



# **karisgossarna.fi**

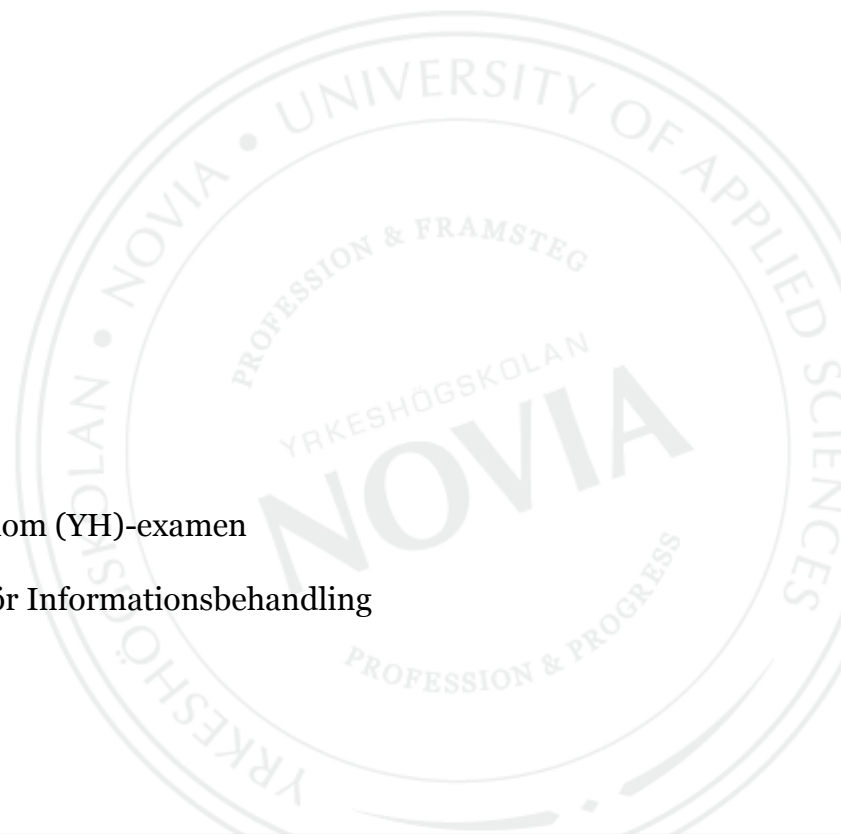
**En webbcommunity**

Tommy Söderlund

Examensarbete för Tradenom (YH)-examen

Utbildningsprogrammet för Informationsbehandling

Raseborg 2012



## Innehållsförteckning

Abstrakt.....	
Abstract.....	
1 Inledning .....	1
1.1 Teoretisk bakgrund .....	1
1.2 Omfattning .....	1
1.3 Syfte och målsättning .....	2
2 Karis Gossarna r.f. ....	3
3 Utvecklingsverktyg .....	4
3.1 WAMP.....	4
3.2 Annan programvara .....	5
4 CMS-System.....	6
4.1 Alternativ .....	6
4.2 CMS-Systemet Drupal.....	7
4.2.1 Drupal - möjligheter och utgångspunkt .....	7
5 Översikt av webbcommunityn .....	8
6 Extern Design.....	9
6.1 Header.....	9
6.2 Bakgrunden.....	10
6.3 Avatar för användare .....	10
6.4 Favikon .....	11
6.5 Väggen och den egna streamen .....	11
7 Intern Design.....	13
7.1 Filstrukturen.....	15
7.1.1 Root-katalogen.....	15
7.1.2 Modules .....	15
7.1.3 Includes.....	16
7.1.4 Themes.....	16
7.1.5 Sites .....	16
7.2 Databasen.....	17
7.3 Om webbcommunityns moduler.....	18
7.4 Användarrättigheter .....	21
8 Medlemsinformationen .....	22
8.1 Spara i databas .....	22
8.2 Cron.php och automatiska e-postfunktionen, skriva till fil .....	24
8.3 Vid användarregistreringen .....	27
9 Testprocessen .....	27
9.1 Buggar i sidan och lösningar .....	27
10 Problem och lösningar under utvecklingen.....	29
10.1 Påträffade i lokal miljö .....	29
10.2 Påträffade vid uppladdning och i Nettihotellis miljö.....	31
10.3 Övriga utvecklingsutmaningar.....	32
10.4 SPAM .....	32
11 Diskussion och resultat .....	33
Källförteckning .....	36

## Abstrakt

### EXAMENSARBETE

Författare: Tommy Söderlund

Utbildningsprogram och ort: Informationsbehandling, Raseborg

Handledare: Klaus Hansen

**Titel: karisgossarna.fi – En webbcommunity**

---

Datum 31.08.2012

Sidantal 37

Bilagor 0

---

#### Sammanfattning

Detta examensarbete är ett webbutvecklingsprojekt av en webbcommunity, enligt traditionell metod för webbutveckling. Arbetet gjordes som ett beställningsarbete åt scoutrörelsen Karis Gossarna r.f.. Denna webbcommunity är ämnad att dels förbättra kommunikationen medlemmar, ledare och styrelsen emellan och dels tillföra mer aktuell medlemsinformation i medlemsregistret genom att sporra medlemmarna att själva fylla i sin medlemsinformation på webbcommunityn. CMS-systemet som används som bas för webbcommunityn är Drupal 6.

