisää käyttäjä](#)
[Lähetä sähköpostiviesti valituille](#)

Kuva 1. JavaScriptillä toteutettu "valitse kaikki" -toiminto

2.3.3 CSS

CSS (Cascading Style Sheets) on HTML-sivujen tyyliohje. CSS:n avulla hieman suurempien sivukokonaisuuksien muotoilut saadaan helposti yhtenäiseksi. CSS-tyyliohje on usein erillisessä .css-loppuisessa tiedostossa, mutta toisinaan CSS:ää käytetään myös suoraan HTML-sivuun upotettuna. CSS:llä määritellään, kuinka dokumentin eri osat käyttäjän selaimella näytetään. [8.]

CSS:ää voi käyttää kolmella eri tavalla: määrittelemällä HTML-koodissa halutun elementin sisälle erillinen style-parametri, syöttämällä CSS-koodia <style>-tageilla tai liittämällä dokumenttiin linkki erilliseen CSS-tyylitiedostoon. Elementin sisällä CSS-tyyli annetaan seuraavanlaisesti (lihavoitu osa on CSS-koodia): [9.]

```
<a href="http://www.google.com" style="{color:lightblue}">www.google.com</a>
```

Tämän tavan heikkous on, että elementin sisällä olevalla tyylin määrittelyllä ei voi enää nykyisellä CSS-versiolla käyttää pseudo-luokkia. Eräs esimerkki pseudo-luokasta on a:hover, joka määrittää tyylin siirrettäessä hiiri linkin päälle. Hieman parempi tapa määritellä tyyliä suoraan HTML-dokumentin sisällä on lisäämällä <head>-tagien

sisälle <style>-tagit, jonka sisällä sitten halutut tyyliohjeet annetaan. Seuraavassa on esimerkki tästä tavasta, jossa on hyödynnetty pseudo-luokkia: [9.]

```
<head>
<style>
a {color:lightblue}
a:visited {color:darkblue}
a:hover {color:pink}
</style>
</head>
<body>
<a href="http://www.google.com">www.google.com</a>
</body>
```

Paras vaihtoehto lisätä CSS-tyyliohjeita on luoda erillinen CSS-tiedosto tyylimäärittelyksille ja linkittämällä tämä tyylitiedosto haluttuihin HTML-dokumentteihin. Tällä tavalla web-sivuston tyylien muokkaus on helppoa, kun jokaista HTML-dokumenttia ei tarvitse muokata erikseen. Erillinen CSS-tyylitiedosto lisätään HTML-dokumenttiin laittamalla <head>-tagien sisälle seuraavanlainen rivi: [9.]

```
<head>
<link rel="stylesheet" href="tyylitiedosto.css"/>
</head>
```

W3C tarjoaa validaattorin myös CSS-dokumenteille osoitteessa <http://jigsaw.w3.org/css-validator/>. Validaattoriin syötetään CSS-dokumentin web-osoite, ja validaattori tarkistaa dokumentin virheiltä ja määrittysten vastaisuuksilta. Vaakapesäsovelluksen CSS-dokumentti on validoitu tällä validaattorilla.

Vaakapesäsovelluksen ulkoasu ja yksinkertaiset visuaaliset efektit (lähinnä edellä esitelty a:hover) on saatu aikaan CSS:llä. Minulle ei ollut entuudestaan kovin tuttua CSS:n käyttö, joten jouduin opettelemaan paljon asioita vaakapesäsovellusta tehdessäni. Eräs hyödylliseksi osoittautunut sivusto CSS:ään perehtymiseen (tai kertaukseen) löytyy osoitteesta: <http://www.w3schools.com/css/> (CSS Tutorial).

2.3.4 PHP

PHP (lyhenne sanoista PHP: Hypertext Preprocessor) julkaistiin vuonna 1995 [10]. PHP:llä tehdään palvelinpään koodia, eli esimerkiksi tietokantahaut ovat melko yksinkertaisia toteuttaa. PHP-sivut määritellään usein .php-loppuosalla

tiedostonimessä. PHP-sivuilla on yleensä sekaisin HTML:ää ja PHP:ta. Tarvittaessa PHP-sivut voi tehdä puhtaasti PHP:lla, jolloin kaikki HTML:n käyttö tehdään echo-komennoilla (tulostuskomento). PHP-koodin aloitustagi on `<?php>` ja lopetustagi on `<?>`. PHP-tagin sisällä olevaa koodia ajetaan palvelinpäässä. Tämä johtaa siihen, että mikäli palvelimella ei tapahdu jotain kriittistä virhettä, niin esimerkiksi tietokantapalvelimen tunnukset pysyvät turvassa PHP-tagien sisällä.

Vaakapesäsovelluksessa suurin osa logiikasta on ohjelmoitu PHP:tä käyttäen. PHP:n ansiosta sovelluksessa on muun muassa dynaamista dataa, lomakkeiden validointia ja käsittelyä, kirjautumistoiminto ja käyttäjäoikeuksien käsittelyä sekä sähköpostin lähetysoiminto. Dynaamista dataa mahdollistava MySQL:n käyttö web-sovelluksessa on tehty PHP:n tarjoamalla MySQL-rajapinnalla. Lisäksi sovelluksessa on etusivun dynaaminen (ylläpitäjien muokattavissa oleva) teksti toteutettu tiedoston luku- ja kirjoituskäskyillä, jotka PHP tarjoaa.

2.3.5 MySQL

MySQL (lyhenne sanoista My Structured Query Language) on avoimen lähdekoodin relaatiotietokantaohjelmisto. MySQL:n ensimmäinen versio julkaistiin sisäiseen käyttöön vuonna 1995. Ensimmäinen julkinen versio julkaistiin loppuvuodesta 1996. [11.]

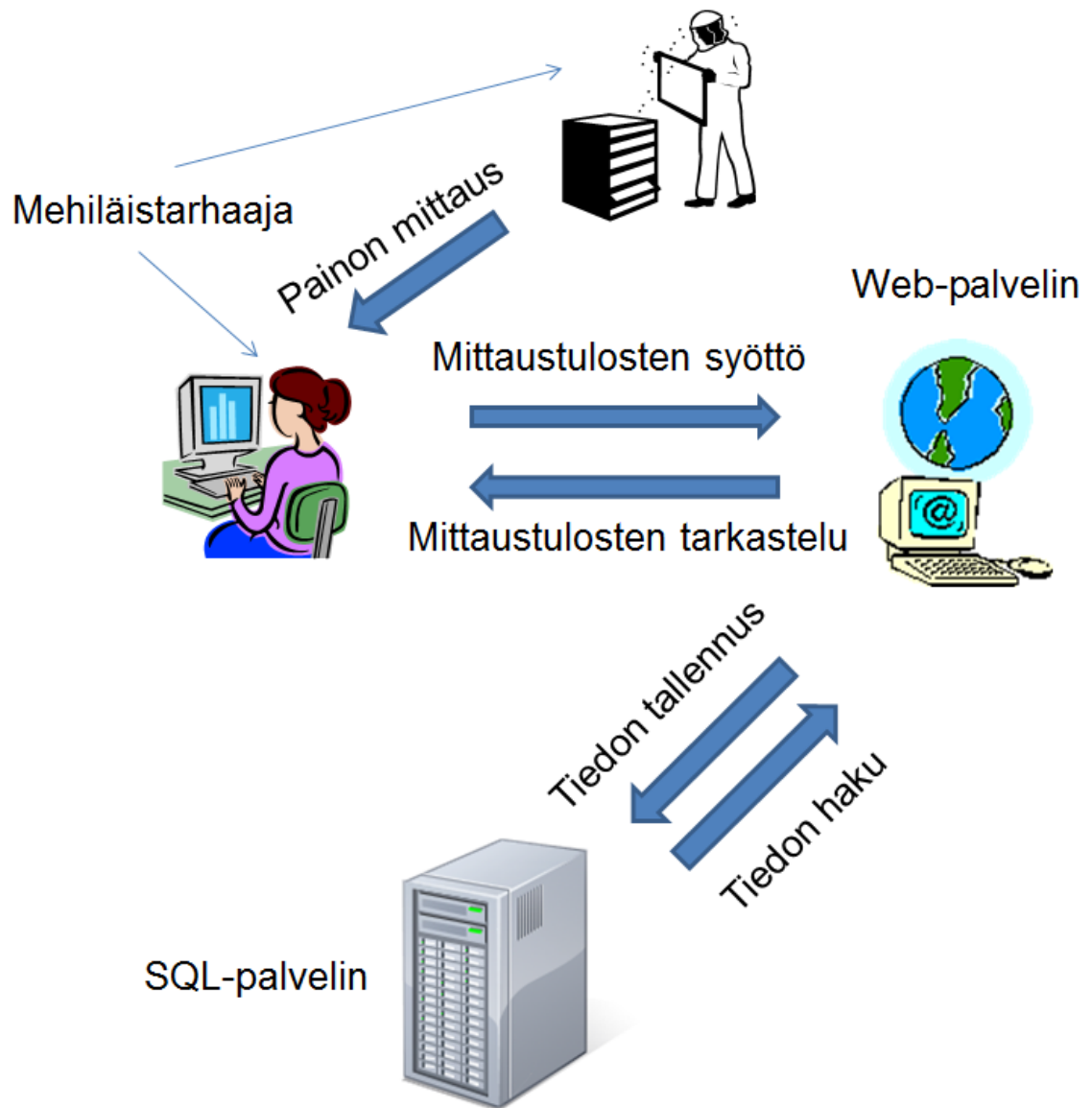
MySQL- tietokantoihin tehdään haku-, muokkaus- ja poisto-operaatioita SQL-lauseilla. MySQL-tietokanta koostuu tauluista, joihin määritellään luonnin yhteydessä kentät (tyypeiltään kokonaisluku, merkkijono, päivämäärä tms.), joista se koostuu. Luonnin jälkeen tauluihin voi lisätä tietoalkioita, kunhan tietoalkioon syöttää kaikkiin pakollisiin kenttiin arvot. Tietoalkioita voi lisäyksen jälkeen hakea, muokata tai poistaa. Taulujen välillä voi olla relaatioita, jotka määritellään taulujen luonnin yhteydessä vierasavaimilla (Foreign Key). Mikäli tarvitaan monta-moneen-relaatiota, täytyy luoda ylimääräinen yhteystaulu kahden taulun (tai tarvittaessa useammankin) välisille yhteyksille.

Vaakapesäsovelluksessa MySQL:ää käytetään PHP:n avulla jatkuvasti. MySQL mahdollistaa luotettavan, nopean ja turvallisen tavan tallentaa sovelluksen dynaamista dataa. Kaikki mittausdata, käyttäjien tiedot, vaakapesien tiedot sekä kuntien tiedot on tallennettu MySQL-tietokantaan, josta niitä haetaan www-sivulle näytettäväksi tai PHP-

logiikassa käsiteltäväksi. MySQL-tietokanta on suojassa ulkopuolisilta urkkijoilta, sillä siihen käsiksi pääseminen vaatii tietokantatunnuksia.

2.4 Sovelluksen rakenne

Vaakapesäsovelluksen voi ajatella olevan osa suurempaa kokonaisuutta, jonka muodostavat itse vaakapesä, mehiläistarhaaja, vaakapesäsovellus sekä SQL-palvelin (kuva 2). Mehiläistarhaaja lukee vaakapesästä painolukeman, palaa kotiinsa ja avaa tietokoneessaan internet-selaimella vaakapesäsovelluksen. Vaakapesäsovellukseen mehiläistarhaaja kirjautuu ja syöttää mittaustuloksen sekä mahdolliset omat havainnot. Vaakapesäsovellus käsittelee syötetyn datan ja välittää sen SQL-palvelimelle, jonne data tallentuu. Tämän jälkeen kuka tahansa sovelluksen käyttäjä voi katsella tätä mittaustulosta vaakapesäsovelluksella.



Kuva 2. Korkean tason kaavio sovelluksen käytöstä

Vaakapesäsovellus koostuu PHP-kooditiedostoista, muutamista kuvista, tyylitiedostosta sekä Highcharts-kirjaston JavaScript-tiedostoista (js-kansio). Kuva 3 esittää vaakapesäsovelluksen tiedostolistauksen. Kuvasta on poistettu tietoturvasyistä tietokantayhteyteen käytetty tiedosto.

js	19.4.2013 11:47	File folder	omatpesat.php	14.3.2013 12:50	PHP Script File	3 KB
banner.jpg	10.4.2013 13:30	JPEG image	omattiedot.php	16.4.2013 12:23	PHP Script File	7 KB
banneriSML.jpg	10.4.2013 13:51	JPEG image	painoform.php	10.4.2013 13:27	PHP Script File	15 KB
etusivu.txt	10.4.2013 14:10	Text Document	painoform2.php	10.4.2013 13:27	PHP Script File	14 KB
hgrad1.png	4.1.2013 13:45	PNG image	pesaform.php	10.4.2013 13:22	PHP Script File	7 KB
hgrad2.png	4.1.2013 13:45	PNG image	pesaform2.php	10.4.2013 13:22	PHP Script File	7 KB
index.php	10.4.2013 14:11	PHP Script File	poistakayttaja.php	13.3.2013 17:12	PHP Script File	2 KB
kayttajaform.php	16.4.2013 12:27	PHP Script File	poistapaino.php	13.3.2013 17:10	PHP Script File	3 KB
kayttajat.php	19.4.2013 11:35	PHP Script File	poistapaino2.php	13.3.2013 17:09	PHP Script File	2 KB
kunnet.txt	14.3.2013 12:16	Text Document	poistapesa.php	13.3.2013 17:09	PHP Script File	3 KB
lahetasahkopostia.php	19.4.2013 11:35	PHP Script File	poistapesa2.php	13.3.2013 17:08	PHP Script File	3 KB
lisaadata.php	8.3.2013 10:15	PHP Script File	restricted.php	5.3.2013 14:18	PHP Script File	2 KB
lisatiedot.php	10.4.2013 13:22	PHP Script File	selaa.php	9.4.2013 13:14	PHP Script File	12 KB
login.php	7.3.2013 20:55	PHP Script File	selaa2.php	9.4.2013 13:14	PHP Script File	11 KB
logout.php	9.1.2013 9:34	PHP Script File	SML.jpg	15.1.2013 20:50	JPEG image	39 KB
luontiskripti.php	14.3.2013 12:14	PHP Script File	style.css	16.4.2013 11:59	CSS Document	3 KB
luontiskripti2.php	14.3.2013 12:12	PHP Script File	suomi.jpg	3.1.2013 1:56	JPEG image	27 KB
luontiskripti3.php	14.3.2013 12:13	PHP Script File	vaakadata.php	25.3.2013 11:46	PHP Script File	5 KB
luontiskripti4.php	14.3.2013 12:13	PHP Script File	vaakadata2.php	25.3.2013 11:47	PHP Script File	5 KB
luontiskripti5.php	14.3.2013 12:13	PHP Script File	Vaakapesaseurannan kayttoohjeet.docx	10.4.2013 15:12	Microsoft Word D...	781 KB
muokkaaetusivua.php	10.4.2013 14:18	PHP Script File	Vaakapesaseurannan kayttoohjeet.pdf	10.4.2013 15:11	Adobe Acrobat D...	691 KB
muokkaapesia.php	14.3.2013 12:50	PHP Script File				

Kuva 3. Sovelluksen tiedostorakenne

PHP-kooditiedostot toimivat sovelluksessa kontrollereina sekä näkymänä, ja ne ovat sovelluksen ydintä. PHP-kooditiedostoista olen hahmotellut arkkitehtuurikuvan (kuva 4), josta selvenee, että minkä vaakapesäsovelluksen osaan liittyy mikäkin PHP-sivu. Olen sisentänyt arkkitehtuurikuvassa PHP-kooditiedostoja sen mukaan, millä hierarkiatasolla ne kyseisessä sovelluksen osassa ovat ja mihin hierarkisesti ylempään tiedostoon ne kuuluvat.

