Opinnäytetyö (AMK)

Tietojenkäsittelyn koulutusohjelma

Tietoliikenne

2014

Henri Turunen

# SUUNNITELMA OMAN JULKAISUJÄRJESTELMÄN KEHITTÄMISEEN

TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietojenkäsittely | Tietoliikenne

Kesäkuu 2014 | 30 sivua

Esko Vainikka

Henri Turunen

# SUUNNITELMA OMAN JULKAISUJÄRJESTELMÄN KEHITTÄMISEEN

Tämän opinnäytetyön tavoitteena on tehdä tekniseen toteutukseen painottuva suunnitelma oman julkaisujärjestelmän ja siihen liittyvän sivuston kehittämiseen. Julkaisujärjestelmällä tarkoitetaan järjestelmää, jolla pystytään helposti hallitsemaan verkkosivustoa selainpohjaisella käyttöliittymällä. Suunnitelman taustalla on erään PK-yrityksen tyytymättömyys heidän tämän hetkiseen järjestelmäänsä, joka on rakennettu avoimen lähdekoodin Wordpress-julkaisujärjestelmän yhteyteen. Uudesta julkaisujärjestelmästä on tarkoitus tehdä vain yrityksen käyttöön räätälöity järjestelmä.

Opinnäytetyön alku koostuu teoriaosuudesta, jossa kerrotaan lyhyesti internetin historiasta ja eri julkaisujärjestelmistä. Tämän jälkeen mietitään syitä siirtyä uuteen järjestelmään ja määritellään, minkälaisia ominaisuuksia siihen tarvitaan. Suunnitelmaa ja määrittelyä varten haastateltiin yrityksen työntekijöitä ja asiakkaita. Lopuksi tarkastellaan oman julkaisujärjestelmän teknisen toteutuksen vaiheet ja mietitään mm. tietoturva-asioita sekä hakukoneoptimointia. Käytettyjä tekniikoita, joita suunnitelmassa otetaan huomioon ovat HTML5, CSS3, MySQL, JavaScript, jQuery, Ajax sekä Smartyn sivupohjajärjestelmä.

Opinnäytetyö sisältää teknisen toteutuksen suunnitelman, joka ottaa huomioon tarvittavan sisällön, vaadittavat tekniikat ja yrityksen sekä asiakkaiden tarpeet.

ASIASANAT:

Julkaisujärjestelmä, Wordpress, HTML-ohjelmointi, CSS3, PHP, Smarty, jQuery

BACHELOR´S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Business Information Technology | Data communications

June 2014 | 30 pages

Esko Vainikka

Henri Turunen

# A PLAN FOR DEVELOPING A COMPANY'S CONTENT MANAGEMENT SYSTEM

The aim of this thesis is to create a plan for developing a content management system. This system is intended to easily manage a web site with a web browser-based user interface. This plan was made for a small company because the employees were unsatisfied with their current system. The company's existing was an open source content management system called Wordpress.

The beginning of this thesis consists of the theoretical part which briefly describes the history of the Internet and the different content management systems. Then, the thesis discusses the reasons for switching to the new system and specifies the features it requires. The company's employees and customers were interviewed for their needs for the plan and their specifications. Finally, the thesis presents all the steps of the technical implementation, security issues, and search engine optimization. The eWeb techniques which were used include HTML5, CSS3, MySQL, JavaScript, jQuery, Ajax and Smarty template system.

As a result a plan was made where all the necessary qualities and technologies for the company's content management system implementation were taken into account.

KEYWORDS:

Template system, Wordpress, HTML programming, CSS3, PHP, Smarty, jQuery

# SISÄLTÖ

# KUVAT



# KUVIOT

# KÄYTETYT LYHENTEET JA SANASTO

| | |
|---|---|
| CSS | Cascading Style Sheets. Tyylitiedosto, jonka säännöillä tehdään verkkosivuston ulkoasua. |
| FTP | File Transfer Protocol. Tiedonsiirtomenetelmä kahden tietokoneen välillä. |
| HTML | Hypertext Markup Language. Avoimesti standardoitu kuvauskieli, joka tunnetaan erityisesti kielenä, jolla internetsivut on koodattu. |
| PHP | Hypertext Preprocessor. Palvelinpuolen komentokieli. |
| SaaS | Software as a Service. Ohjelmiston hankkiminen palveluna perinteisen lisenssipohjaisen tavan sijasta. |
| Selain | Tietokoneohjelma, jolla pystytään katsomaan ja käyttämään WWW-sivuja. |
| SQL | Structured Query Language. IBM:n kehittämä standardoitu kyselykieli, jolla relaatiotietokantaan voi tehdä erilaisia hakuja, muutoksia ja lisäyksiä. |
| URL | Uniform Resource Locator. Käytetään osoittamaan www-sivuja. |
| WWW | World Wide Web. Aliverkkojen muodostama maailmanlaajuinen tietoliikenneverkko, joka koostuu miljoonista tietokoneista. |

# 1 JOHDANTO

Tämä opinnäytetyö on tekniseen toteutukseen painottuva suunnitelma oman julkaisujärjestelmän kehittämisestä eräälle pk-yritykselle, jolla tällä hetkellä on käytössä avoimen lähdekoodin julkaisujärjestelmä Wordpress. Tutkimusote on konstruktiivinen ja tutkimus toteutetaan tapaustutkimuksena.

Lähtökohtana ja ongelmana on, että yrityksen työntekijät ovat tyytymättömiä Wordpressin käyttöliittymään ja sen yhteyteen asennettavien lisäosien epäloogisuuteen. Lisäosat ovat ulkopuolisten tekemiä ohjelmistoja, joilla järjestelmää saa laajennettua ja tuotua siihen lisäominaisuuksia, esimerkiksi gallerian valokuvien näyttämiseen. Lisäosat toimivat harvoin kuitenkaan sellaisenaan, joten niiden lähdekoodeja pitää usein muokata.

Pääsyy uudelle järjestelmälle on sen räätälöitävyys, joten suunnittelussa otetaan huomioon vain ja ainoastaan tämän yrityksen tarpeet ja tarvittavat ominaisuudet. Tavoitteena on tehdä suunnitelma julkaisujärjestelmän ja siihen liittyvän verkkosivuston sisällöstä ja teknisestä toteutuksesta. Tällä järjestelmällä on tarkoitus helppokäyttöisesti internetselaimella hallita verkkosivuston käyttöliittymää, muokata sitä sekä luoda siihen sisältöä.

Julkaisujärjestelmän ja sivuston kehittämistä varten haastatellaan sekä asiakkaita että yrityksen työntekijöitä. Haastattelujen perusteella määritellään suunnitelman sisältö. Teknisen toteutuksen suunnitelmassa kerrotaan järjestelmän ominaisuudet, esimerkiksi vaadittavat ohjelmointikielet, tietoturva-asiat ja hakukoneoptimointi. Työskentelen yrityksessä, joten jatkokehitys järjestelmän valmistuttua tulee olemaan vaivatonta ja sitä voidaan hyödyntää tulevaisuudessa myös muissa vastaavissa kiinnostavissa projekteissa.

# 2 JULKAISUJÄRJESTELMÄT

## 2.1 Verkkosivujen kehitys

Ensimmäinen internetsivu tehtiin vuonna 1990 ja WWW tuli julkiseksi 1993 kaikille. Ensimmäiset selaimet alkoivat nopeasti yleistyä. Netscapen selain julkaistiin vuoden 1994 lopussa ja Internet Explorer vuonna 1995. (The history of the Web.)

