

Tampereen ammattikorkeakoulu  
Tietojenkäsittelyn koulutusohjelma  
Joni Kangaskokko

Opinnäytetyö

DYNAAMISEN VERKKOSOVELLUKSEN SUUNNITTELU  
JA TOTEUTUS

Työn ohjaaja  
Työn teettäjä  
Tampere 3/2009

Petri Heliniemi  
Kodinkone Kangaskokko Ay

Tekijä	Joni Kangaskokko
Työn nimi	Dynaamisen verkkosovelluksen suunnittelu ja toteutus
Sivumäärä	44 + 5
Valmistumisaika	3.3.2009
Työn ohjaaja	Petri Heliniemi
Työn tilaaja	Kodinkone Kangaskokko Ay

---

## Tiivistelmä

Tämän työn tarkoituksena on suunnitella ja toteuttaa yritykselle täysin uudet nykyaikaiset tietokantapohjaiset verkkosivut. Yrityksen aiemmat verkkosivut olivat erittäin raskaat ja hankalasti päivitettävät, jonka myötä niiden ylläpito oli erittäin työlästä. Uusien sivujen päätavoite on yksinkertainen ja nopea päivittäminen. Verkkosovelluksesta tehdään sellainen, että yrityksen kaikki työntekijät pystyvät ylläpitämään niitä.

Tietokantapohjainen rakenne mahdollistaa sen, että kaikki sivuston ylläpito hoidetaan suoraan selaimelta. Yrityksen työntekijät saavat tunnukset ja salasanat, joilla he pääsevät ylläpidon sivuille lisäämään, poistamaan tai muokkaamaan tuotteita. Lisäksi sivujen aukioloaikoja ja etusivun tekstejä pystytään muokkaamaan ylläpitosivuilla. Sivuston ylläpito ei ole laitteistosidonnainen, vaan päivittäminen voidaan suorittaa miltä tahansa internetiin liitetyltä tietokoneelta, jossa on WWW-selain. Yksinkertaisella päivittämisellä taataan se, että tuotteet ja niiden tiedot ovat ajan tasalla ja asiakas saa oikeaa tietoa. Sivustolta asiakkaat voivat tutkia vaivattomasti yrityksen tuotteita, palveluita ja hintoja, mikä puolestaan nopeuttaa ostoprosessia.

---

Keywords

website design, MySQL, PHP, XHTML



Tiivistelmä .....	2
1 Johdanto .....	5
2 Projektin suunnittelu .....	6
2.1 Projektin määrittely .....	7
2.2 Sivuston rakenteen kehittäminen .....	8
2.3 Visuaalinen suunnittelu ja testaus .....	9
2.4 Tuotanto ja laadunvarmistus .....	10
2.5 Julkistaminen ja jälkitoimet .....	11
3 Toteutuksessa käytetyt menetelmät.....	14
3.1 HTML & XHTML .....	17
3.2 Validointi .....	18
3.3 CSS-tyylit.....	18
3.4 Palvelin.....	19
3.5 FTP.....	20
3.6 MySQL.....	21
3.6.1 MySQL:n hyödyt .....	21
3.6.2 Avoimen lähdekoodin tietokantojen edut .....	22
3.6.3 Tietokantajärjestelmän valinta .....	23
3.6.4 Tietokantajärjestelmän tarveanalyysi.....	24
3.6.5 PhpMyAdmin.....	24
3.6.6 Käyttöoikeudet .....	26
3.6.7 Tietokannan suunnittelu.....	27
3.6.8 Tietokannan toteutus .....	28
3.6.9 Tietokannan tarkastaminen .....	30
3.7 PHP .....	31
3.7.1 PHP:n edut .....	33
3.7.2 PHP:n käyttö .....	34
4 PHP ja MySQL .....	35
5 Sivuston rakenne .....	38
5.1 Yleisarkkitehtuuri.....	38
5.2 Tiedostot ja niiden väliset suhteet.....	38
6 Kehittämismahdollisuudet.....	42
7 Päätelmät .....	44

# 1 Johdanto

Suoritin työharjoittelun Kodinkone Kangaskokko Ay -nimisessä yrityksessä vuonna 2007. Yrityksellä oli aiemmin verkkosivut osoitteessa: [www.kangaskokko.com](http://www.kangaskokko.com). Sivut olivat käytettävyydeltään hyvin sekavat. Päivittäminen toteutettiin erittäin monimutkaisella ja aikaa vievällä tavalla. Tämän vuoksi toimeksiantaja teki esityksen helpommin käytettävistä sivuista, joita myös muut työntekijät voisivat päivittää. Uusilta sivuilta asiakkaat voisivat tarkastaa ajankohtaisia tarjouksia ja muita hyödyllisiä tietoja.

Valmiista Internet-sivustosta on hyötyä niin yritykselle, kuin asiakkaille. Yritys voi laittaa tarjouksia ja tuoteuutuuksia sivuilleen ja näin parantaa myyntiään. Asiakkaat voivat helposti tehdä tuotevertailuja, lukea uutisia ja olla yhteydessä yritykseen sivuston kautta. Tämän lisäksi myös sivuston ylläpitäjän työmäärä pienenee, kun sivustoa voivat päivittää muutkin työntekijät.

Opinnäytetyön tavoitteena on suunnitella ja toteuttaa täysin uudet tietokantapohjaiset verkkosivut. Sivuston rakenteen tulee olla sellainen, että kaikki yrityksen työntekijät pystyvät ylläpitämään sivuja, vaikka heille ei olisikaan aiempaa kokemusta sivujen päivittämisestä. Omana tavoitteenani on kerrata ja hyödyntää oppimiani taitoja, mutta mahdollisesti myös oppia uusiakin menetelmiä.

Koska tietotekniikassa kehitys on erittäin nopeaa, on opinnäytetyössä pyritty käyttämään mahdollisimman uusia ja tunnettuja lähteitä. Opinnäytetyöhön liittyvää materiaalia oli tarjolla runsaasti, jonka vuoksi niiden valinnassa joutui käyttämään harkintaa. Suurin osa lähteistä on kirjoja, mutta joukossa on myös muutama verkkosivu ja artikkeli. Verkkosivuiksi on hyväksytty ainoastaan tunnetut sivut ja artikkeliksi sellaiset, joista löytyy julkaisua koskevat tiedot. Näin ollen opinnäytteen lähdekritiikki on pysynyt riittävänä

## 2 Projektin suunnittelu

Yleensä WWW-sivustoprojektilla on tilaaja, joka useimmiten on yritys tai yksityishenkilö. Sivustolla on jokin tarkoitus ja sitä pyritään hyödyntämään jollain tapaa, esimerkiksi lisäämään yrityksen myyntiä tai tunnettavuutta. Tilaaja tekee toimeksiannon ja kertoo verkkosivun toteuttajalle, mistä projektissa on kysymys.

Tämän projektin toimeksiantajana toimii Kodinkone Kangaskokko Ay. Tämä kodin elektroniikkaa myyvä kolme henkilöä työllistävä yritys toimii Pohjois-Pohjanmaalla Kärsämäellä. Expert-ketjuun liittymisen myötä yrityksen tuotevalikoima on laajentunut ja tarjoukset vaihtelevat eri kampanjoiden mukaan. Nämä seikat asettavatkin haasteen koskien sivuston päivitystä. Tuotteet ja niiden tiedot täytyy saada lisättyä verkkosivuille nopeasti ja helposti. Tuotetietoja täytyy myös voida muokata tai poistaa kokonaan. Projektin vaatimusten ja tavoitteiden pohjalta voidaan aloittaa varsinaisen verkkosovelluksen suunnittelu.

Suunnittelun yhteydessä on hyvä tarkastella mahdollisten kilpailijoiden sivustoja ja tutkia niiden hyviä ja huonoja puolia. Samalla olisi suotavaa miettiä, millä tavalla omasta sivustosta saataisiin parempi kuin kilpailijoilla. Projektin suunnittelu ja toteutus voidaan jakaa viiteen eri vaiheeseen:

1. Projektin määrittely.
2. Sivuston rakenteen kehittäminen.
3. Visuaalinen suunnittelu ja testaus.
4. Tuotanto ja laadunvarmistus.
5. Julkistaminen ja jälkitoimet.

## 2.1 Projektin määrittely

Kun projekti on pantu alulle, on suositeltavaa tehdä projektin toteuttamisesta selkeä hyväksyntä kirjallisesti (*Goto & Cotler 2005, 19-22*). Siitä voi olla jatkossa hyötyä, jos tilaaja esimerkiksi syystä tai toisesta luopuu projektista kesken kaiken. Tällöin tehty työ ei mene hukkaan, vaan siitä voidaan vaatia korvausta kirjallista sopimusta todisteena käyttäen.

Projektin määrittely on koko työn tärkein osa, koska siinä määritellään käytännössä koko prosessi. Määrittelystä selviää, mitä halutaan ja kenelle. Sivusto-projektilta kannattaa etsiä sen päätavoitteet asiakas- ja ylläpitoselvityksen avulla. Näillä menetelmillä saadaan selville sivuston tavoitteet ja käyttäjät. Määrittelyssä kannattaa olla tarkkana, sillä pienetkin lisäykset toteutukseen saattavat aiheuttaa ajallisesti huomattavasti ylimääräistä työtä. (*Goto & Cotler 2005, 19*)

Projektin päätavoitteet kannattaa eritellä, jotta suunnittelussa ja toteutuksessa pystytään keskittymään tärkeimpiin seikkoihin. Sivuston tavoitteisiin voi kuulua esimerkiksi helppo päivitettävyyden, myynnin edistäminen, kanta-asiakkaiden hankkiminen, tai tunnettuuden lisääminen. Tässä vaiheessa on suotavaa tutustua yrityksen markkinointisuunnitelmaan ja sen toteutustapoihin. Kun tiedetään yrityksen toimintatavat, voidaan räätälöidä sivustosta juuri heille sopiva kokonaisuus. (*Goto & Cotler 2005, 19-20*)

Projektin tilaajan lisäksi on tiedettävä tarkkaan, ketkä ovat sivuston varsinaisia käyttäjiä. Kohderyhmän tunteminen on tärkeä osa suunnittelua. Sivuston käyttäjät voidaan selvittää käyttäjätutkimuksella, joka voi olla esimerkiksi haastattelu. Kohderyhmän tuntemisen lisäksi on tiedettävä, mitä he sivustolta haluavat. Käyttäjät on hyvä jakaa omiin profiileihinsa. Näin pystytään erittelemään mahdolliset kohderyhmien oikeudet sivustolla. Käyttäjät voivat jakautua esimerkiksi ylläpitoon ja asiakkaisiin. Tämä tarkoittaa käytännössä sitä, että asiakkailta on eritasoiset oikeudet sivustolle kuin ylläpidolla. Eri profiilit vaativat tarkkaa tietoturvaa sivuston rakenteelta ja toiminnalta. (*Goto & Cotler 2005, 20*)

Kun sivuston käyttäjät tiedetään, voidaan tutkia, millaiset tekniset valmiudet heillä on. Tämä tarkoittaa tutkimusta siitä, minkä tyyppisillä sovelluksilla ja laitteilla sivustoa käytetään, ja mitä ominaisuuksia sivusto vaatii. Sivuston toteuttaminen ja testaus tulee tehdä niin, että sivusto toimii ainakin yleisimmissä WWW-selaimissa. Sovellus voi esimerkiksi vaatia WWW-selaimelta Java-liitännäistä, jotta kaikki halutut toiminnot saataisiin näkyviin halutulla tavalla. (*Goto & Cotler 2005, 20*)

Projektin laajuutta ja aikataulua mietittäessä tulee kiinnittää huomiota myös budjettiin. Budjetti rajaa melko tarkasti sen, kuinka paljon aikaa voidaan käyttää esimerkiksi suunnitteluun, toteutukseen, sovelluksiin ja testaukseen. Aikataulu kannattaa suunnitella tarkasti ja sen toteutumista tulee seurata tarkasti, jotta sivuston aikarajat eivät tule vastaan liian nopeasti. Pienikin lisäys sivustoon saattaa aiheuttaa ajallisesti todella paljon töitä. (*Goto & Cotler 2005, 21*)

Kun kaikki aiemmin mainitut seikat on saatu valmiiksi, voidaan kerätä projektin toteutettava ryhmä. Ryhmä voi koostua esimerkiksi projektipäälliköstä, suunnittelijoista, ohjelmoijista, graafisista suunnittelijoista ja testaajista. Kun sopiva tiimi on saatu kasattua, on hyvä pitää aloituskokous, jossa ilmenee yhteiset tavoitteet ja toimintatavat. Alusta asti on erittäin tärkeää, että tiimi toimii jatkuvassa yhteistyössä, jotta projekti etenee suunnitellusti ja sivustosta saadaan yhtenäinen kokonaisuus.