Etusivu

index.php
muokkaaetusivua.php

Omat vaakapesät

omatpesat.php
pesaform.php
poistapesa.php
vaakadata.php
painoform.php
poistapaino.php

Käyttäjät

kayttajat.php
kayttajaform.php
poistakayttaja.php
lahetasahkopostia.php

Selaa vaakapesiä

selaa.php & selaa2.php
lisatiedot.php

Muokkaa vaakapesiä

muokkaapesia.php
pesaform2.php
poistapesa2.php
vaakadata2.php
painoform2.php
poistapaino2.php

Omat tiedot

omattiedot.php

Kirjautuminen

login.php
logout.php

Muut

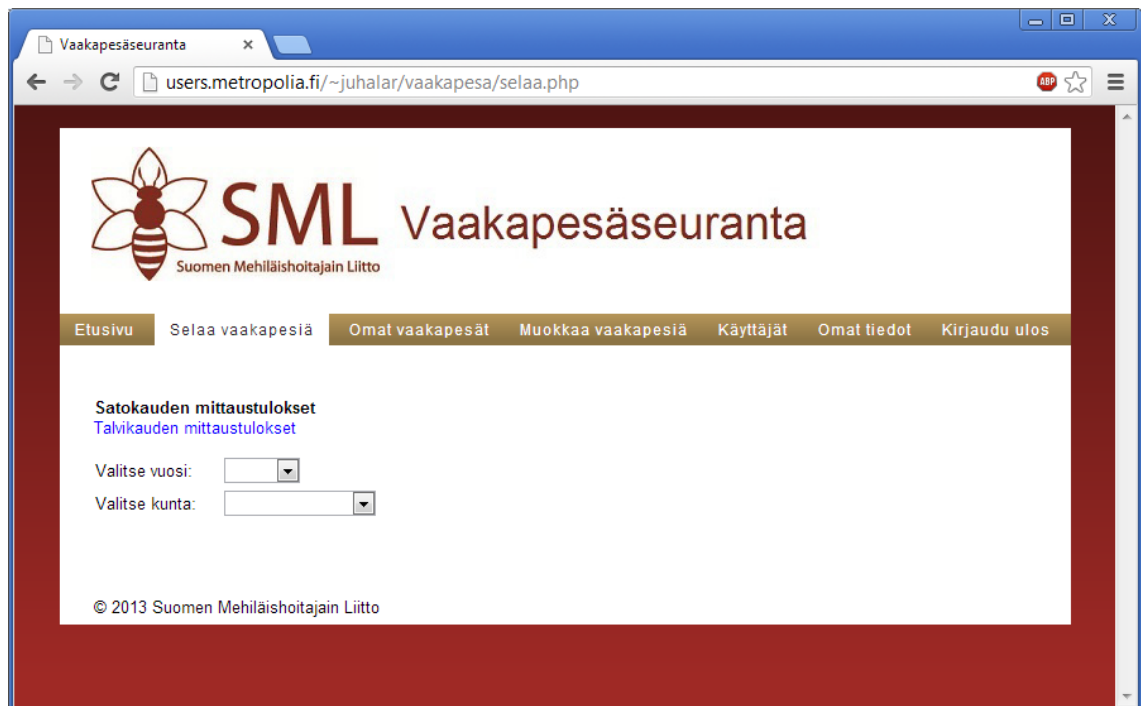
restricted.php
lisaadata.php

Kuva 4. Vaakapesäsovelluksen arkkitehtuurikuva

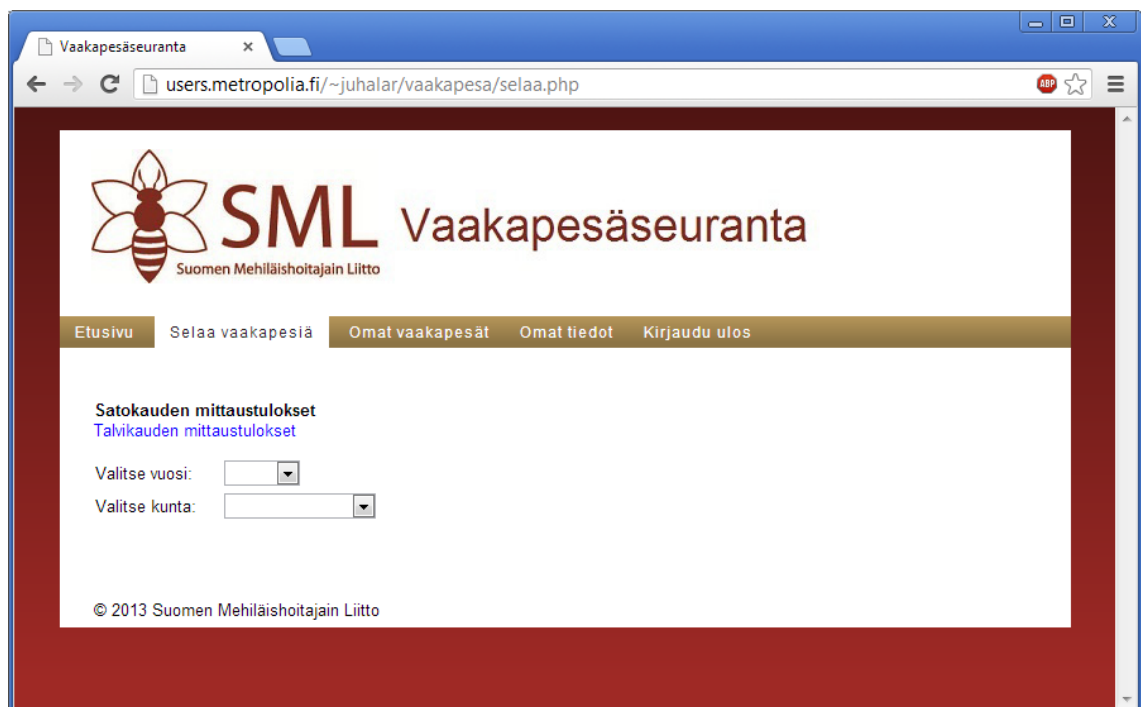
Kirjautumattomat käyttäjät pääsevät sivuille `index.php` (etusivu), `login.php` (kirjautumissivu), `sela.php` (satokauden pesien selaussivu), `sela2.php` (talvikauden pesien selaussivu) ja `lisatiedot.php` (pesien selaussivulla oleva lisätietoja pesästä - popup-sivu). Kaikki sovelluksen kirjautuneet käyttäjät pääsevät myös sivuille `omatpesat.php`, `omattiedot.php`, `painoform.php`, `pesaform.php`, `poistapaino.php`, `poistapesa.php` ja `vaakadata.php`. Ylläpitäjä-oikeustason vaativat sivut ovat `muokkaaetusivua.php`, `kayttajat.php`, `lahetasahkopoistia.php`, `kayttajaform.php`, `muokkaaapesia.php`, `painoform2.php`, `poistapaino2.php`, `poistapesa2.php` ja `vaakadata2.php`. Mikäli käyttäjän oikeustaso ei riitä halutulle sivulle, käyttäjä ohjataan sivulle `restricted.php`. `Lisaadata.php` on ohjelmalliseen vaakapesätietojen tallennukseen tehty lisäosa (lisää tästä voit lukea luvusta 2.8.). Siihen ei ole linkkiä millään sovelluksen sivulla, sillä sitä ei sovelluksen tavallisessa käytössä tarvitse.

2.4.1 Käyttäjäryhmät

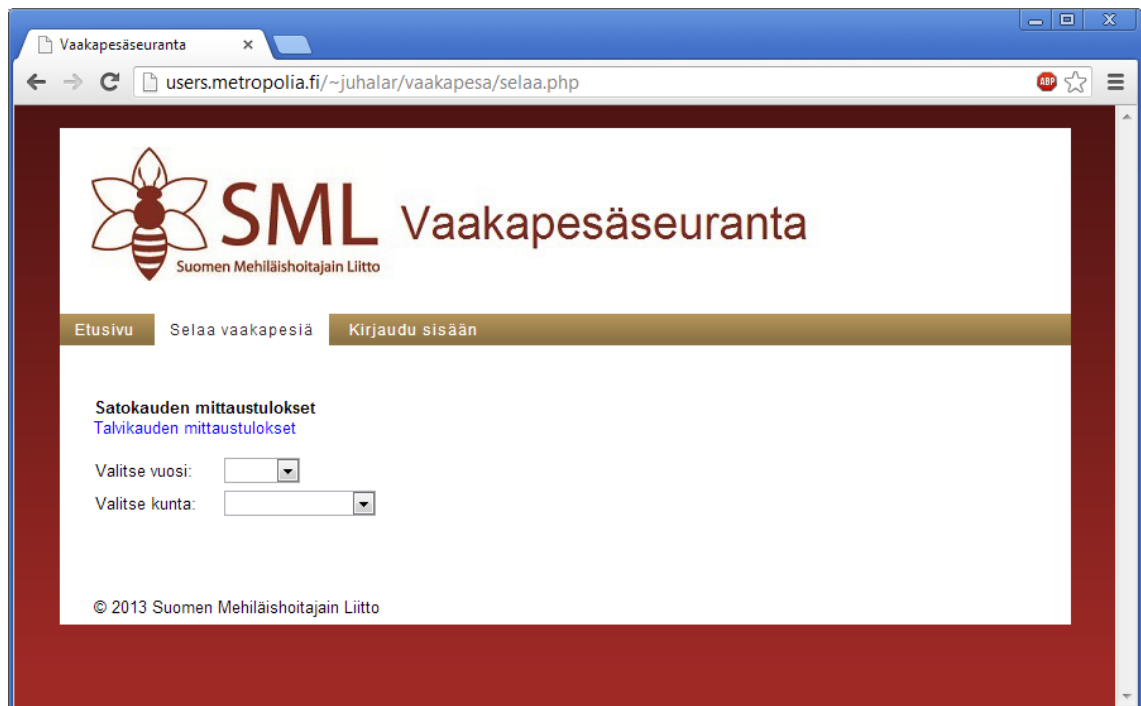
Vaakapesäsovelluksessa on kolme käyttäjäryhmää: ylläpitäjät, tarhaajat sekä kirjautumattomat käyttäjät. Ylläpitäjillä on pääsy sovelluksen kaikkiin osiin. Tarhaajat voivat muokata vain omia tietojaan ja omia pesiään. Kirjautumattomilla käyttäjillä on oikeus pelkästään selata vaakapesiä ja etusivua. Kuvista 5, 6 ja 7 näkyy, miten vaakapesän näkymä eroaa riippuen siitä, mihin käyttäjäryhmään käyttäjä kuuluu.



Kuva 5. Web-sovelluksen näkymä ylläpitäjänä



Kuva 6. Web-sovelluksen näkymä tarhaajana



Kuva 7. Web-sovelluksen näkymä kirjautumattomana käyttäjänä

2.4.2 Etusivu

Etusivulla on yleistä tietoa vaakapesistä (mahdollisesti myös tiedotteita) sekä vaakapesäseurannan käyttöohjeet. Käyttäjän ollessa kirjautuneena ylläpitäjänä tulee etusivun tekstin alle ”Muokkaa tekstiä” -linkki (kuva 8). Vain ylläpitäjät pääsevät muokkaamaan etusivun tekstiä. Tätä napsauttamalla aukeaa yksinkertainen lomake (kuva 9), jossa pääsee muokkaamaan tekstin haluamukseen. Etusivun teksti on tallennettuna yksinkertaiseen tekstitiedostoon, joka tulostetaan etusivulle ennen käyttöohje-linkkejä. Tämä toiminnallisuus tekee sovelluksen ylläpidosta helpomman, sillä etusivun koodiin ei tarvitse koskea, mikäli haluaa muuttaa etusivun tekstiä.



SML Vaakapesäseuranta

Suomen Mehiläishoitajain Liitto

[Etusivu](#)
[Selaa vaakapesiä](#)
[Omat vaakapesät](#)
[Muokkaa vaakapesiä](#)
[Käyttäjät](#)
[Omat tiedot](#)
[Kirjaudu ulos](#)

Näillä sivuilla voi seurata SML:n vaakapesien kehitystä ympäri Suomea. Vaakapesä on mehiläispesä, joka on varustettu tarkkuusvaa'alla ja joka punnitaan säännöllisesti.

Vaakapesätiedot antavat arvokasta tietoa hunajasadon kertymästä, satokauden alkamisesta ja kasvien kukinta-ajoista. Tietojen avulla mehiläistarhaaja voi suunnitella tulevia hoitotoimiaan tehokkaasti.

Vaakapesän sato ei kerro kokonaishunajasadon määrää, koska pesien välillä on suurta satovaihtelua. Usein vaakapesä on tarhaajan vahvin pesä ja parhaalla paikalla.

Mikäli olet kiinnostunut vaakapesistä, ota yhteyttä SML:ään: [sml\(at\)hunaja.net](mailto:sml(at)hunaja.net).

[Muokkaa tekstiä](#)

Vaakapesäseurannan käyttöohjeet:
[PDF](#) | [Word \(.docx\)](#)



Suomen
Mehiläishoitajain Liitto
SML ry




© 2013 Suomen Mehiläishoitajain Liitto

Kuva 8. Etusivun näkymä ylläpitäjänä



Kuva 9. Etusivun tekstin muokkaustoiminto (vain ylläpitäjillä oikeus tälle sivulle)

2.4.3 Selaa vaakapesiä -sivu

Selaa vaakapesiä -sivulle ei ole rajoitettu pääsyä keneltäkään. Sivulta voi valita satokauden tai talvikauden mittaustuloksien tarkastelun. Satokausi on ajanjakso kesällä, jolloin mehiläiset keräävät hunajaa ja pesän paino kasvaa. Talvikaudella mehiläiset pysyttelevät pesän sisällä ja kuluttavat hunajaa, joten talvikaudella pesän paino pienenee.

Kun haluttu kausi (satokausi tai talvikausi) on valittu, valitaan haluttu vuosi alasvetovalikosta. Vuoden valitsemisen jälkeen sivulle latautuu Suomen kartta, jossa näkyy pisteinä ne kunnat, joilta löytyy mittaustuloksia kyseiselle vuodelle. Käyttäjä voi nyt valita haluamansa kunnan joko alasvetovalikosta tai kartan jotain pistettä napsauttamalla. Kun kunta on valittu, sivulle tulostetaan kyseisen kunnan vaakapesät ja niiden mittaustulokset. Lisäksi sivulle tulostuu myös kuvaaja vaakapesien painonkehityksestä (mikäli mittaustuloksia on ainakin yhdellä vaakapesällä useampi kuin yksi). Halutun vaakapesän mittaustulosten alapuolella on "Lisätietoja pesästä" -

linkki, jota napsauttamalla aukeaa uusi ikkuna, jossa kerrotaan vaakapesän nimi, kunta, tilan nimi, tarhaajan nimi sekä mehiläisrotu.

2.4.4 Omat vaakapesät -sivu

Omat vaakapesät -sivulle on pääsy kaikilla sovelluksen kirjautuneilla käyttäjillä. Omat vaakapesät -sivulla voi lisätä, muokata tai poistaa omia vaakapesiä. Lisäksi sivulla voi omien vaakapesien mittaustuloksia tarkastella, lisätä, muokata tai poistaa. Vaakapesän mittaustuloksiin pääsee napsauttamalla halutun vaakapesän kohdalla "Avaa"-linkkiä. Mittaustuloksista näkyvät vain yhden vuoden tulokset kerrallaan (oletusarvoisena tuorein vuosi). Vuoden voi vaihtaa ylhäällä vasemmalla olevasta alasvetovalikosta. Uutta mittaustulosta lisättäessä talvi-valintaruutu rastittuu automaattisesti, mikäli tämän hetkinen kuukausi on jotain lokakuun ja maaliskuun väliltä (mehiläispesien satokausi on suunnilleen huhtikuun puolesta välistä syyskuun puoleen väliin). Talven voi kuitenkin rastittaa pois halutessaan (esimerkiksi vanhoja satokauden mittaustuloksia kirjatessa).

Mittaustuloksia voi tallentaa vain päivän tarkkuudella, ja sovellus hyväksyy vain yhden mittaustuloksen samalle päivämäärälle. Mikäli pesään tehdään jokin muutos (esimerkiksi laatikon lisäys), kirjataan sovelluksessa mittaustulos-kohtaan mittaustulos ennen muutosta. Jos mitään painoon vaikuttavaa muutosta ei pesään ole mittauspäivänä tehty, jätetään mittaustulos muutoksen jälkeen -kohta tyhjäksi.

Satokaudella mehiläistarhaaja saattaa haluta kirjata ylös satokasveja. Satokasveja voi rastittaa haluamansa määrän, mutta muita satokasveja kuin valintalistassa olevia voi olla vain yksi mittaustulosta kohden. Valitut satokasvit näkyvät selaa vaakapesiä -sivun kuvaajassa, kun hiiren osoittimen siirtää kyseisen mittaustuloksen kohdalle.

2.4.5 Muokkaa vaakapesiä -sivu

Muokkaa vaakapesiä -sivulle on pääsy vain ylläpitäjillä. Muokkaa vaakapesiä -sivu on samanlainen kuin Omat vaakapesät -sivu sillä erotuksella, että Muokkaa vaakapesiä -sivulla voi tarkastella, lisätä, muokata tai poistaa kenen tahansa vaakapesiä. Sivun tarkoitus on mahdollistaa ongelmatilanteissa ylläpitäjille pääsyn tarkastelemaan ja muokkaamaan ongelman kohdanneen käyttäjän vaakapesää.

2.4.6 Käyttäjät-sivu

Käyttäjät-sivu on rajoitettu vain ylläpitäjille. Käyttäjät-sivulla näkee jokaisen sovellukseen lisätyn käyttäjän. Halutun käyttäjän tietoja voi tutkia tai muokata napsauttamalla kyseisen käyttäjän kohdalla Muokkaa-linkkiä. Muokkaa-linkin takaa aukeavalla lomakkeella voi muokata (tai vaihtoehtoisesti vain tarkastella) käyttäjän käyttäjätunnusta, salasanaa, etunimeä, sukunimeä, sähköpostiosoitetta, puhelinnumeroa, lähiosoitetta, postinumeroa, postitoimipaikkaa sekä oikeustasoa. Salasanat ovat tallennettuna tietokantaan MD5-kryptattuina, joten niitä ei pääse näkemään vaan ne voi ainoastaan muuttaa. Mikäli jokin sovelluksen käyttäjä on hukannut salasanansa, voi ylläpitäjä vaihtaa salasanan uuteen ja esimerkiksi lähettää sähköpostilla käyttäjälle tämä uusi salasana.

Ylläpitäjä voi poistaa käyttäjiä napsauttamalla halutun käyttäjän kohdalla Poista-linkkiä (itseään ei voi poistaa turvallisuussyistä). Poista-linkin napsauttamisen jälkeen tulee vielä vahvistuskysymys, johon täytyy vastata myöntävästi, mikäli halutun käyttäjän todella haluaa poistaa. Poistamisen jälkeen kyseisen käyttäjän tietoja ei voi enää mitenkään saada palautettua.

