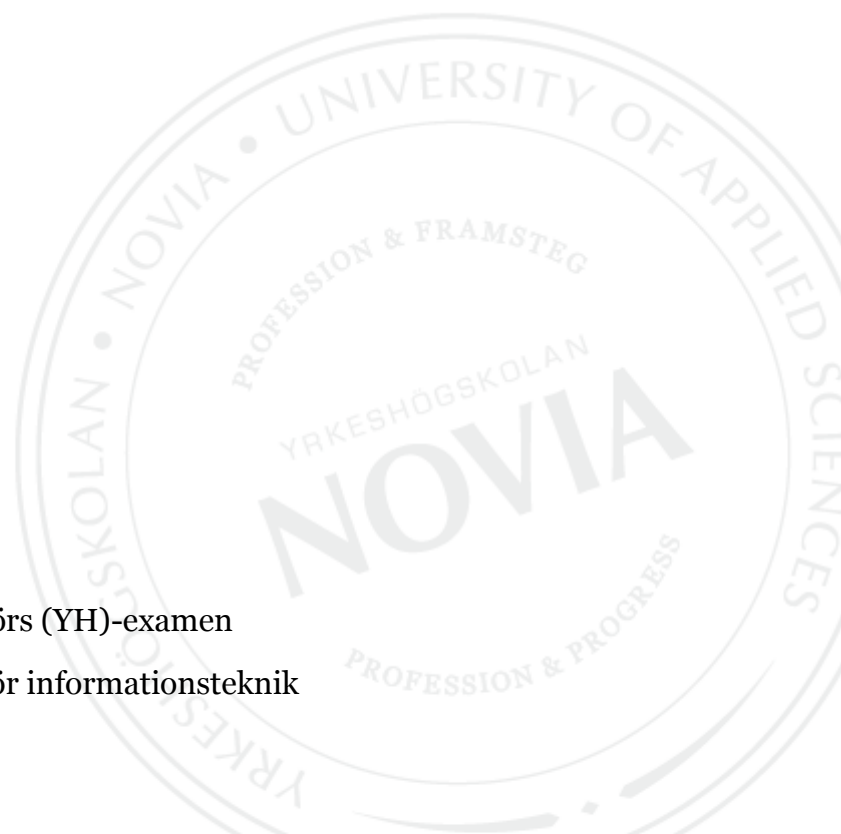


Utvärdering samt val och konfiguration av innehållshanteringssystem

Mats Stolpe

Examensarbete för ingenjör (YH)-examen
Utbildningsprogrammet för informationsteknik
Vasa 2013



EXAMENSARBETE

Författare:

Mats Stolpe

Utbildningsprogram och ort:

Informationsteknik, Vasa

Handledare:

Susanne Österholm

Titel: *Utvärdering samt val och konfiguration av innehållshanteringssystem*

Datum 15.3.2013

Sidantal 33

Abstrakt

Detta examensarbete har utförts åt företaget Anvia i Vasa. Målet med arbetet var att förnya ett innehållshanteringssystem. För att hitta ett lämpligt innehållshanteringssystem gjordes en undersökning av några alternativ. Efter valet av innehållshanteringssystem hade gjorts skulle en webbapplikation skapas för att hantera användare som ville skapa webbplatser.

Resultatet av arbetet är en fungerande installation av innehållshanteringssystemet Wordpress. En webbapplikation har även skapats vid portalen Mitt Anvia för att kunna hantera skapandet av användare samt webbplatser vid Wordpress-installationen.

Språk: svenska

Nyckelord: innehållshanteringssystem, webbapplikation, PHP

Arkiveras: Theseus.fi

BACHELOR´S THESIS

Author:

Mats Stolpe

Degree programme:

Information Technology, Vasa

Supervisor:

Susanne Österholm

Title: *Evaluation, selection and configuration of content management system*

15 March 2013

33 pages

Abstract

This thesis work has been carried out for the company Anvia, located in Vasa. The goal was to renew a content management system. In order to find a suitable content management system a survey has been made. After the selection of a content management system a web application had to be created to handle the users that want to create websites.

The result of this Bachelor´s thesis is an installation of the content management system Wordpress. A web application has also been created at My Anvia, which will take care of the creation of users and websites for the Wordpress installation.

Language: Swedish

Key words: content management system, web application, PHP

Filed at: Theseus.fi

OPINNÄYTETYÖ

Tekijä:

Mats Stolpe

Koulutusohjelma ja paikkakunta:

Tietotekniikka, Vaasa

Ohjaaja:

Susanne Österholm

Nimike: *Sisällönhallintajärjestelmän arviointi, hallinta ja konfigurointi.*

Päivämäärä 15.3.2013

Sivumäärä 33

Tiivistelmä

Tämä opinnäytetyö on tehty yhtiö Anvialle Vaasassa. Tavoite oli uudistaa sisällönhallintajärjestelmä. Jotta löydettäisiin sopiva sisällönhallintajärjestelmä, tehtiin tutkimus. Valinnan jälkeen verkkosovellus rakennettiin, hallitsemaan käyttäjiä jotka haluavat luoda verkkosivustoja.

Opinnäytetyön tulos oli toimiva Wordpress asennus. Verkkosovellus on luotu Mitt Anvia-portaalissa, jotta voidaan luoda käyttäjiä ja verkkosivustoja Wordpress asennuksella.

Kieli: ruotsi

Avainsanat: sisällönhallintajärjestelmä, verkkosovellus, PHP

Arkistoidaan: Theseus.fi

Innehållsförteckning

1	Inledning.....	1
1.1	Uppdragsgivare.....	1
1.2	Bakgrund.....	2
1.3	Uppdrag.....	3
2	Val av innehållshanteringssystem.....	3
2.1	Innehållshanteringssystem.....	3
2.2	Alternativ.....	4
2.2.1	Wordpress.....	5
2.2.2	Joomla.....	7
2.2.3	GetSimple.....	8
2.3	Jämförelse av alternativ.....	10
2.3.1	Administrationspanel.....	10
2.3.2	Uppdateringar.....	10
2.3.3	Teknik.....	11
2.4	Val av innehållshanteringssystem.....	12
3	Tekniker.....	13
3.1	Webb- och skriptspråk.....	13
3.1.1	HTML.....	13
3.1.2	CSS.....	15
3.1.3	PHP.....	17
3.2	Övriga tekniker.....	18
3.2.1	MySQL.....	18
3.2.2	PHP/CURL.....	19
4	Utförande.....	20
4.1	Planering.....	20
4.2	Wordpress.....	20
4.2.1	Installation av Wordpress.....	20
4.2.2	Wpmudev.....	22
4.2.3	Tillägg.....	22
4.3	Webbapplikation.....	23
4.3.1	Mitt Anvia.....	23
4.3.2	Applikationsserver.....	28
5	Resultat och diskussion.....	31
5.1	Vidareutveckling.....	31
5.2	Slutsatser.....	31
6	Källförteckning.....	33

Ordförklaringar

API	Application Programming Interface, kan förklaras som en uppsättning funktionsanrop med vars hjälp man kan kommunicera med en annan programvara.
FTP	File Transfer Protocol, ett filöverföringsprotokoll.
cURL	client URL, ett kommandoradsverktyg som används för att skicka och ta emot information genom att specificera en URL.
CMS	Content Management System, ett system för att visa upp och hantera innehåll, t.ex. visa en bild på en webbsida som skapats med systemet.
CSS	Cascading Style Sheets, ett stilmallsspråk som används för att definiera utseendet på ett sidbeskrivsspråk dokument.
HTML	Hyper Text Markup Language, det huvudsakliga sidbeskrivningsspråket för webbsidor samt annan information som kan visas i en webbläsare.
HTTP	Hyper Text Transfer Protocol, används för att överföra information.
GPL	General Public License, en upphovsrättslicens för fri programvara, användare av programvaran kan modifiera källkoden.
Mitt Anvia	En portal där Anvias kunder kan hantera sina tjänster, t.ex. beställning av tv-kanaler och webbplatsutrymme.
MVC	Model View Controller, ett sätt att dela upp koden i en webbapplikation.
URL	Uniform Resource Locator, en textsträng som används för att identifiera en resurs på webben, t.ex. en webbplats eller ett dokument.
Session	Lagra information för att senare kunna använda sig utav informationen, används i t.ex. inloggningssystem för att se till att användare hålls inloggade.
Tagg	Tagg i detta examensarbete syftar på engelska ordet tag.
WYSIWYG	What You See Is What You Get, används av de flesta CMS för att visa upp hur inlägg och sidor kommer se ut på en webbsida utan att t.ex. HTML-element syns i redigeraren.
XML	Extensible Markup Language, ett sidbeskrivningspråk för dokument.

1 Inledning

Uppdraget gick ut på att ersätta ett föråldrat innehållshanteringssystem genom att undersöka vilka nya innehållshanteringssystem som finns tillgängliga, välja ett och senare även konfigurera det valda systemet. Innehållshanteringssystemet skulle vara enkelt för en användare att förstå och ge goda förutsättningar för att kunna skapa en webbplats snabbt och enkelt. Innehållshanteringssystemet skulle också vara enkelt att underhålla.

1.1 Uppdragsgivare

Anvia är en koncern som består av Anvia Abp (tidigare Vasa Läns Telefon Ab) och dess dotterbolag. Koncernen bytte år 2008 namn från Vasa Läns Telefon Ab som en följd av att man år 2007 hade förnyat sin målsättning. Den förnyade målsättningen var att man skulle sträva efter en starkare ekonomisk tillväxt än förut samt att ha bättre kontakt med sina kunder. Anvia är den fjärde största teleoperatören i Finland med ca 110 000 konsumentkunder och 15 000 företagskunder. Koncernen har tre olika affärsområden: Anvia Securi, Anvia ICT samt Anvia TV.

Anvia Securi är specialist inom trygghetsteknik- och låsteknik. Affärsverksamheten är till största del inriktad på leverans av dörrautomatik och låssystem samt utrustning och tjänster för säkerhetsteknik.

