

VAASAN AMMATTIKORKEAKOULU

Jarkko Puska

Web-pohjainen ohjeiden hallintajärjestelmä

Liiketalous ja Matkailu
2010

VAASAN AMMATTIKORKEAKOULU
Tietojenkäsittelyn koulutusohjelma

TIIVISTELMÄ

Tekijä	Jarkko Puska
Opinnäytetyön nimi	Web-pohjainen ohjeiden hallintajärjestelmä
Vuosi	2010
Kieli	suomi
Sivumäärä	32 + 1 Liitettä
Ohjaaja	Mika Tamminen

Tämä opinnäytetyö esittelee suunnittelun ja toteutuksen web-ohjelmaan. Opinnäytetyö pitää myös sisällään ohjelmointikielet ja tekniikat, joita käytetään tässä opinnäytetyössä. Työn tavoitteena oli rakentaa toimiva web-pohjainen ohjeiden hallintajärjestelmä toimeksiantajalle.

Tässä opinnäytetyössä käytettiin materiaalia Jakob Nielsenilta, Steve Krugilta sekä php.net- ja w3schools kotisivuilta.

Opinnäytetyö on jaettu kahteen osaan, teoriaan ja käytäntöön. Teoriaosuus pitää sisällään suunnittelun, ohjelmointikielet ja tekniikat. Käytännön osuus taas kertoo miten tämä opinnäytetyö tehtiin.

Opinnäytetyön tuloksena syntyi kaksi Internet-sivua, joilla hallinnoidaan MySQL-tietokantaa. Sivujen suunnittelussa ja toteutuksessa on otettu huomioon toimeksiantajan vaatimukset, sekä käytettävyys.

Asiasanat PHP, HTML, AJAX, SQL, Käytettävyys

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Tietojenkäsittelyn koulutusohjelma

ABSTRACT

Author	Jarkko Puska
Topic	Web-based instruction controlling application
Year	2010
Language	Finnish
Number of Pages	32 + 1 Appendices
Name of Supervisor	Mika Tamminen

This thesis studied the planning and implementation of a web application. The thesis also examined the techniques and programming languages used in the application. The goal for this thesis was to build a working instruction controlling application for an employer.

The materials used in this thesis are from the web pages of Jakob Nielsen, Steve Krug, php.net- and w3schools.

The thesis is divided into two sections: the first part consists of theory and the second part consists of a practical case. The theory includes planning, web techniques and languages and the practical case contains information on how the case study was conducted.

The main results of this work were two websites used for controlling MySQL database. In the planning and implementation of the sites regard has been paid to the requirements of the employer as well as usability.

Keywords PHP, HTML, AJAX, SQL, Usability

SISÄLLYSLUETTELO

TIIVISTELMÄ	2
ABSTRACT	3
SISÄLLYSLUETTELO	4
1. JOHDANTO	6
2. OHJELMOINTIKIELET JA -TEKNIIKAT	7
2.1. HTML	7
2.2. PHP	8
2.3. JavaScript	9
2.4. AJAX	9
2.5. SQL / MySQL	10
3. SUUNNITTELU	11
3.1. Suunnittelun teoria	11
3.2. Toimintatavat ennen ohjelmaa	12
3.2. Vaatimusten määrittely	12
3.3. Ehdotus ohjelmiston toteutuksesta	14
4. OHJELMA	17
4.1 Valitut työkalut	17
4.1.1 Usbwebserver	18
4.1.2 Notepad++	18
4.2 Ohjelman tekeminen käytännössä	19
4.2.1. Suunnittelun toteutus	19
4.2.2. Ohjelman toteutus	19
4.2.3. Ohjelman toiminnan kuvaus	20
4.3. Ohjelma käytössä	25
4.4. Ohjelman kehitysmahdollisuudet	28
4.4.1 Sivuston ulkonäön parantaminen	28
4.4.2 Tietoturvan lisääminen	29
4.4.3 Tietokannan varmuuskopiointi	29
4.4.4 Sähköpostin lähetys ohjelmasta	29
5. YHTEENVETO	31

LÄHDELUETTELO.....	32
LIITE 1	33

1. JOHDANTO

Opinnäytetyöni tavoitteena on perehtyä web-pohjaiseen ohjelmiston suunnitteluun, kehitykseen ja toteutukseen. Käyn tässä opinnäytetyössä läpi teoriaa, suunnittelua ja käytännön tekemistä. Kerron myös minkälaisilla työkaluilla, ohjelmointikielillä ja tekniikoilla tätä opinnäytetyötä on tehty.

Suunnittelussa olen ottanut huomioon toimeksiantajan vaatimukset, sekä käytettävyyden. Käyn suunnittelukappaleessa läpi hieman Jakob Nielsenin ja Steve Krugin käytettävyytlakeja. Ohjelmakappaleessa käyn läpi, miten ohjelma toteutettiin.

Kerron taustaa tämän opinnäytetyön ideasta. Sain toimeksiannon yritykseltä, jossa olen aikaisemmin työskennellyt. Heiltä puuttui kunnollinen työkalu ohjeiden säilyttämiseen ja hakemiseen. He halusivat internet-sivuilleen mahdollisuuden hakea sieltä ohjeita.

Yritys toimii tietotekniikan lähituki- ja korjauspalvelu alalla. Yritys toiminta-alueena on koko Suomi. He työskentelevät usein asiakkaan luona, joten ohjeiden haun pitäisi onnistua asiakkaiden tietokoneelta.

Mietimme aluksi perinteisiä sisällönhallintajärjestelmiä, kuten Joomla! tai Drupalia. Päädyimme tekemään ohjelman ilman näitä ohjelmia, koska emme olleet varmoja, voivatko ne tehdä toimintoja, joita toimeksiantaja haluaa.

2. OHJELMOINTIKIELET JA -TEKNIIKAT

Tässä kappaleessa käydään läpi ohjelman teossa käytettyjä kolmea eri ohjelmointikieltä ja kahta eri tekniikkaa. Kappaleessa käydään myös läpi, mitä nämä ohjelmointikieliset ja -tekniikat ovat, sekä kerrotaan millä tavalla niitä käytettiin opinnäytetyön tekemiseen.

2.1. HTML

HTML tulee sanoista ”HyperText Markup Language”. HTML käytetään kuvaamaan web-sivuja. On huomioitava, että HTML ei ole ohjelmointikieli, vaan merkintäkieli. Merkintäkielessä asetellaan haluttu arvo tagien sisään.(W3schools, 1999–2010)

Taulukossa 1 näkyy HTML-koodin perusrakenne. HTML-sivulla kuuluu olla tageina <HTML>,<HEAD> ja <BODY>. On myös tärkeää muistaa sulkea avainsanat. Esimerkiksi <HTML>-tagi suljetaan </HTML>-tagilla. HTML-sivuja yleensä lähdetään tekemään tästä asetelmasta.

Taulukko 1. HTML-koodi perusrakenne

```
<!DOCTYPE HTML PUBLIC"-//W3C//DTD HTML 4.01 Transitional//EN""http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <title>Otsikko</title>
</head>
<body>
Sivulle tulevat tekstit, kuvat, ym. tulevat tähän.
</body>
</html>
```

HTML-ohjelmointiin löytyy paljon tietoa, mutta alla on listattuna muutama tärkeä asia.

