

Jani Puukari ja Hannu Pikkarainen

DYNAAMISET KOTISIVUT VANHAIN YSTÄVÄT RY:LLE

Opinnäytetyö
Kajaanin ammattikorkeakoulu
Luonnontieteiden ala
Tietojenkäsittelyn koulutusohjelma
Syksy 2008



**Kajaanin
ammattikorkeakoulu**

OPINNÄYTETYÖ TIIVISTELMÄ

Koulutusala Luonnontieteiden ala	Koulutusohjelma Tietojenkäsittelyn koulutusohjelma
Tekijä(t) Jani Puukari ja Hannu Pikkarainen	
Työn nimi Dynaamiset kotisivut Vanhain Ystävät ry:lle	
Vaihtoehdot ammattiopinnot Järjestelmän ylläpito	Ohjaaja(t) Liisa Mikkonen Toimeksiantaja Vanhain ystävät ry.
Aika Syksy 2008	Sivumäärä ja liitteet 65 + 9
<p>Tämän opinnäytetyön aiheena oli luoda dynaamiset kotisivut Kajaanissa sijaitsevalle Vanhain ystävät ry:lle. Dynaamisuudella varmistetaan tietojen pysyminen ajan tasalla. Kirjallisessa osiossa käydään läpi eri skriptikieliä kuten PHP, ja niitä vertaillaan toisiinsa. Kirjallinen osio käsittää myös tietoa tietokannoista. Käsiteltäviä tietokantoja ovat MySQL, Oracle ja SQLite. Työ tehtiin Vanhain ystävät ry:n toimeksiannosta. Lisäksi tarkoituksena oli tehdä yhdistyksestä kertova esite. Opinnäytteen tekeminen alkoi jo keväällä 2008 ja se valmistui saman vuoden syksyllä.</p> <p>Aluksi kerrotaan yhdistyksen historiasta ja esitetään mahdollisimman hyvin yhdistyksen toimintamuotoja. Sen jälkeen käsitellään kotisivujen suunnitteluun ja ylläpitoon liittyviä olennaisia asioita. Näihin kuuluvat esimerkiksi sivujen visuaalinen ilme ja päivittäminen. Lopuksi kerrotaan sivujen käytettävyydestä, jossa on otettu huomioon myös sivujen kohderyhmä. Sivuston toimivuutta testattiin eri Internet-selaimilla ja eri ikäisillä käyttäjillä.</p> <p>Työhön kuuluu myös vaatimusmäärittely, joka määrittelee toimeksiantajan tarpeet sivustolle. Usein vaatimukset ovat epäselviä, mutta tässä työssä vaatimukset toimeksiantajan puolelta ovat selkeitä. Toimeksiantaja halusi tuoda toimintaansa tunnetuksi, koska yhdistyksellä ei ole aikaisempia kotisivuja. Tarkoituksena on, että sivustosta kiinnostuneet löytävät haluamansa asiat nopeasti ja helposti.</p> <p>Sivustosta pyrittiin tekemään helppokäyttöinen ja selkeä, koska käyttäjäryhmä on vanhuksia. Myös ylläpitosivusta tehtiin mahdollisimman yksinkertainen, koska kenelläkään yhdistyksen jäsenistä ei ole yhtään kokemusta kotisivujen tekemisestä ja päivittämisestä.</p> <p>Työ oli tekijöille haasteellinen, sillä dynaamisen sivuston toteuttamisesta ei ollut minkäänlaista kokemusta. Tämän vuoksi työn asioiden paneutumiseen meni paljon aikaa. Toimeksiantaja antoi tekijöille kuitenkin vapaat kädet työn toteutuksen suhteen.</p>	
Kieli	Suomi
Asiasanat	dynaamiset kotisivut, php-ohjelmointi, tietokannat
Säilytyspaikka	<input checked="" type="checkbox"/> Kajaanin ammattikorkeakoulun Kaktus-tietokanta <input checked="" type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto

School Business	Degree Programme Data Processing
Author(s) Jani Puukari ja Hannu Pikkarainen	
Title Dynamic Homepages for Vanhain ystävät ry.	
Optional Professional Studies System Support	Instructor(s) Liisa Mikkonen
	Commissioned by Vanhain ystävät ry.
Date Fall 2008	Total Number of Pages and Appendices 65+9
<p>The aim of this thesis was to create dynamic Internet pages for Kajaani-based Vanhain ystävät ry. Having dynamic Internet pages ensures that information is constantly up-to-date. In the theory part, different script languages, such as PHP, are introduced and compared to each other. The theory part also comprises databases, including MySQL, Oracle and SQLite. This project was commissioned by Vanhain ystävät ry. and it also entailed the making of a brochure for the association. The project was initiated in spring 2008 and it was completed in fall 2008.</p> <p>First, the history of the association is discussed and the various activities of the association are presented as well as possible. Next, the essential topics on designing and maintaining Internet pages are explored. These included, for instance, the visual look of the pages and updating. Finally, the usability of the pages is discussed considering the target group of the pages as well. The functionality of the pages was tested in different browsers and by users of various ages.</p> <p>The project also included a definition of competence, which defines the requirements of the client regarding the Internet pages. In many cases, the requirements are not clear, but in this case the client had a clear view. The association wanted to draw attention to the activities of the association, because no previous Internet pages existed. The purpose is that those who are interested in the pages find what they are looking for fast and easily.</p> <p>The Internet pages were intended to be easy to use and straightforward, as the target group was the elderly. Also, the maintenance page was designed to be as simple as possible, because none of the members of the association had previous experience in creating and updating Internet pages.</p> <p>The project presented many challenges for the writers were not familiar with creating dynamic Internet pages. On this account, a lot of time was dedicated to study the themes of the project.</p>	
Language of Thesis	Finnish
Keywords	dynamic homepages, php-programming, database
Deposited at	<input checked="" type="checkbox"/> Kaktus Database at Kajaani University of Applied Sciences <input checked="" type="checkbox"/> Library of Kajaani University of Applied Sciences

ALKUSANAT

Haluamme kiittää Vanhain ystävät ry:tä ja etenkin puheenjohtaja Marja Leskistä siitä, että saimme opinnäytetyön aiheen heidän kauttaan. Kiitokset myös ohjaaja Liisa Mikkoselle hyvästä ja ammattitaitoisesta ohjauksesta.

Kajaanissa 24. Marraskuuta 2008.

SISÄLLYS

SYMBOLILUETTELO	1
1 JOHDANTO	2
2 VANHAIN YSTÄVÄT RY.....	3
3 DYNAAMISET WWW-SIVUT	5
4 VAATIMUSMÄÄRITTELY.....	7
5 KÄYTTÄJÄVAATIMUKSET OHJAAVAT SIVUSTON TOTEUTUSTA.....	10
5.1 Sivuston rakenteen suunnittelu.....	10
5.1.1 Kotisivun suunnittelu.....	10
5.1.2 Kenelle?.....	11
5.1.3 Sivuston perustavoitteet.....	12
5.1.4 Kohderyhmän rajaus	13
5.1.5 Web-sivujen luokittelu	13
5.1.6 Sivuston rakentaminen.....	14
5.2 Sivuston visuaalinen ilme.....	20
5.2.1 Toimivan tekstin rakenne	20
5.2.2 Värien käyttö	22
5.2.3 Taustakuvat.....	23
5.3 Lomakkeet ja niiden perusidea.....	23
5.3.1 Lomakkeen kentät ja yleisrakenne.....	24
5.3.2 Liittymislomake.....	25
5.4 Hallintasivu	25
6 TESTAUS.....	31
6.1 Käytettävyytestaus.....	31
6.2 Sivuston testaus.....	32
7 TYÖVÄLINEET	35
7.1 Macromedia Dreamweaver	35
7.2 PHP	36
7.2.1 Miksi valita PHP?.....	37

7.2.2 PHP:n toiminta ja skriptit.....	38
7.2.3 PHP ja MySQL	39
7.2.4 PHP6	41
7.2.5 PHP ja turvallisuus	42
7.3 ASP	42
7.4 Java Server Pages	44
7.5 CGI-ohjelmat	45
7.6 ColdFusion	47
7.7 Ruby on Rails	47
7.8 MySQL	48
7.9 Oracle	50
7.10 SQLite	51
8 KÄYTETTÄVYYS.....	54
9 POHDINTA.....	58
9.1 Käytännön osio	58
9.2 Kirjallinen osio	59
9.3 Yhteenvedo.....	60
LÄHTEET.....	62

LIITTEET

SYMBOLILUETTELO

Apache	Avoimeen lähdekoodiin perustuva palvelinohjelmisto
ASP	Active Server Pages
ColdFusion	Palvelinpuolen web-ohjelmointi teknologia
Dreamweaver	Ammattilaiskäyttöön tarkoitettu tekstipohjainen html-editori
FTP	File Transfer Protocol
	FTP on tiedostojen siirtokäytäntö, jota käytetään tiedostojen siirtämiseen.
HTML	HyperText Markup Language
JSP	JavaServer Pages
MySQL	Ilmainen SQL-tietokantojen hallintajärjestelmä
Oracle	Relaatiotietokannan hallintajärjestelmä
Paint Shop Pro	Ammattilaiskäyttöön tarkoitettu kuvankäsittelyohjelma
Perl	Practical Extraction and Report Language
PHP	Hypertext Preprocessor
Ruby on Rails	Ruby-ohjelmointikieleen pohjautuva ohjelmistokehys
SQL	Structured Query Language
	SQL on standardoitu kyselykieli, jolla voidaan tehdä relaatiotietokantaan erilaisia hakuja, muutoksia, lisäyksiä ja poistoja.
SQLite	Usealle eri alustalle sopiva SQL-tietokantamoottori
URL	Uniform Resource Locator. Käytetään osoittamaan www-sivuja
WWW	World Wide Web

1 JOHDANTO

Opinnäytetyön aiheena oli tehdä dynaamiset kotisivut Vanhain ystävät ry:lle. Opinnäytetyön aihe löytyi sattumalta Kajaanin Ammattikorkeakoulun Internet-sivuilta. Vanhain ystävät ry halusi kotisivut, koska aikaisempia sivuja heillä ei ollut. Nykyisin suurimmalla osalla yhdistyksistä on kotisivut ja monet yhdistykset, joilla ei ole kotisivuja, todella harkitsevat niitä. Kotisivujen tarkoituksena on tuoda yhdistys ja sen toimintaa tunnetuksi laajemmaltikin. Sivuston oli tarkoitus olla mahdollisimman yksinkertainen, koska kohderyhmä oli ikäihmiset.

Kotisivut on nykypäivänä hyvä tehdä tietokantapohjaisina. Siitä huolimatta, että niiden tekeminen vaatii enemmän työtä kuin tavalliset HTML-sivut, sivuston päivittäminen on helppoa. Päivittämisestä aiheutuvista kustannuksilta vältytään, koska henkilökunta pystyy päivittämään sivut itse selaimen kautta.

Nykyajan ohjelmointikielillä tietokantoja hyväksikäyttäen on kätevää tehdä turvalliset kotisivut. Esimerkiksi PHP:llä rakennetut sivut ovat turvallisuuden kannalta paremmat kuin pelkät HTML-sivut, koska PHP-koodia ei näe suoraan selaimelta. PHP onkin suosittu ohjelmointikieli dynaamisten kotisivujen toteuttamisessa, sillä sen lähdekoodi on vapaasti käytävissä ja muuteltavissa.

Tämän työn tarkoituksena on valaista Internet-sivujen toteutusmenetelmiä ja prosessia. Lisäksi työn tarkoitus on selvittää sivustolle asetettavat vaatimukset, kun käyttäjä- ja kohderyhmänä ovat vanhukset. Työssä käsitellään tarkemmin sivujen suunnitteluun liittyviä asioita ja pohditaan sivujen käytettävyyttä ottaen huomioon kohderyhmä.

Opiskelun tavoitteena oli myös ohjelmointitaitojen parantaminen tulevaisuuden työtehtäviä ajatellen. Alalla on hyvä hallita myös web-suunnittelutehtävien taitoja.

2 VANHAIN YSTÄVÄT RY.

Vanhain Ystävät ry. on Kajaanissa toimiva vanhusten virkistystoimintaa harjoittava yhdistys. Vanhain Ystävät ry:n tarkoituksena on tukea yksinäisten ja eläkkeellä olevien ihmisten oma-toimisuutta ja kokonaishyvinvointia. Yhdistys perustettiin jo vuonna 1923, kun vanhusten asuinolot havaittiin puutteelliseksi. Yhdistyksen alkuperäinen tarkoitus oli perustaa Kajaaniin vanhainkoti turvaamaan vanhuksille paremmat asuinolot Kajaanissa. (Schroderus 2005, 3-4)

Yhdistys järjestää jäsenilleen kerho- ja virkistystoimintaa kuukausittain, joissa ikääntyvät ihmiset voivat viettää aikaa toistensa seurassa. Toimintaan kuuluu luontoretkien järjestämistä lähialueille ja teatterissa käyntiä, joka on saavuttanut suuren suosion yhdistyksessä viime vuosina. Yhdistys tuottaa myös erilaisten esitysten ja aloitteiden kautta kajaanilaisille vanhuksille laadukkaita palveluita ja kehittää koko ajan uusia ajankohtaisia toimintamuotoja. Yhdistys aloitti vuonna 2007 laajan yhteistyön Edukai:n ja KAO:n vanhustyöhön suuntautuvien opiskelijoiden kanssa. Yhteistyötä kehitetään edelleen ja opiskelijat vierailevat kerhossa järjestämässä erilaista toimintaa jäsenille. (Vanhain Ystävät ry. 2007.)

Vanhain Ystävät ry:llä ei ole aikaisempia kotisivuja, joten yhdistys katsoi Internetin olevan paras keino tuoda esille toimintaansa. Myös erilaisia tiedotteita ja ajankohtaisia asioita on nopeaa ja vaivatonta tuoda ihmisten tietoisuuteen Internetin kautta. Yhdistys on tähän saakka tiedottanut toiminnastaan lähinnä pienten lehti-ilmoitusten välityksellä, mutta se katsottiin riittämättömäksi nykypäivän tietoyhteiskunnassa. Sivuston vaatimusmäärittelyssä oli todella tärkeää, että kotisivuille voidaan päivittää juuri näitä ajankohtaisia tietoja ja tapahtumia nopeasti. Näin sivusto saadaan pidettyä ajan tasalla jatkuvasti. Usein sivustoja ei ole päivitetty kovin usein, mikä antaa huonon vaikutelman kyseisen sivuston ylläpitäjästä tai omistajasta.

Sivuston ehkä tärkein tavoite on luoda tunnettavuutta. Kovin monikaan kajaanilainen ei ollut edes kuullut koko yhdistyksestä, kun olemme jutelleet ihmisten kanssa. Kun sivusto saadaan toteutettua ja siirrettyä Internetiin, niin toivottavasti yhdistys tulee paremmin ihmisten tietoisuuteen tätä kautta. Myös yhdistyksen eri toimintamuodot tulevat esille sivustolla. Yhdistykselle tehty esite on tarkoitettu jaettavaksi eri puolille kaupunkia. Esitteen olisi tarkoitus antaa mahdollisimman hyvä kuva yhdistyksen tarkoituksesta ja sen eri toimintamuodoista.

Vanhain Ystävät ry. toivoo myös saavansa sivuston avulla lisää jäsenistöä. Yhdistyksen jäsenmäärä on ollut laskussa viime vuosien aikana, joten kaikki uudet jäsenet ovat tarpeellisia.

Tämän vuoksi sivustolla on erillinen liittymislomake, jonka kautta jäseneksi liittyminen tapahtuu helposti ja nopeasti.

Sivuston yksi tärkeimmistä tavoitteista on sen yksinkertaisuus ja helppokäyttöisyys. Koska kohderyhmänä ovat vanhukset, niin sivuston ei pidä olla liian vaikea käyttää. Tämän vuoksi sivuston ei tarvitse erottua joukosta niinkään hienouden, vaan sen yksinkertaisuuden vuoksi.

Tärkeintä on, että yhdistyksen olemassaolo ja sen toiminta tulisivat tutuiksi lähinnä Kainuun alueella. Yhdistyksellä on paljon toimintaa myös Kajaanin lähikunnissa, joten tämän vuoksi tunnettavuutta toivotaan Kajaanin ulkopuolellakin.

3 DYNAAMISET WWW-SIVUT

Dynaamisten web-sivujen suosio on kasvanut 2000-luvun aikana merkittävästi. Nykypäivän ohjelmointikieliet mahdollistavat monipuolisten ja tasokkaiden dynaamisten kotisivujen kehittämisen ja erityisesti yrityksille ja yhdistyksille tehtävät kotisivut ovatkin nykyään poikkeuksetta dynaamiset. Tavallisia websivuja toki vielä tehdään, mutta dynaamisuudella sivustoista saadaan helpommin päivitettävät ja sivuston sisältöä monipuolisemmaksi.

Internetissä olevat sivustot voivat olla joko staattiset tai dynaamiset. Staattisten sivujen sisältö pysyy muuttumattomana, kunnes ylläpitäjä lataa palvelimelle uuden version sivustosta muokattuaan sitä ensin jollain html-editorilla. Dynaamisten sivujen sisältö puolestaan muuttuu käyttäjän toimenpiteiden seurauksena, esimerkiksi klikkaamalla kuvaa tai tekstiä sivustolla. Myös sivuston ylläpito helpottuu huomattavasti, kun toteutetaan dynaamiset kotisivut. Sivustoa varten rakennetaan erillinen hallintasivusto, jonka kautta ylläpitäjä voi kätevästi muuttaa reaaliajassa sisältöä. Esimerkiksi tapahtumakalenterin sisältöä on erittäin helppo muokata hallintasivustoa käyttäen.

Tietokantapohjaiset websivut ovat ehkä suosituin ja kätevin tapa toteuttaa dynaamiset sivut Internetiin. Tietokannoissa oleva informaatio näkyy sivulla joka kerta kun sivu ladataan selaimelle. Sivuille voidaan ladata tietokannoista kuvia, tekstiä ja muuta sisältöä. Kun tietokannassa olevia tietoja muutetaan, niin muutokset näkyvät automaattisesti myös websivustolla. Tietokannat saadaan toimimaan sivuston kanssa käyttämällä esimerkiksi ohjelmointikieliä PHP, ASP ja JSP. (Mischook, S. 2007.)

Tietokantapohjaisten sivujen avulla voidaan tekstin ja kuvien lisäksi tuottaa monenlaista tietoa julkaistavaksi sekä ihmisten nähtäviksi. Asiakirjoja ja muita dokumentteja voidaan kätevästi lisätä tietokantaan. Myös erilaiset kysely- ja palautelomakkeet voidaan toteuttaa dynaamisesti. Esimerkiksi sivuston toimivuutta koskevat tulokset nähdään suoraan tietokannasta, jolloin paperilomakkeita ei tarvita.

Tietokannoilla toteutettuja dynaamisia sivuja kutsutaan usein palvelinpuolen ohjelmoinniksi. Palvelinpuolen ohjelmoinnilla tarkoitetaan ohjelmointimenetelmää, jossa suorittava ohjelma-

koodi ajetaan palvelimella. Ohjelmaa ei siis tarvitse asentaa työasemalle, vaan sitä käytetään suoraan palvelimelta verkkoyhteyden yli. (Mischook, S. 2007.)

Asiakaspuolen ohjelmointi puolestaan suoritetaan käyttäjän selainohjelmassa www-palvelimen sijaan. Tämän tyyppiset ohjelmat ovat tärkeä osa dynaamisen HTML-kielen (DHTML) konseptia. Ne mahdollistavat sivuston sisällön mukauttamisen käyttäjän syötteen, olosuhteiden (kuten kellonajan) tai muiden muuttujien perusteella. (Mischook, S. 2007.)

4 VAATIMUSMÄÄRITTELY

Vaatimusmäärittely on dokumentti, johon kootaan kehitettävän järjestelmän eri sidosryhmien järjestelmälle asettamat vaatimukset. Vaatimusmäärittelyssä tuodaan esille sidosryhmien tarpeet, mutta ei puututa järjestelmän tekniseen toteutukseen. Vaatimukset luokitellaan toiminnallisiin ja ei-toiminnallisiin vaatimuksiin. Toiminnalliset vaatimukset määrittelevät, mitä järjestelmän odotetaan tekevän, miten se kommunikoi ympäristönsä kanssa ja miten eri sidosryhmien välinen yhteistyö on toteutettu. Ei-toiminnalliset vaatimukset puolestaan määrittelevät millaisten reunaehtojen vallitessa järjestelmä täyttää toiminnalliset vaatimukset. Käytännössä ne siis määrittelevät sellaisia järjestelmän ominaisuuksia kuten vastausajat, kapasiteetti ja käytettävyys. Lisäksi on myös rajoitteet, joilla määritellään rajoituksia toiminnallisille vaatimuksille. (Pohjonen 2002, 28.)

Asiakasvaatimusten kerääminen on vaativaa työtä. Koskaan ei ole vain yhtä yksittäistä keräysmenetelmää, joka takaisi tarpeeksi laajan lopputuloksen. Yleisin tapa vaatimusten selvittämiseen on haastatella järjestelmän eri sidosryhmien edustajia. Oikein hyödynnettynä tällaiset haastattelut tarjoavat mahdollisuuden kerätä tärkeää taustatietoa rakennettavan järjestelmän käyttäjistä sekä muista yleisistä käytänteistä. Täytyy osata valita oikeat henkilöt sidosryhmien sisältä, jotta haastattelut onnistuisivat ja niistä olisi todella hyötyä. (Pohjonen 2002, 28.)

Toinen yleinen tapa vaatimusten keräämiseksi on järjestää erilaisia aivoriihiä ja ideointipalaveriteita. Ensimmäisessä kokouksessa jokainen esittelee omat vaatimuksensa, joista voidaan keskustella ja rakentaa yhteistä kokonaista vaatimusmäärittelyä. Harvoin kuitenkaan päästään yhteisymmärrykseen yhden palaverin aikana. Sen vuoksi on hyvä muodostaa pienryhmiä, joissa määrittelyä työstetään eteenpäin. Lopuksi pienryhmien ideat määrittelyn suhteen tuodaan arvioitavaksi uuteen kokoukseen. Tätä jatketaan niin kauan, kunnes vaatimusmäärittely saadaan valmiiksi. Koska tämäntyyppiseen prosessiin osallistuu huomattavasti pienempi joukko sidosryhmien jäseniä, korostuu pätevien edustajien merkitys prosessin onnistumisen kannalta. (Pohjonen 2002, 29.)

Vaatimusmäärittelyä tehtäessä törmätään usein ongelmaan, että vaatimukset ovat keskeneräisiä ja epäselviä. Tämän vuoksi vaatimuksia täytyy yleensä työstää ennen niiden kirjaamista varsinaiseksi vaatimusmäärittelyksi. Epämääräisesti tulkittavat vaatimukset aiheuttavat on-

gelmia varsinkin kehitystyön tuloksia arvioitaessa. Tämän välttämiseksi vaatimukset täytyisi kirjata hyvin tarkasti. (Pohjonen 2002, 29.)

Vaatimuksia työstettäessä tulee kiinnittää huomiota yksittäisen vaatimuksen todelliseen käyttötarkoitukseen. Vaatimusten merkitystä arvioitaessa täytyy ottaa huomioon, että järjestelmillä tuskin ratkaistaan ongelmia, mihin organisaatio ei itse löydä ratkaisuja. Hyvin usein rakennettavan järjestelmän tietojenkäsittelytehtäviin liittyy organisaatiossa huomattuja ongelmia, joiden syytä tai ratkaisuja ei tunneta. Tämän vuoksi vaatimusmäärittelyyn voi päätyä epämääräisiä vaatimuksia, joiden toivotaan ratkaisevan ongelmat ilman hankkeeseen varattuja resursseja. Tällaiset ongelmat olisikin hyvä ratkaista erillisissä projekteissa. Niiden ratkaiseminen voidaan myös kirjata selvemmin kehityshankkeen tavoitteeksi. (Pohjonen 2002, 30.)

Opinnäytetyön vaatimusmäärittelyä pohdittiin aluksi Vanhain ystävät ry:n hallituksen jäsenten kanssa. Heti ensimmäisessä palaverissa selvisi, että kotisivujen tulee olla yksinkertaiset ja selkeät ottaen huomioon kohderyhmä. Kuitenkaan ensimmäisessä yhteisessä kokoontumisessa ei käynyt ilmi kaikkia vaatimuksia sivuston suhteen. Jatkossa kumpikin osapuoli mietti sivuston vaatimuksia ja ideoita vaihdettiin sähköpostin välityksellä. Sivuston suunnittelun aikana pidettiin myös pari palaveria, joissa keskusteltiin tarkemmin vaatimuksista. Sivuston alkaessa olla loppuvaiheessa, pidettiin kokous, jossa esittelimme yhdistyksen jäsenille sivustoa. Esittelytilaisuudessa saimme kommentteja ja parannusehdotuksia sivujen suhteen. Tilaisuuden jälkeen kaikki vaatimukset eivät siltikään tulleet esille, vaan vasta aivan työn loppuvaiheessa selvisi, että sivustolle olisi pitänyt lisätä vielä yksi osio, mutta saimme ongelman ratkaistua kumpaakin osapuolta miellyttävällä tavalla.

