

Alexi Luoma

# WordPress-liitännäinen

## PC-rakentamiseen

Insinööri (AMK)

Tietotekniikka

Kevät 2017



KAJAANIN  
AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES

## **Tiivistelmä**

**Tekijä:** Luoma Aleksis

**Työn nimi:** WordPress-liitännäinen PC-rakentamiseen

**Tutkintonimike:** Insinööri (AMK), tietotekniikka

**Asiasanat:** WordPress, liitännäinen, PC, verkkosovellus, verkkosivut, AJAX, JavaScript, PHP

Opinnäytetyön tavoitteena oli luoda asiakasyrityksen käyttöön verkkosovellus edistämään räätälöityjen pöytä tietokoneiden myymistä. Sovelluksen tuli integroitua osaksi yrityksen olemassa olevia WordPress-verkkosivuja. Sen tuli olla helppokäyttöinen ja selkeä.

Teoriapohjaa haettiin tutustumalla erilaisiin ohjelmointikieliin sekä erityisesti WordPressiin. Sovellus päätettiin toteuttaa WordPress-liitännäisenä, jotta sen integrointi yrityksen sivuille olisi mahdollisimman helppoa. Projektissa käytettiin virtuaalista kehitysympäristöä ja työ toteutettiin HTML-, CSS-, PHP-, SQL-, JavaScript- ja AJAX-tekniikoilla.

Lopputuloksena oli toimiva prototyyppi. Liitännäisen avulla pystyttiin valitsemaan tarvittavat ja yhteensopivat komponentit pöytä tietokoneen kasaamiseksi haluttujen avainsanojen mukaan. Sovellusta hallinnoitiin suoraan WordPressin admin-ympäristöstä.

Asetetut tavoitteet saavutettiin ja yritys pystyy jatkokehittämään prototyyppiä itsenäisesti.

## **Abstract**

**Author:** Luoma Aleksi

**Title of the Publication:** WordPress-plugin: PC-builder

**Degree Title:** Bachelor of Engineering, Information Technology

**Keywords:** WordPress, plugin, PC, web application, website, AJAX, JavaScript, PHP

The purpose of this thesis was to create a PC building web application to help the commissioner sell custom-built computers. The web application had to be integrated as a part of the company's existing WordPress-site. The application also had to be easy to use.

The theory for this thesis consisted of different programming languages, web development techniques and especially WordPress. The web application was created as a WordPress plugin to make it easy to integrate into the commissioner's website. A virtual development environment was utilized for the project and the used techniques included HTML, CSS, PHP, SQL, JavaScript and AJAX.

Due to difficulties with maintaining the schedule, some features were cut from the application but the end result was a working prototype. The plugin could be used to choose the necessary and compatible components for building a PC. Components could also be filtered by tags. The application was managed directly from the WordPress admin environment.

The set goals were met and the company may continue to develop the prototype as they see fit.

## Sisällys

1	Johdanto.....	1
2	Teoria.....	2
2.1	Pöytätietokoneen rakenne ja toiminta.....	2
2.1.1	Tietokoneen toiminta.....	2
2.1.2	Tietokoneen komponentit.....	4
2.2	WordPress .....	5
2.3	Kielet ja tekniikat .....	6
2.3.1	HTML & CSS .....	6
2.3.2	PHP .....	7
2.3.3	JavaScript.....	8
2.3.4	AJAX .....	8
2.3.5	SQL .....	8
2.4	Virtuaalinen kehitysympäristö.....	9
2.5	Käytettävyys.....	9
3	Tavoitteet .....	12
3.1	Oppimistavoitteet .....	12
3.2	Projektin tavoitteet.....	12
4	Toteutus.....	14
4.1	Kehitysympäristö.....	14
4.2	Liitännäisen luominen.....	14
4.3	Tietokannat .....	15
4.4	Käyttöliittymä.....	15
4.4.1	Käyttöliittymä ylläpitäjälle .....	15
4.4.2	Käyttöliittymä asiakkaalle.....	27
4.5	Komponenttien suodatus.....	32
4.5.1	Avainsanasuodatus .....	32
4.5.2	Yhteensopivuussuodatus.....	32
5	Arviointi .....	34
6	Yhteenveto.....	35
	Lähteet.....	36

## **Symboliluettelo**

AJAX: Asynchronous JavaScript And XML, mahdollistaa dynaamisen sisällön verkkosivuilla.

ALU: Arithmetic Logical Unit eli laskentayksikkö, osa prosessoria.

CPU: Central Processing Unit eli suoritin tai prosessori, suorittaa tietokoneen konekielisiä käskyjä.

CU: Control Unit eli valvontayksikkö, osa prosessoria.

CSS: Cascading Style Sheets on verkkosivujen ulkonäön muokkaamisen tyyliohje.

GPU: Graphics Processing Unit eli grafiikkaprosessori on 2D- tai 3D-grafiikan renderöintiin tarkoitettu erikoistunut mikroprosessori.

HTML: HyperText Markup Language on verkkosivujen luomiseen käytetty standardoitu merkintäkieli.

Modaali-ikkuna: Varsinaisen pääikkunan eteen aukeava alaikkuna.

MySQL: avoimen lähdekoodin relaatiotietokantaohjelmisto.

PC: Personal Computer eli henkilökohtainen tietokone.

PHP: PHP: Hypertext Preprocessor on komentosarjakieli, joka on yleisesti käytössä web-palveluiden luomisessa.

RAM: Random-Access Memory eli keskusmuisti on tietokoneen työmuisti.

SQL: Structured Query Language on standardoitu relaatiotietokantojen kyselykieli.

Vagrant: kehitysympäristöjen rakentamiseen ja jakamiseen tarkoitettu virtualisointia hyödyntävä työkalu.

VVV: Varying Vagrant Vagrants on avoimen lähdekoodin Vagrant toteutus, pääpainona WordPress kehitys.

WordPress (lyh. WP): avoimen lähdekoodin sisällönhallintajärjestelmä.

## 1 Johdanto

Tässä opinnäytetyössä luodaan räätälöityjen PC-kokonaisuuksien myyntiä edistävä työkalu.

Toimeksiantajana opinnäytetyölle on Tietotekniikkapalvelu Valvatti Oy. Tietotekniikkapalvelu Valvatti Oy on Tampereella sijaitseva tietotekniikka-alan yritys, joka tarjoaa asiakkailleen erilaisia huolto- ja neuvontapalveluita. Lisäksi yritys tarjoaa verkkosivujen kehitys- ja ylläpitopalveluja, myy asiakkailleen tarpeen mukaan tietoteknistä laitteistoa sekä kauppa räätälöityjä PC-ratkaisuja.

Projektin taustana on Tietotekniikkapalvelu Valvatti Oy:n räätälöityjen tietokonekokonaisuuksien myynti. Yrityksen verkkosivuille tarvitaan käyttöliittymä yrityksen valikoimien selaamiselle ja tapa edistää kokonaisten PC-ratkaisujen myyntiä. Sivustolle halutaan luoda asiakkaille helppo sovellus komponenttien valintaan.

## 2 Teoria

Seuraavissa luvuissa käydään läpi projektin toteuttamista pohjustavaa teoriaa. Luotavaa sovellusta varten tarvittiin tietoa pöytätietokoneiden rakenteesta, komponenteista sekä niiden kokoamisesta. Näitä asioita käydään tarkemmin läpi luvussa 2.1.

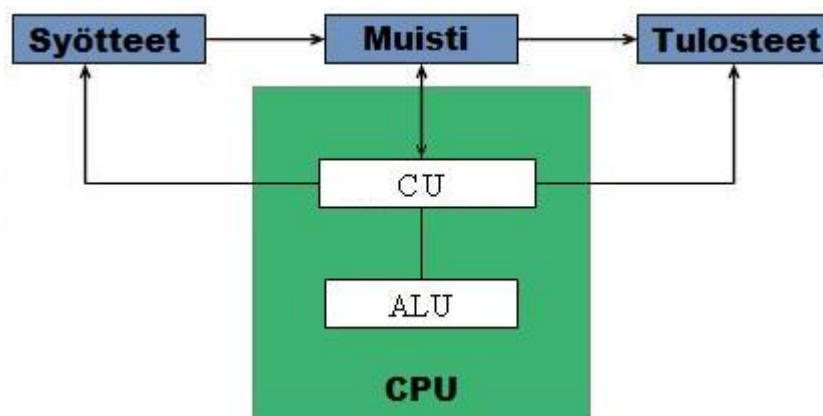
Koska projektissa rakennettava verkkosovellus on toteutettu WordPress liitännäisenä, esitellään WordPress lyhyesti omassa luvussaan 2.2. Lisäksi luvussa 2.3 käydään läpi työn aikana käytetyt työkalut sekä ohjelmointikielet.

Alaluvussa 2.4 tutustutaan virtuaalisen kehitysympäristön teoriaan. Lopuksi avataan käytettävyyden määritelmää alaluvussa 2.5. Luvussa myös pohditaan, mikä merkitys käytettävyydellä on tämän työn kannalta ja mitä tässä työssä tulee ottaa erityisesti huomioon.

### 2.1 Pöytätietokoneen rakenne ja toiminta

Projektia varten tarvittiin taustatietoa pöytätietokoneiden komponenteista sekä PC:n ko-koamisesta ja toiminnasta. Sovellusta varten tuli selvittää, millaisia tietoja erilaisista komponenteista tarvitaan, jotta esimerkiksi yhteensopivuus voidaan varmistaa.

#### 2.1.1 Tietokoneen toiminta



Kuva 1. Tietokoneen lohkokaavio yksinkertaistettuna

Kuvassa 1 nähtävässä lohkokaaviossa on esillä tietokoneen toiminnan ja käytön kannalta välttämättömät osat. Eri osioiden operointi ja merkitys käsitellään tarkemmin tässä luvussa perustuen Dinesh Thakurin artikkeliin "Block Diagram of Computer and Explain its Various Components".

Tietokone tarvitsee syötteitä vastaanottaakseen ohjeita ja dataa operaatioiden suorittamiseksi. Syötteet annetaan tietokoneelle erilaisten syötelaiteiden avulla, esimerkiksi näppäimistöllä ja hiirellä.

Muisti on tietokoneen osio, joka säilyttää dataa ja ohjeita suoritusta varten. Se säilyttää muun muassa laskutoimitusten vaiheet ja tulokset ennen niiden näkymistä tulosteissa. Muistiin voidaan myös tallentaa dataa myöhempää käyttöä varten. Tietokoneen muisti voidaan jakaa päämuistiin ja toissijaiseen muistiin.

Päämuisti eli välimuisti tai keskusmuisti (RAM) on nopeasti käytettävää muistia ja sitä käytetään ajettavan ohjelman säilyttämiseen sekä esimerkiksi syötteiden käsittelyyn. Keskusmuisti on väliaikaista ja mikäli dataa halutaan säilyttää lopullisesti, se siirretään toissijaiselle muistille.

Toissijaista muistia käytetään arkistona. Sille voidaan tallentaa erilaisia ohjelmia, dokumentteja, tietokantoja ynnä muuta. Avattaessa ohjelma toissijaisesta muistista siirretään se suoritukseen ajaksi yleensä keskusmuistiin. Kun ohjelma suljetaan ja sen tulokset tai muokkaamat dokumentit tallennetaan, tehdään se toissijaiselle muistille. Toissijainen muisti on hitaampaa, mutta myös halvempaa kuin päämuisti. Yleisiä toissijaisen muistin laitteita ovat esimerkiksi sisäiset ja ulkoiset kovalevyt sekä CD-levyt.

Tulosteet ovat tapa kommunikoida tietokoneen suorittamien operaatioiden ja laskennan tulokset ulkopuoliselle maailmalle. Tulostelaitteita ovat muun muassa näytöt ja printterit.

CPU (Central Processing Unit) eli suoritin tai prosessori on tietokoneen osa, jossa suoritetaan tietokoneohjelmien operaatiot ja laskenta. Prosessorin keskeiset osat ovat CU (Control Unit) eli valvontayksikkö sekä ALU (Arithmetic Logical Unit) eli laskentayksikkö.

Valvontayksikkö hallitsee muita tietokoneen osia. Valvontayksikkö määrittää mihin tallentaa syötteitä ja se hallitsee datan sekä ohjeiden välittämistä muistista laskentayksikölle ja toisinpäin.



Laskentayksikössä suoritetaan kaikki tietokoneen laskenta. Tietokoneohjelman suorittaessa laskentaa valvontayksikkö välittää muistista tarvittavan datan laskentayksikölle. Laskennan suorittamisen jälkeen valvontayksikkö saa tulokset laskentayksiköltä ja välittää ne muistille ja mahdollisesti siitä eteenpäin tulostelaitteille.

