



TAMPEREEN
AMMATTIKORKEAKOULU

OPINNÄYTETYÖ

WWW-SOVELLUKSEN SIVUPOHJAN KEHITTÄMINEN
Case Mediamaisteri Group Oy

Mikko Siippanen

Tietojenkäsittelyn koulutusohjelma
Toukokuu 2008
Työn ohjaaja: Ville Haapakangas

TAMPERE 2008



Tekijä(t)	Mikko Siippanen	
Koulutusohjelma(t)	Tietojenkäsittely	
Opinnäytetyön nimi	WWW-sovelluksen sivupohjan kehittäminen Case Mediamasteri Group Oy	
Työn valmistumis- kuukausi ja -vuosi	Toukokuu 2008	
Työn ohjaaja	Ville Haapakangas	Sivumäärä: 40

TIIVISTELMÄ

Kehittyvässä yrityksessä ei aina ole aikaa panostaa sovelluksen käytettävyyteen tai miettiä kustannustehokkaampia ratkaisuja totuttujen menetelmien tilalle. Asiat toteutetaan kiireellisen aikataulun vuoksi mahdollisimman nopealla tekniikalla, eikä tämä aina ole paras ratkaisu.

Opinnäytetyön toimeksiantaja, Mediamasteri Group Oy, halusi korjata tuotekehittelynsä puutteita kehittämällä uuden sivupohjan, joka tulisi toimimaan verkko-oppimishjelmista koostuvan mCompass-tuoteperheen sovellusten selkärankana. Opinnäytetyön tavoitteena oli päivittää sivupohjan ulkoasu nykyaikaiseksi, parantaa sen käytettävyyttä sekä selkeyttää sovelluksen rakennetta myös kehittäjän näkökulmasta. Näihin asioihin pyrittiin vaikuttamaan sivupohjan rakenteellisella ja visuaalisella yhdentämisellä sekä kehitysprosessin tuotoksena syntyneen dokumentaation avulla. Työn tulokset dokumentoitiin sähköiseen muotoon tyylikäsikirjaksi, jonka päämääränä on toimia jatkossa oppaana ja tukipilarina mCompass-tuoteperheen sovellusten jatkokehittäjille Mediamasteri Groupissa.

Sivupohjan suunnittelu alkoi hankkimalla tietoa nykyisistä sovelluksista ja niiden kehitystarpeista. Taustamateriaalia kerättiin yrityksen henkilökunnalta haastatteluiden muodossa sekä aiheesta tehtyyn sähköiseen ja kirjalliseen materiaaliin tutustuen. Sivupohjan ulkoasu ja rakenne toteutettiin CSS-tyylimääreitä ja XHTML-kieltä hyväksikäyttäen. Käyttöjärjestelmän dynaamiset osiot luotiin PHP- ja JavaScript-kielillä.

Tämä opinnäytetyö kuvaa projektin kehityskaarta suunnittelusta valmiiksi tuotteeksi. Työn tuloksena syntyi valmis sivupohja ja tyylikirjasto, jotka jäivät valmistumisensa jälkeen Mediamasteri Groupin käyttöön.



Author(s)	Mikko Siippanen	
Degree Programme(s)	Business Information Systems	
Title	Layout development for a web application Case Mediamasteri Group Oy	
Month and year	May 2008	
Supervisor	Ville Haapakangas	Pages: 40

ABSTRACT

An evolving corporation can easily drift to a situation where there's not enough time or resources to invest to the usability and the financial efficiency of the product. Because of the tight schedules the priority is to launch the product as soon as possible and sometimes even without proper testing. In the long run that's not always the best possible practise.

The employer of this thesis was Mediamasteri Group Oy, who wanted to revise their products' deficiencies by making a new layout for an e-learning product family called mCompass. The goals of this thesis were to modernize the template's appearance while improving the usability of the interface and simplifying the structure of the application. To accomplish these goals the aim was to visually and structurally unite the mCompass products. The yield of the process was documented to an e-library of the style sheets used in the layout. The purpose of the library was to work as a structural cornerstone for the web developers in Mediamasteri Group, providing explanations and the logic behind the structural and visual solutions used in the layout.

The designing of the layout started with obtaining information of the existing applications and sorting out the need for redevelopment in the product family. The background material was received in the form of staff interviews and by researching the literature made of the subject. Web techniques used in the project were: CSS, XHTML, PHP and JavaScript.

This thesis describes the trajectory of the project from the designing stage to the completion of the product. As a result a new layout along with a documentation of the style sheets were born. After the completion of the project the end product was left to the use of Mediamasteri Group.

Käsitteet ja lyhenteet

CSS	Cascading style sheet Tyyli tiedosto, joka määrittää WWW-dokumentin ulkoasun.
HTML	HyperText Markup Language Avoimesti standardoitu kuvauskieli, joilla voidaan rakentaa WWW-sivuja.
JavaScript	Web-ympäristössä käytettävä komentosarjakieli.
Palvelin	Tietokoneessa suoritettava palvelinohjelmisto sekä ohjelmistoa suorittava tietokone. Palvelinohjelmistojen tehtävänä on tarjota erilaisia palveluja joko paikallisesti tai tietoverkon välityksellä palvelinta tai sovellusta käyttävälle asiakkaalle
PHP	Hypertext Preprocessor Web-palvelinympäristössä käytettävä ohjelmointikieli.
RGB-värimalli	Väriavaruus, jossa värejä yhdistellään käyttämällä punaista, vihreää ja sinistä väriä. RGB-värimallia käytetään pääosin sähköisissä näyttölaitteissa, kuten tietokoneissa ja televisioissa.
Selain	Tietokoneohjelma, jonka avulla käyttäjä voi katsella WWW-sivuja.
Validaattori	Ohjelma, joka tarkistaa koneellisesti dokumentin oikeellisuuden dokumenttityypin mukaan.
Validointi	Prosessi, jossa tarkistetaan, että prosessin kohde täyttää jotkin tietyt kriteerit. Käyttökelpoisen WWW-dokumentin täytyy olla oikeellinen eli validi.
W3C	World Wide Web Consortium Kansainvälinen yritysten ja yhteisöjen yhteenliittymä, joka ohjaa julkaisukielten kehitystä.
WAI	Web Accessibility Initiative Internetin esteettömyyden parantamiseen tähtäävä ohjeistus.

WWW	World Wide Web Internetissä toimiva hajautettu hypertekstijärjestelmä.
XHTML	eXtensible Hypertext Markup Language HTML:stä kehitetty WWW-sivujen merkintäkieli, joka täyttää XML:n muotovaatimukset
XML	eXtensible Markup Language Rakenteellinen kuvauskieli, jolla tiedon merkitys on kuvattavissa tiedon sekaan. XML-kieltä käytetään sekä formaattina tiedonvälitykseen järjestelmien välillä että formaattina dokumenttien tallentamiseen.

Sisällysluettelo

Käsitteet ja lyhenteet	4
1 Johdanto.....	7
2 Käytetyt tekniikat	10
2.1 HTML	10
2.2 XHTML	11
2.2.1 XHTML kriteerit.....	11
2.2.2 XHTML-dokumentti	12
2.3 CSS-tyyliohjeet	13
2.3.1 Tyylien määrittäminen	13
2.3.2 DIV-elementit	15
2.3.3 Pseudoluokat	16
2.4 PHP	16
2.5 JavaScript	17
2.6 Validointi	17
3 Käytettävyys WWW-suunnittelussa	18
3.1 Rakenne ja navigointi.....	18
3.2 Tekstin luettavuus	19
4 Lähtötilanne.....	20
4.1 Käytettävyyden ongelmat.....	21
4.2 Tekniset ongelmat	21
4.3 Vanha ulkoasu.....	22
4.4 Lomakkeet.....	23
5 Uuden sivupohjan mahdollisuudet ja rajoitukset	24
6 Toteutukseen liittyviä asioita	25
6.1 Rakenteellisten osien jakaminen tiedostoihin	25
6.2 Tiedostojen sijoittelu	25
6.3 Testaaminen	25
7 Uusi sivupohja	27
7.1 Sivupohjan uudistukset	28
7.2 Visuaalinen ilme.....	29
7.3 Ensisijainen navigaatio	29
7.4 Toissijainen navigaatio	30
7.5 Lomakkeet.....	31
7.6 Taulukot	32
7.7 Typografia.....	33
7.8 Otsikot.....	33
7.9 Värit	33
8 Tyylikäsikirja	35
9 Pohdinta	36
9.1 Yhteenveto	36
9.2 Tulosten arviointi ja tulevaisuus	36
Lähteet.....	39
Liitteet	40
Liite 1 – Graafinen ohjeistus	40

1 Johdanto

WWW-selainvalmistajien aggressiivinen kilpailu markkina-asemista luo web-kehittäjille uuden selainversion saapuessa aina kourallisen uusia haasteita entisten rinnalle. WWW-sivustolle on olemassa lukemattomia toteutustapoja, ja ratkaisujen määrä kasvaa päivittäin. Yhden toimivan tekniikan sijaan web-kehittäjille on tarjolla satoja kehoja toteutuksia, joiden käyttöä hankaloittaa selainohjelmien toisistaan eroavat ominaisuudet. Toimivan ratkaisun kehittämien yhdelle selainohjelmalle saattaa tuottaa vain lisää ongelmia toisella järjestelmällä. Oikeaoppisen ja standardien mukaisen sivupohjan luonti on haasteellinen urakka kokeneellekin web-kehittäjälle, ja laajan, dynaamisen sivuston rakentaminen vaatii suurta määrää etukäteissuunnittelua sekä useiden eri tekijöiden huomioimista.

Etukäteissuunnittelun puute näkyy selvimmin tuotteissa, jotka tarjoavat kuluttajalle jotain uutta ja ennennäkemätöntä. Uusia, aiemmin markkinoilta puuttuneita tuotteita luovan kasvavan yrityksen ei ole aina mahdollista vastata tuotteiden suureen kysyntään riittävän nopeasti, etenkin niin että tuotekehittäjissä pystyttäisiin huomioimaan kaikki tarvittavat tekijät tuotteen toimivuuden suhteen. Taloudellisten resursien sekä ajan puute ovat tekijöitä, jotka pakottavat usein yrityksen julkaisemaan tuotteensa keskeneräisinä, ilman asiaankuuluvaa toiminnallisuuden ja käytettävyyden todentamista.

Opinnäytetyön toimeksiantaja, Mediamasteri Group Oy, kaipasi uusia ideoita mCompass-tuoteperheen sovelluksien kehitysmenetelmien ja käyttöliittymän parantamiseksi. Tuotteilla ei ollut yhteneväistä rakennetta ja ulkoasu kaipasi päivitystä. Syyskuussa 2007 alkoi mCompass-tuoteperheen uudistamisen suunnittelu ja toteuttaminen. Uudistusten tavoitteena oli yhdentää tuotteet rakenteellisesti ja visuaalisesti. Uudistusten pohjalta rakennettiin tyylikäsikirja, sähköiseen muotoon dokumentoitu CSS-opas, jonka päämääränä oli jatkossa toimia kulmakivenä tuotteiden käyttöliittymän ja käytettävyyden kehittämisessä.

