



Anu Pohjonen

KÄYTTÖLIITTYMÄN TOTEUTUS VARASTONHALLINTAJÄRJESTELMÄÄN HTML5-TEKNIKALLA

KÄYTTÖLIITTYMÄN TOTEUTUS VARASTONHALLINTAJÄRJESTELMÄÄN HTML5-TEKNIKALLA

Anu Pohjonen
Opinnäytetyö
Syksy 2012
Tietojenkäsittelyn koulutusohjelma
Oulun seudun ammattikorkeakoulu

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma

Tekijä: Anu Pohjonen

Opinnäytetyön nimi: Käyttöliittymän toteutus varastonhallintajärjestelmään HTML5-tekniikalla

Työn ohjaaja: Jouni Juntunen

Työn valmistumiskuukausi ja –vuosi: Syksy 2012

Sivumäärä: 30

Opinnäytetyö tehtiin toimeksiantona Taitonetti-nimiselle oululaiselle pienyritykselle. Työn tarkoituksena oli tutkia HTML5-tekniikan tarjoamia uusia ominaisuuksia sekä käyttää näitä hyödyksi varastonhallintajärjestelmän käyttöliittymän toteutuksessa. Tarkoituksena oli käyttää HTML5-tekniikkaa erityisesti sivuston rakenteessa sekä lomakkeissa siten, että järjestelmä toimisi ongelmitta yrityksessä käytössä olevalla Mozilla Firefox –selaimella.

Teoriaosuudessa käsiteltiin ensin yleisesti HTML-tekniikan kehitysvaiheita. Lisäksi käsiteltiin joitakin HTML4-tekniikan puutteita, jotka ovat johtaneet omalta osaltaan uuden HTML-version kehittämisen tarpeeseen. HTML5-tekniikasta tarkemmin esiteltiin uudet semanttiset elementit ja niiden hyödyntämistä web-sivuston rakentamisessa sekä osa lomake-elementeistä ja attribuuteista. Mediaelementeistä lyhyesti esiteltiin myös video- ja audio-elementit. Viimeisenä tutkittiin uusimpien selainversioiden selainyhteensopivuutta HTML5-tekniikan kanssa. Lähteinä käytettiin suurimmaksi osaksi verkkolähteitä.

Kehitettävänä olleessa järjestelmässä käytettiin onnistuneesti semanttisia elementtejä sivurakenteen kuvaamisessa. Uusista lomake-elementtien tyypeistä käytettiin ainoastaan tel- ja search-tyyppejä, sillä käytettävänä ollut Firefox-selain ei tarjonnut tarvittavaa tukea muille järjestelmän kannalta oleellisille kentille. Attribuutit tarjosivat selkeästi suurimman hyödyn lomakkeiden toteutuksessa, sillä ne vähensivät huomattavasti JavaScriptin tarvetta. Hyöty oli suuri varsinkin selainpuolella tapahtuneessa lomakkeiden validoinnissa.

Avainsanat: HTML5, käyttöliittymä, web-lomakkeet

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Business Information Systems

Tekijä: Anu Pohjonen

Title of thesis: Implementing HTML5 technique in the user interface of a storage management system

Supervisor: Jouni Juntunen

Term and year when the thesis was submitted: Fall 2012

Number of pages: 30

This thesis was commissioned by a small company called Taitonetti located at Oulu. The purpose of the thesis was to explore the new features of the HTML5 technique and use those in the implementation of the user interface of a storage management system. The aim was to use the HTML5 technique especially in the markup of the site structure and in the web forms in a way that the system would work without any problems in the company Mozilla Firefox browser.

The theoretical background of the thesis first dealt with the overall development stages of the HTML technique. There was also discussion on some of the weaknesses in the HTML4 technique which in turn led to the need of developing a new HTML version. More profoundly introduced were the new semantic elements of HTML5 technique and the use of those in the markup of a website and also some of the new form input types and attributes. The video and audio media elements were also briefly showcased. Lastly the browser support of the latest browsers concerning the HTML5 technique was examined. Most of the source material was from web based sources.

In the system implementation the semantic elements in describing the site markup was successfully used. From the new form input types only the tel and search types were used because the Firefox browser did not support any other relevant field types needed in the system. In the making of the forms, the attributes turned out to be most beneficial because they notably reduced the need for JavaScript. The advantage was significant especially in the form validation in the browser.

Keywords: HTML5, user interface, web forms

SISÄLLYS

1	JOHDANTO	6
2	TOIMEKSIANNON ESITTELY	7
2.1	Toimeksiantaja	7
2.2	Tavoite ja käyttötarkoitus	7
2.3	Kehitystehtävät	8
2.4	Toimeksiannon rajaukset	9
3	HTML-TEKNIikka	10
3.1	Kehitysvaiheet	10
3.2	HTML4-tekniikan puutteet	11
4	HTML5	13
4.1	Uudet ominaisuudet	13
4.2	Lomakkeet	15
4.3	Selainyhteensopivuus	16
5	VARASTONHALLINTAJÄRJESTELMÄN TOTEUTUS	18
5.1	Järjestelmän rakenne	18
5.2	HTML5:n hyödyntäminen sivurakenteessa	19
5.3	HTML5:n hyödyntäminen lomakkeissa	20
6	POHDINTA	25
	LÄHTEET	27
	LIITTEET	29

1 JOHDANTO

Web-sivujen rakenteen merkinnässä käytetyn HTML-kielen nykyinen standardi kehitettiin aikana, jolloin internet koostui lähinnä staattisesta sisällöstä. Aikojen muuttuessa on internetistä kuitenkin kehittynyt dynaaminen kokonaisuus, joka rakentuu pitkälti web-sovelluksista ja –lomakkeista. Web-kehittäjien parissa onkin huomattu, että HTML4-standardi ei ole enää riittävä täyttämään modernin internetin vaatimia tarpeita web-kehityksen osalta. Nykyisten sivustojen monimutkaiset rakenteet ja dynaamiset kokonaisuudet yhdessä lisääntyneen multimedian käytön kanssa ovat pakottaneet tarkastelemaan uudelleen HTML-tekniikkaa.

Tuloksena onkin ollut täysin uuden HTML-standardin, HTML5:n, kehitystyön aloittaminen. HTML5:n toivotaan ratkaisevan nykyisen web-kehityksen ongelmia tarjoamalla tähän tarkoituksenmukaiset työkalut. Tähän mennessä HTML5 on tuonut web-kehittäjien käyttöön mm. semanttiset elementit sekä uudet lomake-elementtien tyypit ja attribuutit. HTML5:n käyttö ei kuitenkaan ole ongelmatonta. Virallista standardia ei vielä ole, joten selainyhteensopivuudessa on eroja. Erityisesti Internet Explorerin tapauksessa yhteensopivuus HTML5:n kanssa on tällä hetkellä lähes olematonta.

Opinnäytetyön toimeksiantona oli varastonhallintajärjestelmän toteutus oululaiselle pienyritykselle. Näkökulmaksi valikoitu lopulta käyttöliittymän toteutus HTML5-tekniikalla, sillä se tarjosi hyvät työkalut kyseisen järjestelmän käyttöliittymän toteutukseen. Lisäksi HTML5 on tulevaisuudessa tärkeä osa web-kehitystä, joten sen osaaminen on oman ammattitaidon kannalta tärkeää. Opinnäytetyössä on tutkittu HTML5-tekniikan tarjoamia mahdollisuuksia web-sovellusten kehittämisessä sekä tarkasteltu työn tekemisen ajankohtana vallinnutta selainyhteensopivuutta. Käytännön osuudessa on esitelty järjestelmän toteutuksessa käytettyjen tekniikoiden käyttöä. Huomionarvoista on se, että HTML5:n toimivuus tämän järjestelmän osalta on testattu ainoastaan Mozilla Firefox –selaimella, joten muiden selainten kannalta selainyhteensopivuutta ei voida taata.

2 TOIMEKSIANNON ESITTELY

2.1 Toimeksiantaja

Työn toimeksiantaja on Taitonetti-niminen oululainen pienyritys. Yritys perustettiin vuonna 2005 alun perin nettikahvilaksi. Nettikahvilan pito päätettiin kuitenkin lopettaa, sillä tietokoneiden myynti ja huolto alkoivat viedä enemmän liiketilaa. Vuodesta 2009 lähtien Taitonetti onkin keskittynyt ainoastaan tietokoneiden ja niiden oheislaitteiden myyntiin ja huoltoon.