Toimeksiantaja halusi kotisivut tiedottaakseen yhdistyksen olemassaolosta ja toiminnoista. Sivujen täytyi olla helposti päivitettävät ja lisäksi sivuston sisällölle asetettiin tiettyjä vaatimuksia. Sivuilta tuli löytyä tietoa yhdistyksen historiasta, toiminta-ajatuksesta, hallituksen jäsenistä sekä kokouksien päätöksistä. Lisäksi sivulle haluttiin kuvia, ajankohtaisten asioiden osio, vuosittainen tapahtumakalenteri sekä tiedot yhdistyksen vuokra-asunnoista. Yhdistys haluaa uusia jäseniä, joten sivuston toivottiin sisältävän jäseneksi liittymis- osio, jonka avulla helposti pystyy liittymään yhdistykseen.

Sivuston toiminnallisia vaatimuksia olivat kertoa yhdistyksestä ja sen toimintatavoista sekä antaa mahdollisuus liittyä jäseneksi. Ei-toiminnalliset vaatimukset sivuilla liittyivät käytettä-

vyyteen. Näitä vaatimuksia olivat suuri tekstin fonttikoko, hillityt värit ja selkeä sivuston navigointi. Sivujen tuli myös näkyä selkeästi riippumatta siitä, mitä Internet-selainta käytetään.

Tietyt vaatimukset tuottivat ongelmia sivuston kehittämisessä, koska toteutuksen tekniset ratkaisut aiheuttivat ristiriitaisuuksia joidenkin vaatimusten kanssa. Esimerkiksi sivustolla olevaa kuvaosiota ei saatu toteutettua kuten toimeksiantaja ja tekijät olisivat halunneet. Kyseinen osio kuitenkin toteutettiin erilailla, mutta vaatimusten mukainen lopputulos olisi ollut ehkä mahdollinen käyttämällä jotain toista teknistä ratkaisua. Lisäksi käyttämme html-editori aiheutti ongelmia lähinnä sivuston ulkoasua tehtäessä. Esimerkiksi tekstin sijoittaminen ja asettelu itse sivustolle oli hankalaa ja työlästä. Myös muutamat viime hetkillä tulleet vaatimukset ja lisäsehdotukset aiheuttivat ongelmia niiden sivuille sijoittamisen kanssa. Olimme tehneet sivuille palaverissa jo sovitut asiat tietyllä tavalla, joten nämä viime hetken lisäykset olisivat aiheuttaneet uudelleen suunnittelua.

Suurin osa vaatimuksista kuitenkin saatiin täytettyä sekä toimeksiantajaa että tekijöitä miellyttävällä tavalla. Mielestämme tärkein vaatimus sivuston suhteen oli sen selkeys ja yksinkertaisuus, jossa onnistuimme vaatimusten mukaisesti. Sivuston navigointi ja ulkoasu olivat myös onnistuneita ratkaisuja, kuten esittelytilaisuudessa kävi ilmi. Toimeksiantajan vaatimukset eivät kuitenkaan olleet kovinkaan haasteellisia, koska tekijöille annettiin hyvin vapaat kädet. Asiaan vaikutti myös se, ettei yhdistyksellä ollut aikaisempia kotisivuja, jolloin vaatimustaso ei ollut korkea näitä sivuja suunniteltaessa ja tehtäessä.

Kokonaisuutena vanhuksille suunniteltaessa kotisivuja täytyy ottaa huomioon ennenkaikkea sivuston yksinkertaisuus. Lyhytaikaisessa selailussa liian monimutkaiset sivut hidastavat sivuilla liikkumista. On tärkeää, että sivuista kiinnostuneet ihmiset löytävät haluamansa asiat sivuilta helposti ja nopeasti. Sivustoa suunnitellessamme otimme huomioon tämän asian ja pyrimme erilaisin teknisin ratkaisun pääsemään toivottuun lopputulokseen. Kiinnitimme myös erityisen paljon huomiota sivustolla käytettyyn fonttikokoon ja fontin tyyliin. Liian pieni tai epäselvä fontti ei olisi kenenkään etujen mukaista.

5 KÄYTTÄJÄVAATIMUKSET OHJAAVAT SIVUSTON TOTEUTUSTA

5.1 Sivuston rakenteen suunnittelu

Sivuja suunniteltaessa on otettava huomioon useita eri asioita. On tarkkaan mietittävä sivuston perustavoitteet, käyttäjäryhmä sekä ulkoasuun liittyvät seikat.

5.1.1 Kotisivun suunnittelu

Kotisivun suunnittelussa tulee huomioida, että sivulta täytyy selvittää sivun tunnus ja tehtävä eli mistä sivustosta on kyse ja mitä sillä tehdään. Mieluiten myös syy, jonka vuoksi käyttäjä on siellä eikä jossain muussa sivustossa. Kotisivun on annettava yleiskuva siitä, mitä tarjottavaa sivustolla on –sekä sisällön että toimintojen osalta –ja miten kaikki on järjestetty. Sivustossa tarvitaan usein myös etsintäruutu, joka on sijoitettu näkyvälle paikalle kotisivulla. (Krug 2006, 95.)

Jos sivuston menestys riippuu siitä, että käyttäjät tulevat usein takaisin, kotisivulla pitää olla sisältöä, joka päivitetään säännöllisesti. Vaikka sivustossa ei kaivattaisi säännöllisesti vierailijoita, siinä pitää näkyä jonkinlaisia merkkejä siitä, että sivusto on yhä toiminnassa. Jos sivustoon on mahdollista rekisteröityä, kotisivulla tarvitaan linkit, joiden kautta uusi käyttäjä voi rekisteröityä ja vanha käyttäjä kirjautua sisään. (Krug 2006, 96-97.)

Kotisivun on tehtävä selväksi, miten löydetään etsitty asia –jos se ylipäänsä on jossain päin sivustoa. Samanaikaisesti kotisivun pitää esitellä sivun sisältöä, josta käyttäjä on ehkä kiinnostunut –vaikkei etsisikään sitä. (Krug 2006, 96-97.)

Yksi kotisivun arvokkaimmista paikoista on välittömästi sivuston tunnuksen oikealla puolella. Kun lause, joka on visuaalisesti yhteydessä sivuston tunnukseen, se tunnistetaan heti iskulauseeksi ja sitä pidetään koko sivuston kuvauksena. Iskulauseita on käytetty vuosikausia mainoksissa, viihteessä ja julkaisutoiminnassa. (Krug 2006, 101.)

Tervetuloivotus on sivuston ytimekäs kuvaus, joka on sijoitettu näkyvälle paikalle ja joka voidaan lukea sivua vierittämättä. Huomattakoon, että kaikki käyttäjät eivät tarvitse tätä eikä edes huomaa sitä. Useimmat todennäköisesti yrittävät arvata sivuston laadun kotisivun yleisen sisällön perusteella. (Krug 2006, 101.)

Kotisivutilaa kannattaa käyttää niin paljon kuin tarpeen, mutta ei yhtään sen enempää. Useimmissa sivustoissa ei tarvitse kuluttaa paljon tilaa perusajatuksen esittämiseen. Jos viesti täyttää koko kotisivun, käyttäjät eivät jaksakaan lukea sitä. Ei kannata luetella kaikkia hienouksia, vaan ainoastaan tärkeimmät. (Krug 2006, 103.)

Kun lähdimme rakentamaan kotisivuja Vanhain Ystävät ry:lle, tarkoituksena oli tehdä niistä niin yksinkertaiset kuin mahdollista. Sivuston käyttäjäryhmänä oli vanhukset, joten heille sivujen täytyi näyttää helposti hahmotettavilta ja käytettäviltä. Valitsimme ensin yksivärisen taustaväriin ja olimme lisänneet siihen jo osan linkeistä, kun tajusimme, että se on aivan kamalan näköinen. Hylkäsimme pian tämän vaihtoehdon ja ryhdyimme suunnittelemaan ulkoasua ja taustaa kokonaan uusiksi. Päädyimme tekemään sivustoon otsikkopalkin ja linkkipalkin eri väreillä. Otsikkopalkkiin tuli kunkin linkkipalkilla olevan linkin otsikko. Muu osa johon tulisi kaikki sivuston asiat, jätimme kokonaan valkoiseksi.

Sivuston linkit on valittu niin, että niiden alta löytyy kaikki tarvittava tieto yhdistyksestä kiinnostuneille. Kotisivutilaa on käytetty niukasti eli olemme yrittäneet tuoda perustiedot yhdistyksestä lyhyesti ja ytimekkäästi esille sivustolle. Tekstiä on käytetty hyvin vähän ja osa tekstistä on tekstitiedostossa linkkeinä sivuilla. Sivustolla on oma kuvaosionsa, muuten kuvia on käytetty kuin aivan muutama.

5.1.2 Kenelle?

Kun ryhdytään suunnittelemaan Web-sivuja, niin täytyy miettiä millaisille ihmisille sivusto on ensisijaisesti tarkoitettu. Tämä johtuu muun muassa siitä, että eri ihmiset tarvitsevat erilaisia esitystapoja.

Monet suomalaiset organisaatiot ovat tehneet erilaiset sivustot toisaalta kotimaiseen, toisaalta kansainväliseen käyttöön, jopa sitä myöten, että kotimainen sivusto on .fi-loppuisessa osoit-

teessa, kansainvälinen esimerkiksi .com-loppuisessa. Kansainvälisellä sivustolla voidaan esittää yksi ilmeinen ja tärkeä kohderyhmä ja tarkoitus: sen tulisi antaa kenelle tahansa vähänkin kiinnostuneelle nopea yleiskuva siitä, onko jonkin nimen takana yritys, laitos vai jotain muuta, mitä se tekee, missä se toimii ja miten siihen saa yhteyden. (Korpela, Linjama 2003, 49.)

Vanhain Ystävät ry. on kotimainen yhdistys, joten sen kotisivut on ensisijaisesti tarkoitettu suomalaisille ikääntyville ihmisille. Nämä kotisivut ovat tarkoitettu vain ja ainoastaan kotimaiseen käyttöön. Sivuston sisältö on kokonaan suomenkielinen. Ajattelimme ensin .fi-loppuista osoitetta, mutta yhdistys halusi .net-loppuisen osoitteen sivustolle.

5.1.3 Sivuston perustavoitteet

Web-sivuston tavoitteet voi yleisellä tasolla jäsentää yrityksen sivustoa koskevin. Tavoitteenä on:

- täydentää muita viestintämuotoja tarjoamalla ajankohtaista tietoa. Kun yritys julkaisee uuden tuotteen näyttävästi, sen Web-sivuilla on syytä löytyä siitä kaikki se tieto, mitä lehdistäkin, ja lisäksi tarkempia tietoja
- pitää tarjolla tietovarastoa, josta voi hakea kaikenlaista yritykseen liittyvää yksityiskohtaista tietoa, myös menneistä tapahtumista. Tällaisia tietoja ovat tuotetiedot.
- toimia yleisenä referenssinä, joka sisältää asialliset perustiedot yrityksestä niille, jotka sellaisia asioita Webistä etsivät.

Yleensä sivustoa ei ole mielekästä jäsentää varsinaisesti kohderyhmien mukaan, vaan sivustoon mukaan otettava aineisto määräytyy eri ryhmille hyödyllisen aineiston summana. (Korpela, Linjama 2003, 52.)

Sivuston perustavoitteita mietittäessä, Vanhain Ystävät ry:n kotisivut toimivat yleisenä referenssinä. Sivustolta löytyvät kaikki perustiedot yhdistyksestä niille, jotka näitä tietoja haluavat ja etsivät. Vanhain Ystävät ry. ei ole yritys, joten varsinaista tietovarastoa, kuten tuotetietoja ei sivustolla ole. Kotisivuilla löytyy usein päivittyvää ajankohtaista tietoa käsillä olevista tapahtumista. Vaikka yleensä sivustoa ei ole hyvä jäsentää varsinaisesti kohderyhmien mukaan,

Vanhain Ystävät ry:n tapauksessa tilanne on hieman eri. Todellisia kohderyhmiä kun ei ole kuin yksi eli vanhukset, joten kaikki käytetty aineisto on hyödyllistä.

5.1.4 Kohderyhmän rajaus

Kohderyhmän huomioon otto on erittäin tärkeää Web-sivuja suunniteltaessa. On syytä tavoitella yleistä ymmärrettävyyttä ja kiinnostavuutta. Suomea osaavien ihmisten joukkoon, englantia osaavista puhumattakaan, mahtuu erittäin suurta vaihtelua. Osa ihmisistä ehkä osaa lukea suomea sen verran, että ymmärtää sivun sisällön pääosin, mutta ei tiedä mitä varsinaisesti tarkoittaa jotkut suomalaisuuden termit ja sanat. Suomalaisen yhteiskunnan sisälläkin on suurta vaihtelua ihmisten välillä. Merkittävälle osalle esimerkiksi tavallinen suomen yleiskielikin tuottaa vaikeuksia eri syistä. (Korpela, Linjama 2003, 50-51.)

Tärkeintä on aloittaa siitä, että ei rakenna sivulle tarpeettomia esteitä, kuten jonkin asian sanominen vain kuvan muodossa. Pyrkimyksenä on hahmottaa sivuston ensisijainen kohderyhmä ja ne erityiset edellytykset ja ongelmat, joita siihen kuuluvilla on sivujen aihepiirin kannalta. (Korpela, Linjama 2003, 50-51.)

Työssämme on vain yksi todellinen kohderyhmä. Tietenkin sivuston ylläpitäjät ja muut sivustosta kiinnostuneet ovat kohderyhmiä, mutta vanhukset ovat etusijalla. Koko sivuston sisältö on suomenkielistä. Teksti on hyvin selkokielistä eikä siinä ole outoja sanoja tai termejä, kun suurimman osa sivuston teksteistä saimme yhdistykseltä. Kuvia on käytetty niukasti, joten varsinaisen asian ilmi tuominen on tekstimuodossa.

5.1.5 Web-sivujen luokittelu

Web-sivuja on luokiteltu monin eri tavoin, mutta käyttäjän kannalta olennaista on, että sivun ilmeinen tarkoitus selviää helposti. Kun käydään muiden tekemillä sivuilla, voidaan arvioida sivua sen mukaan, mikä sen tarkoitus näyttää olevan. Tyypillisiä esimerkkejä ovat:

- itseilmaisu: sivun tekijä haluaa kertoa itsestään
- asioiden listaaminen: sivu luettelee tosiasioita, faktoja

- selittäminen: sivu selostaa jonkin asian, ikään kuin esite, oppikirjan luku tai aikakauslehtijuttu
- mielipidekirjoitus: sivu yrittää vaikuttaa lukijan näkemyksiin
- uutisointi: kerrotaan ajankohtaisista tapahtumista samaan tapaan kuin sanomalehdissä
- mainonta: sivu yrittää myydä jotakin tuotetta tai palvelua
- viihdyttäminen: sivulla on ajanvietettä, esimerkiksi pelejä tai pilakuvia
- hakemisto: sivun kautta voi suunnistaa muille sivuille

(Korpela, Linjama 2003, 11.)

Sivun kannattaa olla tunnistettavissa esimerkiksi yritysesitteeksi, viihdesivuksi tai uutiseksi tai johonkin muuhun tyyppiin kuuluvaksi. Samalla sen olisi hyvä ainakin osittain erottua muista samantyyppisistä. Jos kyseessä on viihdesivu, sen pitäisi kertoa, millaista viihdettä se tarjoaa ja miten se eroaa kaikesta muusta viihteestä ja mikä on sen ominaislaatu. Yritysesitteestä pitäisi ilmetä, minkä nimisestä ja minkä alan yrityksestä on kyse, ja uutisia sisältävän sivun pitäisi jo kättelyssä kertoa, tarjoaako se urheilu-, pörssi-, tiede- vai joitakin muita uutisia. (Korpela, Linjama 2003, 12.)

Vanhain Ystävät ry:n kotisivut voidaan luokitella itseilmaisu-, asioiden listaamis- ja jokseenkin uutisointisivuksi. Sivuston päällimmäinen tarkoitus on kertoa yhdistyksestä ja sen toiminnasta. Lisäksi sivustolla kerrotaan yhdistyksestä tosiasioita, joita kaikkien yhdistyksestä kiinnostuneiden tulisi tietää. Uutisoinniksi voidaan laskea se, että sivustolla on kerrottu yhdistystoiminnan ajankohtaisista tapahtumista.

5.1.6 Sivuston rakentaminen

Jos sivustossa on enintään kymmenkunta sivua, sille riittää rakenteeksi, että pääsivulla hyvin lyhyesti asiaa ja linkit alisivuille ja kullakin alisivulla on linkki pääsivuun. Mutkikkaampi rakenne luultavasti vain hämmentää käyttäjää. (Korpela, Linjama 2003, 133.)

Jokaisella alisivulla tulisi olla ainakin yksi viittaus sivuhierarkiassa ylöspäin. Pienessä sivustossa riittää tähän linkki pääsivulle; sivun alussa tai sivun lopussa. Jos kyse on yrityksestä, yhtei-

söstä tai muusta organisaatiosta, on yleensä perusteltua, että heti sivuston alusta ilmenee, että ollaan sen sivulla. Yksinkertainen menettely on, että aivan alussa, vasemmalla ylhäällä, on organisaation nimi, logo tai tunnuskuva. (Korpela, Linjama 2003, 137.)

Vanhain Ystävät ry:n sivustolla on kymmenen sivua, etusivu mukaanlukien. Linkki etusivulle on ensimmäisenä linkkipalkissa ja se näkyy kuten koko linkkipalkki jokaisella sivulla. Tämän vuoksi erillistä "paluu etusivulle"-linkkiä ei tarvita sivujen loppuun.

Sivuston etusivu

Etusivun pitää kertoa jotakin sivuista, niiden tyylistä ja sisällöstä. Ulkoasulla tulisi näin esitetyinä tukea viestiä. Ulkoasu usein vaikuttaa ratkaisevasti ensivaikutelmaan.

Sivuston pääsivulta tulisi heti ilmetä, mistä sivuilla on kysymys, kuten onko kyseessä yrityksen sivusto, julkishallinnon sivusto vai harrastesivusto. Jos sivulla esimerkiksi on esillä yrityksen tuotemerkkejä, on luonnollista olettaa, että se on kaupallinen sivusto.

Yhdistyksen sivusto

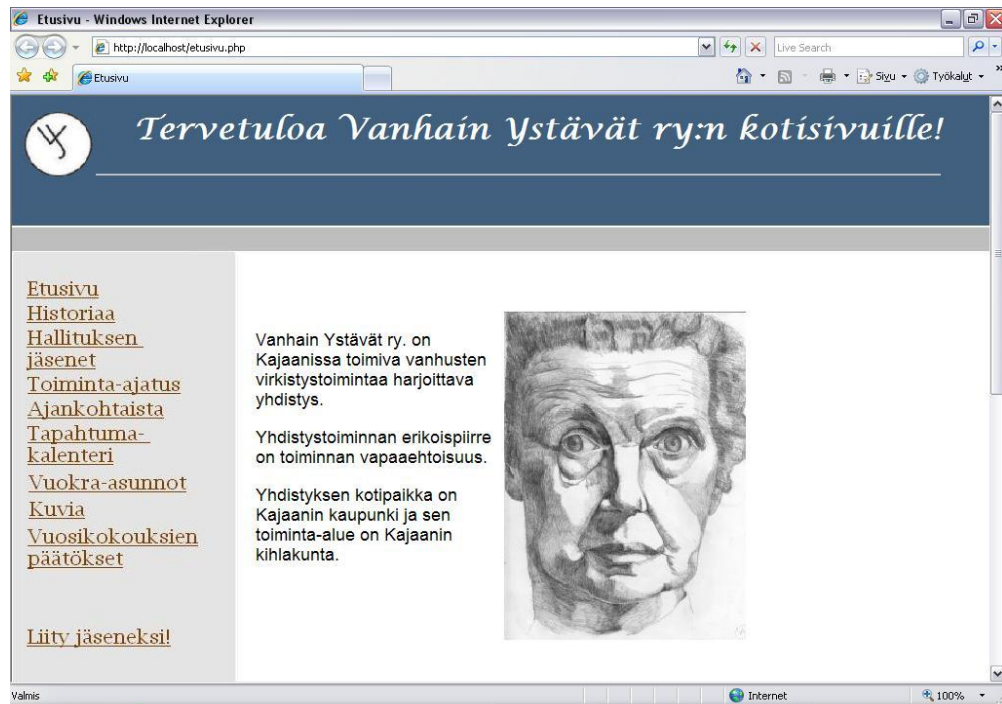
Yhdistyksen pääsivulla voisi olla otsikko, joka sisältää yhdistyksen nimen, lyhyt mutta tiivis kuvaus yhdistyksen tarkoituksesta eli päämäärästä, mahdollisesti kuvaukseen liittyvä iskulause tai kuva, joka havainnollistaa perus- ja toissijaiset linkit. Peruslinkit johtavat keskeisille toimintaa kuvaaville alisivuille. Toissijaiset linkit johtavat hallinnollisempaan aineistoon, kuten yhdistyksen sääntöihin, toimintasuunnitelmaan ja yhdistykseen liittymiseen. (Korpela, Linjama 2003, 144.)

Otsikoksi peruslinkkien listalle voisi sopia esimerkiksi "Tapahtumakalenteri", "Ajankohtaista" tai "Uutisia". Toissijaisten linkkien listalla voisi olla "Hallituksen jäsenet", "Toiminta-ajatus" ja "Vuosikokouksien päätökset". (Korpela, Linjama 2003, 144.)

Tapahtumakalenterista tai uutisista voidaan nostaa erityisen tärkeitä asioita pääsivulle. Tämä kuitenkin edellyttää sivujen päivittämistä.

Yhdistyksen sivustolla on yleensä kolme keskeistä käyttäjäryhmää: jäsenet, jäsenyyttä hakevat ja suuri yleisö, joka on yleensä kiinnostunut yhdistyksestä silloin, kun se on ollut esillä julkisuudessa. Yhdistyksen jäsenet ovat kuitenkin sivujen tärkeimmät käyttäjät eivätkä he tarvitse muistutusta siitä, mikä yhdistyksen tarkoitus on. (Korpela, Linjama 2003, 145.)

Vanhain Ystävät ry:n etusivun otsikoksi valittiin yksinkertainen tervetuloivotus ”Tervetuloa Vanhain Ystävät ry:n kotisivuille”. Ajattelimme, että se olisi paras vaihtoehto tässä tapauksessa. Aivan tervetuloivotuksen viereen vasemmalle puolelle laitoimme yhdistyksen logon kuvaamaan yhdistystä. Etusivulta myös selviää, että Vanhain Ystävät ry. on yhdistys ja mitä toimintaa se harjoittaa. Edellisen lisäksi etusivulla on luonnoskuva vanhan naisen kasvoista. Se kuvastaa sitä, että todella ollaan vanhuksille tarkoitetuilla sivuilla.



Kuva 1. Vanhain Ystävät ry:n etusivu

Etusivulla on erillinen linkkipalkki alisivuille, joka on myös sama kaikilla alisivuillakin. Linkkipalkin peruslinkit johtavat keskeisille yhdistyksen toiminnasta kertoville sivuille. Toissijaisia linkkejä sivustolla ovat yhdistyksen toiminta-ajatus, säännöt ja kokouksien päätökset. Kaikki nämä linkit johtavat Word-dokumentteihin. Peruslinkkeinä linkkipalkissa ovat ”Etusivu”, ”Historiaa”, ”Hallituksen jäsenet”, ”Toiminta-ajatus”, ”Ajankohtaista”, ”Tapahtumakalenteri”, ”Vuokra-asunnot”, ”Kuvia”, ” Vuosikokouksien päätökset” ja ”Liity jäseneksi!”.