HTML-ohjelmointivinkit

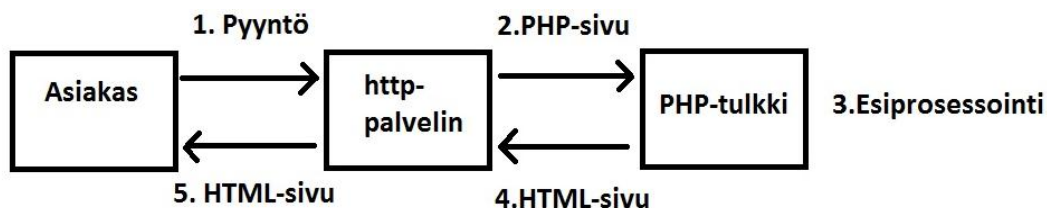
1. HTML-kielessä käytetään isoja ja pieniä kirjaimia(<html> ja <HTML>).
On johdonmukaista käyttää samaa tyyliä kaikissa koodeissa
2. .html ja .htm ovat HTML-tiedoston päätteitä. Tiedoston nimiin ei suositella laitettavaksi välilyöntejä.
3. HTML-komennoista kannattaa tehdä lista, johon kerätään yleisesti käytössä olevat komennot. Näin voidaan vähentää kirjoittamista, kun komento kopioidaan listalta.
4. Huomautuksien lisäämisen avulla voidaan helpottaa ohjelmointikoodin muuttamista.(<!--Huomautuksen teksti tulee tähän -->)
5. -komentoon on hyvä lisätä kuvan mitat, näin kuva voidaan suorittaa selaimessa ilman viivettä.
6. -komennossa olevaa ALT-attribuuttia käytetään vain, kun on esitettävää tietoa olemassa. Tyhjät ALT-määritelmät (ALT=""), voivat ärsyttää tekstipohjaisten selainten käyttäjiä.
7. On hyvä muistaa sulkea annetut komennot (<HTML><TABLE></TABLE></HTML>).

(Pirouz 1999, 3)

Tässä opinnäytetyössä HTML:ää käytettiin sivujen rakenteiden muotoiluun. HTML:n ympärille rakennettiin muista kielistä ja tekniikoista toimiva kokonaisuus.

2.2. PHP

Alunperin PHP muodostui sanoista personal home pages. Nykyään PHP:tä kutsutaan virallisesti ”PHP: Hypertext Preprocessor”. PHP on komentosarjakieli, jota suoritetaan palvelinpuolella. PHP:tä käytetään HTML-koodiin sulautettuna. PHP:ta suoritettaessa mukana tuleva HTML-osa jätetään käsittelemättä ja vain PHP-osa tulkitaan ja suoritetaan. PHP-koodista saatava tulostus yhdistetään suorituksen jälkeen HTML-koodiin ja lähetään käyttäjälle. (Zandra 2001, 20)



Kuvio 1. PHP-koodin käsittely palvelimella vaiheittain.

Kuviossa 1 asiakas suorittaa PHP-sivun selaimen avulla http-palvelimelle (vaihe 1). http-palvelin irrottaa PHP osion HTML:stä ja siirtää sen PHP-tulkille (vaihe 2). PHP-tulkki muokkaa PHP-koodin HTML-muotoon ja palauttaa sen http-palvelimelle (vaiheet 3 ja 4). http-palvelin lisää PHP-tulkilta saamansa HTML-muotoisen koodin alkuperäiseen pyyntöön ja palauttaa sen asiakkaalle (vaihe 5).

PHP-koodia lisätään HTML – koodin mukaan `<?php?>`-tagin avulla. `<?php` osa aloittaa PHP-koodin ja `?>` osa lopettaa sen. Tagin avulla palvelin tietää, mikä osa suoritetaan ja minkä osan selain suorittaa.

PHP:tä käytettiin tässä opinnäytetyössä tietokantayhteyden luomiseen, SQL-kyselyiden suorittamiseen sekä kyselyistä saatujen tulosten lisäämiseksi HTML-sivuun.

2.3. JavaScript

JavaScript on suunniteltu lisäämään HTML-sivujen interaktiivisuutta. JavaScript on komentosarjakieli. JavaScriptiä lisätään suoraan HTML-sivulle. JavaScript on myös tulkkauskieli, joten se voidaan suorittaa ilman kääntäjää. (W3schools 1999–2010)

Opinnäytetyössä JavaScriptiä käytettiin AJAX-tekniikkaan luomiseen, sekä paris- sa kohtaa tiedon tarkistamiseen.

2.4. AJAX

AJAX tulee sanoista ”Asynchronous JavaScript And Xml”. AJAX ei tuo uutta, vaan se yhdistää jo olemassa olevia tekniikoita. Nämä ovat XMLHttpRequest ob-

ject, JavaScript, CSS ja XML. AJAXin avulla voidaan ladata vain osa web-sivusta. Normaalisti koko web-sivu ladataan. AJAX-sovellukset toimivat selaimessa, joten ne ovat alusta riippumattomia. (W3schools 1999–2010)

AJAX:a käytetään opinnäytetyössä ohjaamaan PHP-koodin oikeaan paikkaan. Näin ohjelman tarvitsi päivittää vain osa näkyvästä web-sivusta.

2.5. SQL / MySQL

SQL tulee sanoista ”Structured Query Language” eli rakenteellinen kyselykieli. SQL avulla voidaan hallinnoida tietokantoja. Tietokannasta voidaan hakea, lisätä tai poistaa tietoa SQL-kielen avulla. SQL-kieli on standardoitu, mutta siitä on olemassa erilaisia versioita. Peruskomennot, kuten SELECT, UPDATE, DELETE, INSERT, WHERE ovat tuettuja suurimmissa SQL-ohjelmistoissa. (W3schools 1999–2010)

Taulukko 2. Esimerkki opinnäytetyössä käytetystä SQL-kielestä

```
SELECT ohje.*, hakemisto.Polku
FROM ohje, hakemisto WHERE ohje.HakemistoID
IN (SELECT HakemistoID
FROM hakemisto
WHERE Avainsanat LIKE '%".$uid.%')
AND ohje.HakemistoID = hakemisto.HakemistoID
```

Tässä opinnäytetyössä käytettiin MySQL-tietokantaa metatiedon varastointiin, josta se olisi helppo hakea. MySQL-tietokannassa on kolme taulua. Katso liite 1. Näihin tauluihin sijoitettiin tiedot siitä mitä kyseinen ohje pitää sisällään, missä hakemistossa ohje sijaitsee sekä ohjelman pääkäyttäjien tunnukset ja salasanat.

3. SUUNNITTELU

Suunnittelua on hyvä lähteä tekemään, niistä lähtökohdista mitkä ovat yrityksen tarvitsemat ohjelman ominaisuudet. Tähän ohjelmaan vaadittavat ominaisuudet on kartoitettu kokouksissa yrityksen kanssa. Kokousten perusteella on tehty vaatimusten määrittely, josta kerrotaan kohdassa 3.2 Ohjelman vaatimusten määrittely.

3.1. Suunnittelun teoria

Käytettävyys on tärkeä ominaisuus internet-sivuja tehtäessä. Jakob Nielsen puhuu käytettävyyden viidestä komponentista. Komponenttien avulla voidaan määritellä kuinka, helppokäyttöinen ohjelmisto on. Nämä viisi komponenttia ovat:

- Opittavuus, kuinka helposti sovellusta oppii käyttämään ensimmäisellä kerralla.
- Tehokkuus, kuinka nopeasti oppinut käyttäjä voi suorittaa toimintoja.
- Muistettavuus, kuinka hyvin sovelluksen toiminnot muistetaan tauon jälkeen.
- Virheet, kuinka paljon käyttäjän tekevät virheitä. Kuinka vakavia nämä virheet ovat ja kuinka helposti niistä voidaan palautua.
- Tyytyväisyys, kuinka miellyttävä sovellusta on käyttää.

