

SATAKUNNAN AMMATTIKORKEAKOULU



Ville Nyroos

2007

YRITYKSEN INTERNETSIVUSTON VALMISTUS

Tekniikka Rauma

Tietotekniikan koulutusohjelma

YRITYKSEN INTERNETSIVUSTON VALMISTUS

Nyroos, Ville

Satakunnan ammattikorkeakoulu

Tekniikka Rauma

Tietotekniikan koulutusohjelma

Asiakas: Kaihdinliike Nyroos Ky

Syyskuu 2007

Ohjaaja: Mikko Javanainen

UDK: 004.43

Asiasanat: WWW-sivut, HTML, PHP, JavaScript, CSS

Projektin lähtökohtana oli saada toimiva, selkeä ja edustava internetsivusto yritykselle. Sivuston valmistukseen harkittiin useita eri työkaluvaihtoehtoja. Lopuksi päädyttiin siihen tulokseen, että ohjelmakoodin kirjoittaminen tekstieditorilla on tekijän kannalta paras vaihtoehto.

Opinnäytetyön aiheena oli valmistaa toimiva ja edustava internetsivusto Kaihdinliike K. Nyroos Ky:lle. Yrityksellä ei ollut lainkaan internetsivuja ennen projektia. Sivuston piti sopia yrityksen uuteen ulkoasuun sekä mainostaa asiakkaille yrityksen palveluja.

Sivusto on ohjelmoitu pääosin HTML-kielellä, mutta se sisältää myös PHP:tä sekä JavaScriptiä. Sivuille suunniteltiin lisäksi tietokanta- tai hallinto-ominaisuuksia, mutta niistä luovuttiin, koska niistä olisi ollut enemmän haittaa kuin hyötyä.

DESIGNING AND PRODUCING A COMPANY'S WWW PAGES

Nyroos, Ville

Satakunta University of Applied Sciences

School of Technology Rauma

Information Technology

Commissioned by Kaihdinliike Nyroos KY

September 2007

Tutor: Mikko Javananainen

UDC: 004.43

Keywords: WWW pages, HTML, PHP, JavaScript, CSS

The aim of the project was to get working and simple Internet pages which give a good impression of Kaihdinliike Nyroos. Several different tool choices were taken into consideration for the project, but in the end the conclusion was that writing the program code with a text editor would be the best choice for the author. This method of programming was a little more time-consuming, but much more precise.

The pages are written mostly in HTML, but they also contain JavaScript and PHP. The graphics used on the pages are either created by drawing with the computer, or in the case of other companies' logos, copied from their respective Internet pages.

The pages have a simple layout: navigation links on the left-hand side and contents on the right-hand side. The layout is easy to use and to understand. The pages contain information about the company, advertisements and links to the manufacturers' Internet pages.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

SISÄLLYS

LYHENTEET JA TERMIT

1 JOHDANTO	6
1.1 Yritys	6
1.2 Työn lähtökohdat	6
2 OHJELMOINTIKIELET	7
2.1 HTML	7
2.2 CSS	8
2.3 JavaScript	9
2.4 PHP	11
3 OHJELMISTON ELINKAARI	13
3.1 Vesiputousmalli	13
3.1.1 Esitutkimus	14
3.1.2 Määrittely	15
3.1.3 Suunnittelu	15
3.1.4 Toteutus	16
3.1.5 Testaus	16
3.1.6 Ylläpito	17
4 OHJELMISTON TOTEUTUS	17
4.1 HTML-toteutus	17
4.1.1 HTML validointi	19
4.1.2 HTML rakenne	20
4.2 JavaScript-toteutus	20
4.3 PHP-toteutus	22
4.4 Graafinen toteutus	23
4.5 Sivuston rakenne	24
5. YHTEENVETO	27
LÄHDELUETTELO	28

LYHENTEET JA TERMIT

CSS	Cascading Style Sheet. Kieli, jolla voidaan määrittellä tyyliominaisuuksia.
HTML	HyperText Markup Language, yleinen internetsivujen ohjelmointikieli.
PHP	PHP: Hypertext Preprocessor. Tehokas ja suhteellisen yksinkertainen ohjelmointikieli.
Spagettikoodi	Koodi, joka on sotkuista, monimutkaista ja rakenteetonta.
Unicode	Ohjelmistotalojen kehittämä laaja merkistöstandardi, joka kattaa suurimman osan maailman kirjoitettujen kielten käyttämistä merkeistä.
W3C	World Wide Web Consortium. Yhdistys, joka kehittää Internet teknologiaa ja standardeja.
WYSIWYG	What You See Is What You Get. Lyhennettä käytetään viittaamaan sellaisiin ohjelmistoihin, joissa sisältö näyttää muokattaessa hyvin samalta kuin lopputulos.
WWW	World Wide Web.

1 JOHDANTO

1.1 Yritys

Kaihdinliike K. Nyroos Ky on Raumalla vuonna 1982 perustettu perheyritys. Yritys on tarjonnut laadukasta ja asiantuntevaa apua kodin sisustuksessa ja kaihtimien asennuksessa. Toiminta alkoi omakotitalon kellarista, joka oli muunnettu myymäläksi. Vuonna 1983 myymälä siirrettiin Urheilukadulle, josta tuli ensimmäinen varsinainen myymälä sekä varasto.

Myöhemmin yritystä laajennettiin hankkimalla suurempi tila Metallitieltä, jossa yrityksellä oli myös mahdollisuus valmistaa tuotteita myyntiin. Urheilukadun myymälä myytiin samalla. Metallitien myymälässä syttyi tulipalo, joka tuhosi valmistus- ja myymälätilat. Tällä hetkellä yrityksen toimitilat ovat Monnankadulla, vanhan pankin toimitiloissa. Yritys ei enää valmista omia tuotteitaan, vaan toimii jälleenmyyjänä.