### 2.1.2 Tietokoneen komponentit

Tietokone koostuu kotelon lisäksi emolevystä, prosessorista, keskusmuistista, kiintolevystä ja virtalähteestä. Lisäksi tietokoneessa voi olla erillinen näytönohjain tai äänikortti, tai esimerkiksi Bluetooth-adapteri. (GCF Global.) Kuvassa 2 on esiteltyä esimerkki pöytätietokoneesta ja sen osista.



Kuva 2. PC:n osat (Viogaming)

Tätä työtä varten oletettiin, että kasattavaan pöytätietokoneeseen kuuluu kotelo, emolevy, prosessori, keskusmuisti, kiintolevy, virtalähde sekä näytönohjain. Koska verkkosovelluksen tarkoituksena on auttaa käyttäjää luomaan osista toimiva kokonaisuus, täytyy yhteensopivuutta seurata erilaisilla, osiin sopivilla kriteereillä. Esimerkiksi prosessorissa täytyy varmistaa, että sen kanta sopii valittuun emolevyyn. Kriteerit ja yhteensopivuuden selvittäminen on esitetty tarkemmin käytännön toteutusta kuvaavissa luvuissa.

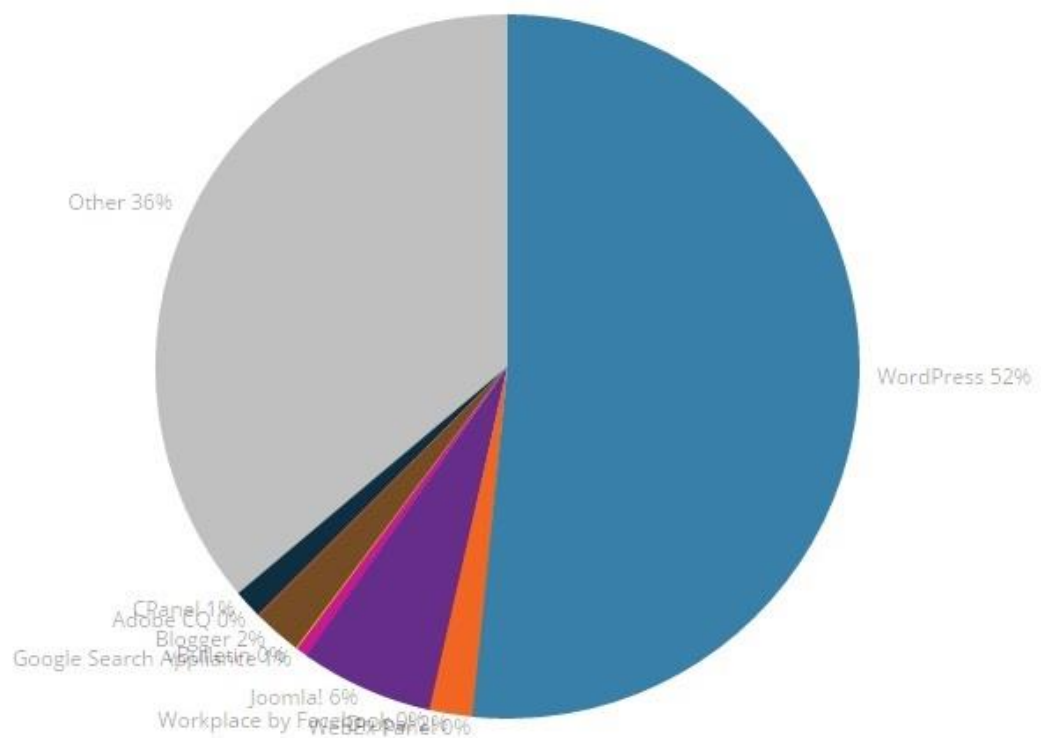
## 2.2 WordPress

Opinnäytetyönä toteutettu kustomoitujen pöytätietokoneiden tilaamista helpottanut sovellus rakennettiin WordPressiä hyödyntäen. Sisällönhallintajärjestelmän toiminnallisuutta laajentamalla toteutettu sovellus on helposti sekä yrityksen että yritysten asiakkaiden ulottuvilla.

WordPress on yksi verkon suosituimmista sisällönhallintajärjestelmistä. Yli puolet maailman verkkosivuista, jotka on rakennettu jonkin sisällönhallintajärjestelmän päälle, on toteutettu WordPressillä (BuiltWith Pty Ltd). Tarkempi käytettyjen sisällönhallintajärjestelmien jakauma näkyy kuvasta 3.

# CMS Usage Statistics

Statistics for websites using CMS technologies



Kuva 3. Sisällönhallintajärjestelmien käytön jakauma (BuiltWith Pty Ltd)

Sisällönhallintajärjestelmä mahdollistaa sisällön julkaisemisen suoraan internetiin sekä sen hallinnoinnin etänä. Sen avulla useampi käyttäjä voi hyödyntää samaa alustaa omien sisältöjensä luomiseen eikä käyttäjien tarvitse ladata erillisiä ohjelmistoja. (Techopedia.)

WordPressin lähdekoodi on vapaasti käytettävissä ja muokattavissa, ja se pohjautuu PHP- ja MySQL-tekniikoille. Alun perin WordPress oli lähinnä bloggaajien käytössä ja sen avulla pystyi luomaan yksinkertaisia sivuja, joille käyttäjät saivat lisätä sisältöä. Nykyään WordPress toimii kuitenkin täysivaltaisena sisällönhallintajärjestelmänä, ja sitä käyttävät esimerkiksi The Walt Disney Company ja Rovion Angry Birds. (Wordpress.org a.)

WordPressin avulla käyttäjä pystyy luomaan juuri tarpeeseensa sopivan sivuston. Sadat tuhannet käyttäjät ja kehittäjät luovat jatkuvasti uusia teemoja ja liitännäisiä, joiden avulla muut voivat muokata ja parantaa sivustojaan. Liitännäisiä on jo yli 45 tuhatta, ja lisää kehitetään jatkuvasti. Niiden avulla käyttäjät voivat lisätä sivustoilleen toiminnallisuutta (kuten verkkokaupan) ilman, että he joutuvat itse koodaamaan. Tässä suhteessa WordPress on erittäin käyttäjäystävällinen, sillä myös aloittelija pystyy laittamaan pystyyn perussivuston ilman pitkää opettelijaksoa. Suuri yhteisö takaa myös sen, että tukea on aina saatavilla muiden käyttäjien muodossa. (Wordpress.org a.)

Sisällönhallintajärjestelmiä on tällä hetkellä useita ja sopivia vaihtoehtoja on paljon. WordPressiin päädyttiin, koska työ tehtiin osaksi jo olemassa olevaa sivustoa, joka on toteutettu WordPressillä. WordPressistä on olemassa kaksi versiota: WordPress.com ja WordPress.org. Tässä työssä painotus on org-versiolla, joka mahdollistaa oman koodin lisäämisen ja paremmat sivuston räätälöintimahdollisuudet.

## 2.3 Kielet ja tekniikat

Seuraavissa luvuissa esitellään lyhyesti projektissa käytetyt kielet ja tekniikat. Lisäksi kuvataan niiden rooli sovelluksen kehityksessä.

### 2.3.1 HTML & CSS

HTML eli HyperText Markup Language on standardoitu merkintäkieli verkkosivujen luomiseen. HTML-koodi kuvailee verkkosivujen rakennetta, ja sen avulla selaimet osaavat

näyttää sivut oikein. Rakenteen merkitsemiseen käytetään avainsanoja kuten paragraph (kappale) ja table (taulukko), jotka sijoitetaan aina kulmasulkeisiin. (W3C b.)

Projektissa HTML-koodia tarvittiin kaiken selaimessa näytettävän sisällön luomiseen. Käyttöliittymiä projektissa on kaksi, yksi ylläpitäjällä ja toinen yrityksen asiakkaille. Suurin osa liitännäisen HTML-koodista on generoitu dynaamisesti PHP:tä hyödyntäen.

CSS eli Cascading Style Sheets on verkkosivujen ulkonäön muokkaamiseen tarkoitettu tyyliohje (W3C a). Sen tärkeimmät käsitteet ovat kaskadi (cascade) sekä periytyvyys (inheritance), jotka määrittelevät tyyliohjeille järjestyksen. Ensisijaisesti elementit ottavat arvon suoraan niille annetusta tyyliohjeesta. Mikäli erillistä arvoa ei ole määritelty, elementti perii tarvittavan arvon vanhemmaltaan eli ylemmän tason elementiltä. (Korpela 2013, 43 - 44.)

Työssä luodun liitännäisen mukana tulee ylläpitosivustoa varten oma tyyliohjetiedosto. Loppukäyttäjän, eli itse sovelluksen, käyttöliittymän CSS-ohjeet tulevat WordPress teeman tyyliohjetiedostosta. Tällöin yrityksen sivuilla asiakkaille näkyvän käyttöliittymän ulkoasu on täysin yrityksen omissa käsissä. Näin varmistetaan myös jo luotujen tyyliohjeiden toimivuus sekä vältetään esimerkiksi periytyvyyteen liittyviltä ongelmilta.

### 2.3.2 PHP

PHP eli PHP: Hypertext Preprocessor on komentosarjakieli, joka on yleisesti käytössä dynaamisten verkkosivujen luomisessa. Se on verkkopalvelimella ajettava ohjelmointikieli, ja se voidaan upottaa osaksi HTML-koodia. Sen avulla voi esimerkiksi hakea tietoa palvelimella olevasta tietokannasta ja näyttää sen halutulla tavalla käyttäjän selaimella. (php.net.)

Projektissa PHP:tä käytettiin tietokannan kanssa keskustelemiseen sekä HTML-sivujen ja sisällön generoimiseen. Tietokantataulujen luonti, niiden päivitys ja taulujen sisällön muokkaaminen suoritetaan liitännäisen php-tiedostoissa. SQL-komennot välitetään WordPressin MySQL-tietokannalle erilaisten php-funktioiden kautta.

### 2.3.3 JavaScript

JavaScript on ohjelmointikieli, jonka avulla verkkosivuista voidaan tehdä interaktiivisia. Se soveltuu käytettäväksi silloin, kun koodi täytyy suorittaa sivun lataamisen jälkeen. (Chapman 2017.)

Sovelluksen käyttöliittymien toiminnallisuus ja HTML-sisällön dynaaminen muuttaminen suoritetaan JavaScript-funktioissa. Liitännäisessä on kaksi erillistä JavaScript-tiedostoa, toisessa ylläpitosivuston ja toisessa asiakkaiden käyttöliittymän tarvitsema koodi.

### 2.3.4 AJAX

AJAX eli Asynchronous JavaScript and XML mahdollistaa sivun osien päivittämisen ilman, että koko sivu täytyy ladata uudelleen. Kun jotain tapahtuu selaimessa, AJAX lähettää palvelimelle pyynnön. Palvelin lähettää vastauksen takaisin selaimelle, jossa uusi tieto käsitellään JavaScriptin avulla. Lopuksi päivitetään muuttunut tieto näkymään käyttäjälle. (W3C c.)

Tässä projektissa AJAX-tekniikkaa hyödynnetään HTML-sisällön päivittämiseen JavaScript- ja php-kieliä käyttäen. JavaScript-funktioista kutsutaan XMLHttpRequest-objektin kautta php-tiedostoja, joista saadaan HTML-elementteihin uusi sisältö.

### 2.3.5 SQL

SQL eli Structured Query Language on standardoitu kyselykieli, jonka avulla relaatiotietokantoihin voidaan suorittaa hakuja sekä tehdä muutoksia ja lisäyksiä. Sitä käytetään usein myös tietynlaisen informaation hakemiseen suuremmista tietokannoista. Yleisimpiä SQL-lauseita ovat valitse (select), lisää (add), päivitä (update), poista (delete) sekä luo (create). (Rouse 2016.)

WordPress käyttää SQL-tietokantaa, ja siksi tässä projektissakin tietokantataulujen luonti, muokkaus ja niiden sisällön hallinta tehdään SQL-kieltä käyttäen. SQL-kyselyt välitetään MySQL-tietokannalle WordPressin omien php-funktioiden kautta.

## 2.4 Virtuaalinen kehitysympäristö

Projektin toteuttamisessa käytettiin virtuaalista kehitysympäristöä. Vaikka yrityksellä oli jo toimivat verkkosivut ja sovelluksen oli määrä tulla osaksi sitä, erillisellä kehitysympäristöllä varmistettiin yrityksen sivujen toimivuus myös kehityksen aikana. Kehitysympäristössä on helppo kokeilla erilaisia ratkaisuja eivätkä esimerkiksi virheet koodissa aiheuta yritysten sivujen kaatumista, jolloin asiakkaat eivät löydä haluamiansa tietoja ja on mahdollisuus tulojen menetykseen.

