

# **Analys om att välja rätt webb- innehållshanteringssystem (CMS) för en skolas webbplats**

Fallstudie med WordPress, Drupal och Joomla!

Marissa Heikel

EXAMENSARBETE	
Arcada	
Utbildningsprogram:	Informations- och medieteknik
Identifikationsnummer:	6117
Författare:	Marissa Heikel
Arbetets namn:	Analys om att välja rätt webb-innehållshanteringssystem (CMS) för en skolas webbplats: Fallstudie med WordPress, Drupal och Joomla!
Handledare (Arcada):	Johnny Biström
Uppdragsgivare:	
<p>Sammandrag:</p> <p>Eftersom det är estimerat att mer än 50 % av webbsidorna har byggts upp med hjälp av ett webb-innehållshanteringssystem så är det nödvändigt att ställa frågan att vad är det som behövs för en lyckad webbplats? Denna avhandling är en fallstudie om att bygga och planera en webbsida för en skola som har som mål att ge ut information till användare som är lärare och som söker information och källor för att förbättra sin kompetens som lärare. Denna avhandling riktar sig in på planering, mål och krav som ställs för att bygga upp grundversionen av webbplatsen och testa vilka av de tre valda öppna källkoderna CMS (WordPress, Drupal och Joomla) som uppfyller kraven bäst.</p>	
Nyckelord:	Webb-innehållshanteringssystem, WordPress, Drupal, Joomla, uppbyggnad och planering av webbtjänst
Sidantal:	36
Språk:	Svenska
Datum för godkännande:	9.5.2017

DEGREE THESIS	
Arcada	
Degree Programme:	Information- and Media Technology
Identification number:	6117
Author:	Marissa Heikel
Title:	The analysis of choosing the right web Content Management System (CMS) for a school website: A case study with WordPress, Drupal and Joomla!
Supervisor (Arcada):	Johnny Biström
Commissioned by:	
<p>Abstract:</p> <p>As it is estimated that more than 50% of websites have been built using web content management systems it becomes important to ask the question of what is needed to successfully build a site with one? This thesis uses a case study for the planning of a site for a school whose goal is to provide the information and resources for teachers to further their own training. This thesis looks into the plan, goals and requirements needed to build the basic version of the site and tests which of three open source CMS (WordPress, Drupal and Joomla) meets the requirements the best.</p>	
Keywords:	Web content management system, WordPress, Drupal, Joomla!, building and planning a school website
Number of pages:	36
Language:	Swedish
Date of acceptance:	9.5.2017

# INNEHÅLL

<b>1</b>	<b>Introduktion och bakgrund</b>	<b>8</b>
1.1	Allmän introduktion och bakgrund	8
1.2	Syfte och mål	8
1.3	Metodik	9
<b>2</b>	<b>Förberedelser</b>	<b>10</b>
2.1	MAMP	10
2.2	WordPress 4.7.2	11
2.3	Drupal 8.0.4	12
2.4	Joomla! 3.6.5	13
<b>3</b>	<b>Fallstudie – VALTERI skola</b>	<b>14</b>
3.1	Bakgrund och information	15
3.2	Plan – mål och krav	15
3.2.1	Strukturkraven	15
<b>4</b>	<b>Analys och resultat</b>	<b>17</b>
4.1	Sidorna – skapa statiskt innehåll	17
4.1.1	WordPress	17
4.1.2	Drupal	17
4.1.3	Joomla!	18
4.2	Menyer och sidofält	18
4.2.1	WordPress	19
4.2.2	Drupal	19
4.2.3	Joomla!	20
4.3	Flerspråkig inställning	20
4.3.1	WordPress	21
4.3.2	Drupal	22
4.3.3	Joomla!	23
4.4	Skapa händelsesidor och händelsekalender	24
4.4.1	WordPress	24
4.4.2	Drupal	26
4.4.3	Joomla!	27

<b>4.5</b>	<b>Användargränssnittet och erfarenhet .....</b>	<b>28</b>
4.5.1	WordPress .....	29
4.5.2	Drupal.....	30
4.5.3	Joomla!.....	31
<b>5</b>	<b>Slutsatser .....</b>	<b>32</b>
<b>Källor</b>	<b>.....</b>	<b>34</b>
<b>Bilagor</b>	<b>.....</b>	<b>36</b>

## Figurer

Figur 1. Screenshot av MAMP.....	10
Figur 2. Screenshot från MAMP WebStart sida, vart man kommer åt phpMyAdmin kontroll.....	11
Figur 3. Screenshot av WordPress admin kontroll panel.....	12
Figur 4. Screenshot av Drupal kontrollpanel .....	13
Figur 5. Screenshot av Joomla! kontrollpanel.....	14
Figur 6. MAMP inställnings val för PHP version .....	14
Figur 7. Screenshot av den svenskspråkiga webbtjänst version (bubbl.us) .....	16
Figur 8 & 9. Polylang språk inställning och språk innehålls val. ....	22
Figur 10 & 11. Språk översättnings knappar. ....	23
Figur 12 & 13. Screenshot tillsatt engelska tillsatt modul och huvudmenyer.....	24
Figur 14. Kalender till Sidans innehåll .....	25
Figur 15. Kalender inställningar i textredigerare vy i sid innehåll Koulutuskalenteri ....	25
Figur 16. WordPress kalender (lista vy) .....	26
Figur 17. Drupal händelse modul.....	27
Figur 18. Joomla JEvents kalender .....	28

## FÖRORD

Den här avhandlingen har skrivits mot avslutet av studierna i informations- och medieteknik på Arcada. Den behandlar frågan "Hur man väljer rätt CMS". Ämnet har valts för att jag har haft kurser i mina studier som har behandlat två olika exempel på innehållshanteringssystem och jag har arbetat på en firma som använder webbpubliceringssystem för att bygga upp och upprätthålla webbtjänster. Det har väckt ett intresse hos mig om den teoretiska metodiken och planeringsprocessen för att uppnå ett webborienterat mål.

Tack går till min studiehandledare för frågeställningarna som inspirerade mig till detta examensarbete och hans tålamod med att det tog lite mer än ett år innan arbetet var klart att inlämnas. Ett särskilt tack går till Valteri skolan för att den gav mig tillstånd att använda innehållet från deras nuvarande webbplats som modell för att arbeta med och planera runt en mera konkret plan som svarar på min fråga. Också ett tack går till mina vänner som hade tålamod att låta mig bolla mina idéer med dem trots att de inte har något särskilt intresse av mitt studieområde.

# 1 INTRODUKTION OCH BAKGRUND

## 1.1 Allmän introduktion och bakgrund

Denna avhandling tar en titt in på användningen av innehållshanteringssystem (CMS) för att uppbygga en organisations webbplats. Mer specifikt så undersöker den tre populära applikationer med öppen källkod som finns gratis tillgängliga för användarna. Jag har valt att använda tillämpningar av WordPress, Drupal och Joomla! på programmen MAMP utan att publicera webbplatsen, för att kunna testa processen och stegen för uppbyggande av webbsidor. Främst undersöker jag vilka krav som behöver ställas för att besluta vilket av de tre innehållshanteringssystemen som bäst kan uppnå det önskade slutresultat. Avhandlingen söker också efter vilken kunskap som behövs mest för att uppfylla de nödvändiga standarderna för användaren som gör sidorna tillgängligt för målgruppen.

Jag strävar till att klargöra begreppet innehållshanteringssystem som är en term som omfattar ett stort område. I denna avhandling används CMS som förkortning för webb-innehållshanteringssystem. Dessa system utnyttjar centrala gränssnitt som tillåter utvecklaren att publicera, redigera, ändra, organisera, ta bort och upprätthålla innehållet på en webbplats/tjänst (*cmscritic, 2017*).

Det beräknas finnas miljontals webbplatser på nätet varav det estimeras att ca 53 % utvecklas självständigt, medan resten har byggts upp och upprätthålls genom användning av en CMS applikation (*3techs.com, 2017*).

## 1.2 Syfte och mål

Frågan som har gett inspiration till denna avhandling lyder "hur väljer du rätt CMS?" Avsikten är att besvara hur planera och påbörja genomförandet av en fallstudie för uppbyggandet av en webbplats med hjälp av de tre valda CMS - WordPress, Drupal och Joomla. Genom fallstudien planerar jag vad som skulle behövas för att uppfylla kraven för webbplatsen och sedan tittar jag på vilka program som mest effektivt möter dessa



krav. Genom dessa iakttagelser analyserar jag skillnader i uppbyggande av de olika CMS och hur de uppfyller kraven. Jag utgår från följande frågor:

- Vilket system är lättare att använda?
- Vilket system kräver mer teknisk kunskap och stöd för att navigera?
- Vilket system skulle rekommenderas för enskilt bruk: för en programmerare eller en amatör? För större projekt eller mindre?