Taitonetin liiketoiminta perustuu sekä käytettyjen että uusien tietokoneiden ja tietokoneiden oheislaitteiden myyntiin. Tärkeä osa Taitonetin toimintaa on myös tietokoneiden huoltaminen. Taitonetissä kiinnitetään erityisesti huomiota asiakaspalveluun ja pyritään näin luomaan pitkiä asiakkuussuhteita.

Liiketoiminnan kasvun vuoksi Taitonetti tarvitsee yksinkertaista varastohallintajärjestelmää, joka on räätälöity heidän tarpeitaan vastaavaksi. Tällä hetkellä yrityksellä ei ole käytössään minkäänlaista järjestelmää tuotteidensa varastosaldojen hallinnointiin. Varastohallintajärjestelmän tueksi yritys kaipaa myös rekisteriä, johon myyntitapahtuman myötä syntyneet takuutodistukset asiakastietoineen tallentuvat.

2.2 Tavoite ja käyttötarkoitus

Tavoitteena opinnäytetyössä on toteuttaa käyttövalmis web-pohjainen varastohallintajärjestelmä, joka tulee toimimaan yrityksen lähiverkossa sen omalla palvelimella. Opinnäytetyön käytännön osuudessa keskitytään HTML5-tekniikalla toteutettuihin ratkaisuihin. HTML5-tekniikkaa käytetään, sillä se tarjoaa HTML4-tekniikkaa paremmat työkalut järjestelmän ulkoasun ja toiminnallisuuden toteuttamiseen. HTML5 helpottaa esimerkiksi järjestelmään tulevien lomakkeiden tekemistä mahdollistamalla tiedon validoinnin ilman JavaScript-ratkaisuja. HTML5 on myös taaksepäin selainyhteensopiva, joten sitä voi käyttää, vaikka nykyiset selaimet eivät tarjoa vielä täyttä tukea. Kehitettävän järjestelmän päätavoitteena on tuotteiden varastosaldojen hallinnointi. Tämän johdosta varastohallinta tulee nopeutumaan, mikä puolestaan vähentää tähän kuluva työaika.

Järjestelmä tulee ainoastaan yrityksen työntekijöiden käyttöön. Kaikilla käyttäjillä tulee olemaan samat oikeudet järjestelmään. Näin ollen eri käyttäjärooleja ja näiden mukaisia näkymiä ei tarvita. Järjestelmään ei myöskään vaadita minkäänlaista kirjautumistoimintoa. Järjestelmää tullaan käyttämään tuotteiden varastosaldojen seuraamiseen sekä nopeuttamaan ja helpottamaan myyntitapahtumaa järjestelmään valmiiksi tallennetun takuutodistusohjan ansiosta.

2.3 Kehitystehtävät

Tärkeimpiä järjestelmän sisältämiä tuotteiden hallintaan liittyviä toimintoja ovat tuotteiden lisäys, muokkaus ja poisto. Tuotteen varastosaldoa on pystyttävä vähentämään ja lisäämään tuotenäkymästä. Lisäksi järjestelmän tulee ilmoittaa käyttäjälle, mikäli tuotteen varastosaldo on liian pieni. Järjestelmässä tulee olla valmiina takuutodistusohja, johon myyntitapahtuman yhteydessä pystytään lisäämään myyty tuote, hinta, takuu, maksaja, puhelinnumero sekä myyntipäivämäärä. Syntynyt takuutodistus on voitava tallentaa tietokantaan.

Tässä opinnäytetyössä kehitettävänä olevan järjestelmän toiminnalliset vaatimukset on kuvattu yksityiskohtaisesti liitteessä 1. Toimintoja on mahdollista järjestelmän valmistuttua lisätä tai muokata asiakkaan tarpeiden mukaan, mutta mahdolliset muutokset jäävät tämän opinnäytetyön ulkopuolelle.

Kehitettävän järjestelmän on oltava helposti asennettava ja käyttöön otettava. Asennuksen jälkeen järjestelmän ylläpidosta huolehtii yrityksen henkilökunta, joten on huolehdittava, että ylläpidettävyys on mahdollisimman yksinkertaista. Tähän pyritään esimerkiksi riittävällä koodin kommentoinnilla. Lisäksi tuotteen mahdollisen jatkokehityksen on oltava helppoa. Järjestelmä on myös tarpeen tullen voitava siirtää ongelmitta toiselle palvelimelle.

Järjestelmän kehitysohjelma tulee olemaan MVC-arkkitehtuuria noudattava CodeIgniter-ohjelmistokehys. Toteutuksessa käytetään PHP-ohjelmointikieltä ja relaatiotietokantana MySQL-tietokantaa. Käyttöliittymä toteutetaan HTML5- ja CSS3-tekniikoilla. Kyseinen järjestelmä vaatii toimiakseen palvelimen, jossa on PHP:stä vähintään versio 5.1.6 sekä MySQL:stä versio 4.1. HTML5:n ja CSS3:n toimiminen vaatii selaimen, jossa on tarvittavat tuet näille. Koska kyseessä on web-sovellus, tarvitaan järjestelmän käyttämiseen lähiverkossa myös toimiva verkkoyhteys sekä Internet-selain. Yrityksessä on käytössä Mozilla Firefox –selain, joten järjestelmän on toimittava vähintään kyseisen selaimen viimeisimmissä versioissa.

2.4 Toimeksiannon rajaukset

Toimeksianto rajoittuu järjestelmän suunnitteluun sekä sen toteutukseen. Näin ollen opinnäytetyön ulkopuolelle rajataan järjestelmän testaus, käyttöönotto sekä käyttäjien opastus järjestelmän käyttöön. Dokumentointi toteutetaan ainoastaan koodin kommentointina. Tuotteiden kirjaaminen järjestelmään, kaikenlainen muu sisällön tuotto sekä mahdollisten kuvien hankinta kuuluu toimeksiantajalle. Toimeksiantajalle jäävät myös järjestelmän toiminnan vaativien teknisten ratkaisujen, kuten Internet-yhteyden ja palvelimen edellyttämät järjestelyt.

Toimeksianto toteutetaan tässä opinnäytetyössä esitettyjen toiminnallisten (liite1) ja ei-toiminnallisten vaatimusten mukaisesti. Mahdolliset lisäykset ja muutokset järjestelmään tämän jälkeen eivät sisälly opinnäytetyöhön.

3 HTML-TEKNIikka

HTML eli HyperText Markup Language on merkintäkieli, jota käytetään web-sivustojen rakenteen kuvailussa. HTML:ää käytettäessä sivun sisältö rakentuu erilaisista elementeistä. (Moore 2005, 2.) HTML:n ansiosta web-sivustojen tekijät voivat mm. julkaista dokumentteja, joissa on esimerkiksi otsikoita, tekstiä, tauluja ja kuvia sekä suunnitella lomakkeita eri tarkoituksia varten (W3C 2012b, hakupäivä 1,10.2012).

3.1 Kehitysvaiheet

Alun perin HTML:n kehitti Tim Berners-Lee työskennellessään CERN:ssä. 1990-luvulla internetin räjähdysmäisen kasvun myötä myös HTML on laajentunut monin tavoin. Vuonna 1995 IETF (Internet Engineering Task Force) julkaisi HTML 2.0 -version. Tämän jälkeen HTML-standardin kehittäminen siirtyi W3C:lle (World Wide Web Consortium), jonka HTML:n kehitysryhmä julkaisi vuonna 1997 suosituksen HTML 3.2:sta. (W3C 2012b, hakupäivä 1,10.2012.)

HTML:n kehittämistä jatkettiin edelleen ja vuoden 1997 lopulla julkaistiin HTML 4.0 eli ensimmäinen versio HTML4:sta. Tätä päivitettiin vuosien 1997 ja 1998 aikana versioon HTML 4.01. (W3C 1999, hakupäivä 1,10.2012.) Vuonna 2000 HTML standardoitiin ISO-standardina nimellä ISO/IEC:2000 (Iso/iec 2000, hakupäivä 1.10.2012). HTML4 toi web-kehitykseen mm. lisää multimediavaihtoehtoja sekä CSS- eli tyylitiedostot (Cascading Style Sheets) (W3C 1999, hakupäivä 1,10.2012). Tämän jälkeen W3C lakkasi kehittämästä HTML:ää ja alkoi työskennellä XHTML:n (eXtensible HyperText Markup Language) parissa (Whatwg 2012, hakupäivä 1.10.2012).

XHTML on tiukempi ja puhtaampi versio HTML 4.01:stä. XHTML kehitettiin yhdistämällä HTML sekä XML, joka on syntaksiltaan erittäin tiukasti määritelty merkintäkieli. XHTML ja HTML eroavat toisistaan mm. tiedoston rakenteessa, elementeissä ja attribuuteissa. (W3schools 2012, hakupäivä 4.10.2012.)