(*Goto & Cotler 2005, 21*)

## **2.2 Sivuston rakenteen kehittäminen**

Sivuston rakenteen kehittämisessä aloitetaan varsinainen käytännön työ. Tähän vaiheeseen kuuluu sisältö- ja tietostrategia. Näin määritellään, kuinka asiat esitetään, ja kuinka käyttäjät löytävät haluamansa tiedon sivustolta. Oli kyseessä miten suuri tai pieni sivusto tahansa, tulee sovelluksen rakenteen olla looginen. Sivuston runkoa ei kannata ryhtyä toteuttamaan ennen kuin sivuston sisältö on tarkkaan mietitty. Sivuston sisällön toimitamisesta kannattaa sopia ajoissa sen toimittajan kanssa, ja siitä kannattaa laatia aikatau-



lu. Näin vältetään sivuston valmistumisen pitkittymiseltä. Lopullisen sivuston rungon määrittää sisältö ja se, miten asiat halutaan esittää. (Goto & Cotler 2005, 23)

Sivustosta täytyy saada luotua selkeä näkemys. Rakennetta kannattaa suunnitella yhdessä toimeksiantajan kanssa. Myös muut projektiin osallistujat voivat osallistua tähän vaiheeseen, jotta tärkeimmät asiat saataisiin esille. Kokouksessa kannattaa tehdä muistiinpanoja suunnitelmista ja sovitusta asioista. Kun sivuston sisältöä ja rakennetta suunnitellaan, kannattaa miettiä, mitä käyttäjät haluavat ja odottavat näkevänsä sivustolla. (Goto & Cotler 2005, 24.)

Suunnittelussa yksi erittäin hyödyllinen menetelmä on sivustokartta. Sen avulla pystytään rakentamaan selkeä näkemys WWW-sovelluksesta. Siitä selviävät sivujen linkitykset ja navigaatio. Navigaatiolla tarkoitetaan sivuilla liikkumista. Sen avulla käyttäjä liitetään sivuston sisältöön. Sivustokarttaan on suositeltavaa nimetä tiedostot tai sivut yhtenevällä tavalla. Myös kuvat ja linkit tulee nimetä ja toteuttaa samalla tyylillä, jotta sivustosta muodostuu kiinteä kokonaisuus. (Goto & Cotler 2005, 24)

### **2.3 Visuaalinen suunnittelu ja testaus**

Visuaalisella suunnittelulla luodaan niin sanottu graafinen käyttöliittymä, joka määrittää sen, miltä sivut tuntuvat ja näyttävät. Sivuston tulisi suositusten mukaan olla vuorovaihteinen, yleispätevä ja sen tulisi täyttää kohderyhmän tarpeet. (Goto & Cotler 2005, 25)

Tässä projektissa sivuston ulkoasulta vaadittiin yhteneväisyyttä Expert-ketjun yleiseen ulkoasuun. Lisäksi ulkoasusta toivottiin yksinkertaista ja selkeää, jotta halutut tiedot löytyisivät sivuilta mahdollisimman nopeasti. Vaatimukseen kuuluu myös se, että etusivun ulkoasua täytyy voida muokata tarvittaessa meneillään olevan Expert-kampanjan teeman mukaiseksi.

Ulkoasulla luodaan sivustolle tunnelma. Sivuston ulkoasun tulisi olla sellainen, että kohderyhmä löytää haluamansa tiedon sieltä helposti ja pystyy liikkumaan yksinkertaisesti kaikilla heille tarkoitetuilla sivuilla. Ulkoasun pitäisi myös sopia yhteen yrityksen imagon kanssa. Monesti on erittäin haastavaa saada tasapaino sivuston ulkoasun ja käytettävyyden välille. Visualisten suunnitelmien kokeiluversiot tulisi näyttää tuotantotiimille, jotta saadaan selvyys siitä, voiko sivuja ylipäättään toteuttaa halutulla tavalla. (*Goto & Cotler 2005, 26*)

Kun muutama konsepti ulkoasusta on saatu tehtyä, voidaan ne esittää toimeksiantajalle. Kun yhteisymmärrys tulevasta visuaalisesta suunnitelmasta ja toiminnallisuuksista on saavutettu, on niistä suotavaa kirjoittaa kirjallinen sopimus. Sopimuksessa voidaan myös eritellä tarkemmin esimerkiksi aikataulusta ja materiaalien toimittamisesta sivuston tekijöille. Sopimuksessa tulisi mainita, kenelle kuuluu sivuston tekijänoikeudet. Mikäli sivustolla on esimerkiksi kuvia tai videoita, tulee pitää huoli, että niitä koskevat tekijänoikeudet ovat kunnossa. (*Goto & Cotler 2005, 26*)

Ulkoasun suunnittelussa kannattaa tehdä niin sanottu tyyliopas, josta ilmenee sivuston ulkonäön kannalta olennaiset seikat. Tyylioppaaseen voi merkitä esimerkiksi värit, reunukset, kirjasimien tyypit, koot värit ja korostukset. Tyyliopas annetaan kaikille, jotka mahdollisesti vaikuttavat sivuston ulkoasuun. Tällä tavalla ulkoasusta saadaan yhtenäinen kaikilta osin. (*Goto & Cotler 2005, 27*)

Sovitun ulkoasuratkaisun pohjalta jatketaan sivuston rakentamista. Tähän vaiheeseen kuuluu navigaation ja toiminnallisuuksien testausta. Sivustosta voidaan luoda yksinkertainen kevyt malli, jossa voidaan testata sivuilla liikkumista. (*Goto & Cotler 2005, 26*)

## **2.4 Tuotanto ja laadunvarmistus**

Tuotannossa sisältö ja ulkoasu yhdistetään yhtenäiseksi sivustoksi. Tähän vaiheeseen kuuluu näkymätön ohjelmointi, eli toteutetaan sivuston taustalla olevat koodit, jotka ei-

vät näy sellaisenaan käyttäjälle. Mahdollinen ohjelmointi liitetään varsinaiseen sivustoon tässä vaiheessa. (*Goto & Cotler 2005, 27*)

Tuotantoon ja laadunvarmistukseen kuuluu erittäin tärkeänä osana kokonaisuuden testaus. Kaikki toiminnallisuudet tulee käydä läpi ja mahdolliset viat sekä toimintahäiriöt tulee saada selville. Ongelmien etsinnässä kannattaa hyödyntää käyttäjiltä saatavaa palautetta. Heiltä kannattaa hankkia tietoa myös käytettävyydestä ja siitä, kuinka helposti haluttu tieto löytyy. (*Goto & Cotler 2005, 27*)

Tiedostorakenteen parantelu, standardien noudattaminen ja sivuston yhtenäisyys tulee tarkastaa testauksen yhteydessä. Näillä pyritään helpottamaan sivuston ylläpitämistä ja mahdollista kehittämistä. Sivuston kevyellä rakenteella taataan nopeat sivujen latautukset ja sivuston päivitykset. Ulkoasu vaikuttaa yleensä merkittävästi sivujen latausnopeuksiin. Tässä vaiheessa projektia, tulisi myös tilatun sisällön olla valmiina. Yleensä sisällön toimittaminen viivästyy, tästä syystä kannattaa olla yhteydessä sen toimittajaan ajoissa. Näin pyritään pitämään aikataulu ja budjetti suunnitelman mukaisina.

Kun sivusto on rakennettu ja testattu, voidaan tarkastaa, onko alussa määritetyt tavoitteet täytetty. Tarkastuksessa selvitetään, ovatko käytetty aika, budjetti ja itse sivusto yhdenmukaiset alkuperäisen suunnitelman kanssa. Tarvittaessa tässä vaiheessa hankitaan lisää resursseja, jotta työ saadaan toteutettua alkuperäisen suunnitelman mukaisesti.

Kun sivusto alkaa olla valmis, testataan sivustokokonaisuus yksityiskohtaisesti, ennen kuin se siirretään julkiselle palvelimelle. Seuraava askel on sovelluksen esittäminen tilaajalle. Mikäli sivusto todetaan yhteismielin valmiiksi, pyydetään asiakkaalta sitä koskien allekirjoitus. (*Goto & Cotler 2005, 29*)

## **2.5 Julkistaminen ja jälkitoimet**

Julkistamisen ohessa huomataan yleensä viimeisimmät ongelmat. Ongelmien varalle kannattaakin tehdä varasuunnitelma. Ennen julkaisua virheet tulisi korjata, tyyliopas

viimeistellä ja lopullinen sovellus kaikkineen dokumentteineen tulisi luovuttaa tilaajalle. Dokumenttipakettiin tulee liittää kaikki projektin aikana tehdyt sopimukset, tarjoukset, hyväksynät, palaverimuistiot ja suunnitteludokumentit. Lisäksi siihen tulisi liittää käyttöä, ylläpitoa ja päivitystä koskevat ohjeet. Kilpailijoiden sivuista on myös suotavaa tehdä analyysi, jotta voidaan vertailla projektisivuston ja kilpailijoiden sivuston ominaisuuksia ja hyviä puolia. (*Goto & Cotler 2005, 30*)

Kun sivuston toteutus on täysin valmis, voidaan siirtyä sen julkistamiseen, ja sen vaatimiin jälkitoimiin. Julkistamisella tarkoitetaan sivuston siirtämistä julkiselle palvelimelle. Suuremmilla yrityksillä saattaa olla oma palvelin ja ylläpidosta vastaava henkilö tai ryhmä. Pienyrityksillä ei ole yleensä tarvetta omalle palvelimelle, jolloin on suotavaa kääntyä jonkin webhotellin puoleen. Webhotellit ovat Internet-verkkoon kiinteästi kytettyjä palvelimia, jotka huolehtivat kotisivujen näyttämisestä kotisivuilla vieraileville asiakkaille. Tässä projektissa verkkosivut siirrettiin kotisivut.com-nimiselle palvelimelle, jossa on PHP- ja MySQL-tuki. Verkkotunnukseksi valittiin [www.kangaskokko.com](http://www.kangaskokko.com). (*Goto & Cotler 2005, 31*)

Kun sivut siirretään julkiselle palvelimelle, kaikilla Internetin käyttäjillä on mahdollisuus vierailla sivuilla. Mikäli sivuston otsikkotunnisteet ja avainsanat on lisätty oikein, löytyy sivusto myös useilla hakukoneilla. Kodinkone Kangaskokon sivut löytyvät hakukoneella esimerkiksi kangaskokko-sanalla. Sivuston toimeksiantajalla on yleensä tiedotussuunnitelma WWW-sivun julkiseksi tekemisestä. Tässä projektissa se tarkoittaa käytännössä verkko-osoitteen liittämistä sanomalehtimainoksiin.

Tuotannosta vastaava henkilö tai ryhmä luovuttaa valmiin sovelluksen ylläpidolle. Luovutuksen ohessa täytyy pitää huoli ylläpidon koulutuksesta, jotta he oppivat käyttämään ja ylläpitämään sivustoa oikealla tavalla. Sivuston päivitystä koskeva suunnitelma voi auttaa sivuston ylläpidossa. Siitä selviävät päivityksen ajankohdat ja tulevat muutokset. Ylläpitotiimin arviointi kuuluu tuotannolle. Heidän tehtävänä on selvittää, riittääkö henkilöiden määrä ja taidot ylläpitämään sivustoa. (*Goto & Cotler 2005, 30-32*)

Jatkuvaa kehittymistä ja oppimista silmällä pitäen projektista vastanneen ryhmän on suositeltavaa pitää loppukokous. Kokouksessa tulisi analysoida kokonaisuutena projektin läpivientiä miettien samalla, mitkä asiat onnistuivat, ja missä on vielä parannettavaa. (*Goto & Cotler 2005, 30-32*)

Kun sivustolle on saatu käyttäjiä, voidaan tutkia, onko projektista ollut käytännön hyötyä yritykselle. Onnistumista voidaan mitata esimerkiksi huolto- ja tukisottojen vähenemisellä, myynnin kasvulla tai ostopäätösten nopeutumisella. Tämän projektin webhottelilla on palvelu, josta pystytään seuraamaan sivuston kävijämääriä. Tämä auttaa arvioimaan sivuston hyödyllisyyttä.

### 3 Toteutuksessa käytetyt menetelmät

Tietokonesovellukset rakennetaan erinäisistä koodeista. Yleensä kaupallisilla sovelluksilla koodit ovat salattuja, eli niiden lähdekoodia kutsutaan suljetuksi. Suljettu lähdekoodi on jonkun omaisuutta, eivätkä siihen pääse käsiksi muut kuin henkilöt, joille on myönnetty tarvittavat oikeudet.

Avoimen lähdekoodin sovelluksessa koodi on nimensä mukaan avoin. Tällaiset sovellukset ovat yleensä ilmaisia, eikä niiden lähdekoodien käyttöön tarvita erillisiä oikeuksia. Koodi on avoin kaikille ja sitä saa kopioida, kehittää, korjata, levittää ja käyttää täysin vapaasti. Kodinkone Kangaskokon verkkosivu projektissa päädyttiin käyttämään avoimen lähdekoodin ohjelmia niistä aiheutuvien kustannusten ja positiivisten kokemusten perusteella.