Kraven på webbcommunityn var bla. sökfunktion, dokumentarkiv, fotogalleri, medlems- samt styrelseforum, profilsidor för användare, kommentarsmöjlighet på profilsidor, galleri för användare, "Wall", gästbok samt en chatt en månatlig epostrapportering. Ytterligare skall webbsidan hittas av Google samt på webben fungera som en informationskanal som Karis Gossarna själva kan uppdatera.

En del av de ursprungliga önskemålen lämnades bort i förverkligandet av olika orsaker. Resultatet av utvecklingen anser jag ändå blev lyckat, då all övrig önskad funktionalitet levererats.

---

Språk: Svenska

Nyckelord: Webbcommunity, community, Drupal, webbutveckling, Karis Gossarna

---

## **Abstract**

### **BACHELOR'S THESIS**

Author: Tommy Söderlund

Degree Programme: Business Information Technology

Supervisors: Klaus Hansen

**Title: karisgossarna.fi – A Web Community / karisgossarna.fi - En webbcommunity**

---

Date 31 August 2012

Number of pages 37

Appendices 0

---

#### **Summary**

This thesis is the web development of a web community, developed through traditional methods for web development. The thesis was done as an order to the scout movement Karis Gossarna r.f.. This web community is intended to improve the communication between members, leaders and the administration as well as enable the member register to be more up to date through encouraging members to update their member information themselves through the web community. Drupal is the CMS system that is used as a base for this Web Community.

The requirements and the functionality that is sought include: search function, document archive, photo gallery, member and administration forums, user profile pages supporting comments, user photo gallery, "Wall", guestbook, a chat and a monthly report sent by email. Additionally, the web page should be found by Google and on the web function as an information channel, which Karis Gossarna r.f. themselves can update.

Some of the original optional requirements were for various reasons not implemented in the realization of this web community. Personally, I feel that the overall result of the development was a success, since the rest of the functionality was delivered.

---

Language: Swedish

Key words: web community, community web page, community, web development, Drupal, Karis Gossarna

---

## **1 Inledning**

Detta lärdomsprov är ett beställningsarbete åt scoutrörelsen Karis Gossarna r.f.. I arbetet beskrivs bland annat själva utvecklingsprocessen, de problem som uppstod under arbetets gång, underliggande teknologier och system samt väsentliga konfigurationer och lösningar för den nya webbsidans funktionalitet.

Mitt huvudmål var att utveckla en webbsida som informationskanal för Karis Gossarna r.f. Efter diskussion med föreningens styrelse beslöts att en "Community" webbplattform vore lämpligt för deras behov. Med en webbcommunity förstås vanligen en mötesplats på nätet, alltså en webbsida där individer kan logga in i med ett eget användarkonto och kommunicera med andra användare via olika funktioner beroende på typen av webbcommunity. Användarna består oftast av individer som har ett eller flera gemensamma intressen, har gemensamma syften eller mål eller är likasinnade i övrigt eller individer tillhörande samma gemenskap, rörelse, förening eller organisation.

Denna webbcommunity är baserad på "Content Management Systemet" (härefter CMS-system) Drupal och fungerar även som en informationssida utåt. Med hjälp av webbsidan ville man förbättra kommunikationsmöjligheterna och samtidigt få ett fungerande system, där medlemmarna själva kan uppdatera sin medlemsinformation och skapa ett medlemsregister med mer korrekta och uppdaterade kontaktuppgifter.

### **1.1 Teoretisk bakgrund**

För att förstå detta lärdomsprov krävs en viss insikt i webbutveckling och webbsidor som exempelvis Facebook, Myspace, Diaspora eller Google+ eller åtminstone koncepten bakom liknande webbsidor. Ytterligare underlättar det att läsaren åtminstone bekantat sig med CMS-System och speciellt med Drupal 6.

### **1.2 Omfattning**

Existerande CMS-system under gpl-licens har den funktionalitet som projektbeställaren önskade i formen av tilläggbara moduler. Ett färdigt CMS-system

minskar på arbetsbördan och projektens egna community förser det med säkerhetsuppdateringar. En annan fördel är minskade skolningsbehov då bland annat guider finns färdigt tillgängliga på internet. Ytterligare funktionaliteter kan också skräddarsys för att uppnå förutbestämda krav samt önskemål av projektbeställaren.