Virtuaalisen kehitysympäristön käytöllä on monia hyötyjä. Niitä voi luoda ja tuhota tarpeen mukaan eivätkä ne ole sidottuja vain yhteen koneeseen. Mikäli jokin menee virtuaaliympäristössä pieleen, se ei myöskään vaikuta fyysiseen koneeseen. Työn varmuuskopiointi on lisäksi helpompaa, sillä virtuaaliympäristön voi kopioida turvaan. Virtuaaliympäristössä pystyy vaikuttamaan myös esimerkiksi keskusmuistin tai prosessorien määrään ilman erillisten fyysisten komponenttien hankkimista. (apriorit.)

Vaikka virtuaaliympäristössä ei voikaan testata toimivuutta kaikilla laitteilla ja kokoonpanoilla, oli sen käyttäminen riittävää tälle projektille. Isommissa projekteissa tulee ottaa myös huomioon virtuaalikoneiden viemä tila ja jättää tarpeeksi aikaa testaukselle myös fyysisillä koneilla. Näin saadaan varmistettua, että sovellus toimii varmasti myös loppukäyttäjän oikealla koneella. (apriorit.)

## 2.5 Käytettävyys

Opinnäytetyön tavoitteena oli luoda sovellus yrityksen asiakkaiden käyttöön, joten suunnittelussa ja toteutuksessa tuli kiinnittää erityistä huomiota lopputuloksen käytettävyyteen ja selkeyteen.

Verkkosivuista puhuttaessa käytettävyydellä tarkoitetaan sitä, että käyttäjä löytää haluamansa tiedon nopeasti ja tehokkaasti. Tiedon tulee olla selkeästi esillä ja elementtien omilla alueillaan. Sivuston käyttäjälle ei pitäisi olla ristiriitaisia ohjeita, mutta huomioon pitää ottaa myös mahdolliset käyttäjien tekemät virheet. (Kelly 2016.)

Jakob Nielsen on määritellyt käytettävyydelle kymmenen heuristista sääntöä. Ne kuvailevat, miten hyvä käyttöliittymä toimii ja mitä sellaisen suunnittelussa tulee ottaa huomioon. Säännöt esitellään tässä lyhyesti pohjautuen Nielsenin artikkeliin " 10 Usability Heuristics for User Interface Design" (1995).

Riittävä palaute: käyttäjän tulisi aina tietää, mitä järjestelmässä tapahtuu. Järjestelmän on hyvä antaa palautetta, jonka käyttäjä ymmärtää, järkevän ajan sisällä.

Käyttäjien oma kieli: Järjestelmässä tulee käyttää termejä ja sanoja, jotka käyttäjän on helppo ymmärtää.

Selkeä poistumistapa eri tiloista ja tilanteista: Käyttäjän tulee pystyä helposti peruuttamaan tekemänsä virheellinen operaatio. "Kumoa" ja "tee uudelleen"-toimintojen tukeminen helpottaa käyttäjää.

Yhdenmukaisuus: Järjestelmässä tulisi noudattaa selkeitä nimeämiskäytänteitä. Yhden sanan tulee aina tarkoittaa samaa asiaa, jolloin välttyään väärinymmärryksiltä.

Virheiden estäminen: Käyttäjät tekevät helposti virheitä, joten järjestelmän olisi hyvä osata ennakoida nämä tilanteet ja estää virheelliset toiminnot.

Käyttäjän muistikuorman minimointi: Käyttäjää ei pidä velvoittaa muistamaan asioita, joita järjestelmässä on aikaisemmin tullut vastaan. Ohjeiden tulee aina olla helposti saatavilla.

Oikopolut: Järjestelmän on hyvä olla tehokas sekä aloittelijoille että kokeneemmille käyttäjille. Siksi olisi hyvä, jos kokeneemmat käyttäjät voivat oikaista käytetyimmissä toiminnoissa esimerkiksi näppäinyhdistelmällä.

Yksinkertainen ja luonnollinen dialogi: Dialogissa ei tule olla ylimääräistä tietoa.

Selkeät virheilmoitukset: Virheilmoitukset tulee olla käyttäjälle selkeästi ilmoitettuna eikä esimerkiksi pelkkinä virhekoodeina. Niiden on myös hyvä ehdottaa virheeseen ratkaisua.

Riittävä ja selkeä apu ja dokumentaatio: Järjestelmän on lähtökohtaisesti hyvä olla käytettävissä ilman erillistä dokumentaatiota, mutta sellainen on kuitenkin hyvä tarjota. Dokumentaation tulee olla helposti löydettävissä ja sisältää käyttäjän toimintoihin liittyviä ohjeita tilanteiden ratkaisemiseksi.

Tärkeimpänä osana tämän projektin käytettävyyttä on asiakkaan tietotason huomioiminen. Kaikki asiakkaat eivät välttämättä itse tiedä, mitä osia tietokoneessa tulisi olla. Tällöin työkalun tulee selkeästi pyytää käyttäjältä kaikkien tarvittavien osien tiedot. Tärkeää on lisäksi se, että työkalu osaa pyytää tiedot oikeassa muodossa ja tarvittaessa huomauttaa, mikäli virhe on tapahtunut.

Kun kaikki tiedot on oikein kerätty, asiakas voi lähettää tilauksensa hyväksyttäväksi yritykselle. Tämän tulee toimia helposti niin, että asiakas varmasti ymmärtää, kun viesti on lähetetty ja tietää, miten prosessi siitä eteenpäin jatkuu.

Toisaalta työssä on kiinnitettävä huomiota myös yrityksen tapaan hallinnoida ja ylläpitää sovellusta jatkossa. Yrityksen edustajan pitää pystyä helposti lisäämään uusia tuotteita valikoimiin sekä tarvittaessa muokkaamaan ja poistamaan vanhoja tietoja.



### 3 Tavoitteet

Tämän opinnäytetyön tavoitteet jaettiin kahteen eri osa-alueeseen.

Ensimmäinen osa-alue on henkilökohtaiset oppimistavoitteet, jotka asetettiin ja joiden saavuttamista arvioitiin itsenäisesti projektin loppupuolella.

Toinen osa-alue on itse projektiin liittyvät tavoitteet, jotka asetettiin yhdessä yrityksen edustajan kanssa. Näiden tavoitteiden toteutuvuutta myös arvioitiin yhdessä yrityksen kanssa.

Asetetut tavoitteet käydään tarkemmin läpi seuraavissa alaluvuissa. Niiden saavuttamisesta kerrotaan luvussa 5 Arviointi.

#### 3.1 Oppimistavoitteet

Projektin aikana pyritään oppimaan WordPress-liitännäisten kehitystä sekä saavuttamaan syvempi ymmärrys WordPressin toiminnasta. Lisäksi erilaisten verkkokehityksessä käytettävien kielten ja tekniikoiden osaamista tullaan kartuttamaan.

Projektin kannalta tärkeimpiä kieliä ovat HTML, CSS, JavaScript, PHP sekä SQL. Projektissa hyödynnettävistä tekniikoista erityisesti dynaaminen verkkosivujen sisällön hallinta AJAXilla sekä MySQL-tietokannan kanssa kommunikointi ovat keskeisessä roolissa. Varsinkin JavaScript- ja PHP-kielistä sekä AJAX-tekniikasta tulisi oppia paljon uutta, sillä niistä minulla ei ole huomattavia määriä kokemusta ja ne ovat hyvin keskeisessä roolissa sovellusta kehitettäessä.

#### 3.2 Projektin tavoitteet

Projektin tarkoitus on luoda toimeksiantajayritykselle räätälöityjen tietokonekokonaisuuksien myyntiä edistävälle verkkosovellukselle prototyyppi.

Verkkosovelluksessa olennaista on sen toimiminen yrityksen verkkosivuston WordPress-ympäristössä. Sovelluksen ylläpitäminen halutaan mahdollisimman helpoksi, joten se tulisi olla integroituna WordPressin ylläpitosivuille.

Verkkosovelluksen avulla yrityksen asiakkaat voivat selata yrityksen PC-komponenttien valikoimaa ja valita sieltä itsellensä osat haluamaansa kokonaisuuteen. Sovelluksen on tarkoitus olla helppo käyttää, hoitaa komponenttien yhteensopivuuden tarkistaminen sekä tarjota asiakkaalle tapa löytää sopivat komponentit käyttötarkoituksen mukaan.

Projektissa luodusta prototyypistä yrityksen on mahdollista jatkaa verkkosovelluksen kehittämistä.

## 4 Toteutus

Projektin toteutus on jaettu eri osa-alueisiin. Näistä ensimmäinen oli kehitysympäristön luominen. Seuraavassa vaiheessa luotiin rakennettavan liitännäisen pohja, ja loput osa-alueet laajensivat liitännäisen toiminnallisuutta. Liitännäisen tärkeimmät toiminnallisuuden luomisen osa-alueet olivat tietokantaoperaatiot, käyttöliittymät sekä komponenttien suodatusominaisuudet.

Nämä vaiheet ja osa-alueet esitellään tarkemmin seuraavissa alaluvuissa.

### 4.1 Kehitysympäristö

Kehitysympäristökseni monistin yrityksen Varying Vagrant Vagrants (VVV) WordPress-kehitysympäristön omalle työskentelykoneelleni. VVV-kehitysympäristö pyörii Ubuntu 14.04 -virtuaalikoneella, jota ajetaan paikallisesti hyödyntäen VirtualBox-ohjelmaa.

VVV on avoimen lähdekoodin Vagrant-toteutus, jonka pääpainona on WordPress-kehitys. Vagrant on työkalu, joka on suunniteltu kehitysympäristöjen rakentamiseen ja jakamiseen hyödyntäen virtualisointia. VVV sopii hyvin WordPress-teemojen, -liitännäisten tai WordPress-ytimen kehittämiseen. (Varying Vagrant Vagrants.)

### 4.2 Liitännäisen luominen

Toteutettavalle liitännäiselle tehtiin oma kansio WordPress 'plugins'-hakemistoon, ja tämän juureen liitännäisen päätiedosto. Liitännäisen päätiedosto sisältää liitännäisen tiedot, kuten liitännäisen nimen, kehittäjän, kuvauksen, versionumeron ja niin edespäin. WordPress hakee ja suorittaa päätiedoston liitännäistä käyttöönottaessa, joten kaikki alustaminen ja vastaavat operaatiokutsut suoritetaan tästä tiedostosta. Tiedosto voi joko suoraan sisältää liitännäisen koodin tai vaihtoehtoisesti se sisältää kutsuja muihin tiedostoihin.

Tämän projektin laajuus huomioon ottaen liitännäisen päätiedosto on toteutettu koontitiedostona, josta vain kutsutaan muualla liitännäishakemistossa sijaitsevia tiedostoja ja niiden funktioita.

### 4.3 Tietokannat

Projektin tarvitsemat SQL-taulut luodaan, kun liitännäinen otetaan ensimmäisen kerran käyttöön WordPress-ympäristössä ja sen päivitystarve tarkistetaan aina kun liitännäinen aktivoidaan WordPress admin-ympäristöstä.

Tietokantataulut luodaan WordPressin dbDelta-funktion kautta. dbDelta on WordPressin suositteleva tapa muokata WordPress-tietokantaa, ja se on tarkoitettu WordPress-tietokantaan taulujen luomiseen tai niiden päivittämiseen. Käytetty funktio on esitetty kuvassa 4.

```
dbDelta( string/array $queries = '', bool $execute = true )
```

Kuva 4. dbDelta-funktio (WordPress.org c)

dbDelta-funktiolle annettavat SQL-kyselyt generoidaan konfiguraatitiedoston pohjalta. Konfiguraatitiedosto sisältää listan eri komponenteista, ja siellä on kullekin komponentille määritelty ominaisuuksien nimet, tietokantataulukkojen sarakkeiden tunnisteet sekä datatyytit. Tietokantataulut luova funktio käy siis läpi jokaisen komponenttityypin ja luo sen ominaisuuksien pohjalta SQL-kyselyt taulujen luomista varten.

Komponentit sisältävien taulujen lisäksi luodaan avainsana-taulu komponenttien luokitte-  
lua varten. Avainsana-taulu sisältää erilaisia termejä, joiden mukaan komponentteja voi-  
daan luokitella. Avainsanojen suhteet eri komponentteihin on toteutettu erillisillä avain-  
sanakarttatauluilla, joissa on listattuna avainsanojen ja komponenttien tunnistepareja.

### 4.4 Käyttöliittymä

Liitännäisessä on kaksi käyttöliittymätoteutusta, toinen liitännäisen hallintaan ja toinen asiakkaalle sovelluksen käyttämiseen. Liitännäisen hallintaa varten luodaan WordPress admin-ympäristöön hallintasivuja.

