

Saimaan ammattikorkeakoulu
Tekniikka Lappeenranta
Tietotekniikan koulutusohjelma
Tietojärjestelmien kehitys

Tuomas Porvali

Parsonrussellinterrierit ry:n terveystietokannan verkkokäyttöliittymä

Opinnäytetyö 2015

Tiivistelmä

Tuomas Porvali

Parsonrussellinterrierit ry:n terveystietokannan verkkokäyttöliittymä, 70 sivua

Saimaan ammattikorkeakoulu

Tekniikka Lappeenranta

Tietotekniikan koulutusohjelma

Tietojärjestelmien kehitys

Opinnäytetyö 2015

Ohjaajat: lehtori Martti Ylä-Jussila, Saimaan ammattikorkeakoulu, Heli Korpinen,

Parsonrussellinterrierit ry

Opinnäytetyön tavoite oli toteuttaa Parsonrussellinterrierit ry:n terveystietokannan verkkokäyttöliittymä. Opinnäytetyön asiakkaana on Parsonrussellinterrierit ry.

Opinnäytetyön teoriaosuudessa käsitellään käyttöliittymän suunnittelua, web-tekniologiaa ja tietoturvaa.

Itse tietokannan toteutti opinnäytetyönään Tero Ranta.

Käyttöliittymän toteutus eteni ketterän kehitysmenetelmiä soveltaen. Asiakaan edustaja arvioi käyttöliittymää viikoittain.

Lopputuloksena saatiin toimiva, asiakkaan vaatimusten mukainen tietokantapalvelu, joka otettiin käyttöön keväällä 2015.

Avainsanat: verkkosivut, käyttöliittymä, verkkopalvelu, web-palvelu.

Abstract

Tuomas Porvali

Parsonrussellinterrieri ry's health database user interface, 70 pages

Saimaa University of Applied Sciences

Technology Lappeenranta

Information Technology

Information systems

Bachelor's Thesis 2015

Instructors: Senior Lecturer Martti Ylä-Jussila Saimaa University of Applied of Sciences, Heli Korppinen, Parsonrussellinterrierit Ry

The goal of the thesis was to produce a web user interface. The project's customer is Parsonrussellinterrierit ry. The Thesis deals with web design, web technology and security.

The database and its tables were created by Tero Ranta as his thesis.

The development of the user interface proceeded by applying agile software development techniques. The customer received every week new iteration of the user interface for evaluation.

A fully functional database service was achieved as a result which was deployed in spring 2015.

Keywords: web technology, web design, web service.

Sisällys

Termit ja käsitteet	6
1 Johdanto	9
2 Ohjelmistotuotannon käytännöt	10
2.1 Projektinhallinta	10
2.2 Vesiputousmalli	11
2.3 Scrum	13
2.4 UML	14
2.5 Tuotteenhallinta ja versiohallinta	20
2.6 Laadunvarmistus	21
2.7 Dokumentointi	21
2.8 Vaatimustenhallinta	22
3 Käyttöliittymän ja käytettävyyden suunnittelu	23
3.1 Käyttöliittymän suunnittelun vaiheita	23
3.2 Visuaalinen suunniteltu	24
3.3 Käytettävyydestaus	26
4 Web-palvelun rakenne ja toiminta	27
4.1 Tietokantapalvelin	28
4.2 Web-palvelin	28
4.3 Verkkoselain	28
4.4 Webhotelli	30
5 Käytetyt ohjelmointikielät	31
5.1 HTML	31
5.2 PHP	32
5.3 JavaScript ja JavaScript-kirjastot	33
5.4 SQL	34
5.5 MySQL ja PhpMyAdmin	35
5.6 CSS	36
6 Käytetyt ohjelmat	37
6.1 XAMPP	37
6.2 Notepad++	38
6.3 FileZilla	39
7 Tietoturva	40
7.1 Roolit	41
7.2 Salaus	41
7.3 SQL-injektio	42
7.4 Brute force -hyökkäys	43
7.5 Istunto	43
7.6 Varmuuskopio	43
8 Opinnäytetyöprojektin toteutus	44
8.1 Projektin suunnittelu ja hallinta	44
8.2 Projektisuunnitelma	45
8.3 Tietokannan suunnittelu	45
8.4 Vaatimusmäärittely	45
8.5 Webhotellin tilaaminen	46
8.6 Verkkosivujen rakentaminen	48
8.7 Testaus	49
8.8 Käyttöönotto	50

9	Lopputuotteen ja ratkaisujen esittely	51
9.1	Verkkosivujen rakenne.....	51
9.2	Käyttötapauskaavio.....	53
9.3	Tietokantakaavio.....	54
9.4	Tietoturvaratkaisut.....	55
9.5	Optimointi ja tehokkuus ratkaisut.....	55
9.6	Samanaikaisuus.....	56
9.7	Käyttöliittymäratkaisuja	57
9.8	Etusivu ja sisään kirjautuminen	58
9.9	Itserekisteröinti	59
9.10	Haku	60
9.11	Käyttäjätili.....	61
9.12	Koiranlisäys.....	62
9.13	Koiran tietojen muokkaus	63
9.14	Koiranselaus	64
9.15	Käyttäjähallinta.....	65
9.16	Käyttäjän lisäys	66
10	Yhteenveto.....	67
	Kuvaluettelo	68
	Lähteet	69

Termit ja käsitteet

.css	Tiedostopäätte CSS-tiedostoille.
.js	Tiedostopäätte JavaScript-tiedostoille.
.php	Tiedostopäätte PHP-tiedostoille.
.sql	Tiedostopäätte SQL-tiedostoille.
Apache	Apache verkkopalvelin. Palvelinohjelmisto, joka mahdollistaa www-sivujen selailun.
Chrome	Googlen kehittämä verkkoselain.
CSS	<i>Cascading Style Sheet</i> , porrastetut tyyliarkit. CSS on tyyliohje www-sivuille, jolla määritellään esimerkiksi tekstin fontti, koko ja väri.
Internet Explorer	Microsoftin kehittämä verkkoselain, joka on valmiina kaikissa heidän käyttöjärjestelmissä.
Firefox	Mozillan kehittämä verkkoselain.
Filezilla	Sovellus, jolla voi siirtää tiedostoja FTP-palvelimelle.
FTP	<i>File Transfer Protocol</i> , tiedoston siirtoprotokolla.
JavaScript	Skriptikieli, jolla on mahdollista toteuttaa interaktiivisuutta www-sivuille.
HTML	<i>Hyper Text Markup Language</i> , merkkäuskieli, jolla tehdään verkkosivuja .
HTML-elementti	HTML-dokumentissa oleva elementti, jolla kuvataan sisältöä. Muodostuu sisäkkäisistä ja peräkkäisistä elementeistä.
MySQL	Tietokantaohjelmisto.
Notepad++	Tekstieditori koodinkirjoittamista varten
Opera	Opera Softwaren kehittämä verkkoselain.

OVH	Webhotelleja ja palvelimia myyvä ranskalainen palvelu. Nimi on lyhenne sanoista 'On Vous Héberge' ja kääntyy suomeksi 'me isännöimme teidän verkkosivut'
PHP	<i>PHP: Hypertext Preprocessor</i> , palvelinkieli. Koodi, joka suoritetaan palvelimella ennen, kun verkkosivut ladataan käyttäjälle näkyviin.
phpMyAdmin	Verkkoselaimen kautta käytettävä tietokantahallintasovellus.
Scrum	Ketterä ohjelmiston kehitysmenetelmä.
SQL	<i>Structured Query Language</i> , relaatiotietokannan kysely- ja hallintakieli.
Safari	Applen kehittämä verkkoselain.
Sisällönhallintajärjestelmä	Järjestelmä, jolla voi hallita verkkosivun sisältöä ja sen rakennetta. Esim. Drupal, Joomla ja Wordpress.
TCP	<i>Transmission Control Protocol</i> , Tietoliikenneprotokolla.
Tietokanta	Järjestetty tietovarasto.
UML	Unified Modeling Language, mallinnuskieli, joka sisältää 13 kaavioita järjestelmän kuvaamiseen.
Vesiputousmalli	Teoreettinen ohjelmistokehityksen vaihejakomalli.
Verkkotunnus	Verkkosivun nimi esim. www.hs.fi.
Webhotelli	Tila, jonka www-palvelimen tarjoaja vuokraa verkkosivuja varten.
Web-palvelu	Verkossa tarjottava palvelu.
WWW-tiedosto	Tiedosto, joka sijaitsee verkkopalvelimella. HTML, PHP, CSS ja JavaScript.

XAMPP

XAMPP on avoin monialusta ohjelmistokokoelma verkkosivustojen ja -ohjelmien kehittämistä varten. XAMPP on lyhenne sanoista Apache, Mysql, PHP ja Perl. X tarkoittaa "cross-platfrom" eli alustariippumaton.

1 Johdanto

Opinnäytetyön tavoite on rakentaa käyttöliittymä Parsonrussellinterrierit ry:n terveystietokannalle. Käyttöliittymä rakennetaan Tero Rannan tekemän toiminnalliset määrittelyn mukaan ja hänen suunnitteleman relaatiotietokannan vaatimusten mukaisesti.

Asiakas on Parsonrussellinterrierit ry ja asiakkaan edustajana toimii koulutuspäällikkö Heli Korpinen. Asiakas tarvitsee tietokannan, johon kaikki Parsonrussellinterrierit ry:n jäsenten koirien tiedot kerätään. Aikaisemmin koirien tiedot ovat olleet ympäri verkkoa muilla sivuilla. Tällä ratkaisulla pyritään tuomaan tiedot yhteiseen ympäristöön.

Toteutettavan järjestelmän ideana on tehdä verkkoselaimessa toimiva tietokantakokonaisuus, jossa käyttäjät voivat selata ja lisätä omia koiriaan tietokantaan.

2 Ohjelmistotuotannon käytännöt

Ohjelmistotuotanto on liiketoimintaa, joka vaatii onnistuakseen hyvin suunniteltuja ja valvottuja prosesseja, tekniikoita, menettelytapoja, työkaluja ja ammattitaitoisen henkilökunnan. Ohjelmistot toteutetaan yleensä projekteina. Projektinhallinta vaatii projektin suunnitelman, ohjausta sekä käyttökelpoisen kehitysmallin. Ohjelmistotuotannon kivijalkatoiminnot ovat tuotteenhallinta, laadunvarmistus, dokumentointi ja vaatimustenhallinta. (Haikala 2011.)

2.1 Projektinhallinta

Projektinhallintaan kuuluvat suunnittelu, organisointi, läpivienti, työmäärän arviointi ja riskinhallinta.

Projektin organisaatioon kuuluvat asiakas, toimittaja ja muut sidosryhmät. Toimittaja asettaa projektiryhmän ja projektipäällikön. Asiakas asettaa asiakkaan edustajan, loppukäyttäjät ja mahdollisen testausryhmän. Tuotteen omistaja ylläpitää yhteyksiä kaikkiin sidosryhmiin.

Projekti jaetaan pieniin osiin, joista arvioidaan työmäärää, ketkä vastaavat mistäkin osista ja tutkitaan, mitä riippuvuuksia eri osilla on, esimerkiksi, voiko tehtävän C aloittaa ennen A tehtävän valmistumista. Suurien osien aika-arvioissa tulee helposti virheitä, joten osittamista jatketaan niin kauan, että osat saadaan hallitavaksi. (Haikala 2011, s. 153 - 167.)

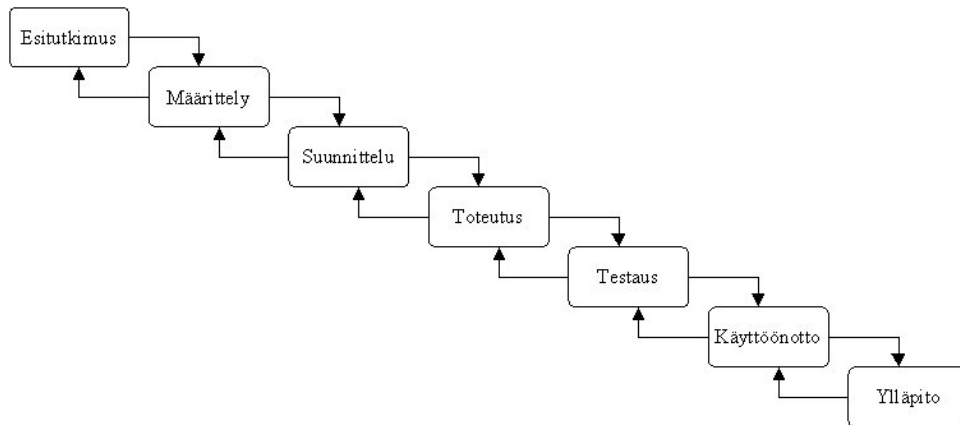
2.2 Vesiputousmalli

Vesiputousmalli on teoreettinen tuotantoprosessi, jossa edetään vaihe vaiheelta eteenpäin: kun määrittely on valmis, edetään suunnitteluun ja määrittelyyn ei enää kosketa. Vesiputousmallissa painotetaan hyvää suunnittelua ja dokumentaatiota. Jokainen vaihe pyritään suunnittelemaan koko järjestelmän laajuisena ja mahdollisimman täydellisesti. (Hiltunen 2004.)

Vesiputousmalli on jaettu seitsemään vaiheeseen (Hiltunen 2004.):

- **Esitutkimuksen** tehtävänä on tuottaa tietoa tietojärjestelmän kehittämistä päättävälle sekä määrittää lähtökohdat mahdolliselle kehityshankkeelle.
- **Määrittelyvaiheessa** kootaan dokumentti, joka sisältää asetetut vaatimukset eri sidosryhmiltä ja mitä rakennettavan järjestelmän tulee tehdä ja sekä analysoidaan muut vaiheet ja vaatimusten toteuttamiskelpoisuus.
- **Suunnittelu** jaetaan arkkitehtuuri- ja moduulisuunnitteluun. Arkkitehtuurisuunnittelussa järjestelmä jaetaan osiin, päätetään osien vastuut, suunnitellaan osien välinen kommunikointi ja rajapinnat. Moduulisuunnittelussa suunnitellaan järjestelmän moduulien sisäinen rakenne.
- **Toteutusvaiheessa** Ohjelmisto toteutetaan joillain ohjelmointikielellä ja moduulit integroidaan toimivaksi kokonaisuudeksi.
- **Testauksessa** etsitään ohjelmistosta virheitä ja suoritetaan toiminnalliset testitapaukset ja suorituskykytestit.
- **Käyttöönnotossa** ohjelmisto on testattu ja tuotteistettu. Ohjelmiston käyttöönotto vaatii käyttäjän hyväksymisen.
- **Ylläpidossa** ohjelmistoon lisätään uusia ominaisuuksia tai sen toimintaa muutetaan sen käyttöympäristön muutosten takia. Tämä vaihe ohjelmistotuotannossa pisin ja työläin.

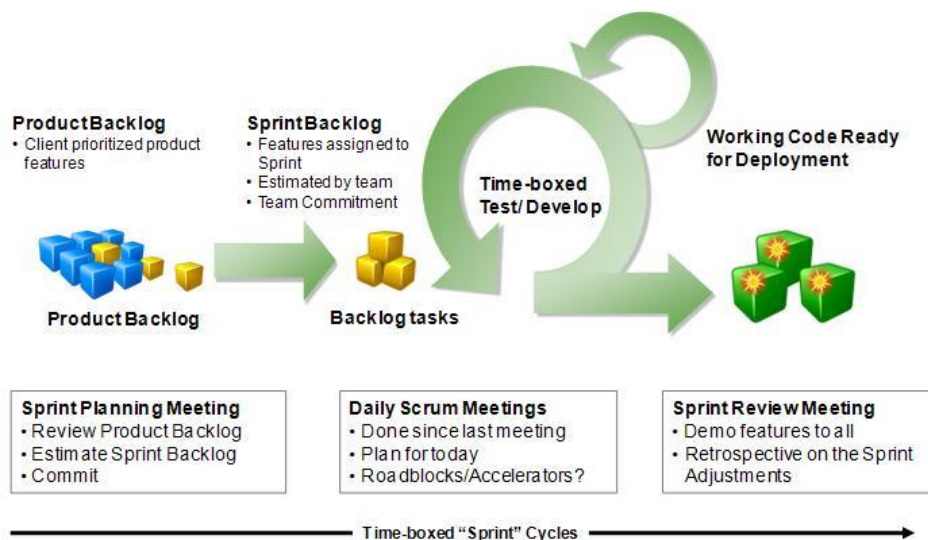
Vesiputousmallin (Kuva 1) etuina on kattava dokumentaatio ja hyvin suunniteltu johtaa merkittäviin säästöihin myöhemmissä vaiheissa. Haittana vesiputousmallissa palaaminen edelliseen vaiheeseen on hyvin vaikeata ja testaus aloitetaan vasta kun koko järjestelmä on toteutettu. Virheen ilmetessä joudutaan palaamaan siihen vaiheeseen, jossa virhe on tehty eli joskus jopa määrittelyvaiheeseen. Toinen haitta on se, että asiakas ei pääse kokeilemaan järjestelmää ennen toteutusta.



Kuva 1. Vesiputousmalli. (Hiltunen 2004.)

2.3 Scrum

Scrum on iteratiivinen kehitysmenetelmä. Scrum-prosessi (Kuva 2) toimii pyrhdyksillä (sprint), jotka kestävät yhdestä kolmeen viikkoon. Prahdyksien alussa pidetaan palaveri (sprint planning meeting), jossa keskustellaan seuraavan pyrahdyksen sisältö (sprint backlog) ja suoritetaan edellisen pyrahdyksen jälkitar-kastus (sprint review meeting). Prahdyksen aikana syntynyt iteraatio toimitetaan asiakkaalle. (Ilkka Haikala, 2011 s.46.)



Kuva 2. Scrum-malli (Lucero 2013.)

Scrum-ryhmä muodostuu tuotteen omistajasta, Scrum-mestarista ja tiimistä.

Tuotteen omistaja pitää työlistaa (product backlog), joka sisältää järjestelmän vaatimukset: järjestelmän ominaisuudet, käyttötapauksia, käyttäjätarinoita, vaatimuksia ja virheraportteja.