Vuonna 2000 W3C julkisti XHTML 1.0:n. Tämän jälkeen se alkoi työstää uutta standardia, XHTML2:ta, joka ei ollut yhteensopiva aiempien HTML:n ja XHTML:n kanssa. Kuitenkin vuonna 2003 alkoi kiinnostus HTML:n kehittämiseen uudelleen web-lomakkeiden tekemiseen käytettävän teknologian kehityttyä. W3C päätti silti jatkaa XML-tekniikkaan perustuvien vaihtoehtojen

kehittämistä, mistä syystä Apple, Mozilla ja Opera perustivat WHATWG:n (Web Hypertext Application Technology Working Group) ja alkoivat omalta osaltaan uuden HTML-standardin kehittämisen. Vuonna 2007 myös W3C päätti lopulta ottaa osaa HTML5:n kehitystyöhön. (Whatwg 2012, hakupäivä 1.10.2012.) Lopullisesti XHTML2:n kehitys päättyi W3C:n osalta vuonna 2009 (W3C 2009, hakupäivä 4.10.2012).

HTML5 on tällä hetkellä vasta kehitysvaiheessaan. Vuoden 2014 loppuun mennessä HTML 5.0:sta olisi tarkoitus saada aikaan W3C:n suositus. Tähän päästäkseen HTML Working Group Chairs esitti vuoden 2012 syyskuussa suunnitelman, jonka mukaan tulisi samanaikaisesti sekä stabiloida HTML 5.0 että jatkokehittää tulevia HTML-ominaisuuksia. Suunnitelmassa mainitaan myös, että ryhmä tulee tuottamaan HTML 5.1 –suosituksen vuoden 2016 loppuun mennessä. (W3C 2012c, hakupäivä 6.10.2012.)

Jokainen uusi HTML-versio on pyrkinyt entisestään yhdenmukaistamaan web-kehittäjien tuottamia dokumentteja, jotta ne pysyisivät käyttökelpoisina pidempään. On tärkeää, että HTML-tiedostot toimivat kehittäjän tarkoittamalla tavalla eri selaimissa ja laitteissa. HTML:n kehitysvisiona onkin ollut pyrkimys siihen, että erityyppisen laitteet, kuten esimerkiksi älypuhelimet ja tietokoneet, olisivat kelpoisia esittämään web-sivujen sisältöä. (W3C 2012b, hakupäivä 1.10.2012.)

3.2 HTML4-tekniikan puutteet

HTML-tekniikka suunniteltiin lähinnä staattisten web-dokumenttien rakentamiseen. Internetin muututtua dynaamisemmaksi kokonaisuudeksi on HTML4:n suureksi puutteeksi muodostunut se, ettei standardi huomioi tarpeeksi web-sovelluksien ja -lomakkeiden toteutuksessa tarvittavia ominaisuuksia. (Whatwg 2012, hakupäivä 12.10.2012.)

Lomakkeiden muodostamista varten HTML4 määrittelee käytännössä ainoastaan tekstikentän, painikkeet sekä valintaruudut. Kenttiin syötettävä tieto on kuitenkin monipuolistunut sähköposti- ja nettiosoitteiden myötä, joten pelkkä teksti-kenttä ei tarjoa tarpeeksi ominaisuuksia erityyppisten tietojen käsittelemiseksi. HTML4 ei sisällä työkaluja mm. lomakekenttiin syötettyjen tietojen tarkastamiseen tai käyttäjälle annettaviin virheilmoituksiin, vaan tämä on suoritettava JavaScriptillä. JavaScriptiä on käytettävä myös tietytyjen käyttäjäkokemusta parantavien ratkaisujen toteuttamiseen. Tällaisia ovat esimerkiksi cursorin automaattinen kohdistus johonkin kenttään tai kentän sisältämä vihjeteksti. (D'Agostino 2011, hakupäivä 12.10.2012.)

Ongelmallista on myös web-sivun rakenteen selkeä kuvaus. Rakenne merkitään HTML4-tekniikalla muodostetussa dokumentissa div-elementeillä, joita saattaa olla yhdessä dokumentissa paljon, sillä jokainen sivun osio sijoitetaan yleensä omaan div-elementtiin. Dokumentit ovat yleensä vaikeaselkoisia, sillä pelkillä div-elementeillä kuvattua sisältöä voi olla vaikeaa hahmottaa eri osiot, kuten esimerkiksi ylä- ja alatunnisteet. (Cox 2011, hakupäivä 12.10.2012.)

Puutteita HTML4:ssä on myös multimedian sisällyttämisessä web-sivustoille. HTML4 ei sisällä standardia tähän, joten selaimet eivät tue luonnostaan videon tai äänen toistoa, vaan vaativat multimedian toistoon erillisen lisäosan asentamisen selaimen. Nykyään multimedia on kuitenkin hyvin oleellinen osa internetiä, joten mediasisällön asettaminen sivustoille sekä sen toistaminen tulisi olla helpompaa ja yhdenmukaista. (Hogan 2011, 122.)

4 HTML5

HTML5 tarjoaa työkalut modernien ja entistä parempien web-sovellusten rakentamiseen. Uusien ominaisuuksien myötä kehittäjät pystyvät luomaan parempia käyttäjäkokemuksia sovellusten käyttäjille. HTML5:n merkintäkieleen on lisätty uusia elementtejä, jotka kuvaavat entistä tarkoituksenmukaisemmin tuottamaansa sisältöä. Myös käyttöliittymiin HTML5 tuo yhdessä CSS3:n kanssa selkeän parannuksen, sillä esimerkiksi taulukoiden muotoilu, reunaviivojen kulmien pyöristys tai tietyt lomaketoiminnot eivät vaadi enää JavaScriptin tai raskaiden tyyli-merkintöjen käyttöä. (Hogan 2011, 1-3.)

4.1 Uudet ominaisuudet

Ensimmäinen huomattava uudistus HTML4:ään verrattuna on HTML5-dokumentin tyyppimäärittelyn yksinkertaistaminen. HTML5-dokumentin alkuun kirjoitetaan ainoastaan seuraava teksti:

```
<!DOCTYPE html>
```

Tämä riittää määrittelemään HTML5-dokumentin ja antamaan käyttöön kaikki käytössä olevan selaimen tukevat HTML5-elementit ja -ominaisuudet. HTML4:n vaatimaan määrittelyyn verrattuna ero on huomattava (kuva 1). (Hogan 2011, 4.)

HTML5 määrittelee täysin uusia elementtejä sekä web-sivun rakenteen että sisällön hallintaan. Rakenteen hallintaan käytettäviä elementtejä kutsutaan semanttisiksi elementeiksi. Semanttisten elementtien avulla voidaan paremmin ymmärtää sivuston rakenne suoraan HTML-syntaksin perusteella. Yleensä web-sivu jaetaan eri alueisiin, kuten esimerkiksi ylä- ja alitunnisteeseen, navigointiin ja sisältöön. Kuten edellisessä luvussa todettiin, HTML4:ssä jokainen osio sijoitetaan omaan div-elementtiinsä. HTML5 kuitenkin tarjoaa mm. näille kokonaan uudet elementit. Sisältö sijoitetaan section- ja article-elementteihin, joilla on helppo jäsenellä rakennetta. Section-elementtiä käytetään sivun jakamisessa eri osiin samankaltaisten sisällönosien mukaisesti. Article taas kuvastaa itsenäistä sisällönosaa, kuten esimerkiksi uutisartikkelia. Aside-elementillä merkitään sellaista sisältöä, joka tarjoaa jotakin lisätietoa pääsisältöön, kuten esimerkiksi diagrammit tai linkit toisille sivustoille. Aside-elementtiä ei siis tule käyttää sivuston sivupalkin merkinnässä, vaan tätä varten käytetään section-elementtiä.

Navigaatiota varten on määritelty nav-elementti. Header-elementillä merkitään yleensä sivun ylätunniste ja footer-elementillä alatunniste. Header- ja footer-elementit voivat kuitenkin sisältää mitä tahansa sisältöä, kuten esimerkiksi yrityksen logon, hakukentän tai navigoinnin. Molempia elementtejä voi myös olla useita yhdellä sivulla, joten esimerkiksi jokaisella artikkelilla voi olla oma header- ja footer-elementti. (Hogan 2011, 16-23; W3C 2012a, hakupäivä 8.10.2012.)

Kuvassa 1 on esimerkki siitä, millainen ero samankaltaisen sivurakenteen tuottamisessa HTML4- ja HTML5-tekniikoilla on.



KUVA 1. Samankaltainen sivu tuotettuna HTML4- ja HTML5-tekniikoilla.