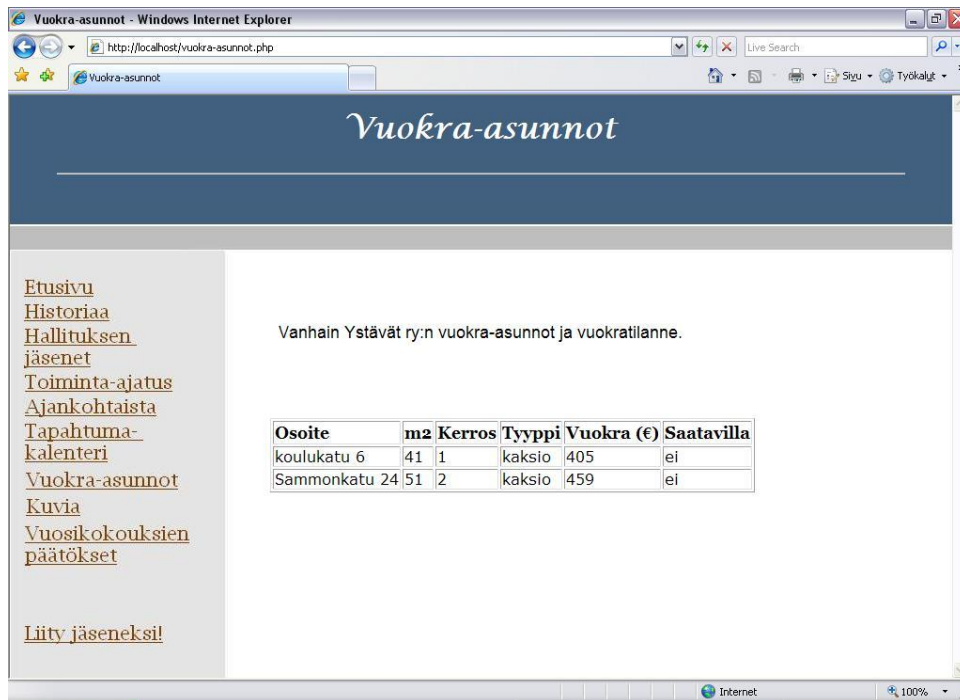
Historiaa-linkin alla on lyhyt, mutta tiivis kuvaus yhdistyksen historiasta ja sen vaiheista sekä kolme aiheeseen sopivaa kuvaa. Hallituksen jäsenet ovat kirjattu taulukkoon, jossa on kunkin jäsenen yhteystiedot.

The screenshot shows a web browser window titled 'Hallitus - Windows Internet Explorer'. The address bar displays 'http://localhost/hallitus.php'. The page content includes a dark blue header with the word 'Hallitus' in white. Below the header, there is a navigation menu on the left with links: [Etusivu](#), [Historiaa](#), [Hallituksen jäsenet](#), [Toiminta-ajatus](#), [Ajankohtaista](#), [Tapahtuma-kalenteri](#), [Vuokra-asunnot](#), [Kuvia](#), [Vuosikokouksien päätökset](#), and [Liity jäseneksi!](#). The main content area is titled 'Hallituksen jäsenet' and contains a table with the following data:

Nimi	Nimike	Puhelinnumero	Sposti
Marja Leskinen	puheenjohtaja	0400-271 006	marja.leskinen@suomi24.fi
Raija Honkanen	varapuheenjohtaja	040-7196 467	
Arja Ohtonen	sihteeri/rahastonhoitaja	0400-410 571	
Elsa Rautiainen	jäsen	050-4141 902	
Esko Jaakola	jäsen	044-3361 319	
Ritva Klemetti	varajäsen	044-5161 549	ritva.klemetti@hotmail.com
Pirkko Teno	varajäsen	040-5739 686	

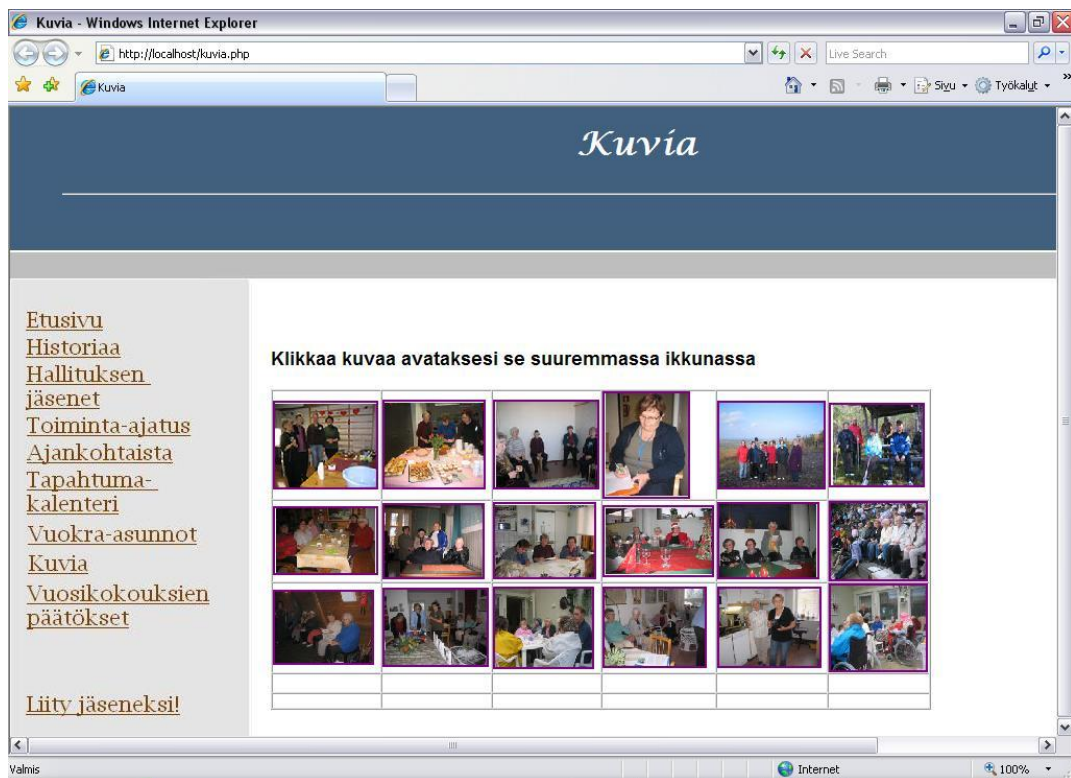
Kuva 2. Hallituksen jäsenet

Toiminta-ajatuksessa on kerrottu lyhyesti yhdistyksen toiminta-ajatus sekä linkit toimintakertomukseen ja yhdistyksen sääntöihin. Ajankohtaista-osiossa on yhdistyksen ajankohtaisia asioita joita päivitetään, kun uutta ilmenee. Tapahtumakalenteri on vuosittainen, kuukausittain jaettu lista, jossa on tarkat päivämäärät yhdistyksen tapahtumista. Vuokra-asunto-linkin alta löytyy tiedot yhdistyksellä olevista vuokra-asunnoista ja niiden saatavuudesta.



Kuva 3. Vuokra-asunnot

Kuvia-osiossa on taulukko, jossa kuvat ovat linkkeinä, pikkukuvina, joita klikkaamalla aukeaa uuteen ikkunaan pikkukuvaa vastaava iso kuva. Pikkukuvat on tehty Paint Shop Pro:lla yksinkertaisesti pienentämällä alkuperäinen kuvasuhde.



Kuva 4. Kuvagalleria

Vuosikokouksien päätöksissä on linkit hallituksen- ja vuosikokouksien päätöksiin. Liity jäseneksi-linkistä löytyy lomake, jossa omat tiedot täyttämällä pystyy liittymään yhdistyksen jäseneksi sekä jäsenmaksun hinta.

The screenshot shows a web browser window titled "Liity jäseneksi - Windows Internet Explorer". The address bar shows "http://localhost/liity.php". The page content includes a navigation menu on the left with links like "Etusivu", "Historiaa", "Hallituksen jäsenet", "Toiminta-ajatus", "Ajankohtaista", "Tapahtumakalenteri", "Vuokra-asunnot", "Kuvia", "Vuosikokouksien päätökset", and "Liity jäseneksi!". The main content area has the heading "Liity jäseneksi" and the text "Voit liittyä Vanhain ystävät ry:n jäseneksi täyttämällä alla olevan lomakkeen. Jäsenmaksu on 5 € /hlö". Below this is a form with the following fields: Sukunimi, Etunimet, Osoite, Puh. nro, and Sähköposti. At the bottom of the form are two buttons: "Lähetä" and "Tyhjennä".

Kuva 5. Jäseneksi liittymis-lomake

Näkyminen Webissä

Yksi tavallisimmista sivun tekijöiden ongelmista on saada sivunsa tunnetuksi. Jos sivulla ei ole mitään varsinaista sisältöä, on parempi, että kukaan ei vielä löydä sitä. Toisaalta, jos sivut suunnitellaan hyvin, ne ovat toimivia ja hyödyllisiä jo varhaisessa vaiheessa, niin silloin on hyvä saada niiden leviäminen hakujärjestelmään käyntiin nopeasti. (Korpela, Linjama 2003, 147.)

On useita tapoja, miten sivu saattaa löytyä. Jos joku mainitsee Web-sivun osoitteen esimerkiksi sanomalehdessä, sähköpostissa tai Internetin keskustelufoorumilla niin, että joku kiinnostuu ja käy katsomassa. Joku saattaa kirjoittaa omalle sivulleen linkin sille sivulle. Sivun saattaa tavalla tai toisella päätyä erilaisiin hakemistoihin. Hakemistosta ihmiset etsivät sivuja jär-

jestelmällisesti. Yksi tärkeimmistä vaihtoehtoista on hakukoneet, joilla haetaan sivuja hyvin laajasta ja järjestämättömästä joukosta sivuja. (Korpela, Linjama 2003, 147.)

Sivuista voidaan tiedottaa antamalla vinkkejä tutuille ihmisille, joiden tiedetään olevan kiinnostunut aihepiiristä. Lehtijutussa tai nettikirjoittelussa voidaan mainita sivu silloin, kun se liittyy asiaan. Voidaan myös ehdottaa muille, että he lisäisivät sivuilleen linkkejä niille sivuille. (Korpela, Linjama 2003, 147.)

5.2 Sivuston visuaalinen ilme

Sivustoa suunniteltaessa täytyy kiinnittää sisällön lisäksi huomiota myös sivujen ulkonäköön. Minkä verran tekstiä käyttää, millaisen fontin valitsee ja koko tekstin kieliasu täytyy olla hallinnassa. Värien ja taustakuvien käyttö sivustolla vaikuttaa merkittävästi sivuston ilmeeseen, varsinkin jos niitä ei ole oikein valittu.

5.2.1 Toimivan tekstin rakenne

Tekstin tyyli laji kannattaa valita ennen kuin ryhtyy kirjoittamaan mitään kotisivulle, koska sen muuttaminen on vaikeaa jälkikäteen. Vakava tyyli sopii, kun vakuutetaan jostakin asiasta tai jaetaan tietoa. Viihtelliseen tekstiin sopii rennompikin tyyli. Tyylin pitää vastata sivun tarkoitusta mahdollisimman hyvin ja kannattaakin valita tyyli, jonka hallitsee ja jota on yleensä käyttänyt. Tekstin kieliasu on viesti. Huoliteltu asu kertoo, että kirjoittaja välittää lukijasta ja haluaa esittää asiansa hyvin. Huolimaton kieliasu taas kertoo, että teksti on kirjoitettu kiireessä. Kirjoitusvirheitä täynnä oleva teksti ärsyttää monia. Erityisen häiritsevää huolimattomuus on pääsivulla, koska sen lukija hyvin usein näkee ensimmäisenä ja koska pääsivun oletetaan olevan harkittu kokonaisuus, jonka tekemiseksi on nähty vaivaa. Usein kirjoittaja jää miettimään kirjoitusasun yksityiskohtia, kuten tuleeko " sen jälkeen" vai "senjälkeen".

Asialla aloittaminen on Webissä erityisen tärkeää siksi, että alku on niin usein ainoa, minkä käyttäjä katsoo tai kuuntelee. Jos tekstin alku ei ala kiinnostamaan, käyttäjä menee muualle. (Korpela, Linjama 2003, 168-169.)

Yleensä asiattylinen kappale kannattaa aloittaa ydinvirkkeellä, joka esittää kappaleen olennaimmman sisällön. Kun asia vaihtuu, vaihtuu myös kappale. Ydinvirkkeillä aloittaminen mahdollistaa sen, että teksti silmäillään läpi lukien vain otsikot ja ydinvirkkeet. Tällöin lukijan pitäisi saada jo selvyys sivun asiasisällöstä. (Korpela, Linjama 2003, 170.)

Virkkeet kannattaa rakentaa kohtuullisen mittaisiksi. Kokematon kirjoittaja usein kirjoittaa niin lyhyitä virkkeitä, että pisteet vastaavat suunnilleen puheen hengenvetoja tai yrittää jäljitellä hienoa kirjallista tyyliä, jossa lukija putoaa kärryiltä. On hyvä lähteä liikkeelle yksinkertaisista lauseista, jotka ilmaisevat tekoa tai tapahtumista: joku tekee jotakin, jotakin sattuu jossakin. Lauseet sitten punotaan yleensä kahden, kolmen yhteenkuuluvan lauseen kokonaisuudeksi, virkkeiksi. Hyvä tavoite on, että virke irralleen otettunakin ilmaisee selvästi jonkin ajatuksen. Jos virke ei sisällä selvää ajatusta, ei yleensä löydy sitten kiinnostusta lukea tekstiä sen ympäriltä. (Korpela, Linjama 2003, 170-171.)

Tekstin selkokieliäisyyttä kannattaa tavoitella silloin, kun kohderyhmänä on mm. vanhuksat, lapset, maahanmuuttajat ja kehitysvammaiset. Selkokieliäisyys siis tarkoittaa yksinkertaistettuna kieltä, joka sopii niille, joiden on hyvin vaikea ymmärtää tavallista yleiskieltä. Selkokielelle ominaista on, että se pyrkii puhuttelemaan sinä-muodossa ja muutenkin kuvaamaan asioita lukijan kannalta pikemminkin kuin yleisin, abstraktisin käsittein. Yleensä selkokieliäisyys merkitsee myös sisällön karsimista olennaisimpaan. Lisäksi sisällöltään yksinkertaistettu esitys auttaa ymmärtämään loput yleiskielisestä esityksestä, kun perusasiat on omaksuttu ensin helposta esityksestä. (Korpela, Linjama 2003, 175-176.)

Kaikki Vanhain Ystävät ry:n sivustolla olevat tekstit on saatu yhdistykseltä. Siksi teksti on hyvin selkokieliäistä. Tekstiä on kuitenkin hyvin vähän, koska vain kaikkein olennaisimmat asiat on tuotu ilmi. Lauseet ovat lyhyitä, yksinkertaisia ja ne ilmaisevat selvästi kerrotun asian. Jo etusivulla on lyhyesti ja ytimekkäästi kerrottu mikä yhdistys on. Kappalejaot on tehty, kun niihin on ollut tarvetta ja olemme tarkistaneet, ettei kirjoitusvirheitä tekstissä ole.

Tekstityylit

Lihavoidulla tai kursivoidulla tekstityylillä tehdyt korostukset toimivat hyvin, jos niitä käytetään normaalin tekstin ohessa rajoitetusti. Ne ovatkin Web-sivuilla käytetyimmät korostus-

keinot. Muita korostustapoja ovat linkit ja listat. Listat erottuvat ensijassa asettelunsa perusteella. (Korpela, Linjama 2003, 383.)

Sivuston etusivulla oleva teksti on tehty lihavoidulla tekstityylillä. Joillakin sivuilla missä ei ole kuin pari lausetta, jotka viittaavat alla olevaan linkkiin on lihavoitu. Joillakin sivulla on jonkinlainen alaotsikko, joka on lihavoitu tai on käytetty isompaa fonttikokoa. Muuten on käytetty Arial-fonttia.

Linkit

Linkit ovat yleensä ulkoisesti erivärisiä ja alleviivattuja, joten ne luovat sivun leipätekstiin väri- ja muotokonstraktin. Linkit ovat hypertekstidokumentin tärkein osa, ja niiden tulisi erottua selvästi normaalista tekstistä. Lisäksi niiden tulee olla esillä sopivalla tavalla. Jos kyseessä on pääsivu, linkkien tulisi olla näkyvissä heti, kun käyttäjä saapuu sivulle. Tämä on sivuston käytettävyyden kannalta ensiarvoisen tärkeää. (Korpela, Linjama 2003, 384.)

Vanhain Ystävät ry:n sivustolla on kuten jo edellä mainittu, erillinen linkkipalkki, jossa on kaikki sivuston päalinkit. Linkkipalkki on tehty niin, että kun jotakin sivua päivitetään, lisätty teksti tai kuva ei tule palkin päälle vaan pysyy sille rajatulla alueella. Linkit ovat alleviivattuja ja niiden väriksi valitsimme ruskean, jotta ne kunnolla erottuisivat sivuilla.

5.2.2 Värien käyttö

Väri on voimakas viesti, ja sillä voi olla monia toiminnallisia käyttötarkoituksia. Sillä voidaan korostaa, järjestellä, osoittaa erilaisia asioita ja erottaa esimerkiksi otsikkotasoja toisistaan. Väri voi toimia tunnistuksen apuna: yrityksen tunnusvärillä varustettu sivu tunnistetaan ensi silmäyksellä. (Korpela, Linjama 2003, 397.)

Värien käytöllä voidaan korostaa sivun osia hyvinkin vahvasti. Jos esimerkiksi muutoin mustavalkoisella sivulla on isohko punainen kuva tai teksti, katse kiinnittyy ensimmäiseksi siihen. Sinistä väriä pidetään etäännyttävänä, viileänä ja rauhoittavana värinä. Punainen taas koetaan kiihdyttävänä, dynaamisena ja lähelle tulevana värinä. Keltainen on valovoimainen väri, ja se

mielletään iloiseksi ja piristäväksi. Oranssi on lämmin, vihreä rauhoittaa ja se yhdistetään yleensä luontoon. (Korpela, Linjama 2003, 398.)

Värien käyttö vaatii kuitenkin hyvin tarkkaa harkintaa. Väärällä värien ja grafiikan valinnalla voi pilata sivusto käytettävyyden. Vaikka värien keskinäinen valinta onnistuisikin, voi valittujen värien sävy, kylläisyys ja käyttötapa poiketa käyttäjän mieltä liikaa sivuston aihepiiristä. (Korpela, Linjama 2003, 398.)

Värejä Vanhain Ystävät ry:n sivuilla ei ole käytetty juuri ollenkaan. Ainoastaan joka sivulla näkyvät otsikkopalkki ja linkkipalkki ovat värillisiä. Otsikkopalkki on vaalean sininen, jonka alla on ohut, tummanharmaa palkki. Linkkipalkki puolestaan on vaalean harmaa ja itse linkit ruskeita.

5.2.3 Taustakuvat

Sivulla ei tarvitse olla taustakuvaa. Monet Webin sivustot olisivat parempia, jos niiden taustakuvat poistettaisiin. Hyvän taustakuvan valinta tai tekeminen vaatii osaamista, harkintaa ja kokeiluja. (Korpela, Linjama 2003, 245.)

Taustakuva luo sivun yleisvaikutelmaa ja tunnelmaa. Varsinkin, jos kyse on koko sivun taustakuvasta sivuston kaikilla sivuilla, se vaikuttaa kaikkialla pelkääntä pinta-alansa voimalla. Taustakuva toimii myös tunnistamisen välineenä, ja se näkyy silloinkin, kun ollaan keskellä sivua, jossa logoa ei näy. Taustakuvilla voidaan myös rakentaa koristekuvitusta kuten sivun vasemmassa reunuksessa oleva ornamentti tai sivun poikki menevä koristeellinen, mutta samaa kuviota käyttävä vaakakuvi. (Korpela, Linjama 2003, 245.)

5.3 Lomakkeet ja niiden perusidea

Suomen kielen sanakirja kuvailee, että lomake on ”täytettävä paperiarkki t. kortti, johon on painettu täyttämiskiinteet ja tyhjää tilaa vastaavia merkintöjä varten”. Html:ään otettiin jo varhain mukaan lomake-elementti, joka monin tavoin vastaa paperilomaketta – ja kuitenkin eroaa siitä ratkaisevasti. Yksi olennainen ero paperilomakkeen ja html-lomakkeen välillä on,

että html-lomakkeella lähetetyt tiedot menevät aina tietokoneohjelman käsiteltäväksi. Toinen olennainen ero on, että html-lomake täytetään ja lähetetään selaimella. (Korpela, Linjama 2003, 272-273.)

Html:ssä lomake, englanniksi *form*, antaa selaimelle ohjeen muodostaa käyttöliittymä, jonka kautta käyttäjä voi kirjoittaa tekstiä tai antaa muuta dataa. Lomakkeeseen liittyy myös action-määrite, joka kertoo selaimelle, mihin osoitteeseen sen tulee lähettää saatu data jonkinlaista käsittelyä varten. Tämä osoite (URL) viittaa palvelimessa toimivaan ohjelmaan tai skriptiin, jota voidaan kutsua palvelinskriptiksi. Skripti tarkoittaa asiallisesti samaa kuin ohjelma, mutta skripteiksi tavataan sanoa pienehköjä ohjelmia tai ns. tulkitsevalla tekniikalla toteutettuja ohjelmia. Palvelinskriptin yksi tavallinen alkeellinen vaihtoehto on, että se vain lähettää saamansa tiedot sähköpostitse johonkin osoitteeseen. (Korpela, Linjama 2003, 272-273.)

5.3.1 Lomakkeen kentät ja yleisrakenne

Lomakkeen sisällä voi olla kokonaisia kappaleita ja jopa taulukoita ja muita lohkorakenteita. Hyvä periaate on, että ennen kutakin syöttökenttää on selitys, joka kertoo kentän merkityksen, ja että yleensä selitys ja kenttä ovat samalla rivillä, jolla ei ole mitään muuta.

Varsinaisten syöttökenttien luomiseen on kolme erilaista elementtiä: input, textarea ja select. Tärkeimmät kenttien tyypit voidaan ryhmitellä seuraavasti:

- Lähetyskenttä eli input- kentässä on painike tai vastaava, jota käyttäen lomakkeen voi lähettää. Vähintään yksi sellainen pitää olla jokaisessa lomakkeessa, koska siitä käyttäjä voi lähettää lomakkeen.
- Piilokenttä välittää palvelinskriptille valmiiksi täytettyä dataa, joka ei ole käyttäjän normaalein keinoin muutettavissa eikä edes nähtävissä.
- Rivinsyöttökenttään käyttäjä voi kirjoittaa yhden rivin tekstiä.
- Tekstialueeseen (textarea), käyttäjä voi kirjoittaa useita rivejä tekstiä.
- Tiedostokentässä käyttäjä voi valita omasta koneestaan tiedoston, jonka sisältö lähtee muiden tietojen mukana.
- Valikkokentässä (select) annetaan käyttäjälle tehtäväksi valita annetusta joukosta vaihtoehtoja.

Näiden lisäksi on muun muassa komento "reset", joka luo painikkeen, joka hävittää kaiken lomakkeeseen kirjoitetun, tarkemmin sanoen alustaa lomakkeen kaikki kentät alkuarvoihinsa. (Korpela, Linjama 2003, 283-284.)

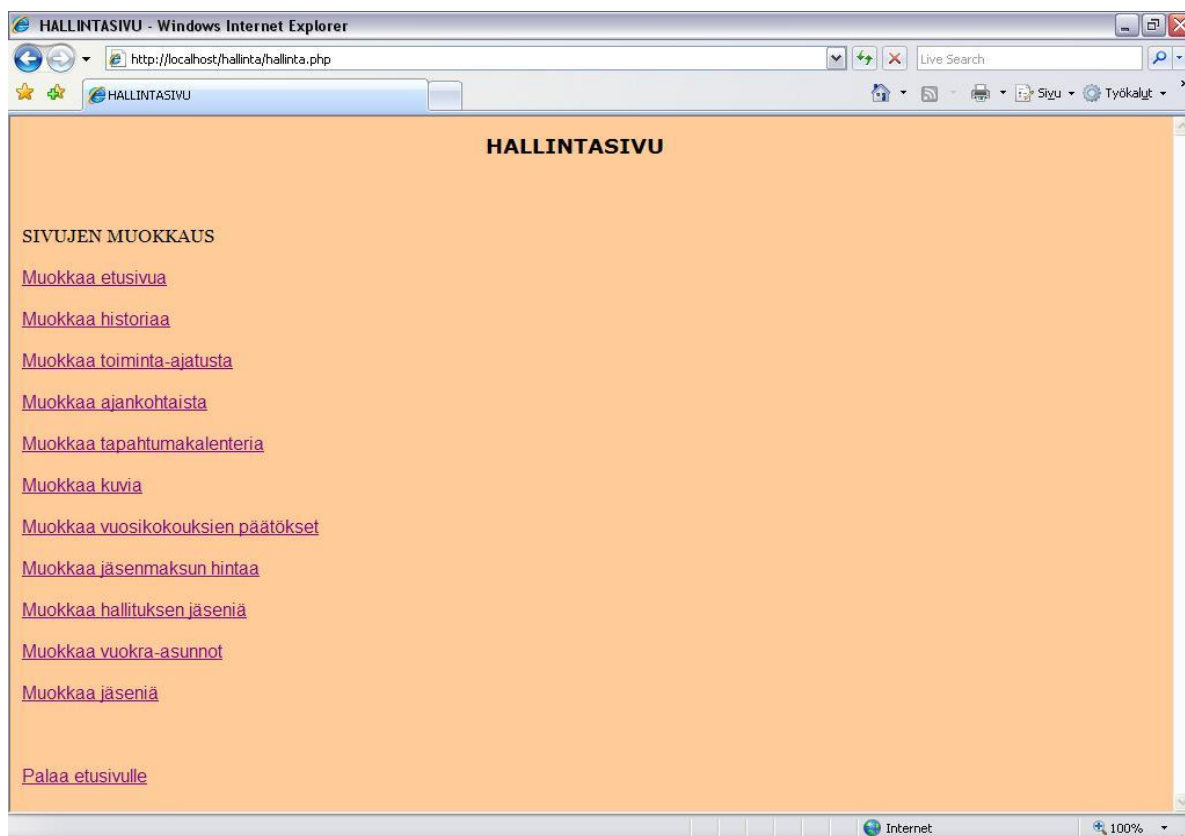
5.3.2 Liittymislomake

Vanhain Ystävät ry:n kotisivustolla on lomake, jossa käyttäjä voi liittyä yhdistyksen jäseneksi. Lomake on hyvin yksinkertainen. Siinä on viisi rivinsyöttökenttää ja lähetykenttä. Rivinsyöttökenttien edessä ovat selitykset "Sukunimi", "Etunimet", "Osoite", "Puh.nro ja "Sähköposti. Lähetyksessä ovat painikkeet "Lähetä" ja "Tyhjennä".