Det slutgiltiga svaret kommer att vara något partiskt och baserat på min egen förståelse och erfarenhet och kommer att vara mer en (så objektivt som möjligt) jämförande studie av olika metoder för att nå samma mål.

### **1.3 Metodik**

Enligt den metod jag har valt att analyseras de tre CMS i flera steg. Detta för att det skall gå bättre att följa med utvecklingen av webbplatserna och jämföra hur lätt eller svårt det är att få önskat resultat. Ordningsföljden är följande:

- Förberedelse och installation – användargränssnittet (*User Interface*)
- Uppbyggande av innehåll och menyer
- Uppbyggande av innehållssidofält för att förenkla navigation
- Undersökning och anpassning av brödsmulor (navigerings element på sidor som visar sammankopplingen och också kan visa användarens klickordning) och webbadresser (navigations tillägg)
- Undersöka möjligheterna för flerspråkiga webbtjänster
- Tillgänglighetskoder och hänsyn till synnedsättning (testning görs med hjälp av Safari och Mac VoiceOver)
- HTML och CSS kodanpassning

## 2 FÖRBEREDELSE

För att uppnå mitt mål med att utreda olika innehållshanteringssystem (CMS), övervägde jag vilka från en lång lista av nuvarande mjukvara och program som jag skulle använda

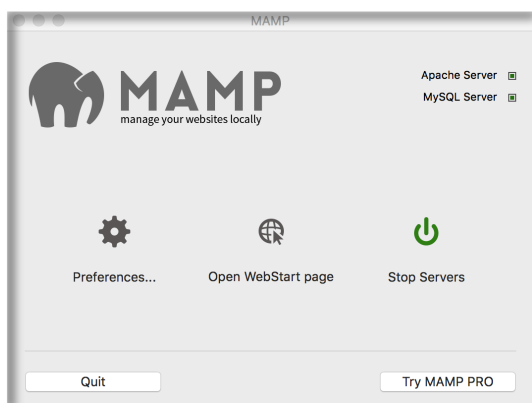
och testa och ställde mig dessa tre frågor:

- Vilka CMS är lätt tillgängliga och aktuella?
- Vilka av dessa är de mest populära?
- Vilka av dessa rekommenderas?

Utifrån dessa frågor valde jag de tre mest populära (*Built with, 2016*) och mest rekommenderade mjukvarorna samt såg till att de var aktuella. Dessutom valde jag dem som hade öppna källkoder (alltså de som är gratis) och givits ut under *GNU General Public License*.

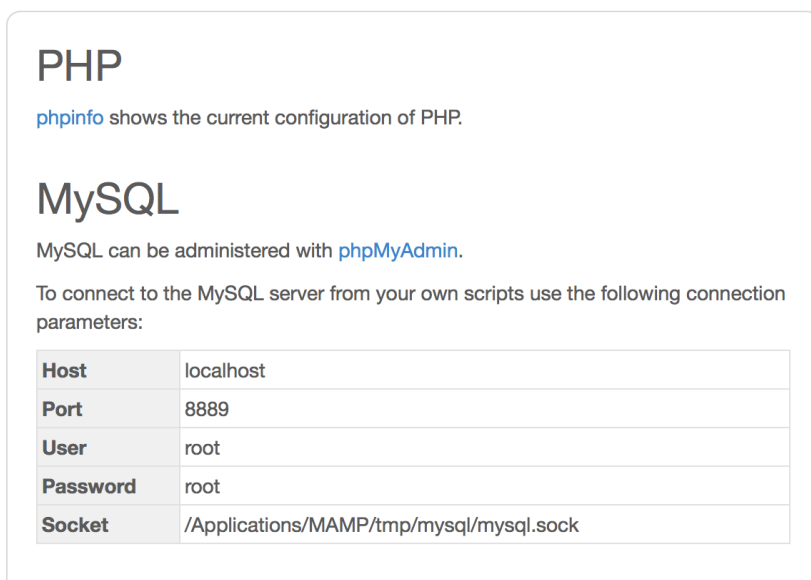
### 2.1 MAMP

Innan jag kunde gå in på de olika CMS som jag hade valt att använda, måste jag först installera applikationen MAMP eftersom CMS behöver en server, för att fungera. Detta är orsaken till att jag laddade ner MAMP som är en öppen källkodad programvara för datorer med Macintosh operativsystem. Detta program installerar en lokal servermiljö som körs internt på datorn utan att publicera någonting på nätet vilket är bra om man vill bygga en ny webbplats och först testa den (Figur 1).



Figur 1. Screenshot av MAMP

MAMP innehåller dessutom program som tillåter dig att manipulera och redigera databaser från *WebStart* sidan (som är tillgänglig från applikationen) varifrån man kan komma åt *phpMyAdmin* (Figur 2). Detta är nödvändigt eftersom det behövs databaser för att kunna installera de tre CMS mjukvarorna på datorn. Dessa är viktiga för att webbplatser behöver någonstans att spara information på sidorna, såsom användare, lösenord, osv.



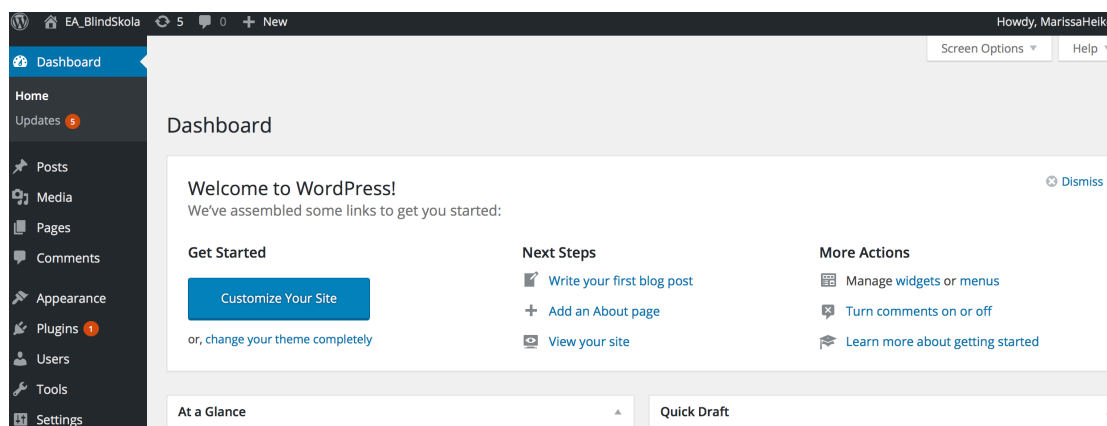
Figur 2. Screenshot från MAMP WebStart sida, vart man kommer åt *phpMyAdmin* kontroll

## 2.2 WordPress 4.7.2

WordPress är en öppen källkod CMS som ursprungligen utgavs den 23 maj 2003 av *Matt Mellenweg* och *Mike Little*. Kärnan i WordPress idag är att den används som en *community* av frivilliga individer för att hålla WordPress i gång samt uppdaterad. Den senaste versionen (vid tiden för mitt arbete) är WordPress version 4.7.2. Det är möjligt att använda WordPress via deras hemsida eller att ladda ner mjukvaran och hantera den själv. Det senare alternativet är det som jag använder. (*Wordpress.org, 2016*)(Figur 3)

Det uppskattas att ungefär 50 % av de nu existerande webbplatserna är uppbyggda med WordPress (*Built with, 2016*). Det är en av huvudorsakerna till mitt val att använda WordPress. Andra orsaker är att den är skriven i PHP och använder *MySQL* som stöds av alla CMS som jag redan har bekantat mig med. Exempel på tre webbplatser som an-

vänder WordPress är Mercedes-Benz hemsida, sångaren Katy Perry, och Flow Festival. (*Elegant Themes, 2016*)