Käyttäjät-sivulla voi myös lähettää sovelluksen käyttäjille sähköpostia. Ylläpitäjän täytyy ensin valita halutut käyttäjät laittamalla valintaruutuihin rastit (mahdollista vain niiden käyttäjien kohdalla, joilla on tietoihin lisätty sähköpostiosoite). Mikäli ylläpitäjä haluaa valita kaikki käyttäjät, onnistuu se helposti napsauttamalla Valitse kaikki -valintaruutuun rastin. Kun halutut käyttäjät on valittu, sähköpostin kirjoittamiseen pääsee napsauttamalla Lähetä sähköpostiviesti valituille -linkkiä. Tämän jälkeen aukeaa sähköpostin lähetys -sivu. Mikäli ylläpitäjä haluaa vaihtoehtoisesti käyttää omaa sähköpostiohjelmaa (esim. Microsoft Outlook) sähköpostin lähetykseen, voi hän vain kopioida Vastaanottaja-kentässä olevan osoitelistan omaan sähköpostisovellukseen ja napsauttaa sen jälkeen vaakapesäsovelluksen Peruuta-nappia. Mikäli ylläpitäjä haluaa kuitenkin käyttää vaakapesäsovelluksen omaa sähköpostin lähetys -toimintoa, täytyy ylläpitäjän syöttää sähköpostiviestille haluamansa osoite sekä itse viesti. Vaakapesäsovelluksen sähköpostin lähetys -toimintoa käyttäessä sähköpostiviestin lähettäjä-kenttää ei pääse muokkaamaan, vaan sen arvoksi on asetettu sml@hunaja.net (Suomen Mehiläishoitajain Liiton oma sähköpostiosoite). Sähköpostiviestin lähetys tapahtuu lomakkeessa olevalla Lähetä-napilla. Mikäli kaikki

sujuu hyvin, Lähetä-napin napsauttamisen jälkeen lomake tyhjentyy ja ylälaitaan tulee viesti ”Sähköposti lähetetty.”.

2.4.7 Omat tiedot -sivu

Omat tiedot -sivulle on pääsy kaikilla kirjautuneilla käyttäjillä. Omat tiedot -sivulla voi tarkastella tai muokata nimensä mukaisesti omia tietojaan. Muokattavat tiedot ovat käyttäjätunnus, salasana, etunimi, sukunimi, sähköpostiosoite, puhelinnumero, lähiosoite, postitoimipaikka ja postinumero. Salasanaa ei näytetä (eikä niitä voidakaan näyttää, sillä salasanat ovat tallennettuna tietokantaan MD5-kryptattuina), mutta sen voi halutessaan muuttaa. Toisin kuin ylläpitäjien Käyttäjät-sivulla, Omat tiedot -sivulla ei pääse muokkaamaan oikeustasoa.

2.4.8 Tietokannan luontiskriptit

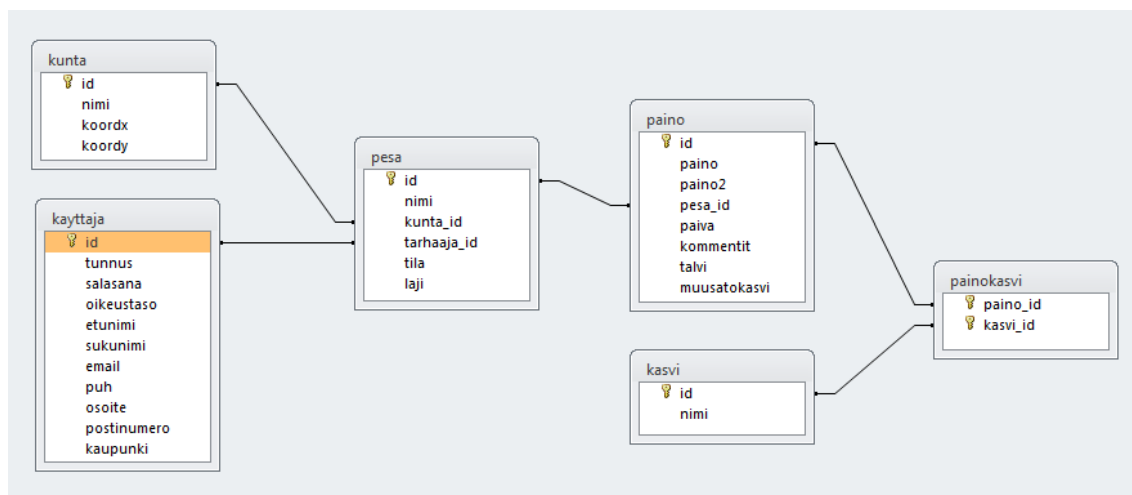
Tein tietokannan rakentamista helpottaakseni viisi erilaista tietokantataulujen luontiskriptiä PHP:lla. Luontiskriptejä ei tarvita enää valmiissa sovelluksessa sen jälkeen, kun ne on ensimmäisen kerran ajettu läpi. Sovelluksen kehittämisvaiheessa jouduin ajamaan luontiskriptejä useasti. Erityisesti luontiskripti2.php:ta tuli ajettua ahkerasti kuntakartan kehitysvaiheessa (helppo poistaa taulu ja luoda se uudelleen uusilla arvoilla luontiskriptin avulla).

Seuraavana on lueteltuna lyhyet kuvaukset kustakin luontiskriptistä:

- Luontiskripti.php: Luo käyttäjä-aulun ja lisää siihen oletusarvoisen admin-käyttäjän, jotta muita käyttäjiä pääsisi sovelluksesta käsin lisäälemään. Admin-käyttäjän salasana kannattaa vaihtaa heti alkuun.
- Luontiskripti2.php: Luo kunta-aulun ja lisää siihen kunnat ja koordinaatit (kuntakarttaa varten) kunta.txt-tiedostosta lukien. Kunta.txt-tiedostossa kuntien tiedot syötetään muodossa kunnannimi|koordx|koordyy. Kunnat erotellaan toisistaan rivinvaihdolla.
- Luontiskripti3.php: Luo pesä-aulun vaakapesiä varten.
- Luontiskripti4.php: Luo paino-aulun vaakapesien mittaustuloksia varten.
- Luontiskripti5.php: Luo kasvi-aulun satokasveille ja lisää siihen satokasvit: pajut, voikukka, omena, puolukka, vadelma, horsma, rypsi, apilat ja kanerva. Tämän jälkeen skripti luo vielä painokasvi-aulun

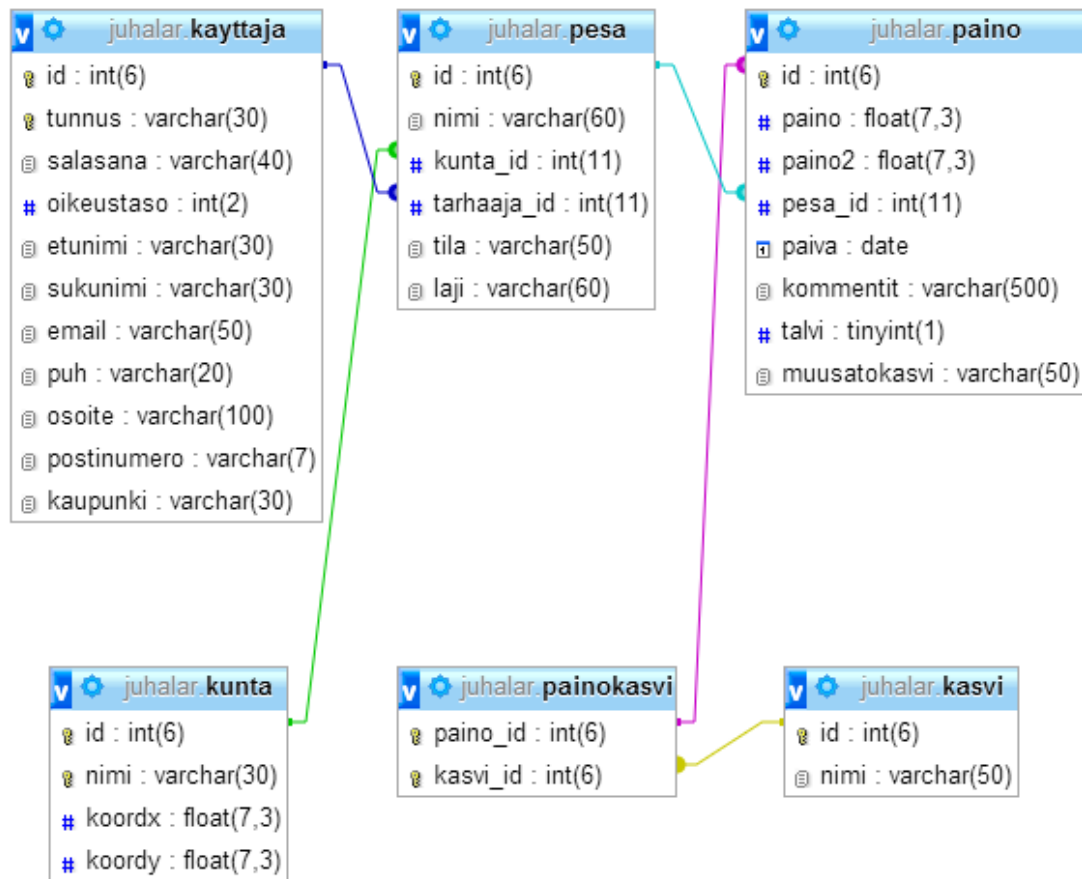
(monta-moneen suhde), jonka avulla voidaan yhdistää painolukemiin satokasveja.

2.5 Tietokannan rakenne



Kuva 10. Vaakapesäsovelluksen tietokantakaavio (Microsoft Access)

Vaakapesäsovellus käyttää kuutta tietokantataulua: kayttaja, kunta, pesa, paino, kasvi ja painokasvi. Mikään tietokantataulu ei ole täysin erillään muista, vaan jokaisella taululla on jokin relaatiotietokantaan suhde johonkin toiseen tauluun. Kaikki tietokantataulut on säädetty käyttämään UTF-8-merkistöä. Microsoft Accessilla tehty sovelluksen tietokantakaavio on kuvassa 10. PhpMyAdmin-työkalulla saa tietokannasta automaattisesti tietokantakaavion (kuva 11), mutta tietokantakaavioon joutuu relaatiot lisäämään itse. Tässä luvussa on kerrottu jokaisesta tietokannan taulusta yksityiskohtaisemmin sekä havainnollistettu kenttien ominaisuuksia taulukoilla. Tietokannassa ei ylläpidetä viite-eheyttä tietokantatasolla, eikä käyttäjä pääse sovelluksesta käsin helposti rikkomaan tietokannan viite-eheyttä. Sovellukseen on tehty käsin joihinkin poistotoiminnallisuuksiin automaattinen lapsitaulun tietojen poisto. Tällaiset poistotoiminnallisuuden yhteydet ovat pesa- ja paino-taulujen välillä sekä paino- ja painokasvi-taulujen välillä.



Kuva 11. Vaakapesäsovelluksen yksityiskohtaisempi tietokantakaavio (phpMyAdmin)

2.5.1 Kayttaja-taulu

Kayttaja-taulun on tarkoitus pitää sisällään tietoja sovelluksen kirjautumista vaativien osien käyttäjistä (tarhaajat ja ylläpitäjät). Kayttaja-taulu koostuu kentistä id (perusavain), tunnus, salasana, oikeustaso, etunimi, sukunimi, email, puh, osoite, postinumero sekä kaupunki. Id-kenttä on yksilöllinen tunnistenumero, jonka ohjelma antaa automaattisesti luodessaan uuden käyttäjän. Tunnus-kenttä on merkkijono, johon annetaan haluttu käyttäjätunnus. Salasana-kenttä on merkkijono, johon tallennetaan MD5-tiiviste halutusta salasanasta. Tämä tarkoittaa myös sitä, että mikäli jokin käyttäjä hukkaa salasanansa, ei sitä ylläpitäjä pysty kaivamaan esiin. Ylläpitäjä voi kuitenkin käydä vaihtamassa käyttäjän salasanan johonkin toiseen. Oikeustaso-kenttä ilmoittaa, onko käyttäjällä ylläpitäjän oikeuksia. Arvo 1 tarkoittaa tavallisia tarhaajan oikeuksia ja arvo 2 ylläpitäjän oikeuksia.

Etunimi-kenttä on merkkijono, johon syötetään käyttäjän etunimi. Sukunimi-kenttä on merkkijono, johon syötetään käyttäjän sukunimi. Email-kenttä on merkkijono, johon syötetään käyttäjän sähköpostiosoite. Email on vapaaehtoinen, mutta suositeltava kenttä, sillä sitä käytetään sovelluksessa mahdollisten tiedotteiden lähettämiseen. Puh-kenttä on merkkijono, johon syötetään käyttäjän puhelinnumero. Postinumero-kenttä on merkkijono, johon syötetään käyttäjän postinumero. Kaupunki-kenttä merkkijono, johon syötetään käyttäjän postitoimipaikka. Taulukosta 1 näkee yksityiskohtaisemmat tiedot kayttaja-aulun kenttien ominaisuuksista.

Taulukko 1: Kayttaja-aulun kenttien ominaisuuudet

kentän nimi	pakollinen?	tyyppi	pituus	yksilöllinen?	oletusarvo
id	autom.	int	6	kyllä	autom.
tunnus	kyllä	varchar	30	kyllä	ei määr.
salasana	kyllä	varchar	40	ei	ei määr.
etunimi	kyllä	varchar	30	ei	ei määr.
sukunimi	kyllä	varchar	50	ei	ei määr.
email	ei	varchar	50	ei	ei määr.
puh	ei	varchar	20	ei	ei määr.
osoite	ei	varchar	100	ei	ei määr.
postinumero	ei	varchar	7	ei	ei määr.
kaupunki	ei	varchar	30	ei	ei määr.
oikeustaso	kyllä	int	2	ei	1

2.5.2 Kunta-aulu

Kunta-aulun on tarkoitus sisältää kaikki Suomen kunnat. Kunta-aulu koostuu kentistä id (perusavain), nimi, koordx ja koordx. Id-kenttä on yksilöllinen tunnistenumero, jonka ohjelma antaa automaattisesti uuden kunnan luonnin yhteydessä. Nimi-kenttä on nimensä mukaisesti tarkoitettu kuntien nimille. Koordx- ja koordx-kentät ovat float-tyyppisiä (liukuluku), ja niihin on määritelty maksiminumeromääräksi 7 ja desimaaliosan maksiminumeromääräksi 3 (esim. suurin luku minkä voi syöttää, on 9999,999). Koordx ja koordx pitävät sisällään kuntien x- ja y-koordinaatit (pikseleinä) kuntakartalla (jpg-kuva, jonka leveys on 300 ja korkeus 522 pikseliä). Taulukosta 2 selviävät yksityiskohtaisemmat tiedot kunta-aulun kentistä.

Taulukko 2: Kunta-aulun kenttien ominaisuuudet

kentän nimi	pakollinen?	tyyppi	pituus	yksilöllinen?	oletusarvo
id	autom.	int	6	kyllä	autom.
nimi	kyllä	varchar	30	kyllä	ei määr.

koordx	ei	float	7, 3	ei	ei määr.
koordx	ei	float	7, 3	ei	ei määr.

2.5.3 Pesa-taulu

Pesa-taulu sisältää vaakapesäsovellukseen lisätyt vaakapesät. Pesa-taulu koostuu kentistä id (perusavain), nimi, kunta_id (vierasavain), tarhaaja_id (vierasavain), tila ja laji. Kenttä id on yksilöllinen tunnistenumero, jonka ohjelma antaa automaattisesti uuden pesän luonnin yhteydessä. Nimi-kenttä on tarkoitettu pesän nimelle ja sen olisi suotavaa olla jollain tapaa vaakapesää kuvaava (esim. Harjula tarkkuusvaaka). Kunta_id-kenttä on tarkoitettu ilmaisemaan missä kunnassa vaakapesä sijaitsee. Kunta_id-kenttään syötetään jonkin olemassa olevan kunnan id-tunnistenumero. Tarhaaja_id-kenttään on tarkoitus syöttää sen käyttäjän id-tunnistenumero, joka luo vaakapesän ja hallinnoi sitä (=syöttää mittauslukemia). Tila-kenttä on tarkoitettu (maa)tilan nimelle, jossa vaakapesä sijaitsee. Laji-kenttään on tarkoitus syöttää pesässä majailevan mehiläisrodun nimi. Taulukko 3 listaa yksityiskohtaisemmat tiedot pesa-tilin kentistä.

Taulukko 3: Pesa-tilin kenttien ominaisuudet

kentän nimi	pakollinen?	tyyppi	pituus	yksilöllinen?	oletusarvo
id	autom.	int	6	kyllä	autom.
nimi	kyllä	varchar	60	ei	ei määr.
kunta_id	kyllä	int	ei määr.	ei	ei määr.
tarhaaja_id	kyllä	int	ei määr.	ei	ei määr.
tila	ei	varchar	50	ei	ei määr.
laji	ei	varchar	60	ei	ei määr.

2.5.4 Paino-tili

Paino-tili sisältää mehiläispesien mittaustuloksia. Paino-tili koostuu kentistä id (perusavain), paino, paino2, pesa_id, paiva, kommentit, talvi ja muusatokasvi. Id-kenttä on yksilöllinen tunnistenumero, jonka ohjelma antaa automaattisesti uuden painon kirjaamisen yhteydessä. Paino- ja paino2-kentät ovat float-tyyppisiä (liukuluku) ja niiden tarkkuudeksi on säädetty 0,001 kg (suurin mahdollinen painolukema on määritetty 9999,999 kg:ksi). Paino- ja paino2-kenttien on tarkoitus ilmaista vaakapesän painoa kilogrammoina. Paino2-kenttään annetaan arvo vain, mikäli pesään on tehty jokin painoon vaikuttava muutos (laatikon lisäys, hunajan poisto tms). Tässä tapauksessa paino-kenttään annetaan paino ennen muutosta ja paino2-kenttään paino muutoksen

jälkeen. Mikäli mitään painoon vaikuttavaa muutosta ei ole tehty, annetaan pelkästään paino-kentälle arvo.

Pesa_id-kenttään syötetään sen pesän id-tunnistenumero, josta painolukema on kirjattu. Paiva-kenttään tallennetaan mittaustuloksen päivämäärä. Paiva-kenttä on date-tyyppiä ja se tallennetaan tietokantaan muodossa vvvv-kk-pp. Paiva-kentän tallennukseen ja näyttämiseen on kuitenkin tehty sellainen muutos, että sovelluksesta käsin paiva-kenttä on suomalaisittain muotoa pp.kk.vvvv. Tietokantaan se kuitenkin tallentuu edelleen muodossa vvvv-kk-pp.