Järkevin tapa erotella ulkoasuun liittyvät ominaisuudet varsinaisesta sovelluksesta on käyttää tyylimääreitä. Tyyli tiedostojen käyttö mielletään web-kehittäjien parissa suhteellisen helposti opittavaksi tekniikaksi, mutta käytön hallitseminen ja tyylimääreiden looginen hyödyntäminen ovat kuitenkin eri asioita. Kokeneetkin CSS-kehittäjät joutuvat uusien ongelmien eteen päivittäin ja päätyvät usein korjaamaan ongelmia tilapäisillä hätäratkaisuilla perehtymättä asianmukaisesti varsinaiseen ongelmaan. Ongelman sivuuttamisesta seuraa aina uusia esteitä kierrettäväksi ja oravanpyörä näyttää loputtomalta.

Toimeksiantaja

Opinnäytetyöni toimeksiantajana oli Mediamasteri Group Oy, joka on asiakaslähtöisen oppimisen ja osaamisen asiantuntijayritys. Yrityksen päätoimiala on koulutus ja konsultointi IT-alalla avoimen lähdekoodin verkko-oppimisympäristöjä hyödyntäen. Mediamasteri Groupin tavoitteena on tuottaa asiakkailleen järkeviä ratkaisuja, jotka mahdollistavat asiakkaan voimavarojen keskittämisen osaamisen kehittämiseen. Yritys aloitti toimintansa Finlaysonin toimitiloissa vuonna 2000, työllistäen kokopäiväisesti neljä henkilöä. Tällä hetkellä Mediamasteri Group Oy on laajentanut konserniaan myös Helsinkiin, Hämeenlinnaan ja Turkuun, työllistäen vakituisesti 30 ihmistä sekä vaihtelevan määrän osa-aikaisia konsultteja, kouluttajia ja freelancer-työntekijöitä.

Mediamasteri Groupilla ei lukuisista tuoteperheistään ja sovelluksistaan huolimatta ole aikaisemmin ollut käytössään valmista muokattavissa olevaa sivupohjaa. Toimeksiantajalla oli ennen työn aloittamista olemassa näkemys siitä mitä vaatimuksia sivupohjan ja tyylikäsikirjan pitäisi täyttää, mutta heiltä puuttui tyylimäärittelyihin perehtynyt osaa ja toteuttamaan työn. Työn projektiluonteen takia tehtävään palkattiin harjoitteluaan tai opinnäytetyötään suorittava opiskelija, jolla oli kokemusta niin palvelinympäristön ohjelmointikielistä kuin tyylimäärittelyillä tehdyistä staattisista sivustoista.

mCompass-tuoteperhe

mCompass on työntekijöiden perustietojen, työ- ja koulutushistorian sekä osaamisen kartoittamiseen ja tallentamiseen tarkoitettu työkalu. Tuote soveltuu parhaiten yrityksille joilla on useita toimipisteitä ja/tai kymmeniä työntekijöitä. Tuote lepää Moodle-alustan päällä. Mediamasteri Group tekee visuaalisen räätälöinnin sovitusti asiakkaan näköiseksi, hoitaa tuotteen ylläpidon ja tukipalvelun, konsultoi asiakasta ja osallistuu tarvittaessa tuotteen sisällön suunnitteluun.

Tavoitteet toimeksiannolle

Opinnäytetyön tavoitteena oli luoda mCompass-tuoteperheen dynaamisille WWW-sovelluksille yhtenäinen visuaalinen ulkoasu sekä tyylikäsikirja, johon on dokumentoitu sivupohjan rakenteelliset ja tyylimäärittelyihin liittyvät ominaisuudet. Uuden sivupohjan vaatimuksia olivat käytettävyys, nykyaikaisuus ja tehokkuus jatkokehityksessä.

Kustannustehokkuuden parantamiseen pyrittiin sovellusten rakenteen uudistamisella ja yhtenäistämällä sekä tyylikäsikirjan luomisella. Päämääränä oli saada sovellukset palvelemaan yrityksen käyttötarkoituksia paremmin lisäämällä sovellukseen tyylimääreitä ja dynaamista palvelintekniikkaa hyödyntäen rakennettuja automatisoituja toiminnallisuuksia. Sovelluskehittäjien urakkaa pyrittiin helpottamaan tyylikäsikirjalla, johon dokumentoitiin ratkaisumalleja työn edetessä vastaan tuleville käytännön ongelmille.

Käytettävyyttä pyrittiin parantamaan käyttöliittymän ja sovelluksen ydinsisällön ulkoasun uudistamisella. Helppokäyttöisyyden sekä sovellusten kevyenä pitämisen lisäksi tuli ottaa huomioon rakenteiden ja niissä käytettävien tyylien ulkoasu markkinoinnin näkökulmasta.

Sovellukset olivat henkilökunnan mielestä ulkoasultaan vanhanaikaisia. Niiden ilme haluttiin päivittää nykyaikaisemmaksi, kuitenkin käytettävyydestä ja luettavuudesta tinkimättä. Yksi ulkoasu-uudistuksen tavoitteista oli yhtenäistää mCompass-tuoteperheen sovellukset graafisesti saman näköiseksi. Ulkoasuun haluttiin piirteitä Mediamaisteri Groupin aikaisemmissa tuotteissa käytetystä meriaiheisesta teemasta sekä graafisen ohjeistuksen linjoista. Ohjeistuksen määrittelyjä noudattaen on toteutettu aikaisemmin mm. painettua markkinointimateriaalia ja Mediamaisteri Groupin WWW-sivut.

2 Käytetyt tekniikat

Verkko-ohjelmointi voidaan Keräsen, Lambergin ja Penttisen (2006: 30) mukaan jakaa kahteen osaan: asiakaspuolen ja palvelinpuolen ohjelmointiin. Asiakkaalla tarkoitetaan tässä kontekstissa selainohjelmaa tai muuta järjestelmää, jolla palveluja käytetään. Selaimilla luettavia asiakaspuolen julkaisukieliä ovat muun muassa HTML, XHTML ja CSS. Kuvauskielillä rakennetaan web-sivuja, joita selataan päätelaitteissa näkyvällä selaimella. Palvelinpuolen ohjelmoinnilla tarkoitetaan palvelimella sovelluksia, joilla tuodaan dynaamista sisältöä web-sivulle. Sovelluksia voidaan ohjelmoida käyttämällä PHP- tai Java-ohjelmointikieliä. Ohjelmointikielillä tuotetaan web-sivuille sellaista sisältöä, mitä julkaisukielillä ei pystytä luomaan.

2.1 HTML

Useimmat WWW-sivut tehdään HTML-kielillä. HTML on yksinkertainen ja helposti opittava julkaisukieli, joka koostuu sivun rakennetta kuvaavista elementeistä sekä niiden attribuuteista. HTML-kielen kehityksestä vastaa W3C, mutta myös selainvalmistajat ovat tehneet omia ratkaisuja selainten tulkitsemaan HTML-kielen. (Keränen ym. 2006: 30.)

HTML-dokumentti

HTML-dokumentti on dokumentti, joka sisältää HTML-kielen muotoilukomentoja ja joka muodostaa siten ohjeet internet-selaimessa esitettävien WWW-sivujen loogiseen tai fyysiseen muotoiluun. HTML-dokumentin tiedostopääte on .htm tai .html. (Linjama 2001: 32.)

HTML-kielen avulla hypertekstidokumentteihin voidaan lisätä otsikotietoja, taulukoita, lomakkeita, listoja, kuvia ym. Hyperlinkkeinä toimivat dokumentin osat, kuten kuvat, sanat ja painikkeet. (Linjama 2001: 13.)

HTML-kieli sisältää elementtejä, jotka on kirjoitettu toistensa sisään. HTML-dokumentilla on puumainen rakenne ja sillä voi vain yksi juurielementti, joka on html. (Linjama 2001: 62.)

HTML-dokumentin täytyy sisältää <html>, <head> ja <body> -elementit. Kuvassa 1 on esimerkki HTML-dokumentin perusajatuksista Linjaman (2001: 62) mukaan. Selvennyksen vuoksi koodiesimerkkiin on lihavoitu elementit sekä laitettu rivinumerot, joita ei tarvitse kirjoittaa varsinaiseen koodiin.

```

01  <html>
02      <head>
03          <title>Esimerkkisivu</title>
04      </head>
05      <body>
06          <h1>Otsikko</h1>
07          <p>Sisältöä</p>
08      </body>
09  </html>

```

Kuva 1. Esimerkki HTML-dokumentin perusrakenteesta.

2.2 XHTML

XHTML on HTML-kielestä kehitetty julkaisukieli, joka sisältää kaikki HTML-kielen elementit ja ominaisuudet, mutta elementtejä käytetään XML-kielen säännöin. XHTML:n ja HTML:n eroina ovat XHTML:n tiukemmat muutosäännöt. XHTML ei salli HTML-kielen tavoin virheellistä koodia, joten sivujen tulee olla aina rakenteellisesti oikeaoppisesti muotoiltuja. (Keränen ym. 2006: 31.)

”XHTML ja XML laajentavat HTML-standardia ja kuvaavat lisäksi dokumentin rakenteen loogisesti järkevämmiin. Tällöin myös internetin hypertekstijärjestelmän valtava tietomassa jäsentyy loogisemmin.” (Linjama 2001: 9.)

2.2.1 XHTML kriteerit

Yleisimmät kriteerit XHTML-dokumentin luonnissa ovat seuraavat:

- Kaikki elementit ja attribuutit kirjoitetaan pienaakkosin.
- Kaikki elementit täytyy sulkea, myös yksiosaiset (
).
- Juurielementti on aina html.
- Elementtien avaus- ja sulkujärjestyksen tulee täsmätä.
- Attribuutin arvo pitää olla lainausmerkeissä.
- Kaikilla attribuuteilla pitää olla arvo.
- Pienempi kuin -merkki (<) ja ampersand-merkki (&) tulee kirjoittaa sisältöön entiteetein < ja &.

2.2.2 XHTML-dokumentti

HTML- ja XHTML-dokumentit tulee teoriassa aina aloittaa dokumentin tyypimäärittelyllä. Se määrittelee dokumentin tyyppin, sen sisältämät elementit ja kieliopin. Tyypimäärittely lisätään dokumentin alkuun käyttäen !DOCTYPE-merkintää. HTML-dokumentit toimivat selaimissa ilman dokumenttityypin määrittelyä, mutta XHTML-dokumentissa tyypimäärittely on välttämätön. (Keränen ym. 2006: 42.)

XHTML-dokumentissa html-elementtiin liitetään lisäksi xmlns-attribuutti, joka kertoo käytettävän nimiavaruuden. Xmlns-attribuutin arvoksi annetaan nimiavaruuden osoite. (Keränen ym. 2006: 43.) XHTML-dokumentin rakenne voi olla esimerkiksi kuvan 2 kaltainen (lihavoituna HTML-elementit).

```

01  <?xml version="1.0" encoding="utf-8"?>
02  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
      Strict//EN"
      "http://WWW.w3.org/TR/xhtml1/DTD/xhtml1-
      strict.dtd">
03  <html xmlns="http://WWW.w3.org/1999/xhtml"
      xml:lang="fi">
04      <head>
05          <title>Otsikko</title>
06          <meta http-equiv="Content-Type"
      content="application/xhtml+xml;
      charset=utf-8" />
07      </head>
08      <body>
09          <p>Sisältöä</p>
10      </body>
11  </html>

```

Kuva 2. Esimerkki XHTML-dokumentin rakenteesta.