1.2 Työn lähtökohdat

Yrityksen uuden, nuorekkaan ulkoasun myötä päätettiin, että yritys tarvitsee myös toimivan ja käytännöllisen internetsivuston. Internetistä asiakkaat näkevät helposti yrityksen tarjoamia palveluja ja tuotteita sekä yhteystietoja. Internetsivut ovat myös edullinen ja helppo tapa mainostaa yritystä ja saada lisää asiakaskuntaa nuoremmista sukupolvista. Yritys toimii lähinnä Rauman ympäristössä, joten sivustot toteutettiin pelkästään suomenkielisinä.

2 OHJELMOINTIKIELET

2.1 HTML

HTML eli HyperText Markup Language on suunniteltu määrittelemään dokumentin looginen järjestys tärkeillä hypertekstijatkeilla. Sitä *ei* ole suunniteltu olemaan kieli WYSIWYG (What You See Is What You Get)-sanaprosessoreille, kuten Word tai WordPerfect. Tämä valinta tehtiin, koska samaa HTML-dokumenttia voidaan katsoa monella eri ”selaimella”, joilla on hyvin erilaiset ominaisuudet. Esimerkiksi HTML sallii merkitä tekstin osiot otsikoiksi tai kappaleiksi ja jättää näiden merkittyjen elementtien tulkitsemisen selaimelle. Esimerkiksi yksi selain saattaa tunnistaa kappaleen alun, kun toinen jättää vain tyhjän viivan.

HTML-ohjeet jakavat dokumentin tekstin lohkoihin, joita kutsutaan elementeiksi. Nämä voidaan jakaa kahteen laajaan kategoriaan – niihin, jotka määrittelevät, miten dokumentin BODY (runko) tulee näyttää selaimessa, ja niihin, jotka määrittelevät tiedon dokumentista, kuten otsikon tai suhteen toisiin dokumentteihin. (Graham 1998.)

HTML:n historia alkoi vuonna 1989, kun Tim Berners-Lee ja Robert Caillau hahmottelivat korviketta CERNin dokumenttien kirjavalle formaatille. Rinnalle suunniteltiin yksinkertainen verkkoprotokolla HTTP. HTML-dokumenttien verkkoa internetissä kutsutaan World Wide Webiksi eli lyhyesti webiksi. Joillekin se on yhtä kuin internet. CERN käynnisti oman WWW-palvelimensa vuonna 1991, ja monet muut ideasta kiinnostuneet liittyivät siihen nopeassa tahdissa. Nykyisin HTML-standardia ylläpitää W3C (World Wide Web Consortium).

HTML:n alkuperäinen tarkoitus oli pikemminkin kuvata WWW-sivun rakennetta kuin sen ulkoasua, mutta sivujen kirjoittajat halusivat ennen pitkää paremmat mahdollisuudet vaikuttaa myös dokumenttiansa ulkoasuun. Selainvalmistajat vastasivat tarpeeseen esittelemällä HTML-määrittelyyn kuulumattomia

elementtejä, joilla esitystapaa saattoi kuvailla. Monet näistä elementeistä jouduttiin myöhemmin ottamaan mukaan viralliseen HTML-määrittelykseen käytännön standardeina. Nykyisin monia elementtejä jälleen puuttuu itse HTML:sta, sillä ulkoasun kuvailussa on siirrytty erillisiin tyyliohjeisiin, jotka kuvataan CSS-kielellä. CSS:ään siirtyminen paitsi yksinkertaisti HTML:ää, myös helpotti sivujen luomista ja päivittämistä.

HTML-dokumentteja voi kirjoittaa millä tahansa tekstieditorilla, mutta tehtävään on tarjolla myös erityisiä HTML-editoreja. Periaatteessa HTML-koodin tulisi aina läpäistä validointi, mutta käytännössä monet selaimet hyväksyvät myös dokumentteja, joiden HTML-koodi ei ole standardin mukaista. Perinteisessä HTML:n validointi on ollut sikäli työlästä, että sen jäsentäminen ei ole aivan triviaalia. (HTML – Wikipedia: 2007.)

2.2 CSS

CSS (Cascading Style Sheets, kaskadityylimääritys) on tyylikieli, jolla voidaan liittää tyylimääritykset HTML-elementteihin. Kullekin elementtityypille voidaan määrittellä yksilöllinen tyyli, joka kuvaa esimerkiksi fontit, värit, marginaalit ja sijainnit. Esimerkki tyylimäärityksestä koodissa 1.

Kun World Wide Web kehiteltiin vuonna 1990, hanke oli puhtaasti tieteellinen. Verkon kautta levitettiin ainoastaan tieteellisiä raportteja, joilla sisältö oli paljon ulkoasua tärkeämpää. Räjähdyksmäinen WWW:n kasvu toi verkkoon myös muunlaisia dokumentteja. Alan nopea kehitys ja standardisoinnin hitaus loivat sekalaisen tilanteen. Suuret selainvalmistajat (Netscape ja Microsoft) loivat uusia tunnuksia, jotka toimivat ainoastaan kyseisen yrityksen selaimilla. Liike-elämän tajuttua WWW:n tärkeyden verkkomainonnasta on tullut yksi tärkeä osatekijä. Perinteisten paperilehtien verkottuminen tuo haasteita HTML:lle, joka ei pyrikään olemaan täydellinen ulkoasun muotoilukieli. Ilman tyylimääritysten käyttöä näyttävien sivujen tekeminen on hankalaa ja joissakin tilanteissa joudutaan tekemään useat, tietyille selaimille optimoidut sivut.

Tyylimäärittelyjen käyttäminen antaa tekijöille lisää mahdollisuuksia sivujen ulkoasun määrittelyille. Määrittelyt voidaan liittää HTML-dokumenttiin tai esittää omassa tiedostossaan, jolloin useampi dokumentti voi käyttää samoja määrittelyjä. Näin mahdollistetaan yhtenäinen ulkoasu ja helppo päivitettävyys. Kaskadi-menetelmällä tyylimäärittelyksiä voidaan tuoda erilaisista lähteistä. Kun tekijällä, käyttäjällä ja selaimella voi olla oma tyylimäärittelynsä, kaskadi-menetelmä ratkaisee ristiriidat eri määrittelysten välillä, koska se valitsee aina vain yhden mahdollisen säännön kunkin elementin esittämiseen. Samalle dokumentille voidaan luoda helposti erilaisia ulkoasumäärittelyksiä, jolloin siitä saadaan erilaisia dokumentteja eri käyttötarkoituksiin. (Pääkkönen 1998.)

```
#markiisit
{
position: absolute;
top: 170px;
left: 0px;
width: 175px;
border: 2px solid gray;
padding: 0px 20px 0px 20px;
background-color: #f0f0f0;
}
```

Koodi 1. Esimerkki CSS-määrittelystä.