Muita uusia elementtejä ovat rakenteellisten elementtien lisäksi niin sanotut mediaelementit eli video ja audio. Näiden mediaelementtien myötä HTML5 pyrkii pääsemään eroon musiikin kuuntelua tai videon katselua varten tarvittavista lisäosista. HTML4 ei siis mahdollista vastaavaa toimintaa, vaan selaimeen on oltava asennettuna median toistoa varten erillinen lisäosa, esimerkiksi Flash tai Quicktime. (Borja 2011, hakupäivä 8.10.2012.) Toistaiseksi kuitenkin mediaelementtien ongelmana on se, ettei yksikään suosituista video- tai audioformaateista toimi kaikissa tunnetuimmissa selaimissa. Esimerkiksi MP3-formaatille ei ole tukea Firefoxissa eikä Operassa. (Hogan 2011, 126.)

4.2 Lomakkeet

HTML 4.01:n muovautumisen aikaan internet sisälsi yleisesti ottaen ainoastaan staattista tekstiä, joten HTML-standardiin ei ollut tarvetta asettaa web-lomakkeita tukevia ominaisuuksia. Internet on kuitenkin muuttunut näistä ajoista dynaamisemmaksi, joten web-lomakkeiden käytöstä on tullut arkipäivää. Tyylikkäiden ja helppokäyttöisten web-lomakkeiden tekeminen on kuitenkin melko työlästä HTML4:llä. HTML5:n tarjoamien uusien lomakeominaisuuksien myötä web-lomakkeiden tekeminen helpottuu ja muuttuu yhdenmukaisemmaksi. HTML5 pyrkii standardoimaan yleisimmät lomakekontrollit, jotta toiminnallisuuksien saavuttamiseksi ei tarvitsisi käyttää monimutkaisia JavaScript-ratkaisuja. (W3C wiki 2012, hakupäivä 9.10.2012.)

Web-lomakkeet ovat web-sovellusten pääasiallinen keino käyttäjän syöttämän tiedon keräämisessä ja sen lähettämisessä palvelimelle. Lomakkeet voivat sisältää mm. painikkeita, valintaruutuja ja tekstikenttiä. HTML-lomake muodostetaan <form>-tageilla. Näin muodostuneen elementin sisälle sijoitetaan tarvittavat lomake-elementit. (Tizag 2008b, hakupäivä 10.10.2012.) Tällaisia elementtejä kutsutaan input-elementeiksi. Input-elementeillä luodaan lomakkeisiin erilaiset kentät ja painikkeet. Type-attribuutilla määrätään, minkä tyyppinen input-elementti on kyseessä. HTML4 määrittelee seuraavat tyypit: text, checkbox, radio, button, submit, reset, password ja hidden. (Tizag 2008a, hakupäivä 10.10.2012.)

HTML5 tuo käyttöön useita uusia input-elementin tyyppisiä, joilla voidaan kattavammin kuvailla tietyn lomakekentän tietotyyppiä. Uusia tyyppisiä ovat mm. email, tel, url, number, color, search ja calendar. Mikäli input-elementin tyyppi on esimerkiksi email, voidaan tällaiseen lomakekenttään kirjoittaa ainoastaan sähköpostiosoite tai useita sähköpostiosoitteita. (W3C wiki 2012, hakupäivä 10.10.2012.) Search-tyyppiä käytetään hakukentän luomiseen. Kenttä itsessään ei kuitenkaan tarjoa toiminnallisuutta haun suorittamiseen, vaan hakutoiminto tulee kirjoittaa koodiin. Kentän ominaisuuksiin kuuluu mm. pieni x-merkki, jolla saadaan tyhjennettyä kentän sisältö. Tällä hetkellä ainoastaan Safari esittää kentän x-merkin kanssa. Useimmat selaimet näyttävät searchin toistaiseksi tavallisena tekstikenttänä. (Harris 2012, hakupäivä 30.10.2012.) HTML5 määrittelee myös uusia attribuutteja, jotka helpottavat joidenkin yleisimpien tehtävien suorittamista ilman JavaScriptiä. Autofocus-attribuutilla voidaan sivun latauduttua asettaa tietty kenttä automaattisesti aktiiviseksi. Placeholder-attribuutilla taas voidaan määrätä, millainen vihjeteksti halutaan tyhjään kenttään. Min ja max määrittelevät number-kenttään hyväksyttävän luvun minimi- ja maksimiarvon. (W3C wiki 2012, hakupäivä 10.10.2012.)

Myös lomakkeen validointi onnistuu HTML5:llä ilman JavaScriptiä, koska uusien input-elementtien ja attribuuttien ansiosta selain tietää, millaisia arvoja kenttiin voidaan hyväksyä. Required-attribuutilla taas voidaan kertoa selaimelle, mitkä kentistä eivät voi olla tyhjiä. Mikäli jokin pakollisista kentistä on tyhjä, selain ei lähetä lomaketta eteenpäin palvelimelle ja antaa käyttäjälle virheilmoituksen puuttuvasta tiedosta. Virheilmoitus tulee myös, mikäli selain havaitsee, että jossain kentässä on väärintyyppistä tietoa. Validointia varten lomake-elementille on aina annettava myös name-attribuutti. (W3C wiki 2012, hakupäivä 10.10.2012.)

Kaikilla kentillä ei kuitenkaan ole olemassa oletusarvoista validointisääntöä. Esimerkiksi email- ja url-kenttien tarkistusta varten selaimissa on valmiina erityiset validointisäännöt, mutta tel-kenttä hyväksyy oletusarvoisesti kaikenlaiset syötteet. Tämä johtuu siitä, että puhelinnumeroilla ei ole yleispätevää formaattia eli ne eivät noudata mitään tiettyä mallia. Mm. puhelinnumeroiden tarkistusta varten HTML5 määrittelee pattern-attribuutin. Tämän avulla kehittäjä voi määrätä tietyn mallin, jota kenttään kirjoitettavan syötteen on noudatettava. Selain suorittaa syötteentarkistuksen annetun mallin mukaisesti. Pattern-attribuuttia voidaan käyttää myös text-, search- url-, email- sekä password-kentissä. (Gravelle 2011, hakupäivä 30.10.2012.) Esimerkki validointimallin määrittämisestä puhelinnumerokenttään:

```
<input type="tel" pattern="[+]\d{2}[(\d{2})]\d{4}[-]\d{4}" title="Puhelinnumero (Format: +99(99)9999-9999)">
```

Lomakkeiden validointi selaimessa ei kuitenkaan aina ole toivottavaa. Kehittäjä saattaa esimerkiksi haluta tarkistaa, kuinka syötteentarkistus toimii palvelimella. Koska lomakkeet validoidaan automaattisesti selaimessa, tarvitaan erityinen attribuutti kertomaan selaimelle, ettei lomaketta tule validoida. Tämä novalidate-niminen attribuutti sijoitetaan form-elementtiin ja se siis syrjäyttää kyseisen lomakkeen kentille annetut validointisäännöt. (Wofoo 2012b, hakupäivä 28.11.2012.)

4.3 Selainyhteensopivuus

Koska HTML5 on edelleen kehitysvaiheessaan, selainyhteensopivuus tulee tulevaisuudessa muuttumaan parempaan suuntaan. Opinnäytetyötä kirjoittaessa parhaiten HTML5-testistä suoriutui Chrome 22. Lähes yhtä hyvin pärjäsivät Maxthon 3.4.5, Opera 12.00, Safari 6.0 ja Firefox 15 jäivät hieman enemmän kahden ensimmäisen suorituksista. Internet Explorer 9 puolestaan sai selkeästi huonoimman tuloksen. Chrome, Opera, Safari ja Firefox tukevat hyvin HTML5:tä myös

hieman vanhemmissa selainversioissaankin. Sitä vastoin Explorerin aiemmissa versioissa tuki on erittäin heikkoa. (Html5test 2012, hakupäivä 6.10. 2012.)

Tällä hetkellä selaimista Opera tarjoaa parhaan tuen uusille lomakkeiden input-tyypeille ja selaimessa tapahtuvalle validoinnille. Muut suosituimmat selaimet ovat työstämässä tuen parantamista. HTML5:n lomake-elementtejä voi silti käyttää, mikäli kehittäjä ottaa huomioon, että tuen puuttuessa selain muuttaa kentän tavalliseksi tekstikentäksi. Käyttäjä voi siis täyttää ja lähettää lomakkeen normaalisti, mutta selain ei esimerkiksi suorita tiedon tarkastamista. (W3C wiki 2012, hakupäivä 10.10.2012.) Tätä kirjoittaessa Internet Exploreria lukuunottamatta selaimet tukevat input-tyypeistä email-, tel- ja url-tyyppejä. Attribuuteille puolestaan löytyy paremmin tukea, tosin jälleen Explorerista tuki puuttuu. Explorer 10:stä eteenpäin tuki kuitenkin pitäisi löytyä kaikille edellämainituille ominaisuuksille. (Wofoo 2012a, hakupäivä 10.10.2012.)