Ensimmäisellä rivillä ilmoitetaan XML-versio, -rakenne ja merkistö. Toisella rivillä on dokumentin tyypimäärittely. HTML 4.01- ja XHTML-kielen 1.0-versioissa käytetään Strict-, Transitional- ja Frameset-dokumenttityyppejä. XHTML:n erona HTML-määrittelyyn on kuitenkin se, että tyypimäärittelyissä on kerrottava dokumenttityypin sijainti, joka määritetään WWW-osoitteella. Kolmannella rivillä määritetään juurielementti, joka on aina html. Juurielementin attribuutissa kerrotaan käytettävä nimiavaruus. Loppuosa koodista on tavallista HTML-kieltä, lukuun ottamatta riviä kuusi, jossa meta-elementillä määritetään dokumenttia kuvaava metatieto.

2.3 CSS-tyyliohjeet

CSS-kielellä laaditaan (X)HTML-dokumentin ulkoasua kuvaavia tyyliohjeita. Tyyliohjeiden tarkoituksena on pitää dokumentin ulkoasuun liittyvät määritykset erillään muusta rakenteesta. Näin dokumentin ulkoasu on helpommin muokattavissa, koska muutokset kohdistuvat jokaiseen dokumenttiin, jossa tyyliohjeita käytetään. Tyyliohjeet laaditaan joko ulkoiseen tyylitiedostoon tai dokumenttikohtaisesti HTML-rakenteen head-osioon. (Keränen ym. 2006: 33, 92.)

Tyylitiedostoja ei kuitenkaan suositella käytettävän sisäisesti, jos tyylimäärittelyt tulevat koskemaan useampaa sivua. Dokumenttikohtaista tyylimäärittelyä käytettäessä muutokset joudutaan tekemään sivu kerrallaan, kun taas ulkoisessa tyylitiedostossa olevat tyylimääreet voidaan päivittää kerralla kaikkiin sivuihin. Ulkoinen tyylitiedosto tallennetaan css-päätteisenä ja se linkitetään jokaisen dokumentin head-osioon import-komennolla. Kuvassa 3 on esimerkki (X)HTML-dokumenttiin linkitetystä tyylitiedostosta (tyylitiedosto.css).

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

Kuva 3. Esimerkki (X)HTML-dokumenttiin linkitetystä tyylitiedostosta.

Sivulle voidaan tuoda tyylimäärittelyitä useammasta kuin yhdestä tiedostosta. Tällöin myöhemmin määritelty tyyli korvaa vastaavat aiemmin määritellyt tyylit.

2.3.1 Tyylien määrittäminen

Tyylien määrittämisen perusidea on muuttaa olemassa olevien HTML-elementtien ominaisuuksia. Yksittäinen tyylivalitsin on nimetty HTML-dokumentissa esiintyvien HTML-elementtien mukaisesti ja valitsimeen tehtävät muutokset vaikuttavat kyseisen HTML-elementin sisällön (esimerkiksi tekstin) ulkoasuun. (Linjama 2001: 504.)

Tyylimäärittelyn perusstyntaksi koostuu valitsimesta, ominaisuudesta ja arvosta. Tyylin määrittely alkaa tyylin nimellä (valitsin), eli sen HTML-elementin nimellä, johon tyylimäärittelyt vaikuttavat. Välittömästi tyylin nimen jälkeen tulee osa, jossa kerrotaan tyylin ominaisuudet. Kaikki tyylin ominaisuudet kerätään lohkokon aaltosulkeiden ({ ja }) väliin. Tyylin ominaisuuden nimi ja siihen liittyvä arvo täytyy antaa kaksoispisteellä (:) toisistaan erotettuna muodossa ominaisuus:arvo;. Puolipisteellä erotetaan saman lohkon sisällä olevat ominaisuuksien määrittelyt toisistaan. (Linjama 2001: 504-505.)

Kuvan 4 esimerkissä määritetään h1-otsikkoelementille väriksi punainen:

```
h1 {
  color:red;
}
```

Kuva 4. Tyylimäärittelyssä H1-otsikkoelementille annetaan punainen väri.

HTML-elementeille voidaan määrittellä aliluokkia, joiden avulla yhdelle HTML-elementille voidaan määrittellä useita erilaisia esitystapoja. Tyylien aliluokkia voidaan määrittellä antamalla tyyllille alkuperäisen tyylin nimestä ja aliluokasta muodostuva yhdistelmänimi. Uuden tyylin nimi on muotoa tyyli.luokka, jossa tyylin ja aliluokan nimi erotetaan toisistaan pisteellä (.). (Linjama 2001: 517.) Aliluokan käytöstä esimerkkinä kuva 5, jossa h1-tyylin aliluokalle (h1.etusivu) määritetään tekstin väri siniseksi.

```
h1.etusivu {
  color:blue;
}
```

Kuva 5. H1-otsikkoelementin aliluokka h1.etusivu määritetään siniseksi.

Jos aliluokkaa ei haluta yhdistää mihinkään tiettyyn tyyliin, voidaan se Linjaman (2001: 518) mukaan esitellä ilman varsinaista tyylin nimeä. Aliluokkaa voi tällöin käyttää minkä tahansa elementin yhteydessä. Kuvassa 6 on esimerkki aliluokasta, jota ei ole yhdistetty tiettyyn tyyliin.

```
.etusivu {
  color:blue;
}
```

Kuva 6. Aliluokka esitettyinä ilman tyylin nimeä.

Kuvassa 7 (X)HTML-dokumentissa tyylien aliluokka otetaan käyttöön määrittämällä halutulle elementille class-attribuutti.

```
<h1 class="etusivu">Otsikkoteksti</h1>
```

Kuva 7. Aliluokka otettuna käyttöön (X)HTML-dokumentissa.

Linjaman (2001: 519) mukaan edelliset esimerkit voitaisiin toteuttaa myös yksilöivän id-attribuutin avulla. Id-attribuutti antaa elementille yksiselitteisen tunnuksen ja samasta dokumentista ei saa löytyä samalla toista id-attribuutilla olevaa elementtiä. Jos samaa aliluokkaa halutaan käyttää useamman elementin yhteydessä, niin tällöin dokumentissa täytyy käyttää class-attribuuttia. Tässä tapauksessa (kuva 8) tyylitiedostossa käytetään class-attribuutin pisteen tilalla id-attribuutin #-merkkiä ja dokumentissa class-sanalla tilalle vaihdetaan sana id.

```
#etusivu {
    color:blue;
}

<h1 id="etusivu">Otsikkoteksti</h1>
```

Kuva 8. Id-attribuutilla yksilöity h1-elementti tyylitiedostossa ja (X)HTML-dokumentissa.

2.3.2 DIV-elementit

Div-elementti on asemointilohkon muodostava elementti. WWW-sivuja tehtäessä on usein tarve rajata jotkin sivun osat toisistaan. Tämä toteutetaan yleensä div-elementillä, koska sen sisään voidaan ryhmitellä useita kappaleen elementtejä. Pääasiassa div-elementillä ohjataan tyylilomakkeiden arvojen kohdistumista HTML-dokumentissa. (Linjama 2001: 202.) Nykysuuntauksen mukaan div-elementillä pyritään korvaamaan raskaat ja kankeat taulukkoelementit sivupohjan taitossa. Alla olevassa esimerkissä (kuva 9) kuvataan kahden lohko-elementin tyylimääritys sekä käyttö (X)HTML-dokumentissa.

```
.leftcontent {
    width:250px;
    float:left;
}

#rightcontent {
    width:500px;
    float:right;
}

<div class="leftcontent">
    <a href="linkki">Linkki</a>
</div>

<div id="rightcontent">
    <h1>Otsikkoteksti</h1>
</div>
```

Kuva 9. Esimerkki lohko-elementtien käytöstä.

2.3.3 Pseudoluokat

Näennäis- eli pseudoluokat ovat tyyli luokkien erikoistapauksia. Näille luokille voidaan antaa ominaisuuksia, joita ei ole mahdollista toteuttaa normaalilla HTML-merkkauksella. Esimerkiksi linkin värin vaihtaminen aktiivisen, vierailun ja vierailemattoman linkkien välillä onnistuu pseudo-luokkien avulla. Tyylimäärittelyissä tyylin nimi saa näennäisluokan nimen liitteeseen, joka erotetaan tyylin nimestä kaksoispisteellä. Esimerkissä (kuva 10) määritetään värit vierailemattomalle, vierailulle ja aktiiviselle linkille.

```
a:link {
    color:blue;
}
a:visited {
    color:purple;
}
a:active {
    color:red;
}
```

Kuva 10. Esimerkki näennäisluokkien käytöstä.

2.4 PHP

PHP on yleisin palvelinsovellusten ohjelmointikieli. PHP perustuu avoimeen lähdekoodiin, mikä takaa sen, että ohjelmointikieli kehittyy jatkuvasti. PHP sisältää kaikki ohjelmointikielelle tyypilliset rakenteet, kuten silmukat, ehtolauseet, muuttujat ja funktiot. PHP mahdollistaa oliopohjaisen ohjelmoinnin määrittelemällä luokkia, ja se sisältää valmiit funktiot mm. tietokantojen, XML-tiedostojen ja sähköpostin käsittelyyn. (Keränen ym. 2006: 35.) PHP-kieltä käytetään usein luomaan dynaamista sisältöä WWW-sivuille yhdessä (X)HTML:n kanssa. Ohjelmakoodi kirjoitetaan (X)HTML-dokumenttiin, josta se tulkitaan palvelimen kautta staattiseksi HTML-koodiksi selaimen. PHP-kielillä tehdyn WWW-sivun tunnistaa php-pääteestä.

PHP:ssä on lukematon määrä erilaisia funktioita. Käytetyimpiä ovat *include-* ja *require-*funktioita. Niiden avulla voidaan toteuttaa mm. sivuston toistuvia osia, kuten navigaatioita. Ulkoiseen PHP-dokumenttiin tallennetaan toistuva osa, jota sivustolta kutsutaan esimerkiksi kuvan 11 osoittamalla tavalla.

```
<?php include_once("navigointi.php"); ?>
```

Kuva 11. Ulkoisen PHP-dokumentin kutsu sivustolta.

2.5 JavaScript

JavaScript on internetin suosituin ohjelmointikieli. Sitä käytetään yleisesti luomaan dynaamisia toiminnallisuuksia käyttöliittymään, joita ei voida tavallisella julkaisukielellä tehdä. Tällaisia toimintoja voivat olla esimerkiksi navigointien alavetovalikot, lomakekenttien kutsut tai selainikkunan avaaminen.

JavaScript-koodia ei tulisi käyttää WWW-sivuilla toimintoihin, jotka ovat sivuston käytön kannalta oleellisia. JavaScript voidaan ottaa pois käytöstä ja siksi sitä pitäisi käyttää esimerkiksi lomakkeiden painikkeiden täydentämiseen, ei kokonaan korvaamiseen. (Korpela ja Linjama 2004: 116.)

2.6 Validointi

HTML-kielelle on määritelty tarkka kielioppi, jota kaikkien WWW-dokumenttien tulisi noudattaa. Ihmiselle virheellisen kielen tulkitseminen on helppoa, mutta selainohjelmat ovat HTML-kieliopin suhteen kriittisempiä. Selaimet yrittävät parhaansa mukaan suodattaa virheitä ja useimmat selaimet pystyvät käytännössä näyttämään koko dokumentin niistä huolimatta.