2.3 JavaScript

JavaScript on Netscape Communications Corporationin kehittämä pääasiassa Web-ympäristössä käytettävä komentosarjakieli. JavaScriptin tärkein sovellus on mahdollisuus lisätä Web-sivuille dynaamista toiminnallisuutta.

JavaScriptin nykymuoto on oliopohjainen komentosarjakieli, jonka syntaksi perustuu löyhästi C-ohjelmointikielen.

JavaScriptin kehitti alun perin Netscapen Brendan Eich nimellä Mocha. Nimi muutettiin LiveScriptiksi ja viimein markkinointisyistä JavaScriptiksi. Java-ohjelmointikielen kanssa JavaScriptillä ei ole kuitenkaan juurikaan yhteistä.

Viimeisin kielen määrittely on JavaScript 1.5 vuodelta 1999, jota vastaa standardi ECMA-262 Edition 3. Standardoitua JavaScriptiä nimitetäänkin nimellä ECMAScript. Useat selaimet tukevat standardin lisäksi lisätoiminnallisuutta, kuten Mozilla-selainten E4X, joka on XML:n käsittelyyn erikoistunut kielen laajennus (standardi ECMA-357). (Javascript – Wikipedia: 2007.)

JavaScript (JS) on Netscapen kehittämä oliopohjainen ohjelmointikieli, jota modernit selaimet yleisesti tukevat. JS on nimensä mukaisesti ns. scriptikieli, jonka koodi tulkitaan, eli sitä ei erikseen käännetä omaksi ohjelmakseen ennen suorittamista kuten sovellusohjelmointikielissä. JS tarvitsee toimiakseen isäntäympäristön, kuten selaimen, jonka JS-tulkki suorittaa koodin rivi kerrallaan. JS:n vahvuus on sen käyttöjärjestelmäriippumattomuudessa, keveydessä, tehokkuudessa, sekä sen yksinkertaisessa ja vapaamuotoista syntaksissa.

JS-koodin tuottamiseen riittää mikä tahansa editori, sillä koodi koostuu pelkästä tekstistä. Tämä teksti taas muodostuu yleisistä ohjelmointikielien perusrakenteista.

Taulukossa 1 esitellään JavaScriptin perustietotyyppejä ja niiden ominaisuuksia. (Saarikumpu 2007.)

Kuvio 1. JavaScriptin perustietotyypit ja niiden ominaisuudet.

Numerot	<p>Numeroita (numbers) ovat positiiviset ja negatiiviset:</p> <ul style="list-style-type: none"> * kokonaisluvut - jotka on helpointa kirjoittaa tutussa kymmenjärjestelmämuodossa * liukuluvut - joissa desimaalierottimena on piste (esim. -7.0) <p>Numerot merkitään koodiin sellaisenaan.</p>
Merkkijonot	<p>Merkkijonot (strings) koostuvat merkeistä, yleensä tavallisesta tekstistä ja mahdollisesti erikoismerkeistä (escape characters), mutta kaikki Unicode-merkit ovat sallittuja. Merkkijonot kirjoitetaan, joko (kaksihipsuisten) lainausmerkkien, tai (yksihipsuisten) heittomerkkien sisälle.</p> <p>Merkkijonot voivat sisältää myös merkkausta, jolloin merkkauskielen sisältämät lainausmerkit voivat aiheuttaa harmeja. Yhdenmukaisuus- ja yhteensopivuussyistä kannattaa käyttää heittomerkkejä JS-merkkijonojen rajoina, koska useimmiten merkkauksieleessä käytetään lainausmerkkejä.</p>
Totuusarvot	<p>(Boolean values) Näitä loogisia arvoja on vain kaksi:</p> <ul style="list-style-type: none"> * true (tosi) * false (epätosi) <p>Totuusarvoja käytetään vertailuissa.</p>
Erikoisarvot	<p>Muuttujilla on kaksi erikoisarvotyyppiä:</p> <ul style="list-style-type: none"> * muuttuja on tyyppiä undefined (määrittelemätön) tilanteissa, joissa muuttuja on luotu, mutta sille ei ole määritetty arvoa * muuttuja on tyyppiä null (pätemätön) tilanteissa, joissa muuttujan arvo on epäkäypä

2.4 PHP

PHP (PHP: Hypertext Preprocessor) on laajalti käytetty avoimen lähdekoodin yleiskäyttöinen skriptauskieli, joka sopii erityisesti nettikehitykseen ja sitä voidaan sisällyttää HTML-kieleen.

Vuonna 1994 tanskalais-grönlantilainen Rasmus Lerdorf kirjoitti pienen kokoelman C-kielisiä CGI-skriptejä nimellä Personal Home Page Tools. Lerdorf julkisti työkalut GPL-lisenssillä 8. kesäkuuta 1995 nimellä PHP/FI (Personal Home Page / Forms Interpreter).

PHP/FI 2.0 julkaistiin marraskuussa 1997 ja sillä oli useita tuhansia käyttäjiä ympäri maailmaa. Ohjelma oli asennettu noin 50 000:een domainiin (1 % kaikista internetin domaineista) ja useat ihmiset osallistuivat sen kehittämiseen. Projekti oli yhä Rasmus Lerdorfin päävastuulla.

Andi Gutmans ja Zeev Suraski totesivat PHP/FI 2.0:n riittämättömäksi internet-kauppasovelluksen tarpeisiin. He kirjoittivat lähes koko lähdekoodin alusta asti uudelleen ja julkaisivat sen kesäkuussa 1998 nimellä PHP 3. Lyhenteen uudeksi merkitykseksi valittiin rekursiivinen PHP: Hypertext Preprocessor.