Internet Explorer ei kuitenkaan ole ainoa HTML5:n käyttöönottoa jarruttava selain. Google, Apple ja Mozilla Foundation käyvät omaa taistoaan video- ja audioformaattien tukemisessa selaimissa ja vaikuttavat näin omalta osaltaan HTML5:n käyttömahdollisuuksiin. Safari esimerkiksi soittaa audio-elementtiin sisällytetyt mp3-tiedostot, mutta ei ogg-tiedostoja. Firefoxissa taas tiedostojen toimiminen on juuri päinvastoin. (Hogan 2011, 9.)

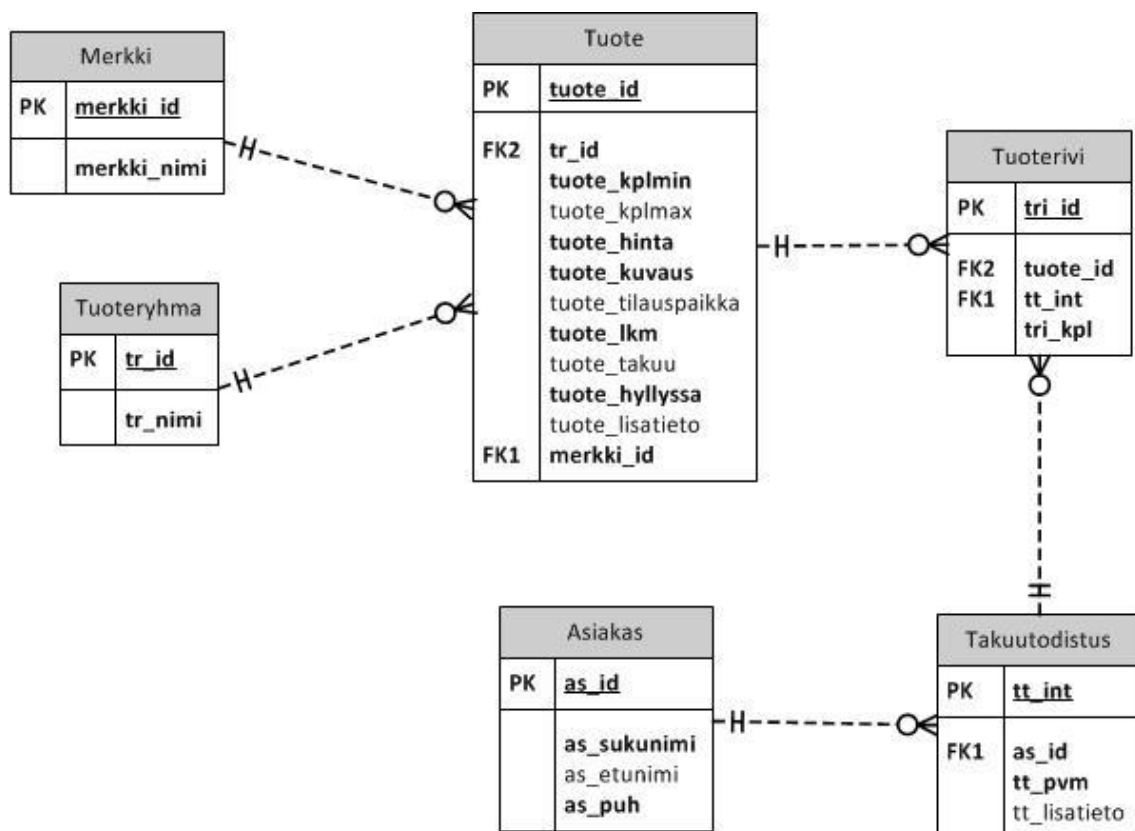
Kuten todettua HTML5 on vasta kehitysvaiheessaan. Lopullista suositusta ei ole annettu, joten nykyinen voimassaoleva spesifikaatio voi milloin tahansa muuttua. Muutosten sattuessa HTML5:ttä vahvasti tukevat selaimet, kuten Firefox, Chrome ja Safari, muuttuvat spesifikaation mukana. Tämä voikin johtaa rikkonaisiin web-sivuihin, mikäli HTML5:ttä käyttävät web-kehittäjät eivät seuraa alan kehitystä ja reagoi muutoksiin. (Hogan 2011, 9.)

5 VARASTONHALLINTAJÄRJESTELMÄN TOTEUTUS

Järjestelmää tullaan käyttämään kohdeyrityksessä ainoastaan Mozilla Firefox –selaimen uusimmilla versiolla. Näin ollen järjestelmän toteutuksessa käytettyjä HTML5-ratkaisuja tarkastellaan tässä opinnäytetyössä ainoastaan Firefoxin kannalta. Työssä ei siis oteta kantaa sivuston toimivuuteen muissa selaimissa.

5.1 Järjestelmän rakenne

Kuvassa 2 on järjestelmän toteutuksessa käytetty tietokanta. Kuvassa PK:lla on merkitty taulun perusavain ja FK:lla mahdolliset viiteavaimet. Tummennetulla tekstillä olevat ominaisuudet tulevat olemaan tietokannassa pakollisia tietoja.

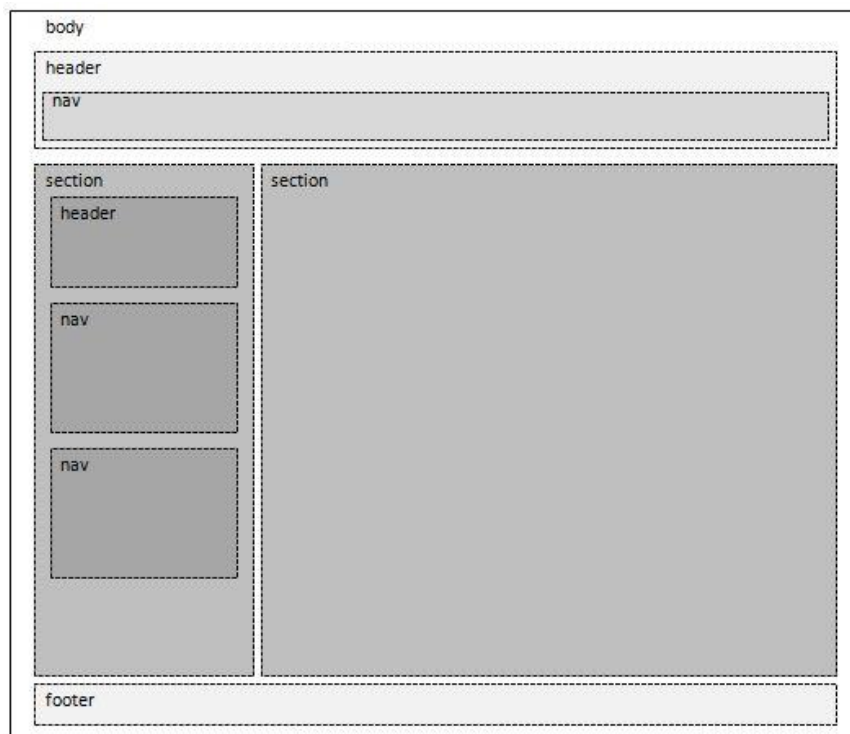


KUVA 2. Järjestelmän tietokanta er-kaaviona esitettynä.

Järjestelmän toteutuksessa on käytetty MVC-arkkitehtuuria noudattavaa CodeIgniter-ohjelmistokehystä. Järjestelmän sivumallia varten on luotu omat view-tiedostot ylätunnisteelle, navigoinnille, sivupalkille sekä alatunnisteelle. Näiden lisäksi on luotu lukuisia yksilöllisiä view-tiedostoja erilaisia sisältönäkymiä varten. Sivumalli-tiedostoja kutsutaan template-tiedoston kautta yhdessä kulloinkin tarvittavan sisältönäkymän kanssa. Sivuston rakenteesta poikkeavat takuutodistus- ja hinnasto-sivut eivät rakennu sivumalli-tiedostoista. Model-tiedostot puolestaan on luotu tuotteelle, merkille, tuoteryhmälle, takuutodistukselle sekä asiakkaalle. Takuutodistustusta lukuunottamatta edellämäinuilte on myös luotu controller-tiedostot. Sivuston pääkontrollina toimiva varasto-controller ohjaa takuutodistukseen ja hinnastoon liittyviä toimintoja. Sivustolla käytetään myös kolmea eri tyylitiedostoa sekä yhtä JavaScript-tiedostoa.