Scrum-mestari pitää huolta, että pyrahdyksen prosesseja käytetään eikä tiimillä ole ongelmia tai esteitä työn suorittamisen suhteen.

Scrum työpäivät ovat 8 tuntia pitkiä työuupumusten välttämiseksi, koska väsynyt ohjelmoija tekee virheitä ja ei ole siksi tuottoisa. Jokaisen päivän alussa pidetään päiväpalaveri (daily scrum meeting), jossa ryhmän jäsenet kertovat mitä he ovat

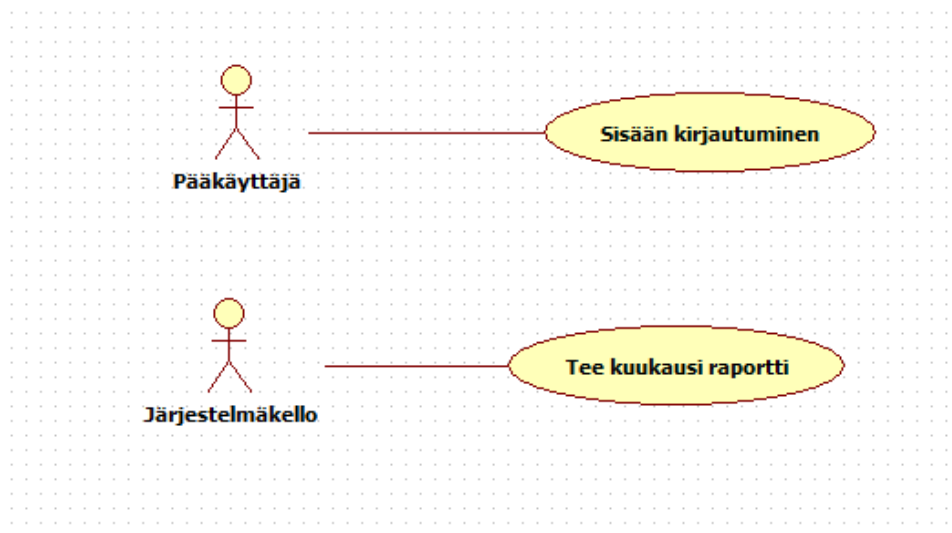
saaneet aikaiseksi, mitä aikovat tehdä ja mahdolliset ongelmat ja esteet. Päiväpalaveri kestää noin 15 minuuttia. (Haikala, 2011.)

2.4 UML

Ohjelmiston vaatimusten ja suunnitelmien dokumentointi perustuu sanallisen selostukseen ja kaavioihin. Kaaviot täydentävät sanallista selostusta. Kaaviot laaditaan standardin kuvauskielen mukaisesti.(Haikala, 2011 s.79.)

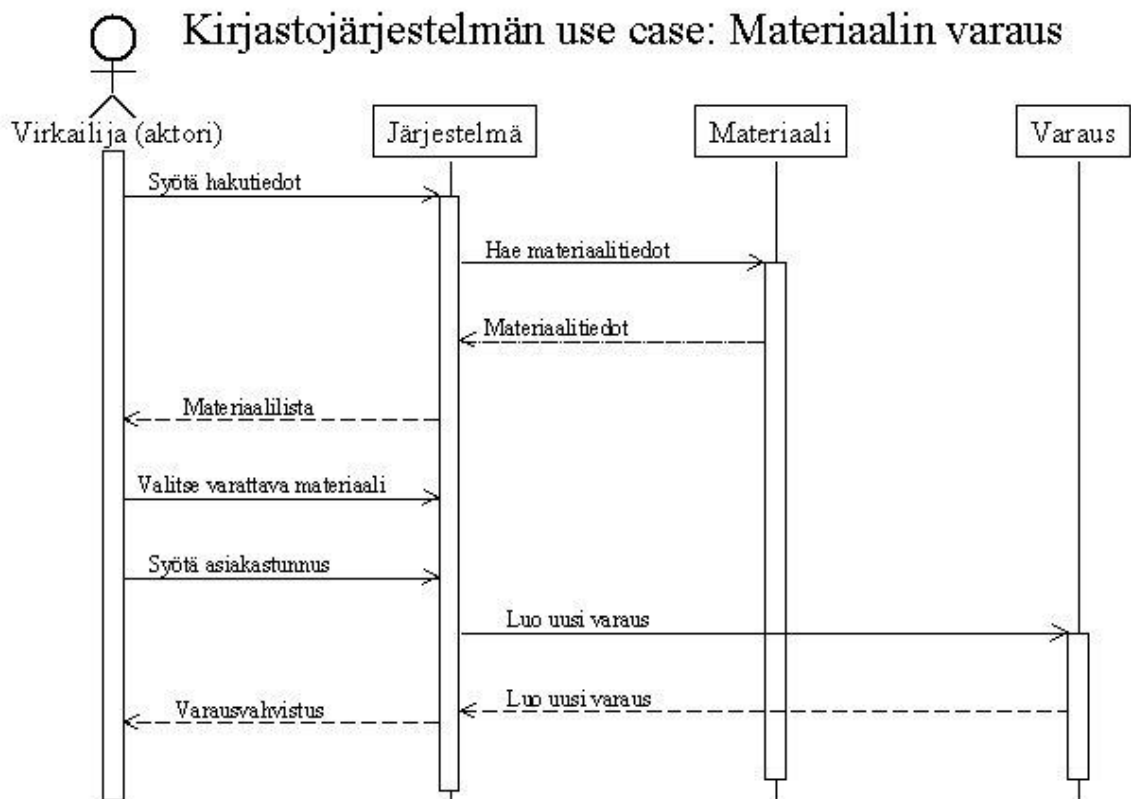
UML (Unified Modeling Language) on graafinen mallinnuskieli, joka sisältää 13 erilaista kaaviota. Kaavioilla kuvataan rakennetta, käyttäytymistä ja vuorovaikutusta.(Haikala, 2011 s.79.)

Käyttötapauskaavio (Kuva 3) on yksinkertainen kaavio, jolla kuvataan järjestelmän käyttäjät (aktorit) ja mitä toimintoja he käyttävät järjestelmässä. Järjestelmän käyttäjiä voivat olla henkilö tai laite (Haikala, 2011 s.79.)



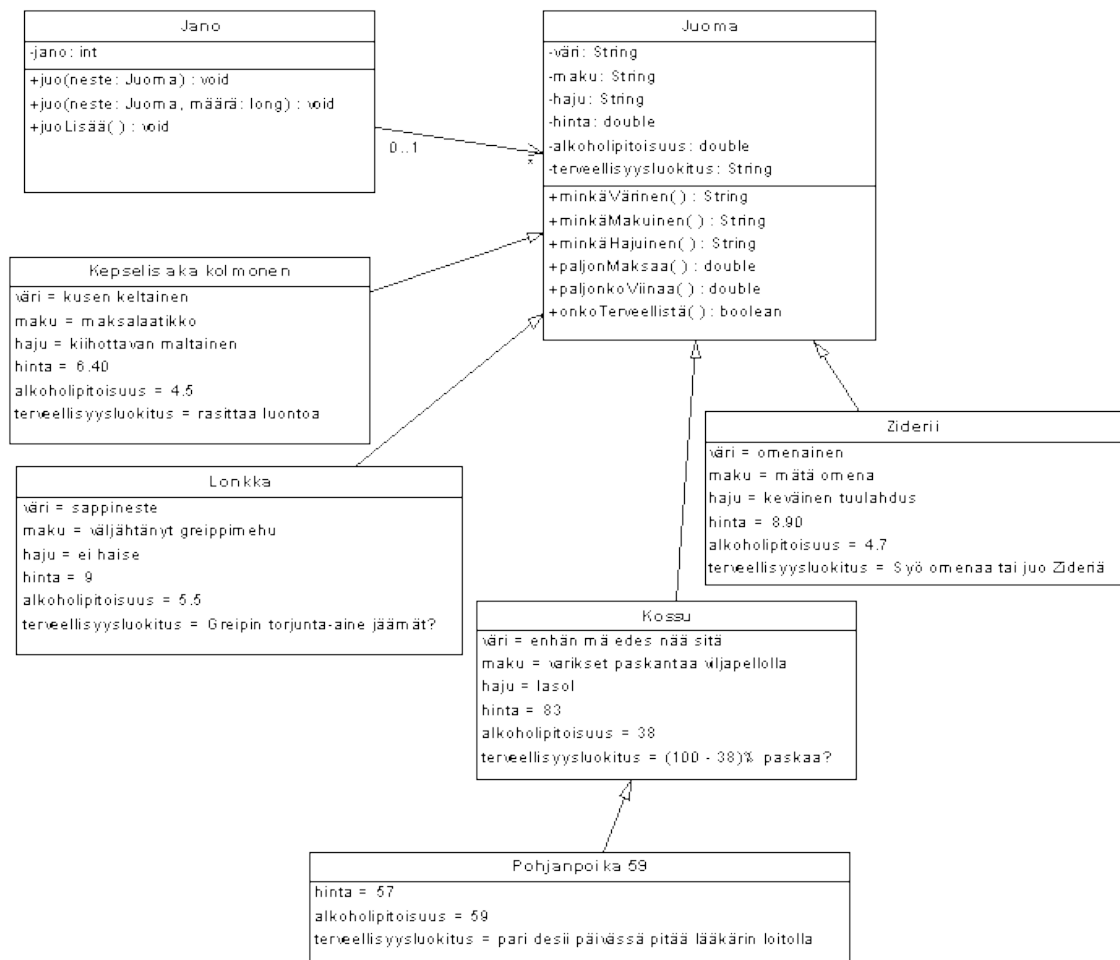
Kuva 3. Käyttötapauskaavio.

Sekvenssikaaviolla (Kuva 4) kuvataan käyttötapauksen kulku järjestelmässä. (Haikala, 2011.)



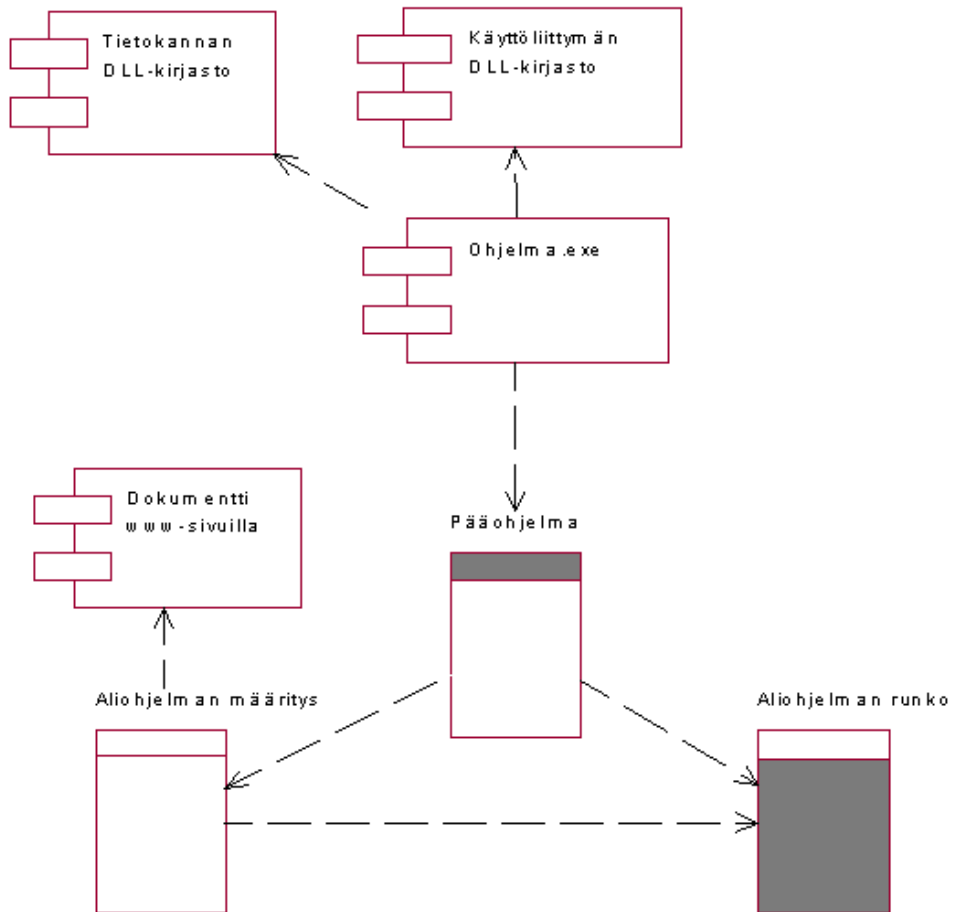
Kuva 4. Sekvenssikaavio. (Oliomallinnusten perusteet, 2015.)

Luokkakaaviossa (Kuva 5) kuvataan ohjelman eri luokkien suhteet toisiinsa ja luokkien sisältö. (Haikala, 2011.)



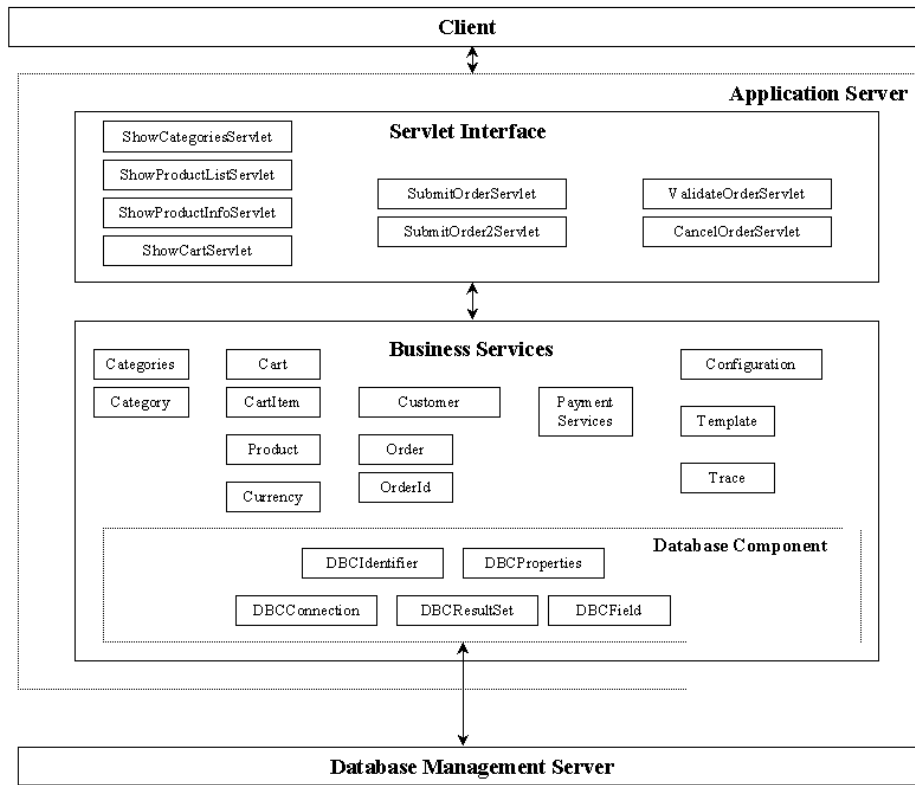
Kuva 5. Luokkakaavio. (Niksula, 2015.)

Komponenttikaaviossa (Kuva 6) kuvataan miten eri komponentit ovat sidottu toisiin komponentteihin tai muihin järjestelmiin. (Haikala, 2011.)



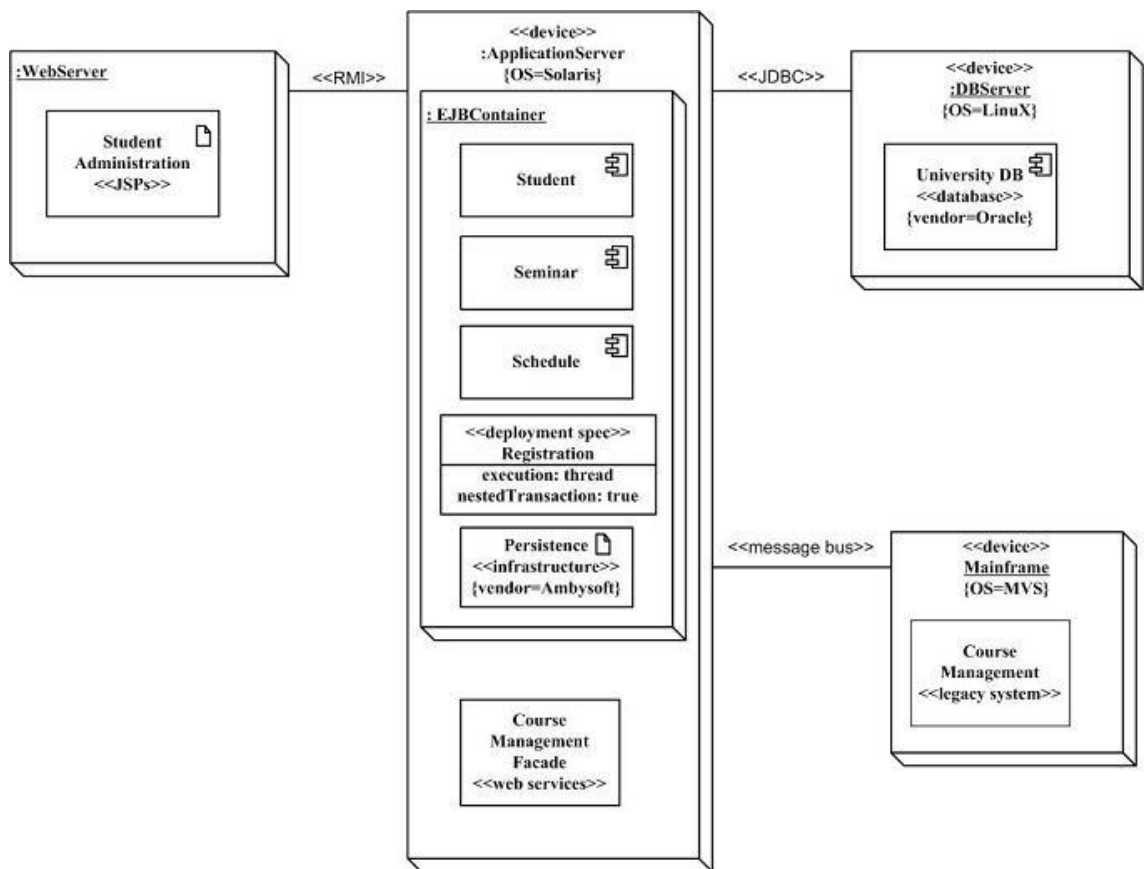
Kuva 6. Komponenttikaavio. (Hämeen ammattikorkeakoulu, 2015.)