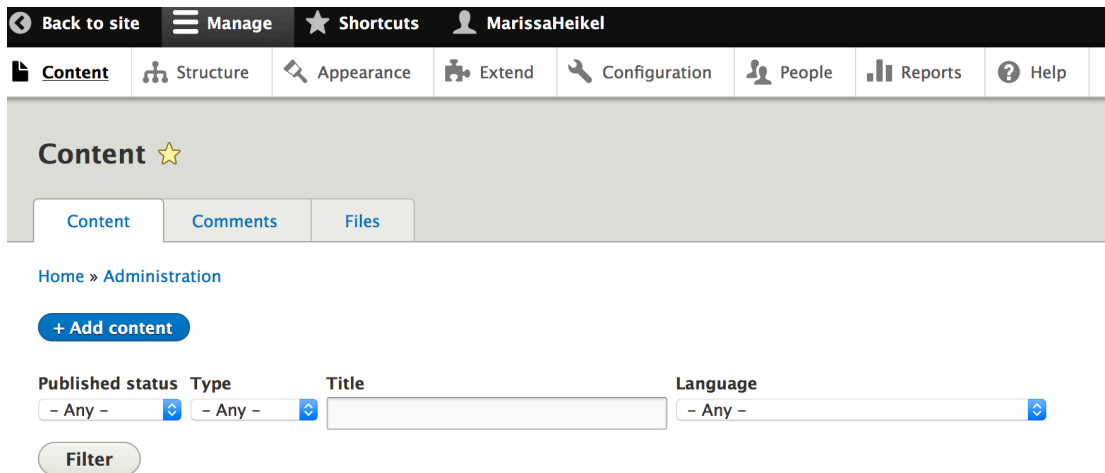


Figur 3. Screenshot av WordPress admin kontrollpanel

## 2.3 Drupal 8.0.4

Den andra öppna CMS källkoden som jag har valt att använda är Drupal version 8. Vilken ursprungligen utgavs på den 18 maj 2000 av *Dries Buytaert*. Som WordPress är Drupal en mjukvara skriven i PHP tillgänglig på webben. Den utnyttjar också en omfattande *community* av individer som håller den uppdaterat. Drupal anses oftast vara ett svårare program att hantera i jämförelse med andra CMS. (*Drupal.org, 2016*)(Figur 4)

Det är estimerat att cirka 2,7 % av de kända webbplatserna är skapade med hjälp av Drupal (*Built with, 2016*). Enligt Drupal hemsida finns det över 1 000 000 webbplatser som använder deras mjukvara, som exempel kan nämnas Drupal hemsida, eBay inc., och Oxford Universitet. (*Drupal Showcases, 2016*)



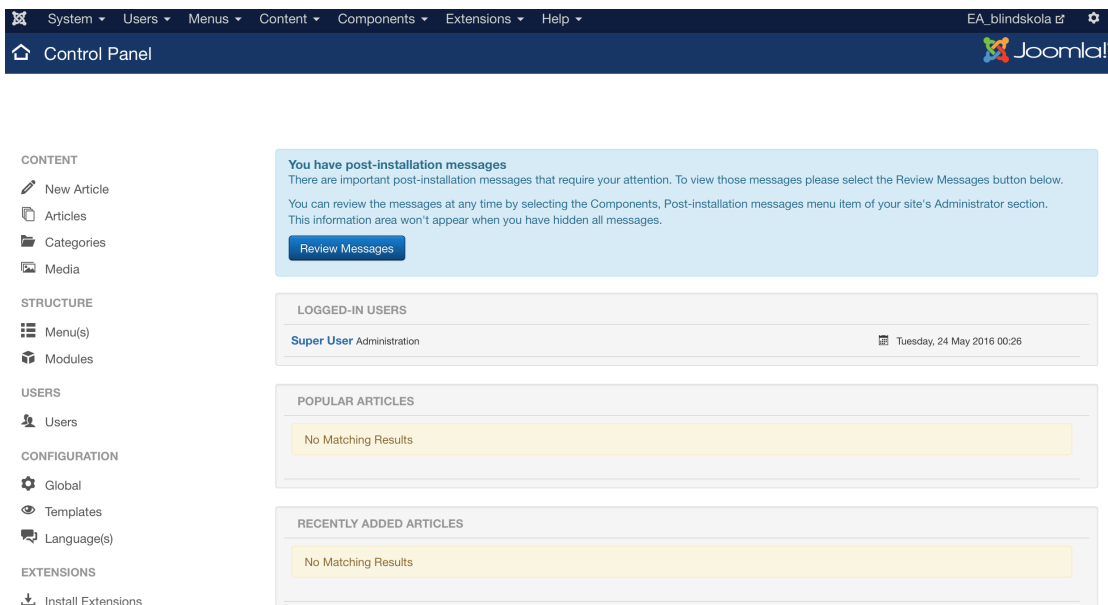
Figur 4. Screenshot av Drupal kontrollpanel

## 2.4 Joomla! 3.6.5

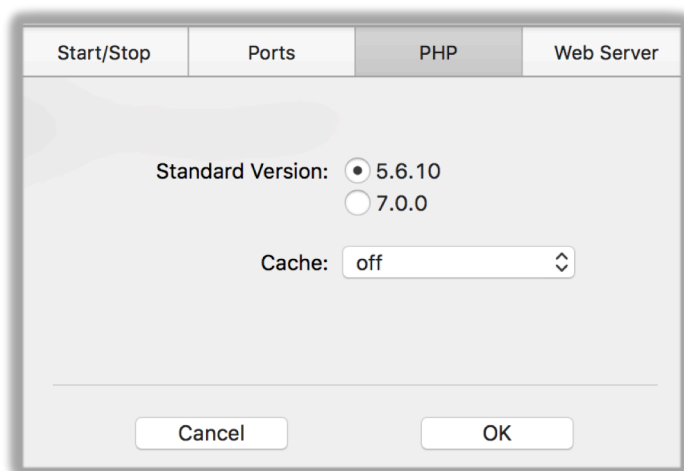
Den sista av de tre CMS som jag har valt att testa är Joomla! version 3.6.5 som utgavs ursprungligen på den 17 augusti 2006 och har utvecklats av projektgruppen Joomla. Det är en mjukvara som likt WordPress och Drupal är skriven med PHP och kan också använda MySQL för att spara data. (Joomla.org, 2016)(Figur 5)

Då jag först installerade Joomla! verkade det som om den inte fungerade med MAMP's PHP version 7.0, så jag måste gå in på MAMP's preferenser och ändra till PHP version 5.5.10 (Figur 6).

Det estimeras att cirka 2,1 % av webbplatser använder Joomla! (built with, 2016). Tre exempel på webbplatser som har använt Joomla! är Linux hemsidan, Harvard Universitet, och Joomla!'s egen hemsida. (CometChat, 2015)



Figur 5. Screenshot av Joomla! kontrollpanel



Figur 6. MAMP inställnings val för PHP version

### 3 FALLSTUDIE – VALTERI SKOLA

Jag valde att använda mig av en fallstudie för att kunna skapa en mer definierad plan med mål och krav som måste uppnås för att så objektivt som möjligt välja ett CMS. Det är en ganska subjektiv process att undersöka rätt val av CMS och det beror på flera faktorer som bl.a. omfattar de tillgängliga resurserna (till exempel: hur mycket pengar man har till förfogande och vilken man möjligtvis då har att betala för en tjänst (plug-in)), ens kunskaper och egna erfarenheter att utveckla en webbplats osv.

### **3.1 Bakgrund och information**

Denna fallstudie är en som imiterar en organisation per definition, eftersom det är en organisation med ett visst mål. Ursprungligen var fallstudien tänkt att byggas runt Skilla, tidigare Svenska skolan för synskadade, men Skilla har slagits ihop med Valteri skolan, som bl.a. är ett resurscenter som riktar sina tjänster till lärare i Finland som vill få stöd och fortbildning.

### **3.2 Plan – mål och krav**

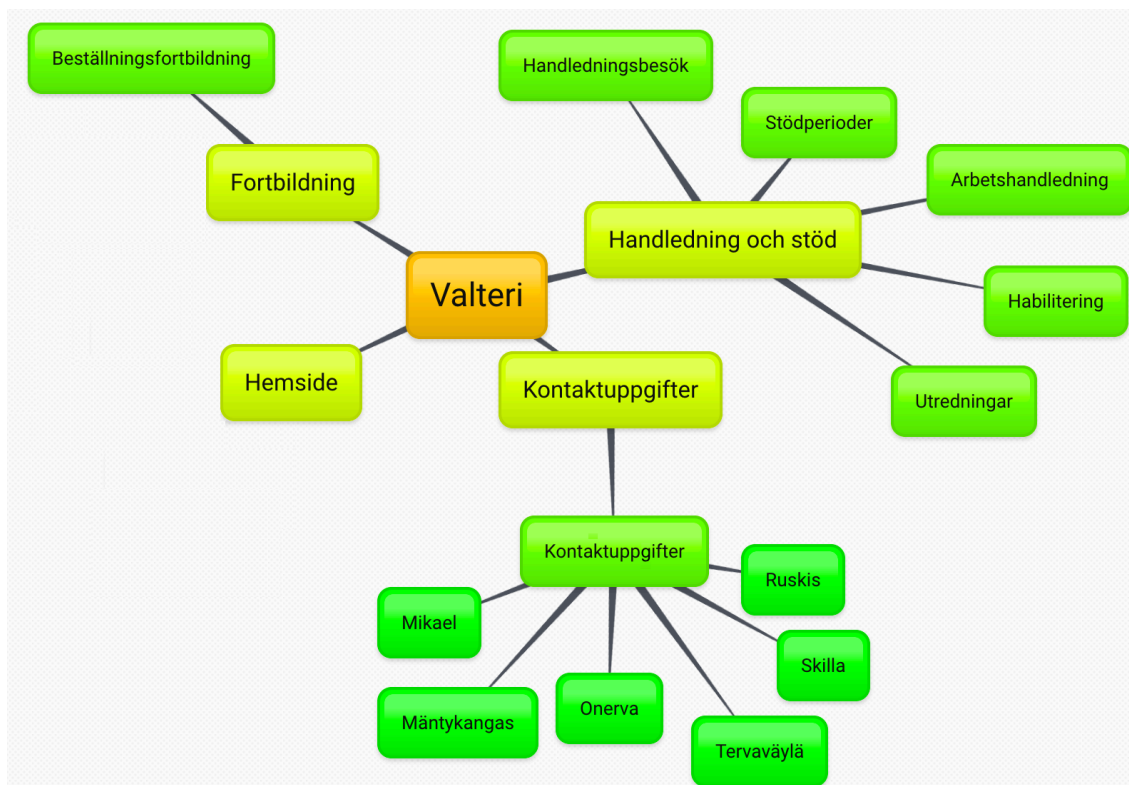
Målet med webbplatsen i det här fallet är att den skall ge information och nödvändiga uppgifter till lärare om tjänster och fortbildningstillfällen samt fungerar som kontaktpunkt för enheterna inom Valteri. Av detta kan man dra slutsatsen att den primära målgruppen kommer att vara lärare och att de viktigaste formerna av information som behöver vara tillgänglig för dem skall finnas på hemsidan, som fungerar som kontaktsida med basinformation om vad Valteri skolan har att erbjuda. Med andra ord det viktigaste målet för webbplatsen är spridningen av information.