Kommentit-kenttä on tarkoitettu mittaustuloksen kommenteille, ja siihen kirjoittaminen olisi suotavaa, mikäli pesään on esimerkiksi tehty jokin painoon vaikuttava muutos. Talvi-kenttä on boolean-tyyppinen (itse asiassa sama asia kuin tinyint(1)) kenttä ja se voi saada pelkästään joko arvon 0 tai 1. Talvi-kentän arvo 0 tarkoittaa satokauden mittaustulosta ja arvo 1 talvikauden mittaustulosta. Talvikausi alkaa mehiläispesissä suunnilleen syyskuun puolessa välissä ja päättyy huhtikuun puolessa välissä. Muusatokasvi-kenttä on tarkoitettu sovelluksen satokasvilistan ulkopuolisille satokasveille (eli mikäli käyttäjä ei löydä haluamaansa satokasvia sovelluksessa olevasta satokasvilistasta). Kuva 12 selventää hieman, mitä muusatokasvi-kentällä tarkoitetaan. Kuvan mukaisella lomakkeen täyttämällä muusatokasvi-kenttä saisi arvon ”mansikka”.

Mittaustulos*:	35,900	kg
Mittaustulos muutoksen jälkeen:		kg
Päivämäärä*:	13.05.2012	syötä muodossa pp.kk.vvvv
Satokasvi:	☐ pajut	Muokkaa Poista
	☐ voikukka	Muokkaa Poista
	☐ omena	Muokkaa Poista
	☐ puolukka	Muokkaa Poista
	☐ vadelma	Muokkaa Poista
	☐ horsma	Muokkaa Poista
	☐ rypsi	Muokkaa Poista
	☐ apilat	Muokkaa Poista
	☒ muu, mikä?	mansikka

Kuva 12. Muusatokasvi-kentän syöttäminen vaakapesäsovelluksessa

Taulukossa 4 on vielä listattuna yksityiskohtaisemmat tiedot paino-tilun kentistä.

Taulukko 4: Paino-tilun kenttien ominaisuudet

kentän nimi	pakollinen?	tyyppi	pituus	yksilöllinen?	oletusarvo
id	autom.	int	6	kyllä	autom.
paino	kyllä	float	7, 3	ei	ei määr.
paino2	ei	float	7, 3	ei	ei määr.
pesa_id	kyllä	int	ei määr.	ei	ei määr.
paiva	kyllä	date	-	ei	ei määr.
kommentit	ei	varchar	500	ei	ei määr.
talvi	kyllä	boolean/tinyint	2	ei	0
muusatokasvi	ei	varchar	50	ei	ei määr.

2.5.5 Kasvi-tilu

Kasvi-tilu pitää sisällään sovelluksen satokasvilistan. Satokasvilistasta tuli lopulta staattinen (muuttumaton). Alun perin oli ajatuksenani tehdä satokasvilistasta dynaaminen siten, että käyttäjät voisivat lisäillä siihen uusia satokasveja. Teinkin jo sovellukseen toteutuksen, jossa oli mukana tämä toiminto sekä ylläpitäjillä mahdollisuus muokata ja poistaa satokasveja. Satokasvilista olisi saattanut kuitenkin ajan kuluessa paisua melko suureksi, joten vaihdoin satokasvilistan muuttumattomaksi. Satokasvilistaksi tuli lopulta kukkimisjärjestyksessä: pajut, voikukka, omena, puolukka, vadelma, horsma, rypsi, apilat ja kanerva. Kasvi-tilu koostuu kentistä id (ensisijainen avain) ja nimi. Id-kenttä on yksilöllinen tunnistenumero, jonka ohjelma antaa automaattisesti uuden satokasvin luonnin yhteydessä. Nimi-kenttä on merkkijono, ja siihen on tarkoitus syöttää kasvin nimi. Taulukosta 5 näkee yksityiskohtaisemmat tiedot kasvi-tilun kahdesta kentästä.

Taulukko 5: Kasvi-tilun kenttien ominaisuudet

kentän nimi	pakollinen?	tyyppi	pituus	yksilöllinen?	oletusarvo
id	autom.	int	6	kyllä	autom.
nimi	kyllä	varchar	30	kyllä	ei määr.

2.5.6 Painokasvi-tilu

Painokasvi-tilu on monta-moneen-tilu, joka ilmoittaa mitä satokasveja (kasvi-tilu) on rastittu mittaustulokselle (paino-tilu). Painokasvi-tilu koostuu kentistä paino_id ja kasvi_id. Paino_id ja kasvi_id muodostavat yhdessä yksilöllisen perusavaimen.

Paino_id-kentälle annetaan paino-aulun jonkin alkion id-tunnistenumero ja kasvi_id-kentälle kasvi-aulun jonkin alkion id-tunnistenumero. Yksityiskohtaisemmat tiedot painokasvi-aulun kentistä näkee taulukosta 6.

Taulukko 6: Painokasvi-aulun kenttien ominaisuudet

kentän nimi	pakollinen?	tyyppi	pituus	yksilöllinen?	oletusarvo
paino_id	kyllä	int	6	ei	ei määr.
kasvi_id	kyllä	int	6	ei	ei määr.

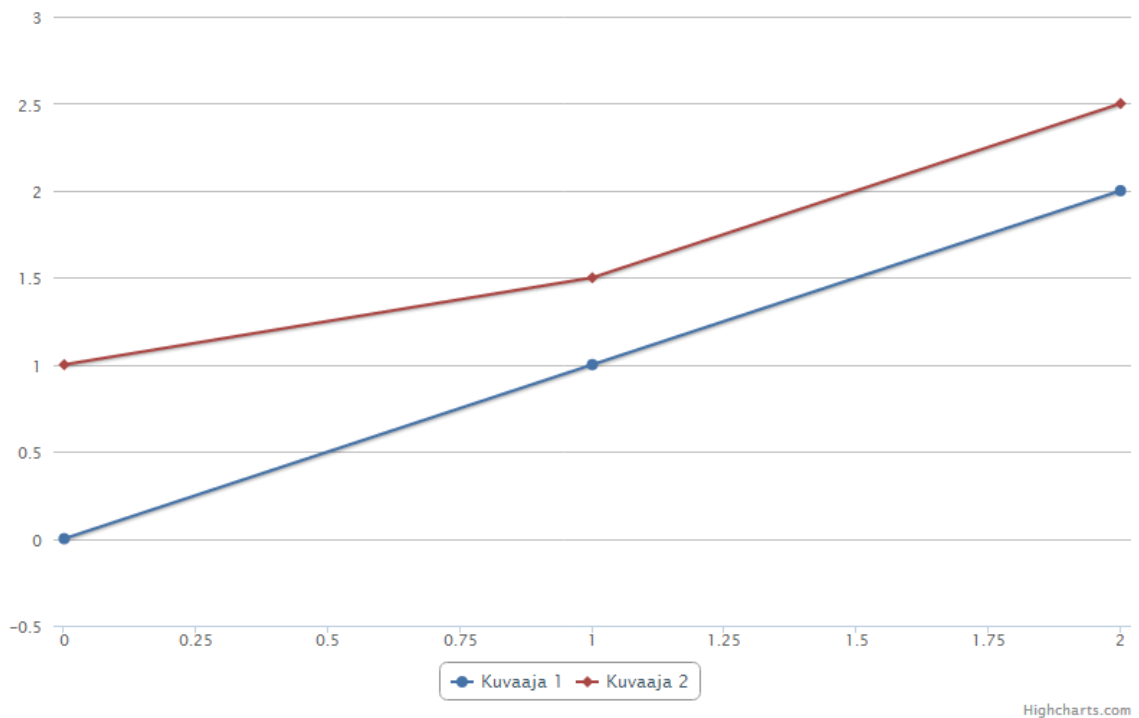
2.6 Highcharts-kirjasto kuvaajien piirtämiseen

Vaakapesäsovelluksen piti pystyä piirtämään kuvaajia vaakadatasta, mikä oli pienoinen ongelma, sillä kuvaajanpiirtotyökalun kehittäminen olisi työläs ja aikaavievä prosessi. Päätin lopulta etsiä valmiin työkalun kuvaajien piirämiseen. Pitkällisen etsinnän jälkeen löysin lopulta JavaScriptillä toteutetun Highcharts-kirjaston, jolla saisi piirrettyä mehiläispesien painonkehityksistä näyttäviä kuvaajia. Highcharts-kirjasto on täysin ilmainen, jos sitä käyttävä sovellus ei ole kaupallinen. Jotta sain kuvaajien piirtoon dynaamista dataa (mittaustuloksia tietokannasta), minun täytyi lisätä JavaScript-koodin sekaan PHP:ta.

Highchartsilla on mahdollista piirtää useita rinnakkaisia kuvaajia samaan kuvaan. Rinnakkaisten kuvaajien ei tarvitse olla synkronissa eli kuvaajissa voi olla toisistaan poikkeava määrä pisteitä. Kuvaajat määritellään series-kohdassa. Seuraavana on esimerkki useamman kuvaajan lisäämisestä samaan graafiin.

```
series: [
// Kuvaaja 1:n kuvaus alkaa
{
    name: 'Kuvaaja 1',
    // Kuvaaja 1:n pisteet
    data: [ { x: 0, y: 0 },
            { x: 1, y: 1 }, { x: 2, y: 2 } ]
},
// Kuvaaja 2:n kuvaus alkaa
{
    name: 'Kuvaaja 2',
    // Kuvaaja 2:n pisteet
    data: [ { x: 0, y: 1 },
            { x: 1, y: 1.5 }, { x: 2, y: 2.5 } ]
}
]
```

Kuva 13 näyttää, miltä yllä olevan esimerkin mukainen graafi näyttää:



Kuva 13. Yksinkertainen Highcharts-esimerkki, johon on piirretty kaksi kuvaajaa

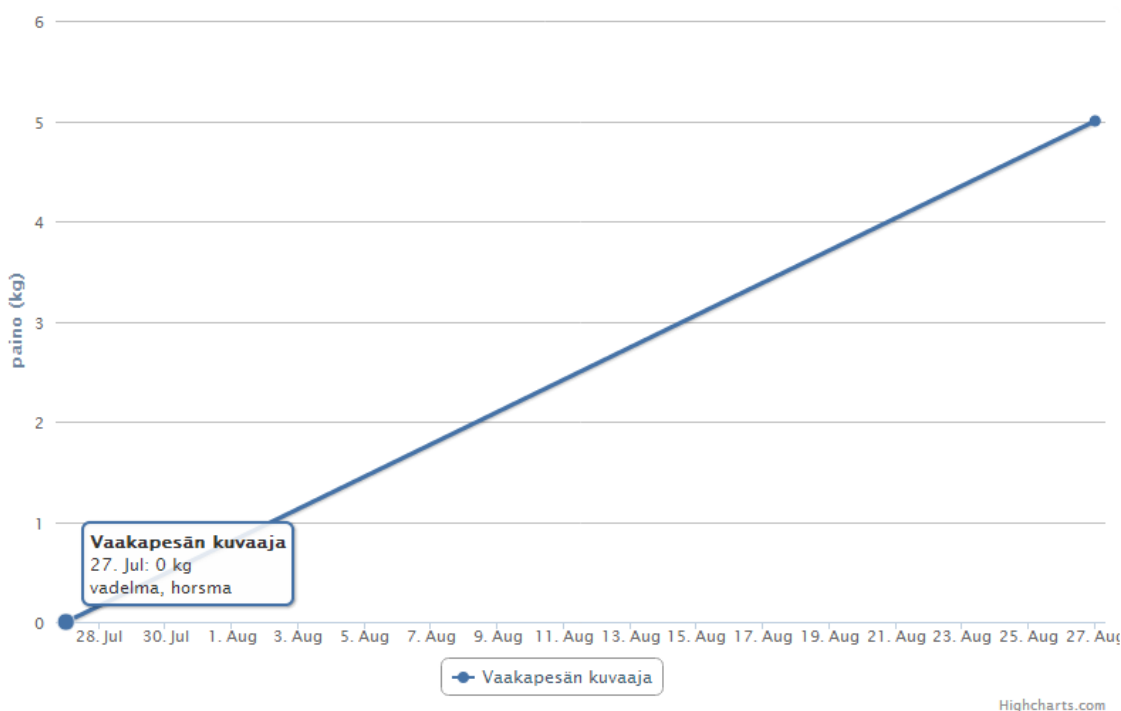
Kuvaajan pisteiden x- ja y-arvojen lisäksi Highchartsilla voi myös tallentaa lisädataa kuhunkin pisteeseen. Hyödynsin tätä ominaisuutta satokasvi-informaation lisäämiseksi kuvaajien pisteisiin. Seuraavassa yksinkertainen esimerkki siitä, miten lisädataa voi tallentaa kuvaajan pisteisiin:

```
name: 'Vaakapesän kuvaaja',
data: [
  {
    x: Date.UTC(1900, 6, 27),
    y: 0,
    satokasvit: 'vadelma, horsma'
  },
  {
    x: Date.UTC(1900, 7, 27),
    y: 5,
    satokasvit: 'mustikka'
  }
]
```

Esimerkin mukainen lisädata tallentuu muuttujaan `this.point.config.satokasvit`. Sen voi tulostaa tooltipiin (pieni informaatoruutu, joka aukeaa siirrettäessä hiiren osoitin pisteen päälle) esimerkiksi seuraavalla tavalla:

```
tooltip: {
    formatter: function() {
        return '<b>' + this.series.name + '</b><br/>' +
            Highcharts.dateFormat('%e. %b', this.x) + ': ' + this.y
            + ' kg<br/>' + this.point.config.satokasvit;
    }
}
```

Kuva 14 havainnollistaa, miltä yllä oleva esimerkki näyttää käytännössä. Kuvassa ei näy hiiren osoitinta (käyttämäni Print Screen piilottaa valitettavasti hiiren osoittimen). Hiiren osoitin on ensimmäisen kuvaajan pisteen päällä.



Kuva 14. Yksinkertainen Highcharts-esimerkki, johon on lisätty tooltipiin ylimääräistä informaatiota

Vaakapesäsovelluksen satokauden kuvaajissa on määritelty kiinteä päivämäärän näyttöalue (x-akseli) huhtikuun puolesta välistä syyskuun puoleen väliin (tämän ulkopuolella olevat satokauden mittaustulokset eivät näy kuvaajassa). Lisäksi satokauden mittaustuloksiin on asetettu kiinteä painon näyttöalue: maksimi +150 kg,

minimi -20 kg (kuvaajan y-akseli on painon kumulatiivinen muutos ensimmäisestä mittauksesta ja täten voi olla myös negatiivinen). Talvikauden mittaustuloksissa näyttöalue skaalautuu automaattisesti (ei ole asetettu kiinteää aluetta). Kuvaajasta saa tarkan päivämäärän ja mittaustuloksen näkyviin viemällä hiiren osoitin halutun mittaustuloksen päälle. Tällä tavalla saa satokauden mittaustuloksissa näkyviin myös satokasvit (mikäli tarhaaja on niitä rastittanut).

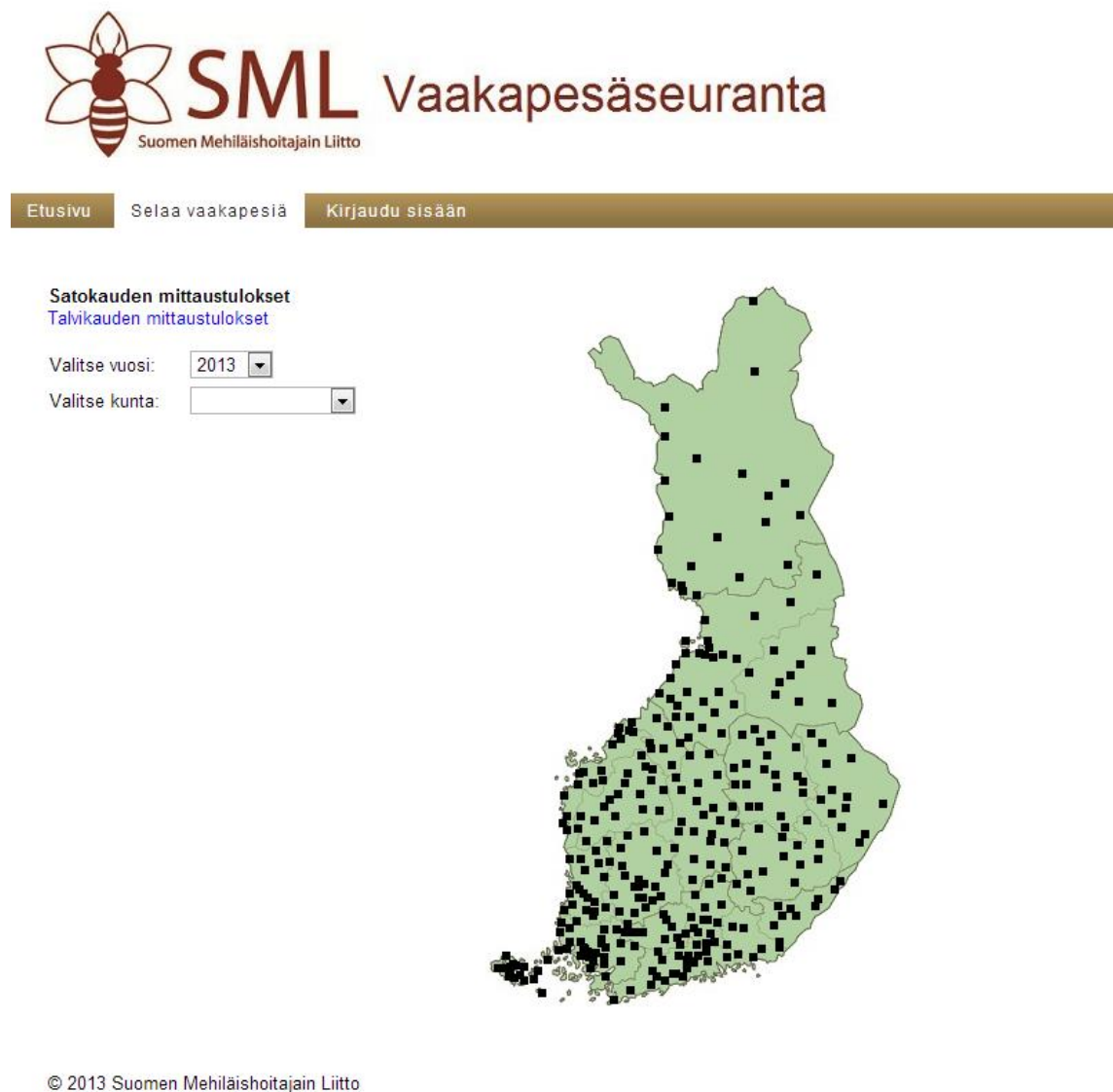
2.7 Kuntakartta

Sovelluksessa tuli olla kuntakartta, josta käyttäjä voisi valita haluamansa kunnan vaakapesät. Kuntakartan pohjana on Suomen kartta, jonka päälle piirretään ohjelmallisesti niiden kuntien pisteitä, joissa on vaakapesien mittaustuloksia valitulla vuodella. Kuntakartassa oli melko suuri haaste saada kuntien pisteet oikeisiin kohtiin kartalle. Kuntia on Suomessa tätä kirjoittaessani 320, joten jokaisen kunnan sijainnin käsin hahmottaminen ei tulisi kyseeseen. Kuntien pituus- ja leveyskoordinaatit löytyivät kyllä Wikipediasta, mutta niiden avulla piti vielä laskea kuntien pikselikoordinaatit karttakuvaan. Hahmottelin ensin muutaman toisistaan etäällä olevan kunnan x- ja y-koordinaatit karttakuvalle. Tämän jälkeen laskin Excelillä x- ja y-koordinaattien yhtälöt pituus- ja leveyskoordinaattien funktiona. Näillä yhtälöillä ratkaisin sitten Exceliä käyttäen jokaisen Suomen kunnan x- ja y-koordinaatit.