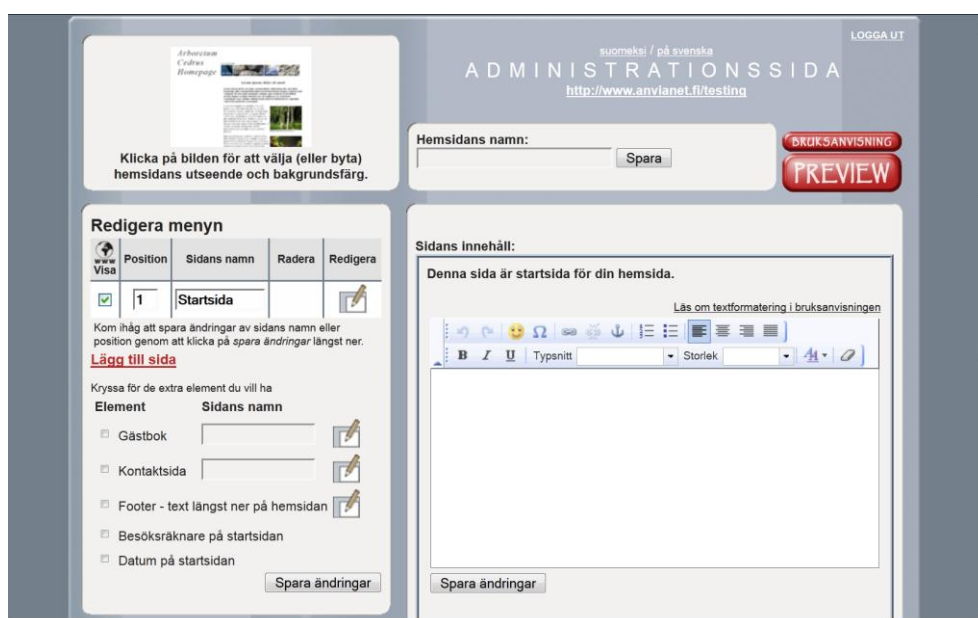
Anvia ICT är den mest betydande leverantören av nätförbindelser i österbottniska landskap, man erbjuder även förbindelser för andra operatörers bruk. Anvia ICT består av tre områden: företagstjänster, konsumenttjänster och nätverk. Till företagstjänsternas utbud hör dataförbindelser, system och utrustning som underhålls, kommunikationslösningar samt support och utbildning som omfattar dessa. De tjänster som är mest betydande för konsumenter är bredband, tv- och samtalstjänster. Samtalstjänster som finns tillgängliga för konsumenter är t.ex. hemtelefonabonnemang och möjlighet att anordna möten via telefon.

Anvia TV erbjuder utrustning och tjänster för mottagning och överföring av rörlig bild och ljud till installationsföretag, operatörer och tv- och radiobolag. Anvia TV utgörs av Anvia TV Oy och Hibox Systems Ab. År 2010 erhöll man nätkoncessioner som en följd av investering i byggandet av antenn-tv-nät för HD-sändningar. Anvia TV är den ledande leverantören av centralantennutrustning i Finland.

Anvia har ca 700 anställda och dess omsättning år 2010 var 108 miljoner euro. Anvias prioriteringar är att ha en bra kontakt med sina kunder samt att ständigt utvecklas. Visionen är att vara kundens bästa samarbetspartner inom kommunikationsteknologi. (Anvia koncernen, u.å.).

1.2 Bakgrund

Anvia erbjuder i nuläget fem stycken webbplatsutrymmen åt sina bredbandskunder. För personer som velat skapa en webbplats, men inte har någon kunskap hur man bygger upp en sida, har man kunnat aktivera en tjänst vid Mitt Anvia som underlättar detta för kunden. Med tjänsten aktiverad kan man logga in på en administrationssida där man kan ändra utseendet på sin webbplats.



Figur 1. Översikt av administrationsverktyget för den nuvarande tjänsten.

Sedan denna webbplats-tjänst togs i bruk år 2006 har tjänsten inte genomgått några stora uppdateringar, vilket lett till att tjänsten börjar bli föråldrad eftersom webbutvecklingen ständigt går framåt och kunder kräver mera möjligheter att hantera sina webbplatser.

För att i fortsättningen ha möjlighet att kunna leverera en tjänst som erbjuder det senaste inom webbutveckling, vill Anvia gå över till att använda ett innehållshanteringssystem med öppen källkod som kunder kan använda för hantering av sina webbplatser.

1.3 Uppdrag

Uppdraget gick ut på att undersöka olika alternativa innehållshanteringssystem med öppen källkod och jämföra dem med varandra. Viktigt i jämförelsen var att hitta en bra balans mellan användarvänlighet, snabbhet och möjlighet för användaren att kunna skapa unika webbplatser. Sedan skulle det valda systemet installeras och konfigureras för att slutligen kunna ge kunder hos Anvia möjlighet att använda det.

Som krav ställdes att systemet kontinuerligt uppdateras av dess utvecklare och att själva uppdateringprocessen enkelt kan genomföras för samtliga sidor då en ny version är tillgänglig. Ett annat krav som ställdes var att systemet ska se bättre ut visuellt och kunden ska ha mera möjligheter att påverka innehållet på sin webbplats än vad man har haft med det gamla systemet. Då systemet konfigurerats skulle även en webbapplikation byggas upp vid Mitt Anvia som kunder skulle ha möjlighet att använda för att lägga till nya webbplatser.

2 Val av innehållshanteringssystem

2.1 Innehållshanteringssystem

CMS står för Content Management System, innehållshanteringssystem på svenska. Då man skapar webbplatser är oftast CMS ett system som hjälper en användare att skapa nytt innehåll på sin webbplats, t.ex. publicering av artiklar eller bilder. Uppdelningen mellan programmering, design och innehåll är det viktigaste målet för ett CMS. Användaren som skapar innehållet på webbplatsen ska inte behöva veta hur personen som skapat designen gjort, samma sak gäller för designern som inte ska behöva veta hur programmeraren gjort. (Hauschildt, 2010, s. 8).

Ett CMS är oftast uppbyggt så att skaparen av material på webbplatsen inte ska behöva ha någon kunskap överhuvudtaget inom programmering för att skapa eller redigera innehållet. Med hjälp av CMS:ets administrationsverktyg kan en inloggad användare skapa nytt innehåll på sin webbplats. Innehållshanteringssystem består av många olika filer som sköter om systemets funktionalitet och i vissa CMS används en databas för att sköta lagringen av innehåll som skapas. (Hauschildt, 2010, s. 8–9).

Fördelarna med ett innehållshanteringssystem gentemot att själv bygga upp en webbplats från grunden är att användare snabbt kan publicera, redigera eller radera innehåll på en

webbplats utan att själva behöva ändra i koden. Många innehållshanteringssystem är även uppbyggda så att de tillåter webbplatsens administratör att lägga till andra användare till sin webbplats med rättigheter att t.ex. skapa nytt innehåll.

2.2 Alternativ

De flesta innehållshanteringssystem är skrivna i PHP, men CMS skrivna i t.ex .NET-språk finns också att välja mellan. Det man bör göra först vid valet av innehållshanteringssystem är att definiera de kriterier som inverkar på valet av CMS. Detta är kriterier som Anvia ansåg hade betydelse:

- Enkelt att påverka innehållet på en webbplats.
- Utbud för att kunna lägga till funktionalitet.
- Stöd för olika språk, speciellt finska och svenska.
- Uppdateringsmöjligheter.

För att snabbt och enkelt kunna testa och jämföra hur administrationspanelen är uppbyggd på olika CMS kan man använda sig av webbplatsen Opensourcecms. Opensourcecms har tillgång till inloggningsuppgifter till över 280 olika CMS, vissa CMS är installerade på Opensourcecms egna webbserver, medan andra finns på innehållshanteringssystemets egen webbplats. Med hjälp av inloggningsuppgifterna kan man sedan logga in som administratör och bilda sig en första uppfattning om hur CMS:et är uppbyggt. (Opensourcecms, u.å.).

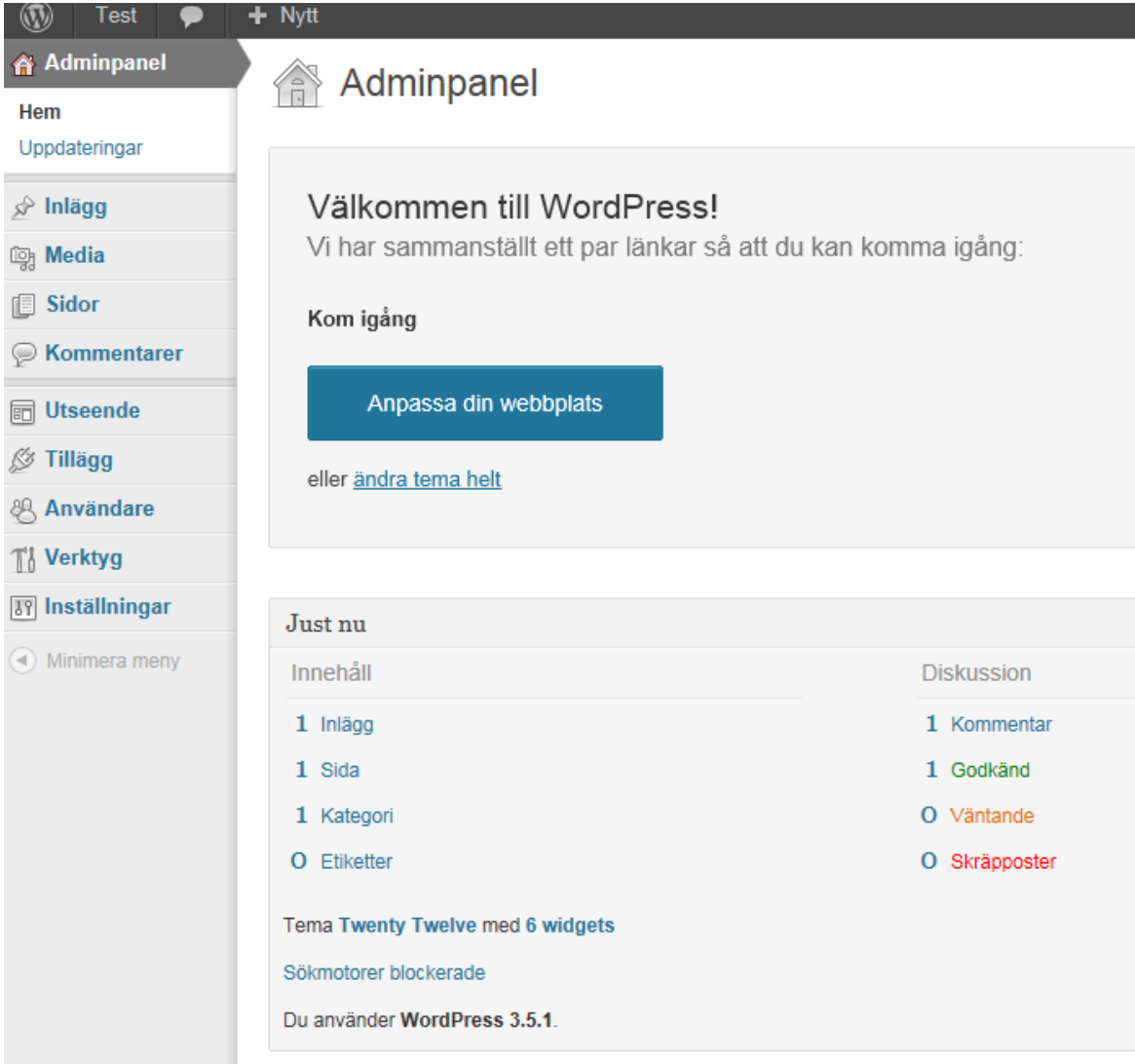
Eftersom det finns flera 100 olika alternativa CMS blev utgångspunkten att undersöka några av de mest populära innehållshanteringssystemen med öppen källkod. Därefter skulle ett av dessa väljas. För att inte enbart ta med CMS-alternativ som använder sig av lagring av innehåll i en databas bestämdes det att i alla fall ett alternativ skulle använda sig av annan metod.