5.2 HTML5:n hyödyntäminen sivurakenteessa

Sivuston HTML-dokumentin alkuun on kirjoitettu dokumentin tyyppin määrittelyksi `<!DOCTYPE html>`, joka siis luvun 4.1 mukaisesti riittää kertomaan selaimelle, että kyseessä on HTML5-tiedosto. Sivuston rakenteen merkinnässä on hyödynnetty samassa kappaleessa esiteltyjä HTML5:n tarjoamia uusia header-, nav-, footer-, section- ja article- elementtejä. Kuvassa 3 on esitetty pääpiirteittäin sivuston sivurakenne.



KUVA 3. Sivuston sivurakenne HTML5-elementeillä.

Sivustolla on siis vasemmalla puolella sivupalkki, jonka sisältö pysyy koko ajan samana. Sivupalkin oikealle puolelle tulostuu kulloisenkin sivun varsinainen sisältö. Sivustolla on myös päänavigointipalkki, ylätunniste ja alatunniste. Luvun 4.1 mukaisesti navigointi on merkitty nav-elementillä, joka puolestaan on sijoitettu header-elementtiin yhdessä ylätunnisteessa olevan sivuston logon kanssa. Alatunniste taas on merkitty sille tarkoitetulla footer-elementillä. Section-elementtiä on käytetty sivupalkin ja sisältö-osion erottamisessa toisistaan. Luvussa 4.1 on todettu, että section-elementillä jaetaan sivu eri osiin samankaltaisten sisällönosien mukaisesti, joten section-elementin käyttö sivupalkin ja varsinaisen sisällön merkinnässä on perusteltua. Sisältö-osioon mahdollisesti tulevat itsenäiset sisällönosat taas merkitään article-elementeillä. Sivupalkki sisältää yhden hakukenttä-osion ja kaksi navigointi-osioa. Nämä on luvussa 4.1 esitetyn ohjeistuksen mukaisesti merkitty header- ja nav-elementeillä.

Takuutodistus- ja hinnasto-sivut eivät noudata pääsivuston sivurakennetta. Molemmat sivut avautuvat uuteen ikkunaan ja ne ovat A4-mallisia näkymiä. Pääsivustosta poiketen kummassakaan ei myöskään ole lainkaan navigointia. Takuutodistuksen header-elementti sisältää näin ollen yrityksen logon, otsikon ja päivämääräkentän. Takuutodistus-sivulla on erillisinä osioina ostajan tiedot, yrityksen yhteystiedot, ostetut tuotteet sekä lisätiedot. Nämä osiot on merkitty sivurakenteeseen article-elementeillä. Yrityksen tiedot sisältävä alatunniste on sijoitettu footer-elementin sisään.

Hinnasto-sivu on hyvin yksinkertainen tuloste hyllyssä myytävänä olevista tuotteista ja niiden hinnoista. Sen rakenteen merkinnässä on riittänyt ainoastaan header- ja section-elementtien käyttö.

5.3 HTML5:n hyödyntäminen lomakkeissa

Sivustolla on käytetty lomakkeita tuoteryhmän, -merkin, tuotteen ja asiakkaan lisäämisessä ja muokkaamisessa. Lisäksi on toteutettu kolme hakukenttää. Uusista lomakkeiden input-elementtien tyypeistä järjestelmässä on käytetty hyödyksi tel- ja search-tyyppejä. Searchille tosin Firefox ei vielä järjestelmää tehdessä tarjoa tukea, kuten luvussa 4.2 onkin todettu. Tämä ei kuitenkaan estä search-tyypin käyttöä, sillä tuen puuttuessa selain kohtelee sitä tavallisena text-kenttänä.

Tel-tyyppiä on käytetty nimensä mukaisesti puhelinnumero-kentissä. Tälle tyyppille myös Firefox tarjoaa tuen. Tel-tyypin yhteydessä on käytetty pattern-attribuuttia, sillä tel-kentälle ei ole

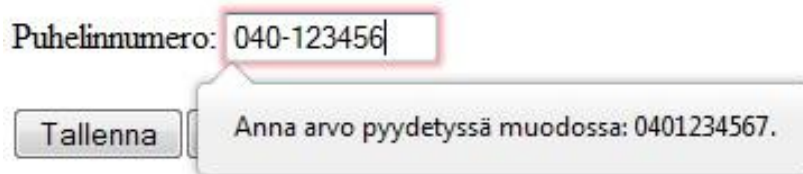
olemassa email- ja url-kenttien tapaan yleistä validointisääntöä. Mikäli siis pattern-attribuuttia ei käytetä, selain ei suorita tel-tyyppisen kentän syötteen validointia. Toimeksiantajan toivomuksesta puhelinnumero saa tässä varastohallintajärjestelmässä sisältää ainoastaan numeroita ja sen tulee olla tasan 10 merkkiä pitkä. Tällainen validointisääntö saadaan luotua antamalla pattern-attribuutille seuraava arvo:

```
\d{10}
```

Mikäli taas puhelinnumeron tulisi sisältää väliviiva kolmen ensimmäisen numeron jälkeen, olisi pattern-attribuutille tällöin annettava seuraava arvo:

```
\d{3}[-]d{7}
```

Mikäli tel-kentälle annetaan title-attribuutti, voidaan tällä ilmaista käyttäjälle missä muodossa puhelinnumero tulee antaa. Jos käyttäjä kirjoittaa numeron väärässä muodossa, selain ympäröi kentän huomiovärillä ja antaa käyttäjälle virheilmoituksen. Tähän ilmoitukseen sisältyy titlen arvo, mikäli se on annettu. Kuvassa 4 on tästä esimerkki.



KUVA 4. Esimerkki virheellisestä syötteestä.

Muita järjestelmän lomakkeissa käytettyjä HTML5-attribuutteja ovat autofocus, placeholder ja required. Luvun 4.2 mukaisesti autofocus määrittelee sen kentän, jonka tulee olla aktiivinen sivun latauduttua. Tämä on käytössä tuoteryhmä-, tuotemerkki-, tuote- ja asiakaslomakkeilla. Kun käyttäjä siirtyy tuoteryhmä- tai tuotemerkkilomakkeelle, on kursori valmiina siinä kentässä, johon kirjoitetaan uuden artikkelin nimi. Tuotelomakkeella aktiivisena on ensimmäinen pudotusvalikko. Asiakaslomakkeelle siirryttäessä kursori asetetaan asiakkaan sukunimi -kenttään. Sama kenttä on aktiivisena myös takuutodistus-sivulle siirryttäessä. Placeholder-attribuuttia on käytetty lähes kaikissa kentissä. Varsinkin search-hakukentissä placeholder oli tarpeellinen, sillä sen avulla voidaan kertoa käyttäjälle kyseisellä hakukentällä haettava artikkeli ilman labelin käyttöä.

Kuvassa 5 on esimerkki hakukentistä ja niiden placeholder-teksteistä. Kuvan ylempi hakukenttä on tuotettu seuraavalla koodilla:

```
<input type="search" name="txtKategoria_etsi" placeholder="Tuoteryhmä/merkki"
maxlength="70" size="16" required />
```

The image shows a screenshot of a web form. It contains two input fields stacked vertically. The top input field has the placeholder text "Tuoteryhmä/merkki" and a search button labeled "Etsi" to its right. The bottom input field has the placeholder text "Tuote" and another search button labeled "Etsi" to its right. The entire form is enclosed in a light gray border.

KUVA 5. Placeholder-attribuutin käyttö hakukentissä

Myös required-attribuuttia on käytetty paljon järjestelmän lomakkeilla. Kaikille niille kentille, jotka ovat kuvassa 2 esitettyssä tietokannassa pakollisia, on annettu tämä attribuutti. Required-attribuutilla siis kerrotaan selaimelle, mitkä kentät eivät voi olla tyhjiä. Esimerkiksi tuotelomakkeella pakollisia tietoja ovat tuotteen kuvaus, hinta, lukumäärä ja minimikappalemäärä. Mikäli jotkut näistä kentistä ovat tyhjiä lomaketta lähetettäessä, ympäröi selain tällaiset kentät huomiovärillä ja antaa ilmoituksen tyhjistä kentistä. Kaikille lomakkeiden elementeille on myös annettu name-attribuutit, sillä luvun 4.3 mukaan kenttien validointi ei muutoin onnistu.