W3C on laatinut ohjeistuksen, joka tunnetaan nimellä WAI. Se määrittelee teknisiä sekä sisällöllisiä toteutustapoja, joita noudattamalla sivujen esteettömyyttä voidaan parantaa. WAI:n lähtökohtana on, että verkkopalvelun tulee olla kaikkien käytettävissä riippumatta käytetystä päätelaitteesta, selaimesta tai itse käyttäjän eritysominaisuuksista. (Keränen ym. 2006: 16.)

WWW-dokumentin pitää olla validi eli oikeellinen, jotta se olisi käytökelpoinen. Verkkosivujen esteettömyys voidaan testata koneellisesti validaattorilla, joka vertaa sivujen teknistä toteutusta WAI-suosituksen mukaiseen toteutustapaan. Validoinnin tuloksena saadaan raportti sivujen esteettömyyteen liittyvistä ongelmista ja ehdotuksista niiden korjaamiseksi. Mikään validaattori ei kuitenkaan korvaa käyttäjien tekemää testausta. Siksi eri kohderyhmiä tarvitaan lopulliseen testaamiseen. (Keränen ym. 2006: 16.)

3 Käytettävyys WWW-suunnittelussa

”Käytettävyys on ominaisuus, joka ilmentää sitä, miten järjestelmä, laite, ohjelma tai palvelu soveltuu suunniteltuun tarkoitukseen tietyille kohderyhmälle.” (Wikipedia)

Nielsenin (2000: 380) mielestä www-sivustojen pitäisi tehdä käyttäjensä tavoitteiden toteuttamisesta mahdollisimman yksinkertaista. Sivustolla saa ja pitääkin olla muitakin ominaisuuksia ja toiminnallisuuksia, mutta yksinkertaiset toimenpiteet pitäisi pystyä suorittamaan mahdollisimman helposti.

3.1 Rakenne ja navigointi

Internetin käytön perustana on liikkuminen eli navigointi. Hyperlinkkiä napsauttamalla käyttäjä kokee vuorovaikutusta selaimen kanssa liikkeessaan valtavassa tietoverkossa. Navigointikäyttöliittymän avulla käyttäjän pitäisi pystyä tiedostamaan missä hän tällä hetkellä on, mistä hän tuli ja minne hän voi sivustolla vielä mennä. (Nielsen 2000: 188.)

Nielsenin (2000: 190) mukaan todennäköisesti tärkeintä käyttäjälle on, että hän ymmärtää missä hän tällä hetkellä on. Käyttäjän sijainti pitää osoittaa kahdella eri tasolla: koko internetin mittakaavassa sekä sivustokohtaisesti. Sivusto on pystyttävä tunnistamaan jokaiselta sen sivulta ja niiden tulee olla yhtäläisiä, sillä käyttäjät voivat soveltaa yhdellä sivustolla oppimaansa kaikilla sivuston sivuilla.

Käyttäjä ei kuitenkaan tiedä millä sivulla hän on, ellei sitä kerrota hänelle. Liikkumisen kannalta tärkeintä onkin että sivun omistajan logo (tai jokin muu tunnus) näkyy sivuston joka sivulla. Logo on tarkoitus sijoittaa sivustolla aina samaan kohtaan, useimmiten sivun vasempaan yläkulmaan. (Nielsen 2000: 191.)

Käyttäjän sijainti sivustolla osoitetaan yleensä näyttämällä osia sivuston rakenteesta ja korostamalla se sivu, missä käyttäjä sillä hetkellä on. Tämän lisäksi sivulla olisi hyvä olla lyhyt, sisältöä kuvaava tai sivun nimen sisältävä otsikko. (Nielsen 2000: 191.)

Sivuston kehittäjät eivät suoraan pysty vaikuttamaan siihen tiedostaako käyttäjä sivut missä hän on jo ollut. Selainohjelmien takaisinpainikkeen ja sivuhistorian lisäksi hypertekstilinkit ovat HTML:n perusmääritysten mukaisesti eri värisiä, mikäli ne osoittavat sivuille joilla käyttäjä on jo käynyt. (Nielsen 2000: 191.)

Hypertekstilinkit rakennetaan nykysuuntauksen mukaan muistuttamaan sivuston muuta värimaailmaa standardiväriytyksen sijasta. Linkit erotetaan muusta tekstistä alleviivaamalla ja/tai muuttamalla linkin väriä ja kirjasimen lihavoitua.

Nielsenin (2000: 198) mukaan käyttöliittymän navigointimallista riippumatta web-ympäristössä liikkumiseen pätee aina samat säännöt: pääperiaatteena on osoittaa käyttäjälle hänen sen hetkinen sijaintinsa ja mahdolliset liikkumisvaihtoehdot. Mikäli rakenne sivustolla on epäselvä, ei edes järkevästi toteutetusta navigoinnista ole käyttäjälle hyötyä. Huonosti suunnitellun informaatioarkkitehtuurin seurauksena myös käytettävyys kärsii.

3.2 Tekstin luettavuus

Tekstin lukeminen näyttölaitteelta on vaivalloisempaa kuin paperilta. Internetissä käytettävällä leipätekstillä on erilaisia vaatimuksia kuin painetulla, esimerkiksi kirjasinvalinnalla on merkitystä. Painetussa materiaalissa esiintyviä pitkiä kappaleita ei web-sivuilla saisi esiintyä, vaan leipätekstin ja otsikoiden pitäisi olla selkeitä ja lyhyitä. Myös kirjasimen ulkoasulla on myös merkitystä, sillä teksti on sivuston ydinsisältöä ja tärkeä osa sivuston graafista ilmettä. (Korpela ja Linjama, 2004: 184-185.)

Sivustojen luettavuudesta voidaan Nielsenin (2000: 125-126) mukaan varmistua, jos suunnittelussa noudatetaan muutamaa perussääntöä:

- Tekstin ja taustan värillä on oltava mahdollisimman suuri kontrasti. Luettavuus on parhaimmillaan jos teksti on mustaa ja tausta valkoista.
- Tekstin taustaksi kannattaa valita joko yksi ainoa väri tai todella rauhallinen taustakuvio. Taustakuvio häiritsee silmän kykyä erottaa kirjainten viivoja ja tunnistaa sanojen muotoja.
- Käytettävän kirjasinkoon pitää olla niin suuri, että myös ihmiset joilla ei ole täydellinen näkökyky, pystyvät lukemaan tekstin.
- Tekstin on pysyttävä paikallaan. Liikkuvaa, välkkyvää tai zoomailevaa tekstiä on huomattavasti vaikeampi lukea kuin paikallaan pysytteleviä sanoja.

4 Lähtötilanne

mCompass-tuoteperheen sovelluksilla ei ole aikaisemmin ollut valmista sivupohjaa, minkä ympärille varsinaisen sovelluksen olisi voinut rakentaa. Usein edellisestä sovelluksesta on kopioitu sivupohjan käytettävimmät ominaisuudet tai joissain tapauksissa koko pohja. Menetelmä on ollut hidasta, ja koodin suora kopioiminen on saattanut aiheuttaa käyttöliittymään toiminnallisia ongelmia.

Tuoteperheen sovellusten alkuperäisissä versioissa ei ole esiintynyt Mediamaisteri Groupin omaa ulkoasun tuomaa leimaa lainkaan. Sivupohjien kopioimisesta on aiheutunut ulkoasun puuroutumisen ja teknisten ongelmien lisäksi myös graafisen identiteetin katoamista. Sovellukset olisivat periaatteessa voineet olla kenen tahansa luomia, eikä asiakas ole tunnistanut sovelluksia Mediamaisterin tekemäksi ulkoasun perusteella. Kuvassa 12 havainnollistetaan tuoteperheen käyttöliittymän ulkoasua Säästöpankille toteutetun Osaamispassi-sovelluksen avulla.



[Poistu]

Sisältö

- Etusivu
- Henkilö- ja yhteystiedot
- Nykyinen työtehtävä
- Työhistoria
- Peruskoulutus
- Koulutus- ja kehityshistoria
- Kielitaito
- Työhön liittyvä osaaminen
- Harrastukset ja vapaa-ajan toiminta
- Kehityskeskustelu

TERVETULOA!

Avasit juuri mCompass-osaamispassin, jonka tarkoituksena on toimia työvälineenä osaamisen johtamisessa ja kehittämisessä. **Huolehdi itse siitä, että osaamispassisi on aina ajan tasalla.**

Suositus on, että esimiehesi tarkastaa osaamispassin ajantasaisuuden ennen kehityskeskusteluita. Esimiehesi näkee osaamispassisi sisällön ja pääsee kommentoimaan sen sisältöä. Kehityskeskustelussa käydään läpi erityisesti työhön liittyvä osaaminen, missä esimiehesi on kanssasi samaa mieltä ja missä eri mieltä. Kehityskeskustelun jälkeen työntekijä käy päivittämässä osaamispassinsa sisällön.

Pankki voi määritellä myös omia käytäntöjä mCompass-osaamispassin käyttöön liittyen.

Osaamispassi tallentaa automaattisesti syöttämäsi tiedot. Poistu aina osaamispassista klikkaamalla oikealla yläkulmassa olevaa Poistu-linkkiä. Voit aloittaa passin täyttämisen valitsemaltasi välilehdeltä.

Voit nyt aloittaa osaamispassisi täydentämisen Henkilö- ja yhteystiedot -välilehdeltä tai päivittää haluamaasi välilehteä.

Ongelmatilanteissa voi ottaa yhteyttä sähköpostitse: mcompasstuki@mediamaisteri.com.

Kuva 12. mCompass-tuoteperheen Osaamispassi-sovelluksen etusivu 4.4.2008.

Visuaalisista ja toteutuksellisista puutteista huolimatta sivupohjat on rakennettu tämänhetkisten standardien mukaisesti. Ulkoasu on toteutettu HTML-CSS-yhdistelmällä ja sovelluksen dynaamiset elementit on ratkaistu PHP-kielen avulla. PHP-kieli on mahdollistanut mm. lomakekenttien kutsut, tallentamiset, poistamiset ja muokkaamiset, ja sen avulla sivuille on voitu luoda toistuvia osioita, kuten navigaatiot sekä ylä- ja alaotsake. Sivupohjien ulkoasuun liittyvät asiat, kuten lohkojen sijainti ja taustakuvien käyttö on määritelty tyyli tiedostoilla.

4.1 Käytettävyyden ongelmat

Käytettävyyden miettiminen yhdessä ulkoasusuunnittelun kanssa on aikaisemmissa sivupohjissa jäänyt vähälle huomiolle. Käyttäjät ovat nostaneet käytettävyyden kannalta suurimmiksi ongelmakohdiksi moduulitilojen välisen navigoinnin puutteen sekä lomake- ja taulukko-kenttien rakenteen kehnot ratkaisut. Navigoinnin puuttumisen takia käyttäjä on joutunut poistumaan väliaikaisesti sovelluksesta päästäkseen toiseen näkymään, mikä on vienyt aikaa ja mielenkiintoa itse lomakkeiden täyttämislä.