De tre populäraste CMS i dagsläget är Drupal, Joomla och Wordpress. Eftersom Drupal kräver mera konfiguration än de två andra samt användargränssnittet för administrationspanelen inte kändes lika användarvänligt som Joomla och Wordpress, valdes Drupal bort från tänkbara alternativ. Det tredje alternativet som togs med i jämförelsen var GetSimple.

2.2.1 Wordpress

Wordpress är ett innehållshanteringssystem som lanserades år 2003. Den första versionen av Wordpress var ämnat att användas som ett bloggverktyg, men i dagens läge är Wordpress ett komplett innehållshanteringssystem. Wordpress är uppbyggt med PHP och MySQL och är licensierad under GPLv2. Nuförtiden hjälper över 100 personer till att förbättra Wordpress. Det används på många miljoner olika webbplatser och dagligen besöks webbplatser uppbyggda med Wordpress av miljontals besökare. (About Wordpress, u.å.).

För att ha möjlighet att installera Wordpress behöver webbservern minst ha stöd för PHP version 5.2.4 och MySQL version 5.0. Då man installerar Wordpress behöver man ha information om MySQL-databasen för att Wordpress ska veta var data ska sparas.



The screenshot shows the WordPress Admin Panel interface. On the left is a sidebar menu with options like 'Hem', 'Uppdateringar', 'Inlägg', 'Media', 'Sidor', 'Kommentarer', 'Utseende', 'Tillägg', 'Användare', 'Verktyg', and 'Inställningar'. The main content area displays a welcome message: 'Välkommen till WordPress!' followed by a 'Kom igång' button labeled 'Anpassa din webbplats' and a link 'ändra tema helt'. Below this is a 'Just nu' section with a table of site statistics:

Innehåll	Diskussion
1 Inlägg	1 Kommentar
1 Sida	1 Godkänd
1 Kategori	0 Väntande
0 Etiketter	0 Skräpposter

Additional information at the bottom of the dashboard includes: 'Tema Twenty Twelve med 6 widgets', 'Sökmotorer blockerade', and 'Du använder WordPress 3.5.1'.

Figur 2. Administrationspanelen i Wordpress.

Via administrationspanelens modul ”Just nu” har man tillgång till att se t.ex. hur många inlägg, sidor, kategorier och kommentarer som för tillfället finns på webbplatsen. Andra menyalternativ ger administratören snabb tillgång till att påverka webbplatsens utseende och diverse andra inställningar.

För att skapa innehåll på en Wordpress-webbplats skapar man antingen inlägg eller sidor. Skillnaden mellan en sida och ett inlägg är att inlägg som skapas publiceras på sidan som är satt som bloggsida, medan en helt ny sida är statisk och innehållet där ändras inte ifall du publicerar ett inlägg. En annan skillnad är att inlägg kan ges olika kategorier medan en statisk sida inte har någon sådan funktion. Det finns ingen gräns för hur många olika statiska sidor som man kan skapa. (Wilson, Palmer, Rennick, & Torbert, 2011, s. 287–288).

För att ändra det grafiska utseendet av Wordpress används teman. Ett tema består av filer som tillsammans beskriver hur en webbplats ska visas. Teman är det första sättet man kan påverka hur en Wordpress-webbplats ser ut, eftersom ett tema innehåller regler för hur sidan ska visas upp. (Wilson, Palmer, Rennick, & Torbert, 2011, s. 471).

En helt ny installation av Wordpress ger användaren möjlighet att skapa en fullt funktionerande webbplats, men genom att lägga till tillägg kan man skapa en mycket mera avancerad webbplats. Om man t.ex. installerar ett tillägg för näthandel kan man skapa en webbplats som innehåller en webbutik. Tilläggen påverkar kraftigt hur webbplatsen fungerar. Det finns sådana tillägg som lägger till någon enstaka funktion, medan andra kan ändra helt på webbplatsens funktionalitet. (Miller, Sabin-Wilson, Palmer, Rennick, & Torbert, 2011, s. 615).

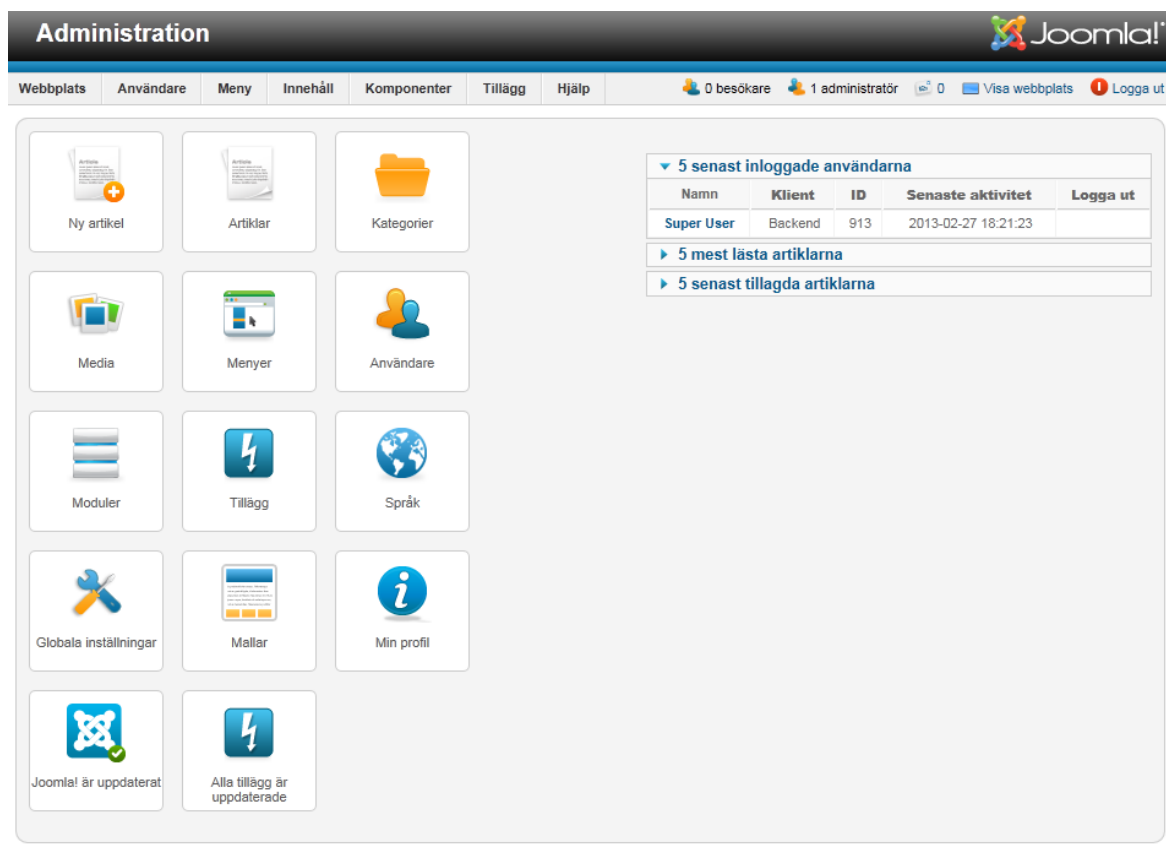
Som anledningar till varför man kunde använda Wordpress kan nämnas:

- Krävs i allmänhet inte mer än fem minuter för att installera Wordpress.
- Administrationspanelen är användarvänlig, till de inbyggda funktionerna hör t.ex. publicering av inlägg och sidor där man även kan inkludera mediafiler.
- Enkelt att utvidga, det finns många olika teman och tillägg som kan installeras till Wordpress.

(Leary, 2010, s. 23–26).

2.2.2 Joomla

Joomla har funnits sedan år 2005 och är licenserad under GPLv2. Versionen som lanserades år 2005 var skapad av utvecklare som tidigare hade utvecklat innehållshanteringssystemet Mambo CMS, som har funnits sedan år 2002. Likt många andra CMS kräver Joomla MySQL och PHP. Att Joomla är ett populärt CMS märks tydligt enligt statistik gjord i januari 2012 då 2,8 % av alla webbplatser i världen drevs av Joomla.



Figur 3. Administrationspanelen i Joomla.

Skapande av innehåll till en webbplats byggd på Joomla görs med artiklar. En artikel gjord med Joomla består oftast av en rubrik, text och inställningar. Artiklar som skapas kan delas in i kategorier för att lättare navigera på webbplatsen, man kan t.ex. skapa en meny för olika kategorier och då besökaren väljer en kategori visas alla artiklar som hör till den kategorin upp.

Mallar i Joomla anger hur sidans innehåll visas upp och placeras. Då man väljer mall ska man gå efter hur man själv vill att webbplatsen ska se ut, t.ex. om man vill ha en webbplats som ska användas för ett företag ska man leta efter en mall som är gjord för detta ändamål. Mallar kommer även med olika alternativ för att påverka placeringen eller

utseende av innehåll. Det betyder att ifall administratören inte är nöjd med att t.ex. ha menyn placerad högt uppe på webbplatsen, kan han utan att ändra mall ändra så att menyn är placerad på ett annat ställe av webbplatsen.

Om man behöver funktionalitet utöver det som kommer med en ny Joomla-installation kan man installera så kallade utbyggnader. Det finns fem olika typer av utbyggnader för att ändra på eller lägga till funktionalitet med Joomla:

- Komponenter, den mest avancerade typen av utbyggnad, t.ex. ett kontaktformulär.
- Moduler, enkla utbyggnader, t.ex. möjlighet att se vem som är inloggad vid webbplatsen.
- Tillägg, avancerade utbyggnader, t.ex. för att filtrera bort otillåtna ord ifall man publicerar en artikel.
- Mallar, ställer in hur sidans innehåll ska visas.
- Språk, ger möjlighet att använda sig av flera språk, både i administratördelen samt och den egentliga webbplatsen.

(Extension types, 2011).