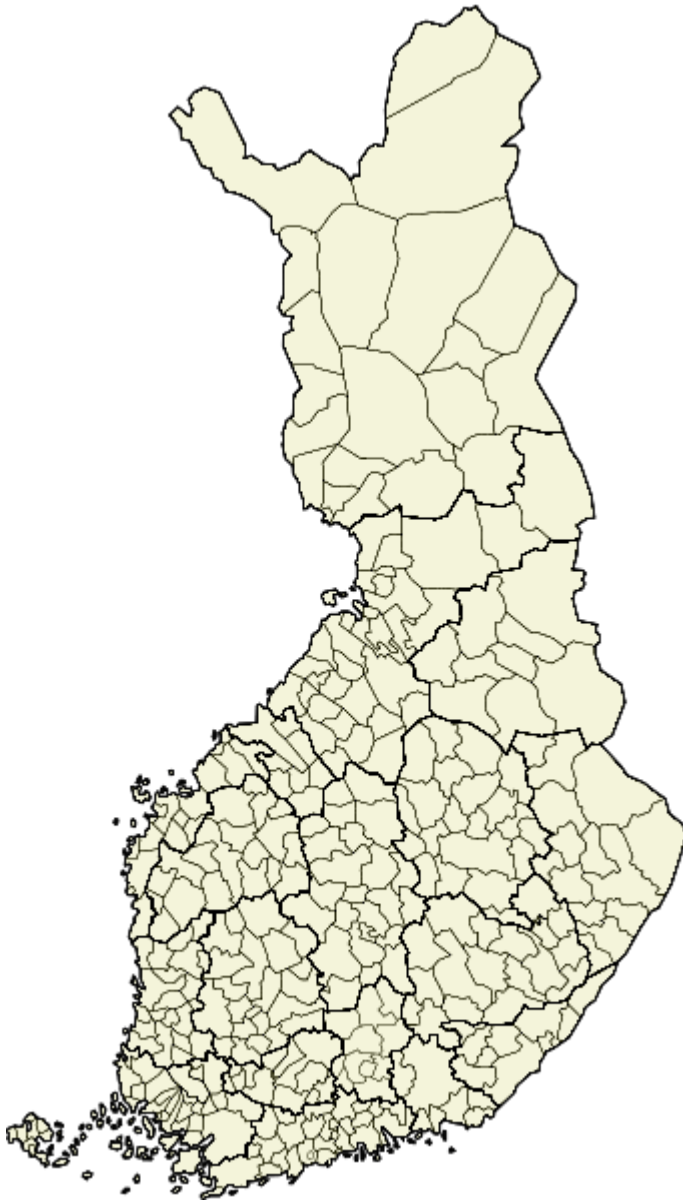
Tämä toimi vain osittain. Etelässä olevien kuntien pisteet kyllä näyttivät asettuvan melko kohdilleen, mutta pohjoiset kunnat ja osittain myös Ahvenanmaan kunnat asettuivat väärin kohtiin. Pohjoisimmat kunnat menivät jopa osittain Suomen rajojen ulkopuolelle. Vääristymät johtuivat siitä, että leveys- ja pituuspiirien linjat eivät kulje pallon pinnalla yhtä suorassa kuin karttakuvan x- ja y-akselit. Mitä lähempänä napoja ollaan, sitä suurempi on linjojen kaartuvuus.

Tämän tason ongelma olisi ehkä vaatinut työlästä kolmiulotteisen trigonometrian käyttöä, mutta en nähnyt tarvetta laskea kuntien sijainteja pikselilleen oikein. Päätin keksiä nopeamman ratkaisun: korjauskerroin vääristymien korjaamiseksi. Korjauskertoimen idea oli hyvin yksinkertainen: mitä pohjoisempana ollaan, sitä enemmän x-koordinaattia korjataan keskemälle. Y-koordinaatteja puolestaan korjailin hieman ylöspäin lännessä sijaitseviin Ahvenanmaan kuntiin. Kuvaan 15 olen laittanut esittämistarkoituksessa kaikki Suomen kunnat vaakapesäsovelluksen piirtäminä.

Kuvan alapuolelle olen vielä lisännyt kuvan 16, jossa on aito Suomen kuntakartta. Kuvia vertailemalla näkee, että lopputulos on varsin onnistunut.



Kuva 15. Kuntakartta (johon laitettu kaikki 320 kuntaa näkyviin)



Kuva 16. Aito Suomen kuntakartta

Käyttöversiossa kunnista näkyvät vain ne, joissa on vaakapesien mittaustuloksia valitulta vuodelta. Kunnan nimen saa näkyviin siirtämällä hiiren kunnan pisteen päälle ja halutun kunnan vaakapesien mittaustulokset saa näkyville napsauttamalla kunnan pistettä. Vaihtoehtoisesti kunnan voi valita myös vasemmalla olevasta alasvetovalikosta. Kuntakartta vie melko paljon tilaa, joten säädin sen katoamaan, kun käyttäjä valitsee jonkin kunnan (ja tällöin kyseisen kunnan vaakapesät mittaustuloksineen tulevat ruudulle).

2.8 Ohjelmallinen vaakapesädatan lähettäminen

Työnantajani pyysi minua tekemään vaakapesäsovellukseen pienen lisäosan, joka mahdollistaisi vaakapesädatan lähettämisen automaattisesti vaakapesistä. Saamani tiedon mukaan tällä hetkellä on olemassa vain puoliautomaattisia vaakapesiä, jolloin käyttäjä saa automaattisesti tiedon painosta esimerkiksi kännykkäänsä, mutta joutuu itse manuaalisesti syöttämään lukemat web-sovellukseen.

Vaakapesäsovelluksen on tarkoitus olla käytössä ainakin seuraavat 10 vuotta, joten siinä ajassa ehtii uskoakseni tulla täysautomaattisiakin vaakapesiä käyttöön. En halunnut, että MySQL-palvelimen tunnuksia jaeltaisiin näihin automaattisiin vaakapesiin, joten tein PHP:lla tietoa vastaanottavan osasen. Tämän osasen avulla vaakapesät voivat lähettää vaakadataa http-pyynnöillä tietynmuotoiseen osoitteeseen. Itse data annetaan osoitteen parametreina ja http-pyyntö lähetetään annettuun osoitteeseen (esim. <http://koti.tnnet.fi/web144/vaakapesa/lisaadata.php?tunnus=tunnus&salasana=salasana&pesa=6&paino=56,79&paiva=14.02.2013>). Automaattiset vaakapesät olisi hyvä ajastaa lähettämään mittaustuloksia 24 tunnin välein tai harvemmin, sillä vaakapesäsovellus ei hyväksy useampaa mittaustulosta samalla päivämäärällä (samaa vaakapesään).

Pakolliset parametrit ovat tunnus, salasana, pesa sekä paino. Parametrien järjestyksellä ei ole väliä. Tunnus-parametrille annetaan käyttäjätunnus. Salasana-parametrille annetaan käyttäjän salasana. Pesa-parametrille annetaan oman vaakapesän ID-tunnistenumero. ID-tunnistenumeron saa selville menemällä vaakapesäsovelluksessa omiin vaakapesiin ja sieltä napsauttamalla ”Muokkaa”, jonka jälkeen pitäisi ylimpänä näkyä ID-numero. Paino-parametrille annetaan vaakapesän mittaustulos kilogrammoina (desimaaliosa erotettuna joko pilkulla tai pisteellä). Paiva- ja talvi-parametrit ovat vapaaehtoisia. Paiva-parametrille annetaan päivämäärä muodossa pp.kk.vvvv. Mikäli paiva-parametrin jättää antamatta, (huom. ei saa merkitä ”paiva=”, vaan pitää kokonaan jättää antamatta: <http://koti.tnnet.fi/web144/vaakapesa/lisaadata.php?tunnus=tunnus&salasana=salasana&pesa=6&paino=56,79>) antaa ohjelma automaattisesti päiväksi tämän päivän.

Talvi-parametrille annetaan joko arvo 0 (satokauden mittaustulos) tai 1 (talvikauden mittaustulos). Mikäli talven jättää antamatta, ohjelma valitsee arvon tämän päivän perusteella (loka-maalis=1, huhti-syys=0). Virhetilanteiden varalta suosittelen, että

automaattinen sovellus tallentaisi lokiin (esimerkiksi tekstitiedostoon) jokaisen lähetyksen yhteydessä HTTP-vastauksen sisällön. Ohjelma tulostaa jokaisen mittaustuloksen lähetyksen yhteydessä yhden rivin ilmoituksen, joka kertoo, onnistuiko lähetys vai menikö jokin vikaan. Taulukkoon 7 on listattu mahdolliset ilmoitukset:

Taulukko 7: Lisaadata.php:n tuottamat mahdolliset ilmoitukset

Ilmoitus
Käyttäjätunnus/salasana virheellinen. Et syöttänyt kaikkia vaadittavia tietoja. Päivämäärän muoto virheellinen. Vaakapesää ei löytynyt. Käyttäjä ei ole pesän omistaja. Vaakapesälle on jo olemassa sama päivämäärä. Datan tallennus onnistui.

2.9 Sovelluksen tietoturva

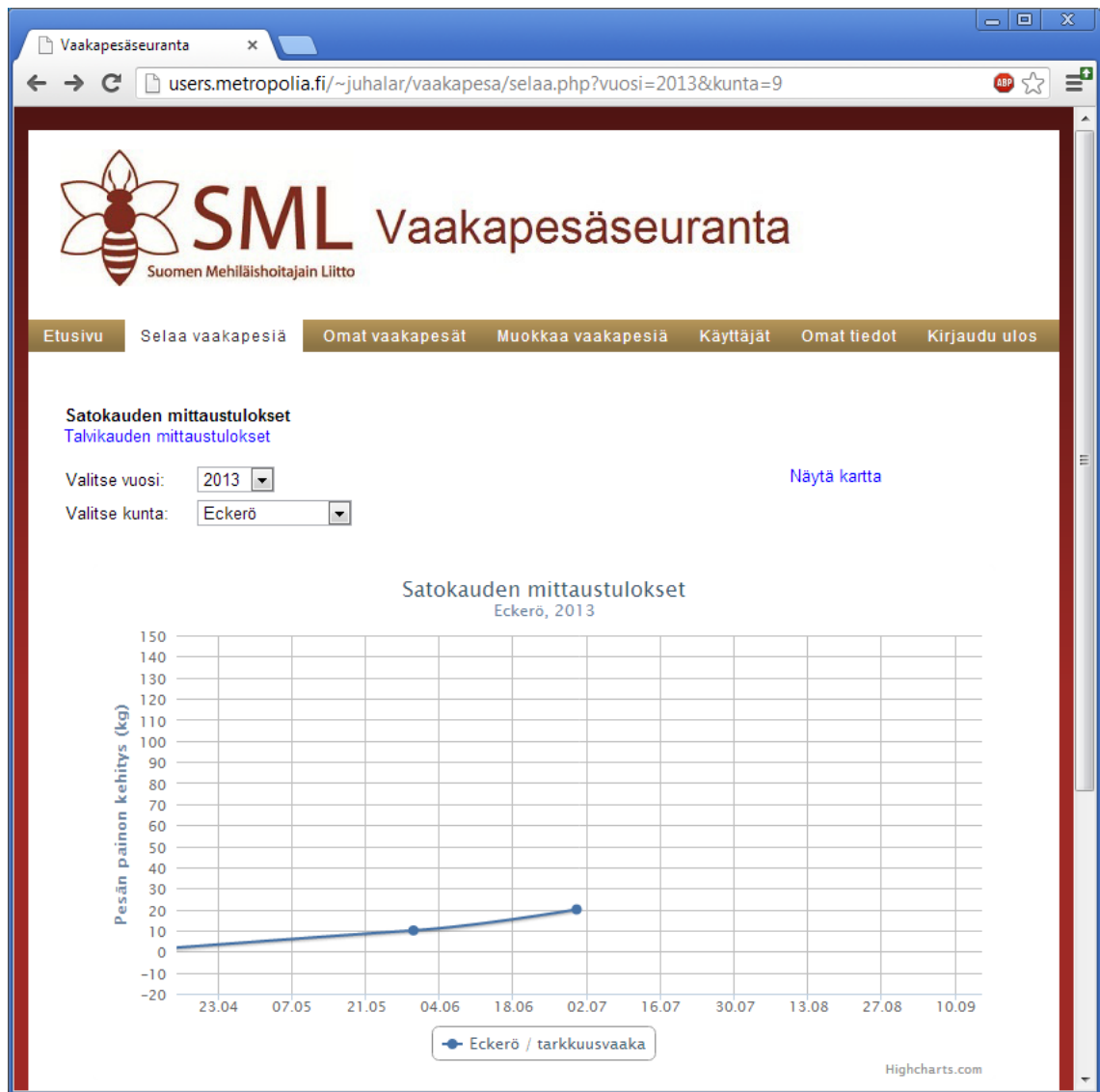
Vaakapesäsovellus on turvassa hakkereilta, sillä kirjautumattomalla käyttäjällä ei ole pääsyoikeutta kuin etusivulle, vaakapesien selaukseen sekä kirjautumissivulle. Ainakin näin luulin. Löysin nimittäin tätä lukua kirjoittaessani (ja asioihin perehtyessäni) vakavan tietoturvariskin, johon kirjautumattomalla käyttäjällä on pääsy. Kerron siitä lisää alaluvussa 2.9.1. Sovelluksen tietoturvan tarkistamiseksi löytyy paljon internet-sivuja, joissa on tarkistuslistoja tietoturva-aukkojen paikkaamiseen. Itse kävin tätä lukua kirjoittaessani läpi seuraavan tarkistuslistan: <http://phpmaster.com/top-10-php-security-vulnerabilities/> ja samalla paikkasin vaakapesäsovelluksen tietoturva-aukkoja. Käyn tässä luvussa läpi yleisimpiä tietoturvariskejä web-sovelluksessa.