Lomakkeet ovat olleet massiivisia, mutta kenttiä ei ole eroteltu visuaalisilla rajaajilla toisistaan, mikä tämän mittaluokan kaaviossa on välttämätöntä. Kenttiä ei ole myöskään ryhmitelty pienempiin kokonaisuuksiin, mikä osaltaan hankaloittaa lomakkeen lukemista. Taulukoista ja lomakkeista on puuttunut looginen rakenne sekä yhtenäinen ulkoasu, mikä on vaikeuttanut niiden hahmottamista ja tulkintaa.

4.2 Tekniset ongelmat

Aikaisemmin mCompass-tuoteperheen sovellukset on rakennettu kii-
reisellä aikataululla ja sivupohjan HTML-koodi ja tyyli tiedostot on kopioitu usein suoraan aikaisemmista sovelluksista. Tämä on aiheut-
tanut päänvaivaa jatkokehittäjille, sillä koodi on muuttunut vaikeasti
tulkittavaksi ja tyylimääritykset ovat olleet epäjohdonmukaisia.
HTML-rakenne on myös ollut sekava, mikä on osaltaan hidastanut
sovellusten kehittämisprosessia. Toistuvat osat on tosin tuotu *include*-
komennoilla sovellukseen, mikä on selkeyttänyt osaltaan rakennetta.
Ongelma on ollut siinä, miten sovelluksen varsinainen HTML-raken-
nepuu on toteutettu. Elementtien sijoittelua HTML-rakenteessa ei ole
ratkaistu järkevästi eikä käytettävyyttä ole mietitty sovelluksen to-
teuttajan kannalta. Lisäksi PHP-kutsuja on esiintynyt sivukohtaisesti
liikaa. Käytännössä sovellus olisi tarvinnut kutsut ainoastaan sivun
ylä- ja alaotsakkeille (header ja footer).

Tiedostojen sijoittelua ja tyyli-tiedostojen jakoa ei ole suunniteltu kunnolla ja ratkaisut vaihtelevat tässä sovelluksittain. Tämä on vaikeuttanut sovelluksen ylläpitämistä. Sovelluksen rakennusprosessissa ei ole ollut aikaa paneutua koodin sijoitteluun, vaan jokainen sovellus on parsittu kasaan edellisten sivupohjien osista. Työ on tehty alusta lähtien uudestaan sen sijaan, että käytössä olisi ollut valmis helposti muokattava sivupohja jota pienillä muutoksilla olisi voitu käyttää kaikissa tulevilla sovelluksissa. Sivupohjassa esiin tulevien ongelmien selvittäminen on ollut hankalaa, sillä myöskään CSS-koodi ei aina ole ollut loogisesti rakennettu, vaan uusi määrittely on useimmiten sijoitettu tiedoston loppuun viimeiselle riville. Tyylimäärittelyitä ei ole myöskään kommentoitu, mikä on aiheuttanut päällekkäisyyksien syntymistä koodissa.

4.3 Vanha ulkoasu

Aikaisempien sivupohjien ulkoasut eivät ole vastanneet värimaailmaltaan ja yleisilmeiltään Mediamasteri Groupin omaa graafista linjaa. Sovelluksia rakennettaessa ulkoasuun ei aina ole ehditty kiinnittämään tarvittavaa huomiota, sillä päätavoite on ollut ydinsisällön lo- make- ja taulukkotietoineen saattaminen toimintakuntoon mahdollisimman nopeasti. Tilattujen pakettien värimaailmaa on kuitenkin räätälöity asiakkaan toiveiden mukaiseksi, mikä on entisestään hävittänyt Mediamasteri Groupin graafista identiteettiä.

Graafinen ilme sivupohjissa on henkilökunnan mielestä ollut vanhanaikainen ja se on kaivannut sekä päivitystä että pysyvyyttä. Ulkoasuun on kaivattu myös markkinointihenkisyyttä, sillä tuotteita esitellään messuilla sekä muissa tapahtumissa, ja henkilökunta kokee nykyaikaisemman ulkoasun vetoavan paremmin potentiaalisiin ostajiin.

Tuotteet eivät ole olleet myöskään typografisesti linjassa Mediamasteri Groupin graafisen ohjeistuksen ja painettujen julkaisuiden kanssa. Web-maailmassa linjausten noudattaminen ei ole kaikilta osin edes mahdollista, sillä käyttöjärjestelmien ja selainten kirjasintuki erikoiskirjasimille on heikko ja vaihteleva. Yleisesti leipätekstissä ja otsikoissa käytetyillä kirjasintyyyleillä on kuitenkin kaikkien selainohjelmien tuki, ja erikoisfontit voi tarvittaessa korvata bittikarttagrafiikalla.

4.4 Lomakkeet

Lomakkeet ovat Mediamaisteri Groupin sähköiseen oppimiseen tarkoitettujen sovelluksien ydinsisältö. Visuaalisesti kunnossa olevalta sovellukselta katoaa pohja, mikäli lomakejärjestelmää ei ole toteutettu riittävän hyvin. Lomakkeet ovat sovelluksien olleet päätiedonlähde sekä loppukäyttäjille että sovelluksen ylläpitäjälle.

Mediamaisteri Group on aikaisemmin toteuttanut myös lomakkeiden määritykset kopioimalla ne aikaisemmista sovelluksista, mikä on johtanut niin rakenteelliseen kuin visuaaliseen epäloogisuuteen. Tämä on saattanut aiheuttaa myös turhautumista loppukäyttäjien osalta, sillä he ovat joutuneet käyttämään ylimääräistä aikaa sovellusten hahmottamiseen, mikä on puolestaan syönyt mielenkiintoa lomakekenttien täyttämiseltä.

5 Uuden sivupohjan mahdollisuudet ja rajoitukset

Pyrittäessä saavutettavuudeltaan ja käytettävyydeltään mahdollisimman hyvään lopputulokseen piti kartoittaa uuden sivupohjan mahdolliset rajoituksia aiheuttavat tekijät, kuten tuotteen kohderyhmä sekä kohdelaitteisto. Kohderyhmän määrittely auttoi käyttöliittymän suunnittelussa erityisesti visuaalisten elementtien osalta sekä antoi osviittaa ohjeistuksen ja johdattelun tarpeellisuudesta.

Kohderyhmän määrittely perustui tässä työssä Mediamasteri Groupin asiakkailta saamaan palautteeseen. Palautteesta selvisi, että loppukäyttäjien taitotaso ja ikäjakauma oli todella laaja. Kohderyhmä koostui täysi-ikäisyyden kynnyksellä olevista aina eläkeikään ehtineistä henkilöistä. Suurin käyttäjäryhmä koostui n. 40–50-vuotiaista henkilöistä. Kohdehenkilöiden atk-taitotaso vaihteli noviisista asiantuntijaan. Näillä tiedoilla voitiin päätellä, että sovelluksen ulkoasu ja käyttöliittymä piti rakentaa ohjeistukseltaan ja käytettävyydeltään palvelemaan kaiken ikäisiä ja tietokoneenkäyttötaidoiltaan eroavia ihmisiä, eikä tiettyä kohderyhmää voitu määrittää.

Tuotetta käyttävien henkilöiden kartoituksen lisäksi piti määrittää ne laitteet ja ohjelmat, joille sivupohja optimoidaan. Kohderyhmän käyttämä laitteisto koostui käytännössä kokonaan PC-järjestelmistä. Tästä syystä Mac-laitteiden tuki jätettiin projektista kokonaan pois. Sivupohja optimoitiin Windows-käyttöjärjestelmien neljälle yleisimmälle internet-selaimelle: Internet Explorer 5.5, Internet Explorer 6.0, Internet Explorer 7.0 ja Mozilla Firefox 2.

Sovelluksen piti palvella ensisijaisesti loppukäyttäjää, mutta myös ulkoasun markkinointihenkisyys oli kehittyessä isossa roolissa. Alan messut ja seminaaritulaisuudet ovat tärkeitä kanavia tuotteiden markkinoinnille, ja hyvän ensivaikutelman luominen on ensisijaisen tärkeää. mCompass-tuoteperheen tuotteen ostopäätöksestä vastaa useimmiten esimiestason henkilö (kuten koulutuspäällikkö, koulutussuunnittelija tai henkilöstöpäällikkö). Iältään henkilö on yleensä yli neljäkymmentävuotias, sukupuolijakauman mennessä kutakuinkin tasan. Ottaen huomioon ostopäätöksestä vastaavan kohdehenkilön koulutuksen, on hänen tietotasonsa verkko-oppimisesta todella laaja: tilaaja saattaa tietää tuotteista ja verkkoympäristöstä tapahtuvasta oppimisesta paljon tai on saattanut olla jopa suunnittelemassa sovellusta. Toinen ääripää suorastaan pelkää verkko-oppimista sekä -työskentelyä. Näinkin hajutetun jakauman kohdatessa täytyi tyytyä kompromisseihin, jotta ulkoasu miellyttäisi ja palvelisi mahdollisimman suurta joukkoa.

6 Toteutukseen liittyviä asioita

6.1 Rakenteellisten osien jakaminen tiedostoihin

Koska kyseessä on erittäin laaja dynaaminen sovellus, täytyi sovelluksen sisäiset toiminnot jakaa eri tiedostoihin. Sivustolla oli paljon toistuvia osia kuten ylä- ja alatunnisteet, navigoinnit ja tyylimäärittelyt, jotka ovat sivuille yhtenäisiä. Näiden osioiden koodia ei ollut järkevää liittää jokaiselle sivulle erikseen, vaan eri osiot kannatti sijoitella omiin tiedostoihinsa joita kutsutaan pääsivulta. Jos esimerkiksi tyylimäärittelyihin tai pelkästään ylä- ja alatunnisteeseen haluttiin tehdä muutoksia, muutoksen tekeminen kymmenille sivuille erikseen olisi ollut todella työlästä. Kun ylä- ja alatunnisteen koodi liitettiin yhdestä tiedostosta joka sivulle, oli muutosten tekeminen helpompaa. Tämä tekee sovelluksen päivittämisestä ja ylläpitämisestä jatkossa huomattavasti kustannustehokkaampaa, toimintavarmempaa ja helpompaa.

6.2 Tiedostojen sijoittelu

Sovelluksen tiedostot sijoitettiin Mediamasteri Groupin testipalvelimelle sekä asiakkaan omalle palvelimelle yksinkertaisella tiedonsiirto-ohjelmalla, joten on tärkeää, että tiedostot ja kansiot ovat loogisessa järjestyksessä. Kuvien, tyyli- ja javascript-tiedostojen, JavaScript- ja muiden ohjelmointikielillä toteutettavien tiedostojen sekä kirjaston ohjaustiedostot ovat omissa kansioissaan. Kansiot nimettiin sisältönsä mukaan. Tämä käytäntö helpottaa tiedostojen päivitettävyyttä ja sovelluksen kokonaisuuden hahmottamista. Tyyli- ja javascript-tiedostot jaettiin kahteen osaan: päätyyli- ja javascript-tiedosto, joka sisältää globaalit tyyli- ja javascript-asetukset sekä ohjelmakohtainen tyyli- ja javascript-tiedosto, johon laitetaan sovelluksittain vaihtuvat tyyli- ja javascript-asetukset.