Julkaisujärjestelmien historia voidaan lyhyesti jakaa kolmeen eri vaiheeseen. Internetsivujen luontia ja muokkaamista varten kehitettyjä ensimmäisiä julkaisujärjestelmiä alkoi esiintyä 1990-luvun puolivälissä. Käyttöliittymän suhteen pääominaisuudet olivat todella suppeita. Ohjelmiston käytössä piti käyttää tageja ja valmiita ulkoasupohjia, koska esimerkiksi WYSIWYG-editoreita ei ollut. Käyttäjien piti olla tietotaidoiltaan melko osaavia, koska käyttöön tarvittiin ohjelmointitaitoja. Ilman ohjelmointia sivujen muokkaus ei onnistunut. (Contegro 2012.)

Julkaisujärjestelmien historiassa toinen vaihe alkoi noin 2000-luvulta lähtien, jolloin ohjelmistoyritykset alkoivat tehdä järjestelmiä, joihin nykyiset järjestelmät vahvasti perustuvat. Ominaisuudet kehittyivät hitaasti, mutta järjestelmät alkoivat sisältää WYSIWYG-editorin, etsimistoiminnon, parannettua HTML-kieltä ja hyödyllisiä lisätoimintoja, kuten esimerkiksi seurantatyökaluja. Myös pelkästään avoimeen lähdekoodiin perustuvat järjestelmät alkoivat yleistyä. (Contegro 2012.)

Tällä hetkellä eletään siirtymävaihetta, jossa verkkosivujen kehityksen kannalta suunnittelijalla on yhä suurempi rooli ohjelmoijaan verrattuna. Tärkeimpiä ominaisuuksia nykyisissä ja tulevissa julkaisujärjestelmissä tulevat olemaan pilvipalveluiden kautta hallinnoitavat sivustot. Tulevissa suuntauksissa näkyy vahvasti integraatio mm. asiakkuudenhallintajärjestelmien, tietokantojen ja verkkokauppaominaisuuksien kanssa. Myös ns. moduulikehitys yleistyy, jolloin muokattua ohjelmointikoodia tehdään mahdollisimman vähän ja käytetään valmiita

pohjia. Palveluntarjoajien hinnoitteluissa tullaan jatkossa käyttämään paljon SaaS-pohjaista hinnoittelua. (Contegro 2012.)

## 2.2 HTML-ohjelmointi vs. julkaisujärjestelmän käyttö

HTML-ohjelmoinnissa ohjelmoidaan internetsivuja HTML-kielellä käyttäen tekstieditoria, FTP-ohjelmalla tiedostot siirretään palvelimelle ja vasta sen jälkeen muutokset ovat näkyvissä kaikille. Tehtäessä julkaisujärjestelmällä internetsivuja järjestelmä itse päivittää sisällön tietokantaan ja muokkaaminen on helpompaa ja yksinkertaisempaa selainpohjaisten käyttöliittymien ansiosta. Nykyään suurin osa verkkosivuista on entistä dynaamisempia ja uutta materiaalia laitetaan esille usein. Julkaisujärjestelmien etu ja helppous on, että samaa sivustoa pystyvät useat ihmiset päivittämään helposti ja vaivatta.

Ohjelmointitaitoiselle yksinkertaisten sivustojen teko HTML-ohjelmoinnilla on huomattavasti nopeampaa kuin julkaisujärjestelmällä. Tällöin julkaisujärjestelmän valintaan ja asentamiseen ei tarvitse käyttää aikaa vaan voi suoraan aloittaa tekemisen. Valmiita HTML-ulkoasupohjia löytyy ilmaiseksi niihin tarkoitetuilta sivustoilta. Valmiisiin pohjiin voi muokata tarvittavat kuvat ja tekstit. Tällä tavoin esimerkiksi pienen yrityksen verkkosivujen ylläpito toimii hyvin ja yksi henkilö saattaa hoitaa koko verkkosivujen kehityksen. Jos jo alusta alkaen tiedetään, että sivuista tulee melko staattiset, eikä päivityksiä ole juurikaan tulossa, niin on kannattavampaa turvautua HTML-ohjelmointiin.

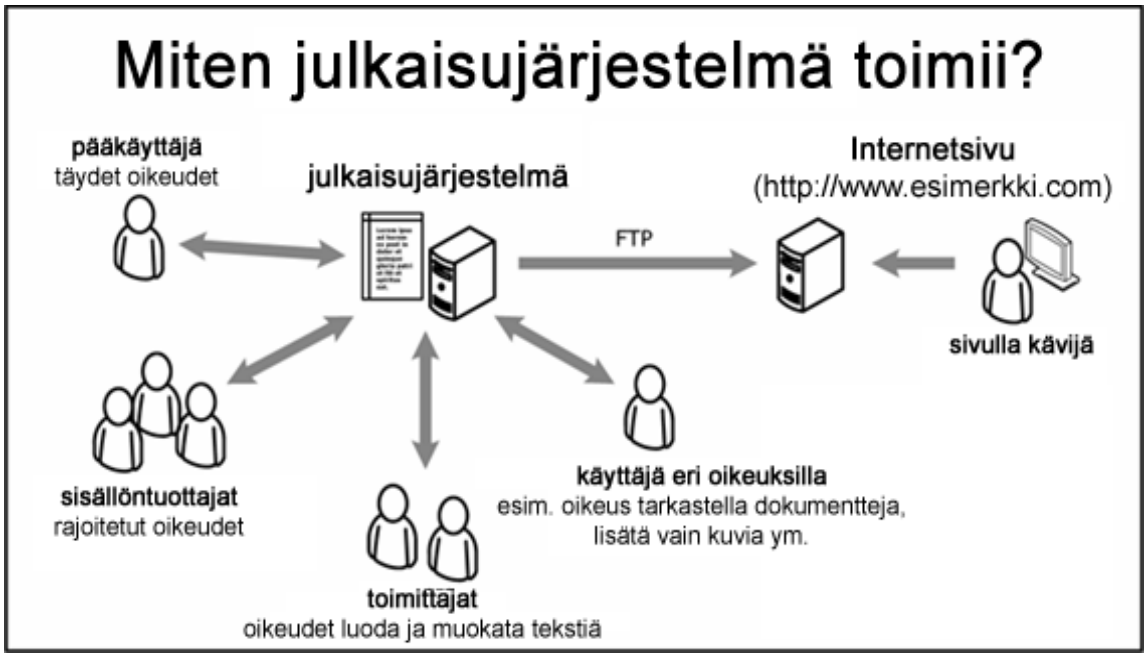
Julkaisujärjestelmä kannattaa ottaa käyttöön, jos verkkosivustosta vastaa useita henkilöitä, joiden pitää päivittää sivustoa, ohjelmointitaitoa ei löydy tai halutaan päästä mistä tahansa muokkaamaan sivustoa. Jos joku ulkopuolinen hallitsee sivustoa, sivuston hallinta ei ole kovin helppoa. Ulkopuoliselle pitää maksaa jokaisesta muutoksesta, esimerkiksi halutessa vaihtaa tiettyä sanaa, lisätä sivuja tai vaihtaa valikoiden paikkaa. Tällaiset muokkaukset onnistuisivat helposti julkaisujärjestelmän avulla.