Steve Krug käsittelee kirjassaan käytettävyyttä. Hän on maininnut kolme keskeistä käytettävyytlakia: ”Älä pakota minua ajattelemaan”, ”Ei sillä ole väliä, monta kertaa minun pitää napsauttaa, kunhan jokainen napsautus on päivänselvä ja yksiselitteinen valinta” ja ”Poista puolet jokaisen sivun sanoista ja sitten vielä puolet jäljelle jääneistä sanoista”.(Krug 2006,11,41)

Krugin ensimmäinen laki: ”Älä pakota minua ajattelemaan” laki tarkoittaa, että web-sivun pitäisi olla niin selkeä, kuin inhimillisesti on mahdollista. Sivun on oltava sellainen, ettei se kaipaa selityksiä. (Krug 2006,11)

Krugin toinen laki: ”Ei sillä ole väliä, montako kertaa minun pitää napsauttaa, kunhan jokainen napsautus on päivänselvä ja yksiselitteinen valinta”. Krug perustelee lakia sillä, että napsautusten määrää ei tule rajoittaa lukumääräisesti, vaan tärkeämpää on, että jokainen napsautus on selkeä, eikä vaadi suurta miettimistä. (Krug 2006:41)

Krugin kolmas laki: ”Poista puolet jokaisen sivun sanoista ja sitten vielä puolet jäljelle jääneistä sanoista”. Krugin mukaan käyttäjä voi saada sanojen määrästä käsityksen, että hänen on luettava koko sivun teksti, jotta hän ymmärtäisi mistä sivulla on kysymys. Krug myös määrittelee, että poistamalla ylimääräiset sanat hyödyllinen sisältö on helpommin löydettävissä. Sivun sisältöä näkyy suurempi osa, koska sivua ei tarvitse vierittää. (Krug 2006,45)

3.2. Toimintatavat ennen ohjelmaa

Ymmärtääkseni yrityksen näkökulmaa selvitin, kuinka heillä toimittiin aikaisemmin ohjeiden suhteen. Selvitin myös miten he olivat hoitaneet ohjeiden ylläpidon ja jakamisen.

Yrityksellä oli kolme eri paikkaa, joihin ohjeita tallennettiin. Ne olivat eri verkkolevyjä. Osaa ohjeista päivitettiin sähköpostin avulla. Eli ohjeeseen tehtiin korjauksia, jotka lähetettiin sähköpostilla käyttäjille. Käyttäjien piti itse tallettaa ohje halumaansa paikkaa talteen. Usein näissä tapauksissa ohje jäi sähköpostiin odottamaan tallennusta.

3.2. Vaatimusten määrittely

Toimeksiantajani kanssa käytyjen kokousten perusteella tein listan vaadituista ominaisuuksista. Listalle kertyi seitsemän tärkeää ominaisuutta. Alla on listattuna vaaditut ominaisuudet tärkeysjärjestyksessä. Tärkein ominaisuus on ylimmäisenä.

1. Ohjelma keskittäisi kaikki ohjeet yhteen paikkaan.
2. Ohjelma olisi helppokäyttöinen.
3. Ohjeita olisi helppo lisätä, muokata ja poistaa.

4. Ohjeita olisi helppo hakea.
5. Ohjelmaa tulisi pystyä käyttämään pääselaimilla(IE,Firefox,Safari,Opera, ym.).
6. Ohjeita saisi lisätä vain ennalta määritellyt henkilöt.
7. Ohjeita pitäisi pystyä hakemaan asiakkaan sekä heidän asiakkaansa mukaan.

Yritykselle oli tärkeää, että ohjeet olisivat yhdessä paikassa. Ohjelmalla pitäisi myös pystyä hakemaan helposti ohjeita, sillä aikaisemmin ohjeet olivat olleet hajallaan, ja uusien ohjeiden ei aina ollut samassa paikassa, kuin vanhat ohjeet.

Yritys myös halusi, että ohjelmaa pystyttäisiin käyttämään monella eri selaimella, joten ohjelmaa on kehitetty myös tästä näkökulmasta. Tämä vaatimus on sen vuoksi, että käyttäjä ei aina pääse omalla tietokoneellaan tähän ohjelmaan. Tällaisia tapauksia on esimerkiksi silloin, kun käyttäjä on asiakkaan luona työtehtävissä ja ei saa omaa tietokonettaan yhdistettyä verkkoon. Näissä tapauksissa voidaan ottaa asiakkaan koneella yhteys ohjelmaan.

Ohjeiden lisääminen ohjelmaan haluttiin rajata, koska näin varmistettiin, että ohjeet tarkistetaan ja laitetaan oikeaan paikkaan hakemistossa. Jos kaikilla olisi mahdollisuus lisätä ohjeita, voisivat samat ohjeet löytyä useasta eri paikasta. Ohjeiden lisäämisen keskittämällä saadaan aikaan se, että ohjeen lisääjä tietää mihin ohjeet kuuluu laittaa.

Yrityksellä on paljon asiakkaita, jotka tuovat mukanaan omia asiakkaita palveltavaksi. Tässä päädyin ratkaisuun, että yrityksen kanssa suoraan asiakkaana oleva yritys sijoitetaan päähakemistoon. Asiakas, joka tulee toisen asiakkaan kautta, sijoitetaan alihakemistoon. Näin pystytään hallinnoimaan ja seuraamaan, kuka on ”suora” asiakas ja kuka asiakkaan kautta tuleva asiakas.

3.3. Ehdotus ohjelmiston toteutuksesta

Ehdotuksessa on otettu huomioon aiemmin määritellyt vaatimukset ohjelman suhteen. Ohjelmisto toimisi Linux-palvelimella, jossa olisi asennettuna Apache http-palvelinohjelmisto. Sivusto olisi tarkoitus toteuttaa HTML, PHP, JavaScript, AJAX ja SQL -tekniikoilla. Näyttöjä tulisi olemaan tässä ohjelmassa kaksi kappaletta. Näytöt voivat olla internet- tai intranet-sivuja. Ensimmäinen oli peruskäyttäjille suunnattu, kun taas toinen olisi vain käyttäjille, joilla on oikeudet lisätä ohjeita tietokantaan. Näytön rakennetta on mietitty käyttäjäystävälliseksi.

Kuva 2. esittää ehdotusta etusivun rakenteeksi. Käyttäjien sivu on jaettu kolmeen osaan. Vasemman puoleissa sarakkeessa ovat kohdat 1, 2 ja 5.

Etusivun rakenne:

- Kohdassa 1 on työpaikan haku. Haku voidaan suorittaa työpaikan mukaan.
- Kohdassa 2 voidaan hakea ohjetietokannasta annetulta hakusanalla. Hakua voidaan myös rajata koskemaan tiettyä kenttää.
- Kohdassa 5 on sähköpostin lähetys pääkäyttäjälle. Käyttäjä voi tätä kautta lähettää viestiä ongelmista suoraan ohjelman pääkäyttäjälle.
- Kesimmäisessä sarakkeessa on kohta 3. Siihen tulostetaan kohdissa 1 tai 2 löytyneet ohjeet.
- Oikean puoleissa sarakkeessa on kohta 4. Sitä käytetään näyttämään kohdassa 3 valittujen ohjeiden metatiedot.

Kuva 2. Ehdotukseni käyttäjälle suunnatun näytön rakenteesta.