Kuvassa 4 olevan lomakekentän toteutuksessa on käytetty kolmea eri HTML5-attribuuttia; pattern placeholder ja required. Tämä kenttä on toteutettu seuraavalla koodilla:

```
<input type="tel" name="telAs_puhnro" placeholder="Puhelinnumero" pattern="\d{10}"
title="0401234567" size="11" required />
```

Titlen sama arvo sisältyy siis tässä tapauksessa mahdolliseen virheilmoitukseen, kuten kuvasta 4 voidaan todeta.

Kuvassa 6 on kuva lomakkeesta, jolla lisätään tuote järjestelmään. Pakolliset kentät on merkitty punaisella asteriskilla. Hinta-, lukumäärä- ja takuu-kentissä on käytetty apuna pattern-attribuuttia. Lukumäärä- ja takuu-kentät hyväksyvät ainoastaan positiivisia kokonaislukuja ja hinta-kenttään puolestaan voidaan kirjoittaa myös desimaalilukuja. Desimaalierottimena voidaan käyttää joko pilkkua tai pistettä. Hinta-kentän toteutuksessa on käytetty seuraavaa koodia:

```
<input type="text" id="txtTuote_hinta" name="txtTuote_hinta" placeholder="Hinta"
maxlength="8" size="7" pattern="\d+(,|.)?(,|.)?\d{0,2}?" title="desimaaleina"
```

Uuden tuotteen lisääminen

Tuoteryhmä:	<input type="text" value="Hiiri"/>
Tuotemerkki:	<input type="text" value="Acer"/>
Tuotteen kuvaus:	<input type="text" value="Tuotteen kuvaus"/>
Hinta:	<input type="text" value="Hinta"/> euroa
Lukumäärä:	<input type="text" value=""/>
Minimikappalemäärä:	<input type="text" value=""/>
Maksimikappalemäärä:	<input type="text" value=""/>
Takuu:	<input type="text" value="Takuu"/> kuukautta
Tilauspaikka:	<input type="text" value="Tilauspaikka"/>
Hyllyssä:	<input checked="" type="radio"/> Kyllä <input type="radio"/> Ei
Lisätietoja:	<input type="text" value="Lisätietoja"/>

KUVA 6. Lomake tuotteen lisäämiseksi.

Takuutodistus sisältää lomakkeen, jossa on kentät ostajan puhelinnumerolle, yritykselle/sukunimelle, etunimelle ja ostosten lisätiedoille. Näistä kaksi ensimmäistä ovat pakollisia kenttiä. Takuutodistus sisältää myös tulosta-, tallenna- ja lopeta-painikkeet. Tallenna-painike tallentaa takuutodistukseen syötetyt ostajatiedot, istuntomuuttujassa olevat tuoterivit sekä itse takuutodistuksen tietokantaan. Tallenna- ja lopeta-painikkeille on liitetty JavaScript-varmistetta ja tallenna-painikkeelle lisäksi JavaScript-ilmoitus tallennuksen onnistumisesta. Järjestelmää tehdessä huomasin kuitenkin, että JavaScript suoritetaan oletusarvoisesti ennen selaimen tekemää validointia. Näin ollen käyttäjälle esimerkiksi tuli ilmoitus takuutodistuksen tallentamisesta, vaikka lomaketta ei todellisuudessa tallennettu puutteellisten syötteiden vuoksi. Koska kuitenkin JavaScript-ilmoitusten on tultava vasta siinä vaiheessa, kun lomake on läpäissyt selaimen validoinnin, oli JavaScript-koodiin lisättävä `checkValidity()`-metodi. Metodi palauttaa arvon `true`, mikäli lomake läpäisee selaimen validoinnin. Kuvassa 7 on esitetty tallenna-

painikkeen JavaScript-koodi. Nähdään siis, että JavaScript-varmistus suoritetaan vain siinä tapauksessa, kun lomake on läpäissyt validoinnin. Muussa tapauksessa selain ilmaisee käyttäjälle virheelliset kentät. CheckValidity()-metodin yhteydessä oleva todistus-sana on tutkittavan lomakkeen id. Lomakkeelle on siis aina annettava jokin id HTML-koodissa, mikäli tähän käytetään checkValidity()-metodia.

```
function varmista_tallennus() {
    if (document.getElementById('todistus').checkValidity()) {
        if (confirm("Haluatko varmasti tallentaa?")) {
            alert("Takuutodistus tallennettu");
            return true;
        }
        else {
            return false;
        }
    }
}
```

KUVA 7. CheckValidity()-metodin käyttö.

Takuutodistuksen lopeta-painike taas tyhjentää istuntomuuttujan ja sulkee ikkunan. Tässä tapauksessa lomaketta ei kuitenkaan tarvitse validoida, joten lomakkeelle annetaan luvussa 4.2 mainittu novalidate-attribuutti. Kuvassa 8 on lopeta-painikkeen JavaScript-koodi, jolla asetetaan novalidate-attribuutti lomakkeelle, jonka id on todistus. Ilman tätä attribuuttia takuutodistusta ei siis voisi sulkea, mikäli lomakkeen kentissä olisi puutteellista tietoa.

```
function varmista_lopetus() {
    document.getElementById('todistus').setAttribute('novalidate', 'true');
    if (confirm("Haluatko varmasti lopettaa?")) {
        return true;
    }
    else {
        return false;
    }
}
```

KUVA 8. Novalidate-attribuutin asettaminen lomakkeelle JavaScriptillä.

6 POHDINTA

Opinnäytetyössä tarkoituksena oli perehtyä HTML5-tekniikkaan sekä sen tarjoamiin dynaamisen web-kehityksen työkaluihin. Toimeksiantona oli varastonhallintajärjestelmän suunnittelu ja toteutus pienyritykselle. Opinnäytetyössä esiteltiin tämän web-pohjaisen järjestelmän käyttöliittymän toteutuksessa käytettyjä HTML5-tekniikoita. HTML5-ratkaisujen oli toimittava moitteetta Mozilla Firefox –selaimessa, joten järjestelmän käyttöliittymän toimivuus varmistettiin ainoastaan kyseisessä selaimessa. Opinnäytetyön tekeminen alkoi syyskuussa ja kohdeyrityksen toiveena oli saada järjestelmä käyttöön vuoden loppuun mennessä. Aikataulullisesti tavoite onnistui eikä työn tekemisessä tullut suurempia ongelmia tai viivästyksiä.

Semanttisten elementtien käyttämisessä selain ei aiheuttanut ongelmia. Firefox osaa esittää elementit oikein sivustolla ja tukee CSS3-tiedostoja. Uudet elementit helpottivat sivurakenteen merkitsemistä sekä hahmottamista verrattuna aiempiin kokemuksiin div-elementtien kanssa. Myös CSS-tiedostojen tyylimäärittelyt olivat selkeämmät toteuttaa käyttämällä semanttisia elementtejä. Suurin käytännön hyöty HTML5-tekniikasta kuitenkin tuli selkeästi ilmi lomakkeita tehdessä. Uusista attribuuteista required ja pattern olivat tämän työn kannalta hyödyllisimmät. Niiden käyttö vähensi huomattavasti JavaScriptin tarvetta, sillä ne mahdollistivat lomakkeiden validoinnin suoraan selaimessa. Myös placeholder- ja autofocus-attribuutteja käytettiin hyväksi lomakkeiden toteutuksessa. Näiden avulla web-kehittäjien on helppo parantaa järjestelmän käyttömukavuutta.

Vähemmälle käytölle jäivät lomake-elementtien eri tyyppien käyttäminen, sillä toistaiseksi Firefoxin tarjoama tuki niille on liian vähäistä. Järjestelmässä hyödynnettiin ainoastaan search- ja tel-tyyppisiä. Search-tyypillekään ei toistaiseksi ole tukea, mutta sen käytölle ei silti ollut esteitä, sillä Firefox kohtelee kyseistä kenttää text-kenttänä. Tel-kenttä oli hyvin hyödyllinen yhdessä pattern-attribuutin kanssa, sillä toimeksiantajan toiveena oli saada puhelinnumero tallennettua tietokantaan tarkkaan määrättyssä muodossa. Mikäli toiveena on tulevaisuudessa jonkin toisenlaisen muodon salliminen, on pattern-attribuutin arvoa helppo muuttaa verrattuna esimerkiksi JavaScriptillä tapahtuvaan syötteentarkistukseen.