## 2.3 Julkaisujärjestelmien tarkoitus

Julkaisujärjestelmät ovat ohjelmistoja, joilla pystytään hallitsemaan verkkosivustoa helposti käyttöliittymän avulla. Niillä pystytään esimerkiksi luomaan uutta sisältöä, muokkaamaan vanhoja artikkeleita ja hallitsemaan sivuston toimintoja. (Content management system 2013.) Julkaisujärjestelmät sisältävät työkaluja sisällön tekemiseen niin teksteille, kuville kuin videoillekin. Ne tarjoavat myös mahdollisuuden hallita sivustojen rakennetta, julkaistujen sivujen ulkoasua ja navigaatiovaihtoehtoja.

Julkaisujärjestelmien tarkoituksena on helpottaa ja nopeuttaa sivustojen hallitsemista sekä tehdä niistä joustavampia. Ne auttavat sellaisia sisällönluojia, jotka eivät osaa ohjelmoida internetsivuja. Suurin osa julkaisujärjestelmistä sisältää selainpohjaisen graafisen käyttöliittymän, jota pääsee hallitsemaan millä tahansa internetselaimella.

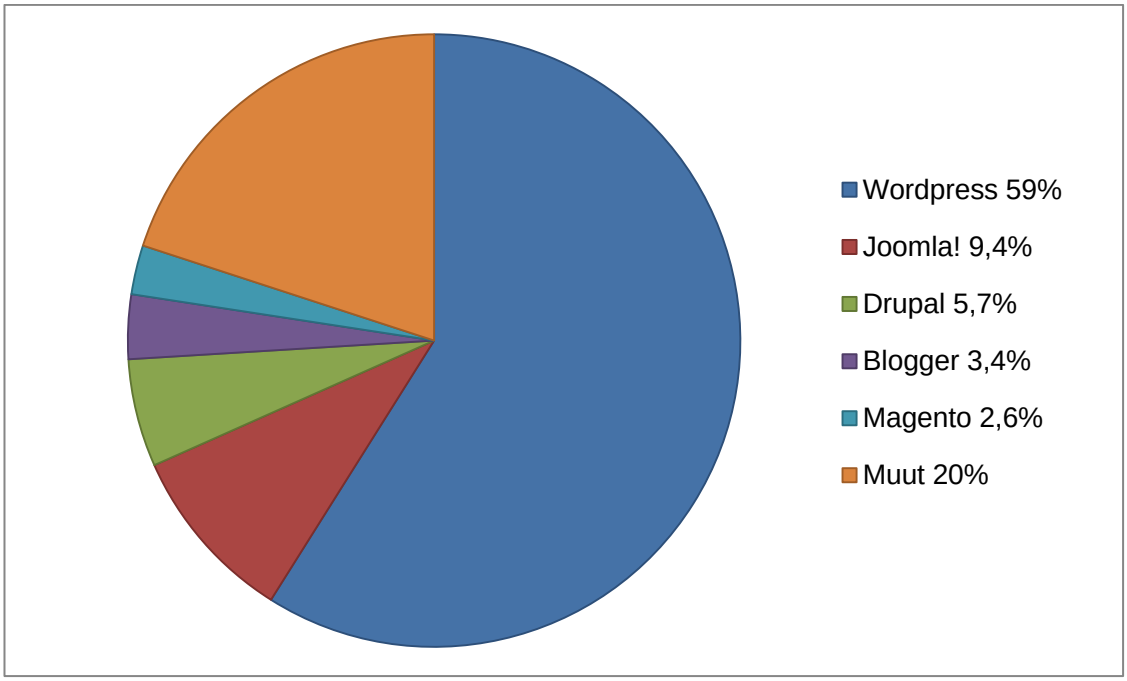
Kuvassa 1 havainnollistetaan kuinka useat eri käyttäjät pystyvät ottamaan osaa sivuston kehitykseen ja ylläpitoon. Käyttäjillä on usein määriteltynä eri oikeustasoja riippuen tehtävästä ja osaamistasosta. Suurin osa nykyisistä julkaisujärjestelmistä tukee eri oikeustasojen antamista ja usein vain pääkäyttäjällä on oikeuksia tehdä sivustolla mitä tahansa. Kuviosta huomataan, kuinka julkaisujärjestelmä hoitaa kaiken työn, jotta sivusto siirtyy sivulla kävijän nähtäväksi. Julkaisujärjestelmä useimmiten joko lataa tiedot FTP:n välityksellä palvelimelle tai tallentaa uudet tiedot omaan tietokantaansa. (Content management system 2013.)




Kuva 1. Esimerkki julkaisujärjestelmän toiminnasta.

## 2.4 Eri julkaisujärjestelmät lyhyesti

Tunnettuja julkaisujärjestelmiä on olemassa satoja. Pääpiirteiltään ne toimivat samankaltaisesti ja useimmiten niillä pääsee samaan lopputulokseen. Wordpress on suosituin avoimien lähdekoodien julkaisujärjestelmissä ja kaikista internetin sivuista 17 % on tehty Wordpressillä (Usage of content management systems for websites 2013). Wordpress hallitsee ylivoimaisesti avoimen lähdekoodin julkaisujärjestelmiä käyttäjien määrässä mitattuna (Kuvio 1).




Kuvio 1. Julkaisujärjestelmien käytön jakautuminen (Usage of content management systems for websites 2013).

Muista käytetyimmistä järjestelmistä melkein kaikki soveltuvat tavanomaisen verkkosivuston hallintaan, mutta ne sisältävät joitakin erityispiirteitä. Joomla soveltuu hyvin verkkokauppojen ja sosiaalisten verkkosivujen tekoon, mutta se vaatii hieman enemmän tietotaitoa kuin Wordpressin käyttö. Drupal on ehkä hankalin oppia, mutta se on suorituskyvyltään nopea, eikä vaadi taustalla olevalta palvelimelta paljoa suorityskykyä. Blogger on Googlen kehittämä ilmainen työkalu blogien tekemiseen. Magento on avoimen lähdekoodin verkkokauppajärjestelmä ja siitä on saatavilla sekä ilmainen että maksullinen versio.

## 2.5 Wordpress

Wordpress on ilmainen avoimen lähdekoodin järjestelmä, joka on suunniteltu pääasiassa blogeja varten. Kehittämisprojekti aloitettiin vuonna 2005 ja sitä johtaa Automattic-niminen yritys. Wordpress on BuiltWithin mukaan maailman kaikkien aikojen suosituin blogijärjestelmä. Vaikkakin Wordpressiä käytetään paljon blogeihin, on se kehittynyt vuosien varrella paljon ja nykyään sitä käyte-

tään kaikenlaisten verkkosivujen tekoon. Varsinkin version 3.0 jälkeen Wordpressiä voidaan pitää täyspainoisena julkaisujärjestelmänä. Yksi tärkeimmistä syistä Wordpressin menestyksen taustalla on erityisen aktiivinen kolmansien osapuolien osallistuminen sen kehittämiseen. Kolmannet osapuolet ovat erittäin aktiivisia tekemään lisäosia ja ulkoasuteemoja, joilla saadaan uniikkia ilmettä ja toiminnallisuutta. Tällä hetkellä ilmaisia lisäosia on yli 11000 ja teemoja 1200. Lisäksi sille löytyy valtava määrä kaupallisia lisäosia. (Pearche 2011, 215).

Wordpressin hallintapaneelin etusivun kautta hoidetaan sivuston artikkelien ja sivujen muokkaus, blogikommenttien hyväksyntä, asetukset, lisäosien asentaminen ja kaikki, mitä hallintaan liittyy (Kuva 2). Näkymä on kaikissa Wordpress-asennuksissa pääasiassa samanlainen, mutta se saattaa hieman poiketa riippuen käyttäjän oikeuksista.