Kuvassa 3. on ehdotus pääkäyttäjäsivun rakenteesta. Tässä tapauksessa sivu on jaettu kahteen osaan. Vasen puoli pitää sisällään kohdat 1–4.

Pääkäyttäjän näytön rakenne:

- Kohdassa 1 on määritelty ohjeiden sijoituspaikka.
- Kohdassa 2 on pääkäyttäjän sähköpostiosoite.
- Kohdassa 3 on ohjeen lisäys tietokantaan.
- Kohdassa 4 on ohjeen poisto tietokannasta.
- Oikean puoleiseen sarakkeeseen on kohta 5, joka on tietojen varmuuskopiointia varten.

Kuva 3. Ehdotukseni pääkäyttäjälle suunnatun näytön rakenteesta.

Toimeksiantajani hyväksyi ehdotukseni ohjelman toteutuksesta. Sovimme toimeksiantajani kanssa, että toteutamme tämän siten, että teen ohjelman kotikoneella ja käyn välillä esittelemässä ohjelman edistymistä. Ohjelman valmistuttua siirämme sen toimeksiantajan Linux-palvelimelle. Samalla teen ohjelmalle asennusohjeen, jotta toimeksiantaja voi tarvittaessa tehdä ohjelman asennuksen uudelleen.

4. OHJELMA

Suunnitelman hyväksymisen jälkeen minun oli valittava työkalut ja ryhdyttävä rakentamaan sivustoa suunnitelman mukaan. Minulla oli muutama työkalu, joita mietin olisivatko ne sopivat opinnäytetyötäni varten. Joomla! ja Drupal ohjelmistot olivat vaihtoehtoina.

Joomla! ja Drupal- ohjelmat ovat sisällönhallintajärjestelmiä. Niiden avulla voidaan luoda www-sivut ja niihin sisältöä. Molemmat ohjelmat kuuluvat avoimiin lähdekoodin ohjelmiin. Avoin lähdekoodi tarkoittaa, että ohjelmaa saa kuka tahansa käyttää.

Joomla! ja Drupal ohjelmia en ottanut opinnäytetyöhöni työkaluiksi, koska tutkiessani ohjelmien tietoja ymmärsin, että ohjelmat eivät olisi ehkä niin helposti muokattavaksi tähän työhön liittyen. Normaaliin www-sivujen tekemiseen ne sopivat hyvin, mutta koska tässä opinnäytetyössä käytetään montaa eri tekniikkaa ja kieltä, ohjelmia voi olla vaikea saada toimimaan oikein. Kohdassa 4.1 Valitut työkalut kerron miksi valitsin toiset ohjelmat opinnäytetyöhöni.

4.1 Valitut työkalut

Pystyäkseni tekemään kotona opinnäytetyötäni minun oli löydettävä oikeanlaiset työkalut sitä varten. Löysin kaksi ohjelmaa, jotka voi asentaa muistitikulle ja suorittaa ne sieltä ilman, että tietokoneelle asennetaan ohjelmaa. Ne osoittautuivat hyödyllisiksi, koska näin en ollut sidottuna yhteen tietokoneeseen vaan ohjelma ja työkalut kulkivat aina mukani muistitikulla. Lisäksi ne tarjosivat mahdollisuuden vapaampaa koodin muokkaamiseen, mitä Joomla! ja Drupal olisivat pystyneet.

Suurin ongelman näissä ohjelmissa on, että työstä täytyy aina erikseen ottaa varmuuskopio, koska muistitikku voi helposti hajota ja samalla tuhota opinnäytetyöni.

4.1.1 Usbwebserver

Usbwebserver on ilmainen webpalvelinohjelmisto. Tämä ohjelmisto pitää sisällään Apache webpalvelinohjelmiston, MySQL-tietokannan, PHP5 ja PHPMyAdmin-ohjelmiston. Ohjelma voidaan helposti asentaa muistitikulle, josta se voidaan suorittaa ilman tietokoneelle asentamista.

Opinnäytetyössäni käytin usbwebserveriä testatakseni ohjelmaa ja tietokannan ylläpitoon. Ohjelmasta löytyy tarkemmin tietoja osoitteesta:

<URL:<http://www.usbwebserver.com/>>.

4.1.2 Notepad++

Notepad++ on ilmainen lähdekoodi editori. Notepad++ korostaa värein koodin eriosia, jolloin saadaan koodista helpommin luettavaa. Värien avulla voidaan korostaa HTML, PHP, JavaScript, SQL ja monia muita ohjelmointikieliä. Kuvassa 4 on esimerkki Notepad++ -ohjelmiston värien korostamisesta.

```
<td width="20%" align="center">
  <p><b>Ohjeiden haku hakemistojen mukaan</b></p>
  <!-- Määritellään pääkansiot jotka ovat tietokanna
  <div style="background-color:#A8A8A8">
    Päähakemisto<br />
    <select name="tyopaikka" id="tyopaikka" onchan
    <option value="0">Valitse ohjekanta</option>
    <?php
    // Luodaan SQL-yhteys
    $tk = mysql_connect("localhost","root","apache
    // Varmistetaan yhteyden onnistuminen
    if(!$tk)
    {
      die("TietoKanta-yhteys ei onnistu: ". mysql
```

Kuva 4. Notepad++ ohjelman värinkorostus esimerkki.

Opinnäytetyössäni käytin Notepad++ koodin korostamiseen, koska näin sain paremmin selville missä kohtaa ohjelmointi virhe voisi olla.

<URL:<http://notepad-plus-plus.org/>>.

4.2 Ohjelman tekeminen käytännössä

Ohjelmaa lähdin tekemään toimeksiantajalle esittelemäni suunnitelman mukaan. Sovimme toimeksiantajan kanssa, että teen työtä kotona ja kun saan uuden ominaisuuden valmiiksi tulen esittelemään sitä.

Ohjelman tekeminen käytännössä toimi siten, että tein pienen ominaisuuden toimimaan ja tämän jälkeen se liitettiin isompaan kokonaisuuteen. Tain esimerkiksi SQL-koodin, joka hakee kaikki hakemistot tietokannasta. Saatuani SQL-koodin toimimaan lisäsin sen PHP-sivulle ja testasin, että se toimii isommassakin kokonaisuudessa.