Talvella 1998 Gutmans ja Suraski aloittivat PHP:n ytimen uudelleen kirjoittamisen. Heidän tavoitteenaan oli ydin, joka tukisi kolmansien osapuolten ohjelmointirajapintoja. Uusi ydin julkaistiin vuonna 1999 nimellä Zend Engine (yhdistelmä nimistä Zeev ja Andi) ja maaliskuussa 2000 julkaistu PHP 4 käyttää sitä ytimenään.

PHP 5 julkaistiin heinäkuussa 2004. Sen ytimenä on Zend Engine II, joka muun muassa tukee olio-ohjelmointia ja sisältää sisäänrakennetun tietokantamoottorin (SQLite). Uusin versio (5.2.1) julkaistiin 8. helmikuuta 2007. PHP 4:ää kehitetään edelleen PHP 5:n rinnalla. (PHP – Wikipedia: 2007.)

Mikä erottaa PHP:n jostain kuten asiakaspuolen JavaScriptistä on se, että ohjelmakoodi suoritetaan palvelimella. Voit jopa konfiguroida nettiselaimesi prosessoimaan kaikki HTML tiedostosi PHP:llä, jolloin käyttäjillä ei ole minkäänlaista mahdollisuutta tietää mitä sinulla on hihassasi.

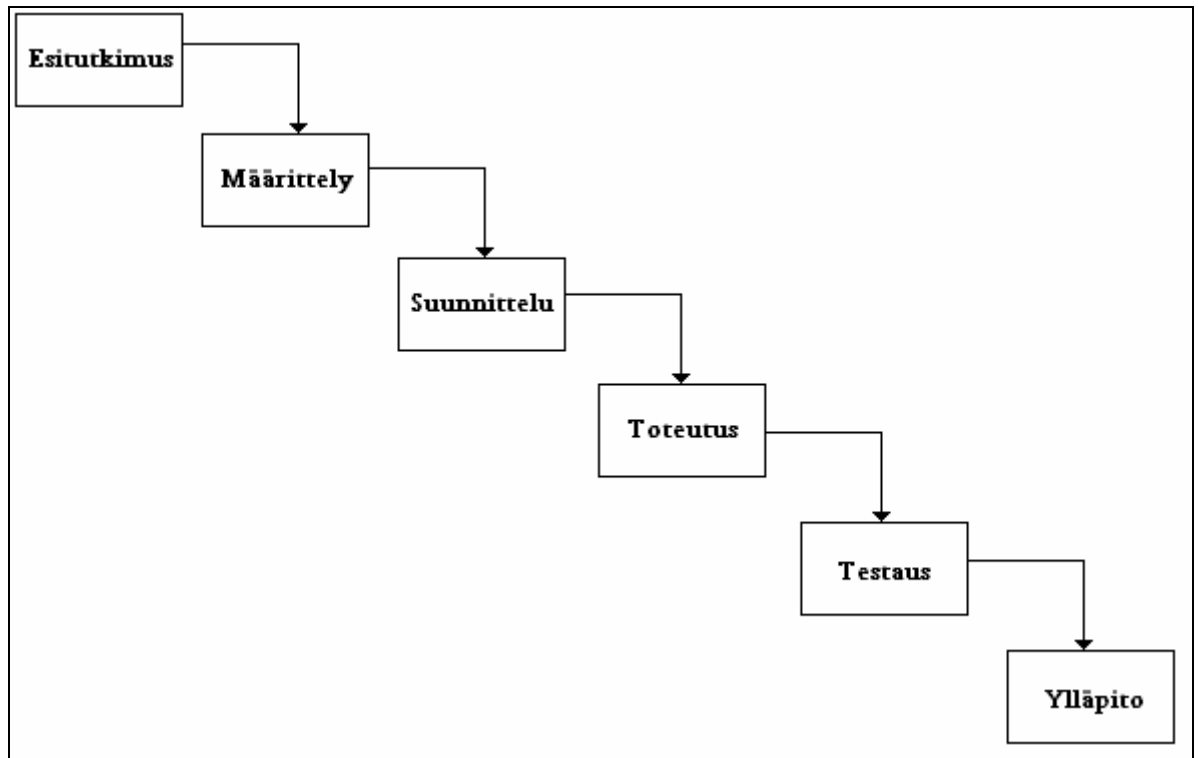
PHP on suosittu yleiskäyttöinen skriptikieli, joka sopii erityisesti Web-sovelluskehitykseen, koska se voidaan liittää suoraan HTML:ään. Lisäksi PHP:n lähdekoodi on avointa, eli sitä saa lisenssin sallimissa rajoissa tutkia, muokata ja levittää. Kieli on syntaksiltaan helppo, se pohjautuu suurimmilta osin C:hen, Javaan ja Perliin. Vaikka PHP onkin pääasiassa suunnattu dynaamisten web-sivustojen tekemiseen, sillä voi tehdä paljon muutakin. (PHP käsikirja: 2005.)

3 OHJELMISTON ELINKAARI

3.1 Vesiputousmalli

Ohjelmiston elinkaarella tarkoitetaan aikaa, joka kuluu ohjelmiston kehittämisen aloittamisesta sen poistamiseen käytöstä. Vaihejakomallilla tarkoitetaan tapaa, jolla ohjelmiston kehitystyö tai koko elinkaari jaetaan vaiheisiin. Tavallisin vaihejakomalli on ns. vesiputousmalli (Kuva 1). Mallista on olemassa useita eri muunnelmia, mutta yleensä niistä voidaan erottaa ainakin määrittely-, suunnittelu- ja toteutusvaiheet. Määrittelyvaihetta edeltää usein esitutkimukseksi tai tarvekartoitukseksi kutsuttu vaihe.