6.3 Testaaminen

Kun uusi sivupohja integroitiin varsinaisen sovelluksen kanssa, todennettiin sivupohjan käytettävyyttä ja toimivuutta Mediamasteri Groupin henkilökunnan toimesta. He antoivat palautetta sovelluksen toimivuudesta sekä käytettävyydestä sähköpostin muodossa tai suullisesti palavereissa. Sivustoa testattiin erilaisilla näyttölaitteilla, joissa eri resoluutioilla, kirkkaussuhteilla ja värisävyasetuksilla todennettiin sivupohjan näkyvyys eri tiloissa. Resoluutiota vaihtamalla havainnoitiin sovelluksen sopivuutta erikokoisiin selainikkunoihin ja värimuutoksilla varmistettiin objektien ja tekstin näkyvyys. Internet-selaimissa tapahtuvaa sivuttaisvieritysefektiä pyrittiin välttämään, jotta sovelluk-

sen käytettävyys ei kärsisi. (X)HTML- ja CSS-dokumenttien oikeellisuus tarkistettiin koneellisesti W3C:n internetsivuilta (<http://www.w3.org>) löytyvillä validaattoreilla.

7 Uusi sivupohja

Sivupohjan rakenne uusittiin kokonaan. Sivujen käytettävyyttä parannettiin lisäämällä sivupohjaan toinen navigaatio, jotta eri tilojen välillä voi liikkua poistumatta sovelluksesta. Sovelluksen ulkoasu päivitettiin nykyaikaisemmaksi ja markkinointihenkisemmäksi. Vanhoista toiminnoista sivupohjassa pysyi sivunavigaatio. Sovellukseen lisättiin toiminto, jolla navigaatiot ja muut toiminnallisuudet saadaan helposti poistettua sovelluksesta. Tämä helpottaa jatkossa huomattavasti sovelluksen rakentamisprosessia. Toimintojen poistot ja lisäykset automatisoitiin, joten jatkossa koodimuutokset voidaan tehdä yksinkertaisesti PHP-kutsuihin hankalien ja aikaa vievien HTML-elementtien muokkauksien sijasta.

Uudet toiminnallisuudet mahdollistavat mm. sivupohjan koko leveyden käyttöönoton leveille taulukoille ja lomakkeille. Sivupohjaan lisättiin tila loppukäyttäjää palveleville sovelluksen sisäisille toimintoille, kuten tulosta- ja poistu sovelluksesta -painikkeille sekä tekstin koon vaihdolle. Valmis sivupohja (kuva 13) integroitiin lopuksi Suomen Sosiaali- ja terveystieteiden ministeriön Osaamispassi-sovellukseen, jolla sitä testattiin käytännössä. Testauksen jälkeen sivupohja tullaan integroimaan myös kaikkiin tuleviin mCompass-tuoteperheen sovelluksiin.

The screenshot shows the Osaamispassi web application interface. At the top, there are navigation tabs: OSAAMISPASSI, ESIMIESNÄKYMÄ, and RAPORTTI. The main header features the Osaamispassi logo and a POISTU button. The page content is divided into a left sidebar and a main content area. The sidebar contains a 'Henkilötiedot' section with a list of evaluation items, including '1 Työsuojeluhallinto osana julkista hallintoa', '2 Hallinto- ja tukitoiminnot', '3 Työsuojeluväily', '4 Työelämän ja sidosryhmien tuntemus', '5 Vuorovaikutustaidot', and '6 Osaaminen suhteessa työtehtäviin'. The main content area is titled 'Henkilötiedot' and includes instructions: 'Ohjeet Osaamispassin täyttämiseen ja osaamiskartoitukseen' and 'Aloita täyttämällä henkilötiedot.' Below this, there are three sections for data entry: 'Nimitiedot' with fields for 'Sukunimi:', 'Etunimet:', and 'Kutsunanimi:'. 'Syntymäaika' with a date selector for 'pv', 'kk', and 'vuosi'. 'Yhteydenotto' with fields for 'Puhelinnumero (työ):' and 'Sähköposti (työ):'. A 'TALLENNNA' button is located at the bottom of the form.

Kuva 13. Uusi sivupohja integroituna Suomen Sosiaali- ja terveystieteiden ministeriölle kehitettyyn Osaamispassi-sovellukseen.

7.1 Sivupohjan uudistukset

Uutta ulkoasua suunniteltaessa ensimmäinen tehtävä asia oli käyttäjäkunnan ja laitteiston rajaaminen. Sovellus optimoitiin 1024x768-resoluutiolle, mutta se suosii myös pienemmän näyttölaitteiden omistajia, aina resoluutioon 800x600 saakka. Sivut, joissa on sisältönä lomakkeita tai taulukkorakenteita, vievät paljon tilaa pystysuunnassa. Tästä syystä lähes jokaisella välilehdellä selaimen oikeaan reunaan ilmestyy sovellukselta tilaa vievä vierityspalkki. Jotta sivupohjasta ei jäisi osia selainikkunan ulkopuolelle edes 800x600-resoluutiolla, määriteltiin sisällön leveydeksi 762 pikseliä. Sovelluksen molempia reunoja sekä alareunaa kehystävät varjoelementit joiden tehtävä on visuaalisesti havainnollistaa sovelluksen sisällön rajat.

Taittoon tuotiin uutena elementtinä erilaisia toimintoja sisältävä otsake-elementti (kuva 14, header), joka sijaitsee sivupohjan yläosassa. Otsake pitää sisällään ensisijaisen navigaation (primaryNav), tilan sovelluksen nimelle tai asiakkaan logolle (companyLogo) sekä työkalulaatikon sovelluksen sisäisille toiminnallisuuksille (tools). Ensisijainen navigaatio sijaitsee taiton ylimpänä ja se on tarpeen mukaan mahdollistaa poistaa sovelluksesta. Ylänavigaation alapuolella on tila logolle ja työkalulaatikon. Asiakkaan logo tai sovelluksen nimi sijaitsee ylänavigaation alla sivun vasemmassa reunassa ja se liitetään taittoon bittikarttakuvana. Tämä mahdollistaa myös niiden kirjasintyylien käytön, joita selainohjelmat eivät yleisesti tue. Tilaa sovelluksen työkaluille on aina logosta sivun oikeaan reunaan asti. Työkalujen lukumäärää ei ole rajoitettu, mutta ne eivät saa ylittää niille annettua vaakasuuntaista tilaa. Myös työkalut voidaan tarvittaessa poistaa kokonaan.

Otsakkeen taustalla on liukuvärillä toteutettu Mediamaisteri Groupin tummansininen taustakuva, joka tuo sovelluksen ulkoasuun kaivattua nykyaikaisuutta sekä samalla havainnollistaa ja rajaa sivupohjan rakennetta: kuva erottaa ylänavigaation sisällöstä ja luo visuaalisen kytköksen työkalujen sekä ydinsisällön välille. Ylänavigaatio on erotettu visuaalisesti varsinaisesta ydinsisällöstä, jotta käyttäjä ei sekoittaisi sitä välilehtikohtaiseen navigointiin. Ylänavigaation ja muun sivupohjan välille luodaan kuitenkin graafinen kytkös aktiivisena olevan linkin kuvatehosteella (sininen nuoli linkin alapuolella).



Kuva 14. Yläotsake (header). Ylimpänä ensisijainen navigointi, jonka alapuolella sovelluksen logo. Oikeassa laidassa työkalulaatikko.

Sovelluksen sisältö voi olla yksi- tai kaksipalstainen. Palstojen määrä määritellään sovellusta kehystävän div-elementin (wrapper) luokan nimellä. Yksipalstaisessa versiossa otsakkeen jälkeinen tila on varattu koko leveydeltään sisällölle. Yksipalstaista versiota käytetään paljon vaakasuuntaista tilaa vaativien lomakkeiden ja taulukoiden kanssa. Kaksipalstaisessa versiossa sisältö jaetaan pystysuunnassa kahteen osaan. Vasemmanpuoleinen tila vie noin yhden kolmasosan sisällön leveydestä ja sillä on harmaa pohjaväri. Tähän tilaan sijoitetaan mm. toissijainen eli sivunavigaatio. Oikealla puolella oleva, leveyssuunnassa noin kaksi kolmasosaa vievä tila on varattu sovelluksen ydin-sisällölle, kuten lomakkeille, tekstile ja taulukoille.

7.2 Visuaalinen ilme

Mediamaisteri Groupilla on käytössään graafinen ohjeistus (liite 1). Aikaisemmin ohjeistusta on käytetty lähinnä painetussa materiaalissa sekä muutamissa sähköisissä ratkaisuissa. Ohjeistus asettaa tarkat määreet ainoastaan painettavalle materiaalille, mutta WWW-sovellusten osalta sitä voidaan noudattaa tarkasti ainoastaan leipätekstin kirjasimen osalta. Muun sähköisen materiaalin, kuten värisuunnittelun osalta määrittelyjä ei ole tehty, joten ohjeistuksessa käytettävät värit ja muut elementit sisällytettiin sivupohjan ilmeeseen silmämääräisesti tai mittaamalla pipetillä RGB-väriarvot.

Vanhojen sovelluksien ideoista käytettiin graafisen suunnittelijan Mediamaisterille luomaa meriteemaa, joka on toistunut mCompass-tuoteperheen sovelluksissa ja vakiinnuttanut paikkansa osana tuotteiden ilmettä. Tuoteperheessä esiintyvä meriteema näkyy sivupohjassa häivyttynä ruorina toissijaisen navigaation alla.

7.3 Ensisijainen navigaatio

Sovelluksessa esiintyy samaan aikaan enintään kaksi navigaatiota: ensisijainen (kuva 15) ja toissijainen navigaatio (kuva 16). Ensisijainen, eli ylänavigaatio mahdollistaa liikkumisen moduulin eri tilojen välillä. Ylänavigaatio on taitossa ollessaan ylin elementti. Navigaation linkit rakennettiin muistuttamaan painikkeita, jotka erotettiin toisistaan reunaviivoilla. Painikkeiden määrää ei ole rajoitettu, mutta linkit eivät saa ylittää vaakasuunnassa otsake-elementin kehyksin määriteltyä aluetta.

Ensisijainen navigaatio toteutettiin hyväksikäyttäen PHP- ja XHTML-kieliä sekä tyylimäärittelyitä. Aktiivinen linkki määritetään sivukohdaisesti PHP:llä. Sovelluksen toiminnallisuus automatisoitiin niin, että ylänavigaation PHP-kutsussa voidaan linkin aktiivisuus jatkossa vaih-

taa ainoastaan yhden merkin muutoksella. Tämä nopeuttaa sovelluksen välilehtikohtaista toteuttamista sekä selkeyttää sivun rakennetta ja koodia. Navigaatio näkyy eri asemilla oleville henkilöille eri tavalla. Esimerkiksi esimiehille tulee näkyviin esimiesnäkyvä, josta he pääsevät seuraamaan ja kommentoimaan alaistensa lomakkeentäyttöprosessia ja tuloksia. Toimihenkilöiltä ylänavigointi on puolestaan poistettu kokonaan. Navigaation poisto ja lisäys tapahtuvat vaivattomalla muutoksella PHP-kutsuun.



Kuva 15. Ensisijainen navigaatio, kohta ”raportointi” aktiivisena.