Kaupallinen sovellus on tehty niin sanotusti valmiiksi tuotteeksi ja laitettu myyntiin. Tämä ei kuitenkaan takaa ohjelman virheettömyyttä. Yleensä kestää todella kauan, kunnes ohjelmaan on saatavilla uusi päivitys, joka kaupallisessa sovelluksessa on yleensä maksullinen.

Avoimen lähdekoodin sovelluksia parannellaan jatkuvasti kehittäjäyhteisöissä, ja näin ollen uusia päivityksiäkin on saatavissa enemmän. Nämä ohjelmat eivät tavallaan ole koskaan valmiita, vaan niitä kehitellään jatkuvasti eteenpäin.

Tässä projektissa käytetyt PHP ja MySQL perustuvat avoimeen lähdekoodin. Niiden valintaan päädyttiin hinnan, helppokäyttöisyyden ja projektin vaatimuksien perusteella. Laajempien ja monipuolisempien tietokannan hallintajärjestelmien käyttöä ei harkittu, koska projektin tietokanta on suppea ja sen käyttö on melko pientä.

Aikaisemmin uskottiin, ettei ilmainen avoimeen lähdekoodiin perustuva sovellus voi menestyä kaupallisen sovelluksen tavoin (Rosen 2004, 10-30). GNU-käyttöjärjestelmän (GNU's Not Unix) ja Linuxin valmistumisen jälkeen yleinen mielipide muuttui. IBM:n

ja HP:n investoitua miljoonia dollareita avoimeen lähdekoodiin perustuviin ohjelmiin, oli sovelluskehityksessä tapahtunut valtava suunnanmuutos. Muutamia tunnettuja vapaan lähdekoodin sovelluksia ovat esimerkiksi OpenOffice, Mozilla Firefox ja PhpMyAdmin.

Vuoteen 2000 mennessä sourceforge.net piti sisällään 17 000 avoimen lähdekoodin projektia. Vuonna 2004 niitä oli kertynyt jopa 74 000 kappaletta. Vuonna 2008 niitä oli jo yli 180 000. Kasvu on ollut erittäin nopeaa ja se jatkuu edelleen hurjaa vauhtia. (Rosen 2004, 10-30)

Jotta sovellukselle voidaan myöntää avoimen lähdekoodin lisenssi, tulee sen täyttää OSI:n (*Open Source Initiative*) kymmenen vaatimusta. Useat organisaatiot ja avoimen lähdekoodin yhteisöt hyväksyvät ainoastaan ohjelmat, joilla on kyseinen lisenssi.

Avoimen lähdekoodin määrittely koostuu L. Rosenin (2004, 10-30) mukaan kymmenestä osasta:

### **1. Vapaa levitys**

*Lisenssiin ei saa liittyä mitään myymiseen liittyvää, eikä luovuttamista osaksi joltain suurempaa sovellusta. Lisenssillä ei saa vaatia mitään maksua.*

### **2. Lähdekoodi**

*Sovellukseen tulee sisältyä lähdekoodi muokattavassa muodossa. Jos lähdekoodi ei ole liitettyä sovellukseen, tulee sen olla saatavilla ilmaiseksi Internetistä. Koodin tulee olla muokattavassa muodossa ja sen tahallinen sotkeminen on kiellettyä.*

### **3. Jatkotyöstäminen**

*Lisenssin täytyy antaa oikeudet muutoksiin ja uudelleen levitykseen alkuperäisten sovelluksen lisenssin ehtojen mukaisesti.*

**4. Oikeudenmukaisuus tekijän lähdekoodiin**

*Lisenssi voi rajoittaa muokatun lähdekoodin levittämistä ainoastaan, jos lisenssi antaa oikeudet korjaustiedostojen (patch files) levittämiseen, sovelluksen rakentamisen aikana. Lisenssi voi vaatia kehitysversiolta eri nimeä tai ohjelmaversion numeroa kuin alkuperäinen ohjelma.*

**5. Ei henkilöiden tai ryhmien syrjintää**

*Lisenssi ei saa syrjiä ketään henkilöä tai ryhmää.*

**6. Ei käyttöympäristöjen syrjintää**

*Sovellusta tulee saada käyttää missä tahansa ympäristössä. Sen käyttöoikeudet eivät voi rajoittua esimerkiksi yhdelle ammattiryhmälle.*

**7. Lisenssin levittäminen**

*Ohjelmaan liitettyjä oikeuksia täytyy soveltaa samalla tavalla kaikkiin sen käyttäjiin.*

**8. Lisenssiä ei saa määritellä tuotteelle**

*Vaikka alkuperäinen laajempi sovelluskokonaisuus jaettaisiin pienempiin osiin omiksi ohjelmiksi, koskevat alkuperäisen sovelluksen oikeudet kaikkia näitä osia. Mikäli näitä osia levitetään, tulee niihin liittää mukaan alkuperäisen sovelluksen oikeudet.*

**9. Lisenssi ei saa rajoittaa muita sovelluksia**

*Lisenssi ei saa sisältää rajoitteita toisille ohjelmille, joita levitetään lisensoidun ohjelman mukana.*

**10. Lisenssin tulee olla teknisesti neutraali**

*Lisenssin sisältö ei saa riippua tietystä tekniikasta, tyylistä tai käyttöliittymästä.*



### 3.1 HTML & XHTML

HTML on lyhenne sanoista (Hypertext Markup Language) (Oliver 2002, 1-2 ). HTML on merkkaukieli. HTML perustui alun perin SGML:n (Standard Generalized Markup Language) syntaksiin. HTML-kielillä kuvataan, kuinka WWW-sivun eri elementit näkyvät käyttäjälle, ja kuinka sivut ovat linkitettyinä toisiinsa. Sivut sisältävät yleensä tekstiä ja kuvia, mutta toisinaan myös videoita ja ääntä. Kun useita sivustoja on linkitettyinä toisiinsa, muodostuu verkosto. Nykyään verkkosivut ovat toteutettu pääsääntöisesti HTML tai XHTML-kielillä.

XML tulee sanoista (eXtensible Markup Language). Se on laajennettava metakieli. XML pohjautuu HTML:n tavoin SGML syntaksiin. Metakieli tarkoittaa kieltä, jolla kuvataan kieltä. Erona HTML-kielen on se, että HTML koostuu ennalta kielen rakenteessa määritellyistä tageista, kun taas XML mahdollistaa omien tagien käytön. XML-kieli antaa runsaasti uusia mahdollisuuksia WWW-sovelluksen monipuolisempaan toteuttamiseen. (Oliver 2002, 9-24)

XML koostuu silmuista, joihin voidaan tallentamaan tietoa. Silmulista voi sisältää esimerkiksi sähköpostiin tarvittavat tiedot. Silmulistan eri kohdilta voidaan hakea tietoa JavaScript-koodin avulla. Näin alkuperäinen tieto saadaan näkyviin WWW-selaimessa. HTML ja XML eivät ole keskenään täysin yhteensopivia. Tämän vuoksi kehitettiin tiukemmat muutosäännöt omaava XHTML-kieli, joka on XML-pohjainen ja sen kanssa yhteensopiva. (Oliver 2002, 9-24)

XHTML:n tärkeimmät säännöt ovat:

- tagit kirjoitetaan pienillä kirjaimilla
- sisällöttömät tagit sisältävät kauttaviivan (<br />)
- kaikilla tageilla on lopputagit
- kaikilla attribuuteilla tulee olla arvo ja niiden tulee olla lainausmerkeissä.

- tiettyjä erikoismerkkejä kuten < ja & sekä ” tai ’ ja vastaavat loppumerkitkin, eli > ja ; ei saa käyttää, koska ne voidaan tulkita tagin osiksi, vaan ne tulisi vaikeuksien välttämiseksi määritellä merkkientiteetteinä: <!ENTITY gt "&#54;">

## 3.2 Validointi

Validointi (validation) on menetelmä, jolla tarkastetaan HTML- ja XHTML-kielen rakenne oikein muodostetuksi ja oikeelliseksi. Validin verkkosivuston kaikki sivut täyttävät niissä käytettyjen kielten kriteerit. (W3C 2001, Modularization of XHTML, 1.3.4 Validation).

Nykyään yhä useammat sivut ovat tuotettu dynaamisesti esimerkiksi PHP:n avulla. Myös näiden dynaamisesti luotujen sivujen tulee olla koodeiltaan valideja. Sivut voidaan tarkastaa W3C:n validaattori-ohjelmalla verkko-osoitteen, tiedostopolun, tai suoraan koodisyötön avulla. Palvelu ilmoittaa virheen tyyppin ja rivin, jolla ongelma ilmenee. Tämän projektin kaikki sivut on tarkastettu W3C:n validaattorilla. PHP-koodi osaa tarkastaa itse itsensä. Mikäli PHP-koodissa ilmenee ongelmia, ilmoittaa WWW-selain sivua avattaessa, mistä ongelmasta on kyse. (W3C 2001, Modularization of XHTML, 1.3.4 Validation).

Tarkastamalla sivut voidaan varmistua siitä, että sivustot ovat nykyaikaiset ja niiden kehittäminen ja päivittäminen tulee jatkossa olemaan vaivattomampaa. Tarkastettujen sivujen yhteensopivuus eri ohjelmointikielten ja selainten kanssa on varmempaa. (W3C 2001, Modularization of XHTML, 1.3.4 Validation).

## 3.3 CSS-tyylit

CSS tulee englannin kielen sanoista Cascading Style Sheets, eli kaskadisetyylitiedosto. CSS:n avulla voidaan luoda sivuille ulkoasusta määräävä tyylitiedosto. Tyylitiedostoon

voidaan määrittää lähes kaikkien sivuston muotoilusäännöistä: väreistä, asemoinnista, koosta ja korostuksista. Lisäksi tyylytiedostolla voidaan määrittää esimerkiksi taustojen värejä, reunoja ja vierityspalkin ulkoasua.

Tyylytiedostoilla säästetään huomattavasti aikaa verkkosivujen rakentamisessa ja päivityksissä. Yksittäisten sivujen ulkoasua ei tarvitse muuttaa, vaan yhtä tyylytiedostoa muokkaamalla saadaan koko sivuston ulkoasu määritettyä samalla kertaa. Halutut sivut linkitetään tyylytiedostoon, jotta määrittäykset saadaan voimaan. Tyylytiedosto luo sivustolle tavallaan tyylistandardin, jota jokainen yksittäinen sivu noudattaa. Näin WWW-sovelluksesta saadaan ulkoasultaan kiinteä yhtenäinen kokonaisuus. (Oliver 2002, 231-246)

### **3.4 Palvelin**

Palvelin on verkkoon kiinnitetty tietokone ja ohjelma, joka vastaanottaa tietoja asiakasohjelmilta ja vastaa niiden tietopyyntöihin. WWW-palvelin jakaa dokumentteja http-protokollalla tietokoneille ja selaimille. Jakaminen tapahtuu WWW-selain-ohjelman avulla. Eräitä tunnettuja selaimia ovat: Mozilla Firefox, Internet Explorer ja Opera. (Heinisuo & Rauta 2007, 12)

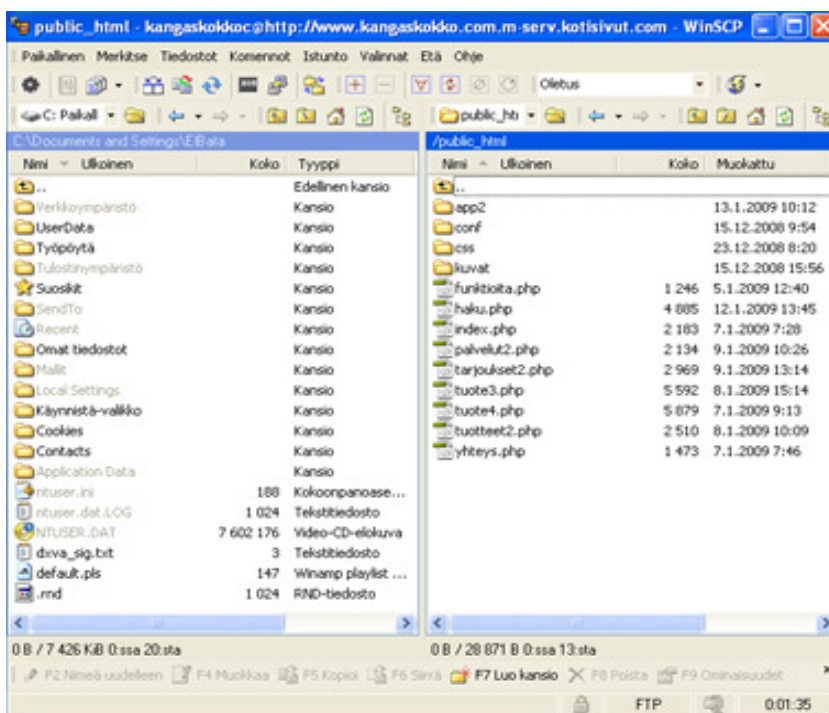
Valmiit WWW-sivut ja sovellukset voidaan siirtää WWW-palvelimelle, jolloin niistä tulee julkisia. Jos WWW-sovellus sisältää esimerkiksi MySQL-tietokannan tai PHP-toimintoja, tulee palvelimen ylläpitäjältä varmistaa, että niille on saatavilla tarvittavat tuet. Yleensä palvelimilta löytyy suoraan PHP-tuki, mutta MySQL-tuen saattaa joutua hankkimaan erikseen pientä korvausta vastaan. (Heinisuo & Rauta 2007, 12-14)