Kaikkiin vaiheisiin liittyy laadunvarmistustoimenpiteitä, kuten tarkastuksia, katselmuksia ja testausta. Tarkastuksilla ja testauksella pyritään kitkemään virheet järjestelmästä mahdollisimman varhaisessa vaiheessa. Katselmuksia pidetään yleensä vaiheiden päätteeksi. Ne ovat tilaisuuksia, joissa todetaan projektin tilanne ja se, että kaikki vaiheeseen liittyvät tavoitteet on saavutettu ja kaikki sovitut dokumentit on tuotettu.



Kuva 1. Vesiputousmalli

3.1.1 Esitutkimus

Esitutkimuksen tehtävänä on asettaa yleiset järjestelmätason vaatimukset. Esitutkimus vastaa kysymykseen, miksi ohjelmisto tai järjestelmä tulisi tehdä. Esitutkimus on siinä mielessä elinkaaren tärkein vaihe, että vääristä asiakasvaatimuksista ei voi päätyä hyvään järjestelmään. (Haikala & Marijärvi 2004, 37.)

Yrityksellä oli hyvin selkeä idea siitä, minkä näköiset ja mitä ominaisuuksia internetsivustolla tuli olla. Sivuston tuli mainostaa yritystä ja laajentaa asiakaskuntaa ja yrityksen näkyvyyttä.

3.1.2 Määrittely

Määrittelyvaiheessa asiakasvaatimuksia analysoidaan ja niistä johdetaan ohjelmistovaatimukset, jotka määrittelevät toteutettavan järjestelmän. (Haikala ym. 2004, 38.)

Sivustolle suunniteltiin niinkutsuttu flash-ruutu, jossa näkyvät yrityksen logo ja motto. Logoa painamalla siirrytään itse varsinaiselle sivustolle. Etusivulla näkyy yrityksen ajankohtaisia tarjouksia ja tuotteita ja sivun vasemmassa reunassa ovat navigointilinkit.

Sivusto sisältää tietoa yrityksestä, yhteystietoja, vastauspyyntöä, kuvia tuotteista, sekä linkit tuotteiden valmistajien internetsivuille.

3.1.3 Suunnittelu

Suunnitteluvaihe jaetaan usein kahteen tasoon. Aluksi järjestelmä jaetaan mahdollisimman itsenäisiin, toisistaan riippumattomiin osiin, moduuleihin. Usein todetaan, että määrittelyvaiheessa kuvataan, ”mitä” järjestelmä tekee, ja suunnitteluvaiheessa vastaavasti vastataan kysymykseen, ”miten” se tehtävänsä suorittaa. (Haikala ym. 2004, 40.)

Tämän sivusto suunniteltiin kahdessa osassa. Ensimmäisessä osassa suunniteltiin sivustojen ulkoasu, toisessa osassa suunniteltiin sivuston sisältö ja toiminnot. Sivuston ulkoasuun piirrettiin siihen sopivat kuvat sekä ohjelmoitiin sopivat CSS-tyylisivut.

3.1.4 Toteutus

Toteutusvaiheessa ohjelmakoodi kirjoitettiin ja tarvittavat kuvat piirrettiin tai hankittiin. Sivut kirjoitettiin HTML-kielellä, tukena PHP- ja JavaScript-kielillä toteutetut toiminnallisuudet.

3.1.5 Testaus

Testauksen tarkoitus on löytää ohjelmistosta virheitä. Varsinkin tuotekehityksessä testaus sekä siihen liittyvä virheiden jäljittäminen ja korjaaminen saattavat aiheuttaa erittäin merkittävän osan järjestelmän kokonaiskustannuksista, jopa 80 %. (Haikala ym. 2004, 40.)

Tämän sivuston valmistuksen aikana testausta suoritettiin lähes jokaisen ohjelmakoodin jälkeen. Virheitä ilmeni tietenkin, mutta ne korjattiin välittömästi. Sivustoa testattiin eri internetselaimilla, kuten Mozilla Firefoxilla ja Internet Explorerilla. Lopuksi sivustojen HTML- ja CSS-koodit validoitiin W3C:n standardien mukaan.

3.1.6 Ylläpito

Ylläpito on asiakkaan ongelmien ratkomista, virheiden korjaamista, ohjelman muuttamista vaatimusten muuttuessa sekä uusien piirteiden lisäämistä. (Haikala ym. 2004, 41.)

Tekijä huolehtii valmiin sivuston ylläpidosta. Sivustolla olevia tarjouksia ja mainoksia täytyy päivittää aika ajoin ja uusia ominaisuuksia lisätä. Sivustoa ei tule todennäköisesti ylläpitämään toista henkilöä.

4 OHJELMISTON TOTEUTUS

4.1 HTML-toteutus

HTML-sivujen toteutuksessa ei käytetty työkaluja, vaan ohjelmakoodi tuotettiin tekstieditorilla. Tästä syystä koodin tuottaminen oli työläämpää, mutta tulos on selkeämpi ja siksi luotettavampi.

Aloitussivulla näkyvät firman logo sekä motto ”Our point is to give you the best.” Logoa klikkaamalla päästään eteenpäin varsinaiselle sivustolle. Sivut koostuvat kolmesta kehyksestä: vasen, oikea sekä pohjakehys. Vasen kehys sisältää kaksi div-osiota. Div ”vasen” määrittelee koko vasemman kehyksen alueen, sisältää kuvan ”vasen.jpg”, joka toimii kehyksen taustakuvana. Divin z-indeksi on -1. Div ”linkit” määrittelee kehyksen keskustan, jossa ovat sivuston navigaatiolinkit. Linkkien kuvat sijaitsevat kansiossa /Kuvat/Linkit. ”Linkit” divin z-indeksi on oletusarvo, eli 0. Z-indeksit määrittelevät, mikä taso on päällimmäisenä, mikä toisena ja niin edelleen. Tässä yhteydessä ”linkit” on siis päällimmäisenä ja ”vasen” sen takana.

Kehyksen tyylitiedosto on ”tyyli.css”, joka sisältää tiedon divien paikasta ja koosta. Koodissa 2 näkyy tyylitiedoston määrittely.

```
<link rel="stylesheet" type="text/css" href="tyyli.css">
```

Koodi 2. CSS-tiedoston määrittely

Pohjakehyksessä näkyy yrityksen motto ja kehys rajoittaa näkyvän osion kokoa.