Examensarbetet omfattar däremot inte produktion av konkret text för sidinnehållet av webbcommunityn, den ansvarar Karis Gossarna r.f. själva för, liksom för information och uppdateringar.

Under examenarbetets gång använde jag mig huvudsakligen av html, css, PHP och SQL. Sql och php framkommer speciellt i utvecklingen av den månatliga automatiska e-postrapporteringen till Karis Gossarna r.f:s styrelse. Css och html används för att få webbsidan att grafiskt se ut som önskat, utöver att förstå hur elementen är ihopsydda i Drupal.

### **1.3 Syfte och målsättning**

Krav, målsättningar och önskemål för den praktiska delen av detta examensarbete förhandlades fram mellan mig och projektbeställaren vid ett möte med Karis Gossarna r.f:s styrelse. Medlemmarna, dvs. huvudparten av de interna slutanvändarna, deltog inte i beslutsfattandet eller i testningsprocessen av webbsidans funktionalitet

Karis Gossarna r.f. hade som huvudmål att få en informationskanal på webben, med allmän information om rörelsen och dess historia, rörelsens kommande och utförda aktiviteter, kontaktuppgifter, grupper, bildgalleri från evenemang och ett skilt forum för användare och ett för styrelsen som endast de har rättigheter till. Styrelsen har moderator rättigheter till webbsidan och kan uppdatera informationen på webbsidan själva. Ett skilt dokumentarkiv var också önskvärt för uppladdning av mötesprotokoll etc.

Beställaren ville dessutom att webbsidan skall hittas av google och ha en egen intern sökfunktion. Webbsidan bör också ha profiler för användare där de kan skriva om sig själva och möjlighet för andra användare att skriva kommentarer till dem. Användare skall också kunna ladda upp egna foton till ett eget galleri.

Medlemmarna skall kunna skriva in sina medlemsuppgifter på webbsidan och endast själva se dem, men månatliga rapporter över ändringar i medlemsinformationen

skickas automatiskt till styrelsen per epost. Slutligen var det önskvärt att webbcommunityn skulle ha en chatt, en ”Wall” och en gästbok.

Genom att integrera populära element från sociala nätverk som exempelvis en facebookliknande "wall" och användarprofiler med egna "walls" eller gästboksfunktionalitet var tanken att stärka gemenskapen och underlätta kommunikationen mellan medlemmarna.

Mina visioner för det slutgiltiga resultatet av karisgossarna.fi var att ha utvecklat ett så funktionskraftigt system att det verkligen tas i bruk på allvar. Det skall ses som ett användbart verktyg och inte som någonting som medför mera arbete utan egentlig nytta.

Målet ställer förstås också vissa krav på styrelsen och scoutledare efter att webbsidan är klar. Styrelsen måste ha intresset att förbättra sitt eget system genom att skapa användarkonton åt sina medlemmar, ledarna bör också ha sporra sina patruller att aktivt delta i communityn samt att uppdatera sin medlemsinformation via detta verktyg.

## **2 Karis Gossarna r.f.**

Scoutföreningen Karis Gossarna r.f. grundades år 1911 under järnvägstjänstemannen Mauritz Sandbergs ledning och år 1978 blev Karis Gossarna r.f. en registrerad förening. Karis Gossarna r.f. (även kallat KG) är en scoutkår för pojkar i Karis, Raseborg, och har för närvarande 71 stycken registrerade medlemmar enligt uppgifter för år 2012. Karis Gossarna r.f. är medlem i både Finlands Svenska Scoutförbund r.f. och Finlands Svenska Scouter r.f. och dess ledning består av ett beslutsfattande samt verkställande organ kallat styrelsen. Styrelsen väljs årligen på kårens årsmöte. (Lokalarkivet arresten)

Föreningens syfte är att bedriva sin verksamhet i enlighet med scoutrörelsens anda. Gemenskapen är en viktig faktor i scoutingen, bland annat lärs team-building, första hjälp, knopar, användning av ”trangia kök”, tältuppsättning, flagghissning, flagghalning och andra praktiska saker för vildmarksliv ut till medlemmarna. Att sporra medlemmarna att bli en ansvarskännande och självständiga individer i samhället kan nämnas som ett av huvudmålen.

## 3 Utvecklingsverktyg

Jag valde att utveckla webbsidan för Karis Gossarna i en lokal utvecklingsmiljö, på detta sätt undviker man långa uppladdningsprocedurer för små ändringar i webbsidan och uppnår därmed en effektivare arbetsprestanda. Efter att testningsprocessen inletts fortsatte utvecklingen på den slutgiltiga servern.