Dessa mål betyder att webbplatsen inte behöver vara särskilt avancerad men måste ändå beakta att den behöver vara rimligt tillgänglig för potentiella användare med nedsatt syn, detta för att en av skolorna som är en del av Valteri är Skilla – tidigare Svenska skolan för synskadade. Webbplatsen måste därför ta i beaktande hur det är möjligt att påverka färgschema och färgkontraster av det valda temat så att det kan undvikas att navigering av tjänsten blir svår för användare med nedsattsyn. Tillgängligheten på CMS testas på webbläsaren Safari med Mac inbyggt VoiceOver applikation.

#### **3.2.1 Strukturkraven**

Den planerade strukturen för fallstudien kommer att vara fyra huvudsidor med tolv undersidor på svenska, fyra huvudsidor och elva undersidor på engelska, och fem huvudsidor plus en huvudnavigeringslänk till voppi.fi med sjutton undersidor för den finska versionen.

Figur 7 visar den planerade strukturen för den svenskspråkiga versionen av tjänsten. Den engelskspråkiga versionen kommer inte att ha undersidor eller sidofält till utbildningssida medan den finska sidan kommer att ha ett objekt till i menyn med tre undersidor. Detta betyder att ett av de krav som ställs på det testade CMS är förmågan att skapa alternativa sidofältsmenyer som bara skulle vara synliga på vissa sidor. Dessutom borde det finnas en logisk struktur till brödsmulorna (navigation) som följer användare navigerings ordning: Hem > huvudnavigeringssida > undersida.



Figur 7. Screenshot av den svenskspråkiga webbtjänst version (bubbl.us, 2017)

Eftersom en språkväxlare krävs, måste jag också undersöka hur det skulle vara möjligt att skapa innehåll som är sammankopplade på tre olika språk så att när en användare tittar på en sida, och byter språk omdirigeras hemsidan automatiskt till den andra språkversionen när en sådan finns. En av de enklaste metoderna skulle ha varit att för varje språk skapa en egen webbplats och sedan länka till varandra. Men jag var mera intresserad av att undersöka en potentiell process för att skapa en enda flerspråkig webbplats, så att tjänsten skulle ha förmågan att känna igen och organisera innehållet på de tre språken.



## 4 ANALYS OCH RESULTAT

Det här kapitlet beskriver resultaten för skapandet av webbplatserna och om hur webbplatserna uppnådde förväntningar och planerade avsnitt för specifika CMS.

### 4.1 Sidorna – skapa statistiskt innehåll

Det första jag undersökte var hur de statiska sidornas innehåll kan skapas och hur de kan genast vara tillgängliga för manipulation, som webbadress och deras automation.

#### 4.1.1 WordPress

WordPress kontrollpanel har två flikar som kan skapa innehåll: *Pages* och *Posts*. Frågan för min fallstudie är vilken av de två typer av innehåll behöver jag eller när ska jag använda det ena eller det andra?

Min definition på WordPress *Page* är en statisk sida som endast ändras när utvecklaren ändrar eller redigerar den. En WordPress *Post* är avsedd att användas för blogginslägg: Poster visas ordnade på basis av publiceringsdatum vilket för min planerade webbsida för Valteri inte behövs. Så för min fallstudie behöver jag bara skapa sidor med *Pages*.

WordPress *Page* skapande är ganska automatiserat med ett titelfält, ett kroppsfält och ett attribut-område där sidan kan ställas in så att den har en föräldrasida/modersida (*parent*) eller inte, vilket också automatiserar webbadressen. Till exempel, när utvecklaren skapar huvudnavigationssida Kontaktuppgifter, fylls webbadressen automatiskt in med '/kontaktuppgifter', men när en barnsida, såsom Ruskis blir inställd, läggs webbadressen automatiskt till föräldrasidan, då fylls den in med '/kontaktuppgifter/ruskis'.

#### 4.1.2 Drupal

Drupal har färdigt installerat två innehållstyper *Articles* och *Basic Pages*. Dessa följer vid användning samma resonemang som WordPress *Pages* och *Post*. Så användning av *basic pages* för att skapa huvudsidornas innehåll var okomplicerat.

Standard *Basic Page* inställningar innehåller titelfält, kroppsfält, meny-länk-inställningar (var i webbplatsens struktur sidan tillhör: modersidor), och manuell webbadressinställning. Den manuella webbadressinstallation tillåter användaren att ha full kontroll över innehållet. Men nackdelen är att om användaren vill skapa en webbadress baserad på tilldelade modersidor så måste stavningen vara korrekt före publicering ifall användaren bestämmer sig för att navigera med webbadressen i stället för menyer eller brödsmulor (navigation).

### **4.1.3 Joomla!**

Joomla har *Articles* och *Featured Articles*, som det är lite svårt att förstå skillnaderna mellan. En synvinkel på *Featured Articles* är att det är något som skulle kunna skapa en mer statisk sida, men Joomlas inre struktur fungerar så att *Articles* fungerar lika bra, så att mappningen av sajtens navigationsstruktur är mer invecklad eftersom för att kunna mappa navigationsmenyn (även en standard version) måste länkarna skapas via fliken Menyer där du väljer *Menu Item Type > Articles > Single Article* och sedan väljs den artikels innehåll som ska läggas till menyn.

Webbadressens redigering är här svårare eftersom då man skapar sidorna så finns det inte något mer än alias för artikeln. Det saknas en automatisk sammankoppling mellan modersidorna och deras barnsidor. Barnsidorna ärver inte automatiskt modersidans alias vid skapandet av nya webbadressen eller då nya menyer skapas (meny skapning diskuteras i nästa avsnitt).

## **4.2 Menyer och sidofält**

En viktig sak för en webbplats är skapandet av navigering mellan innehållet. Så hur varje CMS skapar och manipulerar innehållsnavigering, hur det kan utökas och vad som händer med brödsmulorna (navigation) behöver undersökas. De planerade kraven av webbplatsen behöver inte bara en huvudnavigeringsmeny, utan också en alternativ metod för att visa innehållsspecifika sidofält.

### 4.2.1 WordPress

WordPress inställning av huvudnavigering är det enkelt att direkt påverka eftersom utvecklaren skapar innehållet. Det finns ett attributfält för valet om en sida har toppnivå innehåll eller om den ärver överordnade innehåll (dvs. moderinnehåll). Då det finns en överordnad sida uppdateras innehåll på webbadressen automatiskt så att t.ex. Stödperioder alias /stodperiod ärverHandledning och Stöd alias /handledning-och-stod så att det blir /handledning-och-stod/stodperioder.

För de engelsk- och svenskspråkiga webbplatserna skapas en huvudmeny med 4 menyalternativ och 6 för den finskspråkiga webbplatsen som kan mappas utan förändringar. Det är också möjligt att skapa nya menyer på kontrollpanel-fliken och tilldela dem specifika språkversioner efter att flerspråkig plug-in och moduler har inställts. Det unika språkmenyn lägger till menynamnen med (*Primary language*). Under samma flik kan utvecklaren också skapa menyer som bara ska synas på vissa sidor, men det finns inte färdigt inställda metoder för att mappa vissa menyer på vissa sidor, det behövs en plug-in.

För att kunna mappa vissa menyer på specifika sidor valde jag plug-in *Custom Sidebars (WPMU DEV)* för att redigera och använda de skapade unika menyerna för sidofält i kontroll-panel-flik *Utseende - Widgets*. Det är då möjligt att leda de nya sidofält menyerna till rätt sida när man öppnar innehållet till avsnittet *Sidebar* (inställt sidofält mappas till *right sidebar*).