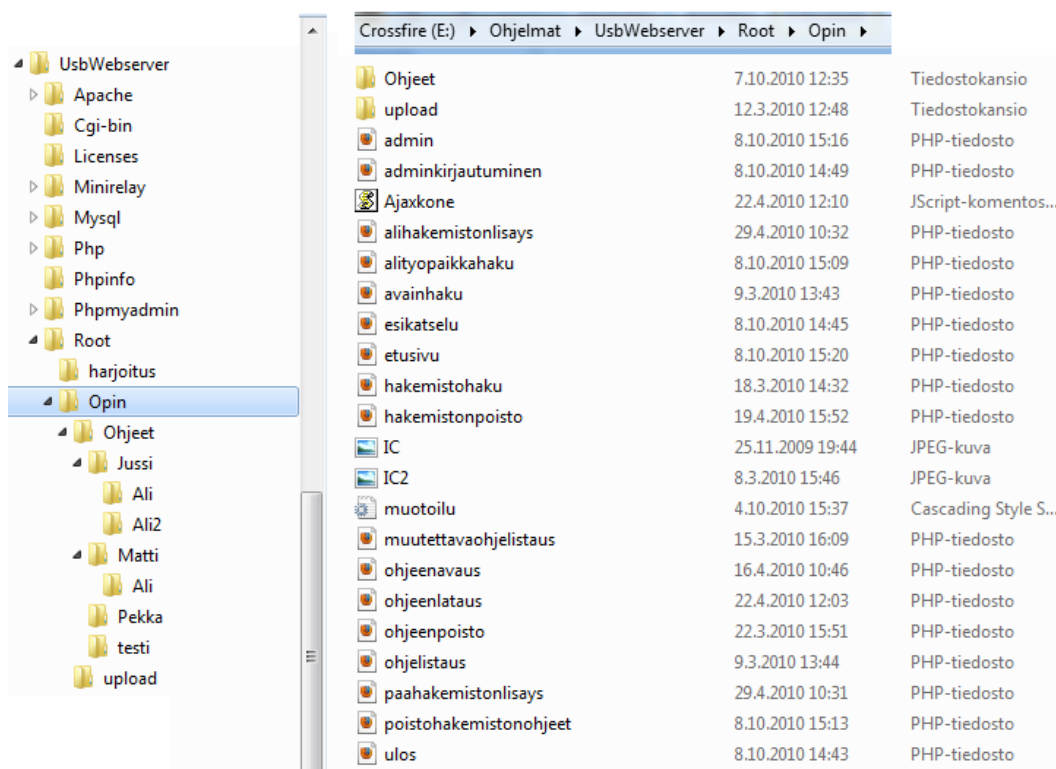
4.2.1. Suunnittelun toteutus

Opinnäytetyöni suunnittelua toteutin kahdessa vaiheessa. Ensimmäiseen vaiheeseen kuului toimeksiantajan vaatimusten sisällyttäminen ohjelmaan, käytettävyyden huomioon ottamista, rajoituksia, joita on otettava huomioon ohjelmoinnissa sekä ohjelman ulkoasun rakenteen päättäminen.

Toinen vaihe kuului oleellisesti ohjelman ohjelmoinnin yhteyteen. Ohjelmointi ei mennyt alkuperäisen suunnitelman mukaan, vaan sitä oli muutettava työn edetessä, koska vastaan tuli ongelmia, joita en osannut ottaa huomioon alkuperäisessä suunnitelmassa.

4.2.2. Ohjelman toteutus

Lähdin toteuttamaan suunnitelmaa rakentamalla HTML-koodilla etusivun pohjan. Pohjan myötä oli helppo lähteä kehittämään vaadittuja ominaisuuksia. Seuraavaksi loin MySQL-tietokannan, joka pitää sisällään hakemisto- ja ohje -taulut. Tässä vaiheessa tein samaan kansioon missä PHP-tiedostot olivat ohjeet kansion. Tähän kansioon laitetaan hakemistotaulussa olevat hakemistot sekä ohjetaulussa olevat ohjeet. Kuvassa 5 näkyy miten ohjelman hakemiston rakenteen. Ohjeet-kansioon sijoitetaan kaikki ohjelman kautta lisätyt ohjeet.



Kuva 5. Näkymä ohjelman hakemisto rakenteesta.

Seuraavana toimenpiteenä tein ohjelmoinnin. Tein ohjelmoinnin siten, että yksi ominaisuus kerralla valmiiksi ja sen jälkeen siirryin eteenpäin. Ohjelmoinnin valmiiksi tultua seuraava vaihe oli testaus. Testauksessa lisättiin ja poistettiin hakemistoja ja ohjeita eritavoilla, jotta saataisiin virhetilanteita aikaiseksi. Viimeisessä vaiheessa muokattiin näyttöjen ulkoasua.

4.2.3. Ohjelman toiminnan kuvaus

Tässä kappaleessa kuvataan miten eri ohjelmointikielet ja –tekniikat saadaan toimimaan keskenään.

Ohjelma on tarkoitettu toimimaan Linux-pohjaisella Apache webpalvelimella. Selaimen avulla voidaan hallita ohjetietokantaa. Selain voi suorittaa hakuja tietokannasta. Hakuja voidaan suorittaa hakemiston tai hakusanan mukaan.

Haut tapahtuvat siten, että käyttäjä valitsee hakemiston mistä haetaan ohjeet. HTML:ssä oleva AJAX-tekniikkakoodi (Katso taulukot 3 ja 4) lähettää pyynnön PHP-sivulle.

Taulukko 3. HTML:ssä oleva JavaScript-koodi, joka lähettää tiedon JavaScript-tiedostolle.

```
<input type="button" value="Hae ohjeet"
on-
click="if((window.document.getElementById('alikansiot')
.value == '0')
|| (document.getElementById('alikansiot').value == ''))
{
var element = document.getElementById('esikatselu');
while (element.firstChild) {ele-
ment.removeChild(element.firstChild);}
ohjehaku(document.getElementById('tyopaikka').value);
}
else
{
var element = document.getElementById('esikatselu');
while (element.firstChild) {ele-
ment.removeChild(element.firstChild);}
ohjehaku(document.getElementById('alikansiot').value);
}">
```

Taulukko 4. JavaScript-funktio, joka lähettää saadun tiedon PHP-sivulle.

```
function ohjehaku(sId)
{
// Luodaan ajax-kone
xmlhttp = luoAjaxKone();
if(xmlhttp == null)
{
```

```

        alert("Selaimesi ei tue Ajax-tekniikka.");
        return;
    }
    // Ajax-kone ok
    var url = "ohjelistaus.php";
    url = url + "?uid=" + sId;
    // Tilanhallintafunktion määrittely xmlhttp-
objectille
    xmlhttp.onreadystatechange = stateChanged;
    xmlhttp.open("GET", url, true);
    xmlhttp.send(null);
}

```

Taulukoissa 5 ja 6 PHP-koodi tekee kyselyn SQL:llä tietokantaan ja muokkaa vastauksen tulostettavaan muotoon.

Taulukko 5. PHP-koodi ottaa yhteyden SQL-palvelimeen ja toteuttaa siellä kyselyn.

```

$uid = $_GET["uid"];
$tk = mysql_connect("localhost","root","apache");
if(!$tk)
{
    die("Tietokanta-yhteys ei onnistu: ".
mysql_error());
}
mysql_select_db("koulu", $tk);
$sql="SELECT ohje.*, hakemisto.Polku FROM ohje, hake-
misto WHERE ohje.HakemistoID IN (SELECT HakemistoID
FROM hakemisto WHERE HakemistoID='".$.$uid."' ) AND oh-
je.HakemistoID = hakemisto.HakemistoID";
$tj = mysql_query($sql);
$rivejä = mysql_num_rows($tj);

```

Taulukko 6. PHP-koodi tekee saadusta tuloksesta HTML-muotoista koodia ja palauttaa sen JavaScript-tiedostolle.

```

if($rivejä > 0)
{
    echo      "<table      border='1'      align='center'
class='black'>
    <tr>
    <th>Valitse</th>
    <th>Nimi</th>
    </tr>";
    $cbnro = 1;
    while($rivi = mysql_fetch_array($tj))
    {
        $cbname = "checkbox_";
        $checkboxid = $cbname ."". $cbnro;
        $testi = $rivi["OhjeID"];
        echo "<tr>";
        echo  "<td  class='grey'><input  type='checkbox'
id='".$checkboxid."'
        value='".$rivi["OhjeID"]."