### 3.1 WAMP

Programvara under klassifikationen WAMP möjliggör användning av Apache, MySQL och PHP på Windows-baserade operativsystem, förkortningen kommer möjligen från just dessa teknologier, då det exempelvis för Linux-baserade system kallas LAMP. För utvecklingen av Karisgossarna.fi testade jag två olika WAMP programvara. Det ena var WampServer som jag under kurserna i PHP i Novia har ansett att varit bättre, bland annat på grund av specifika felmeddelanden och enklare konfigurationsmöjligheter. P.g.a. kompatibilitetsproblem använde jag dock slutligen ett annat program, under GPL licens, vid namnet MoWeS Portable II.

En fördel med MoWes Portable II framom WampServer är att det är "portabelt", dvs. det går att installera på ett USB-minne. Fast denna portabla fördel visade sig sist och slutligen vara mer en nackdel. Jag lade märke till att laddningar av sidan ofta var väldigt långsamma då jag använde MoWeS Portable II installerat på ett USB-minne. Det berodde på att filöverföringen blev långsam pga. det enorma filantalet, samt den minimala filstorleken, som skall flyttas emellan medierna. Jag blev tvungen att överföra allting till ett annat medium för att det inte skulle överskrida vissa "timeout"-begränsningar. Dessa tidsbegränsningar var uppenbara speciellt då databasen skulle uppdateras via 'update.php' och då man väljer vilka moduler som skall användas på webbsidan i Drupals administrativa system.

Teknologierna jag valde att installera med MoWeS Portable II var Apache2, ImageMagick, MySQL5 och PHP5. ImageMagick krävdes för att få mitt galleri att fungera korrekt. Noterbart här är också att SE betyder "Small Edition", inte "Secondary Edition" som jag felaktigt tolkade det som. Detta ledde i sin tur till att jag i min ovisshet valde SE versionerna som ledde till kompatibilitetsproblem.



Ett annat problem var filstorleken vid uppladdning av databasen till phpmyadmin då jag skulle återställa en backup. Drupal's databas hade blivit för stor, detta berodde möjligtvis på att cron inte var konfigurerat att köra automatiskt ännu i det skedet. Problemet löste sig då jag i PhpMyadmins dokumentation under rubriken "1.16 I cannot upload big dump files (Memory, http or timeout problems)" hittade "The first things to check (or ask your host provider to check) are the values of *upload\_max\_filesize*, *memory\_limit* and *post\_max\_size* in the *php.ini* configuration file". Dessa var i sig alla större än databasfilen, men efter att jag läst på samma webbsida "One user also said that *post\_max\_size* and *memory\_limit* need to be larger than *upload\_max\_filesize*." och ändrat värdena samt startat om MoWeS Portable II fick jag allting att fungera korrekt igen. (PhpMyAdmin 2012)

### 3.2 Annan programvara

Dreamweaver är editeringsverktyget jag valde att utnyttja för webbutvecklingen. Dreamweaver har stöd för att upprätthålla flera versioner av webbsidan, så kallad "version control", som jag dessvärre aldrig använde i detta projekt. Ytterligare har Dreamweaver ett designläge, men tyvärr fungerar inte PHP-funktioner i det, vilket betyder att det i stort sett är oanvändbart då det kommer till sidor utvecklade med Drupal som bas.

Utöver Dreamweaver användes "Firebug"- och "Web Developer"-tilläggen till Firefox under utvecklingen av webbsidan. Med hjälp av dessa verktyg får man exempelvis lätt reda på vad olika elements klasser heter, vilka .css-filer som används på sidan som är öppen. En annan bra fördel med Firebug är att det går att göra ändringar i realtid på webbsidan, jag använde dock denna funktion endast tidvis för att försäkra mig om att .css-koden jag skrev fungerade. Åtminstone med senare versioner av Firefox har dock dessa två tillägg börjat slöa ner webbläsaren vilket jag upplever mig åtminstone tidvis ha känt av. (MozillaZine)

Adobe Photoshop har använts vid ett flertal tillfällen till den externa designen och vid enstaka fall använde jag också Microsoft Paint och Notepad. FileZilla som FTP-klient och Putty för SSH-anslutningar till Nettihotelli som Karis Gossarna r.f. valt som leverantör. CMS-systemet Drupal används också som underliggande plattform för webbcommunityn.

## 4 CMS-System

### 4.1 Alternativ

Utöver CMS-systemen Drupal 6 och Drupal 7 begrundade jag att använda wordpress, Joomla och CMS Made Simple. Jag såg också på ett par specifika Community-paket som inte var CMS-System i sig, så som exempelvis bitweaver och Elgg, men kunde snabbt konstatera att de inte var de bästa alternativen för att fylla kraven för webbcommunityn. De integrerade utvecklingsmöjligheterna var inte på långt när tillräckliga. Den enda synbara möjligheten för att utöka funktionaliteten var att tillägga olika 'appar' på olika sidor. Den inbyggda kontrollen blir på detta sätt helt enkelt för liten.