Arkkitehtuurikaaviolla (Kuva 7) esitetään koko järjestelmää ja sen osia. (Ilkka Haikala, 2011.)



Kuva 7. Arkkitehtuurikaavio. (SoberIT, 2015.)

Sijoittelukaaviolla (Kuva 8) kuvataan ohjelmistokomponenttien sijoittelua fyysisiin laitteisiin sekä prosessoreiden väliset verkkoyhteydet ja protokollat. (Ilkka Haikala, 2011.)



Kuva 8. Sijoittelukaavio (Wampler, 2015.)

2.5 Tuotteenhallinta ja versiohallinta

Ohjelmistotuotteet koostuvat tyypillisesti suuresta määrästä erilaisia komponentteja esimerkiksi lähdekieliset moduulit, toiminnallinen määrittely, tekninen määrittely, testaussuunnitelmat ja käyttöohjeet. Komponenteista voidaan koota suurempia kokonaisuuksia, konfiguraatioita. Sekä komponentit ja konfiguraatiot kehittyvät projektin elinkaaren aikana, kun virheitä korjataan, uusia ominaisuuksia lisätään ja komponenteista tehdään uusia variaatioita. Kaikesta tästä huolehtii tuotteenhallinta, jonka osuus ohjelmistotyössä vaihtelee erittäin paljon tapauskohtaisesti tuotteen ja projektin luonteesta. Tuotteenhallintaan sisältyy komponentin ja konfiguraation versiohallinnasta, sen identifioinnista, tuottamisesta ja muutosten hallinnasta ja toimintatavoista. (Haikala, 2011. s.169 - 173.)

Versiohallinnassa seurataan komponentin tai konfiguraation elinkaarta tai niin sanottua versiopuuta. Kun komponenttiin tai konfiguraatioon tehdään muutoksia, kirjataan vastuuhenkilö, muutokset ja työntila. Versiopuu kasvaa ylöspäin eli tehtiin muutoksia versioon 1.0 ja saatiin aikaiseksi uusi versio 1.1. Joskus komponentista tai konfiguraatiosta tarvitaan variaatio esimerkiksi vaikka toiseen käyttöympäristöön, jolloin versiopuuhun kasvaa sivuhaara versio 1.1.1. Haaroittamista tulisi välttää koska se hankaloittaa välillä suurestikin ylläpitäjän työtä. (Haikala, 2011. s.169 - 173.)

2.6 Laadunvarmistus

On olemassa toimenpiteitä ja menetelmiä, joilla varmistetaan ohjelmiston laatu. Yleisimmät laadunvarmistusmenetelmät ovat katselmointi ja testaus. Laatua mitataan erilaisilla mittareilla. Mittareilla mitataan ohjelman luotettavuutta, tehokkuutta, opittavuutta ja taloudellista merkitystä.

Katselmointia tehdään asiakkaan kanssa. Katselmoinnin kohteena voi olla valmisdokumentti, lähdekoodi tai muutosesitys. (Haikala, 2011. s.200.)

Testauksessa varmistetaan, että ohjelmisto toimii määrittelyn vaatimusten mukaisesti. Testejä on useita erilaisia: hyväksymistestaus, kenttätestaus, kuormitustestaus ja luotettavuustesti. (Haikala, 2011. s.206 - 209.)

2.7 Dokumentointi

Ohjelmistokehityshankkeen aikana syntyy runsaasti dokumentteja. (Haikala, 2011.)

Projektista tehdään projektisuunnitelma, jossa määritellään projektin koko, kustannukset, resurssit ja aikataulut.

Vaatusmäärittelyssä kuvataan ohjelmistoprojektin tavoitteita ja mitä vaatimuksia sen on täytettävä ja miten ne täytetään. Vaatimukset jaotetaan toiminnallisiin ja ei-toiminnallisiin vaatimuksiin. Ei toiminnalliset vaatimukset ovat laadulliset vaatimukset ja resurssivaatimukset.

Toiminnallisessa määrittelyssä on tarkkaan määritelty mitä loogisia osia ohjelmassa on ja mitä ne tekevät. Toiminnoista tehdään käyttötapauksia.

Järjestelmän arkkitehtuurisuunnitelmaan kuvataan järjestelmän arkkitehtuuri eli komponentit ja niiden tehtävät sekä komponenttien väliset rajapinnat.

Moduulisuunnitelmaan kuvataan ohjelmistokomponenttien ja moduulien sisäinen rakenne.

Testaussuunnitelma sisältää testausajankohdan, testaushenkilöstön, testausvälineet ja –paikan ja testausprosessin kuvauksen.

Dokumentit muuttuvat projektin aikana ja niiden hallinnasta on huolehdittava.

2.8 Vaatimustenhallinta

Vaatimusten selvittämisestä syntyy toiminnallinen määrittely, joka sisältää alustavat vaatimukset. Toteutusaikana asiakas esittää muutospyyntöjä, jotka analysoidaan ja lopulta hyväksytään tai hylätään.

Ketterissä menetelmissä vaatimusmuutosten käsittelyä ei oikeastaan ole, koska muutokset käsitellään uusina vaatimuksina tuotteen työlistassa tai viimeistään pyrähdysten suunnittelukokouksessa.

Pienuhköissä projekteissa muutostenhallinta hoidetaan usein hyvin kevyellä prosessilla. Vasta kun projekti on päätynyt, siirrytään ylläpitovaiheessa muodollisempaan muutosten hallintaan. (Haikala 2011, s. 67.)

3 Käyttöliittymän ja käytettävyyden suunnittelu

Käyttöliittymän suunnitellun tavoitteita ovat opittavuus, tehokkuus, muistettavuus, virheiden määrä ja miellyttävyys. Käyttöliittymällä on erittäin suuri merkitys ohjelman käytettävyyteen. Huono käyttöliittymä voi pilata ohjelman ja käyttökelpoisuuden ja estää sen myymisen.

3.1 Käyttöliittymän suunnitellun vaiheita

Ennen käyttöliittymän toteutusta tehdään käsitekartta, jolla selvitetään loppukäyttäjän tarvitsema termistö. Tutut termit auttavat loppukäyttäjää käyttöliittymän ymmärtämisessä. (Sinkkonen 2009, s.144 - 147.)

Käyttäjätarinoiden ja -persoonien kirjoittaminen auttaa simuloimaan, miten loppukäyttäjän tulisi käyttää käyttöliittymää. Käyttäjätarinoita käytetään myös olemassa olevan käyttöliittymän ominaisuuksien ja toimintojen parantamiseen. (Sinkkonen 2009, s. 171.)

Käyttöliittymän toteutusta varten tehdään käyttötapaukset. Käyttötapaukset kuvataan määritellyssä. Käyttötapauksessa määritellään toiminnon nimi, sen suorittaja, tarkoitus, alku- ja loppuehdot, kuvaus, vaihtoehdot ja poikkeukset virhetilanteisiin. (Sinkkonen 2009, s.181 - 182.)

Ennen yksittäisten sivujen tekoa on suunniteltava sivustonarkkitehtuuri, eli miten sivulle navigoidaan ja mitä tietoja sillä sivulla esitetään erikseen ja mitä prosesseja se sisältää. Kun arkkitehtuuri on suunniteltu, aletaan tekemään rautalankamalleja.

Rautalankamalleissa sivuihin on asetettu kentät, otsikot, linkit ja kuvat paikoilleen, mutta ne ovat vain paikan varaajia, jotka kertovat visuaalisesta suunnittelusta. Iteoimalla malleja päästään lähelle ihanteellista ratkaisua ja ne toimivat hyvinä dokumentteina, joita voi testata vaikka mitään koodia ei ole edes olemassa. (Sinkkonen 2009, s. 203 - 214.)

3.2 Visuaalinen suunniteltu

Visuaalisessa suunnittelussa minimalistinen lähestymistapa on paras ratkaisu käyttöliittymän suunnittelua aloittavalle henkilölle. Käyttöliittymän elementtien suunnittelu aloitetaan kun rautalankamalli on valmis, mutta sitä tehdään myös projektin rinnallakin. (Sinkkonen 2009, s. 247.)

Yksi visuaalisuuden suunnittelun tavoitteista on esteettisyys eli luodaan positiivisia tunnetiloja, jotka auttavat järjestelmän käytössä. Positiivisilla tunnetiloilla on todettu olevan myönteisiä vaikutuksia esimerkiksi tehokkuuteen, tavoitteiden saavuttamiseen, oppimiseen ja ongelmanratkaisuun. Lisäksi positiivinen tila vaikuttaa käyttäjään siten, että hän sietää pieniä käytettävyyden ongelmia. (Sinkkonen 2009, s. 249.)

Visuaalisessa käytettävyydessä käyttöliittymän ulkonäön pitää auttaa huomaamaan, jäsentämään ja ymmärtämään asiat ja välittää käyttäjälle sivuston antama viesti ja antaa vihjeitä, mitkä asiat käyttöliittymässä ovat tärkeitä.

Visuaalisen suunnittelussa sivun osat tulee jaotella omiin joukkoihin. Nämä joukot on ilmentävä asemoinnissa elementin sisennyksinä tai tekstin muotoilussa. Elementit ohjaavat käyttäjien katsetta. Katsetta ohjataan tyhjillä tiloilla, viivoilla tai väreillä. Taitava visuaalinen suunnittelija saa käyttäjän huomaamaan tismalleen ne oikeat asiat oikeassa järjestyksessä.

Värit ovat voimakkain osa visuaalista suunnittelua ja leimaavat koko käyttöliittymän yleisilmettä. Värit kuvaavat niin käyttöliittymän antamaa viestiä ja tunnelmaa. Erilaisia värejä ei tulisi käyttää liikaa vaan suppea ja hallittu väripaletti antaa varmemmin toimivan ja tyylikkäämmän ulkoasun. (Sinkkonen 2009, s. 252.)

Kuvat auttavat havainnollistamisesta, mutta suuret kuvat tai suuret määrät kuvia hidastavat käyttöliittymän toimintaa. Kuvia kannattaa käyttää harkitusti ja tekijän oikeuksia rikkomatta. (Sinkkonen 2009, s. 249 - 255.)

Käyttöliittymän suunniteletussa hyödynnetään 10 käyttöliittymän suunnittelun sääntöä (Treehouse, 2012.):

- Tunne käyttäjäsi: Teet käyttöliittymää asiakkaalle, et itsellesi
- Tunne mallit: Tee käyttöliittymä samankaltaiseksi olemassa olevien käyttöliittymien kanssa. Tällöin käyttäjien ei tarvitse oppia uuden käyttöliittymän oikkuja.
- Yhdenmukaisuus: Pitämällä käyttöliittymän osat samankaltaisina nopeutat käyttäjän oppimista ja samalla tehostat käyttöliittymän toimintaa.
- Visuaalisuus: Hyödyntämällä värejä ja oikeanlaisia fontteja tekee käyttäjän huomion ohjaamisesta helpompaa.
- Palaute: Käyttöliittymän tulee antaa palautetta, kun käyttäjä suorittaa toimintoja oikein tai väärin.
- Anteesi antava: Käyttöliittymä ei saisi rangaista käyttäjää virheellisen käytön takia.
- Anna valtaa käyttäjälle: Anna käyttäjälle esimerkiksi näppäin oikotiet pura käyttöliittymän monimutkaisuutta.
- Puhu heidän kieltä: Tekemällä selkeät käsitteet toiminnoille pidät viestit yksinkertaisina.
- Pidä yksinkertaisena: Paras käyttöliittymä on näkymätön. Se ei sisällä huomion varastavia ja turhia elementtejä. Suunnittele käyttöliittymään vain ne elementit, mitä käyttäjä oikeasti tarvitsee.
- Iteroi: Käyttöliittymään tulee aina virheitä. Oppi niistä.

3.3 Käytettävyytestaus

Käytettävyyttä mitataan viidellä kriteerillä (Sinkkonen 2009, s. 297.):

- **Opittavuus:** kuinka nopeasti uusi käyttäjä suorittaa tehtävän
- **Tehokkuus:** Kuinka monta tehtävää käyttäjä suorittaa tietyssä ajassa.
- **Muistettavuus:** Osaako käyttäjä suorittaa saman tehtävän kuukauden kulluttua.
- **Virheettömyys:** Kuinka monta virhettä käyttäjä tekee tehtävän suorituksen aikana.
- **Tyytyväisyys:** Kuinka monta käyttäjää pitää testattavaa järjestelmää parempana kuin kilpailevaa järjestelmää.

Käytettävyytestauksella pyritään selvittämään miten aloitteleva käyttäjä tai harvoin käytävä selviää käyttöliittymän kanssa. Testit sopivat siis hyvin mahdollisiin verkkopalvelun ongelmakohtien löytämiseen ja korjaamiseen. Kehittäminen tulee paljon edullisemmaksi, mitä aiemmin virheet saadaan korjattua pois. Kalleimmat virheet ovat ne, jotka jäävät julkaisuversioon.

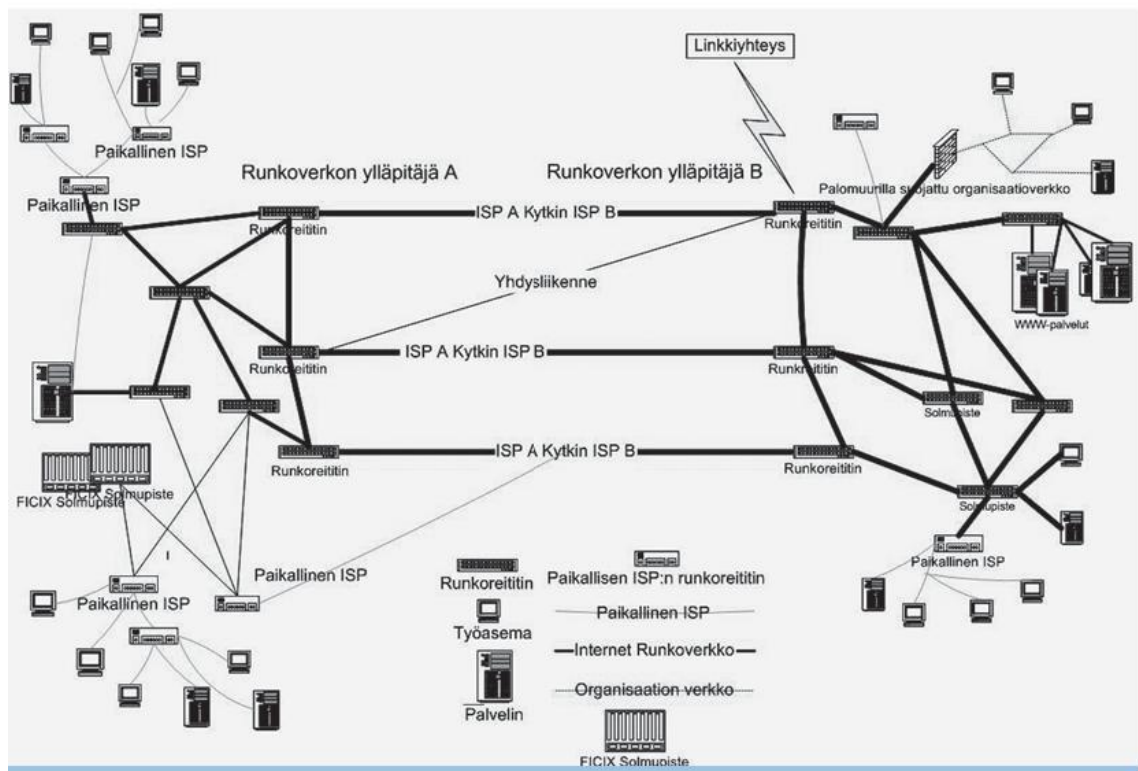
Käytettävyytestauksella käyttöliittymää kehitetään eteenpäin ja varmistetaan tuotteen käytettävyys. Kun huomaataan, mikä on käyttöliittymässä vikana, tehdään toteutus osittain uudestaan. (Sinkkonen 2009, s.299.)

Käytettävyyttä tulisi testata pitkin kehityksen elinkaarta. On havaittu käytännössä, että sarja pieniä ja lyhyitä testejä auttaa löytämään enemmän virheitä kuin yksi iso testi projektin loppupuolella. Isossa testissä virheet peittävät toisensa ja testikäyttäjät kiinnittävät huomionsa väärin asioihin, jotka eivät ole kiinnostavia kehityksen sen hetkisen kehityksen kannalta. (Sinkkonen 2009, s.300.)

Ennen käytettävyytestausta suoritusta määritellään testitehtävät ja testiasetelmat kattavat kiinnostavimmat asiat. (Sinkkonen 2009, s.303.)

4 Web-palvelun rakenne ja toiminta

Internet runko (Kuva 9) on maailmanlaajuinen reitittimien ja niiden välinen tietoliikenneyhteyksien verkko. Reitittimet vastaanottavat tietoliikennepaketteja ja välittävät niitä omien reitittaulutietojensa perusteella edelleen seuraavalle reitittimelle, toivottavasti kohti oikeaa vastaanottajaverkkoa. Lisäksi verkossa on palvelinkoneita kuten nimipalvelimia, sähköposti- ja www-palvelimia. (Valtionvarainministeriö ministeriö, 2009.)



Kuva 9. Verkon rakenne. (Valtionvarainministeriö, 2009.)

4.1 Tietokantapalvelin

Tietokantapalvelin sisältää tietokantasovelluksia ja tarjoaa tietokantaratkaisua ohjelmille tai verkkopalveluille. Tietokanta toimii kyselykielillä, joilla voidaan muokata tietokantaa tai hakea tietoa. Esimerkiksi Oracle ja Microsoft SQL Server ovat tietokantasovelluksia.

4.2 Web-palvelin

Web-palvelin, verkkopalvelin eli www-palvelin voidaan ymmärtää palvelintietokoneeksi tai ohjelmaksi. Yhdessä palvelinkoneessa voi olla muita palvelinohjelmia, verkkosivujen tiedostot ja muut sen toimintaan liittyvät tiedostot. Jotkut palveluntarjoajat vuokraavat kokonaisia verkkopalvelimia tai sitten osan siitä webhotellina.