2.9.1 SQL-injektio

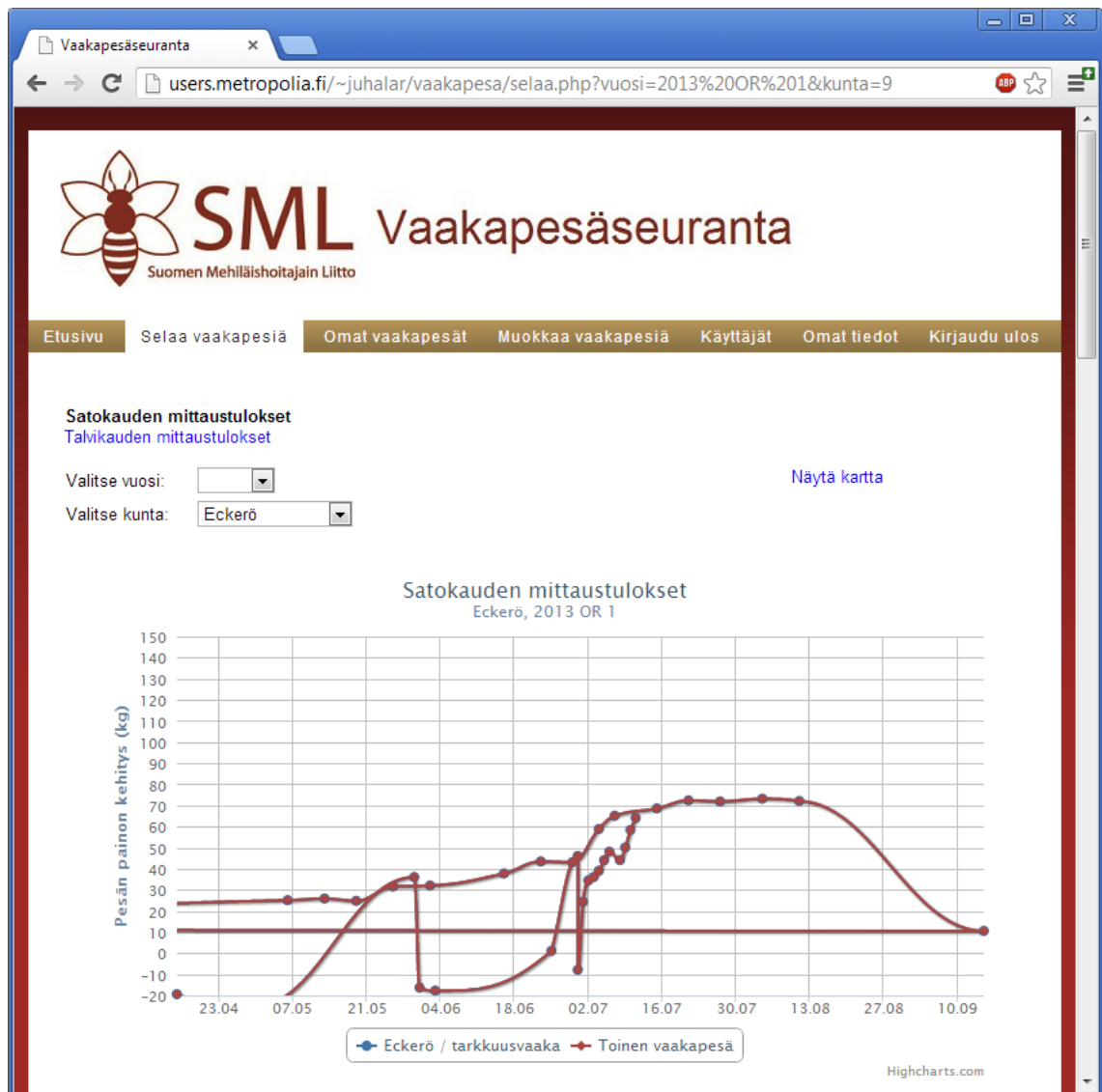
SQL-injektioilla tarkoitetaan SQL-lauseiden ujuttamista alkuperäisen SQL-lauseen sekaan. SQL-injektio on erittäin vaarallinen tietoturvariski, sillä sen avulla vihamielinen hakkeri voi halutessaan poistaa dataa tai jopa suorittaa palvelintason komentoja. Löysin vaakapesäsovelluksestani peräti kaksi mahdollista tapaa suorittaa SQL-injektio ilman kirjautumista sisään. Ensinnäkin kirjautumissivu hakee käyttäjän syöttämän tunnuksen ja salasanan perusteella SQL-lauseella tiedon siitä, onko tunnus/salasana-

pari oikein. Tässä vihamielinen käyttäjä voisi syöttää kirjautumislomakkeeseen tunnuksiksi tai salasanaksi SQL-osalauseen.

Toinen paikka hakkerointiin on vaakapesien selaus -sivu, jossa sivu pyytää GET:llä vuoden ja kunnan. Jälleen kerran vihamielinen käyttäjä voisi syöttää selaimeen vuoden tai kunnan tilalle jonkin haitallisen SQL-osalauseen. Kokeilin vanhalla versiolla vaakapesäsovelluksesta (jossa SQL-injektiosta ei ole suojauduttu) muokata selaimen osoiterivillä olevia parametreja. Korvasin kohdan "vuosi=2013" merkkijonolla "vuosi=2013 OR 1" (selain korvasi välilyönnit merkkijonolla "%20"). Tämä " OR 1" -lisäys muutti SQL-lauseen merkitystä siten, että vuodella ei ole enää väliä ja sovellus hakee tietokannasta kaikkien vuosien mittaustulokset. Seuraavat kuvat (kuva 17 ilman SQL-injektiota ja kuva 18 SQL-injektiolla) havainnollistavat tätä suhteellisen harmitonta SQL-injektiota.



Kuva 17. Näkymä ilman SQL-injektiota



Kuva 18. Näkymä SQL-injektiolla

SQL-injektioilta on onneksi melko helppo suojautua. Eräs keino suojautua injektioilta on karsia kaikki vihamieliset syötteet pois PHP-komennolla `mysql_real_escape_string()` sekä muistaa laittaa jokaiseen mysql-lauseeseen lisättävään käyttäjän syötteeseen lainausmerkit. Seuraavassa esimerkki `mysql_real_escape_string`-komennon käytöstä SQL-injektioilta suojautumiseksi:

```
$query = "SELECT * FROM kayttaja "
        ."WHERE tunnus = '".mysql_real_escape_string($_POST["tunnus"])."' "
        ."AND salasana = md5('".mysql_real_escape_string($_POST["salasana"])."') "
        ."LIMIT 1";
$result = mysql_query($query);
$kayttaja = @mysql_fetch_object($result);
```

Toinen vaihtoehtoinen tapa suojautua SQL-injektioilta PHP-koodissa on käyttää valmisteltuja lauseita (Prepared statements). Valmistelluissa lauseissa muuttujan nimi korvataan kysymysmerkillä. Kysymysmerkin tilalle syötettävä arvo määritellään bindParam-komennolla. Valmistelluissa lauseissa voi myös käyttää kysymysmerkin tilalla nimiparametria. Nimiparametrit merkitään nimekkeellä, jonka eteen laitetaan kaksoispiste (esim. :tunnus). Tällöin bindParam-komentoa käytetään seuraavanlaisesti: \$lause->bindParam(':tunnus', \$_POST["tunnus"]);. Valmisteltujen lauseiden parametreja (kysymysmerkit tai nimiparametrit) ei kuulu laittaa lainausmerkkien sisään, sillä PHP-ohjain laittaa lainausmerkit automaattisesti. Seuraavassa on esimerkki valmistellun lauseen käytöstä kysymysmerkkiparametreilla:

```
$valmisteltulause = $dbh->prepare("SELECT * FROM kayttaja "
    . "WHERE tunnus = ? AND salasana = md5 ( ? ) LIMIT 1");
$valmisteltulause->bindParam(1, ($_POST["tunnus"]);
$valmisteltulause->bindParam(2, ($_POST["salasana"]);
$valmisteltulause->execute();
$kayttaja = $valmisteltulause->fetch();
```

Esimerkissä \$dbh on PDO-objekti (PHP Data Object), jolle luotaessa syötetään tietokantaan liittyvät tiedot (osoite, tietokannan nimi, tunnus, salasana).

Vaakapesäsovelluksessa on käytetty ensimmäistä tapaa (mysql_real_escape_string), sillä se on yksinkertaisempi ja helpompi ottaa käyttöön. Tässä tavassa törmäsin kuitenkin yhteen heikkouteen: null-parametreja syöttäessä ei voi käyttää lainausmerkkejä (joita tietoturvan maksimoimiseksi pitäisi käyttää). Ohjelma tulkitsee null-arvon tekstinä, kun siinä on lainausmerkit. Tilanteisiin, joissa null-arvoja saatetaan tallentaa ja joissa null-arvon tallennus todellakin on välttämätöntä, käytin mysql_real_escape_string-komentoa ilman ympäröiviä lainausmerkkejä.

2.9.2 XSS (Cross Site Scripting)

Cross Site Scripting on eräs web-sivujen tietoturva-aukkoja hyödyntävä menetelmä. XSS hyödyntää mahdollisuutta syöttää web-sivuilla näkyvän datan joukkoon HTML- tai JavaScript-koodia. Vihamielisellä XSS-koodilla voisi esimerkiksi laittaa sivun suorittamaan ulkopuolella olevaa JavaScript-koodia, joka voisi sitten avata käyttäjän selaimeen valekirjautumisruudun tai tonkia istunnon evästetiedot talteen.

XSS-haavoittuvuudet jaetaan ryhmiin DOM-pohjaiset, pysyvät ja ei-pysyvät. Ei-pysyvissä haitallinen koodi ujutetaan sivulle sivun osoitteen parametreissa. Käyttäjä täytyy tämän jälkeen huijata avaamaan vihamielisen koodin sisältävä osoite, joka annetaan esimerkiksi sähköpostin, keskustelupalstojen tai pikaviestimien avulla. Pysyvissä XSS-haavoittuvuuksissa haitallinen koodi tallennetaan kohdesivulle esimerkiksi sivulla olevien lomakkeiden avulla. Tämän tyyppiset haavoittuvuudet ovat vaarallisimpia, sillä ne kohdistuvat kaikkiin sovelluksen pahaa aavistamattomiin käyttäjiin. [12.]

Cross Site Scriptingiltä voi suojautua PHP:ssa käyttämällä käyttäjäsyötteen tulosteisiin htmlentities()-funktia. htmlentities() muuttaa merkkijonossa kaikki erikoismerkit (kuten tagit < ja >) sellaiseen muotoon, että selain tulostaa ne vain merkkeinä eikä tulkitse niitä esimerkiksi html- tai javascript-koodina. htmlentities()-funktia onkin hyvä käyttää kaikkeen PHP-koodissa tapahtuvaan käyttäjän syötteen tai tietokannan datan tulostamiseen.

Internetistä löytyy paljon käytännön esimerkkejä XSS:n käytöstä (testaustarkoitukseen). Itse käytin vaakapesäsovelluksen tietoturvan testaamiseen erästä XSS Filter Evasion Cheat Sheetiä: https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet. Tältä sivulta löytyy lukuisia XSS-esimerkkejä, joita voi omaan sovellukseen kokeilla turvallisesti. Ensimmäisen kerran näitä testatessani totesin, että vaakapesäsovellus vuotaa kuin seula. Esimerkiksi niinkin yksinkertainen koodirivi kuin "<SCRIPT SRC=http://ha.ckers.org/xss.js></SCRIPT>" toimi, kun sen syötti mittaustuloksen lisäys -lomakkeen kommentit-kenttään. Tämä lyhyt lause sai aikaan harmittoman (mutta turhauttavan) popup-ikkunan, joka ponnahtaa esiin jokaisella sivulla, jossa tulostetaan kyseisen mittaustuloksen kommentit (esimerkiksi Selaa vaakapesiä -sivulla). htmlentities() on varsin tehokas suodatin, sillä sen avulla sain kaikki edellä mainitun sivuston esimerkit muutettua toimimattomiksi.

Highcharts-kuvaajan tulosteissa jouduin kuitenkin korvaamaan htmlentities()-funktion htmlspecialchars()-funktiolla ääkkösongelmien takia. htmlspecialchars() toimii samaan tapaan kuin htmlentities(), mutta se muokkaa merkkijonosta vähemmän erikoismerkkejä jättäen esimerkiksi ääkköset rauhaan. Lisäksi törmäsin vaakapesäsovellusta siirrettäessä uudelle palvelimelle toisenlaiseen ääkkösongelmaan htmlentities()-funktion kanssa. Jostain syystä tällä uudella palvelimella kaikki ääkköset php-tulosteissa, joissa käytin htmlentities()-funktia, tulostuivat väärin. Tämä

todennäköisesti johtuu erilaisista palvelimen asetuksista. Ongelman sai kuitenkin korjattua käyttämällä htmlentities-funktiossa kolmatta parametria, joka määrittelee merkitölkoodauksen (character encoding). Koodissa käytin htmlentities()-funktiota lopulta seuraavanlaisesti:

```
echo htmlentities($row['etunimi'], ENT_QUOTES, "UTF-8") . " " . htmlentities($row['sukunimi'], ENT_QUOTES, "UTF-8");
```

Toinen parametri määrittelee, miten erikoismerkkien korvaus tapahtuu (tai mitä erikoismerkkejä korvaus koskee). Mikäli parametrin jättää antamatta, htmlentities()-funktiossa käytetään ENT_COMPAT:ia. Taulukkoon 8 on listattu toiseen parametriin syötettävät vaihtoehdot.

Taulukko 8: htmlentities()-funktion toisen parametrin asetusvaihtoehdot [13]

ENT_COMPAT (1)	Konvertoidaan kaksinkertaiset lainausmerkit (") ja jätetään yksinkertaiset lainausmerkit (') rauhaan.
ENT_QUOTES (1)	Konvertoidaan sekä kaksinkertaiset että yksinkertaiset lainausmerkit.
ENT_NOQUOTES (1)	Ei konvertoida mitään lainausmerkkejä.
ENT_IGNORE (2)	Ohitetaan virheelliset koodin yksikköosat sen sijaan, että palautettaisiin tyhjä merkkijono. Tämän asetuksen käyttöä ei suositella, sillä siinä voi piillä tietoturvariskejä.
ENT_SUBSTITUTE (2)	Korvataan virheelliset koodin yksikköosat Unicode-korvausmerkeillä U+FFFD (UTF-8) tai � sen sijaan, että palautettaisiin tyhjä merkkijono.
ENT_DISALLOWED (2)	Korvataan virheelliset koodipisteet annetun dokumenttityypin mukaisesti Unicode-korvausmerkeillä U+FFFD (UTF-8) tai � sen sijaan, että jätettäisiin ne huomioimatta. Tämä voi olla hyödyllistä esimerkiksi varmistamaan, että XML-koodi, jossa on sulautettua lisäsisältöä, noudattaa syntaksisääntöjä.
ENT_HTML401 (3)	Käsittele koodia HTML 4.01:n mukaisesti.
ENT_XML1 (3)	Käsittele koodia XML 1:n mukaisesti.
ENT_XHTML (3)	Käsittele koodia XHTML:n mukaisesti.
ENT_HTML5 (3)	Käsittele koodia HTML 5:n mukaisesti.

Laitoin jokaisen asetuksen perään sulkujen sisälle numeron, joka jaottelee eri asetusvaihtoehdot ryhmiin. Vaihtoehdoista voi valita useampia, mutta vain yhden samaa ryhmää kohden. Useammat asetukset erotetaan toisistaan tai-merkillä (|). Seuraavassa lyhyt esimerkki tällaisesta useamman asetuksen valinnasta:

```
echo htmlentities($teksti, ENT_COMPAT | ENT_DISALLOWED | ENT_HTML5, "UTF-8");
```

2.9.3 Etätiedostojen käyttö

Etätiedostojen käyttö (Remote File Inclusion) voi aiheuttaa vakavan tietoturvariskin, jos etätiedoston palvelin onnistutaan hakkeroimaan. Esimerkiksi tilanteessa, jossa web-sovellus käyttää etäosoitteessa olevaa tiedostoa `www.jokusivu.com/jquery/1.8.2/jquery.min.js` (kuvitteellinen domain), on vaaransa, mikäli hakkerit löytävät tietoturva-aukon `jokusivu.com`-palvelimelta. Mikäli hakkerit pääsevät muuttamaan `jokusivu.com`-palvelimen tiedostoja, he voivat korvata `jquery.min.js`:n sisällön vihamielisellä koodilla, minkä jälkeen myös oma web-sovelluksemme tietoturva on erittäin vakavassa vaarassa. Etätiedostojen käytöllä on myös se ikävä ominaisuus, että mikäli etätiedosto poistetaan tai sen palvelin kaatuu, myös oma web-sovelluksemme halvaantuu ainakin osittain.

Hakkerit voivat myös muuttaa sovelluksen käyttämään haitallisia etätiedostoja, mikäli sovelluksessa on käytetty esimerkiksi seuraavanlaista ratkaisua:

```
<?php
$sivu = $_GET['sivu'];
require $sivu . '.php';
?>
```

Kyseinen koodinpalanen on siis yksinkertainen kontrolleri, ja eri sivut ladataan `index.php?sivu=sivunnimi-osoitteilla`. Tässä tapauksessa hakkeri voi kuitenkin syöttää osoitteen `index.php?sivu=http://www.jokusivu.com/haitallinenkoodi.php`. Jotta tämä haitallinen PHP-koodi latautuisi, täytyy hakkerin hallinnoimassa `jokusivu.com`:ssa olla PHP-tuki säädettynä pois päältä. [14.]

Etätiedostojen käytön uhkaa ei ole vaakapesäsovelluksessa, sillä edellä mainitun mukaista kontrollerirakennetta ei ole sovelluksessa. Myöskään ulkoisella palvelimella olevia JavaScript-tiedostoja ei käytetä sovelluksessa, sillä päätin jo hyvissä ajoin kopioida kaikki tarvittavat JavaScript-tiedostot (`highcharts.js`, `jquery.min.js`) samaan kansioon web-sovelluksen kanssa. Tämä poistaa tietoturvauhkan lisäksi sen riskin, että sovellus halvaantuisi osittain, mikäli etäpalvelin sattuisi kaatumaan tai etäpalvelimen JavaScript-tiedostot jostain syystä poistettaisiin.

2.9.4 Tallennetut salasanat

Koskaan ei voi täysin varmuudella olla turvassa hakkereilta, mikäli sovellus on julkisella palvelimella, sillä hakkerit keksivät uusia tietoturva-aukkoja web-teknologioista jatkuvasti. Tietokantaan tallennettavat salasanat on hyvä kryptata varmuuden vuoksi. Yksi yleisimmistä keinoista kryptata salasanat on käyttää MD5-algoritmia. MD5 on yksisuuntainen hajakoodausalgoritmi, mikä tarkoittaa käytännössä sitä, että MD5-koodatusta salasanasta on mahdotonta tuottaa alkuperäistä selväkielistä salasanaa.

MD5-algoritmi tuottaa salasanasta (tai mistä tahansa datasta) 128-bittisen tiivisteen. MD5-algoritmissa on kuitenkin eräs heikkous: merkkijonoja, jotka tuottavat saman MD5-tiivisteen on matemaattisesti rajaton määrä. Niitä on kuitenkin erittäin vaikea löytää (vaikka vuonna 2004 löydettiin ensimmäisen kerran kaksi viestiä, jotka tuottavat saman MD5-tiivisteen), joten käytännössä MD5-algoritmia voi pitää turvallisena. [15.]