Palvelimen voi toki rakentaa itsekin vaikka kotiinsa. Tietoturvasta, päivityksistä, varmuuskopioista ja monesta muusta seikasta aiheutuu yllättävän paljon työtä ja lisäkustannuksia. Usein onkin järkevämpää turvautua kaupallisiin palveluntarjoajiin. Internetistä löytyy todella runsaasti palvelinhotelleja, joilta voi vuokrata tilaa käyttöönsä. Palve-

luntarjoajien välillä kannattaa tehdä vertailuja palvelujen ja hintojen suhteen. Yleensä myös verkkotunnus (domain) järjestyy palvelinhotellin tarjoajan kautta. Verkkotunnus on sivuston osoite, joka kirjoitetaan WWW-selaimen osoiteriville. (Heinisuo & Rauta 2007, 12-14)

### 3.5 FTP

FTP (File transfer protocol) on TCP-protokollaa (Transmission Control Protocol ) käytävä tiedostonsiirtomenetelmä. FTP-sovelluksella voidaan muodostaa etäyhteys verkkosivujen palvelimelle. Eräs hyvin suosittu FTP-ohjelma on WinSCP (kuva 1.). Sen avulla on mahdollista ylläpitää palvelimella olevia tiedostoja ja kansioita. Yleensä FTP-ohjelmaa tarvitaan verkkosivuprojektin ollessa julkaisukelpoinen. Kun sivusto siirretään julkiselle palvelimelle, on kaikilla verkon käyttäjillä mahdollisuus nähdä se. (WinSCP 7.6.2008. Introduction WinSCP [online] [viitattu 28.1.2009])



Kuva 1. Ruudunkaappaus kuva WinSCP-ohjelmasta

## 3.6 MySQL

MySQL on tietokannan hallintajärjestelmä (Meloni 2003, 7-16). Se on eräs tunnetuimmista avoimeen lähdekoodiin perustuvista tietokannoista. MySQL:n loivat suomalainen Michael Widenius ja ruotsalainen David Axmark vuonna 1995. Ensimmäinen virallinen versio siitä julkaistiin vuonna 1996, jonka jälkeen on kehitetty jatkuvasti uusia päivityksiä.

SQL-lyhenne tulee sanoista ”Structured Query Language”, joka tarkoittaa suomeksi rakenteista kyselykieltä. SQL on standardin mukainen kieli, mutta MySQL ei noudata standardeja kovinkaan tarkasti. Täysin merkkipohjaisen SQL-kielen tarkoituksena on suorittaa kyselyjä ja määrittelyjä relaatiotietokannoille esimerkiksi näppäimistöllä WinSQL-ohjelman ikkunasta. (Meloni 2003, 7-16)

Tietokannat toimivat tietovarastoina, joista voidaan hakea tallennettuja tietoja eri käyttötarkoituksiin. Relaatiotietokannat rakentuvat yhteen liitetystä tauluista, jotka koostuvat sarakkeista ja riveistä. Taulukot ovat suhteessa toisiinsa tiettyjen sarakkeisiin merkittyjen avainten pohjalta. (Meloni 2003, 11-12)

MySQL RDBMS (relational database management system) huolehtii käyttöoikeuksista, pitää kirjaa käyttäjän toimista ja hoitaa vastaukset kyselyihin. Kyselyiksi kutsutaan SQL-kielillä tehtyjä komentoja, joilla muokataan ja ylläpidetään tietokantoja. Kommunikointi RDBMS:in kanssa tapahtuu ohjelmistorajapinnan, tai kuten tässä projektissa asiakaskäyttöliittymän avulla. Hyvin tyypillinen tietokantaa hyödyntävä sovellus on verkkokauppa, jossa esimerkiksi tuotteiden tiedot säilötään tietokantaan. (Meloni 2003, 11-15)

### 3.6.1 MySQL:n hyödyt

MySQL:n miljooniin käyttäjiin sopii niin yksityisiä henkilöitä, kuin suuryrityksiäkin (Heinisuo & Rauta 2007, 37-40). Nykyaikaiset Internetsivustot ovat usein rakenteeltaan

toteutettu tietokantapohjaisesti, jolloin sivuston varsinainen sisältö haetaan sille luodusta tietokannasta. Tietokantapohjainen WWW-sivusto antaa mahdollisuudet toteuttaa todella monipuolisia sovelluksia.

Yleisimpiä tietokantoja hyödyntäviä interaktiivisia sovelluksia ovat verkkokaupat, keskustelupalstat, linkkikokoelmat ja erinäiset sähköpostilistat. Kaikki nämä sovellukset toimivat tavallisella WWW-selaimella. (Heinisuo & Rauta 2007, 37-40)

MySQL:n suosioon vaikuttavat merkittävästi sen nopeus, siirrettävyys, yhteensopivuus ja hinta. Nopeutta antaa MySQL:n monisäikeinen palvelin. Tämä tarkoittaa sitä, että jokaisen yhteyden muodostuessa aloitetaan aina uusi palvelinprosessi. Jos prosessi päättyy väärällä tavalla tai kuormittaa merkittävästi palvelimen muistia, vain yksi prosessi sulkeutuu, eikä palvelin näin ollen kaadu. (Heinisuo & Rauta 2007, 37-40)

Yhteensopivuus on MySQL:n tärkeä ominaisuus. Se toimii yhteen lähes kaikkien ohjelmointikielien kanssa, mukaan lukien Perl, PHP, Java, C/C++, Python ja Tel. MySQL toimii moitteettomasti eri käyttöjärjestelmissä ja siirto niiden välillä on vaivatonta. (Heinisuo & Rauta 2007, 37-40)

Hinta on merkittävä tekijä tietokantajärjestelmää mietittäessä. MySQL perustuu vapaaseen lähdekoodiin (open source) ja on yleensä ilmainen, eikä esimerkiksi verkkokaupan toteuttaminen kyseistä tietokantaa hyödyntäen vaadi erillistä lisenssiä. Ilmaista asiantuntevaa tukea on mahdollista saada MySQL-postituslistalta. (Heinisuo & Rauta 2007, 37-40)

### **3.6.2 Avoimen lähdekoodin tietokantojen edut**

Yhä useammat yritykset ovat siirtyneet käyttämään jatkuvasti kehittyvää vakaata, huippu suorituskykyistä ja halpaa avoimen lähdekoodin tietokantojenhallintasovellusta. Yritykset voivat valita useasta avoimen lähdekoodin tietokannoista heille parhaiten sopivan vaihtoehdon. (Yuhanna, Gilpin & Schwaber 2004)

Tunnetuimpia avoimen lähdekoodin tietokantoja Yuhannan ym. (2004) mukaan ovat:

- Ingres
- MaxDB
- Firebird
- PostgreSQL
- Berkeley DB
- Derby
- MySQL.

Avoimen lähdekoodin tietokannoilla on edessään kirkas tulevaisuus. Niitä kehitetään jatkuvasti ja ne tarjoavat kaupallisiin kilpailijoihinsa nähden virheettömämpää kokonaisuutta. Alhainen hankintahinta, vahva yhteisön tuki, alhaiset ylläpito- ja tukikustannukset, laajempi sovellusten ja laitteiden yhteensopivuus, turvallisuus sekä vapaa pääsy lähdekoodiin ovat tärkeimmät seikat, miksi yritykset siirtyvät käyttämään avoimen lähdekoodin tietokantoja. (Yuhanna ym. 2004)

### **3.6.3 Tietokantajärjestelmän valinta**

Tietokantojen hallintajärjestelmät voidaan ryhmitellä työpöytä- ja palvelintietokantoihin. Työpöytä-tietokannat on suunnattu lähinnä yhdelle käyttäjälle, joten niistä ei ole merkittävää hyötyä tässä sivustoprojektissa. Palvelin pohjainen tietokannanhallintajärjestelmä takaa toimintavarman ja yhdenmukaisen tiedon usealle samanaikaiselle käyttäjälle. (Chapple, n.d.)

Tietokantasovellusta valittaessa tulee tehdä täsmällinen tarveanalyysi siitä, mitä palveluita yritys tarvitsee. Tarveanalyysin tulisi vastata kysymyksiin: ketkä käyttävät tietokantaa, ja mitä toiminnallisuuksia siltä vaaditaan, kuka ylläpitää tietokantaa, ja kuinka usein sitä päivitetään, mitä laitteistoja voidaan käyttää, ja mikä on projektin budjetti, sekä kuinka tietoon tulee päästä käsiksi? (Chapple, n.d.)

### 3.6.4 Tietokantajärjestelmän tarveanalyysi

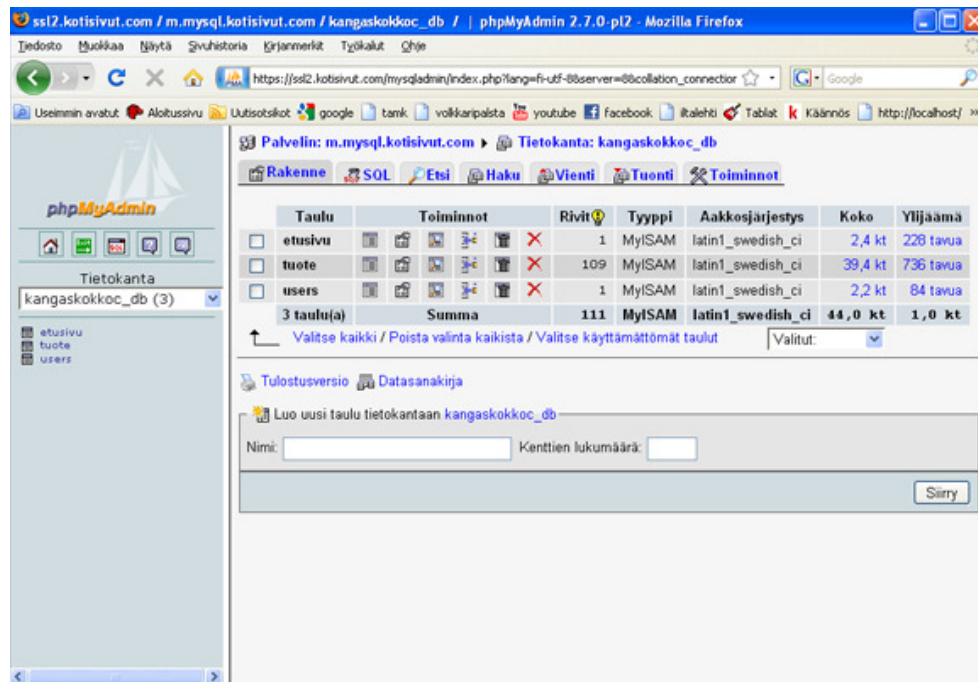
Tämän projektin tietokantajärjestelmän valintaan vaikuttavat erityisesti vaivaton ylläpito, toimintavarmuus, hinta, turvallisuus ja sivuston ylläpitäjän aiemmat kokemukset. Tietokanta rakennetaan niin, että ylläpitäjän tarvitsee puuttua sen rakenteisiin ja toimintaan mahdollisimman vähän. Tämä edellyttää sovellukselta toimintavarmuutta. Yritys vaatii, että sivusto toteutetaan mahdollisimman halvalla turvallisuudesta tinkimättä.

MySQL sopii ominaisuuksiensa ja toimintansa perusteella erittäin loogisesti projektin tietokantajärjestelmäksi. Projektin tietokantaan ei tarvitse tallentaa suuria tietueita, eikä sivustolla vieraile satoja ihmisiä kerralla. Esimerkiksi Oracle on turhan järeä ja kallis tietokantajärjestelmä pienyrityksen sivustoprojektiin. Valintaa tukee Expert Kangaskon palvelimelle valmiiksi asennettu MySQL-tuki ja erittäin käytännöllinen PhpMyAdmin-sovellus, jolla tietokantaa hallitaan.