### 4.2.2 Drupal

Metoden för Drupal för att skapa huvudnavigeringsmeny liknar metoden för WordPress eftersom det kan påverkas direkt och skapas i innehållet i avsnittet "Inställningar för meny". Utvecklaren kan också navigera i kontrollpanelen till Struktur > Menyer och skapa nya menyer och lägga till innehåll direkt. Då den flerspråkiga installationen har slutförts kan utvecklaren också redigera och översätta menyn. Tilldelning av sidofält i Drupal behöver inte någon extra inställning eftersom i Struktur > Blocklayout delen av Drupal har allt man behöver. T.ex. finns det menyer som tillhör alla sidor associerat med Handledning och Stöd som är konfigurerade till *Sidebar First* och de skapade

finskspråkiga huvudmenyerna blev konfigurerad till samma ställe som standardmenyn i Sidhuvud.

### 4.2.3 Joomla!

Menyer i Joomla är inte lika automatiserade som på de andra två CMS eftersom de måste skapas manuellt från början. Joomla har redan från början en standardhuvudmeny på fliken Menyer men på grund av den flerspråkiga installationsprocessen av Joomla; måste en separat meny för varje språk skapas. När menyerna har skapats med sina tilldelade menyalternativ, måste menyerna länkas till en ny modul (antingen från fliken menyer eller från *Extensions > Manage Modules* fliken). Det är i själva modulen som utvecklaren lägger till vilka sidor i menyn som syns. Det betyder att varje gång som ett nytt innehåll skapas, måste den modulen läggas till sidans menyobjekt och sedan göras synlig med *Menu Assignment* inställningarna.

Sidofält kan också skapas som en del av menyprocessen, men problemet med Joomla är att sidofältet inte ärver rätt överordnad alias och brödsmulor från de sammankopplade föräldrasidorna. I Joomla kan en sidomeny inte automatiskt mappa överordnat innehåll till webbadressen eller länkstigen så att navigeringen från hemsidan > kontaktuppgifter > Onerva blir webbadressen /sv/Onerva och brödsmula hem > Onerva istället. Det skulle vara bättre om webbadressen var /sv/kontaktuppgifter/Onerva och brödsmulor hem > kontaktuppgifter > Onerva. Lyckligtvis installeras huvudmenyerna så att menymodulen inte visar undermenyalternativ då det inte ska med utan tillstånd till inställningarna *Show Sub-Menu Items*. Detta gör att när du tittar på webbplatsen från en sida utan barnelement, visar menyn endast huvudnavigering, men om sidan är på till exempel Fortbildning, kommer de ärvda ordnade sidorna att visas som en del av navigationen (åtminstone med standardtemat, det skulle behöva undersökas mera med andra tema för att se om det fungerar alltid eller om en alternativ metod behövs).

## 4.3 Flerspråkig inställning

Den andra installationen för att nå fallstudiens krav är att hitta en metod för att skapa flerspråkigt innehåll. Den enklaste metoden skulle vara att skapa separata tjänster för

varje språk och sedan länka varje språkversion till varandra (eller att skapa innehållet på alla språk och sedan skapa menyer skilt för varje). Men i denna avhandling valde jag att undersöka installationsprocessen för plugin och dess moduler för att kunna bättre organisera innehållet på en webbplats beroende på innehållets språk.

### 4.3.1 WordPress

Flerspråkig inställning på WordPress kräver användningen av fliken plug-in, men om utvecklaren söker under termerna *multi-lingual*, *multi-site* o.s.v., finns det en stor mängd alternativ. Så jag undersökte rekommendationer från olika forum och gjorde video-självstudier och beslöt att installera plugin Polylang (*Frédéric Demarle*). Jag beslöt detta för allt innehållsmaterial redan fanns och det inte behövdes någon utvecklad översättning eller ett överföringssystem. Polylang möjliggör att innehållet som skall skapas på ett annat språk automatiskt kan sammankoppla innehållet med de eventuella alternativa språkens innehåll. Första steget är att inställa språkalternativ på fliken nya språk som läggs till vänster sidan på WordPress kontrollpanel. (Figur 8)

Efter installationen kan innehållet skapas på det nya språket genom att navigera till fliken sidor och klicka på + -symbolen (Figur 9) under kolumnen med rätt språk flagga. Innehållet läggs sedan automatiskt till de relevanta sidorna och menyerna ifall de har byggts på rätt sätt så de registrerar deras associering (vilket innebär att innehållet länkar till samma sida vid byte av sajtspråk). Språkväljaren för WordPress kan tilldelas på meny fliken eller i *Custom Sidebar* widget som har skapats. Helst skall det vara synligt någonstans som är vettigt. För fallstudien är det valda stället för in. Till exempel har den finska sidan *kehittämishankkeet* ingen översättning, så ingen språkväljare syns.

## Languages

### Add new language

Choose a language

Svenska - sv\_SE

*You can choose a language in the list or directly edit it below.*

Full name

Svenska

*The name is how it is displayed on your site (for example: English).*

Locale

sv\_SE

*WordPress Locale for the language (for example: en\_US). You will need to install the .mo file for this language.*

Language code

sv

*Language code - preferably 2-letters ISO 639-1 (for example: en)*

Text direction

left to right

right to left

*Choose the text direction for the language*

Flag

Sweden

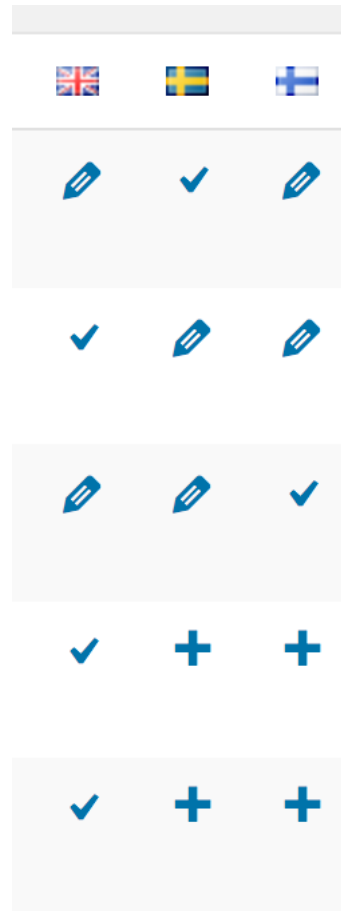
*Choose a flag for the language.*

Order

0

*Position of the language in the language switcher*

**Add new language**



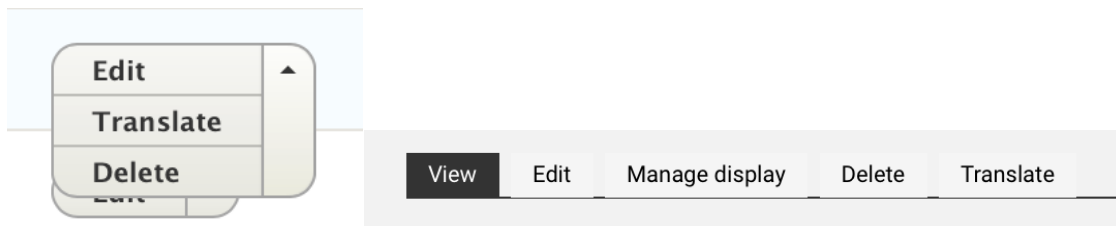
Figur 8 & 9. Polylang språk inställning och språk innehålls val.

### 4.3.2 Drupal

De flesta källor på nätet om flerspråkiga installationer för Drupal är baserat på äldre versioner av Drupal där en plugin behövde installeras. Drupal 8 har nu modulen automatiskt tillgänglig, och måste ställas in av utvecklaren. Installationsprocessen för Drupal flerspråkiga funktioner är inte svårare än för WordPress.

Utvecklaren måste navigera till Utöka fliken och bläddra till det flerspråkiga avsnittet Flerspråkighet och välja alla alternativ för att installera. Efter det är nästa steg att navigera till fliken för konfiguration där alternativen för språk skall läggas till så utvecklaren kan installera ett nytt sajtspråk. Nya språk läggs automatiskt till språkväljaren vid installationen.

Språkväljaren måste sedan läggas till i blocklayout avsnittet i Drupal enligt utvecklarens val. För att kunna lägga till innehållet på ett nytt språk använder man knappen som syns var innehållet syns, och mappar automatisk sammankoppling mellan innehållets språkversioner (likasom med WordPress). (Figurer 10 & 11)



Figur 10 & 11. Språk översättnings knappar.