Wordpress är egentligen avsedd för att skapa bloggsidor med, modulutbudet är inte lika stort som de andra alternativen till denna form av webbsida och möjligheterna till utveckling med detta system ansåg jag var ganska begränsade. På plussidan för detta CMS-System var att jag använt mig av det förut, och det är lätt att förstå sig på enkel användning av det. Det vore således ett bra alternativ för slutanvändaren.

BuddyPress skulle möjligen ha kunnat vara ett lätt och bra alternativ för åtminstone intranätet av Karisgossarna.fi, men då jag stötte på detta alternativ var utvecklingen av Karisgossarna.fi i slutskedet.

Joomla har jag tidigare inte använt mig av desto mera, förutom Moodle, sidorna som Novia och Åbo Akademi använder sig av. Sidor utvecklade med Joomla har editeringsknappar i mitten av sidan, åtminstone då man är inloggad, vilket jag personligen inte anser är estetiska. Det Intryck som användningen av Moodle har lämnat mig är inte positiv. Dessutom finns det aningen mer moduler till Drupal överlag och mer moduler som hanterar den funktionalitet som Karisgossarna.fi önskat sig. Dessa faktorer var avgörande i valet.

CMS Made Simple (härefter CMSMS) fungerar, till skillnad från åtminstone Drupal, mest med .php- och .css-filer. Drupal har en massa .inc-, .module- och .info-filer som är tidskrävande innan man satt sig in i dem. Det som slutligen fällde nådestöten för CMSMS var modulutbudet för CMSMS. Det verkade inte finnas många communityrelaterade moduler som passade till Karis Gossarna r.f:s ändamål. Åtminstone hittade jag inte sådana.

## 4.2 CMS-Systemet Drupal

Efter förundersökningen kom jag slutligen fram till att Drupal i detta fall är det bästa alternativet. Mycket av funktionaliteten finns färdigt i form av moduler, knepigt var dock att veta precis vilka moduler som har de funktionaliteter som behövs samt att få dem integrerade i sidan så att de fungerar korrekt tillsammans.

Drupal 6 är ett "Content Management System" och enligt utvecklarna själva även ett "Content Management Framework" (CMF). Skillnaden mellan de flesta CMS-system och ett CMF-system är som de ganska elegant uttrycker det på drupal.org (fritt översatt): De flesta CMS-system är som en leksaksbil, många antaganden har gjorts om dess användning, antaganden som vore svåra att överskrida. CMF-System förser dig endast med råmaterial - du behöver kunna programmera och ha en klar vision av designen. Drupal är som en legosats. Begåvade utvecklare har redan gjort byggnadsblocken - i formen av samarbetade moduler - som du behöver för att bygga webbsidan som täcker dina behov, om det så är frågan om en nyhetssida, en onlinebutik, ett socialt nätverk, blogg, wikisida, eller någonting helt annat. (Hunter 2012)

Med andra ord är Drupal mer flexibelt än konventionella CMS-system. Ändå känns det ibland som om man bygger lego-satsen helt utan ritning, ifall utvecklarna av modulen inte gjort åtminstone någon offentlig dokumentation, vilket flera gånger är fallet. Drupal 6 distribueras under GNU General Public License (GPL), version 2 från juni 1991. (Drupal)

Drupal har många färdigutvecklade moduler som kan tillämpas så att de uppfyller kraven för detta examensarbete. Det finns inbyggd sökfunktion och forum, användarprofiler fås genom en extra modul, likaså fås bildgalleri och en chattfunktion. En facebook-liknande vägg fås via en modul och gästböcker finns också som en modul. Ytterligare finns det editeringsmoduler som gör det nästan lika lätt för användare att uppdatera sidoinnehåll som det är att skriva ett dokument i Microsoft Word eller OpenOffice.

### 4.2.1 Drupal - möjligheter och utgångspunkt

En stor nackdel som Drupal 6 har är att administrationssystemet är ganska överskådligt och till en början kaotiskt. Det är inte det lättaste av alternativen för

slutanvändaren att sätta sig in i. Valet föll slutligen ändå på Drupal 6 för att det hade flest moduler färdigutvecklade som kunde användas för sidans ändamål och därmed säkrast kunde leverera all önskad funktionalitet. Själv hade jag använt systemet tidigare vid ett tillfälle och var åtminstone delvis bekant med det.