### 3.6.5 PhpMyAdmin

PhpMyAdmin on PHP-kielillä toteutettu sovellus erityisesti MySQL-käyttöön (kuva 2). Sen käyttö on erittäin laajaa ja sitä kehitetään jatkuvasti. Sillä pystytään hallinnoimaan MySQL-tietokantoja, sekä tekemään niihin muutoksia suoraan käsin kirjoitetuilla komennoilla. Ohjelmasta löytyy useita käytännöllisiä toimintoja, kuten SQL-kyselyikkuna, johon tietokantakomennot voidaan syöttää. Uusimman version sovelluksesta voi ladata osoitteesta <http://www.phpmyadmin.net> (Heinisuo & Rauta 2007, 98-102)





Kuva 2. Ruudunkaappaus PhpMyAdmin-sovelluksesta

Jotta sovellusta pääsee käyttämään, tarvitaan tietokantapalvelimen tunnistavat käyttäjätunnus ja salasana. Yleensä tunnukset saadaan palvelimen ylläpitäjältä. Tunnukset tulee säilyttää huolella, ettei sivustolle tule väärinkäyttöä.

Kirjautumisen jälkeen voidaan luoda itse tietokanta. Tietokannan nimi tulee olla yksilöllinen ja se saa sisältää lähes mitä tahansa merkkejä, mutta on suotavaa käyttää normaaleja kirjaimia ja numeroita. Tällä varmistetaan se, että tietokanta toimii myös eri käyttöjärjestelmissä. (Heinisuo & Rauta 2007, 102-104)

Tietokantaan voidaan nyt lisätä taulut ja tarvittavat määrittelyt SQL-kyselyikkunasta. Tietokantaa koskevat tekstit kannattaa tallentaa erilliselle tekstitiedostolle, joka toimii tarvittaessa varmuuskopiona tietokannasta. Tekstiedostoa on myös helppo muokata ja päivittää. Muokattu teksti voidaan helposti liittää kyselyikkunaan. PhpMyAdmin-ohjelma ilmoittaa mahdollisista komentojen virheistä. Hyvin suunniteltu tietokanta ja sivusto on tämän jälkeen käyttövalmis. (Heinisuo & Rauta 2007, 108-112)

### 3.6.6 Käyttöoikeudet

Tietokantaa käyttävät asiakkaat, yrityksen työntekijät ja sivuston ylläpitäjä. Asiakkaille tietokanta tarjotaan tietoa yrityksestä, sen palveluista ja tuotteista. Ylläpidon puolella tehdyt muutokset tallennetaan suoraan tietokantaan. Yrityksen työntekijöille annetaan käyttäjätunnukset ja salasanat, joilla he voivat kirjautua sivuille. Näille tunnuksille on myönnetty oikeus lisätä, muokata ja poistaa tuotteita koskevia tietoja. Lisäksi oikeuksiin kuuluvat aukioloaikojen ja uutisten päivittäminen. Työntekijät voivat tarvittaessa hakea asiakkaiden yhteystietoja tietokannasta. Kaikki työntekijöiden tekemät yhteydenotot tietokantaan suoritetaan selaimen välityksellä.

Sivuston ylläpitäjällä on varsinaiset oikeuden muokata sivuston ja tietokannan kaikkia rakenteita. Ylläpitäjällä on mahdollisuus tarvittaessa laajentaa tietokantaa tai korjata siinä esiintyviä ongelmia suoraan palvelimen tietokannanhallintasovelluksesta. Ylläpitäjä on vastuussa tiedon turvaamisesta. Asiakkaiden henkilökohtaiset tiedot eivät saa päästä leviämään väärin käsiin. Tuotetietoja eivät saa päästä muokkaamaan ketkään muut kuin siihen valtuutetut henkilöt. Nämä seikat tulee ottaa huomioon jo tietokantaa suunniteltaessa.

Palvelinpuolella tietoturva painottuu sivuston konfigurointitiedostoihin. Niissä on tietokantayhteyden tarvitsemat tunnukset ja salasanat, jotka ovat kryptattu eli salattu. Lisäksi konfigurointitiedostojen kansio on salattu .htaccess-tiedostolla. Kyseinen tiedosto piilotetaan kansioon, jonka käyttäjä haluaa salata. Mikäli ulkopuolinen pääsee murtautumaan palvelimelle, täytyy hänen tietää .htaccess-tiedoston tunnus ja salasana, päästäkseen käsiksi salattuun kansioon (Apache HTTP Server Project 2008)

Tietokannan käyttöoikeuksilla on erittäin tärkeä osa sovelluksen toimintaa ja tietoturvaa. Jotta sivuston sisältö pysyy halutunlaisena, täytyy harkita tarkkaan, kenellä on oikeus muokata tietokannan tietoja. (Heinisuo, ym. (2007, 120-126). Käyttäjätunnuksilla pystytään siis määrittelemään se, mitä oikeuksia käyttäjällä on sivustolla. Tietokantaa tehdessä on operaatiot suoritettu oletuksena pääkäyttäjän root-tunnuksella. Jotta kaikki sivuston käyttäjät eivät pystyisi tekemään kaikkia tietokantaan liittyviä toimintoja, raja-

taan heidän oikeuksiaan halutulla tavalla. Oikeuksilla voidaan määrittää palvelimen ja jopa yksittäisen sarakkeen tasolla. Oikeudet koskevat yleensä taulujen tietojen lukua, kirjoittamista ja päivittämistä. MySQL-palvelimelle voidaan lisäksi määrittää jokin tietty tietokone, josta on mahdollista kirjautua sisään tietokantaan tunnuksilla. Tällöin tunnukseen liittyy myös sijainti. (Heinisuo & Rauta 2007, 120-126)

### **3.6.7 Tietokannan suunnittelu**

Ennen sovelluksen toteuttamista on erityisen tärkeää suunnitella tietokannan rakenne huolella. Oikein optimoidut tietokannan suhteet takaavat tehokkaan ja toimivan kokonaisuuden. Tiedon toistamista tulee välttää taulujen rakenteita suunniteltaessa. Jos useassa taulussa on sama tieto, voidaan luoda yksi erillinen taulu, josta tarvittava tieto löytyy. Avainten avulla pystytään näin ollen viittaamaan haluttuun arvoon. Näin kaikki taulut päivittyvät ainoastaan yhtä taulua muokkaamalla

Tässä projektissa tietokanta koostuu kolmesta erillisestä taulusta, jotka eivät ole keskenään suhteessa. Etusivut-tauluun tallennetaan etusivun sisältöön liittyvät tiedot. Tuotetauluun tallennetaan kaikki tuotteita koskevat tiedot ja Users-tauluun tallennetaan ylläpidon kirjautumiseen liittyvät tapahtumat (liite 2).

Tietokannan suunnittelu tulee tehdä huolella ennen varsinaisen sovelluksen toteuttamista. Huonosti toteutettu tietokanta tekee sovelluksesta hitaan ja pahimmassa tapauksessa käyttökelvottoman. On erittäin monimutkaista ja kallista muokata tietokantaa sovelluksen julkistamisen jälkeen.

Tietokannan toteuttaminen aloitetaan tutkimalla, mitä tietoja sovelluksessa käsitellään. Tämä vaihe kannattaa tehdä sovelluksen tilaajan, sen tulevien käyttäjien ja ylläpitäjien kanssa yhteistyössä. Näin varmistetaan, että kaikki tarvittavat osat saadaan mukaan tulevaan tietokantaan. Järkevä tapa on eritellä kaikki sovelluksen päätekijät ja listata niistä tarvittavat tiedot. Esimerkiksi verkkokaupassa päätekijät voisivat olla asiakas, tuote, ti-

laus ja tuotteen toimitus. Näistä kohdista täytyy saada talteen kaikki tarvittavat tiedot, jotta tilattu tuote saadaan toimitettua oikeaan osoitteeseen.

### 3.6.8 Tietokannan toteutus

Kun kaikki tietokantaan liittyvät otsikot on mietitty huolella, voidaan luoda taulut ja tehdä niihin kuuluvat kentät, joihin data tallennetaan. Taulut ovat tietokannan suurimpia elementtejä. Ne koostuvat kolmesta osasta tiedostojärjestelmässä: rakenteen, datan ja indeksit sisältävistä tiedostoista. (Heinisuo & Rauta 2007, 108-109)

Kentille täytyy määrittää tietotyypit sen mukaan, minkälaista tietoa niihin halutaan tallentaa. Tietotyypeillä pyritään rajaamaan tiedon tyyppi, jotta tietokantaan saadaan tallennettua haluttu informaatio. Tietotyypit kannattaa toteuttaa huolella, koska niillä on merkittävä osuus tiedon käsittelyssä ja tietokannan toimivuudessa. (Heinisuo & Rauta 2007, 108-110)

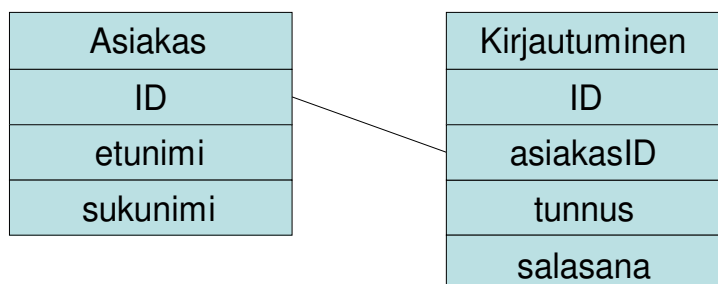
Tyypillisimpiä tietotyypppejä ovat:

- *CHAR – Kiinteän mittainen merkkijono*
- *VARCHAR – Merkkijonot, 1-255 merkkiä*
- *INTEGER – kokonaisluku*
- *DECIMAL – Desimaaliluku*
- *REAL - Liukuluku*
- *DATE – Päivämäärä*
- *TIME – Kellonaika*
- *TIMESTAMP – Päivämäärä ja aika*
- *TEXT – Tekstiä, maksimissaan 65535 merkkiä*
- *BIT – Binääristä tietoa, esimerkiksi kuva.*

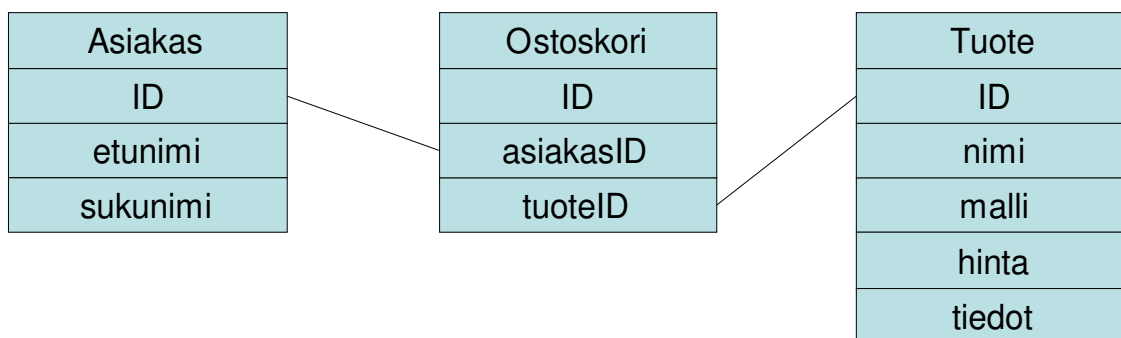
Kun kaikki taulut on luotu ja niiden tietotyypit on määritelty huolella, suunnitellaan taulujen väliset suhteet, eli kuinka ne liittyvät keskenään toisiinsa. MySQL-tauluissa ja tietueissa käytetään avaimia taulujen yhteen sitomisessa. Avaimet voivat olla joko uniikkeja (unique key), tai ensisijaisia (primary key). Uniikilla avaimella määritetään kenttä ainutlaatuiseksi. Tällöin sama tieto ei voi olla useammalla rivillä tai tietueessa. Esimerkiksi tuote-taulussa tuotteen ID-kenttä tulee olla uniikki, jotta tuotteiden tiedot eivät mene keskenään sekaisin. (Heinisuo & Rauta 2007, 108-112)

Uniikkeja avaimia voi olla taulussa vain yksi. Ensisijainen avain (primary key) on hyvin paljon samanlainen kuin uniikki avain, koska molempien avainten tulee saada ainutlaatuiset arvot. Ensisijainen avain toimii linkkinä yhteen tai useampaan tauluun. Taulussa voi olla useampi ensisijainen avain. Yleensä ensisijaista avainta käytetään automaattisen ID:n lisäämisen yhteydessä tauluun. (Heinisuo & Rauta 2007, 108-112)

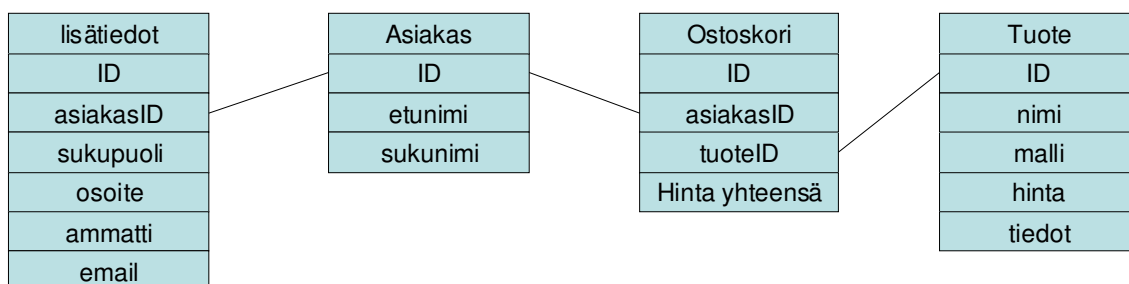
Taulut voivat liittyä toisiinsa kolmella eri tavalla: yksi-yhteen (kuva 3), yksi-moneen (kuva 4) ja moni-moneen (kuva 5). Kun taulujen suhde on yksi yhteen, avain esiintyy vain kerran taulussa, johon viitataan. Yksi moneen suhteessa avaimet yhdestä taulusta esiintyvät useasti viitatussa taulussa. Moni moneen on kaikista ongelmallisoin suhde. Onkin tavallista jaotella monen suhde moneen useaksi yksi moneen suhteeksi. (Meloni 2003, 28-32)