Orsaker till varför Joomla kunde användas är:

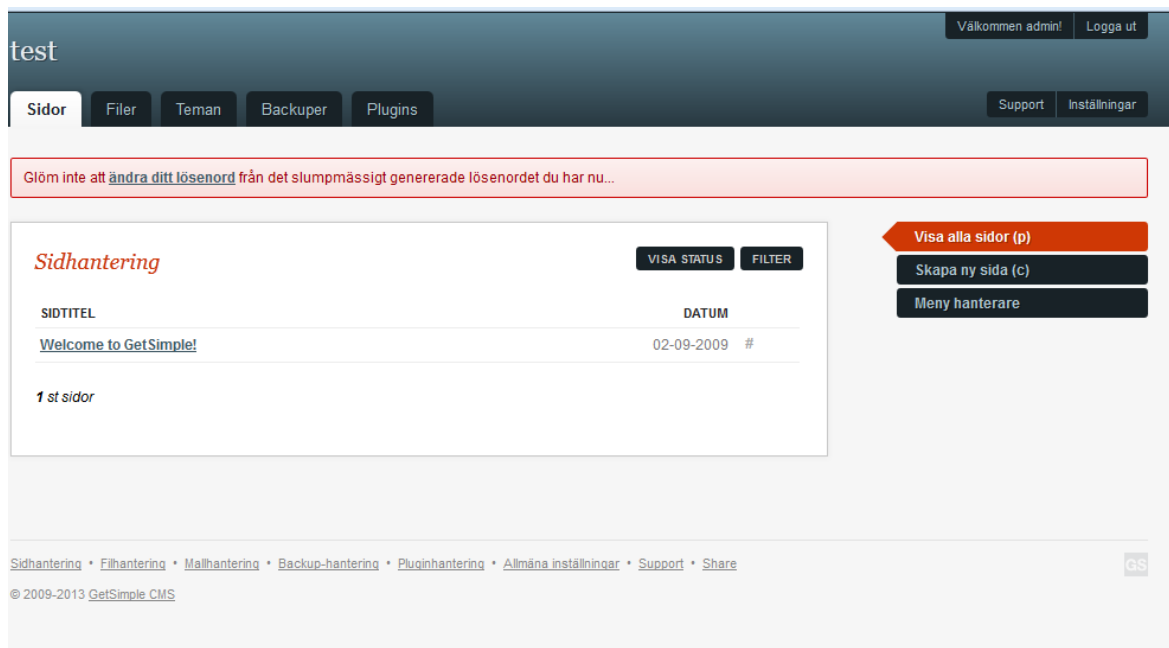
- Det är lätt att använda, efter att man lärt sig grunderna går det snabbt att skapa fina Joomla-webbplatser.
- Det är snabbt, det finns många inbyggda funktioner som gör det lätt att lägga till funktionalitet på webbplatsen.
- Många utbyggnader, på Joomlas webbplats finns över 6000 utbyggnader att ladda ner ifall man behöver ny funktionalitet till en webbplats.

(Burge, 2012, s. 2).

2.2.3 GetSimple

GetSimple lanserades år 2009 under GPLv3-licensen och är baserat på PHP och XML. Eftersom GetSimple lagrar information i XML-format så krävs inte mer av webbservern än

stöd för PHP 5.2 samt att Apache-modulen SimpleXML är aktiverad. GetSimple utvecklas av ett relativt litet utvecklingsteam bestående av sex personer. (About GetSimple, 2013).



Figur 4. Administrationspanelen i GetSimple.

För att skapa innehåll med GetSimple skapar man sidor. En sida består av en rubrik, text samt några inställningar. Eftersom systemet är mycket mindre avancerat än t.ex. Joomla är dessa inställningar också ganska begränsade, vilket inte behöver vara dåligt. Några inställningar man har möjlighet att påverka är:

- Om sidan är publik eller privat, för att ha möjlighet att redigera en sida innan man är redo att visa upp den publikt.
- Etiketters och nyckelord, för att enkelt kunna hitta motsvarande sidor, t.ex. sidor gällande sport används etiketten sport, då besökaren söker innehåll som gäller sport listas alla sidor med den etiketten.
- Meta-beskrivning, ger sidan en beskrivning som används av sökmotorer.

Temat används för att påverka hur innehållet visas upp. För att klassificeras som ett tema behövs det i alla fall en fil som beskriver placeringen av innehåll, en CSS-fil för att beskriva den grafiska designen samt en bild för hur temat ser ut. Därtill kan ett tema innehålla filer som innehåller t.ex. temaspecifika funktioner. För att lägga till mera

funktionalitet till en GetSimple-installation kan man installera tillägg. Till några av de populäraste tilläggen kan nämnas galleri- samt söktillägg.

Några orsaker till varför GetSimple skulle kunna vara ett bra alternativ är:

- Baserat på XML, i stället för att behöva kontakta en databas för att lagra eller visa upp sidor lagras det direkt på webbservern.
- Ångra-knapp, ångra den senaste ändringen direkt efter utförd ändring, t.ex. ångra innehåll på en sida och få tillbaka innehållet som fanns före ändringen.
- GetSimple är skapat för användare som inte behöver skapa avancerade webbplatser.

(About GetSimple, 2013).

2.3 Jämförelse av alternativ

2.3.1 Administrationspanel

De absolut enklaste men också mest lättförståeliga administrationspanelen hade GetSimple. För en användare går det snabbt och enkelt att t.ex. skapa sidor eller byta teman. En annan mycket användbar funktion hos GetSimple är möjligheten att ångra sin ändring genast efter att man sparat en händelse, något som varken Wordpress eller Joomla erbjuder.

Wordpress-gränssnittet kändes lättare att lära sig än Joomlas. En orsak till detta är att Joomla erbjuder mera alternativ för inställningar av innehåll än vad Wordpress gör, och speciellt mycket mera än vad GetSimple gör. Detta behöver dock inte vara något negativt, det är mycket beroende av vad användaren förväntar sig behöva lära sig om innehållshanteringssystemet och vilka inställningar som användaren förväntar sig kunna ändra på.

2.3.2 Uppdateringar

Både Wordpress och Joomla är uppbyggda så att det meddelas via administrationspanelen då uppdateringar finns tillgängliga. Vid uppdateringar är det rekommenderat att ta en säkerhetskopia av mappen med innehållshanteringssystemet samt databasen om sådan används, speciellt om det är en uppdatering av själva innehållshanteringssystemet. Om

man inte gör det kan det vara mycket svårt att återställa systemet ifall problem vid uppdateringar uppstår.

Wordpress erbjuder två sätt att hantera uppdateringar. Det första och enklare sättet för en oerfaren användare är att låta Wordpress själv ta hand om processen. Då uppdateringar finns tillgängliga meddelas det via administrationspanelen att uppdateringar finns tillgängliga och via en knapptryckning kan man uppdatera tillägg, teman eller Wordpress-installationen. För att automatiska uppdateringar ska fungera bör man ha tillgång till Wordpress-mappen via FTP eller SSH, viktigt är att ägaren av filerna motsvarar FTP- eller SSH-användaren. Det andra sättet för att uppdatera är att ladda ner filerna manuellt och ersätta de gamla filerna med de nya. (Leary, 2010, s. 35–37).

Uppdateringsprocessen för Joomla är liknande den som finns tillgänglig för Wordpress, man kan antingen välja att manuellt skriva över gamla filer eller låta Joomla-installationen hämta ner filerna och uppdatera. Det finns undantag när automatisk uppdatering inte fungerar för Joomla och Wordpress, vanligtvis då någon riktigt stor uppdatering lanseras som kräver manuellt arbete eller när filerna till uppdateringen inte finns tillgängliga på Wordpress respektive Joomlas egna webbserver för nerladdning. För att uppdatera GetSimple är enda möjligheten i skrivande stund att ladda ner nya versionen och ersätta de gamla filerna.

Utvecklarna av Wordpress lanserar större uppdateringar tre gånger per år, medan Joomla lanserar två per år. Eftersom GetSimple utvecklas av en mycket mindre grupp är det svårt att veta hur ofta stora uppdateringar sker, men historiskt sett har uppdateringar skett i snitt två gånger per år.

2.3.3 Teknik

Eftersom GetSimple är baserat på lagring i XML-format märks stor skillnad på lagringen som Wordpress och Joomla använder sig av. För att illustrera skillnaden skapades en sida med GetSimple samt Wordpress med likvärdigt innehåll.

```

<?xml version="1.0" encoding="UTF-8"?>
<item>
  <pubDate>Mon, 04 Mar 2013 18:38:35 +0000</pubDate>
  <title><![CDATA[Test sida]]></title>
  <url><![CDATA[index]]></url>
  <meta><![CDATA[]]></meta>
  <metad><![CDATA[]]></metad>
  <menu><![CDATA[Hem]]></menu>
  <menuOrder><![CDATA[1]]></menuOrder>
  <menuStatus><![CDATA[Y]]></menuStatus>
  <template><![CDATA[template.php]]></template>
  <parent><![CDATA[]]></parent>
  <content><![CDATA[&lt;p&gt;Innehåll&lt;/p&gt;]]></content>
  <private><![CDATA[]]></private>
  <author><![CDATA[admin]]></author>
</item>

```

Figur 5. En sida i XML-format skapad med GetSimple.

post_author	post_date	post_date_gmt	post_content	post_title	post_excerpt	post_status
1	2013-03-04 19:43:32	2013-03-04 18:43:32	Innehåll	Test sida		publish

comment_status	ping_status	post_password	post_name	to_ping	pinged	post_modified	post_modified_gmt
open	open		test-sida			2013-03-04 19:53:10	2013-03-04 18:53:10

post_content_filtered	post_parent	guid	menu_order	post_type	post_mime_type	comment_count
	0	http://localhost/wordpress/?page_id=15	0	page		0

Figur 6. Motsvarande sida i en MySQL-databas i en Wordpress-installation.

I figurerna 5 och 6 kan ses hur GetSimple respektive Wordpress sparar en sida. Utöver att Wordpress innehåller mera inställningar för sidan kan man se att sidorna är likvärdiga förutom tekniken att lagra information.

2.4 Val av innehållshanteringssystem

Wordpress blev det alternativ som skulle användas. Detta p.g.a. flera orsaker:

- Wordpress egna bibliotek innehåller över 24 000 tillägg och över 1 700 teman för nerladdning.
- Wordpress administrationspanel kändes lättare att bli bekant med än Joomlas.

- GetSimple påminde om det innehållshanteringssystem som är i användning för tillfället. En del av uppgiften var att hitta ett innehållshanteringssystem som gav möjligheter utöver det som det gamla gav.
- Wordpress erbjuder en inbyggd lösning för att centralisera en installation av Wordpress, vilket ger möjlighet att skapa många webbplatser på en installation.

En annan faktor som påverkade valet var Wpmudev, som beskrivs i kapitel 4.2.3.

3 Tekniker

3.1 Webb- och skriptspråk

3.1.1 HTML

HTML står för Hyper Text Markup Language. HTML är ett märkspråk och använder sig av taggar för att beskriva innehållet på webbsidor. HTML-dokument kan tolkas av webbläsare, såsom t.ex. Internet Explorer och Mozilla Firefox. Webbläsare känner till hur HTML ska tolkas och i stället för att visa taggarna så tolkar den hur innehållet i dokumentet ska visas. (Refsnes, Refsnes, & Refsnes, 2010, s. 1–2).

HTML-dokument skapas av taggar som består av start- och sluttaggar som i sin tur bildar HTML-element. Ett HTML-element är allt innehåll från dess start tagg till sluttagg. De flesta element kan nästlas så att de innehåller flera element. Det element som alltid kommer att innehålla flest element är själva HTML-taggen som definierar att det är ett HTML-dokument. Tomma element är sådana som saknar innehåll och kan avslutas i samma skede som man anger starttaggen, ett exempel på en sådan är ett linjebyte som definieras som `
`. (Refsnes, Refsnes, & Refsnes, 2010, s. 12–14).

För att ge element ytterligare information används attribut. Attributen definieras alltid i starttaggen och är uppbyggda som par med ett namn och ett värde, t.ex. `name="value"`. De attributen som stöds av de flesta HTML-elementen är:

- Id, ger elementet ett unikt id, användbart t.ex. vid manipulering av element med hjälp av JavaScript.
- Class, ger elementet en specifik klass, användbart för att använda en klass definierad i ett CSS-dokument.

- Style, anger CSS direkt vid definieringen av elementet.
- Title, ange vad som visas då man för muspekaren över elementet via webbläsaren.