Lomake toimii siten, että käyttäjä täyttää rivinsyöttökenttien tiedot ja painaa "Lähetä"-painiketta. Siitä avautuu sivu, jossa on teksti: "Kiitos ilmoittautumisesta, otamme yhteyttä!" Uuden jäsenen tiedot tallentuvat hallintasivulle, josta niitä päästään tarkastelemaan. Jos käyttäjä ei kuitenkaan halua liittyä jäseneksi, hän voi alustaa lomakkeen painamalla "Tyhjennä"-painiketta.

5.4 Hallintasivu

Hallintasivun taustaväri on kellertävän oranssi väri. Tarkoituksena oli, että hallintasivu erottuu varsinaisesta sivustosta tällä tavoin. Alue, jossa jokaisen sivun sisältö on jätetty valkoiseksi. Halusimme, että ainoastaan linkit ja otsikot erottuvat edukseen sivustolla.



Kuva 6. Hallintasisivun etusivu

Pelkkää tekstiä on "Muokkaa toiminta-ajatusta", "Muokkaa ajankohtaista", "Muokkaa tapahtumakalenteria", "Muokkaa vuosikokouksien päätökset" ja "Muokkaa jäsenmaksun hintaa"-linkkien alla. "Muokkaa hallituksen jäseniä", "Muokkaa vuokra-asunnot" ja "Muokkaa jäseniä"-linkkien sisältö tulee tietokannoista. Kun esimerkiksi muokataan hallituksen jäsenten tietoja, niin tiedot muuttuvat myös tietokantaan. Samoin, jos lisätään kokonaan uusi jäsen, se tulee myös tietokantaan näkyviin. Tiedot ovat taulukossa kuten itse sivullakin. Taulukon oikeassa laidassa on linkit "Poista" ja "Muokkaa". Kun klikataan "Poista" halutun jäsenen kohdalla, nimi ja kaikki muutkin kyseisen jäsenen tiedot häviää taulukosta. Jos halutaan

muokata jonkin jäsenen tietoja, se onnistuu klikkaamalla "Muokkaa"-linkkiä.

Muokkaa/poista tietoja					
Nimi	Nimike	Puhelinnumero	Sposti		
Marja Leskinen	puheenjohtaja	0400-271 006	marja.leskinen@suomi24.fi	Poista	Muokkaa
Raija Honkanen	varapuheenjohtaja	040-7196 467		Poista	Muokkaa
Arja Ohtonen	sihteeri/rahastonhoitaja	0400-410 571		Poista	Muokkaa
Elsa Rautiainen	jäsen	050-4141 902		Poista	Muokkaa
Esko Jaakola	jäsen	044-3361 319		Poista	Muokkaa
Ritva Klemetti	varajäsen	044-5161 549	ritva.klemetti@hotmail.com	Poista	Muokkaa
Pirkko Teno	varajäsen	040-5739 686		Poista	Muokkaa

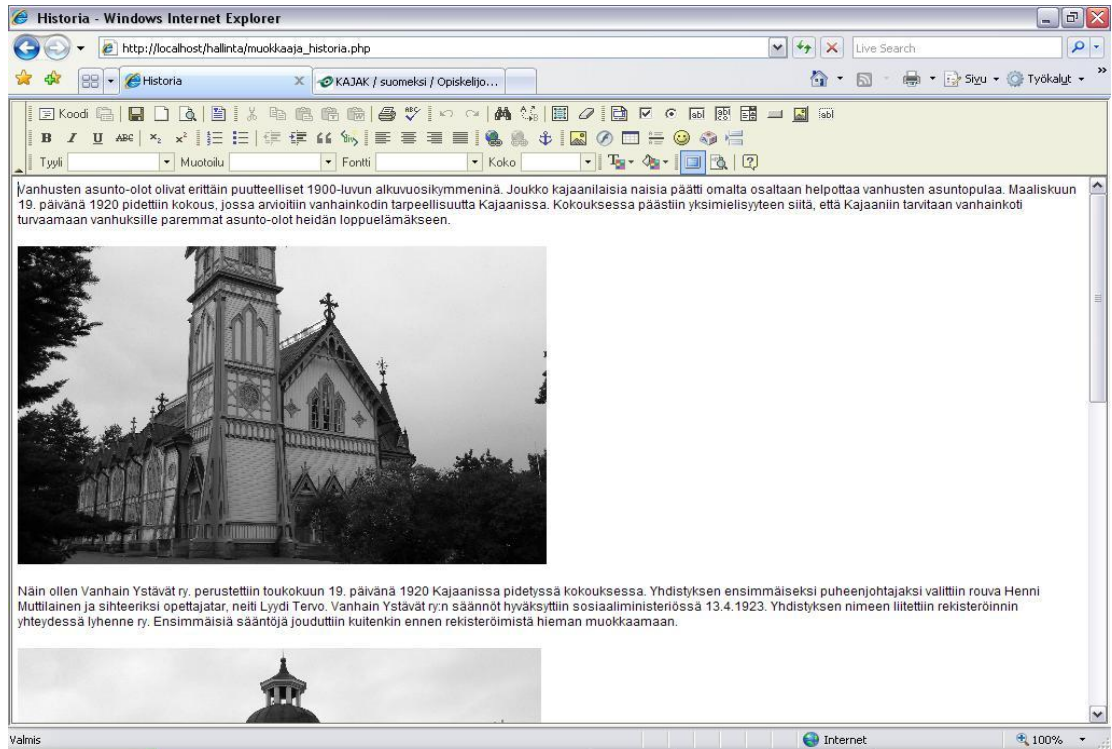
Lisää jäsen:

Nimi:
 Nimike:
 Puhelinnumero:
 Sposti:

[Palaa hallintasivulle](#)

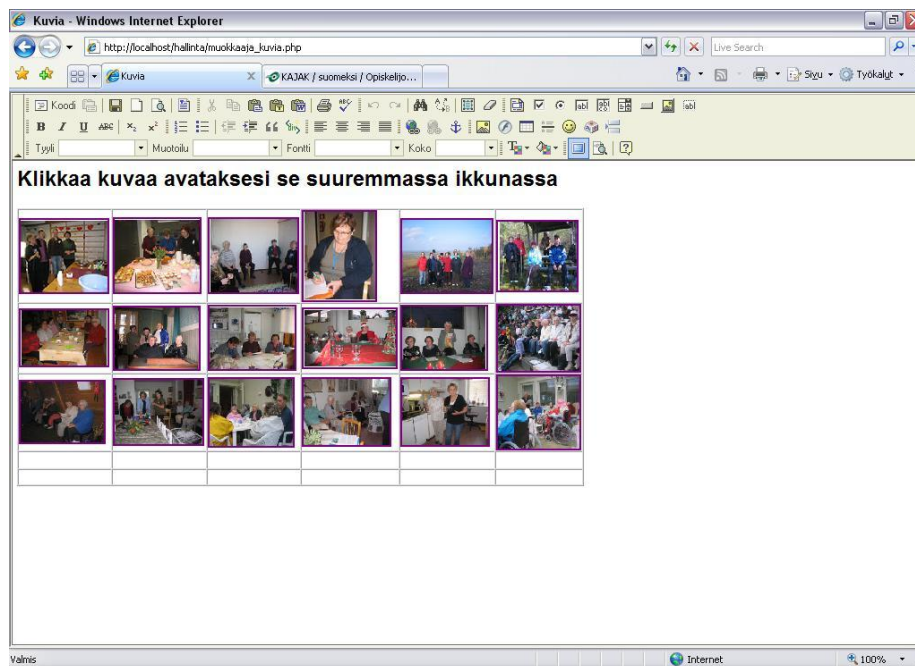
Kuva 7. Hallituksen jäsentietojen muokkaaminen

Hallintasivun Historia-osioon kävimme ottamassa kuvia elävöittääksemme sivuston ulkoasua. Kuvat on otettu Kajaanin tärkeimmistä nähtävyyksistä. Muokkasimme kuvia Paint Shop Pro-ohjelmalla, jotta ne näyttäisivät vanhanaikaisilta. Lisäksi kuvat pakattiin WinZip-ohjelmalla. Näin ollen kuvat latautuvat paremmin ja nopeammin, kun sivu aukaistaan selaimella.



Kuva 8. Hallintisivun historia-osio

Hallintisivun "Muokkaa kuvia"- linkin alla olevat kuvat ovat tehty pikkukuvina taulukkoon. Taulukkoon voidaan lisätä ja poistaa uusia kuvia tekemällä pikkukuvasta linkki sitä vastaavalle isolle kuvalle.



Kuva 9. Hallintisivun kuvia-osio

FCKeditor

FCKeditor on visuaalinen HTML-muokkain, tunnettu myös nimellä WYSIWYG-editori. FCKeditor tekstimuokkaimen avulla voit muokata sivujen sisältöjä kuten millä tahansa tekstinkäsittelyohjelmalla ilman HTML-kielen koodausta tai osaamista. Sillä voidaan helposti lisätä mm. kuvia tekstisisältöjen sekaan. FCKeditorissa on kaikki ominaisuudet, joita tarvitaan tekstinkäsittelytyökalulta. Editorissa on työkalupalkki, jonka alla on WYSIWYG-tekstialue. FCKeditori toimii melkein millä selaimella tahansa ja missä ympäristössä tahansa. (Mohr, J. 2005.)

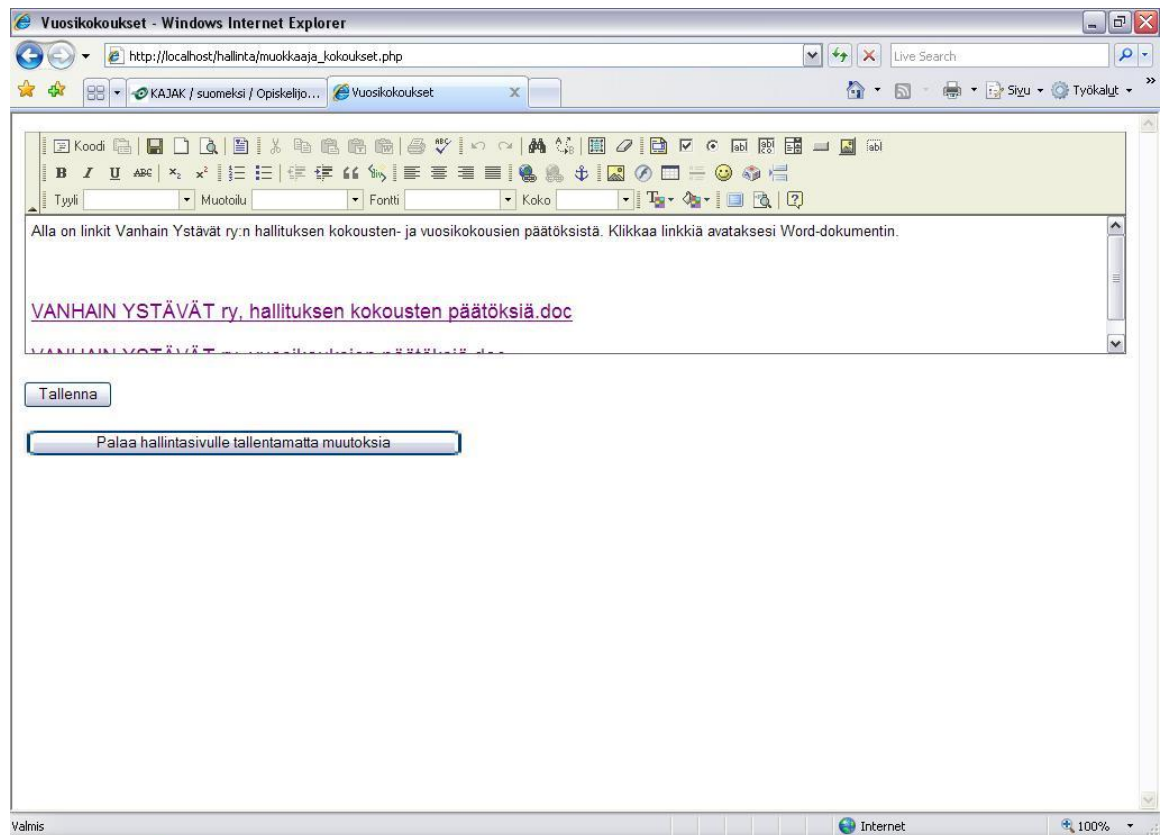
Editori toimii samalla tavalla kuin kaikki muutkin tekstieditorit. Esimerkiksi, voidaan muuttaa tekstin fonttia, fonttikokoa ja tekstin väriä. FCKeditorin työkalupalkista löytyy myös helposti painikkeet, joilla voidaan lisätä ja muokata kuvia, flash-animaatioita sekä luoda linkkejä. Lisäksi työkalupalkissa on muun muassa painike, josta näkyy HTML-lähdekoodi ja sitä voidaan siinä muuttaa. (Mohr, J. 2005.)

Yksi FCKeditorin parhaita ominaisuuksia on taulun luominen ja muokkaaminen. Siinä voidaan määritellä muun muassa, kuinka monta riviä ja saraketta tauluun tulee. Kun taulu on lisätty, sen rajoja voidaan pienentää tai suurentaa. Pystytään myös lisäämään tai poistamaan rivejä ja sarakkeita ja ne voidaan jakaa kahtia. Tauluun on mahdollista lisätä kuvia ja muuttaa sen taustaväriä. (Mohr, J. 2005.)

Tekstin muokkauksen jälkeen käyttäjän täytyy painaa Tallenna-nappia, jolloin muokattu teksti tallentuu automaattisesti kyseiselle sivulle. Toiminta-ajatus ja Vuosikokouksien päätökset on tehty linkeiksi, joita klikkaamalla dokumentit avautuvat. Tekstieditorissa linkkejä on helppo lisätä, muokata ja poistaa.

FCKeditor osoittautui käteväksi ja helposti opittavaksi. Tekstin lisääminen ja muokkaaminen oli helppoa. Editorissa oli lisäksi yllättävän monipuolisia työkaluja, kuten esimerkiksi painik-

keiden ja flash-animaatioiden lisääminen. Näitä ei tosin tarvittu sivustolla. Tosin kuvien asettelu oli todella vaivalloista. Aikaa meni paljon ennen kuin kuva asettui oikeaan kohtaan.



Kuva 10. Vuosikokouksien päätökset FCKeditorissa

6 TESTAUS

Toteutettu tietojärjestelmä täytyy testata ennen käyttöönottoa. Testauksen tarkoituksena on löytää tietojärjestelmästä virheitä. Testaamalla saatetaan löytää virheitä, ja voidaan arvioida testatun järjestelmän toiminnallisuuden luotettavuutta, käytettävyyttä jne. Mutta testaamalla sovelluksen laatu ei parane – tietämys sovelluksen laatusosasta sen sijaan kyllä kasvaa. Ohjelmiston laatu paranee vasta kun testaajien raportoimat virheet on korjattu, ja korjaukset on uudelleentestattu. Laadun parantaminen vaatii siis sekä kehittäjien että testaajien työpanosta. (Patton 2006, 22.)

Testaukseen kuuluu suunniteltujen testien suorittamista toteutetulla sovelluksella. Tämän jälkeen testauksen tulokset ja virheet raportoidaan. Kun havaitut virheet on korjattu, suunnitellaan uusi testitapaus. (Patton 2006, 22.)

Testauksen avulla palaute sovelluksen laadusta, virheistä tai virheettömyydestä saadaan aina jälkikäteen. Toisaalta testauksen huomioiminen ja testausvaiheiden aloittaminen jo sovelluksen varhaisessa kehitysvaiheessa antaa mahdollisuuden arvioida laatua ja löytää virheitä myös määrittely- ja suunnitteluvaiheissa. Testaajien osallistuminen katselmoiteihin, testitapausten suunnittelu määrittely- ja suunnitteluvaiheissa ja pöytätestit tärkeimmille toiminnallisuuksille ovat keinoja aikaistaa testausta. (Patton 2006, 26.)

Vaikka testaus ei löydä virheitä, se ei välttämättä tarkoita, että virheitä ei ole. Testauksen tavoitteena saattaa olla, että ohjelmisto on virheetön. Tällaisessa tapauksessa tavoitteen toteuttamiseksi tarvittava työmäärä kasvaa yksinkertaisessakin sovelluksessa yli mahdollisuuksien. (Patton 2006, 27.)

6.1 Käytettävyystestaus

Käytettävyyden testaus on perusteellinen tapa parantaa Internet-sivujen käytettävyyttä. Käytettävyyden testaamiseen valitaan oikeanlainen testiryhmä ja hankitaan asiantuntija-arvio. Testiryhmän henkilöt käyttävät sivuja tavallisten asiakkaiden tapaan ja asiantuntija seuraa heidän toimintaa. Tilanteet tallennetaan myöhempää analysointia varten. Käytettävyysoongelmia saadaan selville analysoimalla noin 5-6 erityyppisen testikäyttäjän toimenpiteet ja suulliset

kommentit. Näin löydetään kaikki käytettävyysongelmat, jonka jälkeen niihin voidaan puutua. Käytettävyystestauksen perusteella tehdään raportti testin tuloksista ja ehdotukset niiden ratkaisemiseksi. (TukTuk Media 2007)

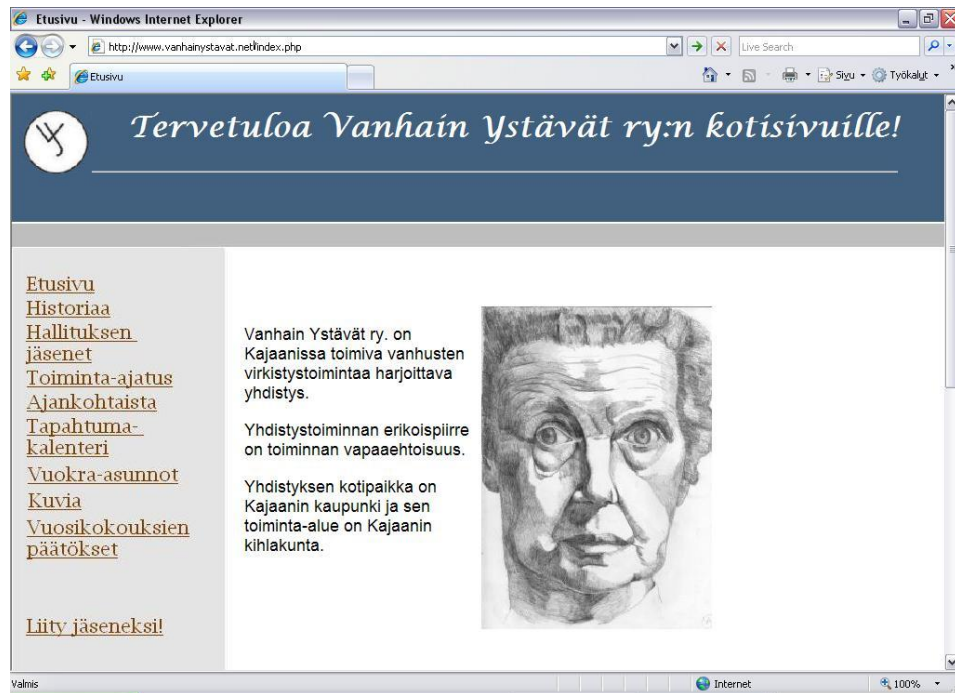
Erityisesti järjestelmän ensimmäisillä käyttökertoilla eteen tulevia ongelmia saadaan hyvin selville käytettävyystesteillä. Käytettävyystestissä testikäyttäjä tekee itsenäisesti testitehtäviä yrittäen käyttää testattavaa järjestelmää parhaaksi katsomallaan tavalla. (Interakta)

Käyttäjätestausta suunniteltaessa kannattaa huomata, että tulokset painottuvat vahvasti ensimmäisten käyttökertojen kokemuksiin, ja pitkäaikaisessa käytössä ilmenevät vaikeudet voivat jäädä piiloon. Järjestämällä käyttäjätarkkailuja, joissa olemassa olevan järjestelmän käyttöä tarkkaillaan muutaman tunnin ajan todellisessa käyttöympäristössä, saadaan selville myös pitkäaikaisen käytön ongelmia. (Interakta)

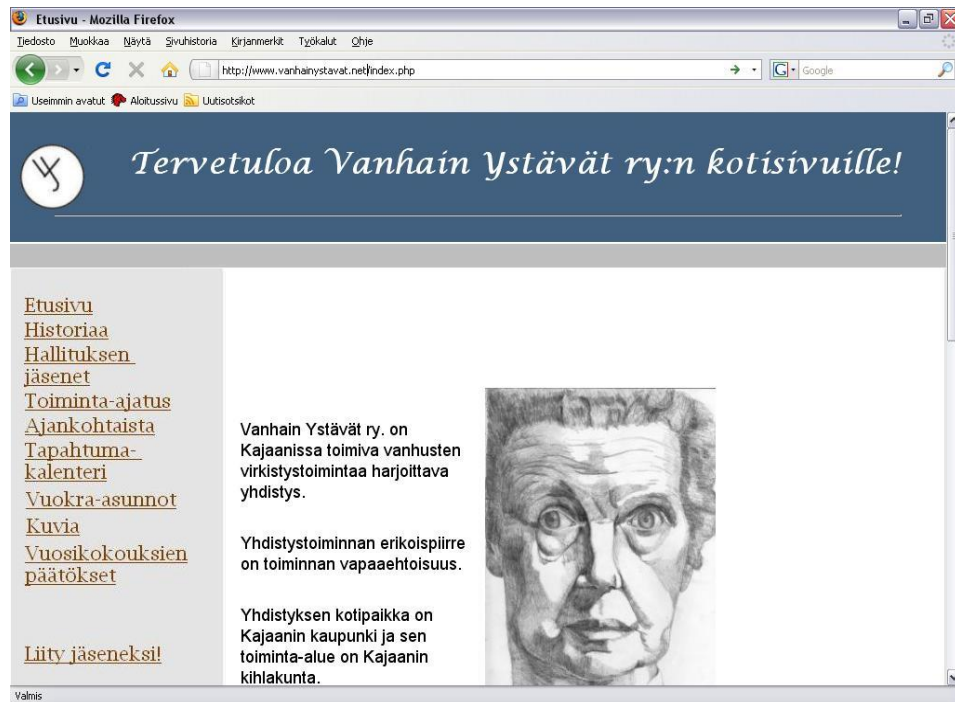
6.2 Sivuston testaus

Sivustoa pyrittiin testaamaan usealla eri käyttäjällä. Testikäyttäjät valittiin eri ikäryhmistä. Näin saimme mahdollisimman hyvät testitulokset. Käyttäjiä pyydettiin testaamaan kuvien ja linkkien toimivuutta. Toiveena oli myös, että he testaisivat sivustoa eri selaimilla, jos se vain oli mahdollista. Saatiin muun muassa seuraavanlaisia kommentteja: "Etusivulla on kerrottu yhdistyksen perustiedot lyhyesti ja ytimekkäästi", "Valikko on selkeä", " Joissakin kohdissa tekstit menevät harmaiden reunapalkkien päälle, jos näytön tarkkuus on 1280 * 1024 kuvapistettä tai suurempi", "Erinomaiset kotisivut", " Hyvin tuotu asioita esille, toiminta-ajatuksineen ja kuvagallarioineen".