*Kuva 3. Esimerkki taulujen välisestä yksi-yhteen suhteesta*



Kuva 4. Esimerkki taulujen välisestä yksi-moneen suhteesta



Kuva 5. Esimerkki taulujen välisestä moni-moneen suhteesta

### 3.6.9 Tietokannan tarkastaminen

Normalisointi on menetelmä, jolla organisoidaan ja viimeistellään tietokannan taulujen väliset suhteet. Normalisoinnissa tulee ottaa huomioon joustavuus, jotta tietokanta voi tarvittaessa kasvaa tulevaisuudessa. (Meloni 2007, 32-37)

Normalisointi voidaan toteuttaa viidessä eri osassa. Tässä tapauksessa tarkastellaan ai-noastaan kolmea olennaisinta normalisoinnin kohtaa. Ensimmäisessä normalisoinnin osassa moniarvoiset kentät, eli attribuutit poistetaan, ja ne siirretään omiin tauluihinsa. Toisessa osassa tarkastetaan, koostuuko taulun avainehdokas useasta attribuutista. Näin ollen attribuutit, jotka eivät ole avaimia, eivät saa olla toiminnallisesti riippuvaisia mistään avainehdokkaasta. Jos attribuutti on riippuvainen koko avaimesta, se saa sijaita taulussa. Kolmannessa normalisoinnin osassa tutkitaan, että taulun ei-avainkenttien tulee riippua avainkentistä. (Meloni 2007, 32-37)

### 3.7 PHP

PHP on lyhenne sanoista *HyperText Preprocessor*. PHP on palvelinpuolen vapaaseen lähdekoodiin perustuva komentosarjakieli (*scripting language*). Alunperin sen suunnitelti tanskalainen Rasmus Lerdorf henkilökohtaiseen käyttöönsä vuonna 1994. PHP:n julkiseksi tuleminen myötä sen suosio räjähti valtavaan kasvuun ja laajenee edelleen to-della nopeasti. HTML-kielen kanssa yhteistyössä toimivan PHP:n alkuperäisenä tarkoi-tuksena oli mahdollistaa monipuolisempien WWW-sovellusten toteuttaminen. (Hei-nisuo & Rauta 2007, 26)

Tässä projektissa valtaosa sivuston toiminnoista on tehty PHP:n avulla (kuva 6). Kysei- sessä koodissa haetaan funktioita.php, jossa sijaitsee tietokantaan ja kirjautumiseen liit- tyviä määrittämiä. Lisäksi koodissa määritellään sivunumerot ja tuotteiden määrä yhdel- le sivulle. Koodiin on määritelty tuotekuvien kansio ja MySQL-komennoilla haetaan tarvittavat tuotetiedot.

Kun tuote, etusivuntiedot tai aukioloaikoja lisätään tietokantaan, täytetään aluksi taval- linen HTML-kaavake, jossa on kaikki tarvittavat tiedot. Tämän jälkeen tiedot viedään PHP-koodiin liitettyllä MySQL-komennolla juuri oikeaan paikkaan tietokannassa. Kun tiedot on tallennettu tietokantaan, näkyvät tuotteet asiakaspuolella automaattisesti, käyt- täjän klikattua Tuotteet-linkkiä.

```

<?php

// Tämä on tuotesivu, jolla listataan tuotteita.

require("funktioita.php");

// Montako tuotteen mallia näytetään kerrallaan?
define("TULOXSIA_SIVULLA", 10);

// Lasketaan mitkä tulokset halutaan näyttää tällä kertaa.
$sivunumero = 0;
if(isset($_GET["sivu"]) && ctype_digit($_GET["sivu"]))
    $sivunumero = $_GET["sivu"];

$limit = intval(TULOXSIA_SIVULLA);
$offset = intval($sivunumero * TULOXSIA_SIVULLA);
$img_folder = "kuvat/tuotekuvat/";
// Haetaan tuotteiden mallit.
$sql = "select sql_calc_found_rows id, nimi, mallinro, ryhma, hinta, tarjoushinta, luotu, kuvanimi from
    tuote order by luotu desc limit $limit offset $offset";
$tulos = mysql_query($sql, $yhteys);
if(!$tulos) exit("Tietokantahaku epäonnistui: ".mysql_error());

// Lasketaan montako sivullista tuloksia on yhteensä.
$sql = "select found_rows() as riveja";
$rivitulos = mysql_query($sql, $yhteys);
$tarjoushinta = "select sql_calc_found_rows tarjoushinta from tuote";
if(!$rivitulos) exit("Tietokantahaku epäonnistui: ".mysql_error());

$riveja = mysql_fetch_assoc($rivitulos);
// Pyöristetään ylös ceil-funktiolla.
$sivuja = ceil($riveja["riveja"] / TULOXSIA_SIVULLA);

```

### *Kuva 6. PHP-koodia*

Dynaamisissa WWW-sivuissa PHP toimii porttina tietokantaan. Se mahdollistaa tiedon haun lisäksi päivittämisen, kuten tiedon lisäämisen ja poistamisen selaimen kautta suoraan tietokantaan. PHP-kielen komennot voivat olla missä tahansa kohtaa koodia tai WWW-palvelimen tekstipohjaista tiedostoa. PHP-käskyt suoritetaan aina ensimmäisenä ennen, kuin WWW-palvelin lähettää sivun selaimelle. (Heinisuo & Rauta 2007, 26-34)

PHP-koodi ei toimi suoraan käyttäjän tietokoneen kiintolevyllä, vaan se vaatii toimiakseen WWW-palvelimen, jossa on PHP-tuki, ja jonne tiedostot on siirrettävä esimerkiksi erillisellä FTP-ohjelmalla. Apache-palvelinta käytettäessä voi PHP toimia sen sisäisenä moduulina, jolloin se on kiinteänä osana WWW-palvelinta. Tämä mahdollistaa sen, ettei PHP tarvitse erillistä tulkkia toimiakseen. Palvelimelle täytyy ainoastaan asentaa PHP-tuki. WWW-selaimen ei kuitenkaan tarvitse asentaa mitään ylimääräistä liitännäistä PHP:n käyttöä varten. Nämä ovatkin avainseikat PHP:n nopeaan toimintaan. Mo-



net muut palvelinperusteiset CGI-ohjelmat (*Common Gateway Interface*) vaativat toimintaansa oman tulkin. PHP:n etuihin lukeutuu myös sen laajennettavuus monipuolisemmaksi eri ohjelmakirjastoilla. (Heinisuo & Rauta 2007, 17-24)

### 3.7.1 PHP:n edut

PHP:n etuihin kuuluvat helppo käytön oppiminen, toiminnallinen kilpailukykyisyys ja edullisuus (McBride 2005, 3-10). PHP:n perustoimintojen oppiminen on nopeaa verrattuna muihin ohjelmointikieliin. Hankalia teoriaosuuksia ei tarvitse tuntea paljoakaan, jotta pääsee toteuttamaan ensimmäisiä PHP-komentoja. PHP tarjoaa yksinkertaisilla koodeilla monipuolisia toiminnallisuksia. (McBride 2005, 3-10)

PHP on erittäin monipuolinen ja helppo kieli. Perustiedoilla pääsee hyvin ohjelmoinnin alkuun. PHP:n oppimista auttaa samankaltaisuus monen muun ohjelmointikielen kanssa. Käyttäjän tietojen ja taitojen kehittyessä on mahdollista toteuttaa yhä tehokkaampia ja laajempia sovelluksia. PHP:n uusimpien versiot mahdollistavat niin sanotun olio-ohjelmoinnin. Olio-ohjelmointi on tuttua esimerkiksi Java-, C++- ja Python ohjelmointikielistä. Oikein käytettyinä ne parantavat sovellusten suorituskykyä, päivitettävyyttä ja vähentävät ohjelmointivirheitä. PHP on yhteensopiva monille eri tietokannoille, sovellusluustoille ja palvelimille. Sitä päivitetään jatkuvasti ja sen sisältämät ongelmatkohtat pyritään korjaamaan heti niiden ilmenemisen jälkeen. (Heinisuo & Rauta 2007, 26)

PHP perustuu avoimeen lähdekoodiin samoin kuin MySQL. Lähdekoodin voi nähdä ja sitä voi muokata vapaasti. Tämän vuoksi PHP:n lataaminen ja käyttö on täysin ilmaista. Niinpä kokonaisuutena, yhteensopivuuden ja monipuolisuuden ansiosta PHP ja MySQL ovat todella oivallinen parivalinta dynaamisen WWW-sovelluksen toteuttamiselle. (McBride 2005, 6)

### 3.7.2 PHP:n käyttö

PHP:n käyttö liittyy yleensä palvelinpuolella tapahtuvaan interaktiivisten WWW-sivujen toteuttamiseen. Sitä voidaan käyttää myös verkkosovelluksissa. PHP-koodit ovat osa WWW-sivujen toiminnallista kokonaisuutta, vaikka ne eivät välttämättä itse selaimessa näykään sivuston käyttäjälle.

PHP-koodia voidaan käyttää myös komentokehoteessa. Yleensä tästä ominaisuudesta ei hyödy monikaan käyttäjä, mutta palvelimen ylläpitäjälle siitä voi olla suurtakin hyötyä. Komentokehoteessa voi kirjoittaa esimerkiksi aikaa säästäviä palvelimen ylläpitoon liittyviä käskyjä. (McBride 2005, 8-9)

Tietokoneille voidaan tehdä myös käyttöjärjestelmän ikkunassa toimivia sovelluksia. Tällainen PHP:n käyttö ei ole kovin yleistä, mutta sillä voidaan toteuttaa useilla eri sovellusalustoilla toimivia ohjelmia. Peliin tai muiden nopeita syötteitä käyttäjältään vaativien sovellusten rakentamiseen PHP ei ole sopiva työkalu. (McBride 2005, 8-9)

## 4 PHP ja MySQL

PHP tekee mahdolliseksi SQL-komentojen välittämisen WWW-sivun ja tietokannan välillä. Näin tietokannasta saadaan haluttu tieto esille suoraan WWW-selaimeen (kuva 7) . Kun PHP ja MySQL toimivat yhdessä, PHP ei käsittele tietokantaa suoraan, vaan sen avulla muodostetaan yhteys tietokantaan, josta poimitaan halutut tiedot MySQL-komentojen avulla. (Meloni 2003, 13-14)



**expert**

**LG**

**32LT75**

- 32 tuuman HD-Ready LCD-DIGITV
- 160GB kiintolevy
- resoluutio 1366x768p pikseliä
- kontrasti 8000:1
- 2 digitaalista viritintä
- 160Gt kovalevy
- 2 x HDMI

Hinta: 999 €

**Tarjoushinta: 899 €**

[Valmistajan kotisivulle](#)

[Siirry takaisin tuotesivulle](#)

Kuva 7. Tietokannasta haetut tuotetiedot WWW-selaimessa

Sivuston suunnittelu tulee aloittaa sisällön määrittelystä. Aluksi täytyy kartoittaa, mitä sivustolla tulee olla ja mikä on kohderyhmä. Tämän jälkeen voidaan suunnitella sisällön pohjalta tietokannan rakennetta esimerkiksi paperille listaamalla. Kun lista on valmis, voidaan rakentaa tietokannat ja normalisoida ne ohjeita noudattamalla. (Meloni 2003, 27-28)

Seuraava vaihe on itse sivuston ja PHP-toimintojen määrittely. Täytyy tutkia, mitä tietoa halutaan syöttää tai hakea tietokannasta millekin sivulle. PHP-toiminnot ja syöteen-tarkistukset tulee testata huolella, jotta tietokantaan ei saada syötettyä turhaa tai väärää tietoa.

Kun nämä vaiheet on suunniteltu ja toteutettu huolella, voidaan aloittaa itse sovelluksen testaaminen. Kaikki kohdat ja toiminnot tulee tarkastaa useaan otteeseen. Erityisen tärkeää on testata syöteentarkistukset, jotta esimerkiksi syntymäaika-kohtaan ei voi syöttää omaa nimeään. Tämä voi aiheuttaa yllättäviä ongelmia jatkossa. Lopuksi koko sovellus on valmis siirrettäväksi palvelimelle julkiseen käyttöön. Täytyy kuitenkin muistaa, että yleensä suurin osa ongelmista ilmenee vasta pitempi aikaisen käytön jälkeen, jonka vuoksi on suotavaa lisätä sivulle toiminto, jossa käyttäjät voivat ilmoittaa sivuston mahdollisista ongelmista.