Drupal 7 var endast i alfa-stadiet då jag började utvecklingen av sidan och hade inte samma modulutbud. I det skedet följde jag även communityns rekommendationer. (Gluskin 2012) I framtiden kommer modulutbudet troligen att ändra och då kanske man kan göra en migrering till Drupal 7 eller till och med Drupal 8, som nyligen släppts ut i alfaversión, fast då är det risk att ett flertal funktioner kommer att sluta fungera.

## 5 Översikt av webbcommunityn

Som oinloggad användare på webbsidan som jag skapat har man tillgång till vanligt sidinnehåll, sökfunktionen samt kan läsa och kommentera i gästboken för hela hemsidan. Via de olika sidorna är det tänkt att Karis Gossarna r.f. skall ge besökarna allmän information om verksamheten och via gästboken kunna ta emot kommentarer, åsikter och intryck från besökare.

Efter att ha loggat in i webbcommunityn kommer man också åt sin egen profilsida och kan ändra på innehållet där. Profilsidan är tänkt att stärka gemenskapen då medlemmar där kan skriva om sig själva så andra medlemmar kan lära känna dem bättre. Profilsidan har också en ”avatar”, så om användaren har en bild på sig som ”avatar” har andra användare lättare att koppla ihop alla namn med ansikten.

Via profilsidan kan medlemmarna också kommunicera med varandra antingen genom varandras gästböcker eller via varandras ”streamar”, dvs. kommenterings- och diskussionsfunktionen på användarprofilerna. Styrelsemedlemmarna kan också skicka kommentarer till alla användares streamar på samma gång ifall de så önskar. Medlemmarna har också möjligheten att chatta med varandra om korrespondensen annars blir för utdragen eller om medlemmar annars så önskar.

Ett annat centralt kommunikationsmedel på webbcommunityn är den allmänna ”väggen” för inloggade användare. Denna vägg följer samma princip som användarnas egna ”streamar”. Här kommunicerar man med hela communityn på en

gång, väggen tillåter delvis html-kod så det går också att ”embedda” videon från exempelvis YouTube om så önskas.

Forumet är ämnat för att få ett strukturerat medium som innehåller nyttig information till medlemmarna och kommunikation som permanent hålls kvar så länge ingen tar bort innehållet. Styrelseforumet är till för att styrelsen privat från medlemmarna skall kunna kommunicera med varandra och exempelvis planera olika evenemang som skall komma som överraskningar för medlemmarna.

Det ”officiella” fotogalleriet, även kallat Fotoarkivet, är tänkt att innehålla bra bilder från olika evenemang som rörelsen varit delaktig i, detta galleri underhålls av styrelsen. Medlemmarna kan också skapa egna gallerier som de kan ladda upp bilder till. Dokumentarkivet är tänkt att innehålla eventuellt protokoll från möten och andra dokument som styrelsen tycker är av central betydelse för rörelsens verksamhet.

## **6 Extern Design**

Den grafiska designen för webbcommunityn skapade jag på basen av det inbyggda temat ”Garland” i Drupal. Jag gjorde bedömningen att det var mer lämpligt att editera ett redan befintligt tema än att bygga upp ett eget från grunden och ansåg att Garland hade den mest önskvärda uppbyggnaden av de medföljande teman. Nedan följer en snabb genomgång på grafiskt producerat material samt det jag anser att var de största designmässiga utmaningarna jag stötte på under utvecklingen av webbsidan.

### **6.1 Header**

Jag gjorde en temporär ”Logo” i form av en panoramabild som skall bytas ut mot ett panoramafoto från Karis Gossarnas ”scoutstuga” efter att någon varit där och fotograferat. Panoramabilden innehåller både själva logon och dessutom panoramabilden bakom. Detta innebär förenklad kontroll över logons position i panoramabilden. För att få kontroll på exakt var på sidan logon skulle visas behövde jag få in bilden i en div-box. Detta gjordes genom att skapa ett block som syns på alla sidor som innehåller div-boxen och bilden för logon. Den temporära logon kan ses i figur 2.

För att få Headern att centreras i Internet Explorer och inte komma på fel plats i Firefox genom ändringen var jag tvungen att lägga till "text-align: center;". Ett annat alternativ hade varit att använda Javascript och Ajax, men jag ansåg att det var lite väl komplicerat endast för detta ändamål och fann ett mindre komplicerat alternativ.

Headern tycks av någon anledning komma på fel plats i webbsidan då man återställer backup, troligtvis har det att göra med filrättigheter under "sites"-katalogen som skyddar vissa filer från överskrift. Lösningen är i dessa fall att ta bort hela "sites"-katalogen helt och sätta in den på nytt från backupen.