```

Taulukko 7. PHP-koodi pystyy lisäämää myös JavaScript-koodia.

```

'onclick=\"if (document.getElementById
('".$checkboxid."').checked == false)
{
    var      oparent      =      docu-
ment.getElementById('esikatselu');
    var  ochild  =  document.getElementById('". $ri-
vi["OhjeID"] . "');
    oparent.removeChild(ochild);
}

```

Taulukko 8. Taulukossa 6 aloitettu PHP-koodin loppuosa.

```

else
{
    esikatselu2(this.value);
}\\" ></td>";
echo"<td class='grey'><a
    id='a_".\$checkboxid."'
href='".\$rivi["Polku"]."/".\$rivi["Nimi"]."'
    target='_blank'>". \$rivi["Nimi"] . "</a></td>";
echo "</tr>";
\$cbnro++;
}
echo "</table>";
echo "<br />";
echo  "<input  type='button'  value='Avaa  ohje'  on-
click='avaaohje(" . (\$cbnro-1). ")' >";}

```

Taulukot 9 ja 10 PHP-sivu palauttaa AJAXille tulostetun vastauksen. AJAX-tekniikka sijoittaa sen HTML-sivun keskimmaiseen sarakkeeseen.

Taulukko 9. JavaScript-funktio sijoittaa PHP:ltä saadun vastauksen HTML-sivulle.

```

function stateChanged()
{
    // Varmistetaan, että palvelupyyntö ja -vastaus on
    suorittu kokonaisuudessaan
    if(xmlhttp.readyState == 4)
    {
        docu-
ment.getElementById("Ohjelistaus").
        innerHTML= xmlhttp.responseText;
    }
}

```



```
}
```

Taulukko 10. HTML-sivussa paikka, jonne JavaScript-tiedosto sijoittaa vastauksen.

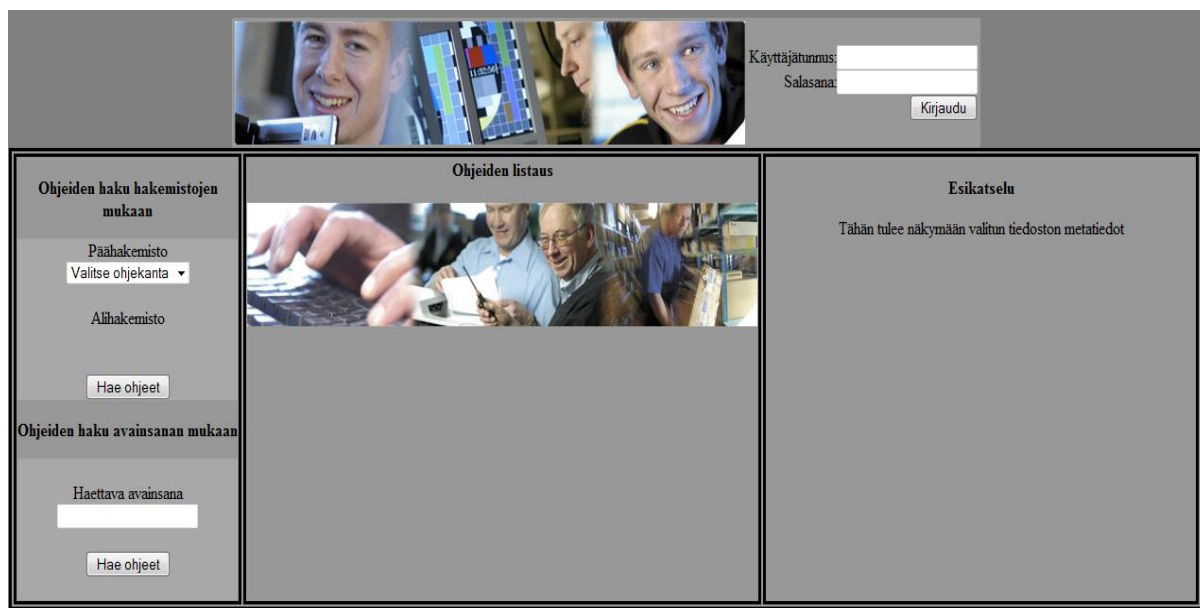
```
<b>Ohjeiden listaus</b>  
<div id="Ohjelistaus"></div>
```

AJAX-tekniikan avulla voidaan päivittää vain osa sivusta. Normaalisti pitäisi päivittää koko sivu ja se aiheuttaa ylimääräistä verkkoliikennettä. Ilman AJAXsia tietoja päivitettäessä valinnat katoavat näytön näkymästä. Koko sivua päivittäessä ohjelma ei muista mitä valintoja on tehty aikaisemmin. Esimerkiksi ilman AJAXsia päähakemisto ei enää näytä hakemiston nimeä, kun taas AJAXsin kanssa nimi pysyy ”muistissa”.

4.3. Ohjelma käytössä

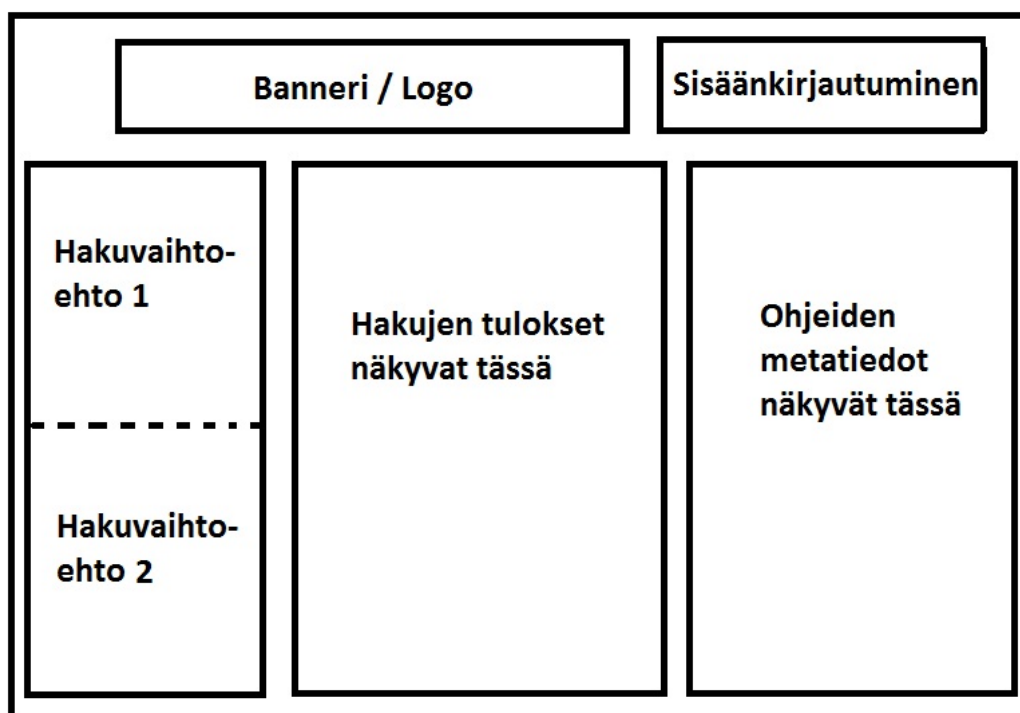
Tässä kappaleessa kerrotaan, miten ohjelma toimii graafisesti. Kummastakin sivusta on näkymä siitä miltä näyttö näyttää selaimessa ja minkälainen on näyttöjen rakenne. Kappaleessa kerrotaan, myös minkälaisia toimintoja voidaan kyseillä näytöllä suorittaa.

Kuvassa 6 näkyy ohjelman etusivu. Tältä näytöltä voidaan suorittaa hakuja, nähdä hakujen tulokset, tutkia ohjeiden metatietoja ja kirjautua pääkäyttäjäksi näytölle.