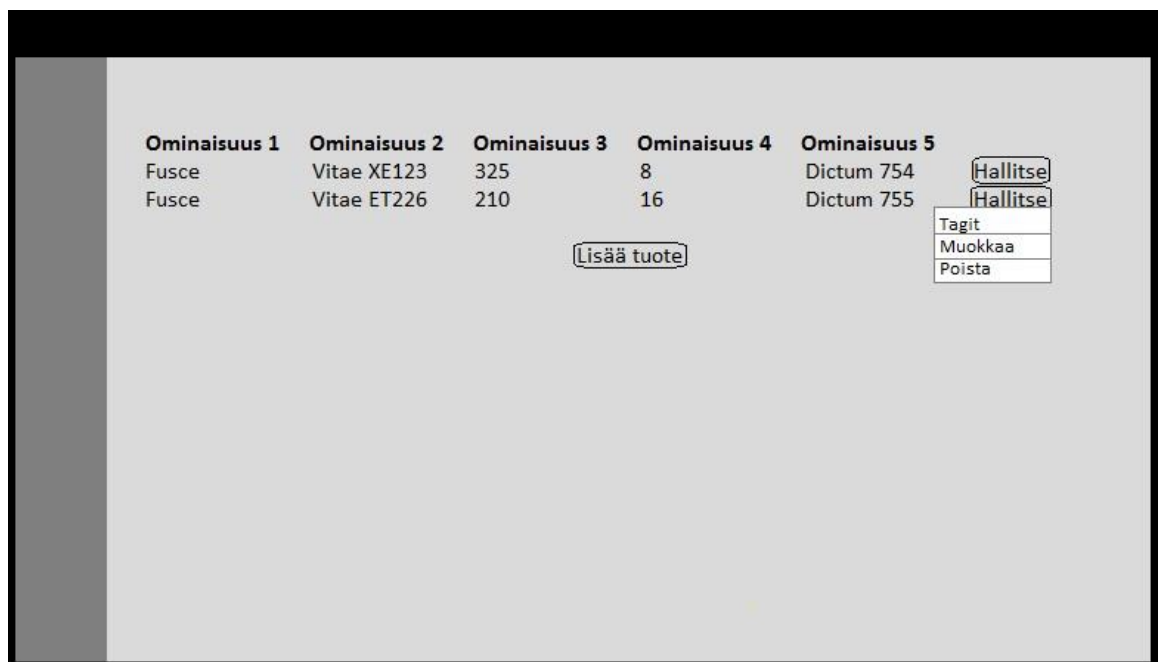
#### 4.4.1 Käyttöliittymä ylläpitäjälle

Admin-sivuilla voidaan hallita komponentteja, avainsanoja sekä näiden suhteita.

Admin-sivut lisätään WordPress admin-ympäristöön liittämällä sivut generoiva funktio WordPressin 'admin\_menu' -toimintakoukkuun. Tällöin WordPress osaa admin-ympäristön valikoita luodessaan lisätä liitännäisen määrittelemät osiot valikkoihin. WordPressin funktioilla voidaan määrittää, mihin osaan valikoita sivut lisätään.

Tässä projektissa luotu liitännäinen luo itsellensä oman valikko-osionsa ja lisää sen alle kullekin komponenttityypille oman aliosionsa. Avainsanojen hallinta löytyy liitännäisen pääosiosta. Osiot generoidaan konfiguraatiotiedostoa hyödyntäen.

Ennen hallintasivujen luomista tehtiin rautalankamallit siitä, miltä lopullinen sivusto tulee näyttämään. Nämä mallit on esitetty kuvissa 5 ja 6.



Ominaisuus 1	Ominaisuus 2	Ominaisuus 3	Ominaisuus 4	Ominaisuus 5	
Fusce	Vitae XE123	325	8	Dictum 754	Hallitse
Fusce	Vitae ET226	210	16	Dictum 755	Hallitse

Lisää tuote

Tagit  
Muokkaa  
Poista

Kuva 5. Ylläpitosivun rautalankamalli

**Ominaisuus**  
Fusce  
Fusce

**Muokkaa tuotetta**

Ominaisuus 1  
Fusce

Ominaisuus 2  
Vitae XE123

Ominaisuus 3  
325

Ominaisuus 4  
8

Ominaisuus 5  
Dictum 764

Tallenna

tse  
tse

Kuva 6. Ylläpitosivun modaali-ikkunan rautalankamalli

Komponenttien ja avainsanojen tiedot haetaan liitännäisen tietokantatauluista WordPressin wpdb-luokan funktiota käyttäen. wpdb on WordPress-tietokannan kanssa kommunikointiin tarkoitettu abstraktio-objekti (WordPress.org b). Se sisältää erilaisia funktioita, joiden avulla voidaan suorittaa SQL-kyselyjä WordPress-tietokantaan. Komponenttien sivuille luodaan taulukko, johon listataan komponentit. Jokaisen komponentin kohdalla on nappi, josta aukeaa pudotusvalikko. Pudotusvalikko sisältää painikkeet komponentin tietojen muuttamiselle, komponentin poistamiselle sekä siihen liittyvien avainsanojen hallitsemiselle. Lisäksi sivulla on painike komponenttien lisäämistä varten.

Toteutunut komponenttien hallintasivun käyttöliittymä näkyy kuvassa 7.

WordPress 4.7.3 is available! [Please update now.](#)

GPU

Manufacturer	Model	Name	Length (mm)	Clock Speed (MHz)	Boost Clock Speed (MHz)	Memory (GB)	Memory Speed (MHz)	Boost Memory Speed (MHz)	URL	
ASUS	GTX 1080 Ti	ROG-STRIX-GTX1080TI-O11G-GAMING	298	1569	1683	11	11010	11100	<a href="https://www.asus.com/ROG-Republic-Of-Gamers/ROG-STRIX-GTX1080TI-O11G-GAMING/">https://www.asus.com/ROG-Republic-Of-Gamers/ROG-STRIX-GTX1080TI-O11G-GAMING/</a>	Manage
ASUS	RX 480	ROG STRIX-RX480-O8G-GAMING	298	1310	1330	8	8000	8000	<a href="https://www.asus.com/Graphics-Cards/ROG-STRIX-RX480-O8G-GAMING/">https://www.asus.com/Graphics-Cards/ROG-STRIX-RX480-O8G-GAMING</a>	Manage

Add Item

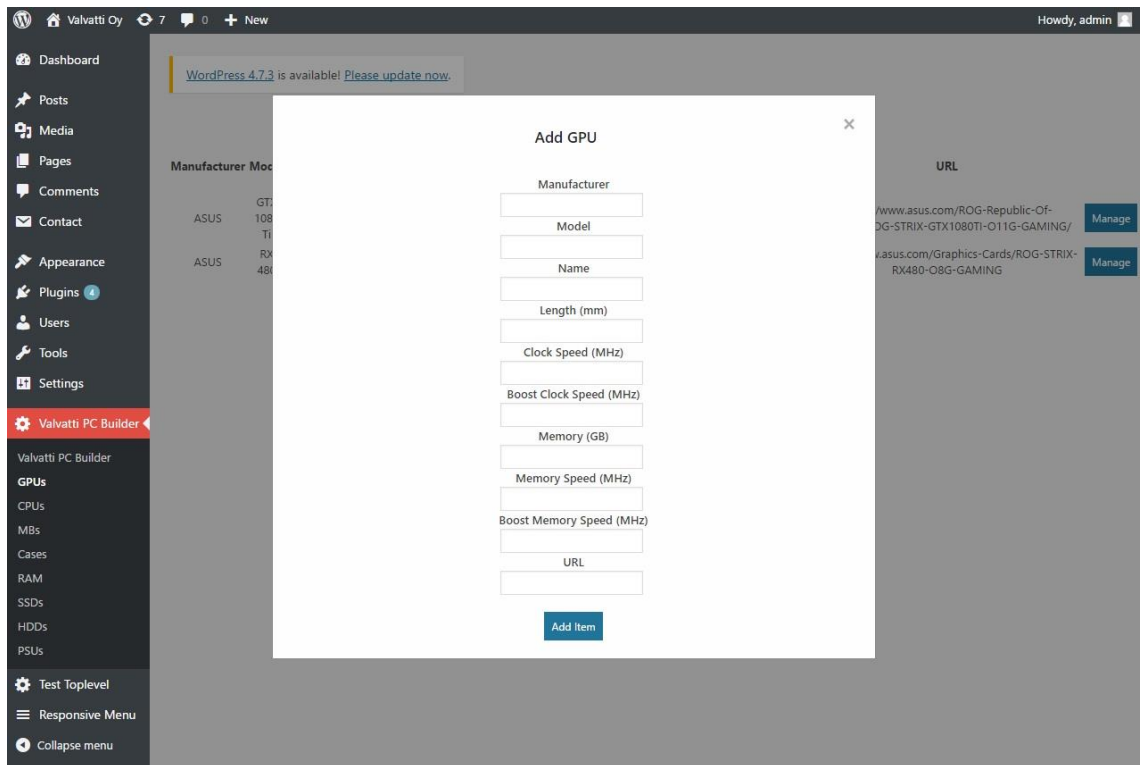
tags  
edit  
delete

Valvatti PC Builder

- Valvatti PC Builder
- GPUs
- CPUs
- MBs
- Cases
- RAM
- SSDs
- HDDs
- PSUs
- Test Toplevel
- Responsive Menu
- Collapse menu

Kuva 7. Kuvankaappaus ylläpitosivuston näytönohjain sivusta

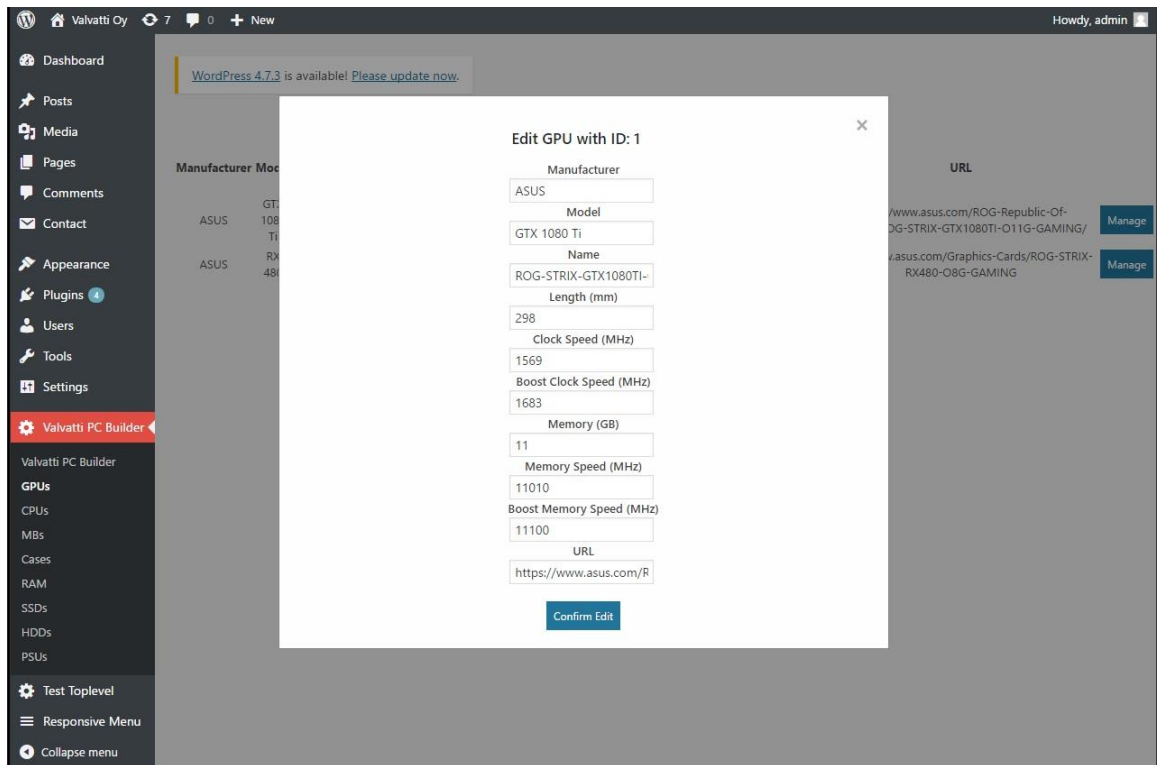
Kaikki pudotusvalikkojen painikkeet sekä komponenttien lisäyspainike avaavat sivulle modaalin (sivun pääikkunan eteen aukeava alaikkuna), josta löytyy kyseisen toimenpiteen käyttöliittymä. Aktiiviset modaali-ikkunanäkymät ovat erikseen esillä seuraavissa kuvissa 8-13.