Kuva 2. Wordpress-hallintapaneelin etusivu

# 3 OMAN JULKAISUJÄRJESTELMÄN MÄÄRITTELY JA SUUNNITTELU

## 3.1 Syyt omaan julkaisujärjestelmään

Miksi lähteä tekemään omaa julkaisujärjestelmää, koska valmiita järjestelmiä on valtavasti olemassa ja varmasti niistä löytäisi omansa? Kaupallista tai avoimen lähdekoodin julkaisujärjestelmää valittaessa herää paljon kysymyksiä, kuten mitä se maksaa, onko se helppokäyttöinen ja onko siinä tarvittavia tai turhia ominaisuuksia?

Suurin syy omaan järjestelmään on sen täysi hallittavuus. Siitä ei tarvitse tehdä tarpeettoman laajaa vaan toteutetaan vain ne ominaisuudet, joita tarvitaan. Kohdeyrityksessä, jossa päätettiin siirtyä omaan järjestelmään, yhdeksi syyksi muodostui nykyisen sivuston hallinnan kannalta tarpeellisten lisäosien sekavuus ja epäloogisuus.

Wordpressiä on helppo käyttää ilman lisäosia, mutta pakolliset lisäosat ovat käyttäjien kannalta vaikeakäyttöisiä. Wordpressiin saatavilla oleva lisäosien määrä on valtava, mutta niiden laatu, ohjelmoinnin selkeys ja tuki vaihtelevat. Kaikki lisäosat eivät tue uuteen Wordpress-versioon päivittämistä. Yhtenä vaihtoehtona näissä tilanteissa on muokata lisäosien lähdekoodia, mutta tällöin on otettava huomioon, että muut ominaisuudet eivät lakkaa toimimasta. Wordpress ei täysin tue hakukoneoptimointia, jolloin sitäkin varten pitää etsiä lisäosa, joka muokkaa ominaisuudet hakukoneystävällisemmiksi. Myös käyttäjätilien oikeuksien kokonaisvaltainen hallinta vaatii lisäosan.

Yhdeksi ongelmaksi muodostui työntekijöiden tietoteknisten taitojen vähäisyys. Yritys ei ole suuri, joten jokaisen työntekijän pitää osata päivittää verkkosivuille ainakin kuvastoja, uutisia, uusia tuotteita ja mainoskampanjoita. Nykyinen sekava järjestelmä ei tee verkkosivujen päivittämistä työntekijöille helpoksi ja loogiseksi.

## 3.2 Asiakkaiden ja yrityksen työntekijöiden haastattelut

Tarkoituksena on tehdä hyvä julkaisujärjestelmä ja verkkosivusto sekä työntekijöiden että asiakkaiden näkökulmista. Päätettiin tehdä kaksi haastattelua, henkilökunnalle liittyen tämänhetkiseen julkaisujärjestelmään ja asiakkaille liittyen nykyiseen internetsivustoon. Haastatteluiden tulokset muokkaavat sivuston toiminnallisuuden ja ulkoasun sopiviksi.

Asiakaslähtöisyys on tärkeää verkkosivuston ulkoasun suunnittelu- ja toteutusvaiheissa. Tästä syystä päätettiin tehdä pienimuotoiset haastattelut yrityksen henkilökunnan lisäksi myös sen asiakkaille. Asiakaskysely toteutettiin puhelinhaastatteluilla ja siihen valittiin 15 yrityksen tärkeää asiakasta. Haastattelut tuottivat olemassa oleviin ongelmiin hyviä ratkaisuja.

Haastatteluissa oli tarkoitus hieman johdattaa asiakasta hänen mielestään oikeaan vastaukseen. Kiinnostava tieto ei ole, onko verkkosivusto hyvä vaan lähinnä minkä takia se on hyvä. Esimerkiksi onko verkkosivusto looginen, millä tavalla tuotteet on lajiteltu, onko mahdollista hakea kategorian ja brändin mukaan ja millä tavalla tilaukset halutaan tehdä.

Yrityksen työntekijät kertoivat nykyiseen julkaisujärjestelmään liittyvistä ongelmista ja esittivät toivomuksia uuden järjestelmän ominaisuuksista.

## 3.3 Haastattelujen tuloksia

Asiakkaiden haastatteluissa tärkeimpinä asioina nousi esille yksinkertainen, looginen toimivuus ja varastosaldojen näkyminen. Moni toivoi saldoon myös nähtäväksi tuotteen saapumispäivän, jos sitä ei sillä hetkellä ollut saatavilla. Lisäpohdittavaa antoivat ideat, että tuotteita voisi laittaa varaukseen, mutta niitä ei tarvitsisi heti tilata. Muutama asiakas koki suosikkitavaroiden listauksen hyvänä ominaisuutena.

Asiakaskartoituksen tulokset antoivat uusia näkökulmia sivuston käytettävyydestä ja ulkoasusta. Moni kaipaa yksinkertaista toimivuutta ja ulkonäköä, mutta

osa kaipaa myös tarkkoja tuotetietoja, kuvastojen sekä ohjeiden lataamista ja erilaisia tilauksensyöttövaihtoehtoja.

Sujuvaa sivuston käyttöä toivottiin mobiliilaitteilla. Erityisesti asiakkaat haluaisivat selata tuotetekstejä helposti ilman ylimääräistä näytön kallistelua ja zoomausta. Tätä varten sivustolle luodaan responsiivinen ulkoasu. Sillä tarkoitetaan ulkoasua, joka skaalautuu näytön koon mukaan. Tällöin ei tarvitse luoda erikseen mobiilisivustoa, vaan sama sivu toimii joka laitteella ja käyttökokemus ei kärsi.

Sisäisessä yrityshaastattelussa huomattiin, että projektista syntyi paljon erilaisia ajatuksia toteutusta varten. Työntekijät toivoivat esimerkiksi valmiita sisältöalueita, joita pystyy helposti muokkaamaan. Sisältöalueen muokkauksessa jokaiselle sisältöalueelle voidaan sijoittaa yksi komponenttilohko. Komponenttilohko sisältää erilaisia komponentteja, joista sivusisältö rakentuu. Tätä varten luodaan dynaamiset URL-osoitteet helpottamaan komponenttilohkojen ja sivuston hallintaa. Tällöin mikä tahansa verkkosivuston sivu sisältää ulkoasupohjan, johon ylläpitäjä voi laittaa haluamaansa sisältöä. Komponenttialueiden ja dynaamisten URL-osoitteiden ansiosta voidaan luopua tarpeettomasta hallintapaneelista, joka on Wordpressissä käytössä. Tämä tekee julkaisujärjestelmän käytöstä henkilökunnalle huomattavasti loogisempaa.

### 3.4 Omalle julkaisujärjestelmälle tulevan sisällön suunnitelma

Asiakas- ja työntekijähaastattelut tuottivat hyviä uusia ominaisuuksia omaan julkaisujärjestelmään:

- Pikahaku. Hakutuloksissa tuotekuva, -numero, -nimi ja linkki tuotteen omalle sivulle.
- Yksi hakutulos ohjaa suoraan kyseiselle sivulle.
- Suosittuja hakusanoja kuukauden ajalta.
- Responsiivinen ulkoasu sivustosta.
- Sivusisällön muokkaus sivuille määriteltyjen sisältöalueiden kohdalla.

- Ylläpitäjänä kaikki sivuston URL-osoitteet ovat dynaamisia.
- Wysiwyg-editori.
- Parempi näkyvyys hakukoneessa.
- Sivuston muokkaaminen ilman erillistä hallintapaneelia.