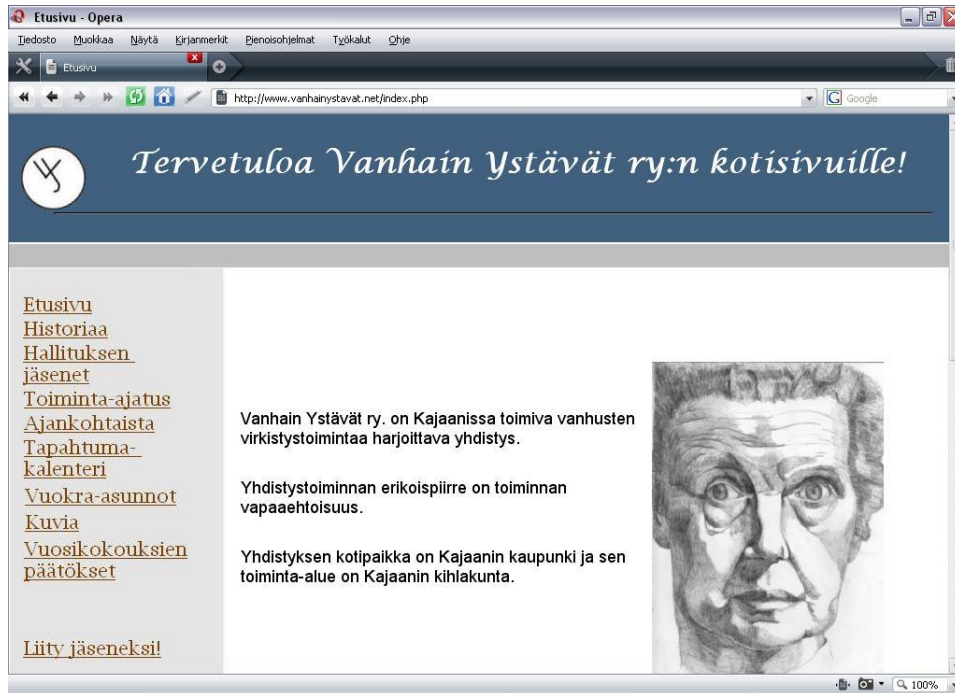
Sivustoa testattiin myös kolmella eri Internet-selaimella. Käytetyt selaimet olivat Internet Explorer, Mozilla Firefox sekä Opera. Selvisi, että jokaisella selaimella sivut näyttivät hieman erilaisilta. Oletusselaimena oli Explorer, jota käytettiin myös sivujen tekovaiheessa. Alla olevista kuvista voi huomata, että Explorerissa etusivun kuva ja teksti ovat niin kuin alun perin suunniteltiin. Firefox-selaimessa teksti ja kuva selvästi putoaa alaspäin ja, jos tarkasti katsoo, niin voi huomata, että otsikkopalkissa oleva logo ja viiva ovat myöskin pudonneet hieman alaspäin. Kaikista erilaisimmalta etusivu näyttää Opera-selaimessa. Kuva on pudonnut alaspäin ja kohdistunut oikeaan reunaan. Tästä johtuen myös teksti on levinnyt. Otsikkopalkin logo on pudonnut alaspäin kuten Firefoxissa ja otsikon alaviiva näyttää mustalta.



Kuva 11. Vanhain Ystävät ry:n kotisivujen etusivu Internet Explorer-selaimessa



Kuva 12. Vanhain Ystävät ry:n kotisivujen etusivu Mozilla Firefox-selaimessa



Kuva 13. Vanhain Ystävät ry:n kotisivujen etusivu Opera-selaimessa

7 TYÖVÄLINEET

Tässä osiossa käsitellään työssä käytettyjä työvälineitä. Skriptikielistä PHP:tä on tarkasteltu eniten, koska sitä käytettiin sivuston rakentamiseen. Muita skriptikieliä on esitelty lyhyesti ja niitä on vertailtu toisiinsa. Tietokannoista MySQL oli eniten esillä, mutta käsiteltiin myös muita vastaavia tietokantajärjestelmiä.

7.1 Macromedia Dreamweaver

Dreamweaver on ammattilaiskäyttöön tarkoitettu tekstipohjainen html-editori. Siinä on mahdollista tarkastella tehtyä työtä ilman, että sitä tarvitsee siirtää palvelimelle tai käynnistää Internet-selainta. Tämä lieneekin syy sen suureen suosioon erityisesti juuri ammattilaiskäytössä. Macromedia julkisti ensimmäisen version Dreamweaverista vuonna 1998. Sen jälkeen vuosien saatossa Dreamweaverista on julkaistu paranneltu ja uusia ominaisuuksia sisältävä versio lähes vuoden välein. (2007 Net Site Story)

Vanhain Ystävät ry:n kotisivut on tehty käyttäen Dreamweaver 8:aa. Kotisivujen tekoon valittiin Dreamweaverin, koska se tukee PHP:tä, skriptikieltä, jota käytettiin työssä. Dreamweaver 8 julkaistiin 2005 ja sen ominaisuuksiin kuuluu mahdollisuus tehdä niin dynaamisia kuin staattisia verkkosivuja. Dreamweaver 8 tukee myös, ASP:tä, JSP:tä sekä ColdFusionia. Sivuja tehtäessä käyttäjä voi valita suunnittelu-näkymän, koodi-näkymän tai näkymän, jossa näkyy kumpikin, suunnittelu- sekä koodinäkö.

Dreamweaver 8:sta löytyy helposti kuvakkeina yleisimmin ja usein web-sivuilla käytetyt komponentit kuten hyperlinkki, sähköpostilinkki, taulun-, kuvan ja median sijoittaminen jne. Myös kaikki tekstinkäsittely toiminnot kuten fontti ja fontin koko ovat ohjelmasta helposti löydettävissä. Dreamweaverin suurimpia heikkouksia on tekstin muokkaaminen ja asettelu. Ohjelmassa on erillinen Templates: editable region- ominaisuus. Sillä voidaan tehdä tekstialue niin, että teksti ei voi mennä sivun muihin kohtiin vaan se sijoittuu sille varatulle alueelle. Jos ei käytä Templates- valintaa, ohjelma tekee tekstin sijoittamisesta ja asettelusta vaikeaa. Aloittelijoina huomasimme tämän Templates-ominaisuuden vasta aivan työn loppupuolella, joten siitä koitui yllättävänkin suuria ongelmia.

Linkkipalkki on tehty käyttämällä ohjelman CSS styles- valintaa. Se on tehty niin, että se asetuu jokaiselle sivulle vain ja ainoastaan vasempaan reunaan.

7.2 PHP

PHP (Hypertext Preprocessor) oli alun perin kokoelma www-pohjaisten sovellusten tekemistä helpottavia rutiineja, eräänlainen palvelimella ajettavien CGI-ohjelmien tekemistä helpottava komentokokoelma. Rasmus Lerdorf aloitti PHP:n kehitystyön vuonna 1994. Ensimmäisestä versiosta hän käytti nimeä "Personal Home Page Tools". Seuraavan version nimenä oli PHP/FI (Personal Home Page / Forms Interpreter). Version PHP/FI 2.0 jälkeen siirryttiin käyttämään lyhennettä PHP (rekursiivinen lyhenne sanoista PHP: Hypertext Preprocessor). (Rantala 2005, 9.)

PHP on ensisijaisesti työväline dynaamisten web-dokumenttien luomiseen. Laajasti ymmärrettynä PHP viittaa sekä varsinaiseen PHP-kieleen että niihin tekniisiin ratkaisuihin, joilla PHP-kielisten ohjelmien suorittaminen on mahdollista. Kokonaisuutena voidaan puhua PHP-ympäristöstä. PHP on suunniteltu erityisesti web-sovellusten kehitystä varten. Tämän vuoksi PHP mahdollistaa ammatillisille web-sovellusten nopean kehittämisen ja vasta-alkajille matalan aloituskynnyksen: jo pienen opiskelun seurauksena voi saada aikaan jotain hyödyllistä. (Rantala 2005, 9.)

Perinteisiä CGI-ohjelmia tehtiin siten, että ohjelmointikielen komentojen avulla tuotettiin myös sivulla tarvittavat html-merkkaukset, kun taas PHP toimii päinvastoin. PHP-kielen komentoja voidaan kirjoittaa mihin tahansa kohtaan html-koodia tai mitä tahansa www-palvelimella olevaa tekstimuotoista tiedostoa. (Heinisuo 2007, 13.)

PHP on tulkittava kieli, eli www-sivun sisällä oleva PHP-koodi ajetaan joka kerta, kun www-palvelin lähettää sivun selaimelle. PHP voi toimia Apache-palvelimen sisäisenä moduulina, jolloin se on kiinteä osa www-palvelinta. Tämän kytköksen ansiosta PHP-koodia sisältävien sivujen ajaminen on suhteellisen nopeaa, sillä erillisen tulkin käynnistämistä ulkoisten CGI-

ohjelmien tapaan ei tarvita. PHP-koodi ajetaan aina palvelimella juuri ennen kuin sivu lähetetään selaimen. (Heinisuo 2007, 13.)

PHP on avoin ohjelmisto ja sen lähdekoodi on vapaasti saatavilla ilmaiseksi. Vaikka PHP:tä käytetään yleisesti juurikin Apache-palvelimen kanssa, PHP:tä voi käyttää aivan hyvin myös esim. Windows-alustalla hyödyntäen vaikkapa Access-tietokantaa.

PHP:n viimeisin stabiili versio on 5 (5.04). PHP5 on tuonut mukanaan merkittäviä parannuksia erityisesti laajojen järjestelmien toteuttamista varten. Erityisesti ovat kehittyneet oliopiiirteet, poikkeusten käsittely, joka mahdollistaa hyvinkin laajojen järjestelmien toteuttamisen sekä xml-muotoisen datan käsittely. (Rantala 2005, 9.)

7.2.1 Miksi valita PHP?

Erilaisia tekniikoita dynaamisten web-sovellusten rakentamiseen on lukuisia. PHP:n etuna on, että se on suunniteltu nimenomaan web-sovellusten rakentamiseen. PHP:n aloituskynnys on matala eli sen perusrakenne html-dokumenttien sisään upotettavana skriptikielenä on yksi helpoimmin opittavista. PHP on heikosti tyypitetty kieli: muuttujien tyyppiä ei tarvitse määrittellä, vaan tyyppi määräytyy sen mukaan, millaista dataa ja millaisin operaatioin muuttujaan sijoitetaan. PHP-kielen syntaksi on lainattu pääosin C-kielestä, joten jos on ohjelmoinut aiemmin C kielellä, PHP on helppo omaksua. (Rantala 2005, 10.)

PHP on tehokas ja sen syntaksi on monipuolinen ja PHP toimii Apachen moduulina erityisen tehokkaasti. Aliohjelmia on tarjolla runsaasti, esimerkiksi eri tietokantojen käyttöä varten. PHP on avoimena ohjelmistona ilmaiseksi saatavilla useille käyttöjärjestelmille ja web-palvelintoteutuksille. PHP:n suosio on nykyään suuri, joten sitä tukevia web-hotelleja on runsaasti tarjolla. Myös muuta dokumentaatiota ja apua löytyy verkosta paljon. (Rantala 2005, 10.)

7.2.2 PHP:n toiminta ja skriptit

PHP-koodi kirjoitetaan tekstinä, ja PHP-tiedostot odottavat palvelimella valmiina ryhtymään töihin, kun niitä kutsutaan. Käytettyjen tiedostojen tunnistus on .php- muotoa. Sitten selaimen käyttäjä pyytää dokumenttia, jonka tunnistus on .php. Tämän palvelin tunnistaa, että kyseessä on PHP-tiedosto ja lähettää pyynnön PHP-jäsentimelle. PHP-jäsentin löytää pyydetyn tiedoston ja hakee siitä PHP-koodit. Kun PHP-jäsentin löytää PHP-koodia, se suorittaa koodin ja sijoittaa tulosteen tiedostossa sille paikalle, jossa koodi oli. (Meloni 2003, 294.)

Tämän jälkeen uusi tulostetiedosto lähetetään takaisin palvelimelle ja palvelin lähettää tiedoston selaimelle. Lopuksi selain näyttää tulosteen. Tässä oli PHP:n toiminta lyhykäisyydessään. (Meloni 2003, 294.)

Tekijän kannalta PHP-skriptit näyttävät ihan tavallisilta html-dokumenteilta sillä erolla, että mukana on (upotettuna) tietyillä tageilla erotettuina PHP-kielisiä komentoja. PHP-osuus on erotettu html-koodista tagein `<?php ja ?>`. `<?php` tarkoittaa PHP-koodin aloitusta ja `?>` on PHP-koodin lopetus-tagi. (Rantala 2005, 17.)

Seuraavaksi pieni esimerkki PHP-koodista:

```
< ?php  
echo "PHP on helppo oppia";  
? >
```

Tagien rajoittaman alueen sisällä on ainoastaan yksi PHP-kielinen komento `echo`, joka toimii tässä tulostuskomentona. Puolipiste rivin lopussa on (komento)lauseen lopetusmerkki. Kun sivulla oleva koodi suoritetaan, näkyy selaimessa ainoastaan ohjelman tulostus. Yllä olevassa esimerkissä selaimen tulostuksi siis ainoastaan teksti "PHP on helppo oppia". PHP-koodi on nähtävissä ainoastaan silloin, kun käsitellään palvelimella olevia tiedostoja esimerkiksi FTP:n tai tiedostojen jaon kautta. Samasta syystä PHP-koodia sisältävät sivut eivät toimi, jos käyttäjä avaa tiedoston suoraan tietokoneen levyiltä – Toisin sanoen PHP suoritetaan `www-` palvelimessa, ja sivut on ladattava `www-` palvelimelta.

PHP sisältää kaikki ohjelmointikielille tyypilliset rakenteet eli esimerkiksi for- ja while-silmukat, if ehtolauseet sekä muuttujat ja funktiot. PHP on myös oliopohjainen, eli sovellukset voidaan haluttaessa toteuttaa luokkina perinteisen funktioihin perustuvan ohjelmoinnin sijaan. (Heinisuo 2007, 13.)

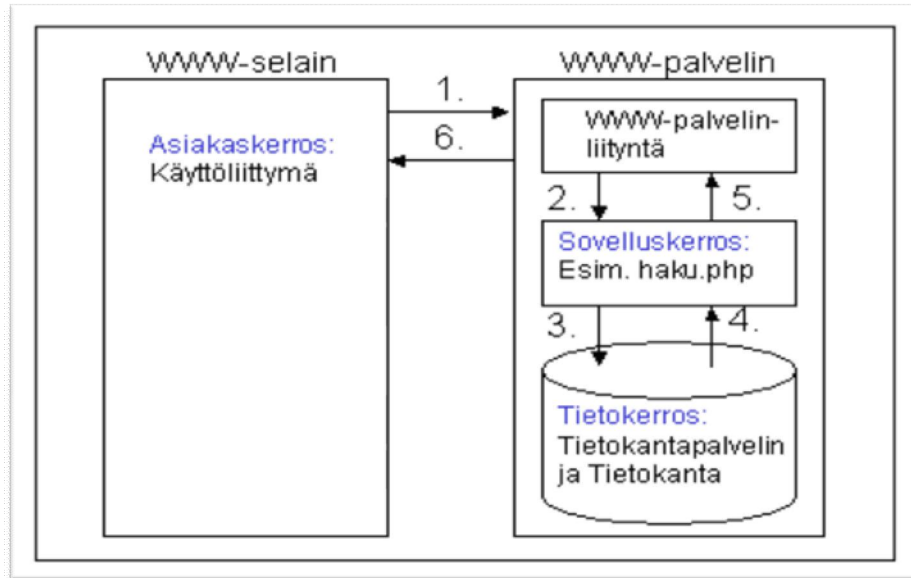
7.2.3 PHP ja MySQL

Usein Web-sovellusten tiedot ovat palvelimella ja relaatiotietokannoissa. Tekstiedostot yleensä riittävät moniin tarkoituksiin, kuten yksinkertaisen vieraskirjan tietojen tallennukseen.. Tietyissä tapauksissa relaatiotietokannat ovat kuitenkin välttämättömiä ja parhaita välineitä tiedon hallintaan. Esim. tehokas tiedon haku ja lajittelu vaativat yleensä tietokannan. Myös suurin osa merkittävistä www-palveluista on tietokantapohjaisia.

7.2.3.1 Asiakas-palvelin –malli

Asiakas-palvelin – mallissa tietokannan sisältävät tiedot ovat tietokantapalvelimella, josta ne haetaan ja jonka avulla ne käsitellään. Tietokantapalvelin on tietokantaohjelmiston mukana tuleva palvelin, ja sen tarkoituksena on välittää tietoja sovellusohjelmasta tietokantaan ja toisinpäin. Palvelinkoneella voi siis pyöriä kaksi palveliohjelmää: www-palvelin kuten Apache ja tietokantapalvelin (esim. MySQL-tietokantapalvelin). Tietokannat ja tietokantapalvelin voivat myös sijaita fyysisesti omalla palvelinkoneellaan. (PHP ja MySQL.)

Tietokantaisessa www-sovelluksessa on yleensä kolme ”kerrosta”. Asiakaskerros on ylin kerros; se on tavallisesti www-palvelimella näkyvä html-käyttöliittymä. Sovelluskerros on ohjelma, jonka avulla hoidetaan yhteys tietokantaan ja tulostetaan tietoa asiakkaalle. Tietokerroksessa ovat tiedot ja tietokantapalvelin, joka välittää tiedot sovelluskerrokselle. (PHP ja MySQL.)



Kuva 14. Asiakas-palvelin –mallin toimintakaavio (PHP ja MySQL.)

Kaavio toimii siten, että ensin asiakas eli selain pyytää palvelimelta haku.php –tiedostoa. Sen jälkeen www-palvelinliityntä tunnistaa .php-päätteen ja antaa PHP-koodia sisältävät osat PHP-tulkille suoritettavaksi. Sitten PHP:n tietokantamoduulin avulla otetaan yhteys tietokantapalvelimeen. Tiedostossa haku.php on SQL-kysely, jonka avulla halutut tiedostot haetaan kannasta. Kun tiedot on haettu, tietokantapalvelin lähettää tiedot ohjelmalle. Lopuksi ohjelma palauttaa www-palvelinliitynnän kautta asiakkaalle html-sivun, jossa PHP-koodi on suoritettu ja tietokannasta haetus tiedot ovat mukana. (PHP ja MySQL.)

Vanhain Ystävät ry:n sivustolla MySQL-tietokantaan on tehty "Hallituksen jäsenet", "Vuokra-asunnot" ja "Jäsenet". Dreamweaverissa on ominaisuus nimeltä Recordset, josta voidaan hakea ohjelmaan tietokannat ja niiden taulut. Tämä edellyttää, että MySQL-tietokantayhteys toimii. Seuraavassa on sivuston liittymislomakkeen php-koodi, jossa uuden jäsenen tiedot siirtyvät tietokantaan:

```
<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{
```

```

die('Could not connect: ' . mysql_error());
}
mysql_select_db("Kotisivut", $con);
$sukunimi = strip_tags(rawurldecode($_POST["sukunimi"]));
    $etunimet = strip_tags(rawurldecode($_POST["etunimet"]));
    $osoite = strip_tags(rawurldecode($_POST["osoite"]));
    $puhno = strip_tags(rawurldecode($_POST["puhno"]));
    $sposti = strip_tags(rawurldecode($_POST["sposti"]));
mysql_query("INSERT INTO jaset2 (sukunimi, etunimet, osoite, puhno, sposti)
VALUES ('$sukunimi', '$etunimet', '$osoite', '$puhno', '$sposti)");
mysql_close($con);
?>

```

Koodista selviää, että ensin otetaan yhteys paikalliseen palvelimeen (localhost) ja MySQL-käyttäjänimi(root). Kun yhteys on muodostettu, valitaan Kotisivut-tietokanta ja lisätään arvot jaset2-tauluun.

7.2.4 PHP6

PHP-kielen seuraavan version ominaisuuksista on olemassa vähän oikeaa tietoa ja paljon arvailuja. PHP-kielen kehittäjät eivät ole vielä varmuudella lyöneet lukkoon sitä, mikä tulee muuttumaan ja mikä tulee pysymään ennallaan. Varmaa on kuitenkin se, että PHP6 rajoittaa ohjelmoijan vapauksia toteuttaa asioita tietoturvatomasti. Samalla tietysti oppimiskäyrä jyrkenee; vasta-alkajilla tulee olemaan vaikeampaa päästä toteuttamaan itseään ilman kunnon opiskelua ja perehtymistä.

Perusvireenä muutoksissa tuntuu olevan niiden asioiden perusteellinen korjaaminen, jotka ovat herettäneet eniten PHP-vastustusta ohjelmoijien parissa. Itse ohjelmointirakenteiden ja menetelmien mielessä harppaus viidestä kuuteen ei ole kovin merkittävä, eli PHP4 ja 5:lle tehdyt ohjelmistot tulee toimimaan suhteellisen mutkattomasti myös PHP6:ssa. (Heinisuo 2007, 396.)

7.2.5 PHP ja turvallisuus

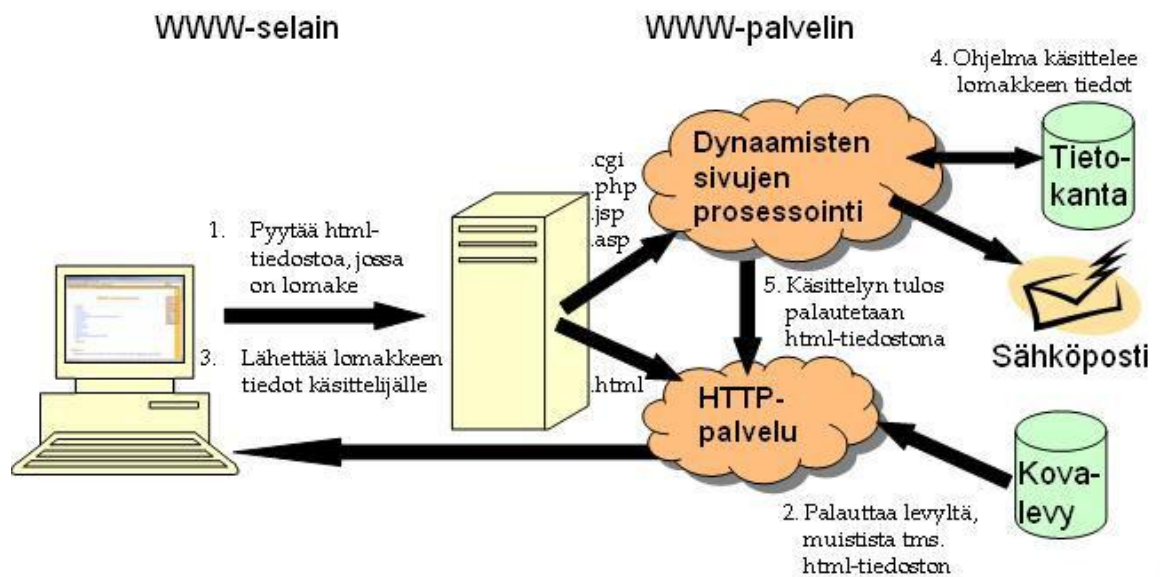
WWW-sovelluksissa tietoturva on erityisen tärkeä asia, sillä sovellukset ovat monesti koko Internetin nähtävillä. Tämä tarkoittaa sitä, että kuka tahansa pääsee käsiksi sovellukseen ja jos sovellus ei ole tarpeeksi turvallinen. Tämä mahdollistaa sovelluksen väärinkäytön joskus jopa vakavinkin seurauksin. Tietoturvaohje ei koske yleensä pelkkää sovellusta, vaan myös sen käyttäjiä ja heidän sinne tallentamiaan tietoja. Pienikin virhe voi paljastaa luottamuksellista tietoa sellaisille tahoille, joiden ei kyseistä tietoa pitäisi päästä näkemään. (Heinisuo 2007, 399.)

Niin PHP:llä kuin muillakin ohjelmointikielillä toteutetuissa sovelluksissa ohjelma saa syötettä ja antaa useimmiten myös jonkinlaisen palautteen muun muassa saadun syötteen perusteella. Sovellus voi myös hakea tietoa itselleen jostakin ulkoisesta lähteestä.

PHP:ssä syötteenä voidaan laskea \$-GET, \$-POST ja \$-COOKIE-taulukot sekä luonnollisesti niiden kaikkien tiedot sisältävä \$-REQUEST-taulukko. Palaute tai vastaus on se, mitä sovelluksen sivu lähettää selaimelle takaisin. Ulkoinen lähde tiedolle on esimerkiksi tietokanta tai tiedostojärjestelmässä olevat tiedostot. Myös \$-SESSION-taulukossa olevat istunto-muuttujat voidaan pienellä varauksella laskea jälkimmäiseen kategoriaan. Tiedostojärjestelmään kuuluvat yleisest myös tiedon tallennuspaikat, ja monesti kyseessä onkin sovelluksen omassa käytössä oleva tietovarasto. (Heinisuo 2007, 399.)

7.3 ASP

ASP (Active Server Pages) on Microsoftin kehittämä tekniikka, jolla voidaan luoda dynaamista tietoa sisältäviä www-sivuja. ASP toimii samalla periaatteella kuten esimerkiksi PHP. Webiselain lähettää lomakkeen palvelimelle, jossa suoritetaan haku tietokantaan. Tämän tehtyään palvelin lähettää käyttäjälle selaimen www-sivun, jossa on haun tulokset halutussa muodossa. (Peltomäki 2000, 3-4; Ek, Arvidsson & Eriksson 2000, 4-5.)