4.3 Verkkoselain

Verkkoselain tulkkaa HTML-tiedoston ja luo sen pohjalta verkkosivun. Selaimet voivat lukea muitakin tiedostomuotoja kuten esimerkiksi kuvia ja PDF-tiedostoja. Selainten toimintoja ja ominaisuuksia voi kasvattaa laajennusosilla.

Jotkin selaimet sisältävät inspect-työkalun, jolla voi tutkia verkkosivun rakennetta ja koodia.

Yleisimmät verkkoselaimet:

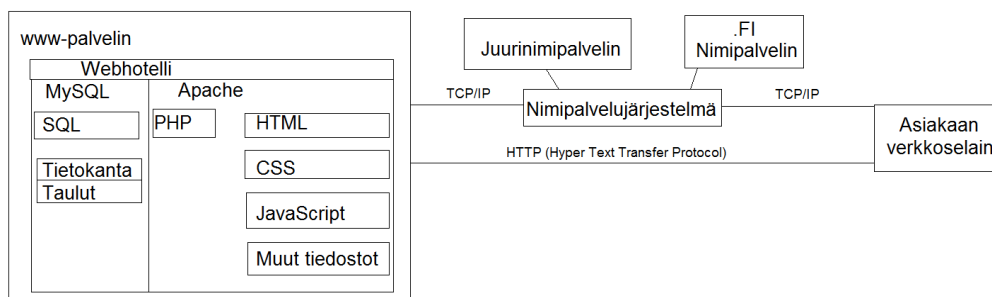
- Internet Explorer on Microsoftin kehittämä verkkoselain, joka on valmiina esiasennettuna kaikissa Windows-käyttöjärjestelmissä.
- Chrome on Googlen kehittämä ilmainen verkkoselain. Chrome tunnetaan sen keveydestä ja nopeudesta.
- Firefox on ilmainen Netscapen avoimeen lähdekoodiin perustuva verkkoselain. Firefoxin toimintoja voi helposti kasvattaa ilmaisilla laajennusosilla. Firefoxia kehittää Mozilla.
- Opera ilmainen verkkoselain, jota kehittää Opera Software. Opera tunnetaan sen laajasta alustatuesta.

- Safari on Applen omistama verkkoselain. Safari on esiasennettu kaikkiin Applen tuotteisiin esim. iMac ja iPhone. Safarista on saatavilla Linux ja Windows versiot

Selain kysyy useilta nimipalvelimilta käyttäjän syöttämää verkkosivun nimeä vastaavaa IP-osoitetta, josta se sitten siirtyy isännöintipalvelulle, josta se lopulta lataa verkkosivun selaimen näkymään (Kuva 10). Verkkotunnuksen muuntaminen IP-osoitteeksi alkaa juurininimipalvelimelta. Juurininimipalvelin sisältää seuraavan tason aluenimipalvelimen osoitteet. Aluenimipalvelin vastaa konkreettisesti omasta alueesta (.fi, .com, .net, .org). Viimeisiltä tasolta kysytään verkkotunnusta vastaavaa IP-osoitetta. Esimerkiksi www.google.com IP-osoitteen haku alkaa juurininimipalvelimelta, joka kysyy com-alueen aluenimipalvelimen vastaavaa IP-osoitetta. Tältä alueelta kysytään google.com nimipalvelimen IP-osoitetta ja vasta viimeisiltä nimipalvelimelta kysytään www.google.com IP-osoitetta.

TCP (Transmission Control protocol, tietoliikenneprotokolla) luo yhteyksiä eri verkkolaitteiden välille, joilla on pääsy internetiin. TCP-yhteyksien avulla tietokoneet voivat lähettää toisilleen tietoa luotettavasti, oikeassa järjestyksessä ja tarvittaessa hävinnyt tieto lähetetään uudestaan. TCP:llä luodaan palvelimen ja selaimen välillä yhteys, jolla palvelin ja selain siirtävät tietoja toisilleen.

Selaimet käyttävät HTTP-protokollaa (Hyper-Text Transfer Protocol, hyper-tekstin siirtoprotokolla) verkkotiedostojen siirtämiseen palvelimilta. Selaimet voivat myös hakea muita tiedostoja palvelimelta verkkotiedostojen lisäksi.



Kuva 10. Web-palvelun arkkitehtuuri

4.4 Webhotelli

Webhotelliilla (web hosting) tarkoitetaan palvelua, jossa palveluntarjoaja vuokraa asiakkaalle web-palvelimelta kiintolevytilaa hänen kotisivuja ja muita verkkopalveluita varten. Webhotellin mukana asiakas saa tarvittavat työkalut hänen verkkosivujen hallintaa. Jotkut palveluntarjoajat antavat mahdollisuuden verkkotunnuksen rekisteröintiin ja käyttää sisällönhallintaohjelmaa.

Webhotellin turvallisuus on palveluntarjoajan käsissä. Koska palveluntarjoajalla on vapaa pääsy webhotelliin, tulee arkojen tietojen säilyttämistä webhotellissa välttää. On säädetty lakeja, jotka sääntelevät palveluntarjoajan toimintaa tietojen säilyttämisessä.

Webhotellit ja verkkopalvelimet ovat alttiita palvelunestohyökkäyksille. Palvelunestohyökkäyksellä nimensä mukaisesti pyritään estämään palvelun käyttö kaatamalla palvelin kuormittamalla sitä hyvin suurella palvelupyynnön määrällä. Hyökkäykseltä ei voi välttyä, mutta siihen voi varautua.

Palvelunestohyökkäystä voi lieventää kasvattamalla palvelimen tehoa, jolloin se kykenee käsittelemään suuren määrän pyyntöjä samalla kertaa tai siirtää kuormaa useammalle palvelimelle. Palvelimen palomuurikin voi virittää estämään useat samat pyynnöt samasta IP-osoitteesta.

Palvelunestohyökkäys ei ole hakkerointitekniikka koska, sillä ei pyritä varastamaan tietoja.

5 Käytetyt ohjelmointikielet

Verkkosivujen tekoa varten tarvitaan seuraavia ohjelmointikieliä: rakenteen kuvaus (HTML), toiminnallisuus (PHP), interaktiivisuus (JavaScript), ulkoasu (CSS) ja tietokannan käsittely (SQL).

5.1 HTML

HTML (Hyper Text Markup Language) on yleisin verkkosivujen kuvauskieli.

HTML:stä on olemassa versiot HTML4, HTML5 ja XHTML1.

HTML toimii käyttämällä elementtejä (<tag>). Elementti koostuu aloitustunnisteesta ja päätöstunnisteesta. Aloitustunniste sisältää attribuutteja, joihin voi sijoittaa arvoja. Alla olevassa koodiesimerkissä on kuvattu kappale.

```
<p><!--Kappaleen aloitus-->
    Tämä on elementin "<p>" -tekstiä
</p><!--Kappaleen päätös-->
```

HTML:llä kuvataan mitkä osat verkkosivulla olevasta tekstistä ovat otsikoita, leipätekstiä ja hyperlinkkejä. HTML:llä tapahtuu myös kuvan ja videon paikan määrittäminen sivulle.

HTML:ää käytetään lomakkeiden rakentamiseen ja sisällön esittämiseen. Tähän käyttötarkoitukseen sopivat elementit <div> ja <frame>

Elementtien ulkonäköä voi muokata määrittelemällä niille tyyliä käyttämällä CSS:ää.

5.2 PHP

PHP (PHP: Hypertext Preprocessor) on verkkopalvelimen skriptikieli. Kun verkkosivua ladataan, suoritetaan ensin verkkosivuun sisältyvä PHP-koodi palvelimessa. PHP:n syntaksi on hyvin yksinkertainen. Toisin kuin muissa ohjelmointikielissä, muuttujille ei tarvitse määrittää tietotyyppiä. Myöskään funktioille ei tarvitse määritellä palautustyyppiä. Alla olevassa koodiesimerkissä on kirjoitettu funktio, joka vastaanottaa kaksi lukua, laskee ne yhteen ja palauttaa lasketun summan. Muuttujat PHP:ssä on merkitty \$-merkillä.

```
Function LaskeSumma($luku1, $luku2){
    $summa = $luku1 + $luku2;
    return $summa;//Palauta summa muuttuja
}
```

PHP-koodi on yleensä omassa erillisessä tiedostossa, mutta myös HTML-koodin sekaan voi sijoittaa PHP-koodia. Alla olevassa esimerkissä PHP-koodi on kirjoitettu HTML "<?php" ja "?>"-elementtien väliin.

```
<p>
<?php
    Echo 'Kaksi plus kaksi on ' .Summa(2,2);
    //Tulostaa tekstin "Kaksi plus kaksi on 4".
?>
</p>
```

Tietojen lähetys HTML-lomakkeelta palvelimelle tietojen käsittelyä varten tapahtuu POST- ja GET-metodeilla. GET-metodissa tiedot siirretään URL:in kautta. POST-metodissa tiedot siirtyvät HTTP-protokollassa. POST-metodi on turvallinen arkojen tietojen käsittelyssä, mutta sen turvallisuus riippuu siitä miten HTTP on turvattu. GET-metodia ei ole suojattu millään, joten kuka vain näkee GET-metodilla lähetetyt tiedot.

PHP:llä suoritetaan myös tietokantaan liittyviä käskyjä.

5.3 JavaScript ja JavaScript-kirjastot

JavaScript on selaimessa suoritettava skriptikieli, jolla voidaan kasvattaa verkkosivun dynaamisuutta ja interaktiivisuutta. Javascriptin syntaksi on hyvin yksinkertainen ja on lähes samanlainen kuin PHP:n syntaksi. Toisin kuin PHP:n koodi, joka suoritetaan palvelimella, javascript-koodia suoritetaan verkkosivun latautumisen jälkeen tai jonkin lomaketoiminnon yhteydessä.

JavaScript-koodi voi olla omassa tiedostossa tai HTML-koodin mukana <script>-elementissä. Alla olevassa koodiesimerkissä on käytetty jQuery-kirjastoa ja JavaScript-koodi on kirjoitettu <script>-elementin sisään.

```
<script src="js/jquery.js"></script>
<script type="text/javascript">
    $(document).ready(function() {
        alert("Terve maailma!");
    });
</script>
```

JavaScript-kirjastot ovat valmiiksi kirjoitettuja JavaScript-ratkaisuja, joilla on helpon kehitystavan kehittää JavaScript-sovelluksia verkkosivuille.

jQuery-kirjastoa ja siihen perustuvia muita kirjastoja käytetään laajasti verkkosivujen toimintojen laajentamiseen ja niiden interaktiivisuuden kasvattamiseen. JQuery-kirjasto sisältää valmiita koodiratkaisuja ja JQuery on yleisin ja kattavin JavaScript-kirjasto.

5.4 SQL

SQL (Structured Query Language) on relaatiotietokannan hallintaa ja tietojen kyselyjä varten tehty kieli. SQL käytetään tietojen lisäämiseen, poistamiseen ja päivittämiseen relaatiotietokannassa. Taulut luodaan SQL:llä tai relaatiotietokannan omalla hallintaohjelmalla. SQL-koodia suoritetaan järjestelmässä PHP:n kanssa. Ennen SQL-koodin suorittamista rakennetaan SQL-lause. Lauseessa määritellään mitä tehdään, mille taululle ja millä ehdoilla. Alla oleva koodiesimerkissä PHP-koodi suorittaa SQL-kyselyn, jolla haetaan kaikki henkilötaulun etunimet ja sukunimet.

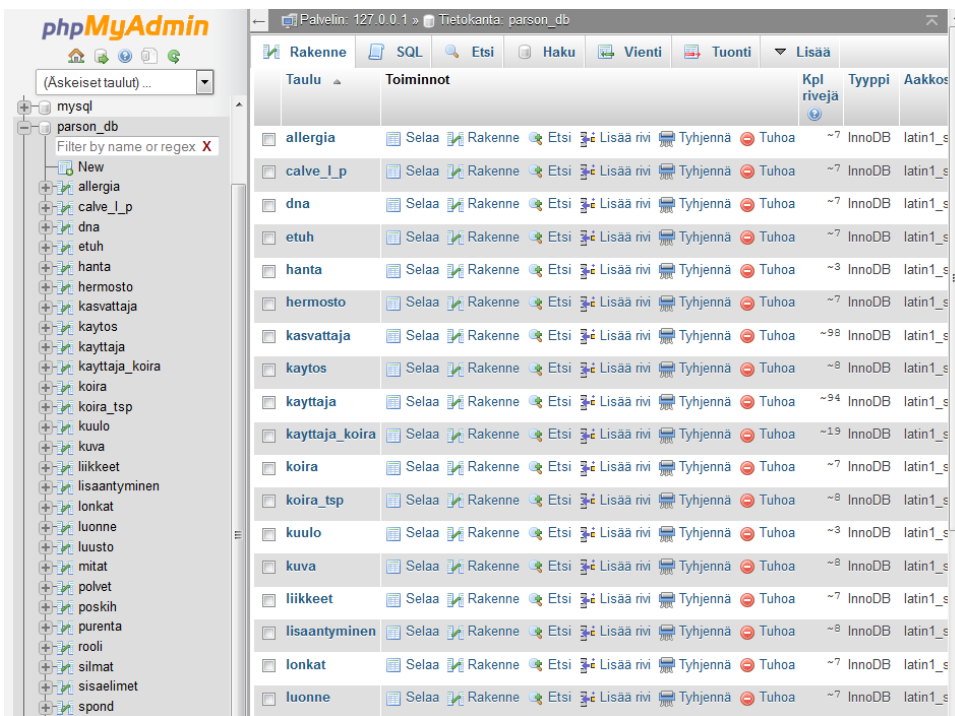
```
mysql_connect("localhost", "root", "salasana");  
mysql_db("tietokanta");  
$kysely = "SELECT etunimi, sukunimi FROM henkilo";  
$kyselynTulos = mysql_query($kysely);  
mysql_close();
```

5.5 MySQL ja PhpMyAdmin

MySQL:n omistaa ja kehittää Oracle. MySQL on suosittu ja yleisti verkkopalveluissa käytetty relaatiotietokanta, koska ilmaiset palvelinohjelmistot, muun muassa Apache, käyttävät MySQL:ää. Myös webhotellien palvelimet sisältävät yleensä MySQL:n.

MySQL:n käyttö tapahtuu komentorivitulkilla, mutta yleisin käytötapa on verkkoselaimessa toimiva PhpMyAdmin.

PhpMyAdmin on verkkoselaimen kautta toimiva käyttöliittymä, jolla kehittäjät pystyvät helposti käyttämään ja hallitsemaan tietokantoja. Yleiset ilmaispalvelinohjelmistot, jotka sisältävät MySQL:n sisältävät myös PhpMyAdmin-käyttöliittymän (Kuva 11).



Kuva 11. PhpMyAdmin.

5.6 CSS

CSS (Cascading Style Sheet) on tyyliohjekieli, jolla määritellään verkkosivun ulkonäköä. CSS-koodi tallennetaan omaan tiedostoon ja ladataan, kun verkkosivu latautuu tai sijoitetaan HTML-elementin style-attribuutin sisään. CSS:n syntaksi muodostuu valitsimesta, ominaisuudesta ja arvosta.

Valitsin on se, mihin HTML-elementteihin kyseinen CSS-koodi vaikuttaa. Valitsin voi olla luokka, joka vaikuttaa kaikkiin HTML-elementteihin, joilla on kyseinen luokka tai perustua HTML-elementin Id:hen tai sitten vaikuttaa kaikkiin HTML-elementteihin. Kun valitsin on valittu, voidaan muokata sen ominaisuuksia uusilla arvoilla. Seuraavassa koodissa määritellään oletusfontti ja sen väri.

```
p{
    font:"Comic Sans";
    color:Red;
}
```

Alla olevassa koodiesimerkissä CSS-koodia on sijoitettu p-elementin style-attribuutin sisään.

```
<p style="font:Comic Sans; color:red;">
    Esimerkki kappaleteksti
</p>
```

CSS-koodin voi sisällyttää <style>-elementin sisään, joka vaikuttaa sen kyseisen koko verkkosivun sisältöön, mutta on suositeltavaa tallentaa se omaan tiedostoon koodin organisoinnin helpottamiseksi.

```
<style type="text/css">
p{
    font:"Comic Sans";
    color:Red;
}
</style>
```

6 Käytetyt ohjelmat

Verkkosivujen kehitykseen, testaukseen ja hallintaan tarvitaan erilaisia ohjelmia. Kehitystyön aikana tarvitaan kehittäjän tietokoneeseen testausympäristö, jossa ohjelman toimintaa kokeillaan. Lähdekoodin kirjoittamiseen tarvitaan tekstieditori. Tiedostojen siirtoon varsinaiselle palvelimelle tarvitaan tiedostosiirtoprotokollaa tukeva ohjelma. Tietokannan luomiseen ja testaamiseen tarvitaan tietokannan hallintaohjelmisto.

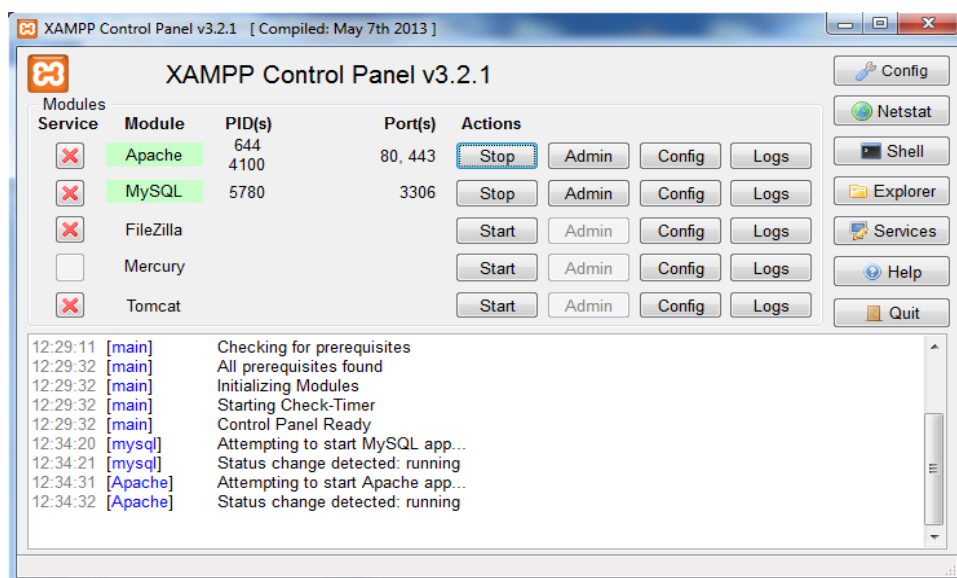
6.1 XAMPP

XAMPP on ohjelmistokokoelma, joka sisältää Apache-verkkopalvelimen, MySQL-tietokannan ja rajapinnat PHP- ja PERL-ohjelmointikielille sekä muita palvelinsovelluksia. XAMPP on tarkoitettu palvelinsovellusten rakentamiseen ja testaamiseen. XAMPP:ia voi käyttää verkkopalvelimena, mutta se ei ole tarkoitettu siihen.