Oikea kehys näyttää varsinaisen sivuston sisällön. Etusivulla näkyy top-10 tuotteet -lista, joka sisältää suosituimpia tuotteita sekä ajankohtaisia tarjouksia (”ajankohtaista” div).

Etusivu sisältää myös Somfy expert -logon (”somfy” div) ja jokainen sivu sisältää yrityksen oman logon (”ikkuna” div, joka määrittelee näkyvän oikean kehyksen alueen). Kaikkien oikean kehyksen sivujen tyylitiedot löytyvät ”tyyli_r.css”-tiedostosta.

Yritys-sivu sisältää tietoa yrityksen historiasta, yrityksen yhteystiedot sekä tietoa tarjotuista palveluista. Sivulla näkyy kolme painiketta ja tekstikenttä. Sivulla on kolme diviä, ”Yritys”, ”Yhteystiedot” ja ”Palvelut”, jotka sisältävät kyseisten osien tiedot. Oletusarvona ”Yritys” on näkyvässä ja muilla diveillä on asetettuna arvo ’style="visibility: hidden”’ eli ne ovat piilotettuna. Kun jotain kolmesta painikkeesta klikataan, suorittaa sivu asianomaisen Javafunktion, joka piilottaa kaikki divit ja sen jälkeen asettaa yhden näkyväksi. Javafunktiot on käsitelty tarkemmin kohdassa 4.2.

Yhteystyökumppanit-sivu sisältää taulukossa linkit kaikkien yrityksen yhteystyökumppanien internetsivuille. Sivun head-osiossa oleva <base target="_blank"> koodi tarkoittaa, että sivulla avatut linkit aukeavat uudessa ikkunassa. Taulukossa on kaksi solua vierekkäin.

Kuvia-sivun toteutus on samantyyppinen kuin Yritys-sivun. Sivulla näkyy neljä painiketta, jotka suorittavat Java-funktioita. Funktiot avaavat eri divit, joissa on jokaisessa 10 painiketta, jotka taas suorittavat Java-funktioita ja vaihtavat sivun

keskellä näkyvää kuvaa. Kuvat löytyvät Kuvat-kansion alla omasta kansiostaan (/Kuvat/Markkiisi, /Kuvat/Muut, /Kuvat/Rulla, /Kuvat/Sale).

4.1.1 HTML validointi:

Useimmat sivut Internetissä on kirjoitettu tietokonekielillä (kuten HTML) ja ne sallivat internetohjelmoijien määrittellä tekstin rakenteen, lisätä multimediasisältöä ja määrittää, minkälaisen ulkoasun tai tyylin tuloksella pitäisi olla.

Kuten jokaisella kielellä, näillä on oma *kielioppi, sanasto ja syntaksi*, ja jokaisen näillä tietokonekielillä kirjoitetun dokumentin oletetaan seuraavan näitä sääntöjä.

Kuten luonnolliset kielet, myös ohjelmointikielissä voi olla syystä tai toisesta kirjoitusvirheitä tai kieliopillisia virheitä. Prosessia, jossa tarkistetaan, seuraako dokumentti tarvittavia sääntöjä, kutsutaan validoinniksi, ja käytettävää työkalua kutsutaan validaattoriksi. Dokumentti, joka läpäisee tämän prosessin, on validi.

Validi internetsivu ei ole välttämättä hyvä sivu, mutta invalidilla sivulla on todella pieni mahdollisuus olla hyvä internetsivu.

Internetselaimet hyväksyvät ja yrittävät näyttää sivut, vaikka sivut eivät olisikaan täysin validia HTML-koodia. Yleensä tämä tarkoittaa sitä, että selain yrittää arvata, mitä sivulla oikein haluttiin olevan. Ongelma on, että eri selaimet (tai jopa eri versiot samasta selaimesta) tekevät erilaisia arvauksia samasta virheellisestä rakenteesta, tai pahempaa, jos HTML on *todella* huonoa, selain saattaa mennä toivottoman sekaisin ja tuottaa vain sotkua tai jopa kaatua. (Help for The W3C Markup Validation Service – W3CSchools: 2007.)

Sivustojen validointi on tehty W3C:n standardien mukaisesti. Osoitteessa http://w3schools.com/site/site_validate.asp löytyy ohjelma, joka tarkistaa internetsivun ja kertoo, sisältääkö sivu virheitä. Validointi tarkistaa, onko sivun koodissa tarvittavat ominaisuudet ja määrittelyt suoritettu oikein. Esimerkiksi sivuilla pitää olla määriteltynä dokumentin tyyppi, käytettävä merkistö ja HTML-koodin pitää olla oikein kirjoitettua.

4.1.2 HTML rakenne

Kun kirjoitettava dokumentti on joko HTML- tai XHTML-kielellä toteutettu, on tärkeää lisätä dokumentin tyyppimäärittely. Määrittelyn pitää olla täsmällinen, jotta sillä olisi haluttu vaikutus. Tyyppimäärittely sisältää tiedon kyseessä olevan sivun rakenteesta ja säännöistä. Yleisimmin käytettyjä tyyppejä (koodi 3) ovat Strict (täsmällinen), Transitional (sama kuin Strict, mutta ei niin tarkka) ja Frameset (kehykset).

```
!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

Koodi 3. Dokumentin eri tyyppimäärittelyt.

Sivun rakenteeseen oleellisena osana kuuluu myös käytettävien kirjasinten määrittely (koodi 4). Tämä on tärkeää etenkin pohjoismaissa tuotetuissa dokumenteissa, joihin täytyy määritellä sopiva kirjasintyyppi pohjoismaisten merkkien näkyvyydeksi.