Kuva 8. Kuvankaappaus komponentin lisäysmodaali-ikkunasta

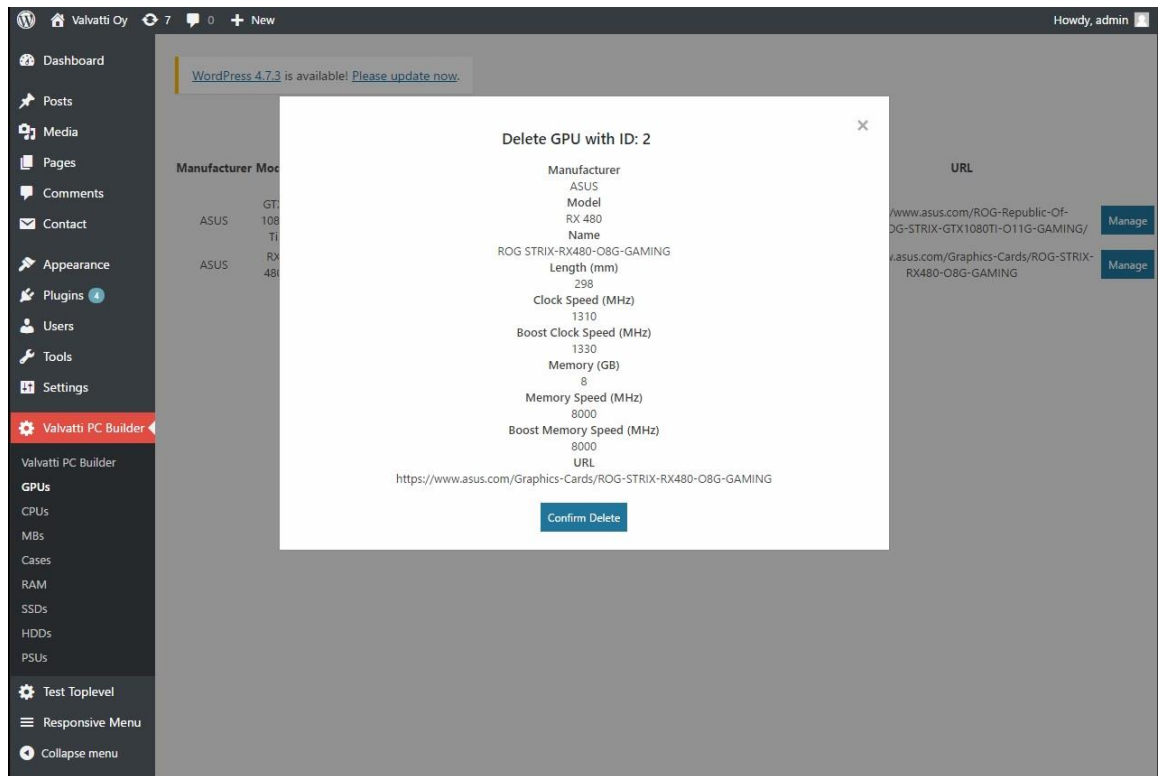
Komponentin lisäysmodaali koostuu komponentin eri ominaisuuksien otsikoista ja niiden alla olevista syötekentistä. Lisäksi modaali-ikkunassa on painike tuotteen lisäämiseksi ja oikeasta yläkulmasta löytyy rasti modaalin sulkemiseen. Lisätäkseen komponentin ylläpitäjä täyttää syötekentät ja painaa tämän jälkeen 'Add Item'-painiketta.





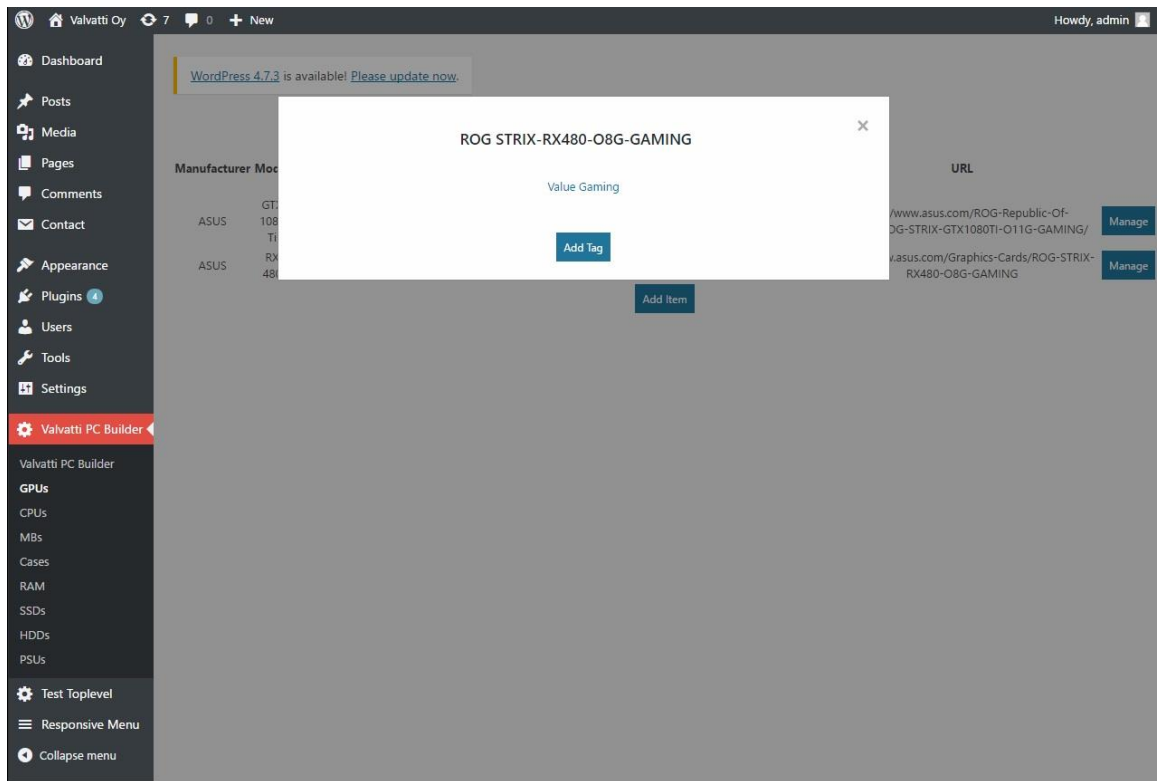
Kuva 9. Kuvankaappaus komponentin muokkausmodaali-ikkunasta

Komponentin muokkausmodaali koostuu samoista elementeistä kuin komponentin lisäysnäkymäkin, erona on se, että syötekentät ovat valmiiksi täytetty komponentin tiedoilla. Lisäksi modaali-ikkunasta löytyy painike tuotteen muokkauksen hyväksymiseksi ja yläkulmasta rasti modaalin sulkemiseksi. Muokatakseen komponentin tietoja ylläpitäjä muuttaa syötekentistä haluamansa osiot ja painaa tämän jälkeen 'Confirm Edit'-painiketta.



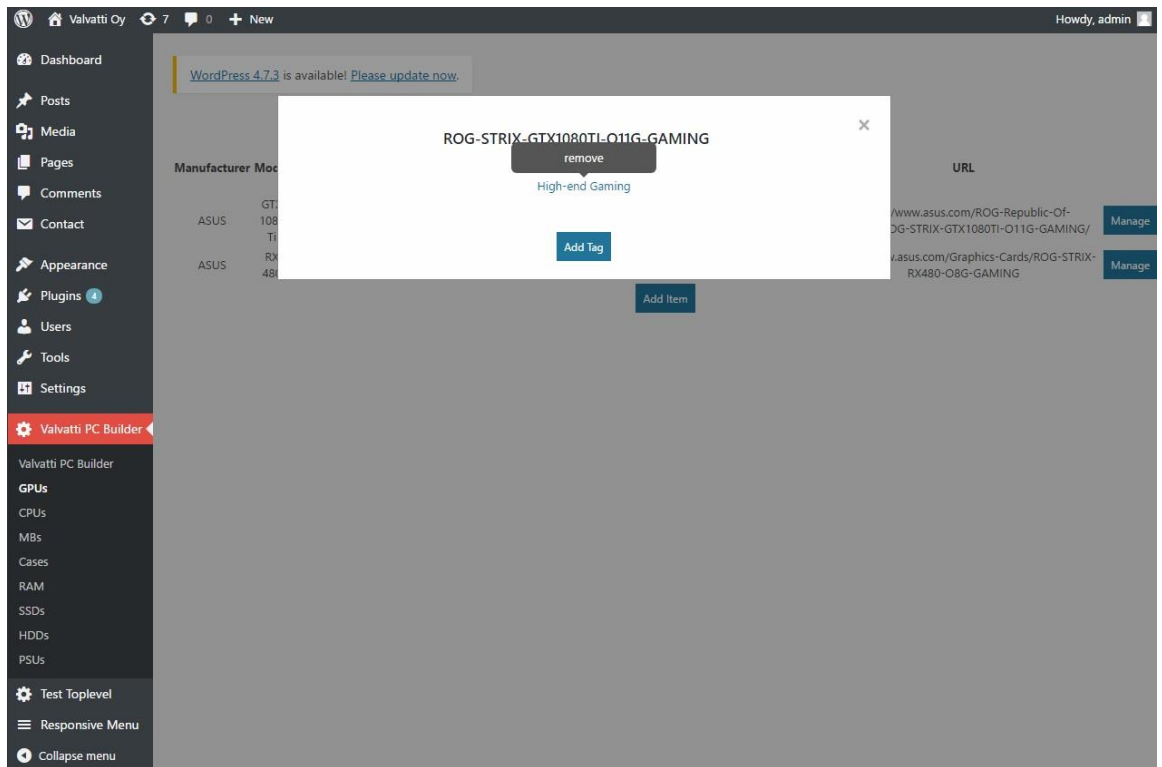
Kuva 10. Kuvankaappaus komponentin poistomodaali-ikkunasta

Komponentin poistomodaali sisältää myös komponentin eri ominaisuuksien otsikot, mutta syötekenttien sijaan niiden alla on yksinkertaisesti esillä komponentin tiedot. Modaalikkunasta löytyy myös painike tuotteen poiston vahvistamiseksi sekä modaalin sulkeva rasti. Poistaakseen komponentin ylläpitäjä tarkistaa valitsemansa komponentin oikeaksi ja painaa 'Confirm Delete'-painiketta.



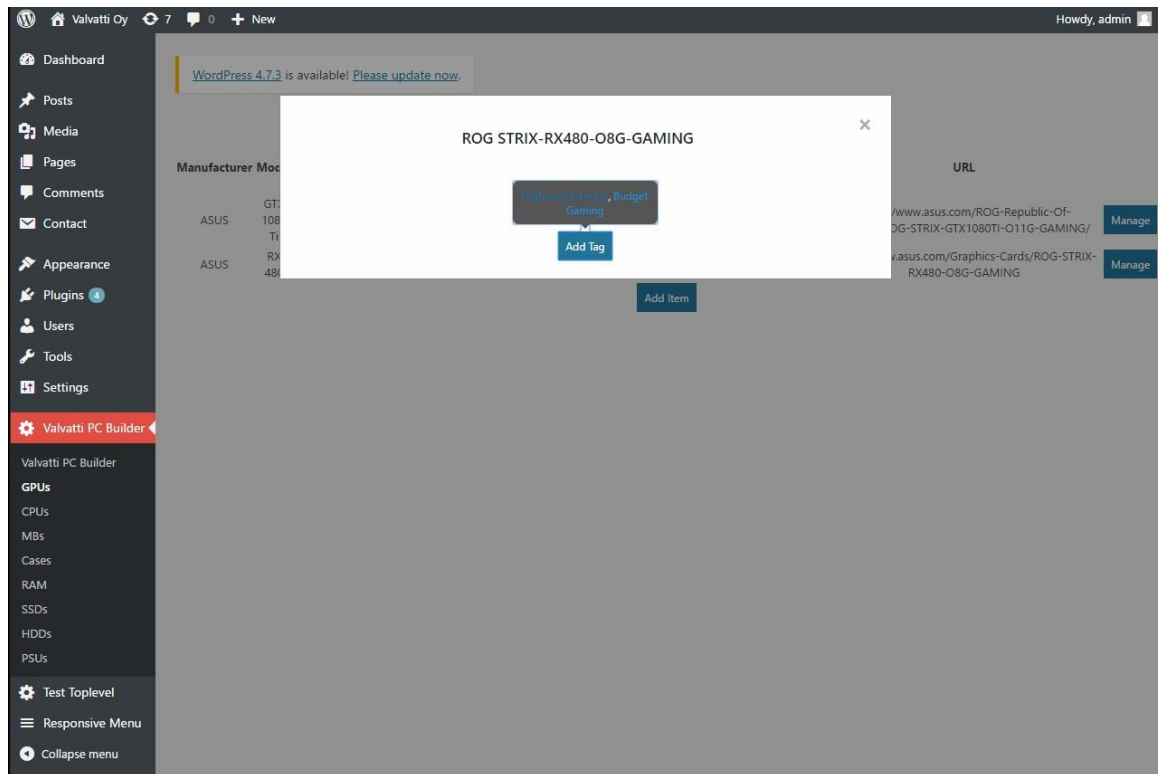
Kuva 11. Kuvankaappaus komponenttiin liitettyjen avainsanojen hallintamodaali-ikkunasta

Komponenttiin liittyvien avainsanojen hallintamodaali sisältää listattuna komponenttiin liitettyt avainsanat sekä painikkeen avainsanojen lisäämiseksi. Lisäksi modaalissa on komponentin nimi ja modaalikkunan oikeassa yläkulmassa on tutusti modaalin sulkeva rasti. Avainsanojen sekä avainsanojen lisäyspainikkeen painaminen avaavat ponnahdusikkunan. Ponnahdusikkunat ovat esillä kuvissa 12 ja 13.