XAMPP tulee sanoista Apache, Mysql, PHP ja PERL. X tarkoittaa cross-platform eli monialusta.

XAMPP:ia kehittää Apache Friends yhteistö.

XAMPP:in osia hallitaan sen omasta hallintapaneelistä (Kuva 12).



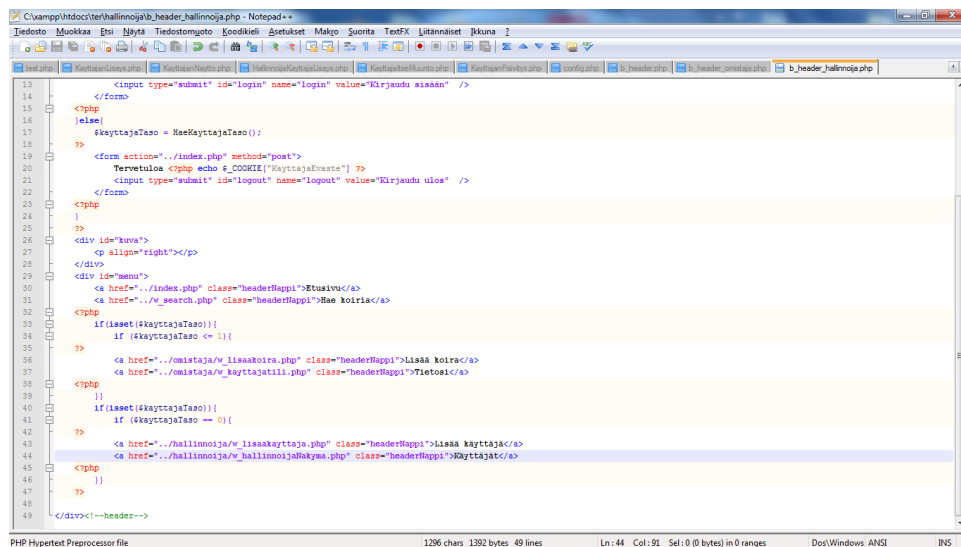
Kuva 12. XAMPP-hallintapaneeli.

6.2 Notepad++

Notepad++ on ilmainen tekstieditoriohjelma. Notepad++ toimintoja voi kasvattaa lataamalla ja asentamalla laajennusosia.

Notepad++ käytetään yksinkertaisena koodin kirjoitusohjelmana.

Notepad++ (kuva 13) sisältää tekstin korostuksen ohjelmointikielille ja rivilaskurin.



```
13 <input type="submit" id="login" name="login" value="Kirjautu sisään" />
14 </form>
15 <?php
16 |else|
17 $kayttajaTaso = HaeKayttajaTaso();
18 >>
19 <form action="..../index.php" method="post">
20 Tervetuloa <?php echo $_COOKIE["KayttajaEvaaste"] ?>
21 <input type="submit" id="logout" name="logout" value="Kirjautu ulos" />
22 </form>
23 <?php
24 |
25 >>
26 <div id="kuva">
27 <p align="right"></p>
28 </div>
29 <div id="menu">
30 <a href="..../index.php" class="headerflappi">Etusivu</a>
31 <a href="..../w_search.php" class="headerflappi">Bae koira</a>
32 <?php
33 |
34 |if (isset($kayttajaTaso))|
35 |if ($kayttajaTaso <= 1)|
36 >>
37 <a href="..../omistaja/w_lisaakoira.php" class="headerflappi">Lisää koira</a>
38 <a href="..../omistaja/w_kayttajaili.php" class="headerflappi">Tietosi</a>
39 >>
40 >>
41 |if (isset($kayttajaTaso))|
42 |if ($kayttajaTaso == 0)|
43 >>
44 <a href="..../hallinnoija/w_lisaakayttaja.php" class="headerflappi">Lisää käyttäjä</a>
45 <a href="..../hallinnoija/w_hallinnoijaakya.php" class="headerflappi">Käyttäjät</a>
46 >>
47 >>
48 </div><!--header-->
49
```

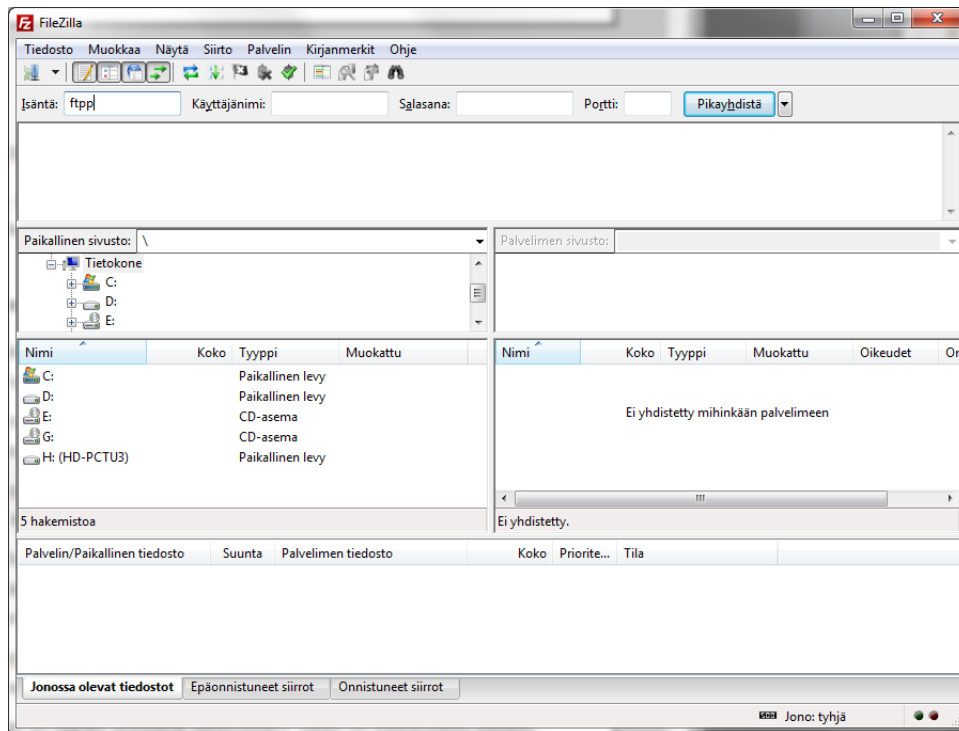
Kuva 13. Notepad++:n ohjelmaikkuna.

6.3 FileZilla

FileZilla on ohjelma, jolla siirretään verkkosivun toimintaan liittyvät tiedostot ohjelmoijan koneelta verkkopalvelimille ja verkkopalvelimilta ohjelmoijan koneelle. FileZilla on ilmainen ja ladattavissa valmistajan kotisivuilta.

FileZilla:lla avataan yhteys antamalla verkkopalvelimen osoite, käyttäjänimi ja salasana. Tietokoneen oma palomuuuri voi haitata yhteyden muodostamista.

FileZilla:ssa (Kuva 14) yläikkunassa tulostetaan kaikki palvelimen FTP:n kautta tapahtuva liikenne. Oikeassa ikkunassa on palvelimen tiedostopuu ja vasemmassa on oma tietokoneen tiedostopuu. Alaikkunassa on sen hetken tiedostojen siirtojen tila.



Kuva 14. FileZilla:n ohjelmaikkuna.

7 Tietoturva

Tietoturvan merkitys korostuu web-sovelluksissa, koska järjestelmä on yhteydessä maailmanlaajuiseen internetverkkoon ja on helposti hakkereiden saavutettavissa, käyttäjiä on paljon ja toteutuksessa käytetään useita eri teknologioita, joiden yhteensopivuudesta ei voi olla varma.

Perinteisessä tiedon arvoon perustuvassa määritelmässä tietoturvallisuus koostuu kolmesta osatekijästä:

- Luottamus: tietoa käsittelevät vain ne henkilöt, joilla on oikeus siihen.
- Eheyys: tieto ei saa muuttua tahattomasti käytössä tai verkkohyökkäyksessä. Muutos tulee havaita.
- Käytettävyys: tieto on saatavilla nopeasti ja oikeassa muodossa.

Luottamuksellisuuden ylläpitoon pyritään suojaamalla tietojärjestelmien laitteet ja tietovarastot käyttäjätunnuksin ja salasanoin. Erilaiset salakirjoitusmenetelmät soveltuvat myös arkaluontoisten tai arvokkaiden tietojen suojaamiseen. (Hakala 2006, s.4.)

Käytettävyyttä ylläpidetään huolehtimalla siitä, että tieto- ja tietoliikennejärjestelmien laitteet ovat riittävän tehokkaita ja että käytettävät ohjelmistot soveltuvat mahdollisimman hyvin järjestelmään tallennettujen tietojen käsittelyyn. Pyritään myös automatisoimaan tiedon jalostus. Käyttäjän tulisi saada haluamansa tiedot oikeassa muodossa. (Hakala 2006, s. 4 - 5.)

Eheyteen pyritään pääasiassa ohjelmointiteknisin ratkaisuin. Sovelluksiin ohjelmoidaan erilaisia syöttörajoitteita tai syötteen tarkistuksia, tallennus- ja tiedonsiirto-operaatioihin varmistussummia tai tiivisteitä. Laitteistotasolla pyritään estämään virheet käyttämällä esimerkiksi virheenkorjaavia muisteja tai väyliä. Tietoliikennetkaisuissa suositaan virheen tunnistus- ja korjausmekanismeilla varustettuja protokollia ja laitteita. Useimmat salakirjoitusmenetelmät ja -tuotteet soveltuvat myös eheyden ylläpitoon. (Hakala 2006, s. 5.)

Klassinen tietoturvallisuuden määritelmä on lyhyt ja tiivis ja se sisältää olennaiset osat, joista on huolehdittava ennen muita tietoturvallisuuden tekijöitä. (Hakala 2006, s. 5.)

7.1 Roolit

Useimmissa järjestelmissä käyttäjien oikeudet on jaettu rooleihin. Roolit määrittelevät käyttäjän oikeudet toimintoihin ja tietoihin.

Hallinnoija-roolilla on kaikki käyttöoikeudet ja hallinnojan roolin omaavia käyttäjiä järjestelmässä on vain ja ainoastaan yksi. Hallinnoijalla on käyttöliittymässä omat työkalunsa, jotka eivät näy muilla rooleilla. Koska hallinnoijalla on kaikki käyttöoikeudet muodostaa, se oman tietoturvariskin.

7.2 Salaus

Järjestelmässä kaikilla käyttäjätileillä on oltava riittävän pitkä salasana, joka on salattava yksisuuntaisella hash-salauksella. Yksisuuntaisella salauksella tarkoitetaan, että kun tieto on salattu, salausta ei voi purkaa. Jos käyttäjä unohtaa salasanan, tulee hänen ilmoittaa siitä hallinnoijalle. Hallinnoija vaihtaa ja ilmoittaa käyttäjälle uuden salasanan.

Käyttäjän tiedot palvelimella eivät ole koskaan selkokielellä luettavissa. Kaksisuuntainen salaus toimii avainsanalla. Kaksisuuntaisella salatut tiedot näkyvät ainoastaan niille käyttäjille, joille ne ovat tarkoitettu.

Salauksen avulla pyritään estämään tiedon käyttö. Salauksessa selkokielineen tieto muunnetaan salausavaimen tai salausmenetelmän avulla kryptatuksi eli lukukelvottomaksi.

7.3 SQL-injektio

SQL-injektio on hakkerointitapa, jolla yritetään saada palvelin suorittamaan haitallista SQL-koodia, jolla ongitaan käyttäjätietoja. SQL-injektioita ei voi 100% estää mutta niiden käytön voi tehdä hyvin hankalaksi.

SQL-injektio tapahtuu syöttämällä verkkoselaimen osoitekenttään tai tekstinsyöttökenttiin SQL-ehto tai -komento.

Esimerkiksi sisäänkirjautumislomakkeella voi käyttäjänimi kenttään syöttää mitä tahansa aakkosia, kun salasana kenttään syöttää `" OR '1' = '1"` Tämä saa SQL-lauseen näyttämään alla olevan koodin mukaiselta. SQL-lauseeseen on asetettu uusi ehto, joka on aina tosi ja päästää käyttäjän sisään järjestelmään.

```
SELECT nimi FROM kayttajat WHERE käyttäjänimi='abc' AND salasana='' OR '1' = '1';
```

SQL-injektio estotapoja on useita. Yleisimmät estotavat ovat syötteen siivoaminen SQL-koodin suoritusta haittaavista merkeistä ja käyttämällä esivalmisteltuja SQL -lauseita, joihin on määritelty haluttu tapahtuma ja muuttujat.

Jos SQL-lauseen suoritus epäonnistuu, ohjataan käyttäjä virhesivustolle, jossa kerrotaan, että virhe tapahtui, mutta ei kerrota miksi se tapahtui, koska hakkeri pystyy tavanomaisesta virheviestistä päättämään minkälainen tietokanta on kyseessä ja siten parantelemaan hänen SQL-injektioitaan.

SQL-injektio estokeinoja käytetään kaikkiin kenttiin, missä käyttäjä voi syöttää tietoja ja verkko-osoitteisiin, jotka sisältävät tietoja.

7.4 Brute force -hyökkäys

Brute force -hyökkäyksellä hakkeri arvaa useita kertoja salasanaa. Tähän käytetään tietokonetta, mikä kokeilee jokaista salasanaa sen salasanalistassa.

Brute forcea -hyökkäystä ei voi estää, mutta sen toimintaa voi hidastaa. Yleisin tapa on käyttää CAPTCHA:aa. Kun hyökkäyskone on yrittänyt kirjautua sisään kaksi kertaa, kolmannella kerralla järjestelmä kysyy CAPTCHA:n, joka on kuva-arvoitus. Koska hyökkäyskonetta ei ole suunniteltu ratkomaan kuva-arvoituksia jää se jumiin. CAPTCHA:lla siis varmistetaan, että käyttäjä on ihminen.

Paras esto menetelmä brute force -hyökkäyksiä vastaan on valistaa käyttäjää välttämään helposti arvattavissa olevia salasanoja ja suosittelemaan käyttämään pitkiä salasanoja.

7.5 Istunto

Yleensä verkkopalveluissa kun käyttäjä kirjautuu järjestelmään, luodaan hänelle istunto, joka on voimassa tietyn ajan. Jos käyttäjä jättää koneen auki ja lähtee pois pitkäksi ajaksi, istunto lopettaa itsensä ja kirjaa käyttäjän automaattisesti pois järjestelmästä. Tällä vältetään, että joku vieras yrittää varastaa käyttäjän tiliä tai tietoja hänen avoimelta tietokoneelta.

7.6 Varmuuskopio

Tietoa kopioidaan tiheään tietokannoista ja muista tietovarastoista mahdollisen ilkkivallan ja laitteisto rikkoutumisien takia. Webhotelleissa on yleensä työkalut varmuuskopioiden luomiselle ja PhpMyAdmin sisältää vienti ja tuonti työkalut tietokantojen varmuuskopiointia varten.

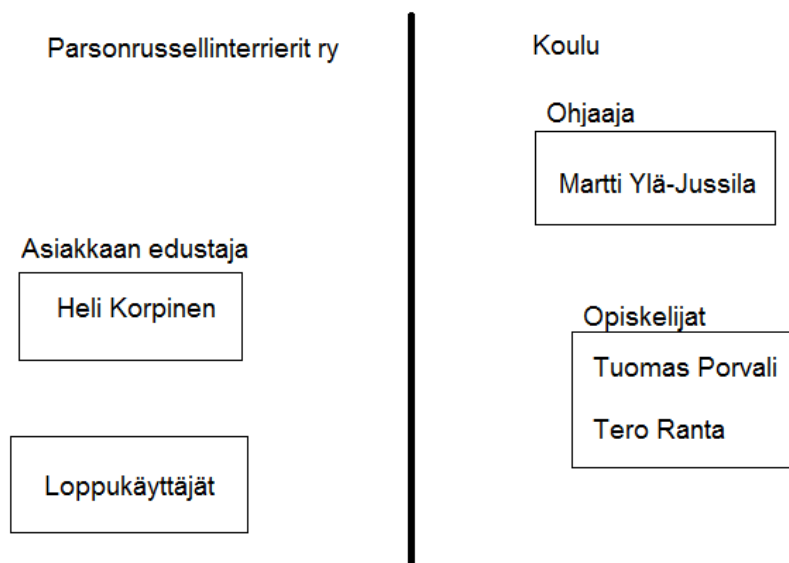
8 Opinnäytetyöprojektin toteutus

Käyttöliittymän toteutus alkoi helmikuussa 2014. Kun tulin mukaan projektiin, järjestelmästä oli olemassa tekninen määrittely ja tietokantakaavio.

8.1 Projektin suunnittelu ja hallinta

Opinnäytetyötä tein yhdessä Tero Rannan kanssa, joka oli jo saanut valmiiksi tietokantakaavion, itse tietokannan, käyttötapauskaavion ja toiminnalliset määrittelyn järjestelmästä.

Scrumia sovellettiin työaikana. Joka viikko pidettiin seurantakokous, joka kesti noin 15 minuuttia ja kokouksesta kirjoitettiin muistio, johon kirjoitettiin aikaansaannokset, tulevat muutokset ja riskit. Kokoukseen osallistuivat asiakkaan edustaja Heli Korpinen, ohjaaja Martti Ylä-Jussila, Tuomas Porvali ja Tero Ranta (Kuva 15).