```
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
```

Koodi 4. Dokumentin kirjasintyyppin määrittely.

4.2 JavaScript-toteutus

Osa sivuista käyttää JavaScriptillä ohjelmoituja funktioita, jotka vaihtavat sivuilla näkyviä kuvia tai tekstejä. Javafunktiot löytyvät projektikansion juuresta sijaitsevassa tiedostossa ”funktiot.js”. Funktioiden vieminen erilliseen tiedostoon selkeyttää ja yksinkertaistaa sekä Java-funktioiden että HTML-sivujen tuotantoa ja lukua.

Ensimmäiseksi suoritettavat funktiot määrittelevät käyttäjän selaimen ja sen mukaan määrittelevät, miten funktiot suoritetaan (koodi 5).

```
var selain;
if (document.layers) {selain = "nn4"}
if (document.all) {selain = "ie"}
if (window.navigator.userAgent.toLowerCase().match("gecko")) {
  selain = "gecko"
}
```

Koodi 5. JavaScript-kielillä ohjelmoitu selaimen määrittely.

Tässä esimerkissä (koodi 5) ohjelma määrittelee selaimen yhdeksi kolmesta tyyppistä, jotka ovat

ie: Internet Explorer

nn4: Netscape Navigator, Mozilla...

gecko: Gecko, Opera...

Java-funktioiden suoritus vaihtelee hieman riippuen käytettävästä selaimesta. Esimerkiksi, jos selaimena on Internet Explorer, käsiteltävä kohde on `document.getElementById("Yritys")` ja jos selain on Mozilla tai vastaava, käsiteltävä kohde on `document.layers["Yritys"]`.

Yritys-sivun nappeja painamalla sivu suorittaa yhden kolmesta funktiosta: `Yritys()` (koodi 6), `Yhteystiedot()` tai `Palvelut()`. Jokainen funktio suorittaa ensimmäiseksi toisen funktion: `PiilotaYritys()`, joka piilottaa kaikki Yritys-sivun kolme diviä. Piilottamisen jälkeen palataan takaisin alkuperäiseen funktioon, joka asettaa yhden divin näkyväksi. Näin näkyvissä on aina yksi div.

```
function Yritys() {
  PiilotaYritys(); //Piilottaa kaiken
  if (selain == "gecko" )
    document.poppedLayer = eval('document.getElementById("Yritys)');
  else if (selain == "ie")
    document.poppedLayer = eval('document.getElementById("Yritys)');
  else
    document.poppedLayer = eval('document.layers["Yritys"]');
  document.poppedLayer.style.visibility = "visible"; //Yksi näkyviin
}
```

Koodi 6. JavaScript-kielillä ohjelmoitu eri layerien näkyvyys.

Kuvat-sivun funktiot toimivat hyvin samantyyppisesti. Sivun neljä nappia suorittavat funktion, joka piilottaa kaikki sivun neljä diviä ja asettaa sen jälkeen yhden näkyviin. Divit sisältävät itsessään kymmenen nappia, jotka taas suorittavat eri Java-funktioita. Nämä napit suorittavat asianomaisen funktion, esimerkiksi koodin 7 SaleKuva1(). Nämä funktiot vaihtavat sivun keskellä näkyvän kuvan lähteen.

```
function SaleKuva1()
{
    document.images["Kuva"].src = "Kuvat/Sale/Sale1.jpg";
}
```

Koodi 7. Kuvan lähteen vaihto.

4.3 PHP-toteutus

Sivujen ”Lisää tietoa” osiossa käyttäjä antaa syöttökenttiin yhteystietojansa ja voi esittää kysymyksen, joka lähetetään yritykselle sähköpostina. Tämä sivu on toteutettu PHP-ohjelmointikielellä.

Sivu sisältää lomakkeen, johon käyttäjä voi antaa nimensä, sähköpostinsa, puhelinnumeron ja esittää kysymyksen tai lähettää viestin. Lomake on ohjelmoitu HTML-koodina ja se on talletettu muuttujaan *\$form_block*. Suorittamalla komento *echo "\$form_block"*; sivu tulostaa koko lomakkeen.

Kun käyttäjä on antanut tietonsa ja painaa Lähetä kysymys -painiketta, ohjelma tarkistaa syötteet (koodi 8). Esimerkiksi nimestä tarkistetaan, onko sen pituus suurempi kuin nolla tai onko annettu arvo tekstiä vai numeerinen. Jos virheentarkistuksessa ilmenee virheitä, ohjelma ilmoittaa niistä käyttäjälle, eikä lomaketta käsitellä loppuun asti. Kun syötteet läpäisevät virheentarkistuksen, annettu kysymys tai viesti lähetetään sähköpostilla yritykselle. Yksinkertainen PHP-funktio *mail(\$to, \$subject, \$msg, \$mailheaders)*; lähettää sähköpostin osoitteeseen *\$to*, lisää sähköpostiin aiheen *\$subject*, itse viestin *\$msg* sekä viestin tietoja *\$mailheaders*.

```
if ($_POST[kysymys_nimi] == "" || is_numeric($_POST[kysymys_nimi])) {  
    $nimi_err = "<font color=red>Anna nimesi!</font><br>";  
    $ok = "no";  
}
```

Koodi 8. PHP-kielillä ohjelmoitu lomakkeen arvon tarkistus.

4.4 Graafinen toteutus

Yritykselle suunniteltiin uusi ulkoasu ja internetsivuille sovellettiin saman tyyppinen ulkoasu (kuva 3). Grafiikat valmistettiin piirtämällä Corel Paint Shop Pro:lla.



Kuva 2: Oikea.jpg

Kuva 2, Oikea.jpg on piirretty yrityksen ulkoasuun sopivaksi. Sivuston vasemmassa reunassa oleva kuva, Vasen.jpg, on saman näköinen kuin Oikea.jpg, mutta se on peilikuva.



Kuva 3: Flash

4.5 Sivuston rakenne



Kuva 4. Etusivun näkymä.

Sivustot sisältävät kolme eri kehystä, joiden määrittely näkyy koodissa 9. Vasen kehys sisältää navigointilinkit, alakehys rajaa sivuston halutun kokoiseksi, ja oikea kehys näyttää varsinaisten sivujen sisällön.