## 6.2 Bakgrunden

Webbsidans bakgrund gjorde jag till en början enligt Karis Gossarnas r.f:s färger, dvs. gul (#FFF003) och blå (#1D1B8E), men efter diskussion och samarbete med en styrelsemedlem gjorde jag om den till blå (#1D1B8E), och vit (#FFFFFF). Jag gjorde den med en gradient i Photoshop, men stötte på en designutmaning. Alternativen jag först såg var att antingen göra en väldigt lång bild (vertikalt) som blir till en färg längre neråt eller så blir sidan vit då bakgrundsbilden slutar i något skede eller så hoppar färgen tillbaka mitt i allt från gul till blå då bakgrundsbilden tog slut. Jag tyckte inte att något av alternativen var särskilt attraktivt. Till slut kom jag på att om jag gör gradienten från blå till gul till blå så slipper jag problemet om bakgrunden upprepas vertikalt, eftersom bakgrunden kan börja på nytt igen och har exakt samma färg som där den slutade. Grafiskt märks det alltså inte när bilden tar slut. Denna bakgrund syns delvis i figur 2, den fortsätter neråt på sidan och sidorna är bortklippta då sidinnehållet är centrerat på skärmen.

## 6.3 Avatar för användare

Jag skapade också en standard-"avatar", dvs. en bild eller ett foto som representerar en användare på profilsidan. Bredvid den finns forumposter eller kommentarer. Den kommer att användas av alla nya medlemmar innan den ändras till en annan bild. Denna avatar syns för användaren test i figur 2. Den är endast avsedd för internt bruk inne på sidan då den är gjord på basen av World Organisation Of The Scout Movements logo, jag ändrade endast färgerna till Karis Gossarna r.f:s färger med en grön bakgrund. Ifall den skall användas någon annanstans bör Creative Commons Public License (ccpl)-licensen följas. (Creative Commons 2011, 2012).

## 6.4 Favikon

Jag skapade en favikon utgående från Karis Gossarna r.f:s logo. Tydligt måste man ladda ner plugin till Adobe Photoshop för att kunna spara i ".ico"-format, detta behövdes för att få favikonen att fungera i Internet Explorer. Det fungerade i Firefox med att bara byta namnet från .png till ".ico" och också till och med att använda en ".png"-bild som favicon, det kan göras via Drupal. Favikoner som fungerar i Internet Explorer kunde dock endast vara 16x16 pixel. Efter att ha ändrat på formatet från 64x73 pixel till 16x16 pixel och sparat favikonen i korrekt formaterad ".ico"-format började det även fungera i Internet explorer. (RealWorld Graphics 2009)

## 6.5 Väggen och den egna streamen

Väggen är motsvarigheten till Facebooks "Wall", på vissa webbcommunityn även kallad "shout". Den egna streamen är motsvarigheten till Facebooks kommentarsmöjligheter på användares profilsidor, den egna streamen kan delvis ses i Figur 2 men restrerande kommentarer följer nedanför det som syns i screenshoten. Views ställde till mycket problem innan jag förstod mig ordentligt på modulen. Det var speciellt problem med Facebook-style Statuses (Microblog). Jag fick inte elementen rätt placerade med hjälp av rätt placerade inbyggda <span>- och <div>-klasser. Egna "custom text" <div>-klasser placerade på rätt platser i Views verkade alltså inte fungera. Slutligen löste jag det genom att visa alla elementen genom samma "custom text" och kalla på elementen som "substitution patterns". Nedanför följer den slutgiltiga "custom text"-koden för "Vägg" i kodexempel 1, streamen för alla status-uppdateringar, med "substitution patterns" i Views.