Nykyiset ominaisuudet, jotka sisällytetään uuteen järjestelmään:

- Etusivu. Sisältää uusimpia tarjouksia ja tuotteita.
- Oma sivu uutisille. Uusimmat uutiset näkyvät myös etusivulla.
- Navigaatioelementit aina samassa paikassa ylhäällä ja alhaalla.
- Kuvastot ja tuotteet omilla sivuillaan kategorian mukaan.
- Tarjoukset ja kampanjat omilla sivuilla.
- Aukioloajat ja yhteystiedot.
- Yhteydenottolomake.
- Rekisteriseloste.
- Sijainnin näyttäminen Googlen karttapalvelussa.
- Googlen analytiikka koko sivustolla.

# 4 OMAN JULKAISUJÄRJESTELMÄN TEKNISEN TOTEUTUKSEN SUUNNITELMA

## 4.1 Palvelin ja tietokanta

Uutta järjestelmää tullaan todennäköisesti ajamaan Linux-käyttöjärjestelmällä ja Apache-palvelimella. Tietokantana tullaan käyttämään MySQL:ää, joka on maailman toiseksi suosituin avoimen lähdekoodin tietokanta (DB-Engines 2014). Myös Linux ja Apache ovat avointa lähdekoodia, joten apua ja teknistä tukea saa helposti esimerkiksi erilaisilta keskustelufoorumeilta.

## 4.2 Sivupohjajärjestelmä

Oma julkaisujärjestelmä voitaisiin toteuttaa yleisellä HTML-PHP-yhdistelmällä, mutta omasta mielenkiinnostani ja jatkokehitystä varten haluan ottaa siihen mukaan Smartyn sivupohjajärjestelmän, jonka päätarkoituksena on erottaa HTML ja PHP toisistaan. Tämä on hyödyllistä, sillä käytännössä se erottaa toimintalogiikan esityslogiikasta, jolloin koodi on huomattavasti selkeämmän näköistä kummassakin päässä. Sivupohjajärjestelmän hyödyllisyys korostuu varsinkin, jos sovelluksen ohjelmoinnista ja vastaavasti käyttöliittymäsuunnittelusta vastaa useampi henkilö. (Smarty 2014.)

Sivupohjajärjestelmää on hyvä havainnollistaa käytännön tilanteessa, esimerkiksi jos ollaan luomassa sivua tietystä tuotteesta, jolloin tuotteen nimi, hinta ja tuoteseloste johdetaan sivupohjajärjestelmän avulla. Sen jälkeen käyttöliittymäsuunnittelija muokkaa ulkoasua käyttämällä ja yhdistelemällä HTML-kieltä ja sivupohjajärjestelmän elementtejä, esimerkiksi taulukoita, taustavärejä, fontin kokoa ja erilaisia tyylimäärityksiä. Tällöin toimintalogiikalla tuodaan tarvittavat tiedot ja käyttöliittymäsuunnittelija laittaa valmiit muuttujat haluamiinsa kohtiin ja ulkoasuun sopivaksi. (Smarty 2014.)

## 4.3 Smarty

Tähän projektiin otetaan käyttöön suosittu Smartyn sivupohjajärjestelmä. Smarty poikkeaa useista muista sivupohjajärjestelmistä siten, että se tarkastaa aina sivupohjaa lukiessaan, onko siitä saatavilla tuore versio välimuistissa. Jos versiota ei löydy, Smarty kääntää sivupohjan puhtaaksi PHP-koodiksi ja sijoittaa sen omaan hakemistoon, josta se voidaan nopeasti hakea. Tuoreen version löytyessä Smarty käynnistää sen heti. Smarty toimii siis erittäin nopeasti, koska PHP:n ei tarvitse suorittaa lukuisia etsi-korvaa-toimintoja sijoittaessaan sisältöä muuttujiin. Myös ohjausrakenteet, kuten toisto ja ehtorakenteet, kääntyvät optimoiduksi PHP-koodiksi.

Esimerkissä (Kuva 3) yhteystiedot tuodaan html-pohjaan muuttujilla, esimerkiksi $contacts.fax. Smarty-tagit ovat oletusarvoisesti { ja } -merkkien sisällä olevia merkintöjä. Muuttujien tuonti käsitellään sovelluslogiikassa (Kuva 4). Ne voitaisiin esimerkiksi tuoda tietystä tietokannasta tai asiakkuudenhallinnan rajapinnasta. Lopputuloksena (Kuva 5) tiedot on tuotu html-pohjaan.

```
{$Contacts.fax}<br />
{$Contacts.email}<br />
{$Contacts.phone.home}<br />
{$Contacts.phone.cell}<br />
```

Kuva 3. Smarty-esimerkki esityslogiikasta

```
<?php
$smarty->assign('Contacts',
    array('fax' => '555-222-9876',
          'email' => 'zaphod@slartibartfast.example.com',
          'phone' => array('home' => '555-444-3333',
                           'cell' => '555-111-1234')
                          )
         );
$smarty->display('index.tpl');
?>
```

Kuva 4. Smarty-esimerkki sovelluslogiikasta.

```
555-222-9876<br />
zaphod@slartibartfast.example.com<br />
555-444-3333<br />
555-111-1234<br />
```

Kuva 5. Smarty-esimerkki lopullisesta tulostuksesta html-pohjaan.

## 4.4 Selaintekniikat

Uuden julkaisujärjestelmän ja sivuston perusrunko rakennetaan käyttäen HTML5:tä ja CSS3:a. HTML5 on HTML-ohjelmointikielen uusin versio, jolla esitetään, millä tavalla verkkosivu rakentuu. HTML on lyhenne sanoista Hypertext Markup Language. Sillä voidaan esimerkiksi merkitä, mikä sivun tekstistä on otsikkoa tai leipätekstiä. Julkaisujärjestelmän editointipainikkeet ja sivuston navigaatioelementit tehdään hyödyntäen HTML:ää ja CSS:ää.

CSS on lyhenne sanoista Cascading Style Sheets. Sillä annetaan sääntöjä, jotka internetselain tulkitsee ja määrittää, miltä HTML-tiedoston rakenne ja elementit näyttävät. Sillä voidaan esimerkiksi määrittää leipätekstin fontin koko, väritys ja tasaus. Kuvassa 6 vasemmalla puolella on HTML ja CSS-määrityksiä ja oikealla puolella voi nähdä, miltä se selaimelta katsottuna näyttäisi. CSS-

määrityksiä pystyy käyttämään suoraan HTML-sivustossa style-tagien väliin laitettuna, mutta tämä ei ole suositeltavaa. CSS-tiedosto kannattaa ladata aina erillisestä tiedostostaan, jolloin sitä pystyy käyttämään monella sivulla helposti. Uudessa julkaisujärjestelmässä responsiivinen ulkoasu tullaan luomaan CSS:n avulla. Responsiivisuuden toteuttamiseksi käytetään CSS:n media queryjä. Niiden avulla pystytään esimerkiksi tunnistamaan, minkä kokoinen käyttäjän selaimen ikkuna on ja sitä kautta tekemään sääntöjä sivuston ulkoasuun.