```

<frameset rows="550,*" border=no frameborder=0 framespacing=0>
  <frameset cols="208,*" border=no frameborder=0 framespacing=0>
    <frame name="left" src="vasen.htm" scrolling="no" noresize target="_self">
    <frame name="right" src="etusivu.htm" scrolling="auto" noresize
target="_self">
  </frameset>
</frameset>
<frameset cols="*" border=no frameborder=0 framespacing=0>
  <frame src="pohja.htm" scrolling="no" noresize>
</frameset>

```

Koodi 9. Kehysten määrittely.

Sivuston etusivuna toimii ns. flash-sivu, jossa näkyvät yrityksen logo ja motto (Kuva 3). Kuvaa klikkaamalla tai 5 sekunnin kuluttua selain siirtyy varsinaiselle etusivulle, jossa näkyy sivun vasemmassa reunassa navigointilinkit ja oikealla ajankohtaisia tarjouksia ja uutisia (Kuva 4).

Yritys-alasivulla käyttäjä voi lukea yrityksen historiaa, sen tarjoamia palveluja sekä yhteystiedot. Sivun oleellinen ominaisuus on JavaScript-funktioilla toteutettu eri div-osioiden näkyvyys.

Yhteistyökumppanit-alasivulta löytyvät kaikki yrityksen tavarantoimittajat ja valmistajat taulukossa (koodi 10). Yrityksistä näkyy logo sekä se, mitä tuotteita se valmistaa. Logoa klikkaamalla aukeaa uusi sivu kyseessä olevan yrityksen omille internetsivuille.

```

<table width="570" cellspacing="20" bgcolor=#f0f0f0 border=1>
  <tr>
    <td width="50%"><center>
      <a href="http://www.solar.fi"></a></center>
      <br>Säleverhot ja hyönteisverkot</td>
    <td width="50%"><center><a href="http://www.turunkaihdin.fi">
      </a></center>
      <br>Säleverhot, rullaverhot ja pliseet
    </td>
  </tr>
</table>

```

Koodi 10. Yhteistyökumppanit-taulukkoa.

Lisää tietoa -alasivulla käyttäjä voi esittää kysymyksen, joka lähetetään sähköpostilla yritykselle. Sivusto on toteutettu PHP-kielillä (kohta 4.3). Käyttäjä

antaa nimen, sähköpostiosoitteen tai puhelinnumeron ja esittää kysymyksen. Mikäli käyttäjä on antanut tarvittut tiedot, ohjelma lähettää sähköpostin. Koodissa 11 on esimerkki tietojen tarkistuksesta.

```
if ($_POST[kysymys_teksti] == "")
{
    $teksti_err = "<font color=red>Anna viesti!</font><br>";
    $ok = "no";
}
```

Koodi 11. PHP-virheentarkistusta.

Kuvia-alasivulla käyttäjä voi katsella kuvia yrityksen tarjoamista tuotteista. Sivusto käyttää samaa periaatetta kuin Yritys-sivu. Nappeja painamalla tietyt osat tulevat näkyviin, kun taas toiset osat piilotetaan.

5 YHTEENVETO

Projektin lähtökohtana oli saada toimiva, selkeä ja edustava internetsivusto yritykselle. Sivuston valmistukseen harkittiin useita eri työkaluvaihtoehtoja, kuten Microsoftin Frontpagea, Zend development studiota sekä muita enemmän tai vähemmän graafisia työkaluja. Pian huomattiin, että graafisilla työkaluilla sivujen valmistus on yksinkertaista ja nopeaa, mutta työkalujen valmistama ohjelmakoodi on lähes käyttökeltontonta. Esimerkiksi ohjelmat kirjoittavat paljon ylimääräistä ja turhaa spagettikoodia.

Lopuksi päädyttiin siihen tulokseen, että ohjelmakoodin kirjoittaminen tekstieditorilla on paras vaihtoehto. Täten koodista tuli selkeää ja helposti luettavaa, ja sivustot toimivat kuten niiden pitää. Tällä menetelmällä ohjelmointi vei hieman enemmän aikaa, mutta se oli varmempaa.

Sivustoilla käytetyt grafiikat on joko valmistettu itse tietokoneella piirtämällä tai toisten yritysten kohdalla kyseessä olevien yritysten omilta internetsivuilta hankituilla kopioilla. Tarvittavat luvat kyseisiin logoihin ja valokuviin on hankittu.

LÄHDELUETTELO

Haikala, Ilkka & Marijärvi, Jukka 2004. Ohjelmistotuotanto. 10. painos. Hämeenlinna: Talentum Media Oy.

Verkkodokumentit:

Pääkkönen, Pekka 1998. CSS – Tyylimäärittely. [Viitattu 5.9.2007] Saatavissa: <http://www.hytti.uku.fi/~ppaakkon/css.html>.

Help for The W3C Markup Validation Service – W3CSchools 2007 [Viitattu 15.6.2007]. Saatavissa: http://validator.w3.org/docs/help.html#validation_basics.

HTML – Wikipedia 2007. [Viitattu 4.6.2007]. Saatavissa: <http://fi.wikipedia.org/wiki/Html>.

Graham, Ian 1998. Introduction to HTML. [Viitattu 18.9.2007]. Saatavissa: http://www.utoronto.ca/webdocs/HTMLdocs/NewHTML/html_intro.html.

Javascript – Wikipedia 2007. [Viitattu 19.9.2007]. Saatavissa: <http://fi.wikipedia.org/wiki/Javascript>.

Saarikumpu, Osmo 2006. JavaScriptin alkeet, osa 1. [Viitattu 19.9.2007]. Saatavissa: <http://weppipakki.com/js/opas/alkeet1.htm>.

PHP: Introduction 2007. [Viitattu 18.9.2007]. Saatavissa: <http://www.php.net/manual/en/introduction.php>.

PHP käsikirja 2005. [Viitattu 4.6.2007]. Saatavissa: <http://www.php.net/manual/fi/preface.php>.

PHP – Wikipedia. 2007. [Viitattu 4.6.2007]. Saatavissa: <http://fi.wikipedia.org/wiki/Php>.