Kuva 6. Näkymä ohjelman etusivusta.

Kuvassa 7 näkyy kuinka näyttö on jaettu osiin. Jokaiselle osiolla on annettu selkeä tehtävä, joka parantaa käytettävyyttä.




Kuva 7. Etusivun rakennekuva.

Kuvassa 8. Esitellään pääkäyttäjien näyttö. Näytöltä voidaan hallita hakemistojen luontia ja poistamista, sekä ohjeiden lisäämistä, päivittämistä ja poistoa. Hakemistoja luodessa voidaan valita tehdäänkö hakemistosta päähakemisto vai alihakemisto. Näytöltä löytyy uloskirjautumiseen mahdollisuus. Näyttöä ei pääse näkemään, muuten kuin etusivun kautta kirjautumalla sisään.

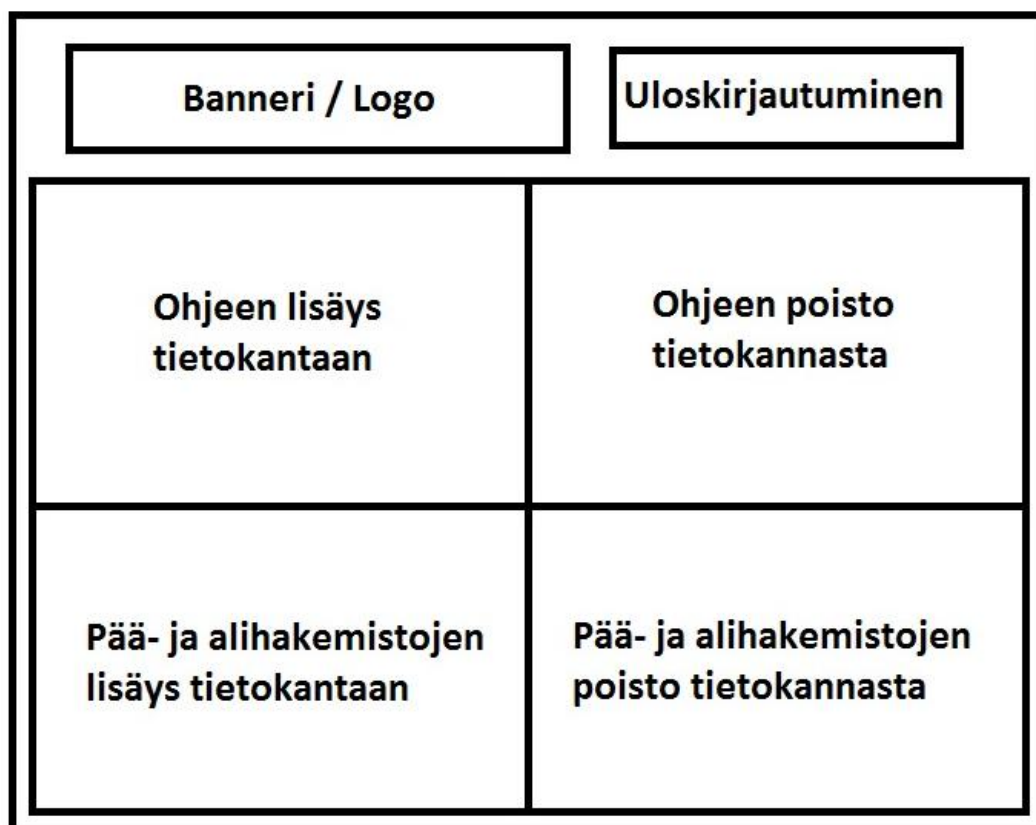


[Kirjaudu ulos](#)

Ohjeenlisäys ja päivitys		Ohjeen poisto	
Valitse kansio mihin ohjemuutokset kohdistetaan	Valitse hakemisto ▾	Valitse mistä kansista ohje poistetaan	Valitse hakemisto ▾
Ylikirjoitus	<input type="checkbox"/> Ylikirjoitetaanko ohje, jos saman niminen tiedosto löytyy?		
<input checked="" type="radio"/> Lisää uusi ohje	Tiedosto: <input type="text"/> Selaa...		
<input type="radio"/> Päivitä olemassa olevaan ohjeeseen Avainsanat tai ohjeen kuvaus			
Lisää avainsanat	Avainsanat: <input type="text"/>		
Kirjoita lyhyesti mitä ohje pitää sisällään	Ohjeen kuvaus: <input type="text"/>		
<input type="button" value="Lisää ohjetietokantaan"/>			
Päähakemiston lisäys		Hakemiston poisto	
Anna päähakemiston nimi:		Valitse poistettava hakemisto	
<input type="text"/>		Valitse hakemisto ▾	
<input type="button" value="Lisää päähakemisto"/>			
Alihakemiston lisäys			
Valitse mihin uusi alihakemisto luodaan			
Valitse ▾			
Anna alihakemiston nimi:			
<input type="text"/>			
<input type="button" value="Lisää alihakemisto"/>			

Kuva 8. Näkymä pääkäyttäjien näytöstä.

Kuvassa 9 näkyy, että näyttö on jaettu kuuteen eri osaan. Kaksi ylimmäistä osaa sisältävät bannerin ja linkin, jonka kautta ulos kirjaudutaan ohjelmasta. Neljä seuraavaa osaa on sijoitettu siten, että eniten käytetty olisi vasemmalla reunalla ylhäällä. Toiseksi lisäys *toiminnot* on keskitetty vasemmalla puolelle, kun taas *poistot* ovat oikealla puolella.



Kuva 9. Pääkäyttäjänäytön rakennekuva.

4.4. Ohjelman kehitysmahdollisuudet

Ohjelmassa on monta eri kehitysmahdollisuutta. En ehtinyt toteuttaa kaikkia, niitä suunnitelmia, joita esittelin suunnittelu vaiheessa. Kehitysmahdollisuuksista tärkeimpänä pidän tietoturvan lisääminen.

Esittelen näissä luvuissa eri kehitysmahdollisuuksia. On muitakin kehitysmahdollisuuksia mitä tässä esittelen, mutta tärkeimmät ovat tähän opinnäytetyöhön kirjattu.

4.4.1 Sivuston ulkonäön parantaminen

Sivuston pohja on suunniteltu toimivaksi, mutta tällä hetkellä ulkonäkö kaipaa parannusta. Sivuston voisi suunnitella sopimaan yrityksen nykyiseen brandiin. Toteutukseen ei tarvita- kuin HTML- ja CSS-koodin muokkaamista yrityksen brandin mukaiseksi. Opinnäytetyössä on jo valmiina CSS-tiedosto.

CSS tulee Englannin kielisistä sanoista ”Cascading Style Sheets”. CSS:ää käytetään www-sivujen ulkoasujen määrittelyyn. CSS on korvannut HTML-koodin sivustojen muotoilussa.

4.4.2 Tietoturvan lisääminen

Ohjelman tietoturva on vielä keskeneräinen. Käyttäjätunnusten varmistusta ja käyttöä tulisi vielä kehittää, sekä ohjelmaa ei ole suojattu esimerkiksi ”SQL injektio” tai ”Cross-site scripting” kaltaisia hyökkäyksiä vastaan.

SQL injektiossa yritetään syöttää omia SQL-komentoja sivujen omien tekstikenttien kautta. Esimerkiksi kirjautumislomakkeet ovat yleisiä kohteita. Näitä voidaan estää tekemällä SQL-kyselyitä talletettuina proseduureina.