Kuva 15. Lomakkeen käsittely selaimen ja palvelimen välillä (Lomakkeiden käsittelytekniikat 2004.)

ASP-sivuja voidaan toteuttaa useilla eri skriptikielillä, joista yleisempiä ovat Visual Basiciin perustuva VBScript, JScript ja Perl. Käyttämällä erityisiä erottimia `<%` ja `%>` voidaan ASP-koodi erottaa tavallisesta html-koodista. ASP-koodia voidaan upottaa joko html-koodin lomaan tai sen ulkopuolelle. Käyttäjät eivät näe palvelinskriptiä sellaisenaan, sillä se on asiakkaan saavuttaessaan muuttunut html-koodiksi, aivan kuten PHP:ssä. (Ek, Arvidsson & Eriksson 2000, 7-9; SoftwareProjects.)

Alla on kuvattu hyvin yksinkertainen VbScriptillä toteutettu ASP-sivu, joka tulostaa tekstin "Hei Maailma!" kolme kertaa. Koodi siis sijoitetaan tagien väliin, jonka sisältö käsitellään palvelimella suoritettavana komentosarjajonona. Palvelin suorittaa komentosarjan ja lähettää käyttäjälle vastauksena pelkän html-koodin.

```
<%@ LANGUAGE=VBScript %>
<% Dim i %>
<html>
<body>
<% For i = 1 To 3 %>
```

```
Hei Maailma!<br>  
<% Next %>  
</body>  
</html>
```

ASP ja PHP muistuttavat tekniikaltaan hyvin paljon toisiaan, mutta erojakin löytyy. ASP on todella suosittu yritysmaailmassa, koska se on Microsoftin kehittämä tuote. Yksityiskäyttäjien keskuudessa PHP on paljon suosittu tekniikka sen ilmaisuuden ja avoimen lähdekoodin ansiosta. PHP on myös helpompi ohjelmointikieli oppia kuin ASP-sivujen kehityksessä suosittut Perl-, VBScript- ja JScript-kielet. Tietenkin asiaan vaikuttaa se, mihin ohjelmointikieleen kukin käyttäjä on mieltynyt. ASP:ssä webpalvelimena käytetään Microsoftin IIS-palvelinta, joka ei ole läheskään saavuttanut niin suurta suosiota tavallisten käyttäjien keskuudessa kuin esimerkiksi PHP:llä usein käytetty Apache. (Jackson, B. 2007.)

Käyttötarkoitukset ovat myös pääasiassa erilaiset molemmilla tekniikoilla. ASP voi olla parempi tekniikka kun toteutetaan verkkokauppasovelluksia, kun taas PHP on suosittu ohjelmointikieli esimerkiksi keskustelupalstojen toteutuksessa. Toki näitä molempia sovelluksia voidaan tehdä kummalla tahansa tekniikalla. Molemmat dynaamisten web-sovellusten toteutukseen pystyvät tekniikat säilyttävät varmasti tulevaisuudessa suosionsa ohjelmoijien keskuudessa. ASP-pohjaiset alustat ovat tosin väistymässä uuden ASP.net version myötä, joka perustuu Microsoftin .net-arkkitehtuuriin. (Jackson, B. 2007.)

7.4 Java Server Pages

JSP (Java Server Pages) on Sunin vastaus Microsoftin ASP-tekniikalle. JSP:ssä koodia voidaan kirjoittaa suoraan html-lomakkeelle. Koodi suoritetaan palvelimella ennen sivun lähettämistä asiakkaalle. JSP muistuttaa hyvin paljon muita web-ohjelmoinnin ratkaisuja, jossa koodia sisältävät html-dokumentit erotetaan muista html-tiedostoista erillisen päätteen avulla: JSP-sivuissa sivut merkitään jsp-merkinnällä. Koodi kirjoitetaan JSP-tiedostoissa <%- ja %>-tagien väliin. (Peltomäki 2000, 644.)

Alla on yksinkertainen esimerkki JSP-koodin käyttämisestä html-dokumentissa. Kyseinen koodi tulostaa html-dokumentin ladattuaan näytölle tekstin "Hei Maailma!".

```

<html>
<head><title>JSP-esimerkki</title></head>
<body>
<h2>
<%
                                out.println("Hei Maailma!");
%>
</h2>
</body>
</html>

```

Kun koodi suoritetaan ensimmäisen kerran, voi sivun lataus kestää yllättävän pitkään lyhyestä koodista huolimatta. Syynä tähän ovat servletit. Servletit ovat sovelluspalvelimessa ajettavia Java-ohjelmia, jotka voidaan käynnistää webpalvelimen kautta. Servletti muodostuu yhdestä pääluokasta ja mahdollisista apuluokista. JSP-sivu muutetaan ensimmäisellä suorituskerralla normaaliksi servletiksi, joka käännetään ja tulkitaan. Seuraavilla suorituserroilla lataaminen onkin jo nopeampaa, koska servletti jää palvelimelle muistiin. (Peltomäki 2000, 645.)

Miksi sitten JSP-tekniikkaa kannattaa käyttää web-sivustojen toteuttamisessa servlettien sijaan? Ensinnäkin JSP antaa vapauden erottaa html/xhtml Java-koodista. Jos Servletin avulla halutaan tulostaa html-rakenteita, on ne tulostettava Javan avulla. JSP:ssä html-merkintä voidaan liittää suoraan JSP-malliin. Lisäksi kun sivustoon tehdään muutoksia, sitä ei tarvitse kääntää erikseen kuten servleteissä, koska palvelin hoitaa JSP-tiedoston kääntämisen servletiksi automaattisesti ensimmäisen kutsun yhteydessä. (2K mediat b.)

7.5 CGI-ohjelmat

Common Gateway Interface eli CGI on Internet-palvelujen yhteydessä käytettävä rajapinta, jonka avulla voidaan luoda interaktiivisia kotisivuja. CGI-ohjelmia voidaan toteuttaa useilla eri ohjelmointikielillä. Perl on ollut suosituin ohjelmointikieli CGI-ohjelmien teossa, mutta viime vuosina C/C++ sekä Php-ohjelmointikielten käyttö on lisääntynyt runsaasti. (Ek, Arvidsson & Eriksson 2000, 2-3.)

CGI-ohjelmassa web-selain ottaa yhteyden palvelimeen ja kertoo haluamansa sivun nimen. Palvelin lähettää sivun selaimelle, jonka jälkeen yhteys sulkeutuu automaattisesti. Tämä on yksinkertaisin perustapaus, sillä palvelimet voidaan myös määrätä suorittamaan sivun lähettämisen yhteydessä tietty sovellus, joka on CGI-ohjelma. (Ek, Arvidsson & Eriksson 2000, 2-3.)

Perl ja CGI-ohjelmat

Perl on ollut suosituin CGI-ohjelmointikieli, vaikka uusia tekniikoita on viime vuosina tullut runsaasti markkinoille. Perl on usein yhdistetty samaksi asiaksi CGI-ohjelmoinnin kanssa, mutta CGI-ohjelmia voi tehdä muillakin vastaavilla ohjelmointikielillä. Perl on ollut suosittu kieli sen ominaisuuksiensa ansiosta. Sillä voidaan helposti tehdä korkeatasoisia dynaamisia sivustoja. Lisäksi sen tekstinkäsittelyominaisuudet on arvostettu korkealle. Perlin etuna on myös, että se pystyy selviytymään pienistä ja keskisuurista ohjelmista tehokkaasti. (Peltomäki 2000, 214; Peltomäki 2001, 4-5.)

Perlin yhtenä haittapuolena on, että se on joissain sovelluksissa suhteellisen hidas sen tulkittavuuden johdosta. Esimerkiksi C-kielillä toteutetut CGI-ohjelmat ovat nopeampia toimimaan, koska siinä on usein valmiina erilaisia luokkakirjastoja, jotka helpottavat sovelluksien kehitystä. Perl-ohjelmien ylläpito tosin on vaivattomampaa, koska Perl-ohjelmoijan ei tarvitse mm. itse huolehtia muuttujien tai olioiden varaaman muistin dynaamisesta vapauttamisesta. (Peltomäki 2000, 215–216.)

Perl sopii moneen eri käyttötarkoitukseen. Dynaamisesti luotavat www-sivut lienevät suurin käyttökohde. Myös erilaiset lokit ja laskurit on helppo toteuttaa Perlin tekstinkäsittelyominaisuuksiensa ansiosta. Perlillä voidaan myös toteuttaa www-käyttöliittymä uutisryhmään tai ftp-palveluun internetpalvelun sijasta. Tällöin pystytään keräämään tietoa myös muista palvelimista kuin pelkästään omasta palvelimesta erilaisten ohjelmien välityksellä. Erilaiset ajanvarausjärjestelmät, verkkokaupat ja kalenterisovellukset ovat olleet myös suosittuja käyttökohteita ohjelmoitaessa Perlillä. (Peltomäki 2000, 217–218.)

Perl ei sen sijaan sovellu ominaisuuksiensa takia käytettäväksi, kun tarvitaan paljon tehoa ja laskentakapasiteettia vaativa CGI-ohjelma. Myös runsaassa käytössä oleva CGI-ohjelma on

hankala toteuttaa Perlillä. Siitä huolimatta ohjelman prototyyppiin voi rakentaa Perl-kieltä käyttäen, koska varsinainen tuotantoon tuleva versio voidaan ohjelmoida esimerkiksi C- tai C++-kielellä. (Peltomäki 2000, 216; Peltomäki 2001, 5-6.)

7.6 ColdFusion

Maailmalla laajan ja monipuolisen käyttäjäkunnan omaava ColdFusion on yksi suosituimmista web-ohjelmointi teknologioista. ColdFusion on erittäin kallis verrattuna moniin muihin vastaaviin web-ohjelmointikieliin, mutta se sisältää monipuolisen valikoiman sisäänrakennettuja piirteitä. Suuren suosion esteenä on kuitenkin ollut vähäinen tunnettavuus, kuten esimerkiksi Suomessa on käynyt. (2K mediat a.)

ColdFusion käyttää ohjelmointikielenä ColdFusion Markup Languagea eli CFML-kieltä, joka muistuttaa vahvasti xml-ohjelmointikieltä. ColdFusion sisältää rajapinnat yleisempiin tietokantoihin sekä monipuolisen virheenkäsittelyn. Lisäksi ColdFusion soveltuu käytettäväksi, kun sivujen toiminnassa ja niiden kehityksessä vaaditaan nopeutta. (2K mediat a.)

ColdFusion sisältää tuen monelle eri tietokantarajapinnalle, jonka ansiosta tietokantayhteyden muodostaminen onnistuu kaikilla alustoilla. Tämän ominaisuuden lisäksi tietoja voidaan hakea erillisen koodin avulla mm. HTML, PDF, DOC -tiedostoista sekä tietokannoista. (2K mediat a.)

7.7 Ruby on Rails

Ruby on dynaaminen, oliopohjainen ohjelmointikieli, joka on hyvin paljon Perl-ohjelmointikielen kaltainen. Rubylla voidaan antaa voimakkaita komentoja vähäisellä koodimäärällä. Ruby on myös hyvin johdonmukainen ja yllätyksetön ohjelmointikieli. Opittuaan kielen periaatteet, ohjelmoijan on hyvin helppo soveltaa sitä erilaisissa tilanteissa. Ennestään ohjelmointia osaavalle Ruby on helppo oppia, koska kieleen on lainattu paljon hyödyllisiä ominaisuuksia muista ohjelmointikielistä. (Wikipedia a.)

Ruby on Rails on avoimeen lähdekoodiin perustuva ohjelmistokehys, joka pohjautuu Ruby ohjelmointikieleen. Rails perustuu MVC-arkkitehtuuriin ja sillä voidaan luoda www-pohjaisia sovelluksia. Muihin www-kehysiin verrattuna Rails vaatii varsin vähän ohjelmakoodia. Rails rakentaa tietokannan taulut lähes kokonaan automaattisesti ja se myös löytää luomansa taulut tietokannasta päätellen mitä sarakkeita siinä on. (Wikipedia b.)

Rails toimii useassa eri käyttöjärjestelmässä ja webpalvelimessa. MySQL, PostgreSQL ja SQLite ovat käytetyimmät tietokannan hallintajärjestelmät Railsissa. Myös kaupallinen Oracle on hyvin suosittu ratkaisu. (Wikipedia b.)

7.8 MySQL

MySQL on laajimmin käytetty relaatiotietokantojen hallintajärjestelmä ja se perustuu avoimeen lähdekoodiin. Joukossa on niin yksityishenkilöitä kuin suuria yrityksiä, jotka käyttävät tietokantoja verkkosovelluksissaan. (Meloni 2003, 11-12.)

MySQL osaa varastoida kaikki tietokannat, taulut, sarakkeet ja rivit. MySQL käsittelee niitä yhtenä kokonaisuutena. Käyttäjille voidaan antaa erikseen käyttöoikeuksia tietokantoihin. MySQL pitää kirjaa kaikkien käyttäjien toimista ja hoitaa myös vastaukset tietokantakyselyihin. Kyselyt ovat SQL-kielellä kirjoitettuja komentoja; niiden avulla tietokantaan voidaan lisätä, poistaa ja muokata tietoja. (Meloni 2003, 11-12.)

MySQL sisältää piirteitä, joiden ansiosta se on saavuttanut suuren suosionsa sovelluskehittäjien keskuudessa. MySQL on hyvin nopea ja se ei ole läheskään niin monimutkainen järjestelmä kuin esimerkiksi Oracle. Yksi suosiota nostattanut tekijä on se, että MySQL toimii lähes jokaisen ohjelmointikielen kanssa (esimerkiksi PHP, Perl C-kielet ja Python). Varsinkin PHP:n ja MySQL:n yhteiskäyttö web-sovelluksien kehittämisessä on suosittua. (Meloni 2003, 12-13.)

Myös käyttöjärjestelmän ja ohjelmointikielen vaihto onnistuu kesken kehitystyön. Tiedot voidaan ottaa talteen käyttämällä erillistä mysqldump-ohjelmaa. Vastaavasti tiedot saadaan

uuteen käyttöjärjestelmään käyttämällä mysqlimport-ohjelmaa. Näiden ominaisuuksien lisäksi MySQL on yleensä saatavilla ilmaiseksi. (Meloni 2003, 12-13.)

Tietokannat tulee suunnitella hyvin, kuten mikä muukin tahansa sovellus. Hyvällä suunnitellulla saavutetaan tietokantojen optimaalinen suorituskyky. Jos tietokantaa ei ole oikein normalisoitu, niin se ei pysty toimimaan tehokkaasti. Normalisoinnin avulla tietokannan tietoja pystytään tallentamaan ehjästi ja tietoja saadaan paremmalla tehokkuudella haettua. (Meloni 2003, 28.)

Pelkkä suorituskyvyn optimointi ei kuitenkaan riitä tietokantojen tehokkaaseen toimintaan. Tietokannan ylläpitämisen täytyy olla mahdollisimman yksinkertaista. Tämän vuoksi toistuvaa dataa pitäisi olla mahdollisimman vähän. Jos jokin tämän datan ilmentymä muuttuu, täytyy muutos tehdä kaikkiin datassa oleviin ilmentymiin. Tämä taas vaikeuttaa ja hidastaa tietokannan ylläpitoa. Tällainen tilanne vältetään tekemällä useita tauluja, jotka sisältävät kaikki mahdolliset arvot. Jos jokin arvo muuttuu tietokannassa, niin muutos tapahtuu vain päätaulussa. Viittaus sen sijaan pysyy samana kaikissa muissa tauluissa. (Meloni 2003, 28.)

Hyvin suunniteltu on puoliksi tehty. Mitä enemmän työskennellään etukäteen tietokannan suhteen, sitä vähemmän tarvitsee käyttää aikaa mahdolliseen korjaustyöhön. Kaikista pahin tilanne on, jos tietokanta joudutaan suunnittelemaan uudelleen sitä käyttävän sovelluksen julkistamisen jälkeen. Tämä aiheuttaa paljon lisäkustannuksia. Suunnitteluun kannattaa käyttääkin runsaasti aikaa, vaikka se tuntuisi työläältä. (Meloni 2003, 29-30.)

Suunnitteluprosessin alkaessa täytyy huolellisesti arvioida tietokannan sisältö, tietojen suhde toisiinsa ja tietokannan mahdollinen laajennettavuus. Ennen kuin aletaan rakentaa varsinaista sovellusta, on tärkeää suunnitella tietokantoja käyttävä sovellus tarkasti. Tietokannat on helppompaa rakentaa, kun tiedetään millainen sovellus kehitetään. Sen jälkeen täytyy tehdä tietokantojen normalisointi sekä tarkastella alustavaa datamallia sovelluksen näkökulmasta. Kun tämä suunnitteluprosessi hoidetaan huolella ja kerralla kuntoon, säästetään paljon aikaa ja rahaa. (Meloni 2003, 36-38.)

Tässä projektissa MySQL valittiin tietokantojen rakentamiseen, koska sen käyttämisestä on paljon kokemusta jo muutaman vuoden ajalta. SQL-kyselykieltä on opiskeltu paljon koulutuksena, joten nämä olivat luonnollinen ratkaisu tietokantojen rakentamiseen sivustolle. PHP:n

ja MySQL:n yhteiskäytön suuri suosio websovelluksien rakentamisessa myös vaikutti ratkaisuun valita tämä kyseinen tietokantojen hallintajärjestelmä.

MySQL:ää oli erittäin vaivatonta käyttää. SQL-kyselykielen avulla oli helppoa ja nopeaa luoda tietokantaan tauluja. Myös erilaiset kyselyiden suorittamiset sujuivat vaivattomasti. Toisaalta tietokantojen taulut eivät olleet kovinkaan monimutkaisia, joten mikä tahansa muu tietokantojen hallintajärjestelmä olisi voitu valita käyttöön. Mutta valittu ratkaisu oli kuitenkin oikea.

Alla olevassa esimerkissä otetaan yhteys "Vuokra-asunnot"- tietokantaan käyttämällä PHP:tä. Kun hallintasivustolla päivitetään kyseisen tietokannan tietoja, tiedot näkyvät automaattisesti myös sivustolla. Tiedot siis luetaan suoraan tietokannasta. Näin ollen sivuston hallinta helpottuu huomattavasti. Näin ollen itse sivustolle ei tarvitse tehdä muutoksia, vaan riittää kun muokataan tietokantaa.

```
$sql="UPDATE $tbl_name SET osoite='$osoite', m2='$m2', kerros='$kerros', tyyppi='$tyyppi', vuokra='$vuokra', saatavilla='$saatavilla' WHERE osoite='$osoite'"; $result=mysql_query($sql);
```

7.9 Oracle

Oracle on relaatiotietokanta ja sitä valmistaa Larry Ellisonin johtama yritys. Oracle on julkaistu jo 1970-luvulla ja uusin saatavilla oleva versio siitä on 11g (grid) joka julkaistiin vuonna 2006. Oraclesta on yleisimmin käytössä versioita 7, 8, 8i ja 9i. Versiosta 8i Oracle koki ison muutoksen koskien lähinnä käyttöliittymää, sillä silloin siirryttiin sqlplus ohjelmassa java-pohjaiseen toteutukseen. Tietokantaan tuli myös joitain piirteitä helpottamaan www- ja java sovellusten tekoa. (Oracle a.)

Oraclessa tieto tallentuu loogisesti taulualueisiin (tablespace) ja fyysisesti tiedostoihin. Taulualueilla voi olla erilaisia lohkoja (segments), kuten esimerkiksi indeksilohkoja ja datalohkoja. Segmentit puolestaan koostuvat ilmenemistä (extents). Ilmenemät ryhmitellään yhtenevien datablokkien perusteella. Datablokki on tiedon talletuksen perusyksikkö. Fyysisellä tasolla tiedostot muodostuvat yhdestä tai useammasta datablokista, joissa blokkikoko voi vaihdella. (Oracle b.)

Oracle- tietokantaan voi tallentaa myös funktioita, ja ne voidaan suorittaa tietokantaohjelmiston sisällä. Funktiot kirjoitetaan SQL-, PL/SQL- tai Java-ohjelmointikielellä. Oracle hallinnoi kaikkea talletettua tietoa systeemitaualueella olevan tiedon avulla. (Oracle b.)

Tietokannan käsittelyä varten perustyökaluna Oraclessa on sqlplus, jolla voidaan toteuttaa kaikki tietokannan käsittelyoperaatiot. Muita Oraclen ominaisuuksia on mm. forms, jolla dataa voidaan syöttää lomakepohjista, reports, jonka avulla saadaan tulostettua erilaisia raportteja ja graph, jolla voidaan tietokannasta tulostaa graafisia esityksiä (Oracle Developer). Tietokannan ylläpitoa varten on Oraclella graafinen työkalu Enterprise manager. (Oracle a.)

Perinteisesti Oracle-tietokannan esimerkkikäyttäjä on scott ja salasana tiger. Bruce Scott oli yksi Ellisonin perustajakumppaneista ja Tiger oli hänen kissansa.

7.10 SQLite

SQLite on erittäin tiivis ja monelle alustalle sopiva SQL-tietokantamoottori, joka on kirjoitettu C-kielellä. Se tarjoaa monia tietokantahallinnon ydinpiirteitä, jotka ovat käytettävissä kilpailevissa tuotteissa kuten MySQL:ssä ja Oraclessa. Se tarjoaa silti huomattavia säästöjä mitä tulee kustannuksiin, oppimiskynnykseen ja hallintaan sijoitettuun aikaan. Sen ominaisuuksiin kuuluvat mm. seuraavat asiat:

- Koko tietokannan tallennus tapahtuu yhteen tiedostoon sallien helpon varmuuskopioinnin ja siirron.
- SQLite:n koko tietokantaturvallisuusstrategia perustuu kokonaan suoritettavan käyttäjän tiedosto-oikeuksiin. Joten esimerkiksi käyttäjä "web" voisi omistaa Web-palvelimen palvelinprosessin ja tuolla palvelimella suoritettua skriptin kautta tehdään yritys avata ja kirjoittaa SQLite-tietokantaan nimellä pmnb.db. Pystyykö käyttäjä tekemään näin, riippuu täysin pmnb.db-tiedostoon asetetuista oikeuksista.
- SQLite on ilmainen ja se on saatavilla yleisen toimialuelisenssin alaisuudessa Windows- ja Unix-alustoille. (Gilmore 2005, 493-494.)

SQLite:n komennot

SQLiten komentorivirajapinta on samantyyppinen kuin MySQL:n rajapinta siten, että se tarjoaa yksinkertaiset keinot vuorovaikutukseen SQLite-tietokantapalvelimen kanssa. Tällä työkalulla voidaan luoda ja ylläpitää tietokantaa, suorittaa hallinnollisia toimintoja kuten varmuuskopiointi ja skriptit sekä säätää asiakasohjelman toimintaa. SQLiten komentoriviohjelmassa taulu ja testidata luodaan seuraavasti:

Luodaan taulu työntekijä:

```
sqlite>create table työntekijä
...>empid integer primary key,
...>name varchar(25),
...>title varchar(25));
```

Lisätään muutama datarivi:

```
sqlite> insert into työntekijä values(NULL,"Aku Ankka","Toimitusjohtaja");
sqlite> insert into työntekijä values(NULL,"Lines Ankka","Sihteeri");
```

Suoritetaan kysely:

```
sqlite>select * from työntekijä;
Poistutaan rajapinnasta:
sqlite>.quit
```

sqlite_open()

```
resource sqlite_open(string filename [, int mode [, string & error_message]])
```

sqlite_open()-funktion avaa SQLite-tietokannan luoden ensin tietokannan, jos se ei jo ole olemassa, *filename*-parametri määrittää tietokannan nimen, valinnainen *mode*-parametri määrittää pääsyn oikeustason, minkä alaisuudessa tietokanta tullaan avaamaan ja *error_message*-parametrille annetaan automaattisesti arvo, joka määrittää virheen, jos tietokantaa ei voida avata. (Gilmore 2005, 496.)

```
sql_close()
void sqlite_close(resource dbh)
```

sqlite_close()-funktio sulkee yhteyden tietokantaresurssiin, jonka on määrittänyt *dbh*. Sitä kutsutaan sen jälkeen, kun kaikki tarvittavat tietokantaan liittyvät tehtävät on suoritettu.

```
<?php
    $sqldb = sqlite_open("pmnb.db");
    // suorittaa tarvittavat tehtävät
    sqlite_close($sqldb);
?>
```

```
sql_query()
resource sqlite_query(resource dbh, string query)
```

sqlite_query()-funktio suorittaa SQL-kyselyn tietokantaan, jonka on määrittänyt *dbh*-parametri. Jos kyselyn on tarkoitus palauttaa tulosjoukko, FALSE palautetaan, jos kysely epäonnistuu. Kaikki muut kyselyt palauttavat TRUE, jos kysely oli onnistunut. (Gilmore 2005, 497.)