### 4.3.3 Joomla!

Flerspråkig installation med Joomla kräver att utvecklaren navigerar till kontrollpanelen för inställningar > språk och därifrån väljer och installerar vilka språk som används på webbplatsen. När språken har installerats måste innehållet delas beroende på innehållets språk. Den rekommenderade metoden för detta är att skapa olika kategorier för varje språk och tilldela innehållet till rätt språkkategori och inställa innehållets språk. Detta gör att systemet kan bättre separera innehåll efter språk för manipulation, särskilt när det gäller att skapa menyerna. Vilket är orsaken till att hela processen är den svåraste av de tre CMS. (Figurer 12 & 13)

Det tog lite tid och ansträngning för att hitta rätt dokumentation för att testa de nyare Joomla versioner som var i bruk eftersom språkmoduler hade uppdaterats och förenklats. Det var också viktigt att ta reda på hur mappa den interna webbplatsens struktur så att när en användare byter språk så dirigeras användaren inte automatiskt tillbaka till hemsidan (till skillnad med WordPress eller Drupal). Joomla kräver att utvecklaren manuellt skapar en sammankoppling i menyalternativet för att känna igen sina motsvarande språkinnehåll och inte dirigerar användaren bort från samma sida. Detta behöver göras bara en gång, men det är en av orsakerna varför den interna språk- och navigeringsstruktur av Joomla är den mest komplicerade och tar mest tid.



Figur 12 & 13. Screenshot tillsatt engelska tillsatt modul och huvudmenyer

## 4.4 Skapa händelsesidor och händelsekalender

En annan aspekt av fallstudien är att det finns ett behov för en sida som dirigerar användare till kommande evenemang. Jag skall undersöka metoden att skapa både den enskilda händelsesidan och en kalendersida med en aktuell kalender eller en listmodul av något slag för att visa de ankommande händelserna. Ett möjligt sätt för att visa händelser är att helt enkelt manuellt skapa informationsinnehållet av händelserna och sedan hitta ett sätt att använda kategorier och taxonomier för att organisera innehållet.

Det går också att skapa innehåll med hjälp av plug-in och moduler. Detta är något som först skapades i den finskspråkiga versionen, eftersom det var det enda språk den ursprungliga Valteri webbplatsens innehåll var skrivet på.

### 4.4.1 WordPress

Wordpress har många möjliga alternativa plug-in som kan användas för att skapa en kalender för att hålla reda på uppkommande evenemang vilket jag tror är viktigt med tanke på fallstudien för det behöver finnas någonstans på webbplatsen för användaren att hitta kommande fortbildningar. Jag bestämde mig för plug-in *Event Calendar WD (WebDorado)*. Jag använde basversionen av plug-in eftersom jag inte vill betala för extra funktioner. Detta begränsar de teman som kan väljas från det nedladdade plug-in men jag behöver det bara för att uppnå till mina tre baskrav (förutom skapande av en kalender):

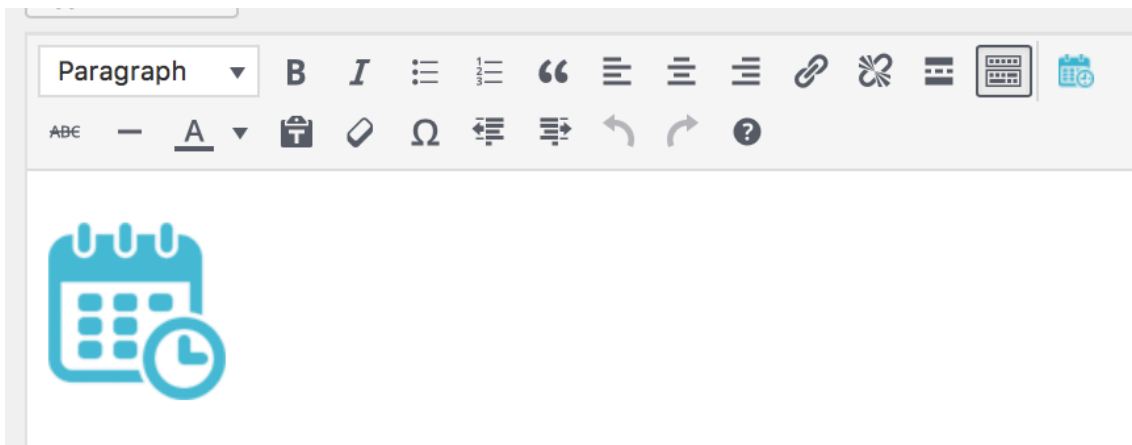
- Ett sätt att skapa fortbildningsanordnaren av händelserna (skolorna som är ansvariga för fortbildningsprogrammet)
- Information om själva evenemanget



- Information om platsen för evenemanget

Först skapas mötesplats, anordnare och kalender för annars blir det nödvändigt för utvecklaren att bläddra tillbaka för att ställa in rätt information på rätt ställen. Därefter skapas en händelse på innehållets högra sida i de befintliga sektionerna var utvecklare lägger till i kalendern vem som är anordnare av händelsen.

För att ställa in en kalender på en sida kan utvecklaren enkelt klicka på kalenderikonen på sidans innehåll (Figur 14) och sedan i textredigeraren ändra koden till hur kalendern skall synas (Figurer 15 & 16).



Figur 14. Kalender till Sidans innehåll

```
[ecwd id="402" type="full" page_items="5" event_search="yes" display="list" displays="list,full,week,day" filters=""]
```

Figur 15. Kalender inställningar i textredigerare vy i sid innehåll Koulutuskalenteri

## Koulutuskalenteri

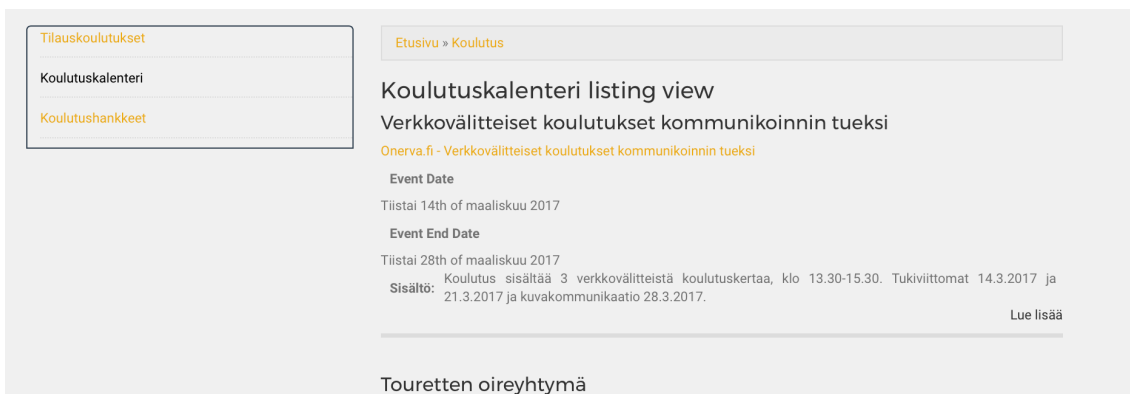
< maaliskuu	< 2017 > Huhtikuu	toukokuu >		
Lista	Kuukausi	Viikko	Päivä	Q
24. Huhtikuu. Maanantai		Toimintakyvyn, toimintarajoitteiden ja terveyden kansainvälinen luokitus (ICF)– koulutus		
🕒 07:00 - 18:00	📅 24-04-2017-27-04-2017			
👤 Oppimis- ja ohjauskeskus Valteri, Tervaväylä				
📍 Ajankohta: 26.04.2017-27.04.2017				
📄 Sijainti: Oppimis- ja ohjauskeskus Valteri, Tervaväylä, Lohipadon yksikkö, Myllytullinkatu 7, Oulu				

Figur 16. WordPress kalender (listvy)

### 4.4.2 Drupal

Skapandet av evenemang i Drupal är mycket svårare. Det tog tid att hitta materialet som behövs för att börja skapa. Orsaken till detta är att innehållstypen för evenemang existerar inte och måste skapas av utvecklaren. Detta betyder att det är möjligt att skapa en händelses innehållstyp som inkluderar exakt den information som avses vara tillgänglig. Innehållstypen skapas i Struktur > Innehållstyper och har ett fält för start datum, komplett beskrivning om all befintlig information om evenemangen, kort (*teaser*) beskrivning, slutdatum (om evenemangen sträcker sig över flera dagar), sponsorer (arrangörerna och ansvariga), ämnen (kategorier) och länk till evenemangens officiella hemsida. (Figur 17)

En annan typ av innehåll som skapades var användargrupper för ansvariga skolor, med all relevanta information inlagd (som skolans kontaktuppgifter samt allmänna information). Jag bytte ut det ursprungliga innehållet som skapats till den nya formen för innehållet eftersom det hade mera information i bättre format med länkar till evenemang som tillhör skolan. Det innehållet var mycket bättre eftersom det skapade bättre sammankopplade innehåll som ledde en användare att möjligtvis söka efter information om skolor som är nära dem och sedan bestämmer sig för att se vilka händelser de kan vara intresserad av att delta i.