Kuva 15. Organisaation rakenne.

Kesäkuusta lähtien aloin luovuttamaan iteraatioita asiakkaalle testattavaksi.

Kehityksessä käytettiin HTML4 versiota ja osittain HTML5 versiota. HTML5 tuki on kaikilla selaimilla vielä kehno, joten kokonaan sivuja ei voitu tehdä HTML5:lla.

8.2 Projektisuunnitelma

Projektisuunnitelman kirjoitti Tero Ranta. Siihen kirjoitettiin projektin tavoitteet, organisaatio ja vastuut, riskit ja projektin vaiheet, tehtävät ja niiden työmääräarviot sekä aikataulu.

8.3 Tietokannan suunnittelu

Tietokannan suunnitteli Tero Ranta. Projektin aikana siitä muutettiin jatkuvasti asiakkaan tai käyttöliittymän vaatimusten mukaan.

8.4 Vaatimusmäärittely

Työ alkoi helmikuussa 2014. Järjestelmästä oli silloin olemassa Tero Rannan tekemä tietokanta. Heli Korpiselta saamme kuvan toisista terveystietokannasta, Suomen Valkoinen paimenkoira.

Vaatimusmäärittely kirjoitettiin IEEE830-standardin pohjalle. Vaatimusmäärittely dokumentti tehtiin hyvin myöhään kehityksen aikana. Osallistuin sen kirjoittamiseen, kun käyttöliittymä oli testausvaiheessa. Käyttöliittymä ja toiminnot suunniteltiin ja toteutettiin toiminnallisen määrittelyn ja käyttötapausten pohjalta ja asiakkaan palaute sen hetkisestä iteraatiosta.

Itse kirjoitin vaatimusmäärittelyyn käyttötapaukset, kaaviot ja kuvat.

8.5 Webhotellin tilaaminen

Webhotelli ja verkkotunnus tilattiin OVH-hosting-palvelulta. Tilattu webhotelli sisältää kaikki tarvittavat verkkosivun isännöintiin tarvittavat työkalut, mm. FTP, phpMyAdmin ja muu tarpeellinen.

Tilaukseen tarvittiin OVH-asiakastili. Asiakastili luotiin aluksi yksityishenkilönä, koska luomisyhteudessa emme ymmärtäneet, että y-tunnus tarkoitti myös yhdistyksen rekisterinumeroa. Valitsimme pienen webhotellin, joka tarjosi riittävät ominaisuudet tarpeisiin. Webhotelli tilattiin OVH:n kotisivuilta (Kuva 16).



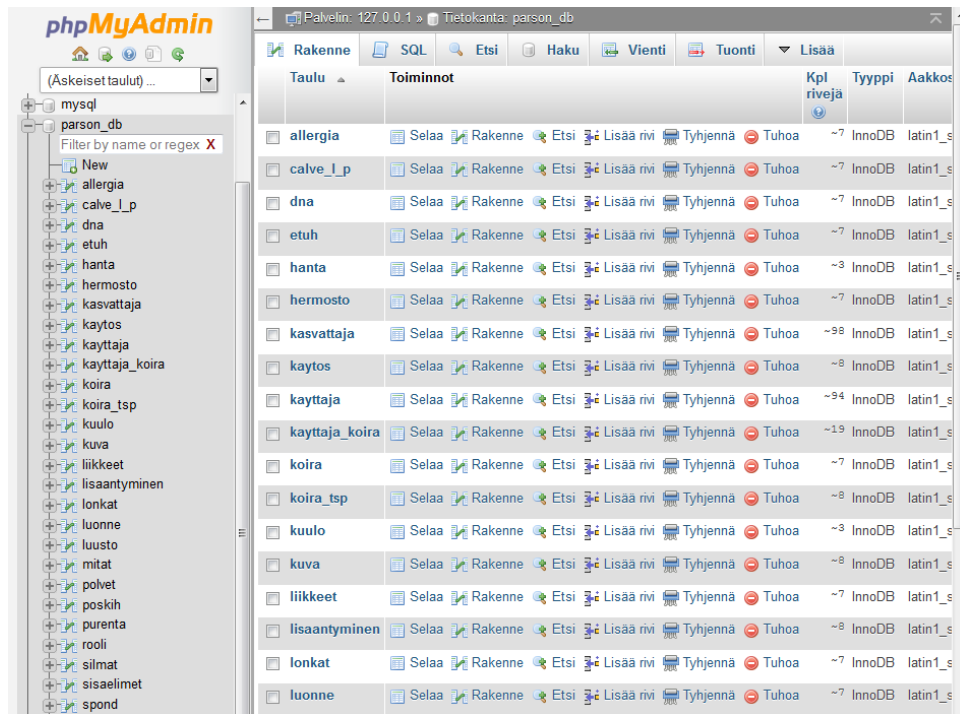
Kuva 16. OVH:lta tilattiin webhotelli.

Verkkotunnukseksi päätettiin parsonterveys.fi, joka rekisteröitiin yhdistykselle. Webhotellin vuokraus ja verkkotunnuksen rekisteröinti vuodeksi maksoi yhteensä noin 45 euroa.

Heti, kun lasku oli maksettu, OVH asensi webhotellin valmiiksi, mutta verkkotunnuksen rekisteröinnissä OVH:llä oli ongelmia. Viikkojen jälkeen otin yhteyttä OVH:n asiakaspalveluun hitaan verkkotunnuksen rekisteröinnin takia. Asiakaspalvelu korjasi asian ja pahoittelivat verkkotunnuksen rekisteröinnin hitautta.

Oletin alkusi, että tietokanta luodaan webhotellin rekisteröinnin yhteydessä, mutta tietokanta luodaan erikseen. Vasta luotua tietokannan sain tunnukset tietokannanhallintasivulle.

Tietokannan hallinta tapahtui täsmälleen samalla työkalulla kuin XAMPPin hallinta. Kopioin tyhjän tietokannan SQL-tiedostoon ja siirsin sen webhotellin tietokantaan käyttämällä OVH:n phpMyAdminia (Kuva 17).



Kuva 17. PhpMyAdmin

Siirsin Filezillailla HTML-, PHP-, JavaScript- ja CSS-tiedostot webhotellin tiedostopalvelimelle.

Aluksi sivusto ei toiminut. Haussa ja kirjautumisessa oli ongelmia ja viaksi paljastui suuraakkoset taulujen nimissä SQL-koodissa. Tietokannassa taulujen nimet oli kirjoitettu pienaakkosilla.

8.6 Verkkosivujen rakentaminen

Ensimmäiset sivut, jotka tein, olivat Etusivu, Koiranlisäyssivu, Hakusivu ja yksinkertainen Koirannäyttösivu. Tosin sivut toimivat vain ulkonäön puolesta, ne eivät sisältäneet mitään toiminnollisuutta. Sivuilla oli tarkoitus esitellä mahdollista ulkonäköä ennen varsinaista toiminnallisuutta. Ensimmäisen toiminnallisuus, jonka tein, oli sisäänkirjautuminen.

Ensimmäinen versio sisäänkirjautumista toimii siten, että verkkokäyttöliittymä tallensi evästeeseen käyttäjän ID:n, käyttäjänimen ja käyttäjäröolin. Koska tämä ratkaisu sisälsi tietoturvariskin, muunnettiin se myöhemmin sellaiseksi, joka tallentaa evästeeseen käyttäjän generoidun ID:n ja käyttäjänimen. Generoidusta ID:stä ja käyttäjänimestä verkkosivu päättelee käyttäjäröolin ja niiden on oltava toisiinsa täsmääviä. Jos ne eivät ole, käyttäjän istunto lopetetaan.

Kun kaikki sivut saatiin valmiiksi, alkoi niiden toiminnallisuuden rakentaminen. Toiminnallisuuden rakentamisessa ei ilmennyt mitään työtä estäviä ongelmia.

8.7 Testaus

Testausvaiheessa XAMPPia käytettiin varsinaisen palvelimen korvikkeena, jolla testattiin sivuston koko toimivuus. Aina, kun toiminto saatiin valmiiksi, testattiin se kaikilla yleisillä selaimilla.

Itse testasin käyttöliittymää koko kehityksen ajan. Asiakas itse testasi viikoittaisia iteraatioita kannettavalla tietokoneella kesäkuussa kunnes siirryttiin kokonaan webhotellin käyttöön lokakuussa 2014.

Sivustoa alettiin testaamaan, kun sillä pystyi itserekisteröitymään ja lisäämään koiria. Testauksessa auttoi kesäkoulun työnjohtaja Anne Paakko ja asiakkaan edustaja Heli Korppinen. He testasivat ja raportoivat löytämänsä ohjelmointivirheet, kielioppivirheet ja Heli Korppinen esitti joitakin muutospyyntöjä.

Itse testasin käyttöliittymää heinäkuun ja elokuun ajan. Oma testaus oli kokonaisuudessaan positiivista testausta.

Syyskuun lopussa saatiin sivusto webhotelliin ja aloitettiin koekäyttö. Koekäytössä ei ilmennyt suuria käyttöongelmia ja testiaikana tehdyt muutokset oli pieniä.

Kun käyttöliittymä siirtyi koekäyttöön, aloin testaamaan käyttöliittymää muilla verkkoselaimilla. Testaus oli järjestelmällistä, eli tein listan testattavista asioista ja testasin ne läpi kaikilla kohdeselaimilla.

Ammattikorkeakoulu järjesti testauskurssin, jossa opiskelijat testasivat käyttöliittymää. Testauskurssilla muodostettiin ryhmä, joka kehitti testitapaukset minun antamieni esimerkkitestitapausten mukaan. Testaajat oli jaettu pareihin, jotka testasivat käyttöliittymän eri selaimilla ja lähettivät testitapausten tulokset minulle. Testisuunnitelma myöhästyi pahasti. Kurssiryhmä käytännössä kävi minun esimerkkitestitapauskiani läpi ilman mitään suunnitelmaa. Testitapaukset ajettiin kolme kertaa läpi kaikilla selaimilla. Testauskurssi meni hyvin ja vikoja löydettiin. Testauksesta huolimatta pikkuvirheitä löytyy yhä.

8.8 Käyttöönotto

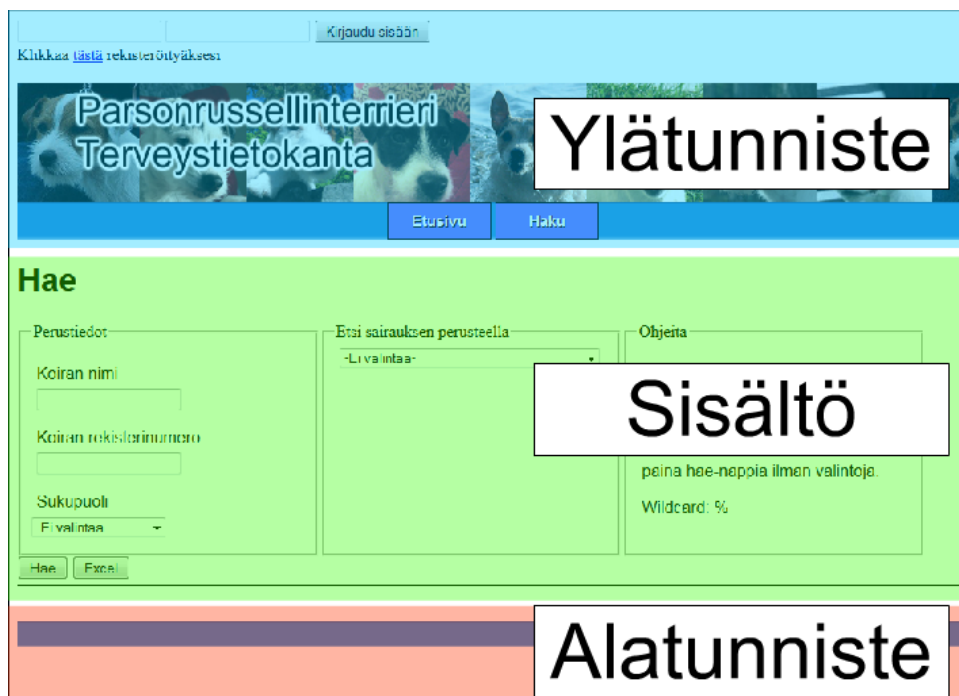
Käyttöönotto tapahtui keväällä 2015. Asiakas teki omat hyväksymistestit. Asiakkaalle kirjoitettiin pääkäyttäjänohje ja loppukäyttäjille kirjoitettiin aloitusohje.

9 Lopputuotteen ja ratkaisujen esittely

Käyttöliittymän suunnittelun toteutus aloitettiin tekemällä malleja tulevista sivuista. Sain Tero Rannalta, silloiselta projektipäälliköltä, mallin hakusivusta. Hakusivusta tein pohjan muille sivuille.

9.1 Verkkosivujen rakenne

Verkkosivu rakentuu ylätunnisteesta, sisältölohkosta ja alatunnisteesta (Kuva 18). Ylätunnisteloiko sisältää kaikki linkit muille sivuille ja kentät sisäänkirjautumisille ja uloskirjautumiselle. Sisältöloiko kenttää sisältä esimerkiksi hakusivulle tarvittavat ohjelmointikoodit ja skriptit, jotka mahdollistavat haut tietokannasta. Alatunnisteloiko on symbolisoimassa verkkosivun loppua.



Kuva 18. Verkkosivun rakenne.

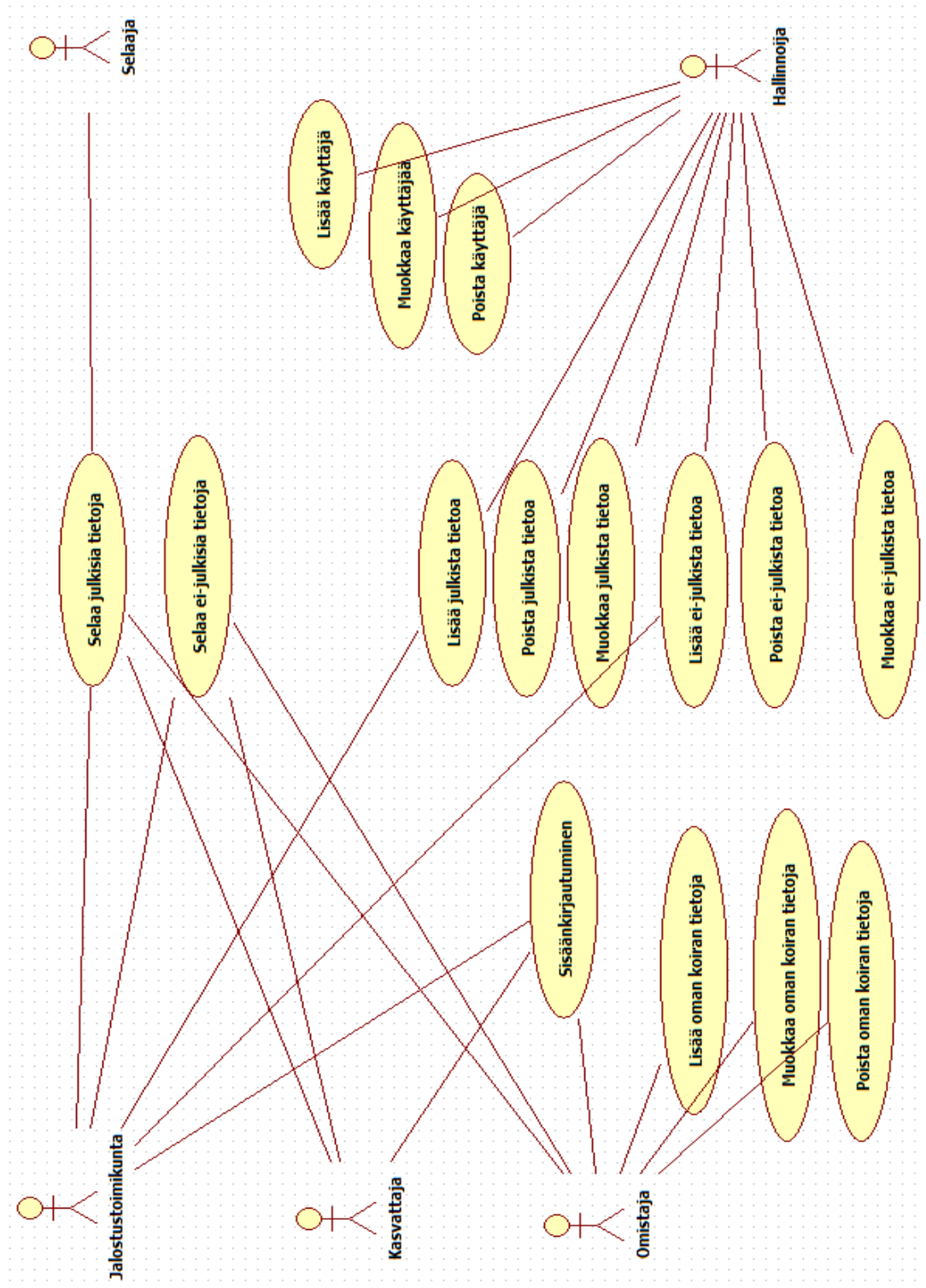
Ylätunnisteloheko, sisältöloheko ja alatunnisteloheko ovat omia PHP-tiedostoja. Lohkotiedosto on nimetty niiden käyttötarkoituksen mukaan ja nimi sisältää "b_" -etuliitteen. Verkkosivutiedosto sisältää yllä mainitut olevat lohkot ja tiedosto nimetään sen käyttötarkoituksen mukaisesti ja nimi sisältää "w_" -etuliitteen

Käyttöliittymässä on hakusivu, joka on nimetty "w_haku" ja se sisältää lohkot "b_ylätunniste", "b_alatunniste" ja "b_haku". Kuvassa 18 kuvataan hakusivun rakenne. Päädyin tällaiseen ratkaisuun, on luodun koodin voi jakaa useisiin helposti hallittaviin osiin. Tiedosto hallinnassa päätin myös, että kaikki CSS-koodi on omassa tiedostossaan.