Seuraavassa tulostuu työntekijän nimi ja ID eli tunnus:

```
<?php
    $sqldb = sqlite_open("pmnb.db");
    $results = sqlite_query($sqldb, "select * from työntekijä");
    while (list ($empid, $name) = sqlite_fetch_array($results)); {
        echo "Name: $name (työntekijä ID: $empid) <br />";
    }
    sqlite_close($sqldb); (Gilmore 2005, 497.)
?>
```

8 KÄYTETTÄVYYS

Verkkopalvelun toteuttamisen onnistuminen on mitattavissa siitä, kuinka käytettäväksi ihmiset kokevat palvelun. Mikä sitten tekee tuotteesta tai www-sivustosta käyttökelpoisen? Käytettävyys on sitä, että käyttäjä kokee palvelun käytön helpoksi, miellyttäväksi ja saa tehtyä palvelussa halutut toimet. Joskus kuvitellaan, että sivuston visuaalinen ilme pitäisi olla näytävä. Ulkoasun pitää kuitenkin olla tarkoituksenmukainen. Visuaalisen ulkoasun tulisi tukea viestiä, mutta myös tehdä palvelun käytöstä miellyttävää ja mieleenpainuvaa. Käyttöliittymässä on tärkeää tuoda esiin tarjottava sisältö tai toiminnallisuus mahdollisimman opittavasti, ymmärrettävästi ja tehokkaasti.

WWW-sivuston käytettävyyttä suunniteltaessa on tärkeintä miettiä, ketkä sivustoa tulevat käyttämään. Miksi he tulevat tälle sivustolle, miksi he tulevat juuri tälle yksittäiselle sivulle ja mitä he tältä sivulta saavat? Sivusto on tarpeeksi helppokäyttöinen, jos kaikki käyttäjät saavat sillä helposti tavoitteeseensa ikään ja toimintakykyyn katsomatta. Miellyttävyys rakentuu monesta eri osatekijästä. Ulkoasu, navigointi ja värit auttavat käyttäjää hahmottamaan sivuston toiminnot. Vanhat ja uudet mielikuvat puolestaan vaikuttavat sivuston miellyttävyys-teen jo ennen käytön aloittamista. (Nielsen 2000, 236-237.)

Hyvä verkkopalvelu on myös esteetön ottaen huomioon erityiskäyttäjryhmien kuten esimerkiksi näkövammaisten ja vanhusten tarpeet. Esteettömyys tarkoittaa tiettyjen käyttäjätyyppien huomioon ottamista esimerkiksi niin, että sivustot pyritään toteuttamaan ilman tiedossa olevia esteitä. Esteettömyyttä on usein tarkasteltu erillään käytettävyydestä. Joskus myös ajatellaan, että esteettömyys on jotain ylimääräistä tai että esteettömyys voidaan lisätä pintasilaukseksi käyttöliittymän päälle. Käytettävyydestäkin ajatellaan, että se on jotain irrallista, joka puolestaan voidaan lisätä valmiin tuotteen päälle. (Nielsen 2000, 239-240.)

Seniorikäytettävyys

Verkkosivujen tekijät ovat useimmiten alle viisikymppisiä ja tottuneita verkkokäyttäjiä. Heidän voi olla vaikea asettua iäkkäämmän käyttäjän asemaan. Usein kyse on vain tiedon puutteesta. Jo lapsina tai nuorina hiirenkäytön oppineiden voi olla vaikea tajuta, miten joku ei kykene valitsemaan useampia kohtia alavetovalikosta tai ymmärrä, mitä on salattu kirjautuminen. (Rossi, T.)

Senioreiden verkko-osaaminen vaihtelee aloittelevasta käyttäjästä rutinoituneeseen konkariin. Huomattavaa on, että suurin osa varttuneista käyttäjistä navigoi sivustolla hitaammin kuin nuoret ihmiset. Seniorit lukevat sivustolla olevat selitykset huolellisemmin ja välttävät turhia klikkailua päästääkseen päämääräänsä. Varttuneemmat käyttäjät myös viihtyvät sivuilla pidempään tutkaillen rauhassa sen sisältöä. Tämä vaikuttaa myös suuresti siihen, että palaavtko käyttäjät enää sivustolle ollenkaan. Jos seniorikäyttäjä havaitsee sivuston käytön vaikeaksi, on erittäin epätodennäköistä, että hän enää ikinä palaa sen pariin. (Rossi, T.)

Sivuston harkitseva selailu aiheuttaa sille myös omat haasteensa. Etusivun täytyy tarjota mahdollisimman riittävästi tietoa helposti omaksuttavassa muodossa. Varttunut käyttäjä poistuu lähes varmasti sivuilta, jos hänen etsimänsä tieto on monen hiiren napin painalluksen takana. Pelkkä houkutteleva sisältökään ei riitä, vaan muutamilla käytettävyysvirheillä voidaan aiheuttaa kokonaisen ikäluokan turhautumisen sivustoon. Kukaan ei jaksaa viihtyä kauaa sivustolla, jos sitä on vaikea käyttää tai tekstiä on vaikeaa lukea. (Rossi, T.)

Sivuston selkeys, loogisuus ja helppo navigaatio palvelevat kaikkia ikäryhmään katsomatta. Niin nuoret kuin vanhatkin voivat iloita toimintojen kunnollisesta palautteesta ja hyvistä hakutoiminnoista. Usein on kuitenkin tarpeellista tehdä peruskäyttäjiä iäkkäämpien ihmisten tarpeisiin sovitettu käytettävyysanalyysi. Siinä pyritään löytämään pelkästään tämän kohderyhmän olennaisimmat asiat. (Rossi, T.)

Vakiintuneet ratkaisut verkossa on kaikista tehokkain tapa houkutellessa senioreita käyttämään www-sivuja. Uudet ja standardeja rikkovat navigointi-ideat haittaavat kohderyhmää, jolla riittää tarpeeksi haasteita jo pelkässä verkon käytössä. Yleinen ongelma on sivujen erilaiset alisivut. Nämä alisivut noudattavat usein omaa logiikkaansa navigoinnin osalta ja ne myös yleensä ovat ulkoasultaan erilaiset kuin palvelun muut osaset. Varsinkin monimutkaisilla, pal-

jon linkkejä sisältävillä sivuilla tämä on ongelmana. Yhden käyttötavan omaksuttuaan seniorikäyttäjän on toisinaan vaikeaa mukautua uuteen tapaan edetä. (Rossi, T.)

Tulostusmahdollisuus täytyy myös ottaa huomioon, kun toteutetaan seniorikäyttäjille web-sivuja. Pisimmät artikkelit halutaan lukea rauhassa paperilta tottumuksen tai huonon näön vuoksi. Mitä varttuneempia käyttäjät ovat, sitä tärkeämmäksi tulostusvalinnat tulevat. Tämän vuoksi kehyksien käyttöä sivustolla olisi syytä välttää, koska se häiritsee tulostamisen laatua oleellisesti. Tulostusta tärkeämpää on käyttäjän mahdollisuus halutessaan vaihtaa sivustolla olevaa tekstin kokoa ja fonttia. Usein selaimen asetukset kuitenkin muuttavat vain perustekstin, mutta valinnan pitäisi myös vaikuttaa navigointiin sekä linkkeihin. Tämä ominaisuus on kuitenkin valitettavan usein jäänyt huomioimatta web-sivuston kehittäjiltä. (Nielsen 2000, 252-253.)

Läkkäämpien ihmisten hiirenkäsitteilytaito ei useinkaan ole nuorten tasolla, ja virhepainallusten riski on suurempi. Tämän vuoksi linkkien täytyisi olla normaalia suurempia fonttiltaan ja niiden ympärille pitäisi jättää riittävästi marginaalia. Linkkien täytyy myös erottua muusta sivustosta selvästi ja niiden täytyy reagoida käyttäjän toimiin näkyvästi. Sivujen rakenteen monimutkaistuessa navigointiin liittyy myös takaisin-nappula, joka on hyvä tarjota käyttäjille verkkopalveluissa yhtenä navigoinnin mahdollisuutena. Käytettävyyden kannalta on olennaista että takaisin-komento toimii loogisesti. Käyttäjää ei myöskään saa missään tapauksessa jättää odottamaan tapahtuuko tietyllä napsautuksella tai painalluksella jotakin. (Nielsen 2000, 256-257.)

Seniorikäytettävyyttä saadaan parannettua kiinnittämällä huomiota pieniin, mutta merkityksellisiin asioihin. Ongelmana on usein, että sivuston kehittäjä keskittyy liikaa sivuston kokonaisuuteen seniorikäytettävyyden kustannuksella. Käytettävyyteen liittyviä ongelmia on myös vaikea löytää, koska seniorikäyttäjien tyytymättömyys voi tulla täytenä yllätyksenä verkkopalvelun tekijöille. Kohdennettu käytettävyyksianalyysi on kuitenkin nopea ja vaivaton tapa korjata käytettävyyteen liittyvät ongelmat ja näin ollen pienelle vaivalle saadaan suuri vastine. (Rossi, T.)

Vanhain Ystävät Ry:n sivustosta pyrittiin rakentamaan mahdollisimman helppokäyttöinen. Etusivu on tehty mahdollisimman yksinkertaiseksi, jotta käyttäjät eivät turhaudu heti ensimmäisellä käyntikerralla. Sivuston navigointi on toteutettu mahdollisimman yksinkertaisesti ja selkeästi. Lisäksi sivuilla olevat linkit eivät sisällä mitään monimutkaisia polkuja alasivuille.

Fontin koko tarpeeksi isoksi, jotta seniorikäyttäjät näkevät tekstin mahdollisimman hyvin. Myös linkkien fonttikoko on suuri, jotta virhepainalluksilta välttyttäisiin. Lisäksi linkkien väri on pyritty tekemään näkyväksi ja selkeäksi. Sisällön määrää on myös pyritty laittamaan sivustolle rajoitetusti. Tekstin ja kuvien määrä on hyvä pitää sopivana, jotta käyttäjä löytää haluamansa tiedot helposti ja nopeasti.

9 POHDINTA

Opinnäytetyön aihe löydettiin sattumalta opinnäytetyöpakista Kajaanin ammattikorkeakoulun Internetsivuilta. Mahdollisia tutkielma-aiheita etsimme useista eri paikoista, niitä kuitenkin löytämättä. Toteutettavan sivuston piti olla dynaaminen, jotta se olisi tarpeeksi vaativa kahden opiskelijan työksi. Kotisivuston rakentamisessa päätettiin käyttää PHP:tä ja MySQL:ää. Vaikkakaan aikaisempaa kokemusta varsinkaan PHP:stä tekijöillä ei ollut. Kotisivujen tekemiseen olisi ollut useita eri vaihtoehtoja ja työkaluja, mutta PHP vaikutti helposti opittavalta ohjelmointikieleltä ja MySQL:stä oli jo ennestään hieman kokemusta. Mahdollisiin ongelmiin haluttiin löytää apua helposti, joten työn toteutukseen valittiin yleisimmät ohjelmat ja ohjelmointikielet.

9.1 Käytännön osio

Työ aloitettiin käytännössä keväällä 2008, jolloin aloimme suunnitella sivuston ulkoasua toimeksiantajan toiveiden mukaiseksi. Tuolloin aloimme myös opiskella sivustolla käytettävää PHP-ohjelmointikieltä. Kesäkuussa aloitimme sivuston toteuttamisen.

Html-editoriksi valitsimme Macromedia Dreamweaver 8, koska se on laajasti ammattilaiskäytössä oleva editori. PHP:tä tukevana Dreamweaver soveltui hyvin työhömmö. Dreamweaver ei kuitenkaan ollut kovinkaan hyvä, sillä esimerkiksi sivuston ulkoasun tekeminen oli hankalaa. Käytettävissä ei ollut kuitenkaan muita html-editoreita, joissa olisi ollut PHP-tuki eikä työn aikataulu sallinut muihin editoreihin tutustumista.

PHP-ohjelmointikieli sopi mielestämme hyvin dynaamisen sivuston toteuttamiseen. Koodin kanssa ei ollut suurempia ongelmia missään vaiheessa. Jotkut kirjoitusvirheet tosin aiheuttivat pohdittavaa koodissa, kun selain ilmoitti virheestä. Ajan kanssa kuitenkin saimme ongelmat selvitettyä ja erilaiset PHP-oppaat olivat hyvänä tukena toteutusvaiheessa. MySQL-tietokantojen rakentaminen oli suhteellisen helppoa, koska niiden toteuttamisesta oli jo kokemusta oppitunneilta. Lisäksi sivustolla olevat tietokannat eivät olleet kovinkaan vaativia, joten senkin vuoksi niitä oli helppo suunnitella ja rakentaa.

Suurin ongelma-alue käytännön osiossa oli kuvagallerian tekeminen. Ensimmäinen versio galleriasta sisälsi pikkukuvakkeita, joita klikkaamalla aukesi isompi kuva. Ongelmana oli, ettei kuvia voinut päivittää hallintaosion kautta. Kuville olisi pitänyt tehdä tietokanta, jonka kautta niitä olisi voinut lisätä ja poistaa. Se kuitenkin osoittautui liian vaativaksi tehtäväksi, varsinkin kun aika alkoi käydä vähiin. Galleria saatiin toteutettua tekemällä FCKeditor-ohjelmalla taulukko, johon tekstilinkkeinä kuvan nimi, jota klikkaamalla aukesi kyseinen kuva. Lopputulos ei aivan tyydyttänyt meitä, mutta toimeksiantaja kuitenkin hyväksyi ratkaisun.

Sivuston tekeminen opetti paljon, koska kummallakaan ei ollut minkäänlaista kokemusta dynaamisen sivuston tekemisestä. Kotisivut valmistuivat kuitenkin suhteellisen helposti ja onnistuimme pysymään myös omassa aikataulussamme. Toimeksiantaja oli myös varsin tyytyväinen tuotokseemme. Jälkikäteen ajateltuna muutama asia olisi voitu tehdä toisin, mutta aikamme oli kuitenkin rajallinen. Sivuston piti olla yksinkertaiset ottaen huomioon kohderyhmä ja mielestämme tässä onnistuimme hyvin. Mitään hienouksia sivusto ei sisällä, mutta niitä ei toimeksiantaja olekaan vaatinut missään vaiheessa projektia.

9.2 Kirjallinen osio

Kirjallisen osion työstäminen alkoi lähteiden kartoittamisella. Lähteitä löytyi runsaasti, niin kirjallisia kuin Internet-lähteitä. Englanninkielisiä lähteitä oli tarjolla todella paljon, mutta valitsimme niistä vain tarkoitukseemme sopivimmat. Aihepiiri oli kuitenkin sen verran yleinen, että myös suomenkielistä kirjallisuutta löytyi monipuolisesti. Kun aloitimme kirjoittamaan opinnäytetyön teoriaa, tarkoitus oli verrata skriptikieliä toisiinsa. Työn ohjaaja vaihtui kuitenkin puolessa välin projektia, joten suurin osa kirjallisesta osiosta täytyi muokata uusiksi. Myös sisällysluettelo täytyi suunnitella melkein kokonaan uudestaan.

Teoriassa oli tarkoitus kertoa yleisesti käyttämistämme työkaluista ja muista vastaavista tekniikoista. Kuitenkaan pelkkä yleinen teoria ei riittänyt, vaan ajatuksena oli kytkeä työssä käytetyt työkalut omaan työhön. Yhtenä näkökulmana oli miettiä, miten tehdään hyvät www-sivut ottaen huomioon työme kohderyhmä. Jouduimme myös tekemään Vanhain Ystävät ry:lle erilliset käyttöohjeet hallintasivun käytöstä. Tämä osoittautui yllättävänkin haasteelliseksi, koska niiden täytyi olla riittävän selkeät ja helposti ymmärrettävät.

Haasteellista oli saada omia ajatuksia kirjoitettua. Joskus ei vaan omaa tekstiä meinannut syntyä millään. Aika paljon teoriasta oli lainauspohjaista, mutta saimme kuitenkin aika paljon omia ajatuksia tuotua esille. Oli haastavaa saada tarpeeksi pituutta kirjalliseen osioon, mutta sen täytyi olla myös täyttä asiaa. Käytännön osio vaati niin paljon aikaa ja resursseja, ettei kirjalliseen osuuteen pystynyt paneutumaan kunnolla.

Työtä helpotti se, kun tekijöitä oli kaksi, joten pystyimme suurimman osan kirjallisesta osiosta jakamaan keskenään. Aiheesta oli kuitenkin jonkun verran aikaisempaa tietoa ja kokemusta, joten kirjoittamiseen oli suhteellisen helppo asennoitua. Parhaimmillaan tekstiä tuli useita sivuja päivässä. Kirjoittaminen oli ajoittain raskasta, mutta onnistuimme mielestämme tyydyttävästi.

9.3 Yhteenveto

Työ oli kokonaisuudessaan haasteellinen, vaikka se tehtiinkin parityönä. Dynaamisen sivuston toteuttamisesta ei ollut minkäänlaista kokemusta, joten asioihin paneutumiseen meni aikaa todella paljon. Toimeksiantajan vaatimukset eivät kuitenkaan olleet kovinkaan vaativia, joten pääsimme alkuun suhteellisen nopeasti ja helposti. Motivaatiota työn tekemiseen riitti, koska valmistuminen oli jo niin lähellä.

Ennen kuin aloittaa dynaamisen sivuston tekemisen, kannattaa suunnitteluun käyttää paljon aikaa. On hyvä hahmottaa vaikka paperille, mitä kaikkea sivut sisältävät ja mitä siellä pitäisi olla. Kun nämä asiat ovat selvillä, sivuston rakentamisesta tulee paljon helpompaa. Kun ryhtyy toteuttamaan sivustoa, kannattaa tarkkaan valita html-editori. Sopiva editori mahdollistaa helposti ulkoasun suunnittelun ja toteutuksen. Ohjelmointitaitoja on hyvä parantaa, jos ne eivät ole kovin hyvät. Koodin kirjoittamiselta ei voi välttyä, kun toteutetaan dynaamisia www-sivuja. Myös tietokantaosaamista on hyvä olla, sillä dynaamiset sivut sisältävät tietokantoja. Internetissä ja kirjastoissa löytyy runsaasti materiaalia, joista pystyy itseopiskelemaan.

Sivuston tekeminen opetti meille hyvin paljon. Projektin aikana ohjelmointitaidot kehittyivät varsinkin PHP:n osalta, joten olemme tyytyväisiä valitsemaamme aiheeseen. Lisäksi opimme, että toimeksiantajan toiveita ja tarpeita täytyy kuunnella tarkasti. Näin varmistetaan, että lopputuloksesta tulee kaikkia osapuolia tyydyttävä. Emme kuitenkaan ole ajatelleet ryhtyä tekemään kotisivuja ammatiksi, koska kummallakaan ei ole kovinkaan hyvät ohjelmointitaidot.

Kotisivun tekoprojekti on lisäksi pitkä ja raskas prosessi, joten kiinnostusta ei tähän sen enempää löydy.

Kaiken kaikkiaan opinnäytetyöprosessi kokonaisuudessaan opetti opiskelijoille projektiluonteisen työn tekemistä käytännössä. Aikataulu oli suhteellisen tarkka, mutta jouduimme hie-
man joustamaan sen suhteen. Koulussa sekä itseopittuja ohjelmointitaitoja käytettiin hyväksi ja kokoajan opiskeltiin lisää. Lopputuloksena syntyi kaikkia osapuolia tyydyttävä työ.

LÄHTEET

Ek, Jesper, Arvidsson, Stefan, Eriksson, Ulrika 2000. Tietokantojen käyttö Internetissä. Espoo : Pagina

Gilmore, W. Jason 2005. PHP & MySQL – tehokas hallinta. Helsinki : Readme.fi

Heinisuo, Rami 2007. PHP ja MySQL - tietokantaiset verkkopalvelut. Helsinki: Talentum

Korpela, Jukka K., Linjama, Tero 2003. Web-suunnittelu. Jyväskylä : Docendo

Krug, Steve 2006. Älä pakota minua ajattelemaan. Helsinki : Readme.fi

Meloni, Julie C. 2003. MySQL: Trainer kit. Helsinki: IT-press

Nielsen, Jacob 2000. WWW-suunnittelu. Jyväskylä : Gummerus

Patton, Ron 2006. Software Testing. Indianapolis : SAMS

Peltomäki, Juha 2000. CGI- ja ASP-ohjelmointi. Jyväskylä: Teknolit

Peltomäki, Juha 2001. Perl. Jyväskylä: Teknolit

Pohjonen, Risto 2002. Tietojärjestelmien kehittäminen. Jyväskylä : Docendo

Rantala, Ari 2005. Web-ohjelmointi. Jyväskylä: Docendo

Schroderus, Anita 2005. "Tallella ikä eletty" Vanhain Ystävät ry:n 85-vuotisjuhlajulkaisu.

Vanhain Ystävät ry. 2007. Toimintakertomus 2007.

Vanhain Ystävät ry. Säännöt

2K mediat a. Yleisesittely.

<http://www.2kmediat.com/internetohjelmointi/coldfusion2.asp> (Luettu 24.6.2008)

2K mediat b. JSP.

<http://www.2kmediat.com/internetohjelmointi/jsp.asp> (Luettu 1.7.2008)

Jackson, Brian 2007. Dynamic Web Pages: ASP vs. PHP.
www.domainmonster.com/editorials/asp_vs_php (Luettu 15.7.2008).

Interakta. Käytettävyyden testaus
<http://www.interacta.fi/testaus.html>

Lomakkeiden käsittelytekniikat 2004.
<http://appro.mit.jyu.fi/2004/kevat/www/luennot/luento5>

Mischook, Stefan 2007. Database driven websites: What are they and how they are built?
<http://www.killerphp.com/articles/wp-content/uploads/2007/04/database-driven-websites.pdf> (Luettu 29.10.2008)

Mohr, James 2005. Editing web-pages with FCKeditor.
<http://www.linux-magazine.com/w3/issue/59/FCKEditor.pdf> (Luettu 18.10.2008)

Net Site Story. Dreamweaver 8. 2007
<http://www.netsitestory.com/dreamweaver/index.html> (Luettu 15.8.2008)

Oracle a. Saatavilla:
http://rinkka.helsinki.fi:7778/orasov/www_apl.ohjeet.tulosta?man=1&kie=fin&luk=2
(Luettu 17.8.2008)

Oracle b. Saatavilla:
<http://fi.wikipedia.org/wiki/Oracle> (Luettu 17.8.2008)

PHP ja MySQL.
http://sinuhe.jypoly.fi/~tuito/i_opinnot/main3.htm#231 (Luettu 22.8.2008)

Rossi, Tino. Käytettävyyden tutkiminen –kohderyhmänä seniorit.
<http://www.steerco.fi/Default.aspx/141/Seniorikäytettävyys> (Luettu 3.11.2008)

SoftwareProjects. What is ASP?
www.softwareprojects.org/asp-tutorial-01.htm (Luettu 15.7.2008).

TukTuk Media 2007. Käytettävyydestaus
<http://www.tuktukmedia.com/kaytettavyys-arviointi/> (Luettu 4.10.2008)

Wikipedia a. Ruby.
<http://fi.wikipedia.org/wiki/Ruby> (Luettu 18.6.2008).