Jos hakkerit onnistuvat pääsemään käsiksi tietokannan MD5-koodattuihin salasanoihin, ainoa keino yrittää muuntaa MD5-koodatuista salasanoista alkuperäisiä selkokieliä salasanoja on Brute Force -menetelmä. Brute Force -menetelmässä tietokone laitetaan kokeilemaan erilaisia salasanoja joko aakkosjärjestykselliseen kombinaatioon perustuen (esim. AAA, AAB, AAC jne.) tai erilaisia valmiita salanalistoja (esim. tiedosto, johon on lisätty kaikki englanninkielen sanat) läpikäyden. Brute Force -menetelmällä on vaikeaa selvittää salasanoja, jotka eivät koostu selkokielistä sanoista ja joiden pituus on tarpeeksi pitkä (yleensä 8 merkkiä ja sitä pidemmät satunnaisista merkeistä koostuvat salasanat alkavat olla jo riittävän turvallisina).

Jo heti sovelluksen alkuvaiheessa halusin tehdä sovelluksen salasanoiden tallentamisesta turvallisen, joten päädyin käyttämään MD5-koodausta salasanoihin. Vaakapesäsovelluksessa käyttäjien salasanoiden tallentuvat palvelimelle vain MD5-tiivisteet. Käyttäjän kirjautuessa sivustolle käyttäjän syöttämän salasanan MD5-tiivistettä verrataan tietokannassa olevaan. Mikäli ne ovat identtiset, käyttäjän kirjautuminen hyväksytään.

3 Tulokset ja jatkokehitys

Vaakapesäsovelluksesta tuli mielestäni onnistunut ja tätä kirjoittaessani siihen on lisätty jo ensimmäisiä oikeita käyttäjiä. Vaakapesäsovellusta on ollut mielenkiintoista tehdä ja olen tyytyväinen lopputulokseen. Opinnäytetyöraporttia kirjoittaessa selvisi paljon uusia erityisesti tietoturvaan liittyviä seikkoja, ja raportin kirjoitusprosessin yhteydessä sovelluksen tietoturvaa sain parannettua huomattavasti. Lisäksi, kuten aina sovelluskehityksessä, vaakapesäsovelluksessa ilmeni muutamia bugeja, jotka olen korjannut.

Kaikki esimieheni toivomat ominaisuudet sain toteutettua ilman suurempia ongelmia. Vaakapesäsovelluksesta tuli tyylikäs ja helppokäyttöinen, joka olikin alkuperäinen tavoite. Lisäksi sovelluksen ylläpito on helppoa ja suoraviivaista, eikä vaadi tietoteknistä osaamista. Pienenä huolenaiheena on tosin sovelluksen datan varmuuskopiointi. Sovelluksessa tai palvelimella ei ole automaattista varmuuskopiointitoiminnallisuutta. PhpMyAdmin-työkalulla voi manuaalisesti ladata SQL-varmuuskopion tietokannasta omalle tietokoneelle, mutta työkalun käyttö ei ole aivan yksinkertaista ohjelmistotekniikkaa tuntemattomalle.

Vaakapesäsovelluksen kehittämisen alussa tein erään virheen: en käyttänyt sovelluksen ylävalikon (ja alaosan copyright-merkinnän) näyttämiseen erillistä template-tiedostoa. Tämä johtaa siihen, että mikäli vaakapesäsovelluksen ylävalikkoa halutaan muuttaa, täytyy sama muutos tehdä lukuisiin eri tiedostoihin. Vaakapesäsovelluksen muuttaminen siten, että se käyttäisi template-tiedostoja, olisi melko työlästä näin jälkikäteen, sillä lähes jokaista PHP-tiedostoa joutuisi muokkaamaan. Tämä pieni kauneusvirhe ei kuitenkaan vaikuta sovelluksen käyttöön mitenkään.

Vaakapesäsovellus on jo tällaisenaan erittäin toimiva, mutta sen toiminnallisuutta voidaan jatkossa kehittää edelleen. Seuraavassa on listattu jatkokehitysideoita:

- Vaakapesäsovelluksen mittausdatan ja kuvaajien tulostamista ei ole vielä tätä kirjoittaessa pohdittu yhtään. Mittausdataa ja kuvaajia olisi hyvä saada avattua erilliseen ikkunaan, josta tulostaminen (ilman mitään ylimääräistä) olisi helppoa. Tämän toteuttaminen olisi melko yksinkertaista, joten saattaa olla, että toteutan tämän vielä ennen työsuhteeni päättymistä.

- Olisi hyödyllistä, jos vaakapesäsovelluksesta saisi mittausdataa ladattua esimerkiksi Excel- tai tekstitiedostona omalle koneelle. Tämä voisi toimia jopa eräänlaisena keinona varmuuskopioida dataa, sillä sovelluksessa ei ole tällä hetkellä mitään varmuuskopiointitoiminnallisuutta.
- Edellistä jatkokehitysideaa jatkaen: sovelluksessa olisi hyvä olla helppokäyttöinen varmuuskopiointitoiminto (tai mieluummin vieläpä automatisoitu), jonka avulla tärkeää mittausdataa ei pääsisi katoamaan, mikäli palvelimella tapahtuu jotain ikävää.
- Automaattivaakapesien ohjelmoimisessa olisi myös työtä, mutta niiden rakentaminen saattaa vaatia myös rautapuolen osaamista. Joka tapauksessa palvelinpään tietoa vastaanottava osa on jo olemassa ja käytettävissä.
- Esimieheltäni tuli vielä eräs jatkokehitysidea: pesästä löydettyjen varroapunkkien lukumäärästä olisi hyvä voida pitää kirjaa. Varroapunkit ovat loiseläimiä, jotka saattavat vähitellen tappaa koko mehiläisyhdyskunnan, mikäli niitä ei torjuta niiden ilmaantuessa pesään. Löydettyjen varroapunkkien lukumäärä voisi olla valinnainen kenttä mittaustuloksia syötettäessä, ja se voisi näkyä esimerkiksi kuvaajan tooltipissä samaan tapaan kuin satokasvit.

4 Yhteenveto

Opinnäytetyössä kehitettiin helppokäyttöinen web-sovellus vaakapesäseurantaa varten. Opinnäytetyöraportissa tutustuttiin yksityiskohtaisesti vaakapesäsovelluksen rakenteeseen ja toiminnallisuuksiin. Lisäksi raportissa esiteltiin lyhyesti vaakapesäsovelluksessa käytettyjä web-teknologioita sekä selvitettiin sovellukseen liittyviä tietoturvariskejä.

Vaakapesäsovelluksen kehittämisessä ei tullut vastaan suurempia ongelmia. Suurimmaksi haasteeksi osoittautui kuntakartan toteutus, jota jo alun perin osasin arvella melko haasteelliseksi. Kuntakartan ei kuitenkaan tarvinnut olla huipputarkka, joten sain vältettyä trigonometristen yhtälöiden käytön yksinkertaisemmalla matematiikalla. Taulukkolaskentaohjelma Microsoft Excelistä oli suuri apu kuntakartan pisteiden sijainteja laskiessa.

Mikäli vaakapesäsovellusta ryhdytään SML:n toimesta joskus jatkokehittämään, tarjoaa tämä dokumentti hyvän kuvauksen vaakapesäsovelluksesta sovelluskehittäjän näkökulmasta. Lisäksi tästä dokumentista voi olla apua esimerkiksi Highcharts-kirjastoon tutustuville tai PHP-sovellusten tietoturvan parantamisesta kiinnostuneille.

Lähteet

- 1 Mehiläispesä - Wikipedia. Verkkodokumentti.
<<http://fi.wikipedia.org/wiki/Mehiläispesä>>. Päivitetty 24.3.2013. Luettu: 19.4.2013.
- 2 Scrum - Wikipedia. Verkkodokumentti. <<http://fi.wikipedia.org/wiki/Scrum>>. Päivitetty 16.4.2013. Luettu 19.4.2013.
- 3 A History of HTML. 2008. Verkkodokumentti.
<<http://www.w3.org/People/Raggett/book4/ch02.html>>. Luettu 17.4.2013.
- 4 HTML - Wikipedia. Verkkodokumentti. <<http://en.wikipedia.org/wiki/HTML>>. Päivitetty: 19.4.2013. Luettu: 19.4.2013.
- 5 HTML - Wikipedia. Verkkodokumentti. <<http://fi.wikipedia.org/wiki/HTML>>. Päivitetty: 7.3.2013. Luettu: 17.4.2013.
- 6 JavaScript - Wikipedia. Verkkodokumentti.
<<http://en.wikipedia.org/wiki/JavaScript>>. Päivitetty: 18.4.2013. Luettu 19.4.2013.
- 7 JavaScript (ja vastaavat). Verkkodokumentti.
<<http://www.cs.tut.fi/~jkorpela/webjulk/3.2.html>>. Päivitetty 21.6.2009. Luettu 19.4.2013.
- 8 CSS - Wikipedia. Verkkodokumentti.
<http://en.wikipedia.org/wiki/Cascading_Style_Sheets>. Päivitetty 18.4.2013. Luettu 19.4.2013.
- 9 CSS-ohje. Verkkodokumentti.
<<http://www.cs.uta.fi/ipoppla/www/ipoppla03/ka/tohje/tyylioh.htm>>. Luettu 19.4.2013.
- 10 PHP - Wikipedia. Verkkodokumentti. <<http://fi.wikipedia.org/wiki/PHP>>. Päivitetty 15.3.2013. Luettu 19.4.2013.
- 11 MySQL - Wikipedia. Verkkodokumentti. <<http://en.wikipedia.org/wiki/MySQL>>. Päivitetty 19.4.2013. Luettu 19.4.2013.
- 12 Cross site scripting - Wikipedia. Verkkodokumentti.
<http://fi.wikipedia.org/wiki/Cross_site_scripting>. Päivitetty 26.3.2013. Luettu 19.4.2013.

- 13 PHP: htmlentities - Manual. Verkkodokumentti.
<<http://php.net/manual/en/function.htmlentities.php>>. Päivitetty 19.4.2013. Luettu 19.4.2013.
- 14 PHP Freaks - PHP Help Tutorial: PHP Security. 2008. Verkkodokumentti.
<<http://www.phpfreaks.com/tutorial/php-security>>. Luettu 19.4.2013.
- 15 MD5 - Wikipedia. Verkkodokumentti. <<http://fi.wikipedia.org/wiki/MD5>>. Päivitetty 7.3.2013. Luettu 19.4.2013.

VAAKAPESÄSEURANNAN KÄYTTÖOHJEET



Sisällys

1	Lyhyesti vaakapesäsovelluksesta	3
2	Vaakapesätietojen selaus	3
3	Kirjautuminen tunnuksilla	8
4	Omien tietojen tarkistus	9
5	Omat vaakapesät	9

1 Lyhyesti vaakapesäsovelluksesta

Vaakapesäsovellus on Suomen Mehiläishoitajain Liiton kehittämä web-pohjainen sovellus mehiläispesien vaakadatan tallentamiseen ja tarkasteluun. Vaakapesätietojen selaus on kaikille avoin, mutta vaakapesädatan lisäämiseen tarvitset tunnukset järjestelmään. Vaakapesäsovellus löytyy osoitteesta <http://koti.tnnet.fi/web144/vaakapesa/>.

2 Vaakapesätietojen selaus

Vaakapesien selaukseen pääsee klikkaamalla ylävalikosta ”Selaa vaakapesiä”. Talvikauden pesien selaukseen pääsee halutessaan klikkaamalla ”Talvikauden mittaustulokset” -linkkiä. Aktiivisena oleva talvi- tai satokausi näkyy sivulla lihavoituna mustalla. Aina ”Selaa vaakapesiä” -sivulle tultaessa aluksi on aktiivisena satokausi, joten edellä mainituista asioista ei tarvitse välittää, mikäli haluaa selata satokauden mittaustuloksia. Alasvetovalikosta valitaan ensin haluttu vuosi (listauksessa näkyvät vain ne vuodet, joilta löytyy järjestelmästä mittaustuloksia). Kun vuosi on valittu, ilmestyy sivulle Suomen kartta, jossa näkyvät mustina neliöinä ne kunnat, joilta löytyy valitulta vuodelta mittaustuloksia.

[Etusivu](#) [Selaa vaakapesiä](#) [Kirjaudu sisään](#)

Satokauden mittaustulokset
[Talvikauden mittaustulokset](#)

Valitse vuosi:

Valitse kunta:



Kuntien nimiä saa näkyviin siirtämällä hiiren osoittimen halutun neliön kohdalle. Nyt käyttäjä voi klikata haluamaansa kuntaa kartalta tai vaihtoehtoisesti valita kunnan vasemmalla olevasta alasvetovalikosta.

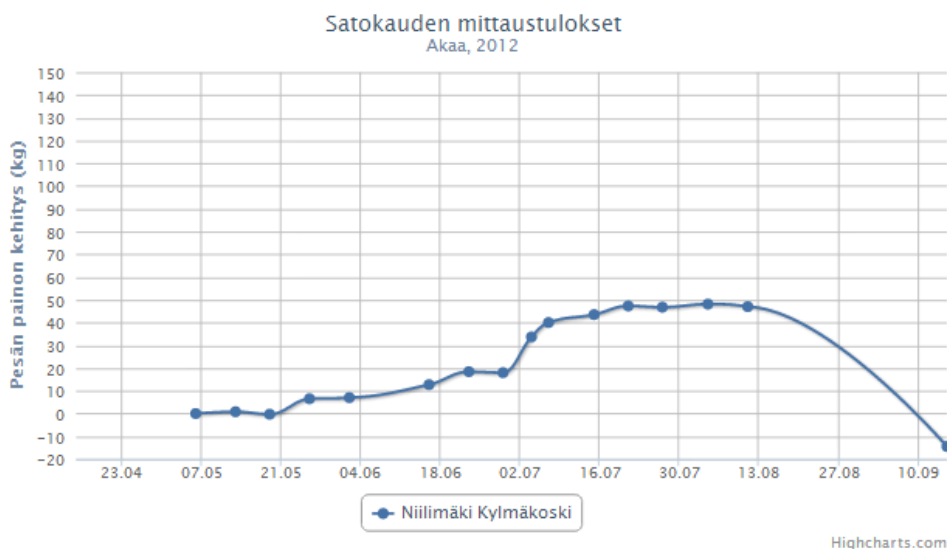
Kun kunta on valittu, kartta katoaa ruudulta ja sivulle ilmestyy kuvaaja sekä kunnan vaakapesät mittaustuloksineen valitulta vuodelta. Alla oleva kuva havainnollistaa, miltä sivu voi näyttää tässä vaiheessa.

Satokauden mittaustulokset
Talvikauden mittaustulokset

Valitse vuosi: 2012

[Näytä kartta](#)

Valitse kunta: Akaa



Niilimäki Kylmäkoski (Tarhaaja: Teppo Testaaja)

Mittauspäivä	Mittaustulos (kg)	Mittaustulos muutoksen jälkeen (kg)	Painonmuutos	Kumulatiivinen muutos	Kommentit
06.05.2012	35,1		0,0	0,0	
13.05.2012	35,9		0,8	0,8	
19.05.2012	34,8	44,4	-1,1	-0,3	Laatikko lisää. Erilaisia havaintoja löytyy. Akaan kaupungin etusivulla on hieno makea kaupunki

Kuvaaja ilmestyy sivulle vain, jos jossakin kunnan vaakapesässä on kaksi tai enemmän painomittauksia kyseiseltä vuodelta. Kuvaajan pisteistä saa tarkempia tietoja siirtämällä hiiren osoittimen jonkin kuvaajan pisteen päälle. Tällä tavalla kuvaajan pisteestä saa näkyviin päivämäärän, painon sekä mahdolliset satokasvit (mikäli niitä on annettu mittaustulosta tallennettaessa).