PHP:n käyttöön on saatavilla valtavasti ohjeita ja esimerkkejä (liite 1). Helpoin ja nopein tapa on etsiä niitä Internetistä. Suomenkielellä löytyy melko paljon apua, mutta englanninkielellä sitä löytyy huomattavasti enemmän.

Erittäin suositeltavaa on hankkia alan kirjallisuutta, joissa opastetaan alkuunpääsyyn esimerkkien avulla. Kirjallisuutta hankkittaessa kannattaa pitää silmällä julkaisuvuosia. PHP kehittyi koko ajan ja tällä hetkellä uusin versio on 5.2.8. Tämän vuoksi uusien painosten tarjoaa aina ajankohtaisempaa tietoa. Esimerkkejä tekemällä ymmärtää helpoiten, mistä PHP-kielessä on kysymys, ja kuinka se toimii.

Keskustelupalstat tarjoavat ajankohtaista tietoa, joihin voit esimerkiksi lähettää ongelmakohdan sisältävän PHP-koodinpätkän. Asiantuntevaa apua on yleensä tarjolla nopeasti.

PHP:n opiskelun voi aloittaa näistä (*The PHP Group, 2001-2009*):

- Internet dokumentit
- Kirjat
- Alan lehdet
- sähköpostilistat
- Uutisryhmät
- Käyttäjryhmät
- Tapahtumat ja koulutukset
- Esimerkit
- Reaaliaikaiset keskustelusovellukset
- PHP.net-sivuston ylläpitäjät

## 5 Sivuston rakenne

Dynaaminen verkkosivusto koostuu yleensä useasta eri elementistä, joiden rakenteiden kuvaaminen pelkällä tekstillä voi olla erittäin haastavaa. Teknillisen dokumentaation tarkoituksena on havainnollistaa tarkemmin, mistä elementeistä verkkosivusto koostuu ja millä lailla ne ovat kytköksissä toisiinsa. Arkkitehtuurin avulla on myös helpompi suunnitella sivuston muokkaamista ja ratkoa mahdollisia ongelmia. Yleisarkkitehtuurissa käydään läpi sivuston toiminta pääpiirteittäin.

### 5.1 Yleisarkkitehtuuri

Suunnittelussa tuli ottaa huomioon, että asiakkaat eivät saa päästä ylläpidon puolelle. Ennen varsinaista sovelluksen toteuttamista päätettiin, että sivusto koostuu erillisistä asiakkaan- ja ylläpidon -osiosta (liite 3). Liitteestä näkee, kuinka asiakas- ja ylläpitosivustot ovat kytköksissä toisiinsa. Asiakas-osiossa on mahdollista ainoastaan selailta eri sivuja. Etusivulla näkyy otsikkoteksti, uutiset ja yrityksen aukioloajat. Tuotteet- ja tarjous-sivuilla voi tutustua yrityksen tarjoamiin tuotteisiin hintoineen ja tuotetietoineen. Yhteystiedoista löytyy tarkemmat tiedot yrityksestä ja palvelu-sivulla kerrotaan, mitä yleisesti yrityksen tarjoamista palveluista.

### 5.2 Tiedostot ja niiden väliset suhteet

Projektin verkkosivuston rakenne koostuu yhteensä 31:stä tiedostosta (Liite 4). Luku pitää sisällään sivuston rakenteen määrittelevät tiedostot, ulkoasun määrittelevän tyyli-tiedoston ja tietokantayhdeydestä vastaavat konfigurointitiedostot. Koodaamiseen kannattaa varata riittävästi aikaa, koska pienetkin toiminnot saattavat vaatia paljon työtä ja aiheuttaa ongelmia. Tämän lisäksi ulkoasun viimeistelystä vastaavat useat kuvatiedostot. Liitteessä on merkattu nuolilla, millä tavalla tiedostot ovat suhteessa toisiinsa.

Tyylitiedosto (tyyli.css) on tärkeä osa jokaista sivua (kuva 8). Asiakaspuolen sivuilla tyylitiedosto liitetään osaksi funktioita.php-tiedostoon, joka puolestaan on liitettyinä kaikkiin asiakaspuolen sivuihin. Tyylitiedostossa määritellään verkkosivun kullekin elementille määrytykset. Määrytykset voivat koskea esimerkiksi elementin ulkoasua ja asemointia. Funktioita.php-tiedostoon on liitettyinä konfigurointitiedostot, joilla muodostetaan yhteydet tietokantaan.

Lopuksi sivuston koodit käytiin läpi ja ne siistittiin turhista osista ja kommentteista.

Kaikki sivuston toiminnallisuudet, linkitykset ja ulkoasu testattiin tunnetuimmilla www-selaimilla.

```
#leftnav
(
float: left;
width: 180px;
min-height: 400px;
height: 600px;
margin: 0;
padding: 1em;
background-image: url(../kuvat/pikkukuvat/leftnav.jpg);
border-right-style: solid;
border-right-color: #E6E6E6;
)

#content
(
padding: 1em;
min-height: 400px;
height: 600px;
width: auto;
overflow: auto;
text-align: left;
background-image: url(../kuvat/content.jpg);
background-repeat: repeat-y;
```

*Kuva 8. Esimerkki tyylitiedoston rakenteesta*

Asiakaspuolen etusivulta voidaan kirjautua tunnuksen ja salasanan avulla ylläpidon puolelle, jossa voidaan suorittaa kaikki tarvittavat päivitykset. Ylläpidon puolella tyylitiedosto ja kirjautumistoiminto on liitetty yläosa.php-tiedostoon, jotka puolestaan ovat liitettyinä jokaiseen ylläpidon sivuun. Ylläpidon puolella etusivulta löytyy linkki, josta avautuu etusivun tietojen muokkaus kaavake (kuva 9).

**Etusivun otsikko**

Tervetuloa Expert Kangaskokon uusille verkkosivuille.

**Etusivun teksti:**

Tähän voidaan kirjoittaa mitä tahansa tekstiä.

**Uutisotsikko:**

Uutiset:

**Uutinen:**

Tähän tulee uutisia.

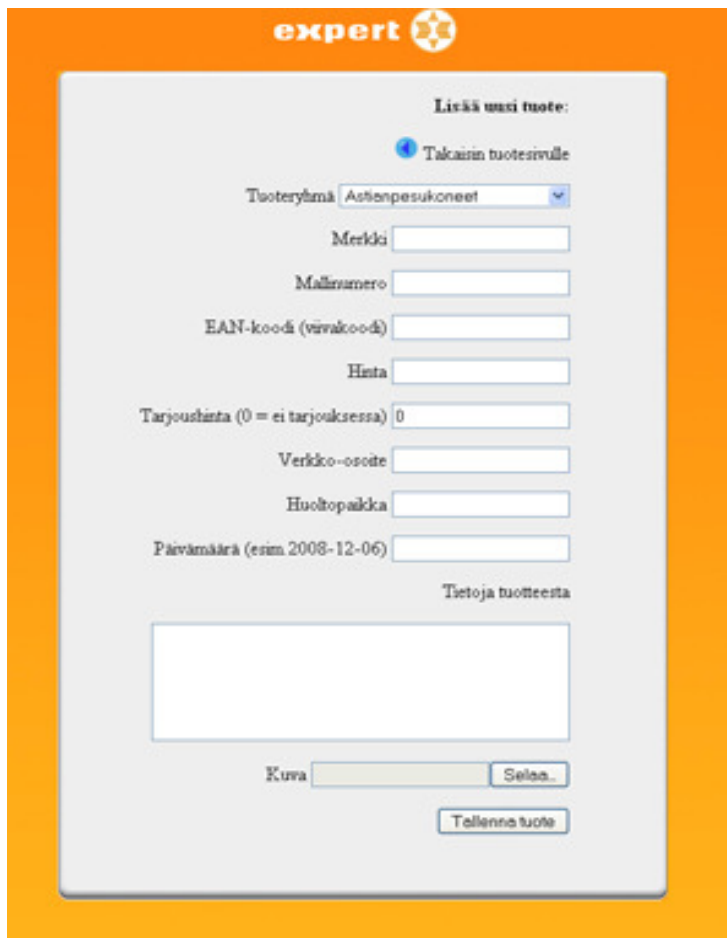
**Aukioloajat:**

Expert palvelee  
Ma-Pe 9-17  
La 9-13

*Kuva 9. Etusivun muokkauskaavio*

Kirjautumalla ylläpidon puolelle, on käyttäjällä mahdollisuus lisätä tuotteita Tuotteet-sivulta (Kuva 10). Kun tuote on lisätty, voidaan sen tietoja muokata jälkikäteen valitsemalla tuote listasta ja painamalla muokkaa-linkkiä. Muokkaa-kaavio on ulkoisesti samannäköinen kuin Tuotteen lisäämisen-kaavio, mutta sen sarakkeisiin haetaan tiedot automaattisesti muokkaamista varten. Tuotteet-sivulta voi myös tarvittaessa poistaa koko tuotteen. Kaikki ylläpidon puolella tehdyt muutokset näkyvät suoraan asiakaspuolen sivuilla.





**expert**

**Lisää uusi tuote:**

Takaisin tuotesivulle

Tuoteryhmä

Merkki

Mallinumero

EAN-koodi (viivakoodi)

Hinta

Tarjoushinta (0 = ei tarjouksessa)

Verkkosoite

Huoltopaikka

Päivämäärä (esim. 2008-12-06)

Tietoja tuotteesta

Kuva

*Kuva 10. Tuotteen lisääminen*

## 6 Kehittämismahdollisuudet

Expert Kangaskokon WWW-sovellusta olisi tarvittaessa helppo kehittää pitemmälle. Sivuston toimintaan olisi mahdollista liittää verkkokauppa, yhteys varastokirjanpitoon, elektroniikka-uutiset, kanta-asiakaspisteet, asiakkaiden ostot, sekä sähköpostilista.

Sivustolle olisi melko yksinkertaista toteuttaa verkkokauppa, jolloin asiakkaat voisivat tehdä tilauksia suoraan yrityksen WWW-sivuilla. Jotta tuotteiden lähettäminen onnistuisi, tulisi asiakkaiden rekisteröityä sivuille täyttämällä käyttäjä- ja yhteystiedot. Tämän jälkeen asiakkaille annettaisiin tunnus ja salasana, jolloin heillä olisi mahdollisuus ostaa tuotteita. Ostosten maksu tulisi suorittaa verkkopankin kautta, luottokortilla tai suoralla tilisiirrolla. Tässä projektissa verkkokauppaa ei kuitenkaan toteuteta, koska se vaatisi ylimääräistä työvoimaa, jota pienyrityksellä ei välttämättä ole tarjota.

Sivuston tietokantaan olisi mahdollista lisätä yhteys yrityksen kassaohjelman varastotietokantaan. Tämän avulla tuotteiden tiedot, kuten kappalemäärät ja hinnat näkyisivät reaaliajassa myös WWW-sivuilla. Tällä tavalla sivuston päivittäminen helpottuisi entisestään ja aikaa säästyisi.

Tietokannan asiakas-tauluun voitaisiin kerätä verkkokaupan myötä tarkempia tietoja asiakkaiden toimista, kuten ostoista. Tällä tavalla voitaisiin seurata, mitä tuotteita kukin asiakas yleensä ostaa. Tämä mahdollistaisi esimerkiksi täsmällisen tuotetarjousten ja mainosten lähettämisen asiakkaille, mikäli he suostuvat siihen. Verkkoo-ostoista voisi kertyä kanta-asiakas pisteitä. Pisteillä voisi saada esimerkiksi alennusta tai ilmaistuotteita, kun pisteitä on kertynyt tarpeeksi.

Kun asiakas täyttää rekisteröitymiskaavaketta, voi hän lisätä itsensä halutessaan sähköpostilistaan. Näin yritys voi lähettää hänelle sähköpostia esimerkiksi uusista tuotteista, tuote-esittelyistä, aukioloajoista ja tarjouksista.

WWW-sivuille voitaisiin tuoda RSS-syötteiden (*Really Simple Syndication*) avulla elektroniikka-uutiset. Täten asiakkaat voisivat seurata yrityksen verkkosivuilla esimerkiksi elektroniikan ja uusien tuotteiden kehittymistä.

Ottaen huomioon yrityksen tämän hetkisen tilanteen, ei yllämainituista lisäyksistä ole niin paljon hyötyä, että niitä lisättäisiin lopulliseen sovellukseen. Yritys tarvitsisi ylimääräistä työvoimaa hoitamaan sivustoa, tuotteiden pakkaamista ja lähettämistä, sekä sähköpostipalvelua. Sivusto rakennetaan kuitenkin niin, että sivuston laajentaminen edellä mainituilla tavoilla on tulevaisuudessa mahdollista.