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>Testisivu</title>

<style type="text/css">
    h1 { color:#F60; text-decoration:underline; }
    p { font-size:14px}
</style>

</head>

<body>
    <h1>Testiotsikko</h1>
    <p>Tässä on leipätekstiä</p>
    <h2>Pienempi otsikko</h2>
    <p>Taas leipätekstiä</p>
</body>
</html>
```

Testiotsikko

Tässä on leipätekstiä

Pienempi otsikko

Taas leipätekstiä

Kuva 6. HTML-esimerkkisivu. Vasemmalla puolella HTML ja CSS-määrityksiä ja oikealla puolella, miltä se selaimella katsottuna näyttää.

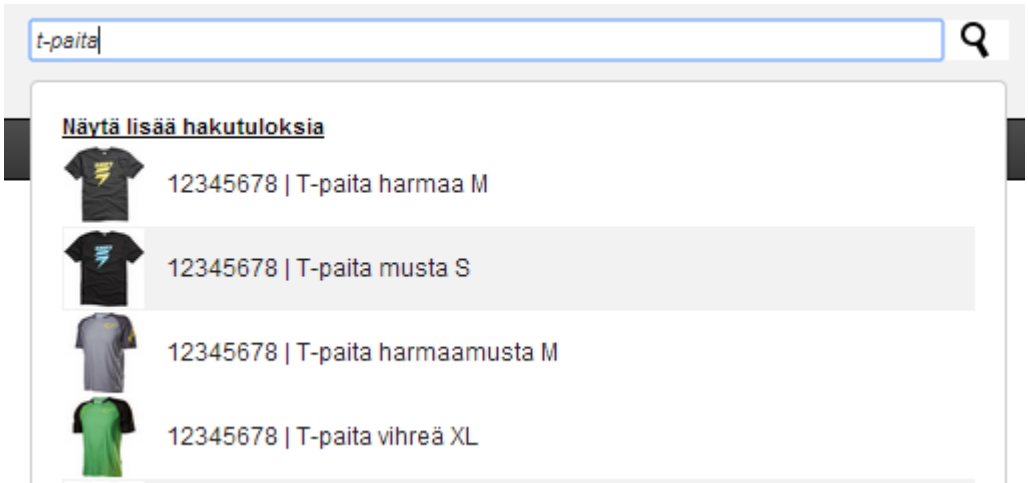
Kaikkia uuden järjestelmän vaatimia ominaisuuksia ja tehtäviä ei pystytä hoitamaan pelkästään palvelimella tai HTML:n ja CSS:n avulla. Niiden lisäksi tarvitaan myös selaimella käytettäviä erilaisia tekniikoita, kuten Javasciptiä, jQuerya ja Ajaxia.

JavaScript on alun perin Netscape Communications Corporationin kehittämä käyttöjärjestelmäriippumaton oliopohjainen ohjelmointikieli. Javascript mahdollistaa dynaamisten toimintojen lisäämisen verkkosivuille. Sillä voi esimerkiksi tehdä lomakkeen tietojen esitarkastuksen, erilaisten välilehtien toteutuksen, animaatioita, uusia tyylimäärityksiä ja evästeiden hallintaa. Javascriptiä ei pidä sekoittaa Javaan, vaikka kielissä onkin yhtäläisyyksiä. Javascriptiä kutsutaan

skriptiksi, koska se tulkitaan suoraan käyttäjän selaimessa, eikä siitä käännetä omaa ohjelmaa. (Javascript introduction 2014.)

jQueryä tullaan käyttämään Javascriptiä enemmän. Se on Javascript-kirjasto, jonka tarkoituksena on helpottaa ja nopeuttaa javascriptin kirjoittamista. jQuery yhdistää Javascript-toimintoja ja sen avulla voi yhdellä rivillä koodia tehdä toimintoja, jotka Javascriptillä yleensä vaatisivat useita rivejä. jQuery sisältää tuen myös AJAXin käytölle. (Jquery introduction 2014)

Yleensä on totuttu siihen, että internetsivun sisällön muuttaminen vaatii sivun uudelleen lataamisen palvelimelta. Aina tällainen ei ole toivottua ja monesti on myös turhaa hakea verkosta koko sivua uudelleen, jos muutokset koskevat vain pientä osaa sivusta. AJAX-tekniikalla selainohjelma vaihtaa dataa palvelimen kanssa taustalla niin, ettei koko verkkosivua tarvitse ladata uudelleen käyttäjän tehdessä muutoksia. Omassa julkaisujärjestelmässä esimerkiksi hakukentän toiminto tullaan toteuttamaan AJAXin ja jQueryn avulla. Käyttäjän syöttäessä hakusanaa, joka näppäimen lyönti tunnistetaan jQueryllä ja AJAXin avulla haetaan tietokannasta hakukentässä olevia kirjaimia vastaavia tuotteita. Tuotteen löytyessä, siitä näytetään heti hakukentän alapuolella tuotekuva, -numero ja -nimi (Kuva 7). Käyttäjän kannalta tämä on hyödyllistä, sillä sivua ei tarvitse ladata uudelleen tai mennä erilliselle hakusivulle.

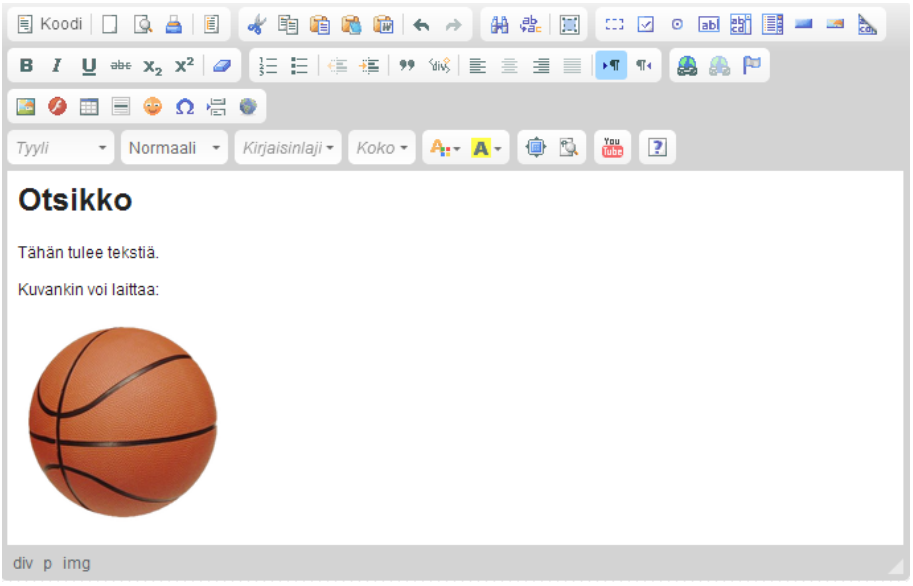



Kuva 7. Hakutulokset näytetään heti AJAXin avulla.

## 4.5 Wysiwyg-editori

Wysiwyg on lyhenne englanninkielen sanoista: What You See Is What You Get. Wysiwyg-editorit muistuttavat ulkoasullisesti paljon yksinkertaisia tekstieditoreita, esimerkikiksi Windowsin Wordpadia. Verkkosivustojen yhteydessä Wysiwyg-editorilla tarkoitetaan sellaista ohjelmistoa, jonka sisältö muokattaessa näyttää mahdollisimman paljon lopulliselta ulkoasulta tallentamisen jälkeen. Wysiwyg-editori siis luo HTML-koodia käyttäjän tekemistä teksti- ja kuvamuokkauksista. Editorissa voi muuttaa fonttien väritystä, tasausta ja kokoa. Suorien linkkien teko onnistuu yleensä yhdellä napin painalluksella ja suurimmassa osassa editoreja onnistuu myös kuvien lisäys suoraan omalta koneelta. Wysiwyg-editorin käyttö on helppoa eikä se vaadi ohjelmointiosaamista.

Tätä projektia varten on tärkeää saada Wysiwyg-editori käyttöön, jotta haluttujen tekstien ja ulkoasujen muokkaaminen onnistuisi ketterästi keneltä tahansa yrityksessä. Ilmaisia Wysiwyg-editoreja on valtava määrä. Tähän projektiin valitaan Ckeditor (Kuva 8), joka on yksi käytetyimmistä editoreista. Olen todennut sen eritysen toimivaksi ja luotettavaksi aikaisemmissa projekteissa.




Kuva 8. Ckeditor on Wysiwyg-editori.

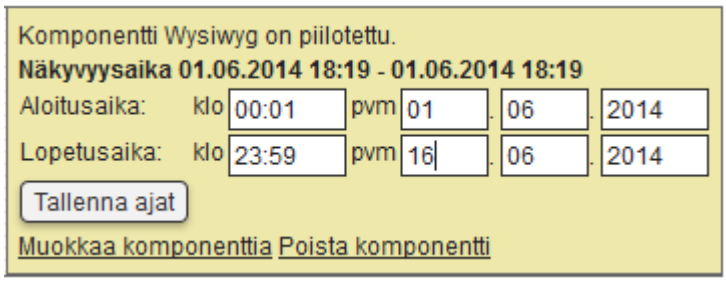
## 4.6 Komponentit

Oman julkaisujärjestelmän ideana on mahdollisimman helppo ja tehokas käyttö, joten tätä varten kehitetään erilaisten komponenttien hyödyntäminen. Varsinaista hallintapaneelia omaan järjestelmään ei tule vaan ylläpitäjäksi kirjautuneena, jokaista sivuston sivua pystyy muokkaaman suoraan.

Wysiwyg-editorilla luodut sisällöt tallennetaan komponentteihin. Komponentteja pystytään tuomaan mille tahansa sivulle valmiiksi luotuihin komponenttialueisiin (Kuva 9). Esimerkiksi, jos tietty mainoslause halutaan näkyväksi usealle sivulla, voidaan samaa komponenttia käyttää usealla eri sivulla. Muutostarpeen ilmetessä muokkausta ei tarvitse tehdä kuin yhteen komponenttiin, jolloin se päivittyy kaikille sivuille, joissa se on käytössä.


Kuva 9. Komponenttialueet.