Då man anger värdet på ett attribut ska man använda små versaler, i HTML4 stöds användning av stora bokstäver, men är inte rekommenderat, i den framtida standarden HTML5 är det inte tillåtet. (Refsnes, Refsnes, & Refsnes, 2010, s. 16–17).

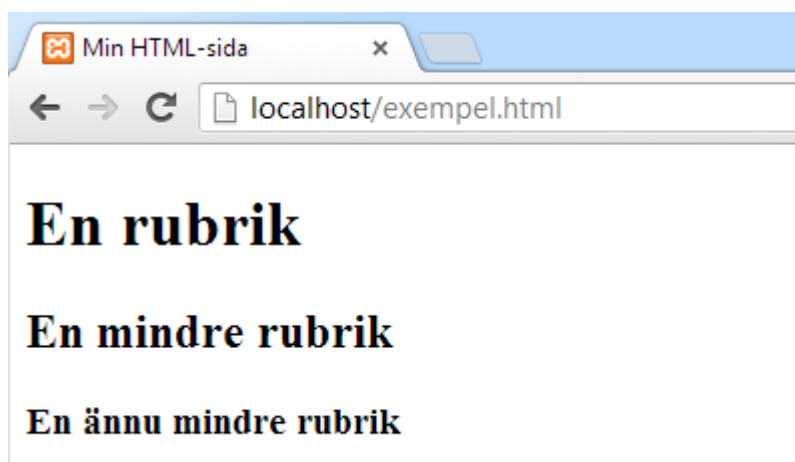
HTML-formulär används för skicka information från element till en server, t.ex. från ett textfält. Den mest använda formulär-taggen är input-taggen. Input-taggens typ definieras med hjälp av attributet Type, de fyra vanligaste typerna är:

- Text, ett textfält som man kan skriva in text i.
- Checkbox, skapar alternativ som går att markera, även fast boxarna har samma namn går det att välja flera.
- Radiobox, skapar alternativ som går att markera, om det finns flera element med samma namn går det endast att markera ett alternativ.

För att skicka informationen i ett formulär används typen submit på taggen input. I taggen form kan man ange attributet action tillsammans med en URL, som har i uppgift att hantera informationen som den tog emot. (Refsnes, Refsnes, & Refsnes, 2010, s. 98–105).

Kodexempel 1. Ett exempel på ett HTML-dokument.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>Min HTML-sida</title>
</head>
<body>
  <h1>En rubrik</h1>
  <h2>En mindre rubrik</h2>
  <h3>En ännu mindre rubrik</h3>
</body>
</html>
```



Figur 7. Hur en webbläsare tolkar HTML-dokumentet.

3.1.2 CSS

CSS är en förkortning av Cascading Style Sheets. Med hjälp av CSS kan man definiera utseendet och kontrollera layouten på webbsidor. Man kan t.ex. skapa skuggor, kontrollera utseendet på kanter eller var bilder placeras. CSS kan jämföras med ett program som Microsoft Word där man kan skapa stil-regler för sitt innehåll, i CSS gäller samma princip. CSS är inte ett alternativ till HTML, det är två helt olika språk, med helt olika uppgifter, stilmallen är till för att ange hur ett HTML-dokument ska visas. (McFarland, 2009, s. 1).

Det finns tre olika sätt att använda sig av CSS. Man kan antingen skapa regler för att definiera ett enskilt elements stil med hjälp av attributet style, skapa en stilmall i samma fil som HTML-dokumentet eller skapa en helt ny fil med filändelsen .css. Det rekommenderade sättet är att skapa en skild fil p.g.a. av att det är mycket enklare att underhålla om man har många olika webbsidor som använder sig av samma stil, det enda som behöver redigeras är en enskild fil.(McFarland, 2009, s. 34–36).

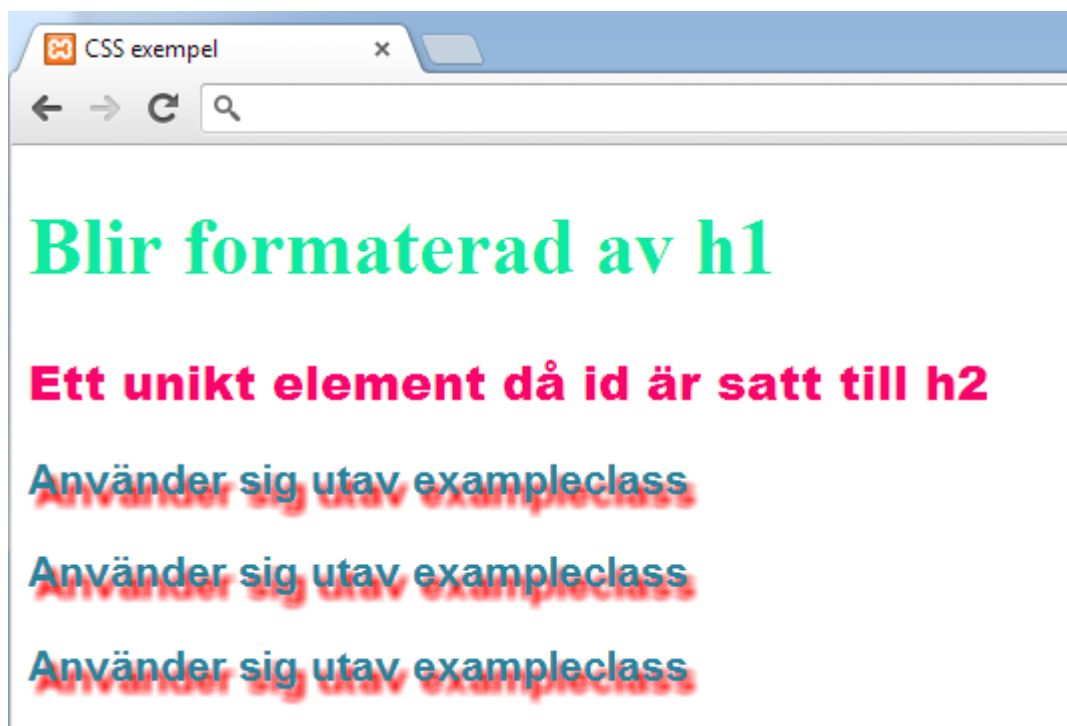
Varje CSS-stil har två olika delar, det första anger vad som ska formateras, t.ex. en paragraf, det andra hur det ska formateras, t.ex. vilken färg paragrafen ska vara. Då man anger regler har man tre alternativ om vad man kan ange som ska formateras. Man kan välja att formatera alla element med samma tagg, formatera element på basen av det id som är satt för elementet eller så kan man välja att skapa en regel för element som använder samma klass. Id används för att formatera unika element medan användning av klass kan användas på flera element. (McFarland, 2009, s. 49–53).

Kodexempel 2. Ett exempel på hur CSS kan användas för att ändra på formatering av element.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>CSS exempel</title>
  <style type="text/css">
    h1 {
      color: #0CE9A0;
      font-size:30pt;
    }
    #h2 {
      color: #F90269;
      font-family:arial black;
    }
    .exampleclass{
      color: #26839B;
      font-family:arial;
      font-size:16pt;
      font-weight:bold;
      text-shadow: red 0.2em 0.3em 0.2em
    }
  </style>
</head>
<body>
  <h1>Blir formaterad av h1</h1>
  <h2 id="h2">Ett unikt element då id är satt till h2</h2>
  <h3 class="exampleclass">Använder sig utav exampleclass</h3>
  <label class="exampleclass">Använder sig utav exampleclass</label>
  <p class="exampleclass">Använder sig utav exampleclass</p>
</body>
</html>

```



Figur 8. Hur webbläsaren visar HTML-dokumentet i kodexempel 2.

3.1.3 PHP

PHP (rekursiv akronym för PHP: Hypertext Processor) är ett skriptspråk som är främst menat till användning inom webbutveckling. Detta skriptspråk är serverbaserat, vilket betyder att till skillnad från t.ex. JavaScript så exekveras PHP-kod på webbservern i stället för på klienten. PHP-kod kan enkelt bäddas in i ett HTML-dokument genom att använda sig utav start- och sluttaggen för PHP, `<?php` och `?>`. Zend Engine, en programtolk, sköter om tolkning av PHP-kod. Ifall PHP-kod förekommer så försöker Zend Engine tyda koden innanför taggarna och skicka tillbaka informationen till klienten så klienten kan visa upp resultatet. (What is PHP?, 2013).

Nuvarande version av PHP är version 5, som blev tillgänglig 2004. Den drivs i huvudsak av den uppdaterade Zend Engine 2.0. PHP användes på över 240 miljoner webbplatser i februari 2013 (PHP just grows & grows, 2013).

Kodexempel 3. Användning av PHP för att visa upp dagens datum och veckodagens namn.

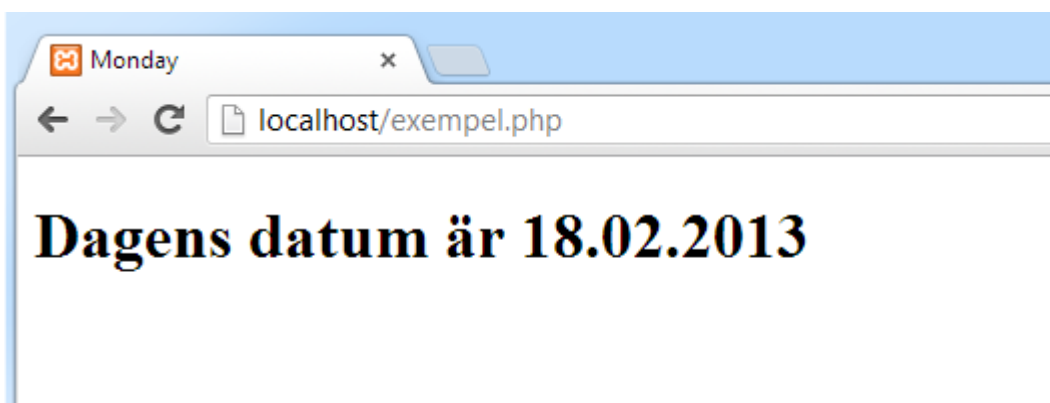
```
<?php

/* Skapa två variabler med hjälp av date klassen
 * En för endast namnet på veckodagen och den
 * andra för att visa datumet
 */

$nameofday = date("l");
$today'sdate = date('d.m.Y');

?>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title><?php echo $nameofday ?></title>
</head>
<body>
  <h1>Dagens datum är <?php echo $today'sdate ?></h1>
</body>
</html>
```



Figur 9. Resultat av kodexempel 3.

3.2 Övriga tekniker

3.2.1 MySQL

MySQL är ett databashanteringssystem som lanserades 1996 av det svenska företaget TcX DataKonsult AB. MySQL har blivit nerladdat över 100 miljoner gånger och har användare som t.ex. NASA och Google. Orsaker till att MySQL har blivit så populärt förutom att det är gratis är bland annat att MySQL stödjer de flesta operativsystemen och är mycket snabbt. MySQL är lanserat under GPLv2-licensen. (Gilmore, 2010, s. 477–478).