Kuva 12. Kuvankaappaus komponenttiin liitetyn avainsanan poistamisen ponnahtusikkunasta modaalissa

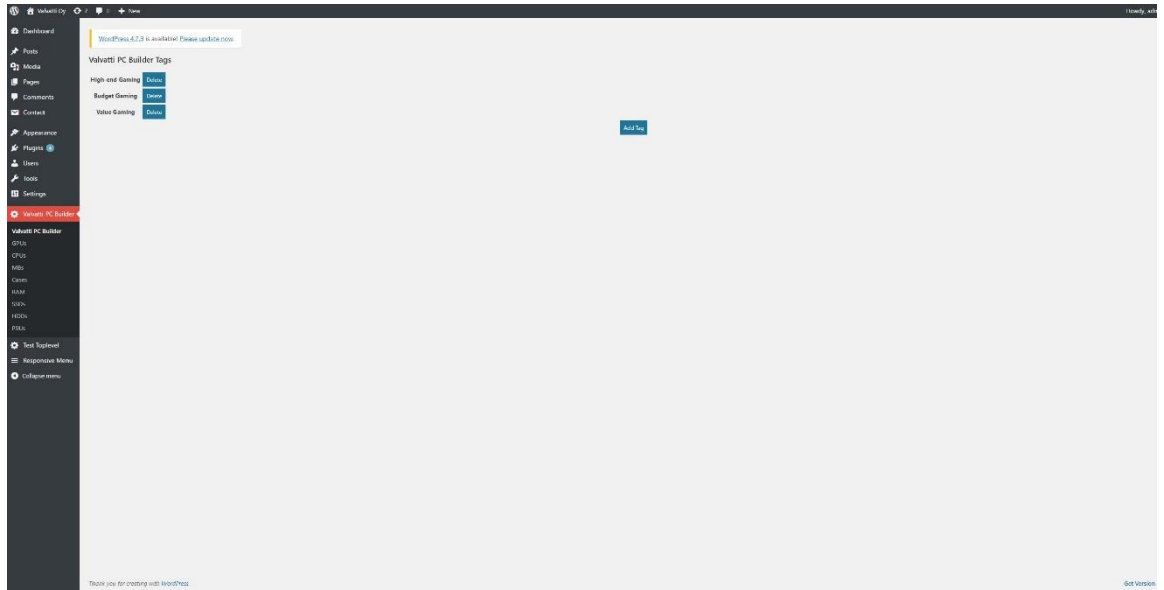
Avainsanojen painaminen avaa ponnahtusikkunan, jossa on painike avainsanan poistamiseksi kyseiseltä komponentilta. Ponnahtusikkunan saa suljettua klikkaamalla sen ulkopuolelle. Painamalla 'remove'-painiketta avainsana poistetaan komponenttiin liittyvien avainsanojen listasta, tällöin myös ponnahtusikkuna sulkeutuu.



Kuva 13. Kuvankaappaus komponenttiin liitettävien avainsanojen lisäyksen ponnahdusikkunasta modaalissa

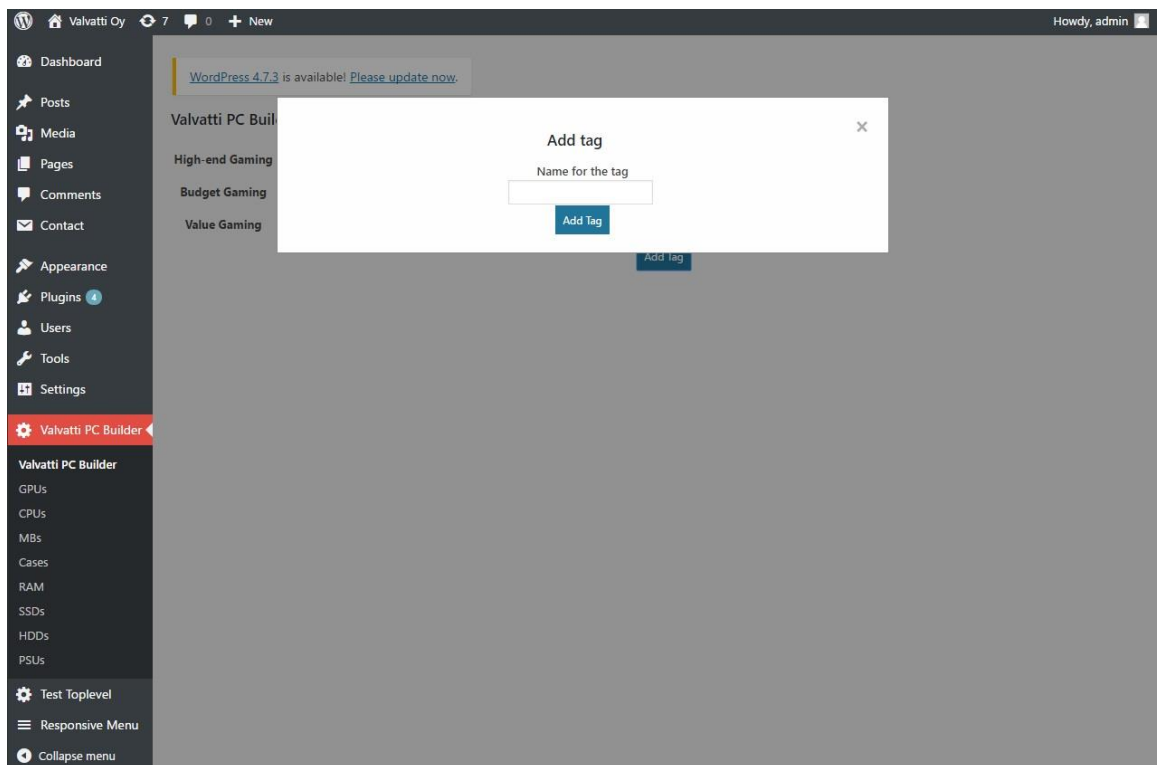
Avainsanojen lisäyspainike avaa myös ponnahdusikkunan. Siinä on listattuna lisättävänä olevat avainsanat. Lisättävien avainsanojen lista on yksikertaisesti lista kaikista olemassa olevista avainsanoista, joita ei vielä ole liitetty kyseiseen komponenttiin. Ponnahdusikkunassa olevaa avainsanaa painamalla voi liittää avainsanan kyseiseen komponenttiin. Ponnahdusikkuna sulkeutuu, kun jotain listan avainsanaa painetaan ja se lisätään komponenttiin liittyvien avainsanojen listaan. Ponnahdusikkuna suljetaan myös silloin, kun ylläpitäjä klikkaa ponnahdusikkunan ulkopuolelle.

Liitännäisen ylläpitosivujen etusivuna toimii avainsanojen hallintasivu. Avainsanasivulta löytyy listattuna olemassa olevat avainsanat sekä painikkeet avainsanojen lisäämistä ja poistamista varten. Avainsanasivun painikkeet avaavat myös modaali-ikkunan. Avainsanasivu näkyy kuvassa 14. Avainsanasivun modaalit on esitelty kuvissa 15 ja 16.



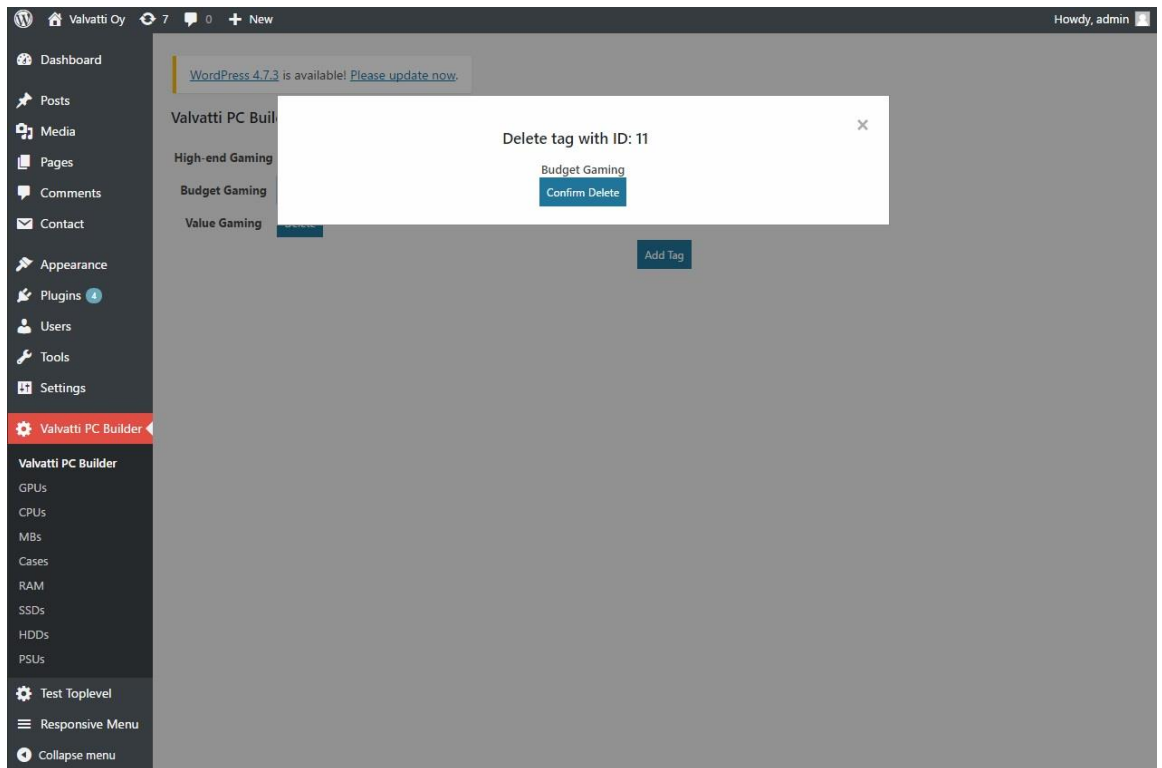
Kuva 14. Kuvankaappaus avainsanasivulta

Avainsanasivulla on lista avainsanoista ja näiden perässä kunkin avainsanan poistoon tarkoitettu painike. Lisäksi listan alta löytyy avainsanan lisäämiseksi painike.



Kuva 15. Kuvankaappaus avainsanan lisääsmodaali-ikkunasta

Avainsanan lisäysmodaali sisältää syötekentän avainsanan nimelle sekä painikkeen sen lisäämiseksi. Oikeasta yläkulmasta löytyy rasti modaalin sulkemiseen. Lisätäkseen avainsanan ylläpitäjä täyttää syötekentän ja painaa tämän jälkeen 'Add Tag'-painiketta.



Kuva 16. Kuvankaappaus avainsanan poistomodaali-ikkunasta

Avainsanan poistomodaali-ikkuna sisältää kyseisen avainsanan nimen sekä tunnusteen. Lisäksi modaalissa on painike poistamisen vahvistamiseksi sekä rasti modaali-ikkunan sulkemista varten. Poistaakseen avainsanan ylläpitäjä tarkistaa valitsemansa avainsanan tiedot ja painaa 'Confirm Delete'-painiketta.

Sivuilla hyödynnetään AJAX-tekniikkaa dynaamisen sisällön mahdollistamiseksi (esimerkiksi modaalin sisältä haetaan dynaamisesti). Admin-ympäristöön luotavia sivuja varten on liitännäisessä oma JavaScript-tiedosto sekä CSS-tiedosto sivujen toiminnallisuuden ja muotoilun toteuttamiseksi. JavaScript-koodin käyttöönottamiseksi WordPressin ylläpitosivuilla käytetään funktiota `wp_enqueue_script`. Vastaavasti tyyliohjetiedoston käyttöönottoon on funktio `wp_enqueue_style`. Näiden funktioiden kutsuminen liitetään 'admin\_enqueue\_scripts' -toimintakoukkuun, jolloin WordPress osaa ottaa koodin käyttöön ylläpitosivustoillaan.