Komponenttien hallintaa helpottamaan tehdään myös ajastus (Kuva 10), jolloin voidaan määrittää se aika, milloin komponentti tulee ja poistuu näkyvistä. Esimerkiksi tietyt aikaan sidotut kampanjat voi laittaa keskiyöllä alkamaan päivän vaihtuessa. Tämä helpottaa yrityksen henkilökunnan työskentelyä, eikä heidän tarvitse olla tietokoneen äärellä normaalin työajan ulkopuolella.




Kuva 10. Komponenttien ajastus.

## 4.7 Hakukoneoptimointi

Kohdeyrityksen näkyvyyden ja kaupallisuuden kannalta on hyötyä näkyä hakukoneissa mahdollisimman hyvin. Hakukoneiden kannalta on tärkeää saada julkaisujärjestelmä tuottamaan mahdollisimman optimoitua koodia. Tärkeää on esimerkiksi, että HTML-otsikoissa on sivun tärkeimpänä otsikkona h1-tagi. Leipätekstit laitetaan p-tagien väliin ja eri sivujen urlit tulevat selkokielisenä ja sisällön kannalta tärkeänä. Joka sivulle ja kuvalle laitetaan omat kuvaukset ja otsikot.

Hakukoneet laskevat arvostusta, jos sivuilta löytyy paljon identtistä sisältöä. Esimerkiksi tietyt tuotekategorioiden sivut saattavat sijaita kahdessa tai useammassa kohdassa tuotepuussa, jolloin on hyvä, että niille annetaan ns. kanonisointi-url. Tällöin hakukoneet eivät laske sisältöä samanlaiseksi, vaan ottavat vain kanonisoidun urlin huomioon.

Sivuston nopeus ja koodin oikeellisuus vaikuttavat myös jonkin verran hakukoneissa näkyvyyteen. Järjestelmän toteutusvaiheessa huomioidaan välimuistin käyttö eli kaikki staattiset elementit tallennetaan palvelimen välimuistiin. Tällöin ei esimerkiksi suoriteta tietokantakyselyitä, jotka hidastavat sivuston käyttöä vaan näytetään suoraan staattinen html-sivu. Koodi tullaan tarkastamaan myös siihen tarkoitetulla W3-validaattorilla, joka käy läpi ohjelmointikielen oikeellisuuden ja virheet.

On tärkeää korostaa henkilökunnalle ja sisällönluojille hakukoneoptimoinnin hyödyllisyyttä. Kannattaa esimerkiksi luoda yksittäisestä tärkeästä aiheesta tai tuotteesta kokonaan oma sivunsa ja oma kuvaava otsikkonsa. Hakukoneet arvostat näitä ns. laskeutumissivuja enemmän kuin sellaisia, joissa on paljon erinäköistä sisältöä. Optimoinnin kannalta on myös tärkeää, että teksisisältöä löytyy ja se vastaa ainoastaan sillä sivulla olevia asioita.

## 4.8 Tietoturva

Omassa järjestelmässä on tietoturvan kannalta hyviä ja huonoja puolia. Tunnettuja avoimen lähdekoodin järjestelmiä kohtaan suunnataan valtavasti massahyökkäyksiä ja hyödynnetään tunnettuja haavoittuvuuksia. Omassa järjestelmässä näistä ei ole ole vaaraa, mutta on vaarana joutua hyökkäyksen kohteeksi esimerkiksi ohjelmointivirheen takia. Ohjelmointivirheet ovat useimmiten suurin syy tietoturvauhille. Ohjelmoitaessa on varmistuttava, että ohjelmakoodi on validia eikä ristiriitaisuuksia synny.

Omaa järjestelmää tehtäessä on erityisen tärkeää huolehtia, että tietoturva-asiat ovat kunnossa. On syytä tarkistaa säännöllisesti, että esimerkiksi Smartystä, jQuerystä ja Wysiwyg-editorista on uusimmat suositellut versiot käytössä, käyttäjille annetaan vain tarvittavat käyttöoikeudet ja varmuuskopioita järjestelmästä otetaan säännöllisesti. Käytettävien ohjelmistojen tunnetut haavoittuvuudet kannattaa ehdottomasti tarkastaa kansainvälisestä haavoittuvuustietokannasta käyttäen sen hakukonetta (NIST 2014).

OWASP on avoimen lähdekoodin projekti, joka keskittyy internetohjelmistojen tietoturvaan. Se on koonnut hyödyllisen listauksen kymmenestä yleisimmästä tietoturvauhasta (Kuva 11). Nämä kaikki uhat tullaan huomioimaan ja varmistutaan esimerkiksi, että SQL-injektioita ei ole mahdollista tehdä sivuston tekstikenttiä hyödyntäen. Uhkien läpikäyntiin saa erinomaisen ohjeistuksen OWASP Top Ten 2013 projektin www-sivustolta (OWASP 2014).

| OWASP Top 10 – 2010 (Previous) | OWASP Top 10 – 2013 (New) |
|---|---|
| A1 – Injection | A1 – Injection |
| A3 – Broken Authentication and Session Management | A2 – Broken Authentication and Session Management |
| A2 – Cross-Site Scripting (XSS) | A3 – Cross-Site Scripting (XSS) |
| A4 – Insecure Direct Object References | A4 – Insecure Direct Object References |
| A6 – Security Misconfiguration | A5 – Security Misconfiguration |
| A7 – Insecure Cryptographic Storage – Merged with A9 → | A6 – Sensitive Data Exposure |
| A8 – Failure to Restrict URL Access – Broadened into → | A7 – Missing Function Level Access Control |
| A5 – Cross-Site Request Forgery (CSRF) | A8 – Cross-Site Request Forgery (CSRF) |
| <buried in A6: Security Misconfiguration> | A9 – Using Known Vulnerable Components |
| A10 – Unvalidated Redirects and Forwards | A10 – Unvalidated Redirects and Forwards |
| A9 – Insufficient Transport Layer Protection | Merged with 2010-A7 into new 2013-A6 |

Kuva 11. OWASP:n määrittelemät kymmenen tietoturvauhkaa vuosilta 2010 ja 2013 (Zer0byte 2014).

Tietoturvaongelmat eivät johdu pelkästään ulkoisista uhista vaan myös sisäiset uhat pitää huomioida. Erilaisten käyttöoikeuksien määrittely kannattaa tehdä ja miettiä, millaisia oikeuksia kullekin käyttäjälle annetaan. Tällöin vältytään esimerkiksi vahingossa tehdyiltä muokkauksilta tai poistoilta sisällön tai käyttöliittymän suhteen. Käyttöoikeuksia tullaan rajamaan työtehtävien ja tietotaidon mukaan.