För att hantera MySQL-databaser och dess innehåll kan användas en klient, antingen en klient med grafiskt användargränssnitt eller ett textbaserat gränssnitt. I figur 10 används en textbaserad mysql-klient för att skapa tabellen *athletes*. Tabellen *athletes* består av fyra kolumnder, *id* används som identifieringskolumn, medan de tre andra kolumnerna används för att spara information om personen.


```

c:\wamp\bin\mysql\mysql5.5.24\bin\mysql.exe
mysql> CREATE TABLE athletes (
-> id INT NOT NULL AUTO_INCREMENT,
-> firstname VARCHAR(25) NOT NULL,
-> lastname VARCHAR(25) NOT NULL,
-> sport VARCHAR(45) NOT NULL,
-> PRIMARY KEY(id));
Query OK, 0 rows affected (0.00 sec)

mysql> INSERT INTO athletes(firstname,lastname,sport) values ('Petter','Northug',
,'Skiing'),('Andreas','Romar','Alpine Skiing');
Query OK, 2 rows affected (0.01 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> select * from athletes;
+----+-----+-----+-----+
| id | firstname | lastname | sport |
+----+-----+-----+-----+
| 1 | Petter | Northug | Skiing |
| 2 | Andreas | Romar | Alpine Skiing |
+----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> _

```

Figur 10. Skapande samt insättning av data i MySQL-tabell.

3.2.2 PHP/CURL

cURL står för Client for URLs. Med hjälp av cURL kan man skicka filer från och till en webbserver. Protokoll som stöds av cURL är t.ex. HTTP, HTTPS och FTP. I PHP-applikationer är det vanligt att man använder sig av det portabla biblioteket libcurl som innehåller ett användarvänligt gränssnitt för att använda sig av cURL. För att kunna använda sig av libcurl behöver webbservern ha stöd för libcurl i PHP-installationen. Genom användning av cURL kan webbservern agera som om den vore en klient som skapar eller skickar information på samma sätt som en webbläsare gör. (Khalid, 2006, s. 7–8).

Några vanliga användningar av PHP/CURL är hämtning av webbsidor, skickande av information och autentisering på webbsidor. I kodexempel 4 visas hur man kan visa upp en webbplats med hjälp av PHP/CURL.

Kodexempel 4. Visning av en webbplatsens innehåll med hjälp av PHP/CURL.

```

<?php
$url = "http://www.vasabladet.fi";
$ch = curl_init(); // Skapa en CURL session.
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1); // Returnera sidans innehåll
curl_setopt($ch, CURLOPT_URL, $url); // Skicka URL som en parameter
$result = curl_exec($ch); // Ta emot sidans innehåll

curl_close($ch); // Stäng CURL sessionen.
echo $result; // Visa upp sidans innehåll.
?>

```

4 Utförande

Denna del omfattar det praktiska utförandet av examensarbetet. Till det praktiska hörde att installera en fungerande installation av Wordpress samt att skapa en webbapplikation för att lägga till webbplatser.

4.1 Planering

Efter att innehållshanteringssystemet var valt blev nästa skede att planera hur installationen skulle förverkligas. Eftersom Wordpress kommer med inbyggt stöd för att kunna driva en installation som ett nätverk beslutades det att den funktionaliteten skulle utnyttjas. Tillsammans med uppdragsgivaren beslutades att i första stadiet av lanseringen skulle varje kund ha möjlighet att lägga till endast en webbplats. Vid planeringen bestämdes även att de teman och tillägg som ska finnas tillgängliga för kunderna ska vara uppdaterade så att de fungerar med den senaste versionen av Wordpress samt att de är översatta till svenska respektive finska.

För att ge kunder möjlighet att lägga till webbplatser skulle en webbapplikation vid Mitt Anvia portalen skapas. Tillsammans med uppdragsgivaren beslutades det att applikationen skulle vara liknande den som varit i användning för den gamla webbplats tjänsten. Den skulle kunna hantera skapande, radering samt byte av lösenord för användare.

4.2 Wordpress

För att kunna installera Wordpress behövdes en databas. På webbservern där Wordpress skulle installeras skapades därför en databas, för att öka säkerheten valdes ett annat namn än Wordpress på databasen. Det skapades även ett MySQL-användarnamn som endast hade tillgång till den valda databasen för att vid möjliga intrång endast riskera att Wordpress-databasens data skulle bli tillgänglig.

4.2.1 Installation av Wordpress

För att installera Wordpress behövdes information om MySQL-databasen som skapades. Förutom informationen om databasen fick man även ange vilka inloggningsuppgifter som skulle gälla och vad namnet på webbplatsen skulle vara.

Nedan anger du dina uppgifter för databasanslutningen. Om du är osäker på dessa uppgifter, kontakta ditt webshotell/administratör.

Databasnamn	<input type="text" value="wordpress"/>	Namnet på databasen du vill köra WP under.
Användarnamn	<input type="text" value="användarnamn"/>	Ditt användarnamn för MySQL
Lösenord	<input type="text" value="lösenord"/>	...och ditt lösenord för MySQL.
Databasvärd	<input type="text" value="localhost"/>	Du får denna information från ditt webshotell om localhost inte fungerar.
Tabellprefix	<input type="text" value="wp"/>	Om du vill köra flera installationer av WordPress i en och samma databas, ändra det här.

Figur 11. Information om MySQL-databasen som ska användas för Wordpress.

Efter installationen hade man en fungerande Wordpress-webbplats. För att göra så att Wordpress-installationen skulle kunna driva många olika webbplatser ändrades installationen om till en nätverksinstallation. För att ändra om Wordpress till en nätverksinstallation gjordes följande:

1. Lade till i Wordpress konfigurationsfil "wp_config.php" följande:


```
define('WP_ALLOW_MULTISITE',true).
```
2. Efter inloggning fanns nu ett nytt val i administrationspanelen, installera nätverk.
3. Vid valet att installera nätverk tog Wordpress hand om installationen och angav vad som behövde ändras i konfigurationsfilen och i filen .htaccess.

Nu hade Wordpress möjlighet att skapa nya webbplatser för olika användare. Följande steg blev att installera tillägg och teman som skulle vara tillgängliga över nätverket.

4.2.2 Wpmudev

Wpmudev är en webbplats där man kan köpa specifika teman och tillägg till Wordpress. Man har även möjlighet att köpa en licens som ger rättighet till alla teman och tillägg. Förutom tillgång till nedladdningar av tillägg och teman ger även en licens kunden tillgång till diverse manualer och support för tillägg. Allt på Wpmudev är licenserad under GPL, vilket innebär att ifall man behöver redigera ett tillägg som man installerat på sin Wordpress-installation, så har man tillåtelse att fritt göra det. För att ha tillgång till uppdateringar av tillägg och teman samt support behövs en aktiv licens. (Wpmu Premium, u.å.).

Valet att använda sig av Wpmudev grundar sig på den stora tillgången på olika tillägg som kan anpassas till en nätverksbaserad installation av Wordpress. Att på förhand veta att de tillägg och teman som köptes från Wpmudev skulle bli uppdaterade i samband med Wordpress-uppdateringar var även en betydande orsak.

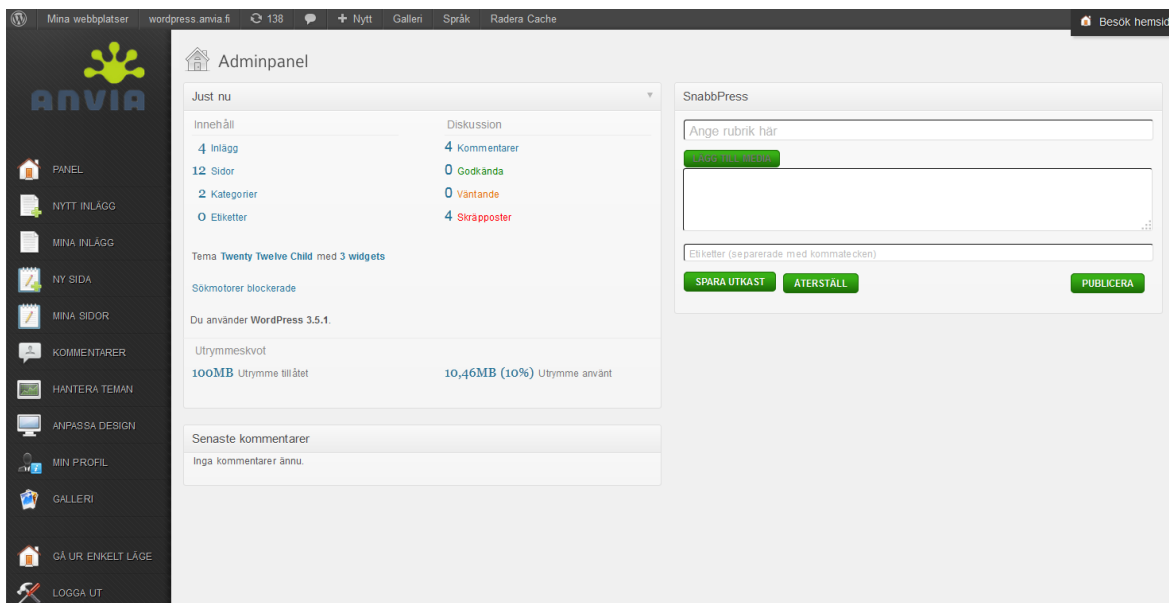
4.2.3 Tillägg

Eftersom målet med uppgiften var att skapa webbplatser via Mitt Anvia, användes ett tillägg för att möjliggöra detta. För skapande av webbplatser och användare installerades ett tillägg som hette WHMCS Multi-Site Provisioning. Det gav möjligheten att skapa användare och webbplatser med hjälp av PHP/CURL.

Eftersom nätverkslösningen av Wordpress riskerade att bli långsammare då många webbplatser skapas, bestämdes det att datalagringen skulle delas upp på flera databaser. Wpmudev erbjöd en enkel lösning för att möjliggöra detta, Multi-DB. Enligt Wpmudev skulle en nätverkslösning kunna driva 5 000 webbplatser på endast en databas utan problem, men efter 5 000 webbplatser skulle Wordpress-installationen riskera att bli långsammare. Därför valdes det att använda 16 databaser för att driva en Wordpress-installation, vilket var Wpmudev:s rekommendation för att driva 5 000 – 50 000 webbplatser. För övergången till 16 databaser behövde man skapa nya databaser samt ange i Multi-DB:s konfigurationsfil namnet på dessa databaser. För att dela upp datan i 16 databaser hade Multi-DB ett skript som kunde användas.

EasyBlogging är ett tillägg som används för att ändra om administrationspanelen hos Wordpress. Detta tillägg kommer från Wpmudev och är i praktiken endast en vy för att ge användare en ännu enklare administrationspanel än den panel som Wordpress erbjuder.

Genom detta tillägg har användare möjlighet att välja mellan det vanliga Wordpress-gränssnittet och det som visas i figur 12.



Figur 12. Wordpress-administrationspanel med tillägget Easy Blogging installerat.