Funktiot `wp_enqueue_script` ja `wp_enqueue_style` näkyvät kuvissa 17 ja 18.

```
wp_enqueue_script( string $handle, string $src = '', array $deps = array(),
    string|bool|null $ver = false, bool $in_footer = false )
```

Kuva 17. wp\_enqueue\_script-funktio (WordPress.org d)

```
wp_enqueue_style( string $handle, string $src = '', array $deps = array(),
    string|bool|null $ver = false, string $media = 'all' )
```

Kuva 18. wp\_enqueue\_style funktio (WordPress.org e)

#### 4.4.2 Käyttöliittymä asiakkaalle

Yrityksen verkkosivuille tulevan loppukäyttäjän (asiakkaan) käyttöliittymä on toteutettu siten, että sen voi lisätä halutulle verkkosivulle kutsumalla liitännäisen tarjoamia funktioita.

Rautalankamallit loppukäyttäjän käyttöliittymästä näkyvät kuvista 19 ja 20.

ETUSIVU PALVELUT HINNASTO YHTEYSTIEDOT PC-BUILDER

PC BUILDER

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin posuere diam ex, a malesuada ante ultricies in. Vivamus mattis lectus eu sapien finibus, eu egestas erat viverra.

Suodata:  avainsana 1  avainsana 2  avainsana 3

Komponentti 1: lisää

Komponentti 2: lisää

Komponentti 3: lisää

Komponentti 4: lisää

Nimi:

Sähköposti:

Puhelinnumero:

Lähetä tilaus

Kuva 19. Rautalankamalli loppukäyttäjän käyttöliittymästä, tyhjä



ETUSIVU PALVELUT HINNASTO YHTEYSTIEDOT PC-BUILDER

PC BUILDER

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin posuere diam ex, a malesuada ante ultricies in. Vivamus mattis lectus eu sapien finibus, eu egestas erat viverra.

Suodata:  avainsana 1  avainsana 2  avainsana 3

Komponentti 1: Morbi dictum 124XER [lisätietoja](#) [poista](#)  
 Komponentti 2: Odio et hendrerit 6GB [lisätietoja](#) [poista](#)  
 Komponentti 3: Felis ex 1Tt [lisätietoja](#) [poista](#)  
 Komponentti 4: Dignissim ipsum TG3241X [lisätietoja](#) [poista](#)

Nimi: Assi Asiakas  
 Sähköposti: assi@asiakas.fi  
 Puhelinnumero: 040 123 456 78

[Lähetä tilaus](#)

Kuva 20. Rautalankamalli loppukäyttäjän käyttöliittymästä, täytetty

Käyttöliittymään generoidaan konfiguraatitiedoston pohjalta taulukko, josta löytyy paikka jokaiselle komponenttityypille. Aluksi taulukko sisältää vain painikkeet komponenttien lisäystä varten. Käyttöliittymään generoidaan myös valintaruudut avainsanoille hakemalla avainsanat tietokannasta. Komponentin lisäystä varten olevia painikkeita klikkaamalla aukeaa modaali-ikkuna, joka sisältää listan kyseisen tyypin komponenteista.

Toteutunut käyttöliittymä esitellään kuvissa 21 - 23.

VALVATTI

TIETOTEKNIKKAPALVELU

ETUSIVU PALVELUT HINNASTO YHTEYSTIEDOT PC BUILDER

## PC BUILDER

Työkalu räätälöidyn PC:n tilaamiseen.

Työkalu auttaa sinua valitsemaan sinulle sopivat komponentit käyttötarkoituksesi mukaan.

SUODATA KOMPONENTTEJA:

High-end Gaming  Budget Gaming  
 Value Gaming

Näytönohjain (GPU) ASUS ROG-STRIX-GTX1080TI-O11G-GAMING  
 Lisätietoja Poista

Proessori (CPU) Lisää

Emolevy Lisää

Kotelo Lisää

Välimuisti (RAM) Lisää

Kovalevy (SSD) Lisää

Kovalevy (HDD) Lisää

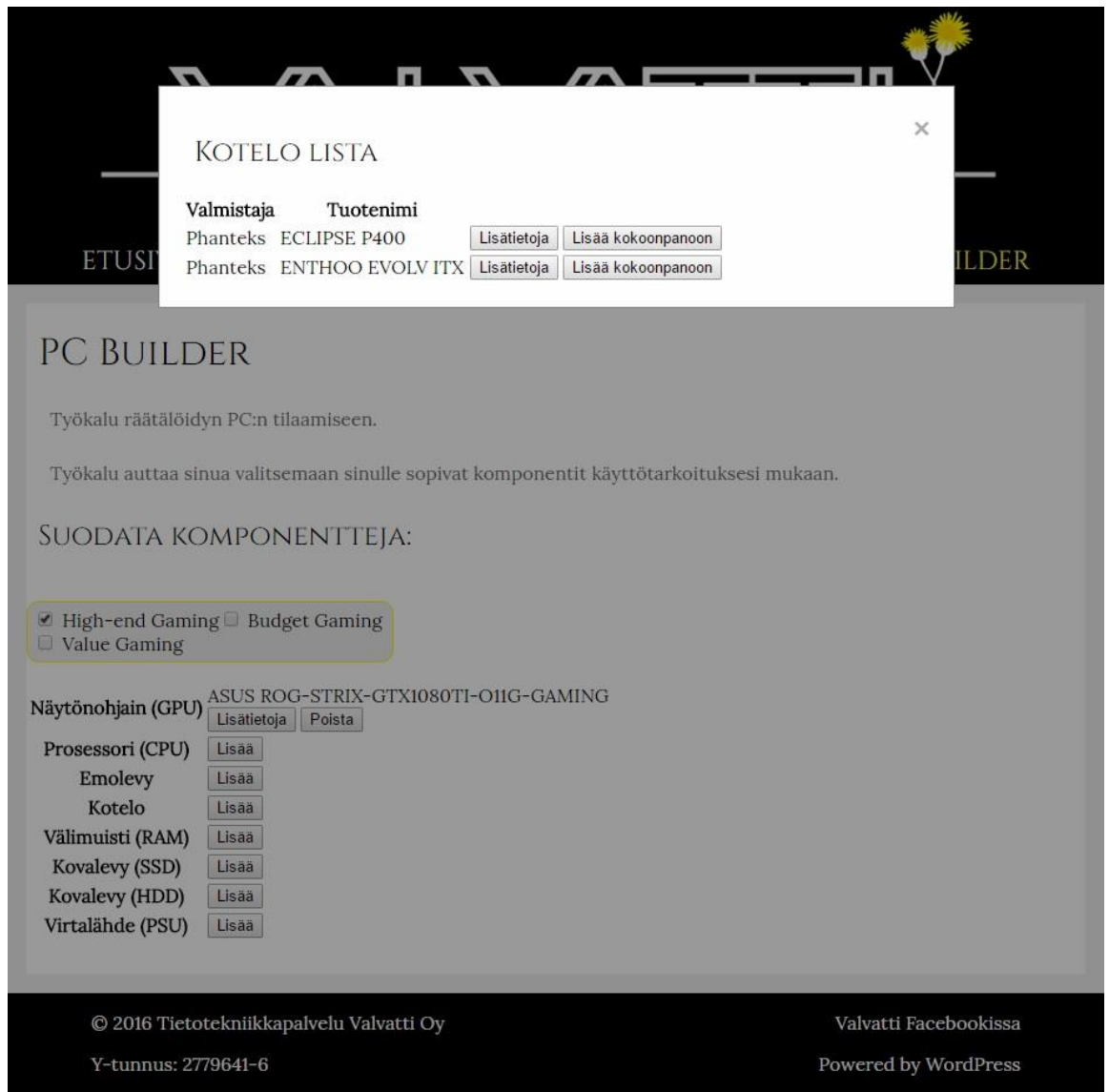
Virtalähde (PSU) Lisää

© 2016 Tietotekniikkapalvelu Valvatti Oy  
 Y-tunnus: 2779641-6

Valvatti Facebookissa  
 Powered by WordPress

Kuva 21. Kuvankaappaus loppukäyttäjän käyttöliittymästä

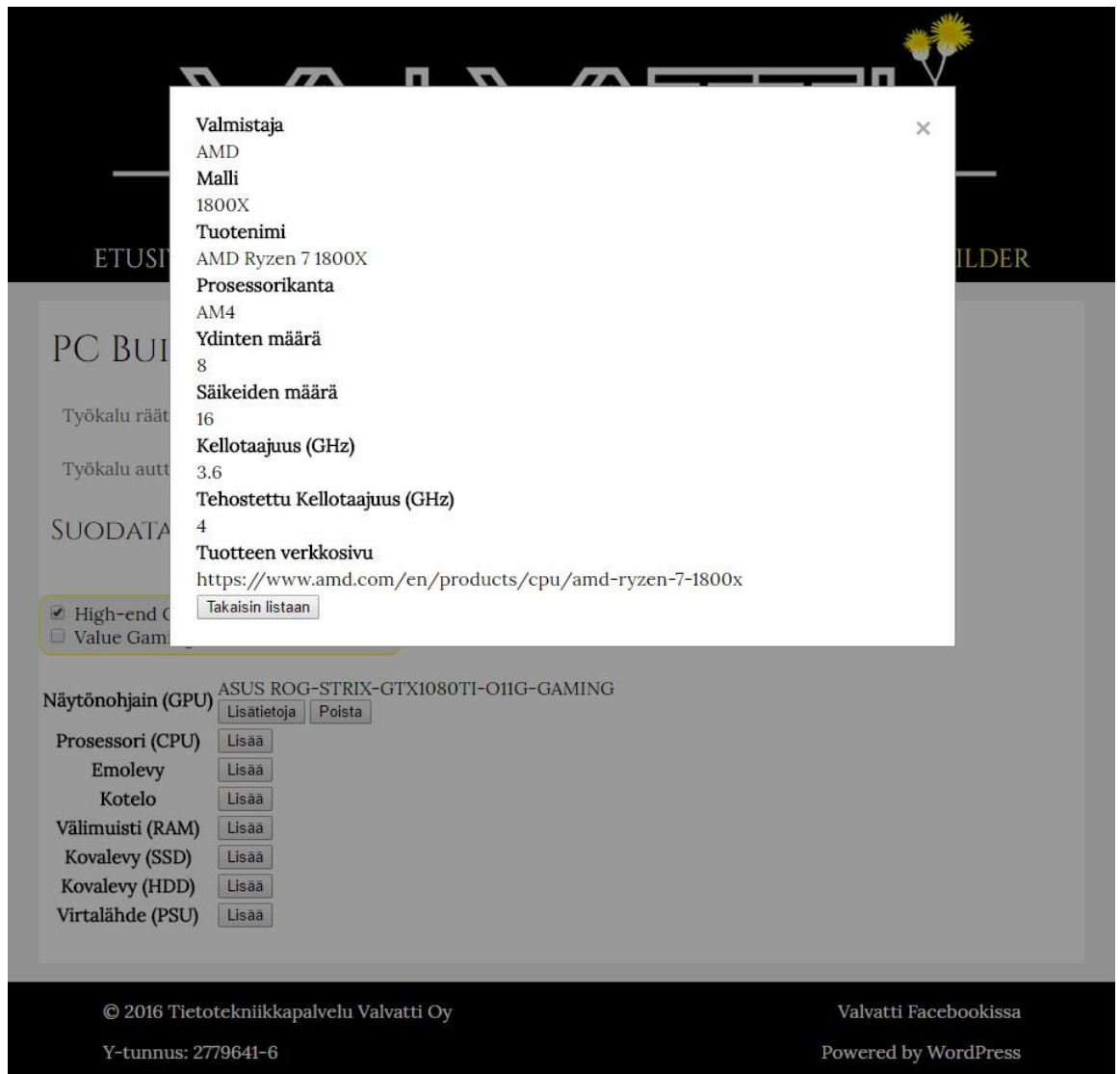
Kun komponentti lisätään kokoonpanoon, tuotteen alle lisätään 'Lisätietoja'- ja 'Poista'-painikkeet. Painikkeiden avulla kyseisen tuotteen saa poistettua kokoonpanosta tai sen tietoja voidaan tarkastella. Jos tuote poistetaan, näkyviin palautuu 'Lisää'-painike.



Kuva 22. Kuvankaappaus loppukäyttäjän käyttöliittymän modaali-ikkunasta

Tuotteet haetaan liittäen tuotetyypin vastaavasta tietokantataulusta. Listanäkymä sisältää valmistajan, tuotenimikkeet sekä painikkeet tuotteen lisätietojen tarkastelemiseen ja tuotteen lisäämiseksi kokoonpanoon. Modaali-ikkunan yläkulmassa on myös rasti modaalin sulkemiseksi. 'Lisätietoja'-painike päivittää modaali-ikkunan sisällön tuotetieto-näkymään, josta voidaan painikkeella palata takaisin listaan. 'Lisää kokoonpanoon'-painike sulkee modaali-ikkunan ja päivittää kyseisen komponentin kentän taulukosta näyttämään valittu tuote.