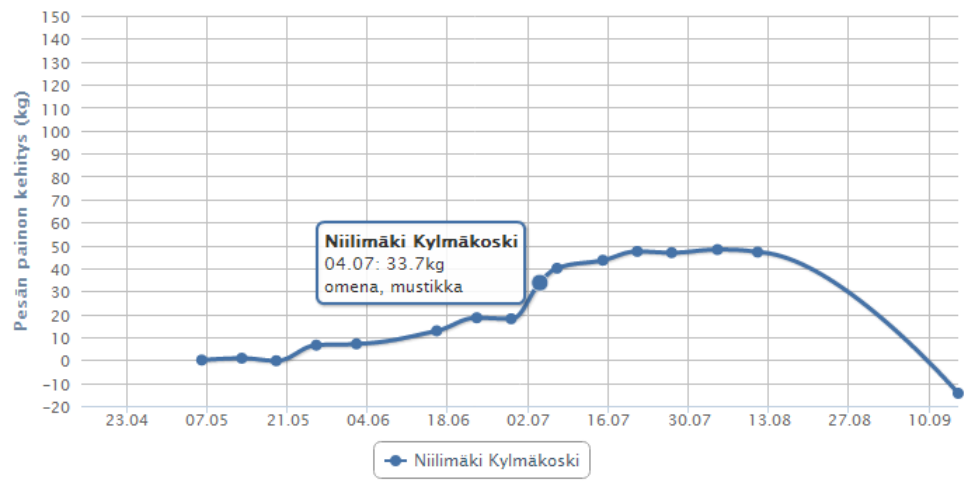
Satokauden mittaustulokset
[Talvikauden mittaustulokset](#)

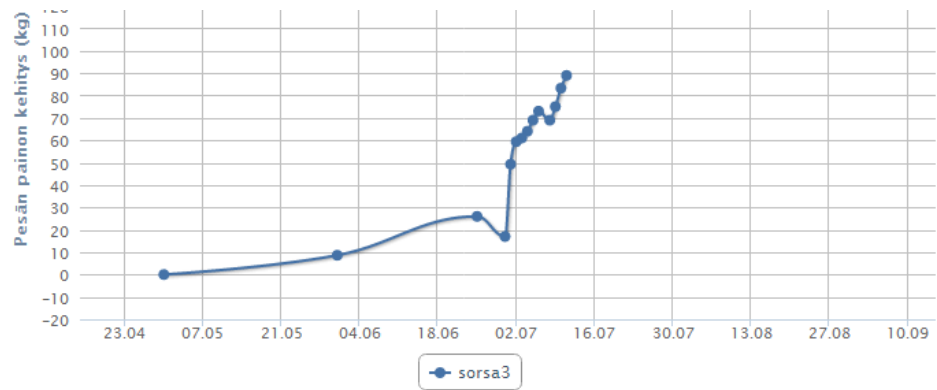
Valitse vuosi: 2012 ▼

[Näytä kartta](#)

Valitse kunta: Akaa ▼

Satokauden mittaustulokset
Akaa, 2012





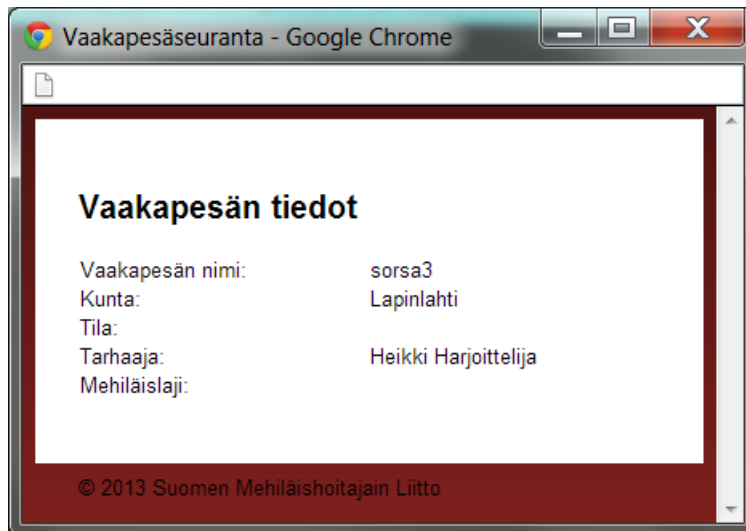
sorsa3 (Tarhaaja: Heikki Harjoittelija)

Mittauspäivä	Mittaustulos (kg)	Mittaustulos muutoksen jälkeen (kg)	Painonmuutos	Kumulatiivinen muutos	Kommentit
30.04.2013	-1,0		0,0	0,0	
31.05.2013	7,6		8,6	8,6	
25.06.2013	25,0		17,4	26,0	
30.06.2013	16,0		-9,0	17,0	
01.07.2013	48,3		32,3	49,3	
02.07.2013	58,4		10,1	59,4	
03.07.2013	60,0		1,6	61,0	
04.07.2013	63,0		3,0	64,0	
05.07.2013	68,0		5,0	69,0	
06.07.2013	72,0		4,0	73,0	
08.07.2013	68,0		-4,0	69,0	
09.07.2013	74,0		6,0	75,0	
10.07.2013	82,3		8,3	83,3	
11.07.2013	88,0		5,7	89,0	

[Lisätietoja pesästä](#)

© 2013 Suomen Mehiläishoitajain Liitto

Kunkin vaakapesän mittaustulosten (tekstimuotoinen esitys) alla on "Lisätietoja pesästä" -linkki (ylläoleva kuva). Sitä klikkaamalla aukeaa uusi ikkuna, jossa on listattuna tiedot vaakapesästä. Tila ja mehiläisrotu ovat vapaaehtoisia tietoja vaakapesää luodessa, ja näin ollen niitä ei aina ole annettu.



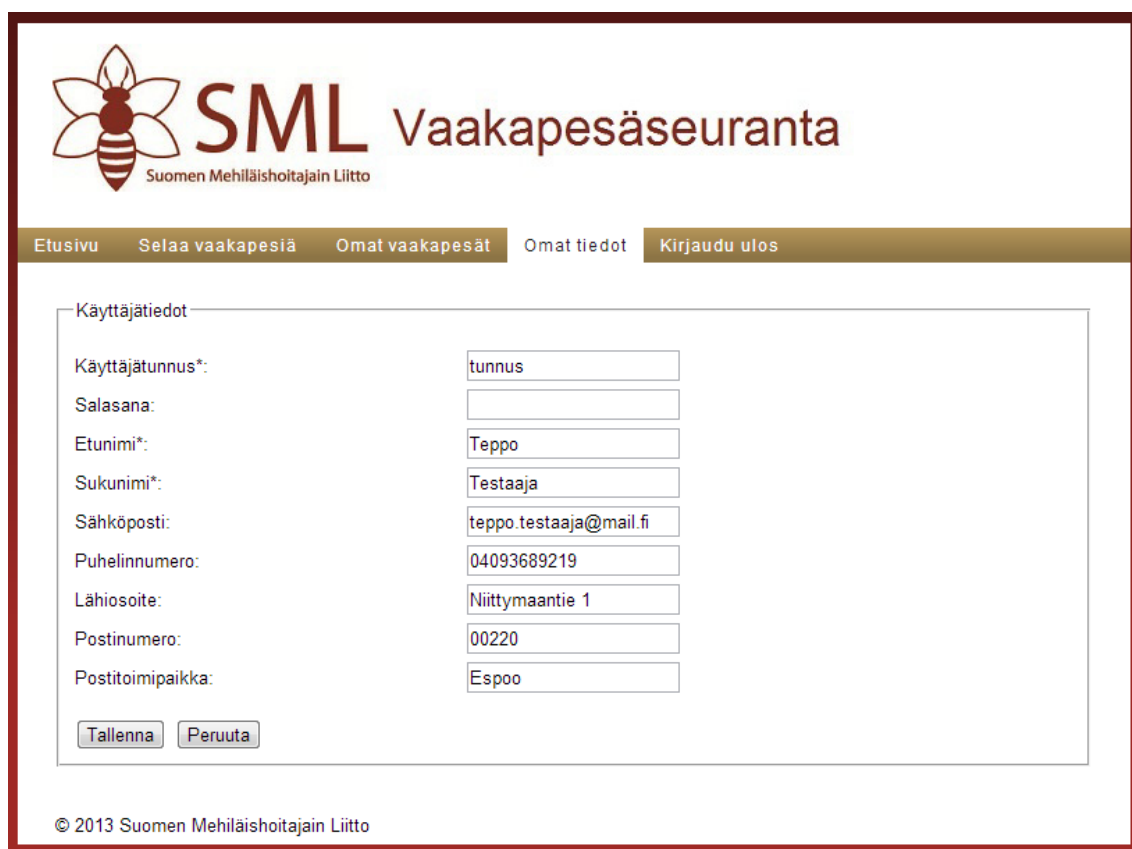
3 Kirjautuminen tunnuksilla



Kirjautuessasi sisään, sinulla täytyy olla olemassaoleva käyttäjätunnus sekä salasana. Tunnus ja salasana ovat merkkikokoriippuvaisia, mikä tarkoittaa sitä, että kirjain "A" ei ole sama asia kuin kirjain "a". Mikäli olet unohtanut salasanasi, voit ottaa yhteyttä sml@hunaja.net, jotta salasanasi voitaisiin vaihtaa uuteen.

4 Omien tietojen tarkistus

Ensimmäisen kerran kirjautuessa on hyvä tarkistaa, että omat tiedot ovat kunnossa. Omiin tietoihin pääsee ”Omat tiedot”-valikon kautta. Omissa tiedoissa salasanaa ei näytetä, mutta sen voi vaihtaa syöttämällä siihen uuden haluamansa salasanan. Mikäli salasana-kentän jättää tyhjäksi tiedot tallentaessa, säilyy salasana entisellään. Tähdellä merkityt kohdat ovat pakollisia syöttää. Eli käyttäjätunnus-, etunimi- sekä sukunimi-kenttiä ei voi jättää tyhjiksi.



 **SML** Vaakapesäseuranta
Suomen Mehiläishoitajain Liitto

Etusivu Selaa vaakapesiä Omat vaakapesät Omat tiedot Kirjaudu ulos

Käyttäjätiedot

Käyttäjätunnus*:	tunnus
Salasana:	
Etunimi*:	Teppo
Sukunimi*:	Testaaja
Sähköposti:	teppo.testaaja@mail.fi
Puhelinnumero:	04093689219
Lähiosoite:	Niittymaantie 1
Postinumero:	00220
Postitoimipaikka:	Espoo

© 2013 Suomen Mehiläishoitajain Liitto

5 Omat vaakapesät

Omat vaakapesät -sivulla tarhaaja voi lisätä mittaustuloksia vaakapesän painon kehityksestä. Ensimmäistä kertaa ”omat vaakapesät” -sivua käytettäessä tarhaajan on kuitenkin luotava uusi vaakapesä, sillä vaakapesälista näyttää tyhjää. Uuden vaakapesän luontiin pääsee klikkaamalla ”Lisää vaakapesä” -linkkiä.

**SML** Vaakapesäseuranta
Suomen Mehiläishoitajain Liitto

EtusivuSelaa vaakapesiäOmat vaakapesätOmat tiedotKirjaudu ulos

Omat vaakapesät

Nimi	Toiminnot
------	-----------

[Lisää vaakapesä](#)

© 2013 Suomen Mehiläishoitajain Liitto

Nyt aukeaa lomake, johon tulee syöttää joitakin tietoja vaakapesästä. Tähdellä merkityt kohdat ovat pakollisia eli siis vaakapesän nimeä tai kuntaa ei voi jättää antamatta. Mehiläisrotu-kenttä on vapaaehtoinen ja siihen voi syöttää mehiläisrodusta joko suomen- tai latinankielisen nimen.

**SML** Vaakapesäseuranta
Suomen Mehiläishoitajain Liitto

EtusivuSelaa vaakapesiäOmat vaakapesätOmat tiedotKirjaudu ulos

Lisää vaakapesä

Pesätiedot

Vaakapesän nimi*:

Tilan nimi:

Kunta*:

Mehiläislaji:

TallennaPeruuta

© 2013 Suomen Mehiläishoitajain Liitto

Kun vaakapesän tiedot ovat syötetty ja on klikattu Tallenna-nappia, pitäisi uuden vaakapesän ilmestyä Omat vaakapesät -listaan. Tämän jälkeen mittaustuloksia voi

alkaa syöttämään klikkaamalla vaakapesän kohdalla Avaa-linkkiä. Nyt sivulle aukeaa tyhjä mittaustulos-taulukko (ellet ole jo ehtinyt syöttää mittaustuloksia vaakapesääsi). Mittaustulokset näkyvät aina vuosi kerrallaan, ja ylälaidassa olevasta alasvetovalikosta voi valita haluamansa vuoden mittaustulokset näytettäväksi.

**SML** Vaakapesäseuranta
Suomen Mehiläishoitajain Liitto

EtusivuSelaa vaakapesiäOmat vaakapesätOmat tiedotKirjaudu ulos

Valitse vuosi:

Tarkkuusvaaka / Ylöjärvi

Mittauspäivä	Mittaustulos (kg)	Mittaustulos muutoksen jälkeen (kg)	Kommentit	Toiminnot
--------------	-------------------	-------------------------------------	-----------	-----------

[Lisää mittaustulos](#)

© 2013 Suomen Mehiläishoitajain Liitto

Uuden mittaustuloksen lisäämiseen pääsee klikkaamalla sivun alalaidassa olevaa "Lisää mittaustulos" -linkkiä.

**SIVEL** Vaakapesäseuranta
Suomen Mehiläishoitajain Liitto

EtusivuSelaa vaakapesiäOmat vaakapesätOmat tiedotKirjaudu ulos

Lisää mittaustulos

Vaakadata

Mittaustulos*: kg

Mittaustulos muutoksen jälkeen: kg

Päivämäärä*: syötä muodossa pp.kk.vvvv

Satokasvi:

☐ pajut
☐ voikukka
☐ omena
☐ puolukka
☐ vadelma
☐ horsma
☐ rypsi
☐ apilat
☐ kanerva
☐ muu, mikä?

Kommentit:

Talvi:☐

Tallenna

Peruuta

© 2013 Suomen Mehiläishoitajain Liitto

Lisää mittaustulos -lomakkeessa pakollisia kohtia ovat mittaustulos ja päivämäärä (eli tähdellä merkityt jälleen kerran). Mittaustulos syötetään kilogrammoina siten, että desimaaliosa erotetaan pilkulla tai pisteellä (esim. 93,6). Mikäli mittauspäivänä on tehty jokin painoon vaikuttava muutos, syötetään "Mittaustulos"-kenttään paino ennen muutosta ja "Mittaustulos muutoksen jälkeen"-kenttään paino muutoksen jälkeen. Mutta mikäli mitään painoon vaikuttavaa muutosta ei ole tehty, "Mittaustulos muutoksen jälkeen" -kenttä jätetään tyhjäksi.

Päivämäärä-kenttään syötetään päivämäärä suomalaisessa muodossa pp.kk.vvvv (esim. 16.07.2013). Huom! Etunollat ovat pakollisia päivässä ja kuukaudessa, ja vuosi täytyy ilmoittaa nelinumeroisena. Satokasvi-kohtaan voi rastittaa haluamansa määrä

arvelemiaan satokasveja (jotka sillä hetkellä kukkivat). Satokasveista voi valita jonkin/jotkin seuraavista: pajut, voikukka, omena, puolukka, vadelma, horsma, rypsi, apilat sekä kanerva. Lisäksi voi halutessaan syöttää jonkin muun satokasvin ”muu, mikä?”-kenttään, kunhan muistaa myös laittaa rastin tämän eteen.

Kommentit-kenttään kannattaa kirjoittaa ainakin pesiin kohdistuneista toimenpiteistä (laatikoiden lisäyksestä, hunajan keruusta yms.). Lisäksi kommentteihin voi halutessaan merkitä esimerkiksi säähavaintoja ja muita mehiläispesien kasvuun liittyviä tekijöitä.

Talvi-kohta on tarkoitettu erottelemaan talvikauden pesät satokauden pesistä. Mikäli mehiläispesä on talvehtimassa, ja tarhaaja haluaa tallentaa talvikaudelta mittaustuloksia, täytyy tähän valintaruutuun laittaa rasti. (Huom. Valintaruutu saattaa olla jo valmiiksi rastitettu, mikäli mittaustulosta tallentaessa on palvelimen kellon päivä jotakin väliltä 1.10.-31.3.)

Lopuksi mittaustulos tallennetaan klikkaamalla ”Tallenna”-nappia. Tämän jälkeen lisätty mittaustulos tulee näkyviin mittaustulos-listaan. Mittaustulokset näkyvät päivämäärän mukaisessa järjestyksessä (valitun vuoden vanhin ylimpänä ja uusin alimpana). Mittaustuloslistalla painot pyöristyvät yhden desimaalin tarkkuuteen (eli ne näkyvät 0,1kg tarkkuudella). Mittaustuloksia voi jälkeinpäin muokata klikkaamalla ”Muokkaa”-linkkiä mittaustuloksen kohdalla. Mittaustuloksia voi myös poistaa klikkaamalla ”Muokkaa”-linkin vieressä olevaa ”Poista”-linkkiä.

Ohjelmallinen vaakapesädatan lähettäminen

Voit syöttää vaakadataa järjestelmään ohjelmallisesti, mikäli sinulla on vaadittavat tunnukset sekä pesä luotuna järjestelmään. Tarvitset tähän lisäksi pesäsi ID-numeron. Saat sen selville menemällä vaakapesäsovelluksessa omiin vaakapesiin ja sieltä klikkaamalla halutun pesän kohdalla ”Muokkaa”. Tämän jälkeen sinulle pitäisi aueta sivu, josta näkyy kaikki vaakapesän tiedot. Ylimpänä tiedoista on pesäsi ID-numero. Kirjoita se jonnekin muistiin.

Parametrin nimi	Kuvaus
tunnus	Käyttäjätunnuksesi. Tunnus on pakollinen kenttä.
salasana	Salasanasi. Salasana on pakollinen kenttä.
pesa	Vaakapesäsi ID-numero. Pesa on pakollinen kenttä.
paino	Mittaustulos kilogrammoina (desimaalit erotettuna joko pilkulla tai pisteellä). Paino on pakollinen kenttä.
paiva	Päivämäärä muodossa pp.kk.vvvv. Paiva ei ole pakollinen kenttä. Mikäli päivämäärää ei syötetä, ohjelma merkitsee päivämääräksi tämän päivän.
talvi	Talvikausi vai satokausi? Tähän syötetään 0, jos kyseessä on satokauden mittaustulos ja 1, jos kyseessä on talvikauden mittaustulos. Talvi ei ole pakollinen kenttä. Mikäli talvi-kohdan jättää antamatta, ohjelma valitsee sen automaattisesti nykyhetken mukaan: lokakuu-maaliskuu talvi=1 ja huhtikuu-syyskuu talvi=0.

Esimerkki:

<http://koti.tnnet.fi/web144/vaakapesa/lisaadata.php?tunnus=tunnus&salasana=salasan a&pesa=2&paino=41,2&paiva=08.03.2013>

Esimerkin mukainen osoite kirjautuu järjestelmään käyttäjätunnuksella: tunnus sekä salasanalla: salasana, ja sen jälkeen tallentaa pesään, jonka ID-numero on 2, painon arvolla 41,2kg päivälle 8.3.2013.

Huom! Vaakapesäsovellus ei salli useamman mittaustuloksen lisäämistä samalla päivämäärällä (samaan vaakapesään), joten automaattiset sovellukset on hyvä ajastaa lähettämään uusimpia mittaustuloksia maksimissaan kerran vuorokaudessa.