JavaScript-koodi sisällytetään samaan lohkotiedostoon vaikka sen voi laittaa omaan tiedostoon. Syy tähän ratkaisuun oli hallittavuus. Koodia on paljon helpompi hallita, kun lohkon sisältö ja JavaScript-koodi ovat samassa tiedostossa. PHP-tiedosto ja JavaScript-kirjastot sijoitettiin omiin kansioihin järjestelmässä.

9.2 Käyttötapauskaavio

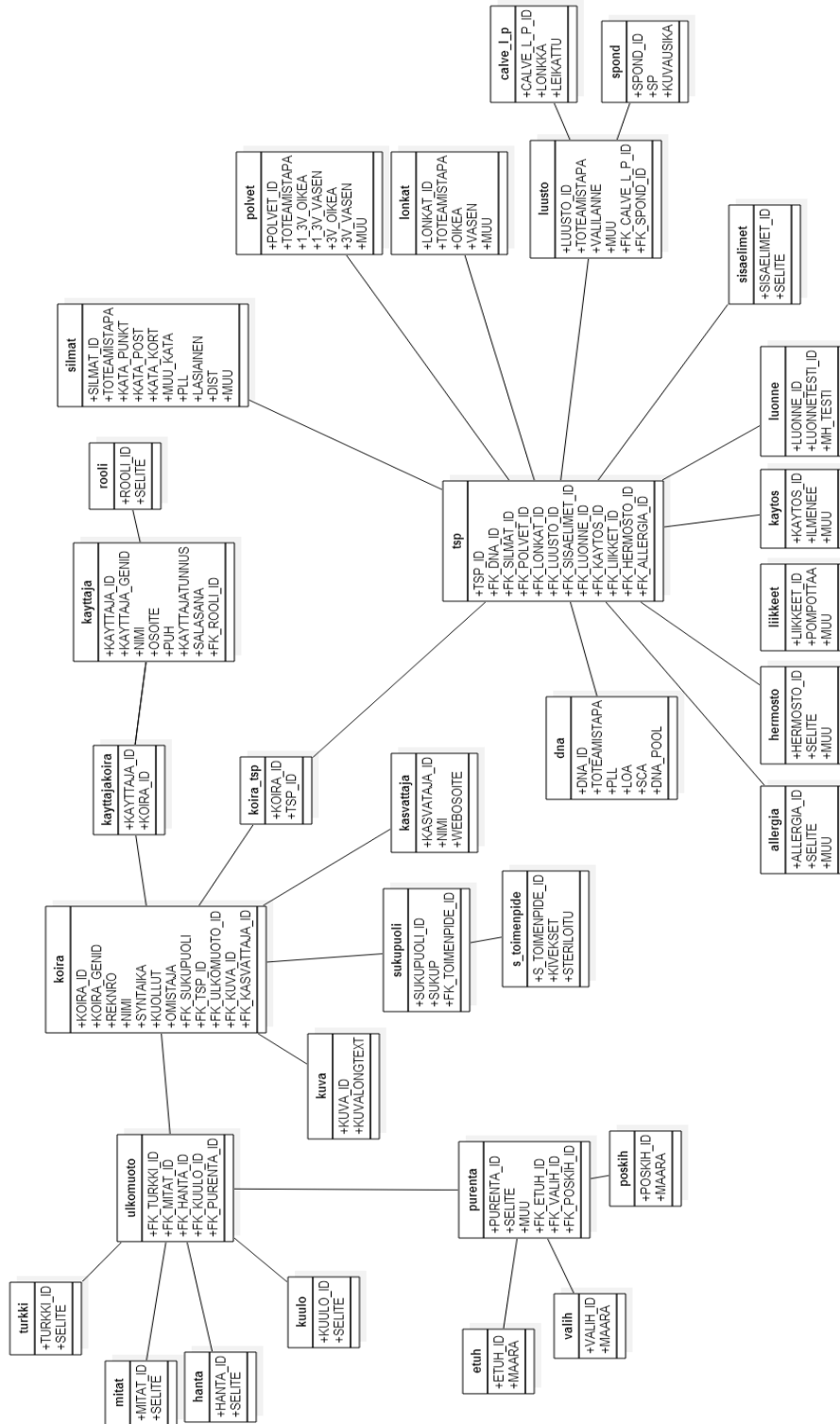
Järjestelmän kaikki käyttäjät ja heidän käyttötapaukset (Kuva 19).



Kuva 19. Käyttötapauskaavio.

9.3 Tietokantakaavio

Järjestelmän tietokantakaavio (Kuva 20).



Kuva 20. Tietokantakaavio.

9.4 Tietoturvaratkaisut

SQL-injektio oli minulle täysin uusi asia. Pian opin tekemään omia SQL-injektioita ja murtautumaan omaan järjestelmään. Sen hetkiset SQL-kyselyt piti kirjoittaa täysin alusta. Uudelleenohjauksilla pyritään estämään luvaton pääsy tietoihin, joihin käyttäjälle ei ole oikeutta. Istunnon luonti, salaukset ja varmuuskopiointi ovat perusratkaisuja, jotka ovat löydettävä mistä tahansa järjestelmästä.

9.5 Optimointi ja tehokkuus ratkaisut

Huomasin, että aina kun yhteys avattiin tietokantaan ja suljettiin, kasvoi sivun latausaika. Sovelsin ratkaisuna keinoa, että yhteys tietokantaan avataan vain kerran. Latausaika pieneni 12 sekunnista alle 2 sekuntiin.

Hakusivulla haku-aika kesti yli 20 sekuntia vaikka tietokannassa oli vain yksi koira. Vanha haku toimi sitten, että se linkitti kaikki taulut hakua varten ja se kasvatti haku-aikaa, koska haku kävi kaikki linkitetty taulut läpi.

Ratkaisuna oli linkittää vain haulla välttämättömät taulut ja ohjelma katsoo onko tarpeellista linkittää muita tauluja. Haku-aika pieneni alle 5 sekuntiin.

Ensimmäiset sivut, jotka tein olivat Etusivu, Koiranlisäyssivu, Hakusivu ja yksinkertainen Koirannäyttösivu. Tosin sivut toimivat vain ulkonäön puolesta, ne eivät sisältäneet mitään toiminnollisuutta. Sivuilla oli tarkoitus esitellä mahdollista ulkonäköä ennen varsinaista toiminnollisuutta. Ensimmäisen toiminnollisuus, jonka tein oli sisäänkirjautuminen.

Ensimmäinen versio sisäänkirjautumista toimii siten, että verkkokäyttöliittymä tallensi evästeeseen käyttäjän ID:n, käyttäjänimen ja käyttäjäröolin. Koska tämä ratkaisu sisälsi tietoturvariskin, muunnettiin se myöhemmin sellaiseksi, joka tallentaa evästeeseen käyttäjän generoidun ID:n ja käyttäjänimen. Generoidusta ID:stä ja käyttäjänimestä verkkosivu päättelee käyttäjäröolin ja molemmat on oltava toisiinsa täsmäviä. Jos ne eivät ole, käyttäjän istunto lopetetaan.

9.6 Samanaikaisuus

Testauksessa koirien tietojen muokkauksessa ilmeni ongelmia. Jos kaksi käyttäjä muokkaa saman koiran tietoja yhtä aikaa, se käyttäjä, joka tallentaa viimeisenä. Hänen muutokset jäävät voimaan. Ne käyttäjät, jotka tallensivat aikaisemmin, heidän muutokset katoavat. Ratkaisuna tähän tilanteeseen ilmoitetaan muille käyttäjille yhtäaikaisesta muokkauksesta asettamalla lippu yhtäaikaisesta muokkauksesta. Lippu yhtäaikaisesta muokkauksesta pysyy 22 minuuttia voimassa tai kunnes sen asettaja klikkaa tallenna-painiketta. Aikarajoitus poistettiin käytöstä palveluntarjoajan asettamien rajoitusten takia.

9.7 Käyttöliittymäratkaisuja

Käyttöliittymän suunnittelussa hyödynsin nykytrendejä ja omia mieltymyksiä. Esimerkiksi päävalikon painikkeet ylätunnisteessa on järjestetty vaakariviin. Ennen ne olisi järjestetty sivun sisällön viereen. Käyttöliittymän ulkoasusta tein mahdollisimman pelkistetyn: ei liukuvärejä, vähintään kaksi – kolme väriä yhtä aikaa, ei kuvia.

Koiran tietojen selaussivulla tiedot on järjestetty eri välilehtiin. Valitsin välilehtiratkaisun, koska se järjestää tiedot selkeästi ja tekee koodin hallinnasta helppoa. Vaihtoehtoisina ratkaisut oli jäsentää tiedot omiin sivuihin mutta se olisi tehnyt tietojen selaamisesta hyvin hidasta ja vaivalloista. Toinen vaihtoehtoinen ratkaisu olisi ollut sijoittaa kaikki tiedot samalla sivulle näkyviin, mutta se olisi tehnyt sivun selaamista vielä vaivalloisempaa. Vaivalloisuutta olisi voinut vähentää lisäämällä valikon, joka vie halutulle lomakkeen kohdalle mutta ei olisi eliminoinut edestakaisin vieritystä lomakkeella.

Kaikki mahdollinen, mikä helpottaa käyttäjäkokemusta, otettiin käyttöön. Esimerkiksi riittää, kun klikkaa valintalaatikon vieressä olevaa tekstiä toiminnon aktivoimiseksi (Kuva 21).

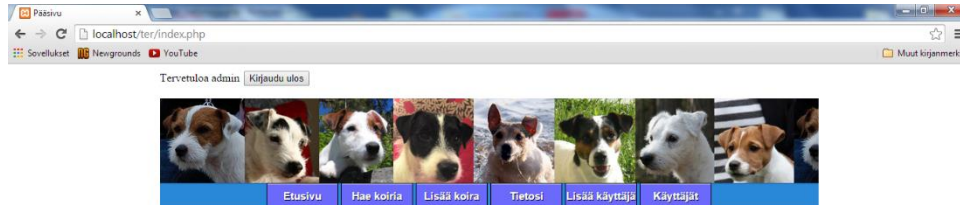


Kuva 21. Käytön helppottamista.

Myös asiakkaan toiveet otettiin huomioon käyttöliittymä suunnittelussa. Esimerkiksi päivämäärän valinnassa kuukauden ja vuoden valinta tehtiin helpoksi.

9.8 Etusivu ja sisään kirjautuminen

Etusivu tarjoaa linkit muihin sivuille. Sisäänkirjautumiseen tarvittavat kentät tarjotaan aina kaikilla sivuilla.



Kuva 22. Etusivu.

9.9 Itserekisteröinti


Itserekisteröintisivu tarjotaan sisäänkirjautumiskentän vieressä. Kaikki käyttäjät, jotka rekisteröityvät itse, saavat omistajakäyttöoikeudet.

Jos käyttäjällä ei ole tunnusta sivulle kirjautumiseen, voi hän rekisteröityä sivulle. Rekisteröinti tapahtuu rekisteröintisivulla (kuva 23), jossa käyttäjän on syötettävä koko nimensä, puhelinnumero, osoite, käyttäjänimensä ja salasana kahdesti.

Kun tiedot on syötetty ja käyttäjä klikkaa rekisteröidy, tietokantaan lisätään hänen tiedot ja kanta rakentaa tarvittavat tunnistetiedot. Käyttäjät, jotka itserekisteröivät itsensä järjestelmään, saavat omistaja-roolin, eli he voivat saman tien lisätä koiria järjestelmään.

Jos käyttäjä haluaa vaihtaa roolia, hänen täytyy lähettää pyyntö siitä hallinnoijalle.

Klikkaa [tästä](#) rekisteröityäksesi



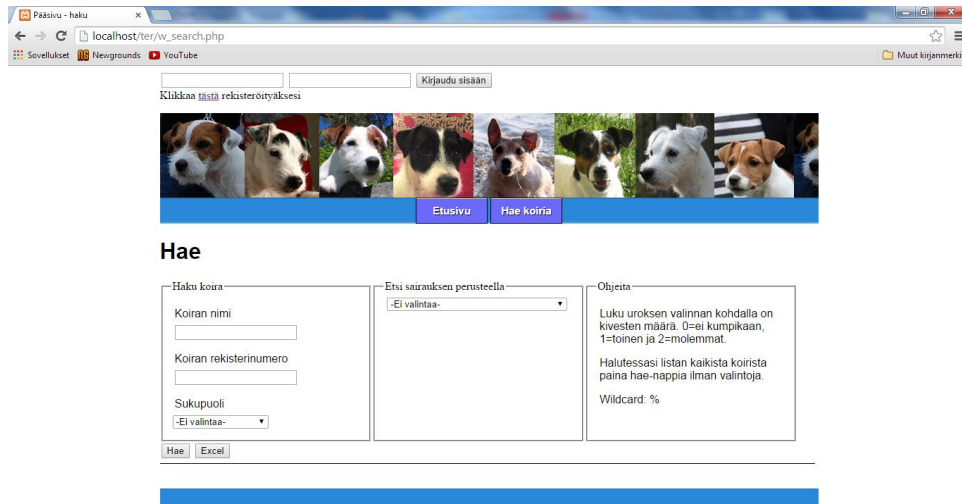
Rekisteröinti

Henkilötiedot	Käyttäjätiedot
Nimi <input type="text"/>	Käyttäjä nimi: <input type="text"/>
Kaupunki <input type="text"/>	Salasana: <input type="text"/>
Osoite <input type="text"/>	Salasana uudelleen: <input type="text"/>

Kuva 23. Rekisteröintisivu.

9.10 Haku

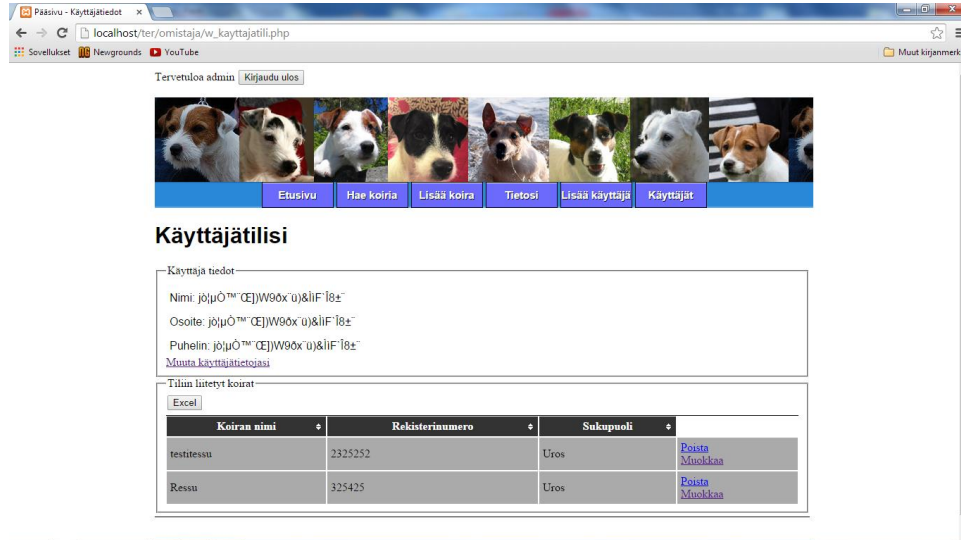
Koiran tapahtuu hakusivulla (kuva 24). Haku toimii koiran nimen, rekisterinumeron, sukupuolen ja sairauksien perusteella. Hakutoiminto hakee kaikki hakuehdot täyttävät koirat, joita ei ole asetettu yksityiseksi.



Kuva 24. Hakusivu.

9.11 Käyttäjätili

Käyttäjillä on oma hallintasivu (kuva 25). Käyttäjä voi muuntaa omia tietoja ja vaihtaa salasanaa. Sivulta löytyy myös listat kaikista tiliin liitetyistä koirista.



Kuva 25. Käyttäjätilisivu.

Koiran poisto tapahtuu yksinkertaisesti klikkaamalla käyttäjätili näkymässä koiran kohdalla olevaa **Poista**-painiketta. Ainoastaan omistaja ja hallinnoija voi poistaa koiria.

Käyttäjä voi selata omia tietoja ja vaihtaa tilin perustietoja ja salasanan.

9.12 Koiranlisäys

Tällä sivulla käyttäjä voi lisätä koiria tietokantaan. Sivun (Kuva 26) kysyy ensimmäisenä koiran nimeä, rekisterinumeroa, syntymäaika ja sukupuolelta. Kuvan koirasta voi myös laittaa mutta se on vapaaehtoinen. Lopuksi käyttäjä klikkaa **Lisää koira** -painiketta.

Tervetuloa Tuomas

**Parsonrussellinterrieri
Terveystietokanta**

Lisää koira

<p>Koiran tiedot</p> <p>Nimi <input type="text"/></p> <p>Rekisterinumero <input type="text"/></p> <p>Syntymäaika <input type="text"/></p> <p>Kasvattaja Muu kasvattaja ▾</p> <p>Sukupuoli Uros ▾</p> <p>Kivesten määrä: <input type="text"/></p> <p>Kastroitu <input type="checkbox"/></p> <p><input type="button" value="Lisää"/></p>	<p>Koiran kuva</p> <p><input type="button" value="Selaa..."/> Ei valittua tiedostoa.</p>	<p>Viesti jalostustoimikunnalle</p> <p><input type="text"/></p> <p>Oma email: <input type="text"/></p>
---	---	---

Kuva 26. Koiran lisäys –sivu.

9.13 Koiran tietojen muokkaus

Sivulla (Kuva 27) koiran tietoja voi muokata yksityiskohtaisesti.

Kuva 27. Koiran tietojen muokkaus –sivu.