7.4 Toissijainen navigaatio

Toissijainen navigaatio eli sivunavigaatio sijaitsee nimensä mukaisesti sovelluksen sivussa, sisällön vasemmalla puolella. Linkit on sijoitettu allekkain ja tasattu vasemmalle. Sivunavigaation alla on pystysuunnassa koko sisällön mittainen harmaa taustaväri, joka jakaa sisällön visuaalisesti kahteen osaan: sivunavigaatioon ja ydinsisältöön. Linkkien väri on perustilassa vaalean sininen. Aktiivisena linkki vaihtuu tummansiniseksi. Sivunavigaatioiden linkkien aktiivisuus toteutettiin JavaScript-kielellä. Selainpohjaisen ohjelmointikielen käyttö voi tiettyissä tapauksissa aiheuttaa saavutettavuusongelmia, mutta Mediamaisteri Groupin vanhoista sovelluksista saadun palautteen perusteella loppukäyttäjien selaimissa JavaScript on täysin tuettu kieli.



Kuva 16. Toissijainen navigaatio (vasemmalla). Aktiivisena linkkinä ”Henkilötiedot”.

7.5 Lomakkeet

Käyttäjät täyttävät lomakkeisiin perustietonsa, työ- ja koulutushistoriansa sekä oman osaamisensa arvioinnin. Lomakkeet rakennettiin sivupohjan tapaan kokonaan alusta palvelemaan paremmin jatkokehittäjien tarpeita.

Lomake-elementtien ulkoasuun voidaan tyylimäärittelyillä vaikuttaa vain rajallisesti. Asetettujen tyylimääreiden vaikutus vaihtelee suuresti selainvalmistajien ja jopa selainversioiden mukaan. Jotta lomakkeiden ulkoasu saataisiin täsmälleen samannäköiseksi selaimesta riippumatta, jouduttaisiin käyttämään selainpohjaista ohjelmointia, kuten JavaScript-koodausta. Tämä kuitenkin kuormittaisi jo valmiiksi raskasta sovellusta, joten lomakkeiden ulkoasu päätettiin toteuttaa käyttämällä pelkästään staattisia tyylimäärittelyjä.

Lomakkeet koostuvat kentistä ja niiden otsikoista, jotka sijoitetaan fieldset-elementin sisälle. Fieldset-elementin avulla pystytään ryhmittämään kenttiä eri kokonaisuuksiin ja antamaan kokonaisuuksille ryhmää kuvaava otsikko. Ryhmän otsikolle annetaan nimi fieldset-elementin sisään tulevaan legend-elementtiin, joka on pakollinen HTML-syntaksin mukaan.

Lomakkeiden tyylimäärittely fieldset-elementtiä hyväksikäyttäen on mielletty web-kehittäjien keskuudessa haasteelliseksi tehtäväksi. Tästä syystä lomakekenttien ryhmittely ratkaistaan usein muilla menetelmillä. Pienellä vaivalla fieldset-elementillä pystyy luomaan kuitenkin näyttävän ja ennen kaikkea loogisen tavan ryhmitellä kentät isoimmiksi kokonaisuuksiksi. Kokonaisuudet tekevät lomakeviidakosta helppolukuisemman ja visuaalisesti miellyttävämmän.

Lomakkeiden taustaväri on hillitty valkoisesta vaalean siniseen juokseva liukuväri. Fieldset-elementit (kuva 17) on rajattu reunaviivoilla mikä helpottaa ryhmien erottamista toisistaan. Jokaisella yksittäisellä syötekentällä tulee olla selite tai otsake, joka toteutetaan label-elementillä. Label-elementti on aina lomakekentän edessä vasemmalle tasattuna. Vasemmalle tasattu, aina samasta kohdasta alkava lomakekenttä on isoissa kokonaisuuksissa luettavampi, kuin keskitetty tai oikealle tasattu ratkaisu. Label-elementtien oikealla puolella olevat syötekentät tasataan myös vasemmalle syötteen tyylistä riippumatta. Syötekentät erotetaan toisistaan liukuvärillä ja kentän alareunassa olevalla katkoviivalla.

Nimitiedot

Sukunimi:

Etunimet:

Kutsumanimi:

Yhteydenotto

Puhelinno (työ):

Sähköposti (työ):

Kuva 17. Kaksi fieldset-elementin avulla ryhmitettyä lomakekenttäkokonaisuutta.

7.6 Taulukot

Raporttinäkömystä (kuva 18) käyttäjät voivat tarkistaa erilaisista taulukoista täyttämiensä lomakkeiden tuloksia. Taulukot voivat toimia myös pelkästään esimiehillä näkyvänä yhteenvetona työntekijöiden raporttien tilasta. Taulukoiden juurielementtinä on table, jonka sisään ladotaan yksittäisistä soluista koostuvia rivejä. Tavallista tietoa sisältävän solun taustalla on sama liukuväri kuin lomakkeissa. Lisäksi rivit ja solut erotetaan toisistaan reunaviivoilla. Palstojen otsikkosoluissa on tummansininen taustaväri, jotta ne erottuisivat muusta taulukkotiedosta. Otsikkotekstin kirjasin on lihavoitu ja sen väri on valkoinen. Yleensä taulukon viimeisenä rivinä on yhteensä-sarake, joka on erotettu muista riveistä tummemmalla taustavärillä ja reunaviivalla sekä lihavoidulla tekstillä. Taulukko skaalautuu automaattisesti koko sisälön levyiseksi. Tämä helpottaa rakennusprosessissa kehittäjän työtä, sillä taulukoille ei tarvitse erikseen määrittää leveyttä.

Henkilötiedot Toimintayksikkö 1 Tukihenkilöstö KOOSTA RAPORTTI

Henkilötiedot

Otoksen koko: 21

Numeroiden selitykset

- ei vastausta
- 1 ei osaamista
- 2 perusosaaja
- 3 osaava suorittaja
- 4 ammattilainen
- 5 ekspertti

	- (kpl)	Osaaminen					Keskiarvo
		1	2	3	4	5	
1. Alaotsikko							
1.1 Ensimmäinen rivi	0	0	3	2	1	2	3,25
1.2 Toinen rivi	0	0	1	2	2	0	2,67
1.3 Kolmas rivi	0	2	0	2	4	0	3
Yhteensä	0	2	4	6	7	2	2,97

Kuva 18. Raporttinäkömä. Alareunassa taulukko. Ylälaidassa valintalaatikko raportin koostoa varten.

7.7 Typografia

Kohderyhmän ikäjakauman laajuus täytyi huomioida erityisesti tekstin kokoa valittaessa, sillä iäkkäämmät käyttäjät näkevät usein pientä tekstiä heikommin. Tekstille täytyi määrittää riittävän iso kiinteä koko, jotta sovellus pysyisi helppolukuisena myös iäkkäämpien ja heikonäköisempien käyttäjien keskuudessa.

Varsinaisen sisällön otsikoiden ja leipätekstien kirjasintyyliksi valittiin Verdana. Se on web-ympäristössä yleisesti käytetty ja näyttölaitteilla helppolukuinen. Leipätekstin ja hyperlinkkien koko on kiinteä 12 pikseliä. Ylänavigaatiossa linkkien tekstit määritettiin versaaliksi.

Lomakkeissa fieldset-elementein ryhmitetyillä kentillä on usein kokonaisuutta kuvaava nimi, joka annetaan legend-elementissä. Kirjasimen koko on 12px ja se on lihavoitu. Kenttien selitykset merkataan label-elementillä. Label-kirjasimen koko on sama kuin leipätekstillä.

7.8 Otsikot

Jokaisella sivulla on H1-tason otsikko, josta selviää millä sivulla käyttäjä on. Otsikon koko on kiinteä 20px ja HTML-syntaksin perusasetuksena oleva lihavoitu on poistettu. Otsikon alapuolella juoksee 1px leveä viiva, joka erottaa otsikon varsinaisesta sisällöstä ja selkeyttää rakennetta. Muita otsikkotyylejä ei tässä työssä käytetty, mutta jatkoa ajatellen määritettiin myös muut otsikkotyylit vastaamaan Mediamaisteri Groupin ilmettä. Otsake-lohkossa sijaitsevan sovelluksen otsikon kirjasimena on Mediamaisterin käyttämä Eurostile LT. Fontin koko on 32px ja värin tulee erottua taustan väristä.

7.9 Värät

Värimaailmassa noudatettiin Mediamaisteri Groupin graafisessa ohjeistuksessa määriteltyjä linjoja. Graafisen ohjeistuksen typografiamäärittelyt olivat käytännössä painetulle materiaalille tehty, mutta niitä voitiin soveltaa myös web-ympäristöön.

Sisällön taustaväri on valkoinen, joten leipätekstiksi valittiin siitä parhaiten erottuva musta. Sivuston linkit ovat vaaleansinisiä. Vietäessä osoitin linkin päälle väri muuttuu tummansiniseksi ja linkki alleiviivataan, jotta käyttäjä erottaa linkin leipätekstistä. H1-otsikoiden väri on tummansininen ja niiden alapuolella oleva viiva on vaaleansininen. Legend-elementin väri on Mediamaisterin tukiväriä käyttämä oransi.

Navigaatioiden värit noudattavat muun sovelluksen linjaa. Ylänavigaation ollessa aktiivinen väri on tumman oranssi. Linkki on perustilassaan harmaa ja vietäessä osoitin sen päälle se muuttuu tummansiniseksi. Lisäksi aktiivista linkkiä korostaa tekstin alapuolella oleva sininen kolmio. Sivunavigaation tekstin väri on perustilassa vaaleansininen ja linkin edessä on linkin havainnollistamista helpottava vaaleansinisellä pohjalla oleva valkoinen nuoli. Linkin ollessa aktiivinen muuttuu linkin väri tummansiniseksi ja vaaleansininen pohja nuolessa vaihtuu tummansiniseksi.

8 Tyylikäsikirja

Jatkoprojekteja varten luotiin uudesta sivupohjasta dokumentoitu tyylikäsikirja. Dokumentti päätettiin kirjoittaa HTML-muotoon, mikä tekee tiedon etsimisen ja koodin kopioimisen helpommaksi jatkokehittäjille. Käsikirjassa selitetään yksityiskohtaisesti sivupohjan rakenne niin HTML- kuin CSS-koodin osalta. Rakenteen selvityksen lisäksi dokumentoinnissa näytetään esimerkein mitä elementtejä HTML-tiedostossa tarvitsee olla ja miten ne tulee nimetä, jotta sivupohja pysyisi ehyenä. Käsikirjassa selvitetään myös miten sivupohjan ulkoasuun voi tehdä muutoksia tyyli-tiedostosta käsin.

Tyyli-tiedostossa kommentoidaan isommat kokonaisuudet ja elementti-ryhmät, jotta koodi pysyy lukijalle helppolukuisempana ja tyyli-tiedoston rakenne selkeänä. Perustelujen lisäksi dokumentoinnissa kerrotaan esimerkein rakenteen oikeaoppiset muokkaustavat, esimerkiksi miten navigaatioiden ja muiden toiminnallisuuksien poistot ja lisäykset toteutetaan HTML-koodissa.