Tuotelistoissa näytettävät tuotteet suodatetaan jo valittujen komponenttien yhteensopivuuden sekä valittujen avainsanojen mukaan. Komponenttien suodatusta käydään tarkemmin läpi luvussa 4.5.



Kuva 23. Kuvankaappaus lisätietojamodaali-ikkunasta

'Lisätietoja'-painikkeesta avautuva tuotetiedot näyttävä modaali-ikkuna sisältää komponentin eri ominaisuuksien otsikot sekä niiden alla näytetään itse komponentin tiedot. Tietojen alapuolella modaali-ikkunassa on painike, josta pääsee takaisin komponenttien listaan. Lisäksi oikeasta yläkulmasta löytyy rasti modaalin sulkemiseksi.

Käyttöliittymän toteutuksessa hyödynnetään AJAX-tekniikkaa dynaamisen sisällön mahdollistamiseksi (modaali-ikkuna sekä komponenttien valintakentät). Käyttöliittymän toiminnallisuuden mahdollistamiseksi on liitännäisessä oma JavaScript-tiedosto. Kuten ylläpitosivuillakin, koodin käyttöönottamiseksi käytetään WordPressin funktiota `wp_enqueue_script`. Toisin kuin ylläpitosivuston tapauksessa, itse verkkosivuille JavaScript-koodin lataava funktiokutsu liitetään `'wp_enqueue_scripts'` -toimintakoukkuun.

## 4.5 Komponenttien suodatus

Asiakkaan selatessa kokoonpanoonsa lisättäviä komponentteja näytettävien komponenttien listaa suodatetaan kahdesti. Nämä käydään tarkemmin läpi seuraavissa alaluvuissa.

### 4.5.1 Avainsanasuodatus

Käyttöliittymässä olevia avainsanojen valintaruutuja valitsemalla asiakas voi valita, mitä avainsanoja sisältäviä komponentteja sovellus listaa.

Ennen kuin komponenttilista näytetään sovelluksen modaali-ikkunassa, suodatetaan se siten, että vain valittuja avainsanoja vastaavat komponentit ovat listattuna. Komponenttityypin listalle suoritetaan yksinkertainen tarkistus, jokaisesta tuotteesta käydään läpi siihen liitetyt avainsanat ja katsotaan, vastaako mikään niistä yhtäkään valituista avainsanoista. Mikäli näin ei ole, poistetaan kyseinen tuote näytettävästä listasta.

### 4.5.2 Yhteensopivuussuodatus

Sovelluksella PC-kokoonpanoa rakennettaessa eri komponenttilistoissa näytetään vain tuotteet, jotka ovat yhteensopivia jo kokoonpanoon lisättyjen tuotteiden kanssa. Yhteensopivuutta tarkasteltaessa on paljon erilaisia asioita, jotka tulee ottaa huomioon valittavasta ja jo valituista komponenteista riippuen.

Eri komponenttien eri ominaisuuksien riippuvuussuhteet on listattu liitännäisessä omaan konfiguraatitiedostoonsa. Konfiguraatitiedosto sisältää listattuna jokaisen riippuvuussuhteen. Listaan riippuvuudet on merkitty siten, että jokainen listan jäsen sisältää komponenttien tyypit, komponenteissa toisistaan riippuvien ominaisuuksien nimet sekä sen operaation, jolla ominaisuuksien yhteensopivuus määritetään.

Esimerkiksi näytönohjaimella ja tietokoneen kotelolla on riippuvuussuhde, sillä näytönohjaimen tulee mahtua koteloon. Tämän riippuvuussuhteen merkintä konfiguraatitiedostossa on havainnollistettu alla.

```
[ 'GPU', 'Case', 'gpulength', 'gpumaxlength', '<=' ]
```

Esimerkkitapauksessa tulee näyttöohjaimen pituuden (gpulength) olla pienempi tai yhtä suuri kuin kotelon varaaman tilan näyttöohjaimelle (gpumaxlength).

Komponentteja listatessa käydään siis läpi komponenttityypin riippuvuussuhteet jo valittuihin komponentteihin ja suoritetaan konfiguraatitiedostossa määritellyt vertailuoperaatiot. Mikäli jokin suoritettavista operaatioista palautuu epätotena, poistetaan kyseinen komponentti näytettävien tuotteiden listasta.

Sovelluksen prototyypissä toistaiseksi tehtävät tarkistukset ovat: näyttöohjaimen, virtalähteen ja emolevyn mahtuminen koteloon sekä prosessorin ja emolevyn saman kantaisuus. Uusien perusoperaatioiden avulla tehtävien tarkistusten lisääminen liitännäiseen on helppoa siihen tarkoitettua konfiguraatitiedostoa muokkaamalla.

## 5 Arviointi

Projektissa saavutettiin asetetut minimitalvoitteet. Projektin tuotoksena aikaansaatu prototyyppi tarjoaa mahdollisuuden jatkokehittää yritykselle sopiva kustomoitujen PC-koneiden kasaamiseen ja myymiseen tarkoitettu verkkosovellus.

Oppimisen kannalta projekti oli mainio ja tietotaito WordPress-liitännäisten kehittämisestä ja verkko-ohjelmoinnista yleensä on kasvanut huomattavasti. Projektin aikana opin, miten luodaan WordPress-liitännäisiä ja miten niitä voidaan integroida jo olemassa oleville WordPress-sivustoille.

Lisäksi opin käyttämään sujuvammin HTML-, CSS-, PHP- ja SQL-kieliä. Erityisesti JavaScriptin ja AJAXin käyttö opetti paljon, sillä en ollut aikaisemmin ollut niiden kanssa juurikaan tekemisissä. Jatkossa osaan luoda interaktiivisia ja sisällöltään dynaamisia verkkosivuja.

Suurin haaste projektissa oli uuden opittavan paljous ja siitä johtuen alkuperäisessä aikataulussa ei aivan pysytty. Joitakin alkuperäisen suunnitelman ominaisuuksista ja joidenkin ominaisuuksien laajuutta jouduttiin karsimaan tämän oppinäytetyön osalta.

Projektin lopuksi luotu sovellus käytiin läpi yhdessä yrityksen edustajan kanssa. Yrityksen puolesta prototyyppi vastasi heidän toiveitaan suurilta osin, vaikka joitakin osa-alueita jäi puuttumaan. Kuitenkin kaikki mitä sovellukseen saatiin tehtyä, sopi yritykselle hyvin eikä nykyisissä ominaisuuksissa ollut mitään, mitä olisi tarvinnut poistaa tai suuremmin muokata. Yritys pystyy jatkamaan sovelluksen kehittämistä ja ominaisuuksien laajentamista itsenäisesti tarpeen mukaan.

Yrityksen edustajan mukaan sekä yritykselle tarkoitettu ylläpitokäyttöliittymä, että asiakkailla näkyvä käyttöliittymä olivat sopivan selkeitä. Asetettujen tavoitteiden mukaisesti ylläpito tehtiin helpoksi siten, että liitännäisen ylläpitosivut ovat osana WordPressin admin-ympäristöä.

## 6 Yhteenveto

Projektissa luotiin verkkosovellus yrityksen PC-myyntiä varten. Sovellus toteutettiin WordPress-liitännäisenä hyödyntäen erilaisia ohjelmointikieliä, joista pääpaino oli PHP:llä sekä JavaScriptillä. Projektin aikana kohdatuista ongelmista huolimatta lopputuloksena oli toimiva prototyyppi, josta yrityksen on mahdollista jatkaa verkkosovelluksen kehittämistä.

Ongelmat liittyivät suurilta osin uusien asioiden kohtaamiseen ja sopivien ratkaisujen löytämiseen. Heti alussa ongelmaksi muodostui oikean lähestymistavan valitseminen, sillä kokemusta vastaavista projekteista ei ollut. Lopulta päädyttiin luomaan sovellusta WordPress-liitännäisenä, koska sillä pystyi tekemään kaiken tarvittavan ja sen pystyi helposti liittämään jo olemassa olevalle yrityksen WordPress-pohjaiselle sivustolle.

Lisäksi ongelmia tuli ja aikaa kului selvittäessä sekä mietittäessä, miten tarvittavat tiedokannat kannatti luoda.

Näiden alkuongelmien jälkeen projekti lähti sujumaan paremmin, vaikka silloin tällöin vastaan tulikin kohtia, joiden selvittämiseen kului enemmän aikaa. Näistä johtuen ennen projektin aloittamista laaditusta aikataulusta jäätin hieman jälkeen ja lopulta jouduttiin jättämään osa halutuista ominaisuuksista pois.

Tuloksena oli yrityksen toivomalle verkkosovellukselle hyvä pohja ja prototyyppi. Projektin päättyessä verkkosovelluksella pystyi jo luomaan PC-kokonaisuuksia toivottujen kriteerien mukaan. Uusien komponenttien lisääminen ja vanhojen tietojen muokkaaminen onnistuivat kätevästi ylläpitosivujen kautta.

WordPress-liitännäisten käyttäminen on kätevä tapa lisätä WordPress-sivustojen toiminnallisuutta. Liitännäisillä voi erottaa uuden toiminnallisuuden perussivusta ja niitä voi siirtää halutessaan eri sivustoille.



## Lähteet

apriorit. Virtualization in Software Testing: Advantages and Disadvantages. Haettu 12.04.2017, osoitteesta <https://www.apriorit.com/qa-blog/223-virtualization-in-testing>

BuiltWith Pty Ltd. CMS technologies Web Usage Statistics. Haettu 11.04.2017, osoitteesta <https://trends.builtwith.com/cms> 15.4.2017

Chapman, Stephen 2017. What Is JavaScript? Haettu 11.04.2017, osoitteesta <https://www.thoughtco.com/what-is-javascript-2037921>

GFC Global. Computer Basics: Inside a Computer. Haettu 11.04.2017, osoitteesta <http://www.gcflearnfree.org/computerbasics/inside-a-computer/1/>

Kelly, Luke 2016. Usability: The Core Principle of Web Usability. Haettu 12.04.2017, osoitteesta <http://www.bordercrossingmedia.com/2016/11/web-usability-core-principles/>

Korpela, J. 2013. CSS – uudet mahdollisuudet. Jyväskylä: Docendo.

Nielsen, Jakob 1995. 10 Usability Heuristics for User Interface Design. Haettu 20.04.2017, osoitteesta <https://www.nngroup.com/articles/ten-usability-heuristics/>

php.net. PHP: What is PHP? Haettu 11.04.2017, osoitteesta <http://php.net/manual/en/intro-what-is.php>

Rouse, Margaret 2016. What is SQL (Structured Query Language)? Haettu 11.04.2017, osoitteesta <http://searchsqlserver.techtarget.com/definition/SQL>

Techopedia. What is a Content Management System (CMS)? Haettu 11.04.2017, osoitteesta <https://www.techopedia.com/definition/24075/content-management-system-cms>

Thakur, Dinesh. Block Diagram of Computer and Explain its Various Components. Haettu 20.04.2017, osoitteesta <http://ecomputernotes.com/fundamental/introduction-to-computer/draw-the-block-diagram-of-computer-and-explain-its-various-components>

Varying Vagrant Vagrants. Haettu 14.04.2017, osoitteesta <https://varyingvagrantvagrants.org/>

Viogaming. PC Parts For Dummies. Haettu 20.04.2017, osoitteesta <https://viogaming.neocities.org/final2.html>

W3C a. CSS Introduction. Haettu 12.04.2017, osoitteesta [https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)

W3C b. Introduction to HTML. Haettu 12.04.2017, osoitteesta [https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)

W3C c. What is AJAX? Haettu 12.04.2017, osoitteesta [https://www.w3schools.com/php/php\\_ajax\\_intro.asp](https://www.w3schools.com/php/php_ajax_intro.asp)

WordPress.org a. Haettu 11.04.2017 osoitteesta <http://wordpress.org>

WordPress.org b. Class Reference/wpdb << WordPress Codex. Haettu 15.04.2017, osoitteesta [https://codex.wordpress.org/Class\\_Reference/wp](https://codex.wordpress.org/Class_Reference/wp)

WordPress.org c. dbDelta() | Function | WordPress Developer Resources. Haettu 16.04.2017, osoitteesta <https://developer.wordpress.org/reference/functions/dbdelta/>

WordPress.org d. wp\_enqueue\_script() | Function | WordPress Developer Resources. Haettu 16.04.2017, osoitteesta [https://developer.wordpress.org/reference/functions/wp\\_enqueue\\_script/](https://developer.wordpress.org/reference/functions/wp_enqueue_script/)

WordPress.org e. wp\_enqueue\_style() | Function | WordPress Developer Resources. Haettu 16.04.2017, osoitteesta [https://developer.wordpress.org/reference/functions/wp\\_enqueue\\_style/](https://developer.wordpress.org/reference/functions/wp_enqueue_style/)