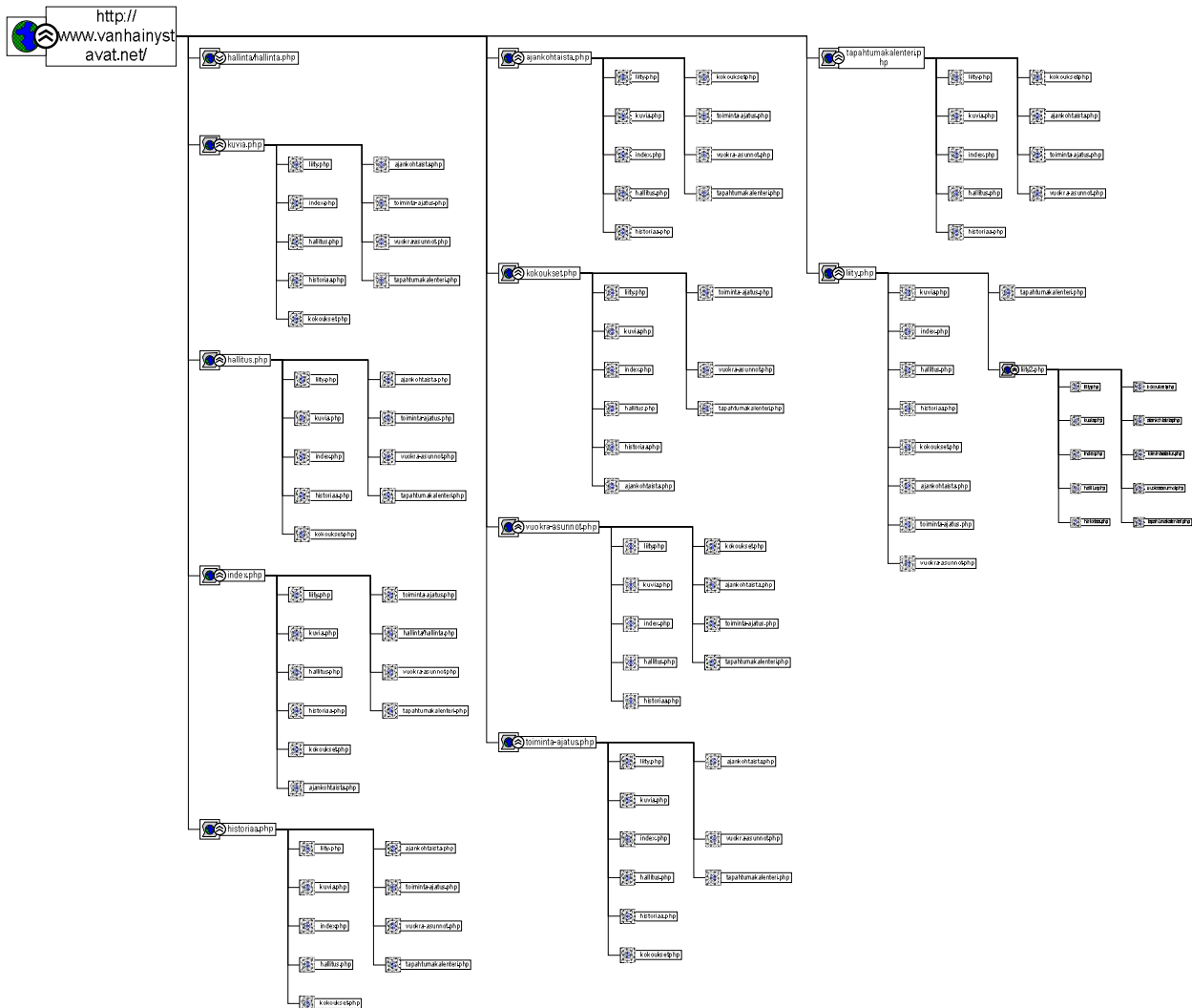
Wikipedia b. Ruby on Rails.
http://fi.wikipedia.org/wiki/Ruby_on_Rails (Luettu 16.6.2008).

LIITTEIDEN LUETTELO

Liite1: Vanhain ystävät ry:n sivukartta

Liite2: Hallintasivun käyttöohjeet

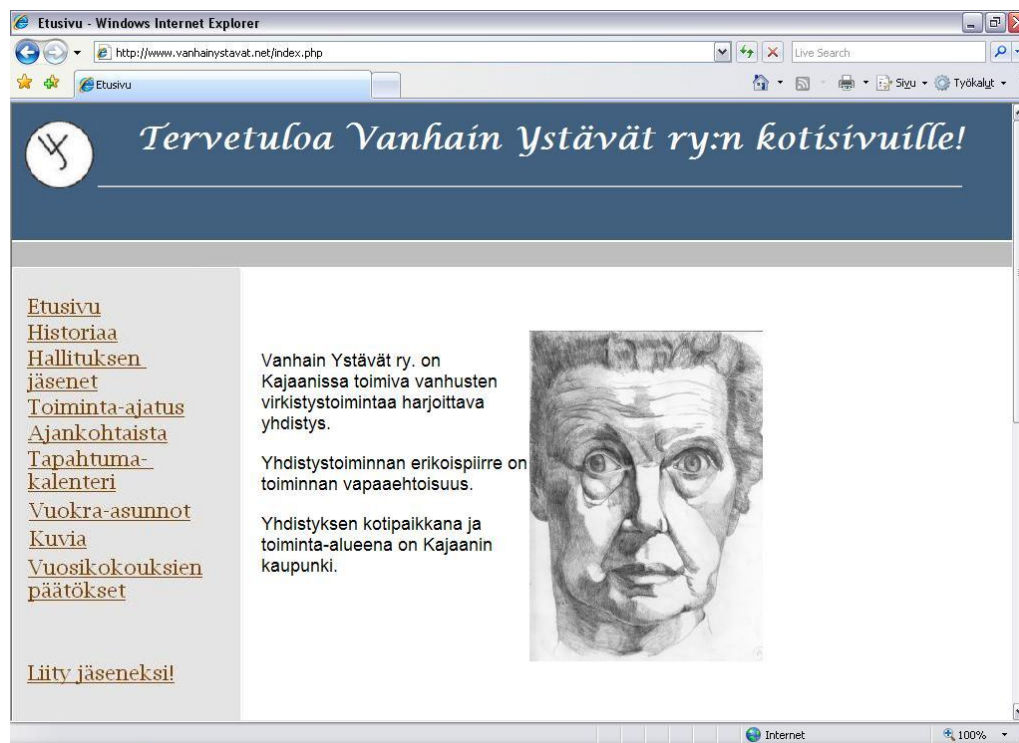
Vanhain ystävät ry:n sivukartta



Hallintasivun käyttöohjeet



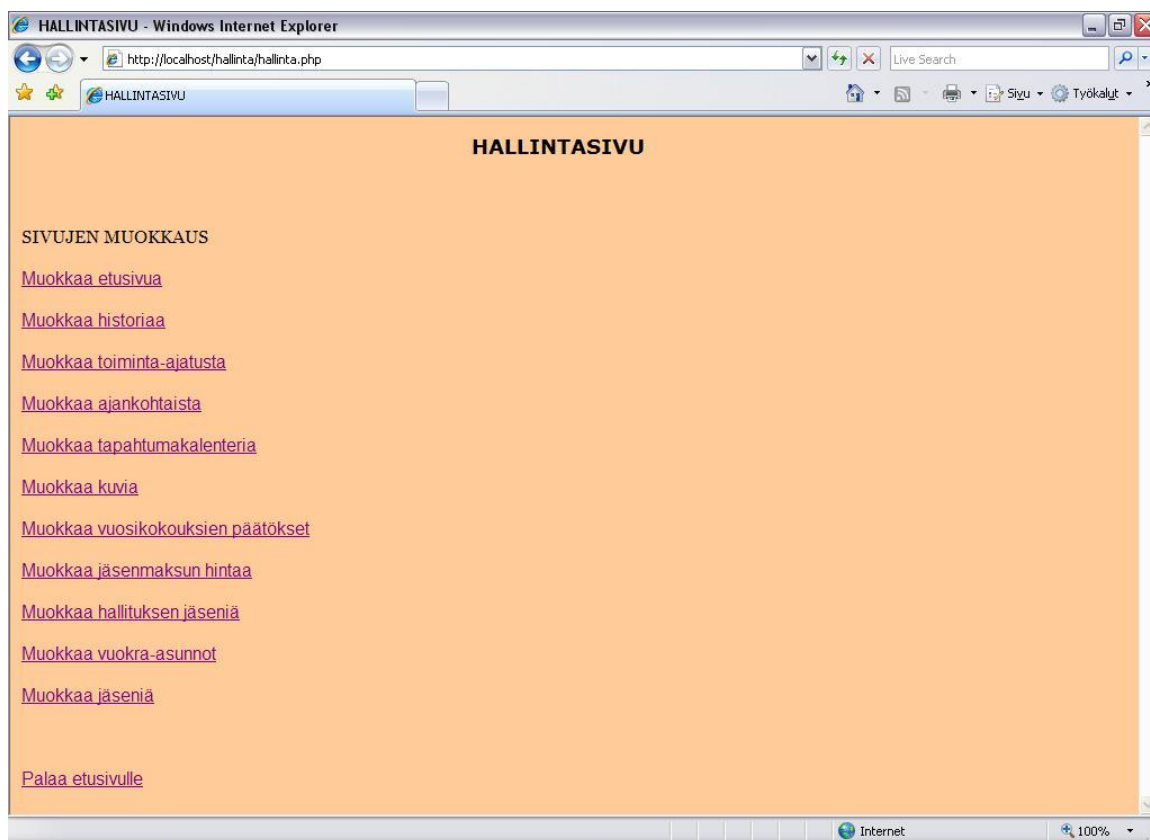
Klikkaa etusivun vasemmassa yläkulmassa olevaa logoa päästäksesi hallintasivun kirjautumissivulle.



Kirjoita oikea käyttäjänimi ja salasana ja klikkaa "OK"-painiketta.



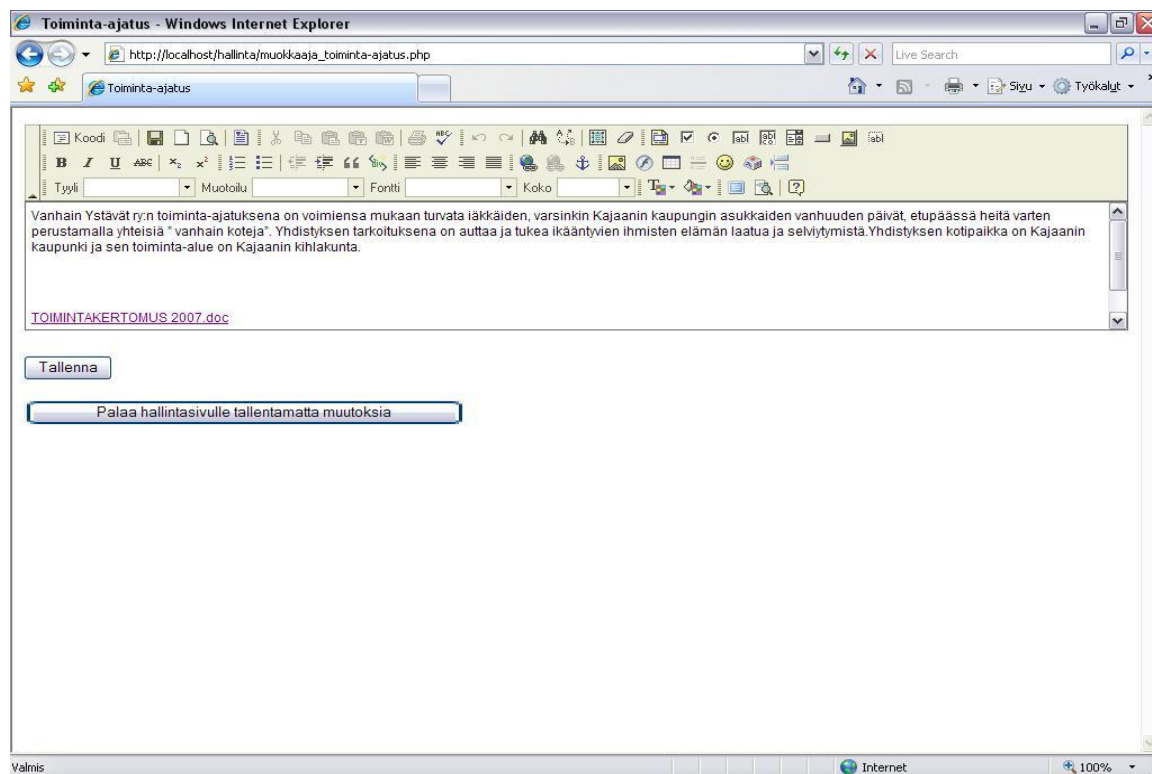
Jos kirjautuminen onnistui, niin pääset hallintasivulle.



Tekstin muokkaaminen


Pelkkää tekstiä on "Muokkaa toiminta-ajatusta", "Muokkaa ajankohtaista", "Muokkaa tapahtumakalenteria", "Muokkaa vuosikokouksien päätökset" ja "Muokkaa jäsenmaksun hintaa"- linkkien alla.

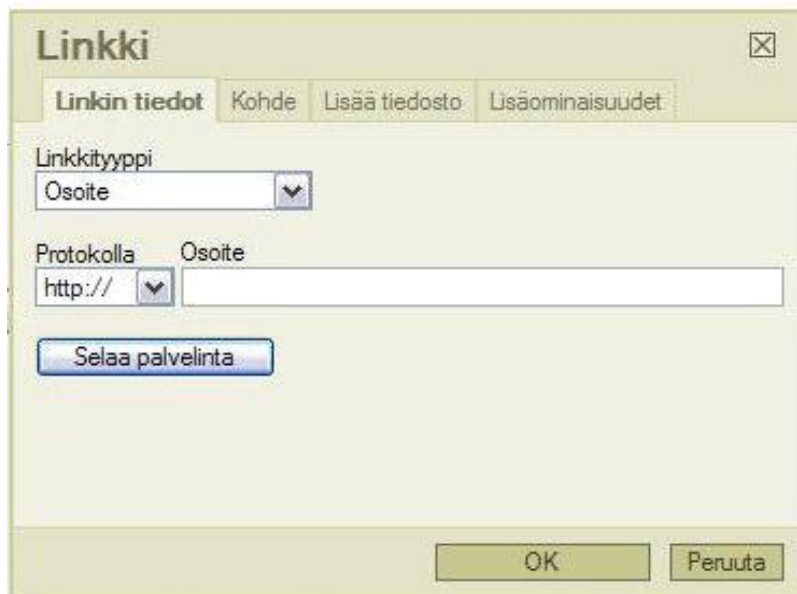
Alla oleva kuva on "Muokkaa toiminta-ajatusta"- linkistä, johon pääset klikkaamalla linkkiä. Kaikki muokkaukset tehdään tässä samassa editorissa.



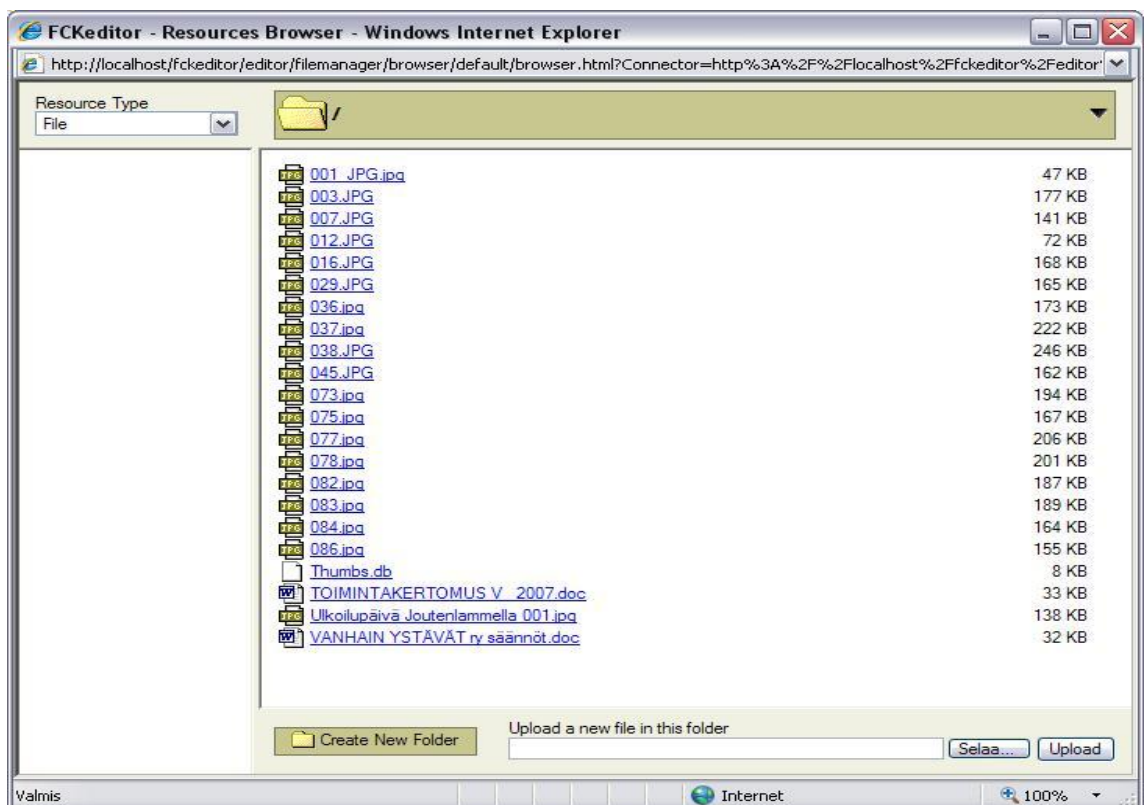
Tekstiä pystyy lisäämään yksinkertaisesti kirjoittamalla edellisen tekstin perään. Jos haluaa poistaa koko tekstin, nopeimmiten se käy "maalaamalla" koko tekstin ja painamalla näppäimistöllä "Delete"-näppäintä. Myös tekstin tyyliä, fonttia ja kokoa pystyy kätevästi muuttamaan. Ne kaikki löytyvät editorin työkaluista.

"Toiminta-ajatus"-sivulla on myös kaksi linkkiä: TOIMINTAKERTOMUS 2007.doc (näky kuvassa) ja VANHAIN YSTÄVÄT ry säännöt.doc. Nämä ovat kumpikin Word-dokumentteja.

Uusi linkki tehdään ensin klikkaamalla haluttuun kohtaan editorissa, jonka jälkeen klikkaamalla  painiketta editorin työkalurivillä. Tästä aukeaa seuraavanlainen ikkuna:



Tässä painamalla "Selaa palvelinta"- painiketta avautuu alla oleva ikkuna.



Hakeaksesi uuden tiedoston, kuten Word-dokumentin paina "Selaa.." - painiketta, jonka jälkeen haet haluamasi tiedoston tietokoneelta. Kun olet valinnut tiedoston paina "Upload"-painiketta, jolloin tiedosto ilmestyy listalle (kuten kuvassa näkyy). Sitten vain klikkaat tiedostoa, jolloin pääset takaisin edelliseen ikkunaan (Kuva5) ja painat "OK" - painiketta.

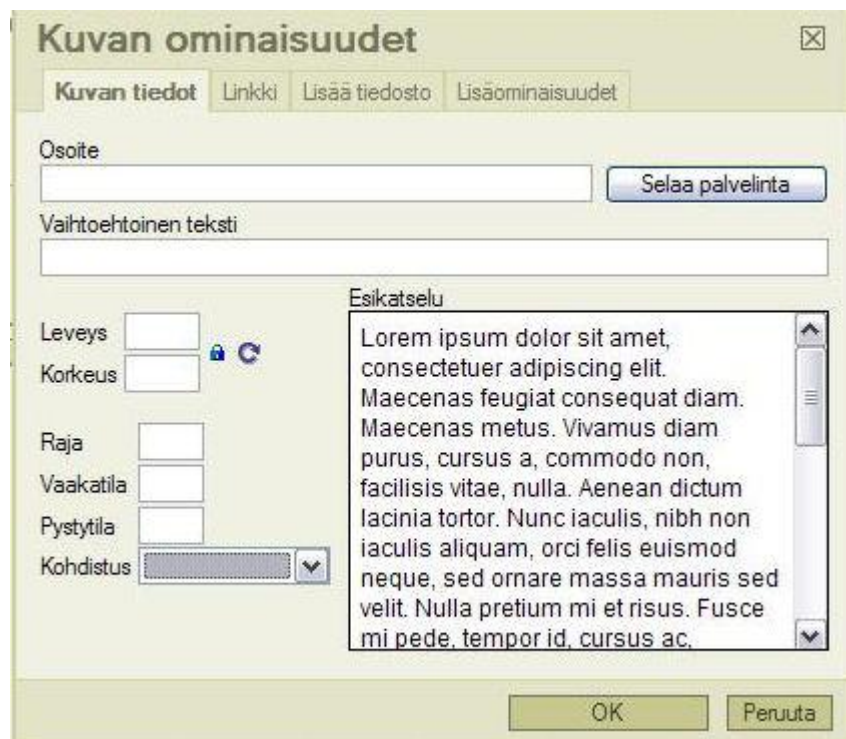
Nyt linkki on ilmestynyt editoriin, mutta tiedostonimi on jotain epämääräistä. Sen voi nimetä uudelleen, kunhan muistaa jättää .doc- päätteen loppuun. Täytyy ottaa myös huomioon, että editori ei tunnista kuin .doc- päätteiset tiedostot. Eli jos olet tehnyt tiedoston uudella Wordin versiolla, päätte on muotoa .docx. Tiedosto täytyy siis tallentaa vanhaan Word 97-2003-tiedostomuotoon.

Kun halutut muutokset/lisäykset on tehty, ne tallennetaan painalla "Tallenna"- painiketta (Kuva4). Tästä avautuu sivu, josta klikkaamalla "Avaa muokattu sivu"- painiketta päästään katsomaan, että muokkaus on onnistunut. Jos et halua tehdä muutoksia sivuun, pääset takaisin hallintasivulle klikkaamalla editorissa "Pala hallintasivulle tallentamatta muutoksia"- painiketta.

Kuvien lisääminen

Uusi kuva lisätään sivulle ensin klikkaamalla haluttuun kohtaan jonka jälkeen klikkaamalla

editorissa  painiketta. Kuvapainikkeesta avautuu "Kuvan ominaisuudet"- ikkuna:



Kuvan hakeminen toimii samantapaisesti kuin linkin teko. Ensimmäinen painetaan "Selaa palvelinta"-painiketta, jonka jälkeen avautuu sama ikkuna kuin edellä (Kuva6). Kun haluttu kuva on ha-

ettu se ilmestyy äskeisen kuvan esikatselu ikkunaan pikku kuvana. "Kuvan ominaisuudet"-ikkunassa on kohta "Kohdistus", josta pystyy valitsemaan miten kuva asettuu tekstiin, jos se halutaan tekstin sekaan. Kun halutut muutokset on tehty painetaan "OK" ja sitten editorissa "Tallenna"-painiketta.

Hallintasivun "Muokkaa kuvia"-linkin alla olevat kuvat ovat tehty pikkukuvina taulukkoon. Pikkukuvat ovat nimetty kuva1, kuva2, kuva3... jne. Kuvia on tällä hetkellä 31 eli viimeinen pikkukuva on nimellä kuva31. Ne kaikki löytyvät edellisen kuvan listalta (Kuva6.) Lisätäkseen uusia kuvia, klikkaa taulukkoon, hae haluamasi pikkukuva ja tee siitä linkki sitä vastaavalle isolle kuvalle. Tällä hetkellä taulukkoon pystyy lisäämään 12 uutta kuvaa. Jos Internetistä otatte kuvia, niin jotkut niistä ovat tekijänoikeuksien alaisia eli niitä ei periaatteessa saisi ottaa sieltä.

Henkilö- ja asuntotietojen muokkaaminen

Hallintasivulla on linkit "Muokkaa hallituksen jäseniä", "Muokkaa vuokra-asunnot" ja "Muokkaa jäseniä". Näitä tietoja muokataan hieman eri tavalla kuin muita. Alla oleva kuva on "Muokkaa hallituksen jäseniä"-linkin alta.

Nimi	Muokkaa/poista tietoja			Poista	Muokkaa
	Nimike	Puhelinnumero	Sposti		
Marja Leskinen	puheenjohtaja	0400-271 006	marja.leskinen@suomi24.fi	Poista	Muokkaa
Raija Honkanen	varapuheenjohtaja	040-7196 467		Poista	Muokkaa
Arja Ohtonen	sihteeri/rahastonhoitaja	0400-410 571		Poista	Muokkaa
Elsa Rautiainen	jäsen	050-4141 902		Poista	Muokkaa
Esko Jaakola	jäsen	044-3361 319		Poista	Muokkaa
Ritva Klemetti	varajäsen	044-5161 549	ritva.klemetti@hotmail.com	Poista	Muokkaa
Pirkko Teno	varajäsen	040-5739 686		Poista	Muokkaa

Lisää jäsen:

Nimi:
 Nimike:
 Puhelinnumero:
 Sposti:

[Palaa hallintasivulle](#)

Tässä tiedot ovat taulukossa kuten itse sivullakin. Taulukon oikeassa laidassa on linkit "Poista" ja "Muokkaa". Kun klikataan "Poista" halutun jäsenen kohdalla, nimi ja kaikki muutkin kyseisen jäsenen tiedot häviää taulukosta. Jos halutaan muokata jonkin jäsenen tietoja, se onnistuu klikkaamalla "Muokkaa"-linkkiä. Aukeaa seuraavanlainen sivu:

Muokkaa tietoja

Nimi	Nimike	Puhelinnumero	Sposti
Marja Leskinen	puheenjohtaja	0400-271 006	marja.leskinen@st

Tässä voidaan muuttaa jäsenen tietoja, kuten vaihtaa puhelinnumero tai sähköpostiosoite. Kun muutokset on tehty, painamalla "Lähetä"- painiketta ne tallentuvat taulukkoon ja itse sivustolle.

Jos halutaan lisätä kokonaan uusi hallituksen jäsen, se voidaan myös tehdä tässä (Kuva8). Lisäämällä tiedot "Nimi", "Nimike", "Puhelinnumero" ja "Sposti" lomakkeeseen ja painamalla "Lisää"- painiketta, uusi jäsen ilmestyy taulukkoon ja samalla se tallentuu myös itse sivustolle.

Hallintasivulla olevat "Muokkaa vuokra-asunnot" ja "Muokkaa jäseniä"- linkkien alla olevat tiedot muokataan täysin samalla tavalla kuin hallituksen jäsenten tiedot.