Alunperin järjestelmässä oli tarkoitus käyttää myös number-tyyppiä, mutta ongelmia ilmeni attribuuttien kanssa. Käyttöliittymää toteuttaessa kävi ilmi, että number-tyypin kanssa ei virallisen ohjeistuksen mukaan tule käyttää maxlength- ja size-attribuutteja, vaan näiden tilalle on number-

tyypin omat min- ja max-attribuutit. Lomakkeet toimivat ongelmitta maxlength- ja size-attribuuttienkin kanssa, mutta W3C:n virallisesta validoinnista dokumentti ei mennyt läpi virheittä. Koska Firefoxin ei toistaiseksi tue number-tyyppiä, päädyttiin siis kenttien muotoilun yksinkertaistamiseksi käyttämään järjestelmän lukumääräkentissäkin text-tyyppiä. Selaimen tuen kehittyessä kenttien tyypit on kuitenkin helppo myöhemmin muuttaa number-tyypeiksi, mikäli toimeksiantaja näkee tämän tarpeelliseksi.

Toinen hieman ongelmallinen seikka oli JavaScript-varmistusten ja ilmoitusten käyttö yhdessä HTML5-tekniikalla toteutettujen lomakkeiden yhteydessä. Koska JavaScript suoritetaan ennen selaimen validointia, antoi järjestelmä käyttäjälle virheellisiä ja turhia ilmoituksia, mikäli lomake ei läpäissyt validointia. Tähän kuitenkin auttoi checkValidity()-metodi, jolla voidaan ohjata tarkistusten kulkua. Myös novalidate-attribuutti oli tarpeellinen, sillä lomakkeen validointi ei ollut aina suotavaa.

Kaiken kaikkiaan kokemus HTML5:n käytöstä web-sovellusten kehittämisessä oli erittäin positiivinen erityisesti lomakkeita tehdessä. Tosin jotain muuta selainta käytettäessä olisi HTML5:stä ollut vielä enemmänkin hyötyä järjestelmässä, sillä nyt Firefoxin kanssa sekä search-että number-tyyppien tarjoamat ominaisuudet jäivät toistaiseksi kokonaan käyttämättä. Tulevaisuudessa kuitenkin selainyhteensopivuuden koko ajan parantuessa myös Firefoxin käyttäjät pääsevät hyötymään näistäkin.

LÄHTEET

- Borja, J. 2011. Guide to HTML5's New Media Tags – Audio and Video. Hakupäivä 8.10.2012, <http://speckyboy.com/2011/05/02/guide-to-html5s-new-media-tags-audio-and-video/>.
- Cox, P. 2011. Top 10 reasons to use HTML5 right now. Hakupäivä 12.10.2012, <http://tympanus.net/codrops/2011/11/24/top-10-reasons-to-use-html5-right-now/>.
- D'Agostino, A. 2011. The New and Improved Way to Create Forms with HTML5. Hakupäivä 12.10.2012, <http://webdesignledger.com/tips/the-new-and-improved-way-to-create-forms-with-html5>.
- Gravelle, R. 2011. HTML5 Forms: How To Use The New Email, URL, and Telephone Input Types. Hakupäivä 30.10.2012, <http://www.htmlgoodies.com/html5/tutorials/whats-new-in-html5-forms-email-url-and-telephone-input-types.html#fbid=e6Bfbp5JZu2>.
- Harris, A. 2012. New Form Input Types in HTML5. Hakupäivä 30.10.2012, <http://www.dummies.com/how-to/content/new-form-input-types-in-html5.html>.
- Hogan, B. 2011. HTML5 and CSS3. Pragmatic Bookshelf.
- Html5test. 2012. The HTML5 test – how well does your browser support HTML5? Hakupäivä 6.10.2012, <http://html5test.com/results/desktop.html>.
- Iso/iec. 2000. Information technology — Document description and processing languages — HyperText Markup Language (HTML). Hakupäivä 1.10.2012, <https://www.cs.tcd.ie/misc/15445/15445.HTML>.
- Moore, A, 2005. Multimedia Web Programming. Palgrave Macmillan.
- Tizag. 2008a. HTML – input element(s). Hakupäivä 10.10.2012, <http://www.tizag.com/htmlT/htmlinput.php>.
- Tizag. 2008b. HTML - web forms. Hakupäivä 10.10.2012, <http://www.tizag.com/htmlT/forms.php>.
- W3C. 1999. HTML 4.01 Specification. Hakupäivä 1.10.2012, <http://www.w3.org/TR/html401/>.
- W3C. 2009. Frequently Asked Questions (FAQ) about the future of XHTML. Hakupäivä 4.10.2012, <http://www.w3.org/2009/06/xhtml-faq>.
- W3C. 2012a. HTML5 differences from HTML4. Hakupäivä 8.10.2012, <http://www.w3.org/TR/2012/WD-html5-diff-20120329/>.
- W3C. 2012b. Introduction to HTML 4. Hakupäivä 1.10.2012, <http://www.w3.org/TR/html401/intro/intro.html#h-2.2.1>.
- W3C. 2012c. W3C Blog. Hakupäivä 6.10.2012, http://www.w3.org/QA/2012/09/getting_html5_to_recommendatio.html.
- W3C wiki. 2012. HTML5 form additions. Hakupäivä 9.10.2012, http://www.w3.org/wiki/HTML5_form_additions.

W3schools. 2012. HTML xhtml. Hakupäivä 4.10.2012,
http://www.w3schools.com/html/html_xhtml.asp.

Whatwg. 2012. HTML living standard. Hakupäivä 1.10.2012, <http://www.whatwg.org/specs/web-apps/current-work/multipage/introduction.html#history-1>.

Wofoo. 2012a. The Current State of HTML5 Forms. Hakupäivä 10.10.2012,
<http://wufoo.com/html5/>.

Wofoo. 2012b. The novalidate Attribute. Hakupäivä 28.11.2012,
<http://www.wufoo.com/html5/attributes/11-novalidate.html>.

LIITTEET

Toiminnallisuus	Prioriteetti
Käyttäjän tulee voida lisätä uusi tuote järjestelmään	1
Käyttäjän tulee voida muokata järjestelmässä olevan tuotteen tietoja (1)	1
Käyttäjän tulee voida poistaa järjestelmässä oleva tuote	3
Käyttäjän tulee voida hakea tuotteita tuotteen tuoteryhmän tai merkin perusteella	1
Käyttäjän tulee voida vähentää tai lisätä tuotteen varastosaldoa tuotenäkymästä	1
Järjestelmän tulee ilmoittaa käyttäjälle, mikäli tuotteen varastosaldo saavuttaa sille asetetun minimikappalemäärän	1
Käyttäjän tulee voida lisätä uusi tuoteryhmä järjestelmään	1
Käyttäjän tulee voida muokata järjestelmässä olevaa tuoteryhmää	1
Käyttäjän tulee voida poistaa järjestelmässä oleva tuoteryhmä	3
Käyttäjän tulee voida lisätä uusi merkki-kategoria järjestelmään	1
Käyttäjän tulee voida muokata järjestelmässä olevaa merkki-kategoriaa	1
Käyttäjän tulee voida poistaa järjestelmässä oleva merkki-kategoria	3
Käyttäjän tulee voida hakea asiakkaita nimen tai puhelinnumeron perusteella	1
Käyttäjän tulee voida muokata järjestelmässä olevia asiakastietoja	2
Käyttäjän tulee voida poistaa järjestelmässä oleva asiakas	3
Käyttäjän tulee voida lisätä tuoterivi takuutodistukseen (2)	1
Järjestelmän tulee vähentää tuotteen varastosaldoa, mikäli tuote merkitään takuutodistukseen	1
Käyttäjän tulee voida tallentaa takuutodistus tietokantaan	1
Järjestelmän tulee tallentaa tietokantaan asiakkaan nimi ja puhelinnumero, mikäli käyttäjä tallentaa takuutodistuksen	1
Käyttäjän tulee voida tulostaa takuutodistus	1
Käyttäjän tulee voida poistaa takuutodistus tietokannasta	3
Käyttäjän tulee voida nähdä yhteenveto hyllyssä myytävien tuotteiden hinnastosta	2
Käyttäjän tulee voida tulostaa hinnasto	2

Taulukossa on kuvattu järjestelmälle asetetut toiminnalliset vaatimukset prioriteetteineen. Käyttäjällä tarkoitetaan tässä yrityksen työntekijää.

(1) Tuotteen tietojen muokkaamisella tarkoitetaan järjestelmässä olevan tuotteen ominaisuuksien (kuvaus, minimikappalemäärä, maksimikappalemäärä, hinta, tilauspaikka, varastolukumäärä, korvaava tuote, takuu, lisätietoja, hyllyssä) muokkaamista.

(2) Tuoterivi sisältää tuotteen kuvauksen, tuoteryhmän, tuotemerkin, kappalemäärän, hinnan ja tuotteen takuuajan.