Alla inlägg som installerats togs inte med i detta arbete eftersom den listan blir rätt lång. Som exempel kan nämnas tillägg som ger dessa möjligheter:

- Påverka alternativen som användare har tillgång till i administrationspanelen.
- Möjlighet att skapa galleri med olika möjligheter för att visa upp bilder på en webbplats.
- Möjlighet för användare att skapa ett kontaktformulär.

4.3 Webbapplikation

4.3.1 Mitt Anvia

Webbapplikationen följer MVC-strukturen i Zend ramverket såsom resten utav Mitt Anvia. Det finns fyra olika sidor för att visa upp innehåll:

- En startsida, ifall användaren har en webbplats visas information om webbplatsen, sidan ger även tillgång till att radera eller byta lösenord. Om användaren inte har en webbplats visas en länk till sidan där användaren kan skapa en webbplats.

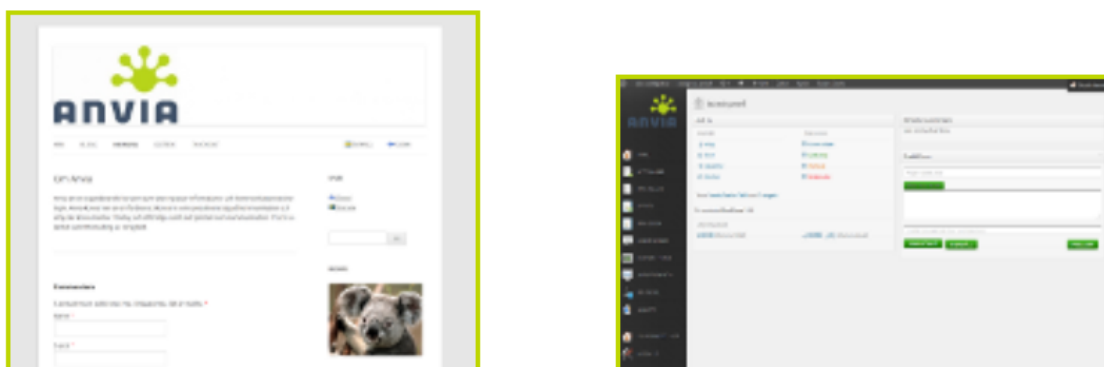
- En sida där användaren kan skapa en webbplats.
- En sida där användaren kan radera webbplatsen.
- En sida där användaren kan byta lösenord.

Startsidan för Wordpress-tjänsten ser ut som i figur 13. Bilderna som syns i figur 13 är skärmdumpar från en webbplats som skapats vid Wordpress-installationen, ifall besökaren klickar på bilderna visas de upp i en större storlek.

Wordpress webbplats

Webbplats	Användarnamn	E-post
-----------	--------------	--------

[Skapa en ny webbplats](#)



Figur 13. Webbapplikationen vid Mitt Anvia.

En kund som är inloggad vid Mitt Anvia kan nå sidan i figur 13 ifall kunden har ett aktivt internetabonnemang hos Anvia. Ifall kunden inte har en Wordpress-webbplats kan han välja att skapa en. Sidan som hanterar skapandet av webbplatser är ett HTML-formulär (se kodexempel 5 och figur 14). För att placera samt ändra färger på element används CSS.

Kodexempel 5. En del av formuläret för att skapa en webbplats.

```
<form action="" id="f1" method="post" name="addWordpressPage">

<fieldset>
<legend><?=$this->translate('Give information for you website'); ?></legend>
<div class="infieldset">
  <input type="hidden" name="token" value="<?=$this->escape($this->token); ?>">
  <label class="con1" for="homepagename"><strong><?=$this->translate('Website name'); ?></strong>
  </label><br/>
  <input id="homepagename" type="text" name="homepagename" size="25"
  value="<?=$this->escape($this->homepagename); ?>"/>
  .wordpress.anvia.fi
  <br />
  <label class="con1"><strong><?=$this->translate('Homepage language'); ?></strong><br />
  <select name="language">
    <option value="fi" <?=$fi?><?=$this->translate('Finnish'); ?></option>
    <option value="sv_SE" <?=$se?><?=$this->translate('Swedish'); ?></option>
    <option value="en_US" <?=$en?><?=$this->translate('English'); ?></option>
  </select>
</label>
```

Wordpress webbplats

[« Tillbaka](#)

Ge information om din webbplats

Webbplatsens namn
 .wordpress.anvia.fi

Webbplatsens språk

Användarnamn

E-postadress

Ange lösenord, minst 8 tecken

Lösenord **Bekräfta lösenordet**

Skapa webbplats

Figur 14. Sidan som används för att skapa en webbplats.

Efter att kunden tryckt på ”Skapa webbplats” skickas formuläret och ”Controller” klassen gör diverse kontroller, se kodexempel 6. Ifall inga fel uppstår i kontrolleringen kontaktas ”Model” klassens funktion ”createPage”.

Kodexempel 6. Exempel på kontroller av lösenordet.

```
if(strlen($this->view->password) < 8){
    $this->view->errorMessage = 'Password too short';
    $this->view->successmessage = NULL;
    return $this->render('create');
}
if($this->view->password != $this->view->password2){
    $this->view->errorMessage = 'Passwords does not match';
    $this->view->successmessage = NULL;
    return $this->render('create');
}
if(!empty($this->view->homepage) && !empty($this->view->language) && !empty($this->view->username)){
    try{
        $res = $this->model->createPage($contractid,NULL,$customerid,$this->view->email,$this->view->homepage,'100',
            $this->view->username,$this->view->password,$owner_firstname,$owner_lastname,$this->view->language);
    }
```

Funktionen ”createPage” som anropas på näst sista raden i kodexempel 6 tar emot information om webbplatsen som ska skapas och skickar den vidare till klassen vid applikationsservern, läs mer i kapitel 4.3.2.

Ifall skapandet av webbplatsen lyckas sänds ett e-post-meddelande till vald adress med en bekräftelse på användarnamn samt lösenord. Kunden kan nu välja att börja skapa innehåll på sin webbplats, byta lösenord för sitt användarnamn eller radera sin webbplats.

Wordpress webbplats

Skapande av Wordpress webbplats lyckades				
Webbplats	Användarnamn	E-post		
thesis	demo	mats.stolpe@gmail.com	Byt lösenord	Radera

Figur 15. Användaren har skapat webbplatsen thesis.

Eftersom kunden nu har skapat en webbplats har han tillgång till att byta lösenord eller radera webbplatsen. I figur 16 och 17 visas webbsidorna som är gjorda för att byta lösenord respektive radera en Wordpress-webbplats.

Wordpress webbplats

[« Tillbaka](#)

Uppdatera lösenord

E-postadress

Lösenord

Bekräfta lösenordet

Uppdatera lösenord

Figur 16. Sidan för byte av lösenordet till Wordpress-webbplatsen.

Wordpress webbplats

[« Tillbaka](#)

Radering av Wordpress webbplats

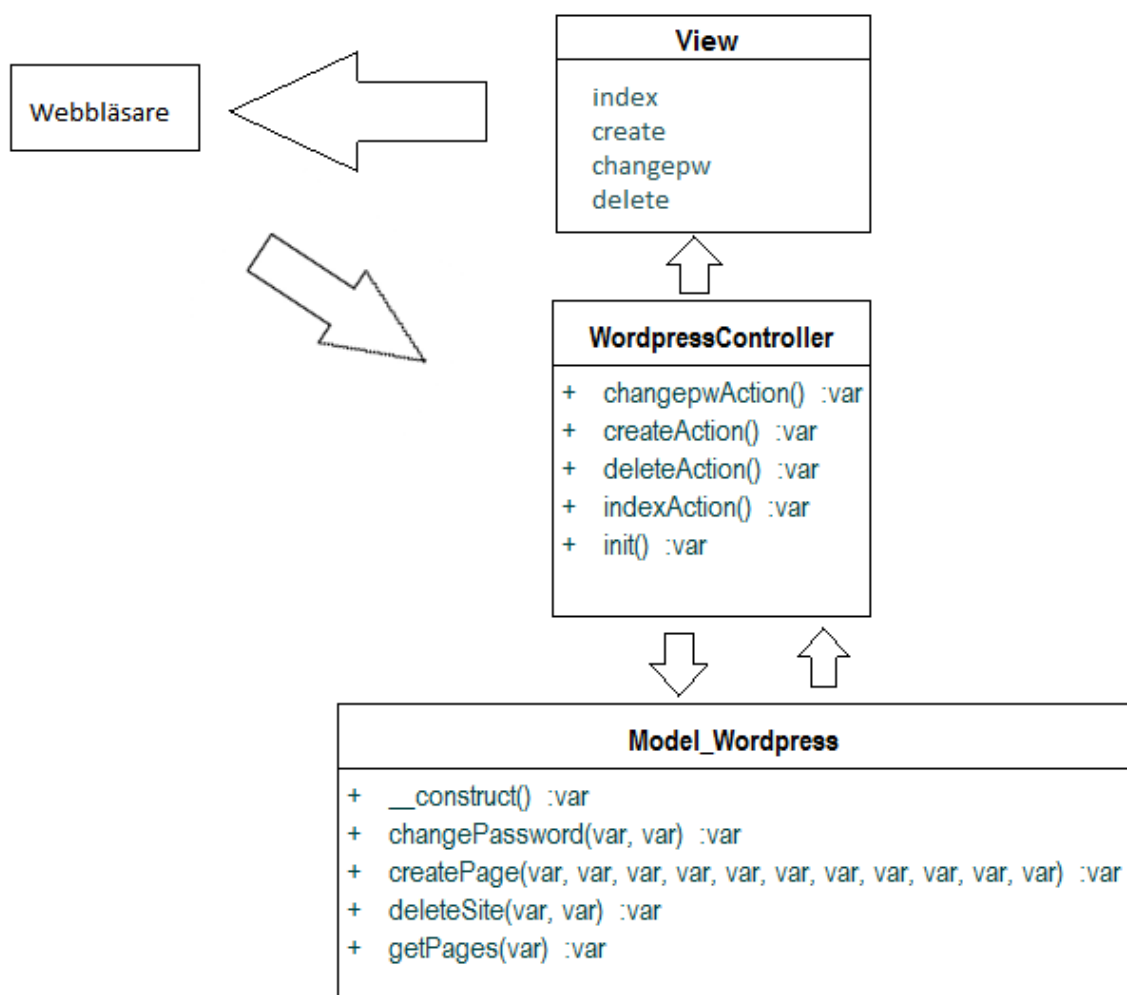
Webbplatsens namn
 .wordpress.anvia.fi

E-postadress

Radera

Figur 17. Sidan för att radera en Wordpress-webbplats.

I figur 18 visas en förenklad bild på hur MVC-strukturen är uppbyggd. WordpressController är den som klass som är utgångspunkten för alla förfrågningar som webbläsaren gör. I WordpressController finns instruktioner för vilket nästa steg som ska göras ifall webbläsaren gjort en förfrågning t.ex. kontakta funktioner i Model_Wordpress eller säga till vilken sida som ska visas upp i webbläsaren.



Figur 18. Hur webbapplikationen är uppbyggd.