Figur 17. Drupal teaser händelsemodul

Vyn på kalendern var avsedd att skapas i Struktur > Visningslägen men på grund av att jag gjorde några dåliga programmeringar, kunde jag aldrig få det att fungera. Jag beslöt mig för att i stället skapa en vy med en lista på händelserna som syns på kalendersidan. Detta hände när jag försökte följa videohandledningar för att skapa en kalender och jag gjorde det i fel ordning så att något fel hände i installationsprocessen som jag inte ännu har hittat lösningen på. Detta ledde till upptäckten av att skapa evenemang och användargrupp innehållstyp som fungerade bra för att skapa bättre anslutna sidor.

#### 4.4.3 Joomla!

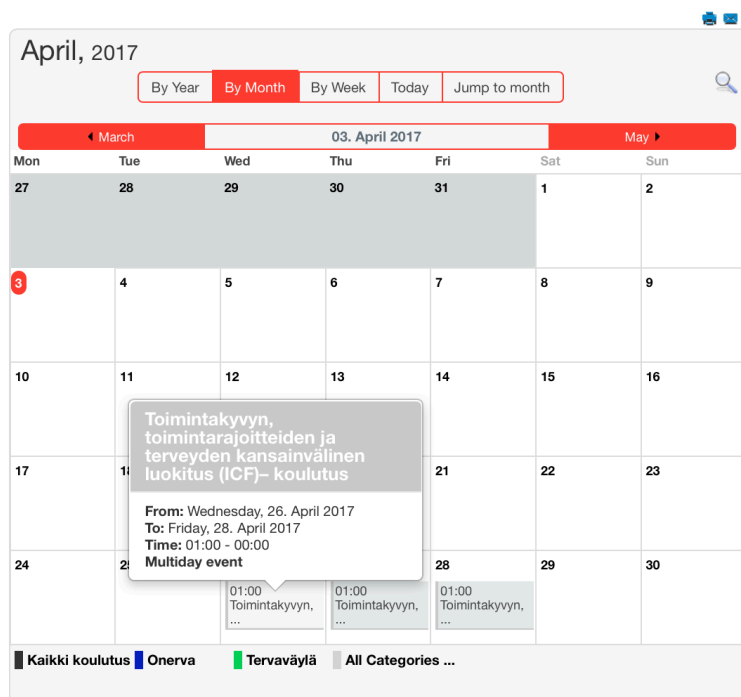
JEvents (*Geraint Edwards*) är plug-in för att skapa evenemang och kalendrar och den installeras från fliken Tillägg> Pluginer genom att söka efter JEvents samt välja de viktigaste komponenterna för plugin; verktyget blir då tillgängligt i Komponenter > JEvents.

Den första delen är en grundläggande konfiguration; det finns tre alternativ att välja från (grundläggande, medel och avancerad). I de här exemplen konfigurerade jag inställningarna med alternativet medel. De mest relevanta inställningarna för kalendern var temat (inställd på *flat*), kalender bakgrundsfärg (ljus – eftersom en mörk bakgrund skulle ha troligen påverkat synligheten av texterna), datumformat (inställt så att veckan börjar på en måndag), och alternativen för att skriva ut eller mejla kalendern.

När evenemangen hade skapats i *events manager*, märkte jag att kategorier kan användas för att automatiskt separera evenemang i kalendern. Så med fem evenemang fanns det totalt 3 kategorier: en för alla evenemang att synas tillsammans, en för ett av Onerva anordnat evenemang och ett för Tervaväylä. (Figur 18)

Processen som används för att lägga till en sida till en meny kan användas för kalendern. Utvecklaren behöver bara i stället välja i menyobjekt typen *JEvents Core > View by Month* och justera inställningarna och godkänna. Resten är detsamma som för att mappa sidor till menyn.

## Events Calendar



Figur 18. Joomla JEvents kalender

## 4.5 Användargränssnittet och erfarenhet

Användargränssnittet hänvisar till interaktionen mellan en användare och ett program eller en hemsida. I det här avsnittet hänvisar jag till användargränssnittet och erfarenheter mellan utvecklaren och CMS och mellan tjänsten och tjänstens målgrupp. I detta sammanhang hänvisas det till webbplatsens initialtillgänglighet för användaren, undersökt med Mac laptop och det inbyggda skärmläsarprogrammet VoiceOver.

### 4.5.1 WordPress

Som nämnts i tidigare avsnitt om erfarenheterna av fallstudien är WordPress gränssnittet enkel att använda och att navigera med. Det behövs inte en massa teknisk kunskap för att börja byggprocessen. De flesta av de nödvändiga komponenterna finns och kan lätt hittas. Den har ingen komplex navigering (åtminstone i jämförelse med de två andra CMS).

När innehållet har installerats och allt som gäller för att kunna tillämpa flerspråkigheten, skapas själva sajten på ett sådant sätt att det är lätt att navigera genom menyer och sidofält. Logiken bakom modulerna tillåter att allt sammankopplas bra utan komplikationer. Den enda punkten som inte fungerar lika bra som jag hade planerat är kontaktuppgiftsidorna. Idealet skulle ha varit att jag skulle ha hittat en metod för att skapa en jämnare navigerbar sida. Fast alla nödvändiga kontaktuppgifter är på sidan med ett sidofält som länkar till individuellt valda skolsidor, och kontaktinformation saknas. Från kontaktsidan är det möjligt att klicka på skolrubriken att hoppa till informationssidan, men det enda sättet att bläddra tillbaka (förutom att använda webbläsarens tillbakaknapp) är genom huvudmenyn eller brödsmlor. Jag skapade inte de enskilda kontaktsidorna med att inkludera kontaktuppgifter.

Teman är också en viktig del av en webbplats, när jag använder termen tema här menar jag webbplatsens visuella design. WordPress har omfattande färdigt skapade teman för att använda gratis, men de flesta jag hittade var för bloggar. Det var inte lätt att hitta ett tema som kunde användas för testning. Lyckligtvis är det möjligt för utvecklaren att navigera till Utseende > Anpassa > Ytterligare CSS för att skapa egen CSS kod. Till exempel på sidan för kontaktuppgifter sattes en kod för att kunna länka med ett div element med egen klass som listar skolorna överst på sidan i en rad som har kodats med Flexbox för att ha lika mycket utrymme mellan varandra.

Efter att skapandet hade slutförts testade jag tillgängligheten för skärmläsare. För det mesta i preliminär testning. Alla sidor och element var läsbara med VoiceOver programmen med undantag för kalendern i månadsvy. Vilket inte är det sämsta resultatet

eftersom list, vecka och dag vyer fungerade helt bra. Det enda felet var att programmen läste ut datum nummer som två-tre istället för tjugotre.

All media i WordPress kan ändras så bilder kan antingen läsas av skärmläsaren med alt-text eller så lämnas det tomt. Med WordPress är det bättre att ge bilderna en alternativ text för att bilder annars läses som "(blank), du är på en bild", vilket ger ingen information. Det skulle vara bättre att antingen göra bilden och dess rubrik icke-läsbar för skärmläsare eftersom informationen inte är tillgänglig för dem eller att ge bilden en lämpligt detaljerad alt-textbeskrivning.

#### **4.5.2 Drupal**

Användargränssnittet för Drupal är svårare att navigera från början jämfört med WordPress. Men det resulterade i en mer grundlig undersökning för hur Drupal skulle användas för att komplettera webbplatsen som igen medförde att jag lärde mig mera utförligt om Drupal som resulterade i att detta program hjälpte mig att bäst nå mina uppställda mål. Dessutom, användning av blocklayout för att styra innehåll gjorde det så mycket lättare att kontrollera placeringen av menyer och sidopaneler. Detta ledde till att mera sidor anslöts utan att försämra flödesstrukturen av webbadresserna och brödsmulorna. Alla komponenter och moduler som jag behövde var redan installerade förutom de flerspråkighetsinställningar som behövdes.

Det enda problemet var ett personligt misstag i byggprocessen för evenemangskalendern (nämnt i ett tidigare avsnitt). På grund av det var jag fast flera dagar innan jag återvände till nybörjarhandledningen för att leta efter lösning på problemet. Detta misstag i byggprocessen hade den bieffekten att jag skapade bättre anslutna och mer informativa kontaktuppgiftssidor som kunde läggas som objekt till innehållet. Detta gjorde det möjligt att lägga till komponenten som tillät evenemang att vara kopplade till sina respektive arrangörer.