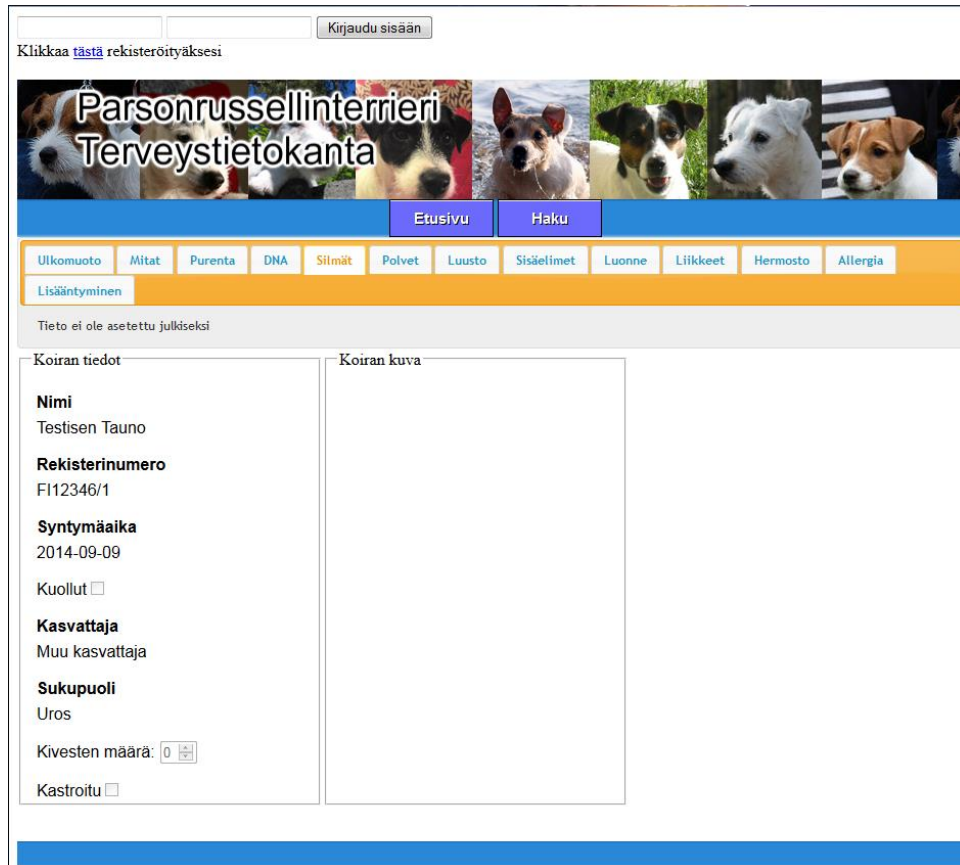
Koiran tiedot on jaettu useisiin välilehtiin, jotka on nimetty seuraavasti: mitat, hampaat, dna, polvet, käytös, liikkeet, hermosto, allergia ja lisääntyminen. Välilehden tiedot voi asettaa yksityisiksi, jolloin ne eivät näy selaajalle. Koko koiran voi asettaa yksityiseksi, jolloin se ei näy haussa.

Myös perustietoja voi muuntaa, kuten esimerkiksi nimi, rekisterinumero ja kuvan vaihto.

Sivu tarjoaa mahdollisuuden tulostaa lomakkeen käyttämällä selaimen tulostus-toimintoa.

9.14 Koiranselaus

Sivu (Kuva 28), missä selaaajat, kasvattajat ja muut koirien omistajat voivat selata ja lukea koiran tietoja.

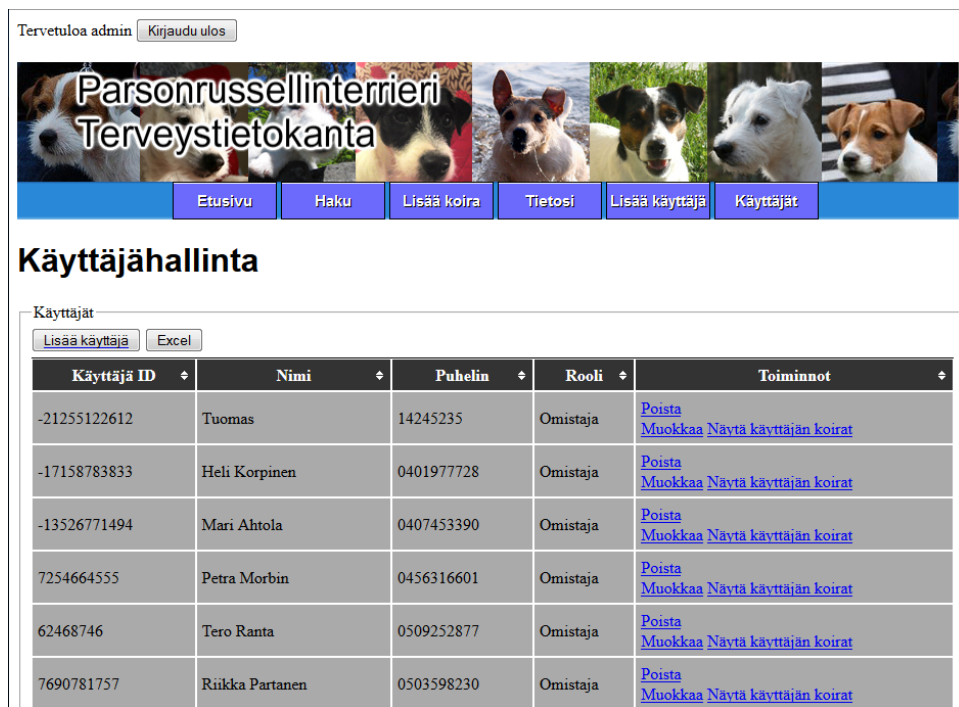


The screenshot shows a web application interface for a dog health database. At the top, there is a navigation bar with a login button 'Kirjaudu sisään' and a link 'Klikkaa tästä rekisteröityäksesi'. Below this is a banner image featuring several Parson Russell Terrier dogs with the text 'Parsonrussellinterrieri Terveystietokanta'. The main navigation menu includes buttons for 'Etusivu' and 'Haku', and a secondary menu with categories: 'Ulkomuoto', 'Mitat', 'Purenta', 'DNA', 'Silmät', 'Polvet', 'Luusto', 'Sisäelimet', 'Luonne', 'Liikkeet', 'Hermosto', and 'Allergia'. A 'Lisäntyminen' button is also present. A message states 'Tieto ei ole asetettu julkiseksi'. The main content area is divided into two columns: 'Koiran tiedot' (Dog information) and 'Koiran kuva' (Dog photo). The 'Koiran tiedot' column contains the following data: Name: Testisen Tauno; Registration number: FI12346/1; Birth date: 2014-09-09; Deceased: ; Breeder: Muu kasvattaja; Sex: Uros; Stones: 0; Castrated: .

Kuva 28. Koiran tietojen selaus -sivu

9.15 Käyttäjähallinta

Hallinnoijan työkalut (Kuva 29) käyttäjien muokkaamiseen, poistamiseen ovat samankaltaiset kuin käyttäjien työkalut koirien lisäämiseen, muokkaamiseen ja poistoon. Kun käyttäjän poistaa tietokannasta, poistetaan myös tiliin liitetyt koirat. Hallinnoija voi selata käyttäjän koirakin (Kuva 30).



Tervetuloa admin [Kirjautu ulos](#)

Parsonrussellinterrieri Terveystietokanta

[Etusivu](#) [Haku](#) [Lisää koira](#) [Tietosi](#) [Lisää käyttäjä](#) [Käyttäjät](#)

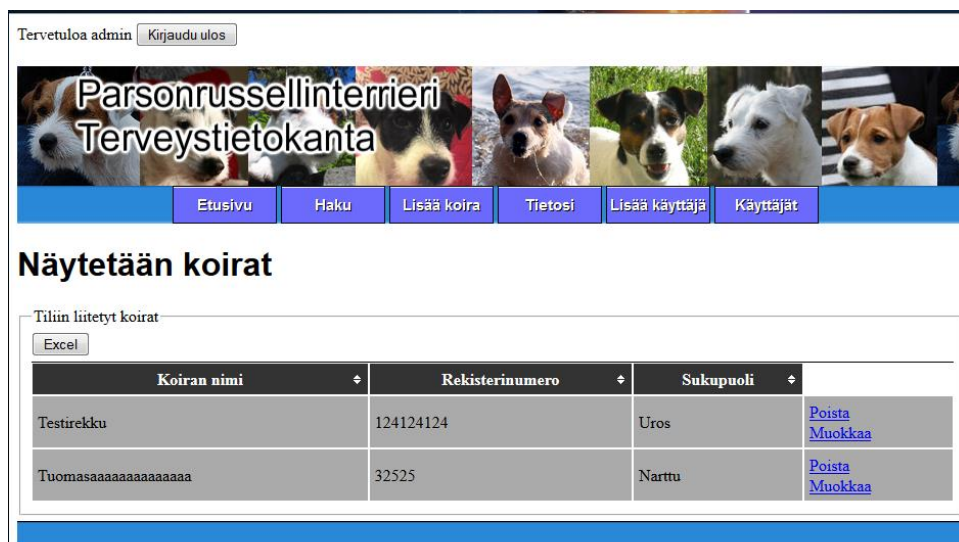
Käyttäjähallinta

Käyttäjät

[Lisää käyttäjä](#) [Excel](#)

Käyttäjä ID	Nimi	Puhelin	Rooli	Toiminnot
-21255122612	Tuomas	14245235	Omistaja	Poista Muokkaa Näytä käyttäjän koirat
-17158783833	Heli Korpinen	0401977728	Omistaja	Poista Muokkaa Näytä käyttäjän koirat
-13526771494	Mari Ahtola	0407453390	Omistaja	Poista Muokkaa Näytä käyttäjän koirat
7254664555	Petra Morbin	0456316601	Omistaja	Poista Muokkaa Näytä käyttäjän koirat
62468746	Tero Ranta	0509252877	Omistaja	Poista Muokkaa Näytä käyttäjän koirat
7690781757	Riikka Partanen	0503598230	Omistaja	Poista Muokkaa Näytä käyttäjän koirat

Kuva 29. Käyttäjähallinta-sivu



Tervetuloa admin [Kirjautu ulos](#)

Parsonrussellinterrieri Terveystietokanta

[Etusivu](#) [Haku](#) [Lisää koira](#) [Tietosi](#) [Lisää käyttäjä](#) [Käyttäjät](#)

Näytetään koirat

Tiliin liitetyt koirat

[Excel](#)

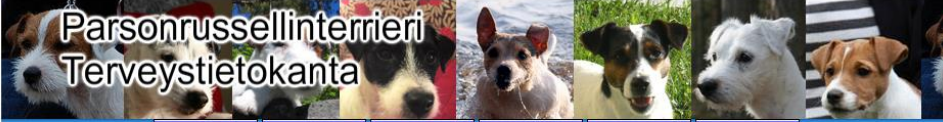
Koiran nimi	Rekisterinumero	Sukupuoli	Toiminnot
Testirekku	124124124	Uros	Poista Muokkaa
Tuomasaaaaaaaaaaaaa	32525	Narttu	Poista Muokkaa

Kuva 30. Käyttäjien koirien selaus -sivu.

9.16 Käyttäjän lisäys

Ainoastaan hallinnoija voi lisätä kasvattaja- ja jalostustoimikuntaroolin omaavia käyttäjiä, ja tämän sivun (kuva 31) käyttötarkoitus on juuri se.

Tervetuloa admin



Lisää käyttäjä

Käyttäjä tiedot	Käyttäjänimi ja salasana
Nimi <input type="text"/>	Käyttäjänimi: <input type="text"/>
Puhelin <input type="text"/>	Salasana: <input type="text"/>
Kaupunki <input type="text"/>	
Käyttäjätaso Omistaja Omistaja Kasvattaja Jalostustoimikunta	

Kuva 31. Käyttäjän lisäys -sivu.

10 Yhteenveto

Ennen opinnäytetyötä minulla oli aikaisempaa kokemusta verkkosivujen teosta, mutta en koskaan soveltanut kokemusta laajaan kuukausia kestävään projektiin. Ensimmäiset kuukaudet kuluivat sivuston ulkoasun rakentamiseen ja vasta sen jälkeen alkoi toiminnollisuuden rakentaminen.

Omasta mielestäni työ sujui hyvin. Projektin suurimmat kompastuskivet olivat tietoturva, suorituskyky ja webhotellin pystytys. Varsinkin SQL-injektio kasvatti työmäärä kahdella viikolla, mutta suorituskykyongelmat ratkesivat päivässä.

Ongelmat webhotellissa johtuivat kokemattomuudestani. Itse en ole koskaan asentanut verkkopalvelinta tai webhotellia ja sen kanssa epäröinti hidasti prosessia. Jos webhotelliasiat olisivat olleet tuttuja, webhotelli olisi ollut käyttövalmiina syyskuun alussa lokakuun alun sijaan.

Tämän opinnäytetyön aikana opin paljon verkkoselaimen tietoturvasta paljon ja webhotellin toiminnasta. JavaScriptia käytin ensimmäistä kertaa hyvin laajasti verkkosivuilla.

Asiakaan tavoite uudesta yhteisestä terveystietokannasta toteutui. Työ mahdollisesti jatkuu pienillä ylläpitopäivityksillä.

Kuvaluettelo

Kuva 1. Vesiputousmalli. (Hiltunen 2004.)	12
Kuva 2. Scrum-malli (Lucero 2013.)	13
Kuva 3. Käyttötapauskaavio.	14
Kuva 4. Sekvenssikaavio. (Oliomallinnusten perusteet, 2015.)	15
Kuva 5. Luokkakaavio. (Niksula, 2015.)	16
Kuva 6. Komponenttikaavio. (Hämeen ammattikorkeakoulu, 2015.)	17
Kuva 7. Arkkitehtuurikaavio. (SoberIT, 2015.)	18
Kuva 8. Sijoittelukaavio (Wambler, 2015.)	19
Kuva 9. Verkon rakenne. (Valtionvarainministeriö, 2009.)	27
Kuva 10. Web-palvelun arkkitehtuuri	29
Kuva 11. PhpMyAdmin.	35
Kuva 12. XAMPP-hallintapaneeli.	37
Kuva 13. Notepad++:n ohjelmaikkuna.	38
Kuva 14. FileZilla:n ohjelmaikkuna.	39
Kuva 15. Organisaation rakenne.	44
Kuva 16. OVH:lta tilattiin webhotelli.	46
Kuva 17. PhpMyAdmin	47
Kuva 18. Verkkosivun rakenne.	51
Kuva 19. Käyttötapauskaavio.	53
Kuva 20. Tietokantakaavio.	54
Kuva 21. Käytön helpottamista.	57
Kuva 22. Etusivu.	58
Kuva 23. Rekisteröintisivu.	59
Kuva 24. Hakusivu.	60
Kuva 25. Käyttäjätillisivu.	61
Kuva 26. Koiran lisäys –sivu.	62
Kuva 27. Koiran tietojen muokkaus –sivu.	63
Kuva 28. Koiran tietojen selaus -sivu	64
Kuva 29. Käyttäjähallinta-sivu	65
Kuva 30. Käyttäjien koirien selaus -sivu.	65
Kuva 31. Käyttäjän lisäys -sivu.	66

Lähteet

Haikala, I. & Mikkonen, T. 2011. Ohjelmistotuotannon käytännöt. Talentum: Helsinki.

Hakala, M. & Vainio, M. & Vuorinen, O. 2006. Tietoturvallisuuden käsikirja. Docendo: Jyväskylä.

Hiltunen, M. 2004. Vesiputouksen malli. Oulun seudun ammattiopisto, Kaukuvainion liiketalouden yksikkö. http://www.okol.org/verkkokurssit/datanomi/tietojarjestelmien_kaytto_ja_kehittaminen/johdatus_tietojarjestelmiin/kehittamistyön_vaiheet_ja_elikaarimallit/kehittamistyön_vaiheet_ja_elinkaarimallit_asia.htm. . Luettu 15.4.2015

jQuery Foundation 2014. jQuery. <http://api.jquery.com/>.

Keränen, V., Lamberg, N & Penttinen, J. 2006. Web-julkaiseminen ja multimedia. Docendo: Jyväskylä.

Komponenttikaavio -kuva. Hämeen ammattikorkeakoulu. 2015. <http://tari-nani.net/osma/Diat/UML-kalvot/sld025.htm>. . Luettu 15.4.2015

Kuutti, W. 2003. Käytettävyys, suunnittelu ja arviointi. Talentum oyj: Helsinki.

Lucero M. 2013. Coaching Agile Teams. Scrum. <http://mariolucero.cl/agile-points/what-is-scrum/>. Luettu 15.4.2015

Nielsen, J. 2000. WWW-suunnittelu. Oy Edita Ab / IT Press: Jyväskylä.

Nikander, P. 2015. <http://www.niksula.hut.fi/~mkomu/drinkit/kaavio.html>. . Luettu 15.4.2015

Oliomallinnuksenperusteet. 2014. Oulun seudun ammattiopisto. http://www.okol.org/verkkokurssit/datanomi/tietojarjestelmien_kaytto_ja_kehittaminen/johdatus_tietojarjestelmiin/oliomallinnuksen_perusteet/oliomallinnuksen_perusteet_asia.htm. . Luettu 15.4.2015

Rantanen, J., Sainio, A., Laiho, M., Renkonen, E. & Silpiö, K.1989. Relaatiotietokannat. ATK-instituutti: Helsinki.

Rossi, M. 2001. Tietokantapalvelin. Helsingin kauppakorkeakoulu http://www.cs.jyu.fi/el/tjt154_00/Matti/Jyv%E4skyl%E421022000/sld019.htm. Luettu 10.4.2015

Sijoittelukaavio -kuva. Scott Wambler, S. 2015. Agile Modeling. <http://www.agile-modeling.com/artifacts/deploymentDiagram.htm>. Luettu 15.4.2015

Treehouse. 2012. 10 käyttöliittymän suunnittelunsääntöä. <http://blog.teamtreehouse.com/10-user-interface-design-fundamentals>. Luettu 15.4.2015

Sinkkonen, I., Nuutila, E. & Törmä, S. 2009. Helppokäyttöisen verkkopalvelun suunnittelu. Tietosanoma Oy: Helsinki.

SoberIT. 2015. <http://www.soberit.hut.fi/tik-76.115/99-00/palautukset/groups/TelaMiitti/lu/tekninen.html> . Luettu 14.4.2015

Suomenvalkoinenpaimenkoira. 2014. Mallisivu. <http://www.asiakas-tila.fi/VALTE/index.php?s=hae>.

Valtionvarainministeriö 2015. Verkkon rakenne. <https://www.vah-tiohje.fi/web/guest/tekninen-internet-infrastrukturi-ja-tietoturvallisuus>. Luettu 17.4.2015

W3Schools 2014. SQL, PHP, HTML, Javascript. <http://www.w3schools.com/>. Luettu 15.1.2015