Cross-site Scripting on hieman erilainen hyökkäystapa. Toiselta nimeltään XSS. Sitä välitetään web-lomakkeiden avulla. Lomakkeen avulla voidaan asentaa palvelimelle komentosarjakielisen haittakoodi. Tämän jälkeen tulevat käyttäjät voidaan ohjata hyökkääjän serverille. Nämä ovat suurimmat tietoturvaongelmat tässä ohjelmassa.

4.4.3 Tietokannan varmuuskopiointi

Tietokannan varmuuskopiointi oli mukana alkuperäisissä suunnitelmissa, mutten ehtinyt toteuttaa tätä, koska aikaa ei ollut tarpeeksi. Tällä hetkellä varmuuskopiointi toimii manuaalisesti, koska tätä kehitysmahdollisuutta ei ehditty toteuttaa.

Ideana oli, että ohjelma ottaisi tietokannasta SQL koosteen, jokaisena arkipäivä kello 20 jälkeen. SQL kooste on teksti tiedosto, joka pitää sisällään tietokannan SQL-lauseet, joilla voidaan luoda tietokanta uudestaan. SQL koostetta varten on oma toiminto, mutta ajastettuna tätä toimintaa en löytänyt.

4.4.4 Sähköpostin lähetys ohjelmasta

Ohjelman piti myös sisältää sähköpostin lähetyksen admille, mutta toteutukseen asti tämä idea ei ehtinyt aikataulullisista syistä. Toiminta-ajatuksena tässä oli se,

että ohjelmasta voisi lähettää suoraan viestiä pääkäyttäjille ongelmatilanteesta. Näin pystyttäisiin puuttumaan ongelmiin mahdollisimman nopeasti.

Tämä idea pystytään toteuttamaan PHP-koodilla ja palvelimelle asennetulla sähköpostipalvelin-ohjelmiston avulla. PHP.net sivulta löytyy ohjeita mail-funktion käyttöön

5. YHTEENVETO

Onnistuin täyttämään minulle määritellyt vaatimukset ohjelman suhteen. Ohjelmaa pystytään käyttämään, vaikka muutamaa ominaisuutta ei ehditty tehdä. Ohjelman perustoiminta ohjeiden hakeminen, lisääminen ja poistaminen toimivat, joten ohjelma voidaan ottaa käyttöön. Mielestäni paras onnistumiseni on, että sain ohjelman perustoiminnot toimimaan kunnolla.

Opin paljon tämän työn tekemisestä. Mielestäni tärkeimpänä asiana opin, että miten lähdetään selvittämään ohjelmointivirheitä. Välillä mietin pitkän aikaa, kuinka voisin ongelman ratkaista, mutta vastausta ei vain tahtonut löytyä. Seuraavana päivänä asia kuitenkin ratkesi, kun kokeilin uutta ideaa. Näin kävi useasti. Toinen asia, jonka opin on, että ymmärrän nyt paljon paremmin HTML:n, AJAXin, PHP:n, JavaScriptin ja SQL:n yhteistyöstä.

Kokonaisarviona prosessista antaisin hyvän. Ohjelman perustoiminnot toimivat ja ohjelmaa on helppo käyttää. Ainoat miinukset asiasta tulevat ajankäytöstä ja siitä, että en ehtinyt tehdä kaikkia ominaisuuksia, joista keskustelimme toimeksiantajan kanssa.

Suunnittelun osalta teoria oli helppo liittää käytännön tekemiseen. Huomioon piti ottaa pääasiassa käytettävyys. Toteutuksessa teorian ja käytännön yhdistäminen oli hankalampaa. Esimerkiksi ohjeita, joita löytyy Internetissä ei voi suoraan käyttää. Ne pitää muokata sopiviksi omaan ohjelmaan.

LÄHDELUETTELO

Painetut teokset

Pirouz, Raymond (1999). HTML Expert. 1Painos. Helsinki: IT Press.

Zandstra, Matt (2001). PHP Trainer Kit.1 Painos. Helsinki: Oy Edita Ab

Krug,Steve (2006). Älä pakota minua ajattelemaan. 1 Painos. Helsinki:Readme.fi

Internetsivut

Nielsen, Jakob 2003. Usability 101: Introduction to Usability. [viitattu 14.11.2010]. Saatavilla Internetissä: <URL:http://www.useit.com/alertbox/20030825.html>.

W3schools, 1999-2010. HTML Introduction. [viitattu:14.11.2010]. Saatavilla Internetissä: <URL:http://www.w3schools.com/html/html_intro.asp>.

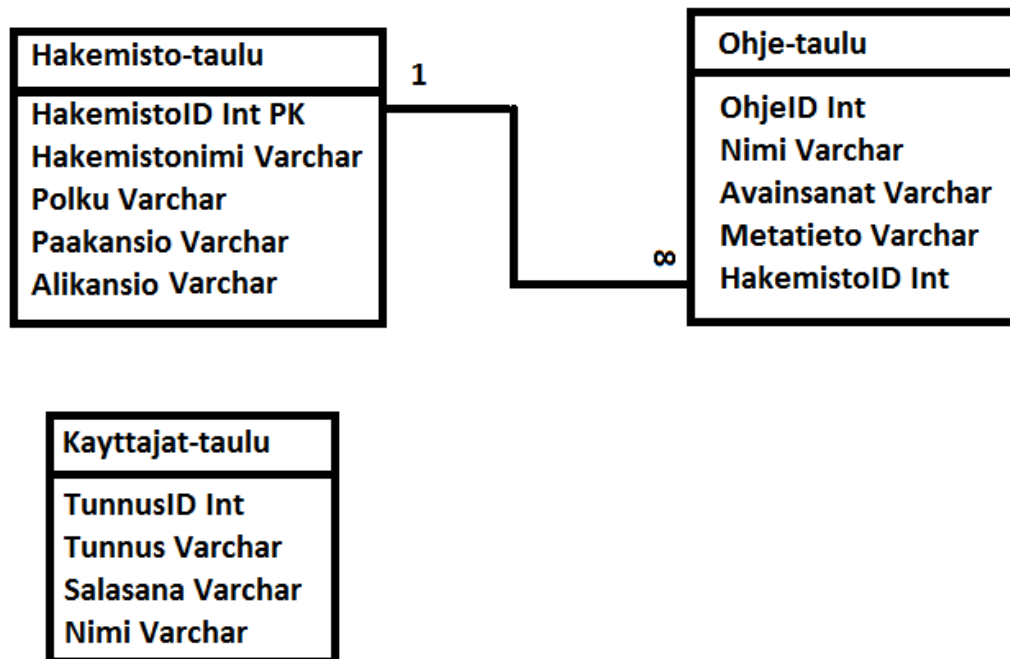
W3schools, 1999-2010. JavaScript Introduction. [viitattu:14.11.2010]. Saatavilla Internetissä: <URL: http://www.w3schools.com/js/js_intro.asp>.

W3schools, 1999-2010. AJAX Introduction. [viitattu:14.11.2010]. Saatavilla Internetissä: <URL: http://www.w3schools.com/ajax/ajax_intro.asp>.

W3schools, 1999-2010. SQL Introduction. [viitattu:14.11.2010]. Saatavilla Internetissä: <URL: http://www.w3schools.com/sql/sql_intro.asp>.

PHP.NET, 2010. mail function. [viitattu:14.11.2010]. Saatavilla Internetissä: <URL: http://php.net/manual/en/function.mail.php>.

LIITE 1



Kuva 10. MySQL-tietokannan taulut