Det finns många gratis teman för Drupal, men utvecklaren måste hitta ett tema någonstans ifrån ([drupal.org/project](http://drupal.org/project) till exempel) och sedan antingen ladda ner filerna som innehåller temauppgifter eller installera den till CMS eller använda webbadresslänk för

att ladda ner den. Detta är motsatt till hur WordPress tillåter användaren att surfa direkt och installera teman med en knapptryckning. Dessutom är några personliga tillägg som en utvecklare önskar skapa till CSS koden inte lika enkelt. I stället måste utvecklaren söka fram temats CSS-filer i mapparna och redigera därifrån.

Efter testning fungerade skärmläsaren bra med Drupal, och skärmläsaren kunde läsa allt relevant innehåll (inklusive evenemanglistorna). Tyvärr finns det inget sätt att veta säkert om kalendervy modulen fungerar, men teaser händelse-vy och deras fulla text vy kunde läsas ut.

Det verkar också som om Drupal hade programmerats på ett sätt som läser bild alt-texterna beroende på hur det har inställts av utvecklaren. Detta testades på den engelska språkiga hemsidan med två bilder bredvid de två rubrikerna. Den första bilden gavs alt-texten av "" där skärmläsaren helt hoppade över bilden (även utelämnade beskrivningen av bilden från skärmläsaren), medan den andra bilden gav en alt-text som registrerades av VoiceOver och läste bilden med den kodade alt-texten. Drupal är då mer tillgängligt CMS än WordPress.

### **4.5.3 Joomla!**

Användargränssnittet för Joomla är det mest komplexa av de tre på grund av dess mer komplexa navigationssystem. Det tog längre tid att lära sig de enskilda komponenterna för att kunna uppnå kraven för fallstudien.

Joomla har också den mest komplicerade menyn och sidofältsmappningen (det har också det mest komplexa sambandet mellan meny och brödsmulor). Den initiala uppbyggnaden gjordes med separata menyer som sedan tilldelades specifika sidor och innehåll. Detta fungerade inte eftersom skapandet av separata sidofältmenyer gjorde att överordningen blev fel. Det gjorde att användaren navigerade från (till exempel) handledning och stöd till stödperioder och brödsmulorna läste som hem > stödperioder, den saknade överordnade brödsmula. Det är inte nödvändigt men jag är av den åsikten att brödsmulor ska reflektera användarens klickordning så bra som möjligt.

Joomla har också en omfattande samling av teman tillgängliga för nedladdning och kan hittas från flera källor med sökorderna *Joomla Themes*. Personlig CSS-kod kan läggas till på liknande sätt som med Drupal där utvecklaren antingen kan redigera CSS *stylesheet* som tillhör teman eller skapa ett nytt tema och lägga till det i `index.php` filen för att integrera den. Då jag skapade med Joomla låg mitt fokus på att få det att fungera så jag hade inte tillräckligt med tid för att fullt undersöka temaval och redigeringsprocesserna.

Skärmläsaren på Joomla kunde läsa alla texter och moduler, inklusive kalendern i månadsvy. Skärmläsaren kunde automatiskt hoppa över bilder, men det fanns inget område för att lägga en alt-text, utan i stället skulle det behöva läggas till innehållet direkt. Tillgängligheten för Joomla var mycket bra på en grundläggande nivå.

## 5 SLUTSATSER

Mina slutsatser till frågan om hur man kan välja rätt innehållshanteringssystem (mellan WordPress, Drupal och Joomla!), var inte så lätt att få ett svar på. Att fatta ett beslut på denna fråga är storligen beroende av utvecklaren och dess förståelse för vad det är som de bygger och vilket det önskade slutresultatet är.

När man överväger vilka kraven för fallstudien var, så uppnådde all tre CMS dem. Denna framgång gällande resultatet beror på tolkningen av omfattningen av just hur mycket som hade lyckats. Uppfattningen efter testning är att WordPress var både det enklaste att skapa med och det snabbaste. Medan resultatet med Drupal var att det klarade att nå de flesta kraven ur en teknisk och funktionell synvinkel med det mer fullständiga innehållssammankopplandet. Joomla! verkade då vara det svåraste att manipulera men har den största potentialen, om utvecklaren har tid och tålamod att upptäcka alla tekniska möjligheter.

Om jag skulle ha haft mera tid och mera innehåll att skapa tror jag att det skulle ha varit klokt att göra en närmare undersökning av hur kategorier och taxonomier kan användas för att sammankoppla och manipulera tekniska möjligheter för de skapade webbplatserna med till exempel WordPress och Joomla!. En riktigt väl genomtänkt sammankopp-



ling av skolornas kontaktuppgifter och de olika anordnarnas händelser på samma sätt som det lyckade motsvarande sättet som skapas i Drupal.

När det gäller min fallstudie av Valteri webbsida vars mål var att ge information om fortbildning och lärande för lärare, var Drupal den som bäst uppfyllde de uppsatta målen och kraven. Särskilt i den finskspråkiga versionen av sajten lyckades jag omfatta innehåll som sammankopplade smidigt med händelser och skolornas information som skapades från genom manipulering av Drupal innehållstyper.

## KÄLLOR

CMS Critic, 28 July 2010, Johnston, Mike

Tillgänglig: <https://www.cmscritic.com/what-is-a-cms>

Hämtad: 19.02.2017.

Built With, 2016, CMS

Tillgänglig: <http://trends.builtwith.com/cms>

Hämtad: 18.02.2017

W3techs, Web Technology Surveys, 2003-2017 Q-Success, ”Usage of content management systems for websites”.

Tillgänglig: [https://w3techs.com/technologies/overview/content\\_management/all](https://w3techs.com/technologies/overview/content_management/all)

MAMP, 2004, Hemsida

Tillgänglig: <https://www.mamp.info/en/>

Hämtad: 24.02.2016

WordPress, 2003, Hemsida

Tillgänglig: <https://wordpress.org/>

Hämtad: 26.03.2016

Drupal, 2000, Hemsida

Tillgänglig: <https://www.drupal.org/>

Hämtad: 26.03.2016

Joomla!, 2005, Hemsida

Tillgänglig: <https://www.joomla.org/>

Hämtad: 26.03.2016

Valteri, 2013

Tillgänglig: <https://www.valteri.fi/SV/valteri.html>

Hämtad: 12.01.2017

Bubble.us Mind Map, LKCollab, LLC

Tillgänglig: <https://bubbl.us/042403310904017694>

Hämtad: 26.02.2017

Youtube: Evolving Web, ”Drupal 8 Multilingual Basics”. Augusti 2, 2016

Tillgänglig: <https://www.youtube.com/watch?v=uO1UKIggTwk>

Hämtad: 25.02.2017

Youtube: OStraining, ”The Drupal 8 Beginner Tutorial Class. 19.11.2015. Uppdaterad den 27.02.2017.

Tillgänglig:

[https://www.youtube.com/playlist?list=PLtaXuX0nEZk9MKY\\_CIWcPkGtOEGyLTyCO](https://www.youtube.com/playlist?list=PLtaXuX0nEZk9MKY_CIWcPkGtOEGyLTyCO)

Hämtad: 02.03.2017

Youtube: WPBeginner – Wordpress Tutorials, ”How to Easily Create a Multilingual Wordpress Site”. March 13, 2015

Tillgänglig: <https://www.youtube.com/watch?v=dKnQh0Td9L4>

Hämtad: 02.02.2017

Youtube: Brian, ”Joomla 3.x - How to create a multilingual site in 6 steps”. April 3, 2014

Tillgänglig: [https://www.youtube.com/watch?v=XNI\\_B3RezgQ](https://www.youtube.com/watch?v=XNI_B3RezgQ)

Hämtad: 27.02.2017

Eric Tiggeler, ”Creating a multilingual site in Joomla 3 (Joomla 3 Beginner’s Guide)”. March 11, 2013

Tillgänglig: <https://www.slideshare.net/erictiggeler/creating-a-multilingual-site-in-joomla-joomla-3-beginners-guide-eric-tiggeler>

Hämtad: 27.02.2017

Thebalance, 2017 About, Inc., 22.11.2016, Mindy Lilyquist, ”How to Plan a Website”

Tillgänglig: <https://www.thebalance.com/how-to-plan-a-website-1794805>

Hämtad: 01.02.2017

Kartchner, Chris. "Content Management Systems: Getting from Concept to Reality." *Journal of Electronic Publishing* 3.4 (1998). Article reference from ‘the Journal of Electronic Publishing on the 18.11.2015 (<http://quod.lib.umich.edu/cgi/t/text/idx/j/jep/3336451.0003.408?rgn=main;view=fulltext> )

Dan, Noah, and Albert C. Brown. "System for web content management based on server-side application." U.S. Patent No. 6,560,639. 6 May 2003. Patent referenced 18.11.2015 (<https://www.google.com/patents/US6560639> )

Patel, Savan K., V. R. Rathod, and Satyen Parikh. "Joomla, Drupal and WordPress-a statistical comparison of open source CMS." *Trendz in Information Sciences and Computing (TISC), 2011 3rd International Conference on*. IEEE, 2011.

## **BILAGOR**