9 Pohdinta

9.1 Yhteenveto

Mediamaisteri Group etsi syksyllä 2007 opinnäytetyöntekijää tai harjoittelun suorittajaa, jonka tehtäväkuvana oli laatia mCompass-tuoteperheelle yhtenäinen visuaalinen ulkoasu käytettävyyks ja markkinointihenkisyys huomioon ottaen. Opinnäytetyöntekijän tai harjoittelijan tuli määrittellä tuoteperheessä käytettävät tyylit CSS-muotoisena ja luoda Mediamaisterin käyttöön tyylikäsikirja joka sisältää tuoteperheessä käytetyt tyylimääritykset. Mediamaisteri otti yhteyttä alan opiskelijoihin koulun postituslistojen kautta, ja vastattuani viestiin sain haastattelun kautta opinnäytetyöpaikan 3.9.2007 alkaen.

Projekti alkoi Mediamaisterin toimintatapoihin ja tuotteisiin perehdyttämisellä sekä henkilökuntaan tutustumisella. Tavoitteiden läpikäynnin jälkeen alkoi sivupohjan hahmottelu ja mahdollisten uusien ominaisuuksien kartoittaminen. Viikoittaisissa palaverissa käytiin läpi projektin etenemistä. Myöhemmässä vaiheessa järjestettiin myös markkinointiryhmän kokouksia, joihin osallistui Mediamaisterin imagon vahvistamiseen palkattu alan erikoisosaja. Näissä tapaamisissa käytiin läpi erityisesti Mediamaisterin graafisen ilmeen vahvistamista sekä sitä, miten uusia linjoja tullaan noudattamaan tulevissa tuotteissa.

Sivupohjan ulkoasun suunnitteluun ei annettu kovin tarkkaa määrittelmää, mikä ohjeiden ja mielipiteiden erotessa oli lopulta melko turhauttavaa. Käyttöliittymän rakenteen ja ulkoasuratkaisujen osalta ei myöskään tullut toimeksiantajan puolelta selkeää noudatettavaa linjaa. Markkinoinnin palaverien jälkeen asiat kuitenkin selkeytyivät ja sivupohjan työstäminen helpottui. Aikaisempien sivupohjien teemoja haluttiin käyttää vielä hyväksi, mutta samalla haluttiin jotain uutta, joka tulisi pysymään pitkään mCompass-tuoteperheen ilmeessä. Tämä loi etenkin projektin alussa lieviä paineita työn onnistumiselle.

9.2 Tulosten arviointi ja tulevaisuus

Projekti onnistui mielestäni hyvin ja se saatiin valmiiksi määräaikaan mennessä (21.12.2007). Uusi sivupohja on jo integroitu moneen mCompass-tuoteperheen sovellukseen. Käytännön työn valmistuttua olen kuullut henkilökunnalta ja asiakkailta käyttäjäkokemuksia, mitkä ovat järjestäen olleet positiivisia. Käyttäjien puolelta varsinkin käyttöliittymän ulkoasusta ja käytettävyydestä on pidetty. Myös uusien työkalujen helppokäyttöisyyttä ja toiminnallisuutta on kiitelty asiakkaiden toimesta.

Työtä tehdessäni opin jatkuvasti uutta ja kehityin paljon varsinkin CSS-koodauksessa. Työn edetessä aikaisemmat ratkaisut tuntuivat jälkikäteen oudoilta ja niitä jouduttiin usein korjaamaan järkevämpään muotoon. Tämä aiheutti tunteen jatkuvasta hienosäädön tarpeesta, ja työskentelymenetelmänä tuntui olevan kokeileminen ja korjaaminen. Tyyli tiedostot tulivat projektin aikana erittäin tutuiksi ja rutiini koodaamiseen kehittyi isoin harppauksin. Myös PHP-ohjelmointi kertaantui projektin edetessä.

Koska sovellukset ovat laajoja ja rakenteeltaan monimutkaisia, on sovellusten yhtenäisyys ja loogisuus loppukäyttäjälle tärkeää. Välilehtien, lomakkeiden ja taulukoiden tulee olla yhdenmukaisia ja käyttäjän tulee pystyä päättelemään nopeasti kenttien täyttöjärjestys. Sivupohjan rakennetta ja ulkoasua selkeytettiin paljon niin loppukäyttäjän kuin jatkokehittäjänsäkin näkökulmasta. Sovellukseen tuotiin myös uusia toiminnallisuuksia, jotka lisäävät helppokäyttöisyyttä.

Visuaalinen näyttävyys ja markkinointihenkinen ulkoasu olivat tärkeitä seikkoja, ja niiden tuomisessa uuteen sivupohjaan onnistuttiin hyvin. Sovelluksen käytettävyys ei kuitenkaan saanut kärsiä niiden takia. Tasapainottelu näyttävyuden ja käytettävyyden suhteen onnistui kuitenkin hyvin ja lopputuloksena on selkeä ja myyvä paketti.

Sivupohjan koodipuolen rakenne ja toiminnallisuus haluttiin uudistaa myös jatkokehittäjien kannalta. Mielestäni tämä toteutui työn tavoitteista parhaiten, ja kehittäjiltä saatu palaute on ollut positiivista. Varsinkin navigaatioiden aktiivisten linkkien vaihdon automatisointi ilman JavaScript-ohjelmointia oli haastavaa, mutta olen lopputulokseen tyytyväinen. Uuden sivupohjan suunnitteluvaiheessa oli alun perin tarkoitus luoda työkalu tekstin koon vaihdolle sekä tulostusnäkyville, mutta ajanpuutteen vuoksi niiden toteutus jätettiin muhimaan myöhemmäksi ajankohdaksi. Ominaisuuksien työstäminen koettiin lopulta todella haastavaksi; tässä voisikin olla aihe seuraavalle opinnäytetyöntekijälle Mediamaisteri Groupissa.

Työlle asetetut tavoitteet täyttyivät ja opinnäytetyön pohjalta uusien sovelluksien rakennusprosessista tuli tehokkaampi sekä helpompi ja sovelluksen graafisesta ilmeestä markkinointihenkisempi. Lisäksi sovellus sai lisätyökaluja, jotka helpottavat ylläpitoa ja parantavat käytettävyyttä loppukäyttäjien osalta. Tyylikäsikirja ei vielä ole ollut käytännön kokeilussa, mutta uskon sen toimivan hyvänä apuvälineenä jatkokehittäjille antaen heille valmiudet kehittää sovellukseen uusia ratkaisuja ja helpottavan sivupohjan ongelmien selvittämisessä.

Lähdekritiikki

Opinnäytetyötä tehdessäni ei minulla ollut varsinaista päälähdettä. Lähdeaineistoa WWW-sivujen luonnista löytyy todella paljon, mutta akuuttien käytännön ongelmien ratkaisuun käytin ajankohtaisia sähköisiä lähteitä. Näistä hyvänä esimerkkinä web-kehittäjille tarkoitettu kehityssivusto W3 Schools (<http://www.w3schools.com/>) Lisäksi prosessin edetessä sain käsiini Andy Buddin kirjoittaman CSS Mastery: Advanced Web Standards Solutions (2006) kirjan, josta löytyi myös apua käytännön ongelmiin ja vinkkejä järkeviin CSS-koodin toteutustapoihin. Kirjoittamisprosessin aikana tärkeät lähteet olivat Vesa Keräsen, Niko Lambergin ja Jukka Penttisen Web-julkaiseminen & multimedia (2006) sekä Tero Linjaman XHTML-kirja (2001).

Internetin ja sivunrakennustekniikoiden kehittyessä myös toteutustavat ja -tekniikat lisääntyvät. Ajankohtaisista ja pienistä mutta tärkeistä käytännön seikoista on erittäin vaikeaa, ellei jopa mahdotonta löytää kirjallisessa muodossa olevaa lähdeaineistoa. Internet-foorumeilla ja -sivustoilla on todella suuret määrät tietoa web-kehittämiseen liittyvistä asioista. Lainatessa tekstiä verkkosivuilta tulee kuitenkin ottaa huomioon kysymys materiaalin luotettavuudesta, sillä internetissä kirjoittaja voi olla kuka tahansa ja nimimerkkien taakse on helppo piiloutua. Pyrin kuitenkin käyttämään mahdollisimman luotettavia lähteitä kuten verkkoartikkeleita (kuten A List Apart, <http://www.alistapart.com/>) joiden kirjoittajat ovat alansa tunnustettuja ammattilaisia. Foorumeilta puolestaan saa usein ilmaiseksi hyviä neuvoja toteutustapoihin ja käytännön ongelmien ratkaisuun. Näiden neuvojen toimivuus selviää yksinkertaisesti kokeilemalla.

Lähteet

Painetut lähteet:

Keränen, Vesa, Lamberg, Niko & Penttinen, Jukka 2006. Web-julkaiseminen & Multimedia. Jyväskylä: Docendo.

Korpela, Jukka K & Linjama, Tero 2004. XHTML-käsikirja. Jyväskylä: Docendo.

Linjama, Tero 2001. XHTML. Jyväskylä: Docendo.

Nielsen, Jakob 2000. WWW-suunnittelu. Suomentaja Timo Haanpää. Jyväskylä: IT Press.

Zeldman, Jeffrey 2003. Designing with Web standards. Indianapolis: New Riders.

Sähköiset lähteet:

A List Apart [online] [viitattu 18.4.2008]. <http://www.alistapart.com/>

CSS Beauty [online] [viitattu 18.4.2008]. <http://www.cssbeauty.com/>

Käytettävyys Wikipedia [online] [viitattu 8.4.2008] .
<http://fi.wikipedia.org/wiki/K%C3%A4ytett%C3%A4vyys>

Mediamaisteri Group Oy [online] [viitattu 19.3.2008].
<http://www.mediamasteri.com/web/>

W3 Schools [online] [viitattu 18.4.2008]. <http://www.w3schools.com/>

Liitteet

Liite 1 – Graafinen ohjeistus

Värit

Mediamaisteri Groupin värimaailma rakentuu vahvan sinisen sävyn ympärille. Väri on ollut käytössä yrityksen alkua ajoista lähtien ja muodostunut lähtemättömäksi osaksi imagoa. Tumma laivas-tonsininen sävy esiintyy hallitsevana värinä yrityksen käyntikorteissa, verkkosivuilla sekä dokumenttipohjien graafisissa elementeissä. Lisäväreinä käytetään raikkaan oranssia, valkoista sekä harkitusti vaaleata harmaata ja mustaa.



CMYK
83
51
27
59

PMS
PANTONE 539 U
SOLID UNCOATED



CMYK
0
30
100
0

PMS
PANTONE 130 U
SOLID UNCOATED

Typografia

Eurostile esiintyy ainoastaan Mediamaisteri Groupin logon kirjasimena, sitä ei käytetä muihin tarkoituksiin. Logon maksimikoko riippuu pitkälti suunnittelijan harkinnasta. Massiivisen suurta fonttikokoa ei kuitenkaan suositella käytettävän.

Logon väritys on kirjoitettuna joko Mediamaisterin sininen, musta tai valkoinen. Valkoisen tekstin taustavärinä käytetään aina joko mustaa tai Mediamaisterin sinistä värialuetta. Mahdollisuuksien mukaan tulisi aina kuitenkin käyttää valmista logotiedostoa.

Verkossa leipätekstinä käytetään Verdanaa.

Universia käytetään konsernin typografista ilmettä tukevana elementtinä. Sitä hyödynnetään esimerkiksi printtigrafiikassa otsikkofonttina sekä verkkosivuilla navigaatioissa ja osa-alueiden otsikoinnissa.