# 5 YHTEENVETO

Opinnäytetyön tavoitteena oli tehdä suunnitelma oman julkaisujärjestelmän ja siihen liittyvän sivuston sisällöstä ja teknisestä toteutuksesta. Tarkoituksena oli, että uudella julkaisujärjestelmällä pystyy helppokäyttöisesti internetselaimella hallitsemaan verkkosivuston sisältöä ja käyttöliittymää. Tarve opinnäytetyöhön tuli yrityksen kautta, jossa oltiin tyytymättömiä heidän tämän hetkiseen Wordpress-sivustonsa epäloogiseen hallintaan. Suunnitelmaa varten haastattelin yrityksen henkilökuntaa ja sen asiakkaita. Haastattelu oli hyödyllistä tehdä, sillä sen avulla sain toiveet ja tarpeet määriteltyä. Määrittelyä silmällä pitäen huomioin tarvittavat selaintekniikat.

Opinnäytetyö sisältää teknisen toteutuksen suunnitelman, joka ottaa huomioon tarvittavan sisällön, vaadittavat tekniikat ja yrityksen sekä asiakkaiden tarpeet. Suunnitelmaa apuna käyttäen julkaisujärjestelmä on jatkossa helppo toteuttaa ja jatkokehittää. Valituissa selaintekniikoissa huomioin myös työn tehokkuuden. Esimerkiksi käyttämällä jQueryä saan Javascriptin tekemisestä ja käyttämisestä huomattavasti nopeampaa. Myös Smartyn sivupohjajärjestelmän hyödyntäminen tulee jatkokehityksen kannalta olemaan järkevää ja tehokasta. Ohjelmoinnin määrä tulee kasvamaan järjestelmää kehitettäessä, joten Smartyn avulla voin helposti itse ottaa roolin pelkästään käyttöliittymäsuunnittelussa ja ulkoistaa sovelluslogiikan esimerkiksi ulkopuoliselle tai uudelle työntekijälle. Responsiivinen ulkoasu takaa, että sivuston käyttö ja julkaisujärjestelmän hallitseminen sujuu miltä tahansa laitteelta sujuvasti.

Omasta julkaisujärjestelmästä tulee vanhaan Wordpress-järjestelmään verrattuna yksinkertainen. Uskon, että kohdeyrityksen henkilökunta selviää sisällöntuottamisessa hyvin WYSIWYG-editorin avulla. Varsinaista hallintapaneelia ylläpitäjille en tule toteuttamaan, sillä komponenttialueet tekevät yksittäisen sivun ja käyttöliittymän muokkaamisesta todella loogista. Tällöin sivun muokkaaja näkee heti mitä sivua ja kohtaa hän on muokkaamassa. Henkilökunta tulee varmasti myös arvostamaan komponenttien ajastusta, sillä nykyisessä Wordpress-

järjestelmässä se on hankalasti käytettävän lisäosan takana, eikä siinä ole mahdollista ajastaa vain tiettyä kohtaa sivulta. Komponenttien ajastuksella kohdeyrityksen henkilökunta pystyy helposti hallitsemaan aikaan sidottuja kampanjoita. Dynaamisten verkko-osoitteiden ansiosta henkilökunta pystyy lisäämään sisältöä mille tahansa haluamalleen uudelle verkkosivulle. Näin vältytään nykyisen Wordpressin erillisestä sivun lisäystoiminnosta hallintapaneelin kautta.

Itselleni suurin osa opinnäytetyössä käytetyistä tekniikoista on ennalta tuttuja, mutta opin myös uutta mm. hakukoneoptimoinnista ja tietoturvasta. Opinnäytetyötä tehtäessä korostui tietoturvan tärkeys eri julkaisujärjestelmissä, joten OWASP:n ja kansainvälisen haavoittuvuustietokannan tietoturvauhat tulevat varmasti aiheuttamaan haasteita julkaisujärjestelmää tehtäessä. On hienoa päästä tulevaisuudessa itse toteuttamaan suunnitelma ja mahdollisesti hyödyntää sitä myös muissa projekteissa.

# LÄHTEET

Contegro 2012. A (brief) history of CMS development. Viitattu 8.12.2013 http://www.contegro.com/info-center/designers-blog/blog-article/_thread_/a-brief-history-of-cms-development.

Content management system 2013. Webopedia. Viitattu 29.11.2013 http://www.webopedia.com/TERM/C/content_management_system.html.

DB-Engines 2014. DB-Engines Ranking. Viitattu 9.6.2014 http://db-engines.com/en/ranking

Htmlgoodies 2013. I Want To Build A Website. Do I Need a Content Management System? Viitattu 31.10.2013 http://www.htmlgoodies.com/beyond/webmaster/toolbox/article.php/3887866/I-Want-To-Build-A-Website-Do-I-Need-a-Content-Management-System-CMS.htm.

Javascript introduction. W3Schools. Viitattu 5.6.2014 http://www.w3schools.com/js/js_intro.asp.

Jquery introduction. W3Schools. Viitattu 20.3.2014 http://www.w3schools.com/jquery/jquery_intro.asp.

NIST 2014. National Vulnerability Database. Search CVE and CCE Vulnerability Database. Viitattu 5.6.2014 http://web.nvd.nist.gov/view/vuln/search.

OWASP 2014. OWASP Top Ten 2013 Project. Viitattu 21.5.2014 https://www.owasp.org/index.php/category:OWASP_Top_Ten_2013_Project.

Pearche, J. 2011. Professional Mobile Web Development. Indianapolis: Wiley.

Smarty 2014. What's a template engine and why should I use one? Viitattu 12.1.2014 http://smarty.incutio.com/?page=SmartyFrequentlyAskedQuestions#basics-1.

The history of the Web. W3C 2014. Viitattu 24.3.2014 http://www.w3.org/wiki/The_history_of_the_Web.

Usage of content management systems for websites. W3Techs 2013. Viitattu 31.10.2013 http://w3techs.com/technologies/overview/content_management/all.

Zer0byte 2014. OWASP Top 10 for 2013 is now officially released. Viitattu 21.5.2014 http://zer0byte.com/2013/06/13/owasp-top-10-2013-officially-released/.