4.3.2 Applikationsserver

Användningen av webbapplikationen Wordpress vid Mitt Anvia går via en separat server. Den hanterar de flesta tjänster som finns på Mitt Anvia. Då en kund väljer att t.ex. skapa en Wordpress-webbplats skickas informationen från Mitt Anvia till denna server över HTTP-protokollet. Allt gällande skapande, radering samt ändring av lösenord för en Wordpress-webbplats går via denna server. För hantering av Wordpress-webbplatser skapades klassen RestServiceWordpress. RestServiceWordpress innehåller funktioner för att t.ex. ta kontakt med ett tillägg för att skapa Wordpress-webbplatser.

Eftersom uppdragsgivaren behöver veta information om kunder som skapar Wordpress-webbplatser skapades en MySQL-databas. Denna databas används för att lagra information om vem som äger webbplatsen.

Kodexempel 7. PHP/CURL för att logga in samt försöka skapa en webbplats genom att kontakta tillägget WHMCS Multi-Site Provisioning i Wordpress-installationen.

```

public function createWordpressSite()
{
    $cookie = tempnam("/tmp", "WordpressCookie");
    $loginquery = array('log' => $this->_wordpress_login, 'pwd' => $this->_wordpress_pwd, 'wp-login' => 'Log In');

    $postfields = http_build_query($loginquery);
    $ch = curl_init($this->_wordpress_login_url);
    curl_setopt($ch, CURLOPT_POST, true);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $postfields);
    curl_setopt($ch, CURLOPT_USERAGENT, $this->_useragent);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie);
    curl_exec ( $ch );
    curl_close($ch);
    $theblog = array( 'action' => 'create', 'blog' => array('domain'=>$this->_request_params['homepage_name'],
        'email' => $this->_request_params['customer_email'],
        'username' => $this->_request_params['username'],
        'password' =>$this->_request_params['password'],
        'last_name' => $this->_request_params['owner_lastname'],
        'first_name' => $this->_request_params['owner_firstname'],
        'nickname' => $this->_request_params['owner_firstname'],
        'title' => $this->_request_params['homepage_name'],
        'language' => $this->_request_params['homepage_language']
    ));

    $postfields= http_build_query($theblog);
    $ch = curl_init($this->_wordpress_network_url);
    curl_setopt($ch, CURLOPT_POST, true);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $postfields);
    curl_setopt($ch, CURLOPT_USERAGENT, $this->_useragent);
    curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

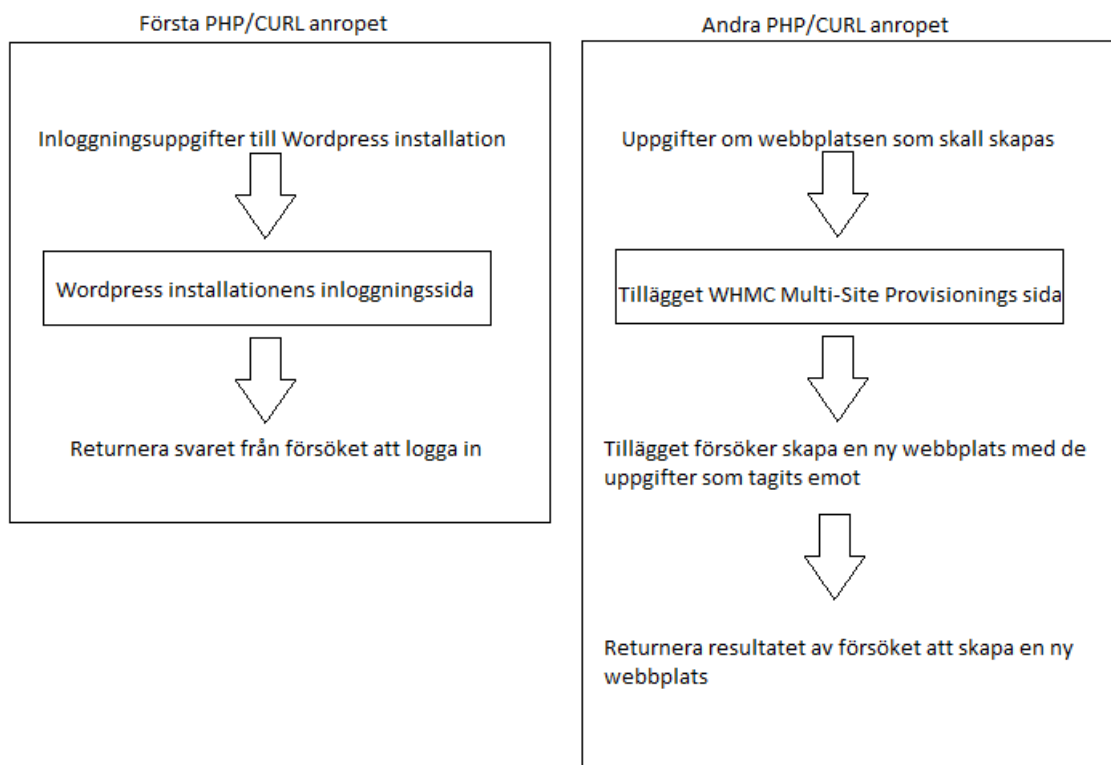
    $curlInfo =curl_exec($ch);
    $statusCode = curl_getinfo ( $ch, CURLINFO_HTTP_CODE );
    $resp =json_decode($curlInfo);

    curl_close($ch);

    $this->_newpassword = $resp->pass;
    if(isset($resp->error) || $resp->success != 1){
        throw new Exception( "Failed to create new blog error: " . $resp->error);
    }

    return $curlInfo;
}

```



Figur 19. Hur PHP/CURL används i kodexempel 7.

Det som figur 18 visar är hur koden i kodexempel 7 fungerar. Först används PHP/CURL för att skicka inloggningsuppgifter för en användare till sidan där man kan logga in till Wordpress-installationen. Svaret som bland annat innehåller information om inloggningen lyckades skickas därefter tillbaka till applikationsservern. För det andra PHP/CURL-anropet anges som URL tilläggets sida i Wordpress-installationen samt information om webbplatsen som ska skapas. Efter att tillägget försökt skapa en webbplats skickas information tillbaka till applikationsservern innehållande bl.a. om en ny webbplats kunde skapas.

Det som inte fanns med i kapitel 3.2.2 är att CURL även tillåter sessioner. Det som sker först i kodexempel 6 är att en tom fil skapas och sedan anges i `CURLOPT_COOKIEJAR` var den filen finns. Då det första PHP/CURL-anropet görs lagras information om det anropet i filen. I det andra PHP/CURL-anropet anges som värde på `CURL_COOKIEFILE` var filen som användes i det första anropet finns, för att kunna fortsätta i samma session. Med hjälp av informationen som finns i filen kan man därför fortsätta vara inloggad som administratör och ha tillgång till att skapa användare och nya webbplatser.

Ifall en webbplats kunde skapas läggs det till information om användaren i en separat MySQL-databas. Det finns även funktioner som kontrollerar t.ex. ifall namn på webbplatser eller användarnamn är upptagna. Detta för att slippa göra ett onödigt PHP/CURL-anrop. Det sista som sker vid applikationsservern är att den skickar tillbaka ett svar till webbapplikationen innehållande information om t.ex. en webbplats kunde skapas.

5 Resultat och diskussion

Resultat av examensarbete är en fungerande Wordpress-installation som tillåter att många webbplatser delar på en installation. Skapandet av webbplatser görs via en webbapplikation som finns vid Mitt Anvia. För att kunna kontrollera vem som har skapat webbplatser används en MySQL-databas innehållande information om kunden.

I skrivande stund väntas beslut om och när det nya systemet blir tillgängligt för kunder.

5.1 Vidareutveckling

Den första versionen är endast en grund för tjänsten. Det finns flera möjligheter till vad som skulle kunna implementeras eller möjligen förbättras i vidareutvecklade versioner.

I Wordpress-versionen kunde man implementera olika nivåer av användare för att kunna erbjuda olika teman och tillägg för användare. I den första versionen sparas användares filer i Wordpress-installationen. Det vore bra om man skulle få dem länkade till webbutrymmen som kunderna skapat, och på så sätt kunna slå ihop Wordpress-tjänsten och webbutrymme-tjänsten. Wpmudev erbjuder många tillägg för förbättring, men det krävs även tid till att konfigurera dessa tillägg. Man kunde även utveckla egna teman och tillägg som man erbjuder kunderna.

Man skulle även kunna se över om det finns bättre alternativ för att hantera skapandet och raderingar av webbplatser samt byte av användares lösenord än PHP/CURL, som för tillfället används.

5.2 Slutsatser

Detta har varit ett mångsidigt och intressant arbete, men utmanande. Eftersom det finns så många olika innehållshanteringssystem var det svårt att veta var man skulle börja för att hitta möjliga alternativ. Det som kanske var mest utmanande var att jag själv skulle välja

vilket system som tilltalar en stor grupp människor och som kan användas av i princip vem som helst. Eftersom jag inte hade någon erfarenhet av innehållshanteringssystem sedan tidigare, förutom att veta namnet på några få, har jag lärt mig mycket gällande olika innehållshanteringssystem. Mina kunskaper inom webbutveckling har också förbättrats mycket genom detta arbete.

6 Källförteckning

About GetSimple (2013)

<http://get-simple.info/start/> (Hämtat: 25.2.2013)

About Wordpress (u.å.)

<http://wordpress.org/about/> (Hämtat: 23.2.2013)

Burge, S. (2011) *Joomla! Explained: Your Step-by-Step Guide*. Boston: Pearson Education

Extension types (2011)

[http://docs.joomla.org/Extension_types_\(general_definitions\)](http://docs.joomla.org/Extension_types_(general_definitions)) (Hämtat: 27.2.2013)

Företagspresentation (2012)

<http://anvia.sharepoint.com> (Hämtat: 19.12.2012)

Gilmore, J. W. (2010) *Beginning PHP and MySQL: From Novice to Professional*. New Jersey: Wiley Publishing

Hauschildt, S. (2010) *CMS Made Simple 1.6: Beginner's Guide*. Birmingham: Packt Publishing

Khalid, M. I. (2006) *cURL PHP Book*

Leary, S. (2010) *Beginning Wordpress*. New York: Springer Science+Business Media

McFarland, D. J. (2009) *CSS: The Missing Manual*. Sebastopol: O'Reilly Media

Miller, C., Sabin-Wilson, L., Palmer, K., Rennick, A. & Torbert, M. (2011) *Wordpress All-in-One For Dummies*. New Jersey: Wiley Publishing

Opensourcecms (u.å.)

<http://www.opensourcecms.com> (Hämtat: 21.2.2013)

PHP just grows & grows (2013)

<http://news.netcraft.com/archives/2013/01/31/php-just-grows-grows.html>
(Hämtat: 18.2.2013)

Refsnes, H., Refsnes, S., & Refsnes, K. J. (2010) *Learn HTML and CSS with w3schools*. New Jersey: Wiley Publishing

What is PHP? (2013) <http://php.net/manual/en/intro-what-is.php> (Hämtat: 18.2.2013)

Wpmu Premium (u.å.)

<http://premium.wpmudev.org/join> (Hämtat: 26.2.2013)