## 7 Päätelmät

Projektin tavoitteet huomioon ottaen, verkkosivujen suunnittelu ja toteutus onnistuivat erittäin hyvin. Työ eteni alkuperäisen suunnitelman mukaan ja aikataulussa pysyttiin melko tarkasti. Yhteensä suunnitteluun ja itse sivuston kehittämiseen aikaa kului noin 160 tuntia. Tähän ei ole laskettu tuotteiden lisäämiseen ja sivuston testaamiseen kulu- nutta aikaa.

Sivustolla ilmeni muutamia ongelmia tietyillä selaimilla. Vierityspalkin toiminnalla ja joissakin tyylitiedoston asetuksissa oli ongelmia, mutta ne saatiin korjattua sivuston tes- taamisen yhteydessä.

Sivuston käyttöä testanneilta henkilöiltä saatiin palautetta, jonka mukaan sivuston ra- kenne, päivittäminen ja helppokäyttöisyys täyttivät alkuperäiset tavoitteet. Testaamiseen osallistui yrityksen työntekijät ja yrityksen asiakkaat, sekä muutamat verkkosivujen ke- hittämiseen perehtyneet henkilöt. Muutamia huomautuksia tuli hieman vääristä tuotetie- doista, mutta nopean päivittämisen ansioista ne oli mahdollista korjata erittäin nopeasti.

Yrityksen työntekijöille pidettiin lyhyt koulutus liittyen sivuston ylläpitoon. Tämän jäl- keen varmistettiin, että jokainen työntekijä osasi suorittaa sivuston ylläpidon vaatimat tehtävät. Kuvan etsiminen kuvakansiosta aiheutti ainoastaan hieman hankaluuksia tuot- teen lisäämisen yhteydessä. Tuotetietojen muokkaaminen, tuotteiden poistaminen, etu- sivun tietojen muokkaaminen ja aukioloaikojen muokkaaminen onnistuivat ongelmitta.

Omalta osaltani voin todeta, että opinnäytetyö oli melko haastava työmäärältään yhdelle henkilölle. Projekti sujui kuitenkin ilman suurempia ongelmia, ja vieläpä aikataulussa. Projekti kaikkine osineen oli erittäin hyödyllinen oppimisen ja vanhan kertaamisen kan- nalta.

## Lähdeluettelo

- Apache HTTP Server Project 2008. .htaccess-files [online] [30.1.2008]  
<http://httpd.apache.org/docs/1.3/howto/htaccess.html>
- Chapple, Mike. About.com. Choosing a Database for Your Organization: Desktop vs. Server Database Systems. [online] [viitattu 11.11.2008].  
[http://databases.about.com/od/administration/a/choosing\\_a\\_db.htm](http://databases.about.com/od/administration/a/choosing_a_db.htm)
- Goto, Kelly & Cotler, Emily 2003. Verkkopalveluprojekti, Käännös: Riitta Santala-Köykkä. Helsinki: Edita Prima Oy
- Heinisuo, Rami, Rauta, Ilkka 2007. PHP ja MySQL Tietokantapohjaiset verkkopalvelut. Helsinki: Talentum Media Oy
- McBride, Nat 2005. PHP with MySQL. London, Enland: Teach yourself
- Meloni, Julie C. 2003. MySQL Trainer Kit. Helsinki: Edita Prima Oy
- Oliver, Dick 2002. HTML & XHTML Trainer Kit, IT Press, Käännös: Jussi Arola. Helsinki: Edita Prima Oy. ISBN: 951-826-608-5
- Rosen, Lawrence 2004. Open Source Licensing, Software freedom and Intellectual Property Law. New Jersey: Prentice Hall Professional Technical reference
- The PHP Group, 2001-2009. PHP Support. [online] [viitattu 6.11.2008].  
[www.php.net/support.php](http://www.php.net/support.php)
- WinSCP 2008. Introduction WinSCP [online] [viitattu 28.1.2009]  
<http://winscp.net/eng/docs/introduction>
- W3C 2001. Modularization of XHTML, W3C Proposed Recommendation 22.2.2001. [online] [viitattu 10.11.2008]. <http://web5.w3.org/TR/2001/PR-xhtml-modularization-20010222/xhtml-modularization.pdf>

- Yuhanna, Noel, Gilpin, Mike & Schwaber, Carey 29.12.2004. Open Source Databases Come Of Ages. [online] [viitattu 11.11.2008].  
<http://www.mysql.com/why-mysql/analyst-reports/forrester-open-source-databases.pdf>

## Liite 1. Linkkilista hyödyllisistä PHP- ja MySQL-verkkosivuista

### Linkkilista suomalaisista PHP-sivuista

- <http://www.ohjelmointiputka.net/oppaat.php>
- [http://users.jyu.fi/~kolli/ITK215\\_05/php/](http://users.jyu.fi/~kolli/ITK215_05/php/)
- <http://wiki.mureakuha.com/wiki/PHP-opas>
- <http://www11.uta.fi/~jl/php/esim/index.php>
- <http://fi.wikibooks.org/wiki/PHP>
- <http://www.sivut.org/php/>

### Linkkilista englanninkielisistä PHP-sivuista

- <http://www.php.net/>
- <http://www.phpbuilder.com/>
- <http://www.hotscripts.com/PHP/index.html>
- <http://www.planet-php.net/>
- <http://gscripts.net/>
- <http://www.php-scripts.com/>
- <http://www.w3schools.com/PHP/DEfaULT.asP>
- <http://www.tizag.com/phpT/>
- <http://www.thesitewizard.com/archive/feedbackphp.shtml>
- <http://tut.php-quake.net/en/>
- <http://www.freewebmasterhelp.com/tutorials/php>
- <http://www.htmlgoodies.com/beyond/php/article.php/3472391>
- <http://www.phpfreaks.com/forums/index.php/board,1.0.html>

**Linkkilista suomalaisista MySQL-sivuista**

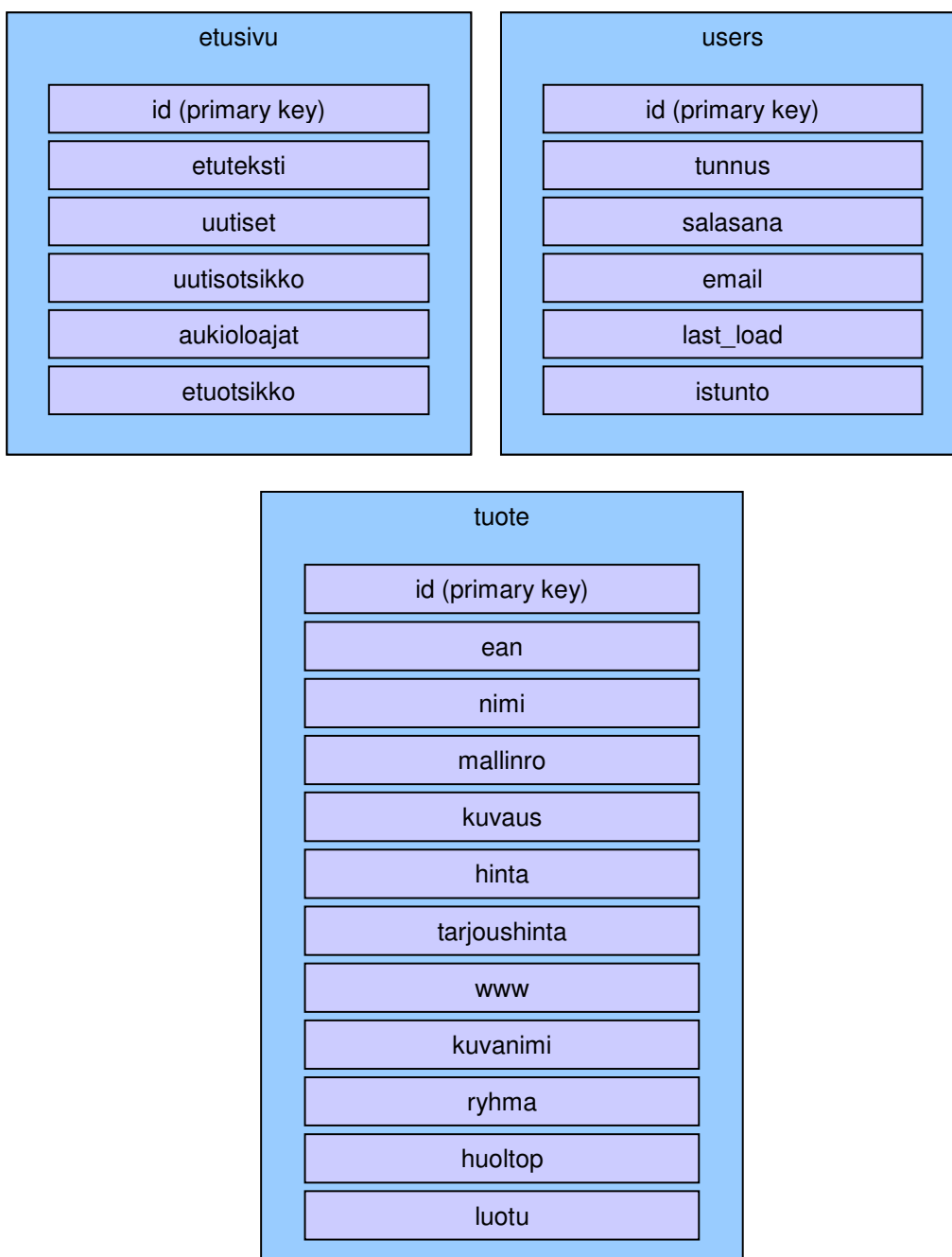
- <http://www.ratol.fi/opensource/mysql/index.htm>
- [http://users.jyu.fi/~kolli/ITK215\\_05/php/?sivu=tietokanta](http://users.jyu.fi/~kolli/ITK215_05/php/?sivu=tietokanta)
- [http://lukk.org/index.php/MySQL:n\\_hallinta](http://lukk.org/index.php/MySQL:n_hallinta)

**Linkkilista englanninkielisistä MySQL-sivuista**

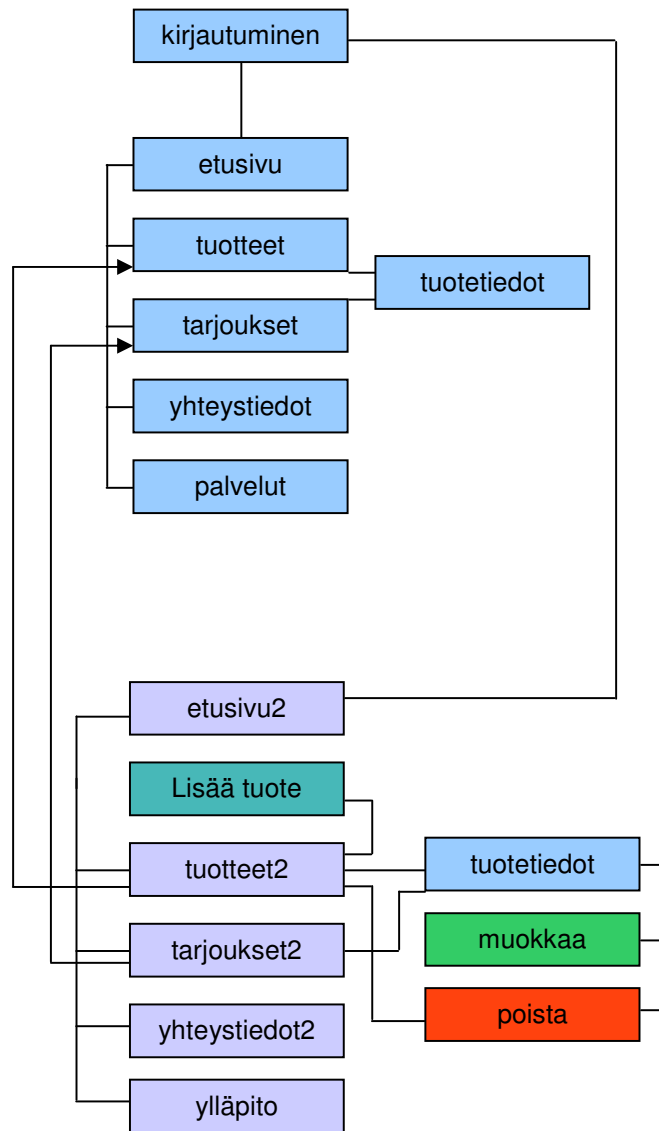
- [www.mysql.com](http://www.mysql.com)
- <http://fi.php.net/mysql>
- <http://forums.mysql.com/>
- <http://www.php-mysql-tutorial.com/>
- <http://www.tizag.com/mysqlTutorial/>



## Liite 2. Tietokantakaavio



### Liite 3. Yleisarkkitehtuuri



## Liite 4. Tiedostot ja niiden väliset suhteet