*Kodexempel 1. "Custom text" modifiering för "vägg"*

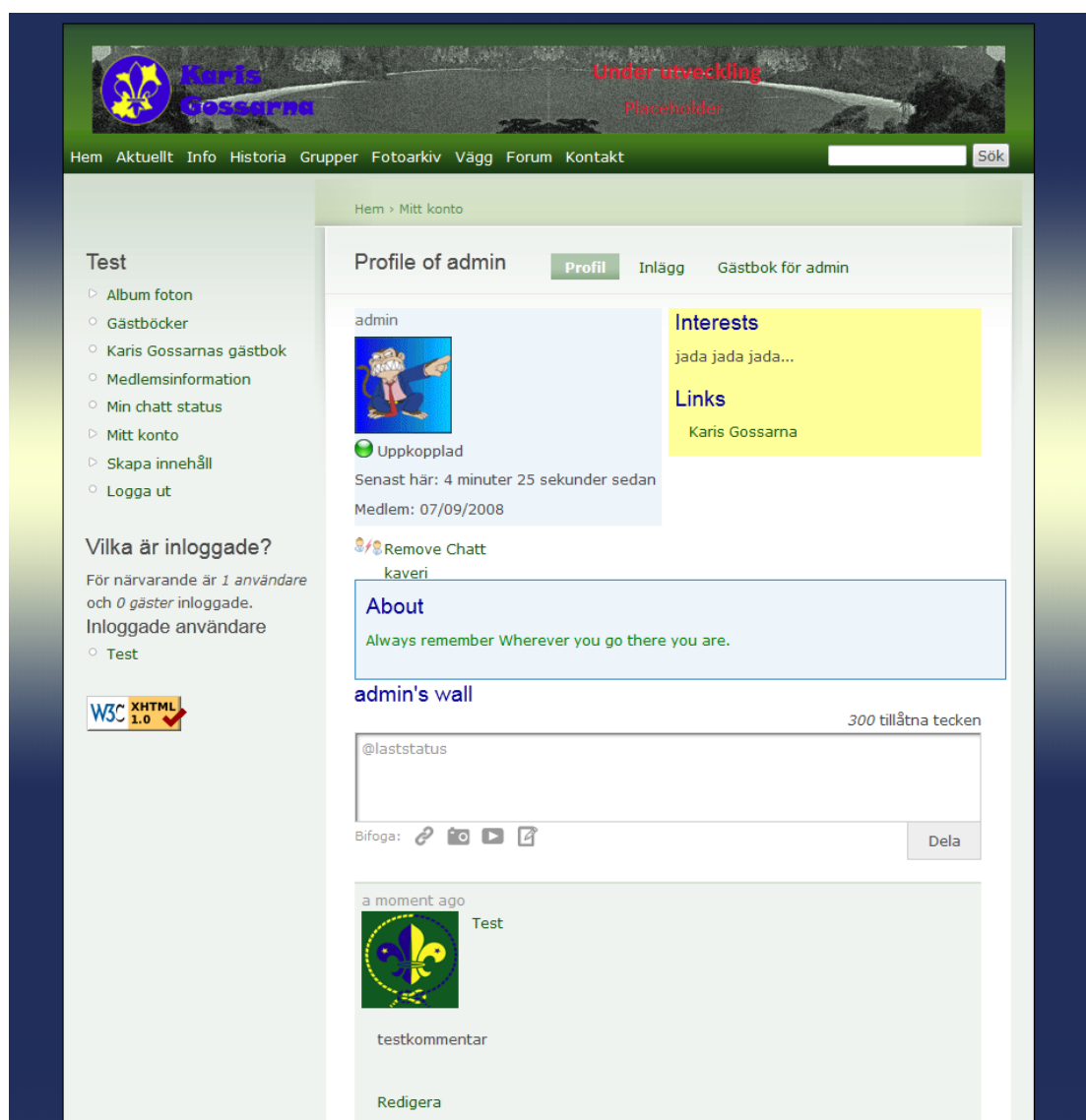
```
<div class="fbss-uinfo1">
<div class="fbss-created1">[created]</div>
<div class="fbss-picture1">[picture] </div>
<div class="fbss-name1">[name] </div>
</div>
<div>
<div class="facebook-status-details">[message] [attachment] [edit] [delete]</div>
<div class="fbss-comments1">[comment-box]</div>
</div>
```

För att få Facebook-style Statuses (Microblog) comments att se ut som vanliga statusar var jag tvungen att ändra i fbss\_comments.module, detta på grund av

omarrangering av \$output variabeln till egna "div-boxar" för att få en sammanhängande och logisk sidupbyggnad som ser bra ut. Denna ändring av modulens kod krävdes på grund av att comments submodulen till Facebook-style Statuses (Microblog) inte har en egen View i Views. Detta innebar att ifall uppdateringar görs till själva Facebook-style Statuses (Microblog)-modulen skulle själva uppbyggnaden av "Väggen" ändras och se "ful" ut. Modulen för comments är numera en undermodul inkluderad i själva Facebook-style Statuses (Microblog)-modulen. Denna ändring av modulens kod krävdes på grund av att comments submodulen till Facebook-style Statuses (Microblog) inte har en egen View i Views. Tyvärr glömde jag bort detta faktum i något skede och installerade en väsentlig säkerhetsuppdatering av modulen utan att lägga märke till designändringen eller spara backup av ändringarna. Men i och med uppdateringen ser statusuppdateringarna ändå någorlunda bra ut på grund av modulens egna ändringar.

Slutledningsvis bör nämnas att jag även integrerade Facebook-style Statuses Micropublisher i webbsidan. Detta innebär att man enkelt kan lägga till länkar, foton dokument eller vilka filer som helst till sina kommentarer. Länkarna hämtar information till statusen automatiskt från sidan man länkat till som exempelvis metadata för beskrivningen av sidan samt logon. Jag tillät även många html-taggar, exempelvis kan man embedda YouTube-videon i sina statusar bara man kopierar embed-koden från youtube-videon. Jag skrev även en guide angående detta på forumet.

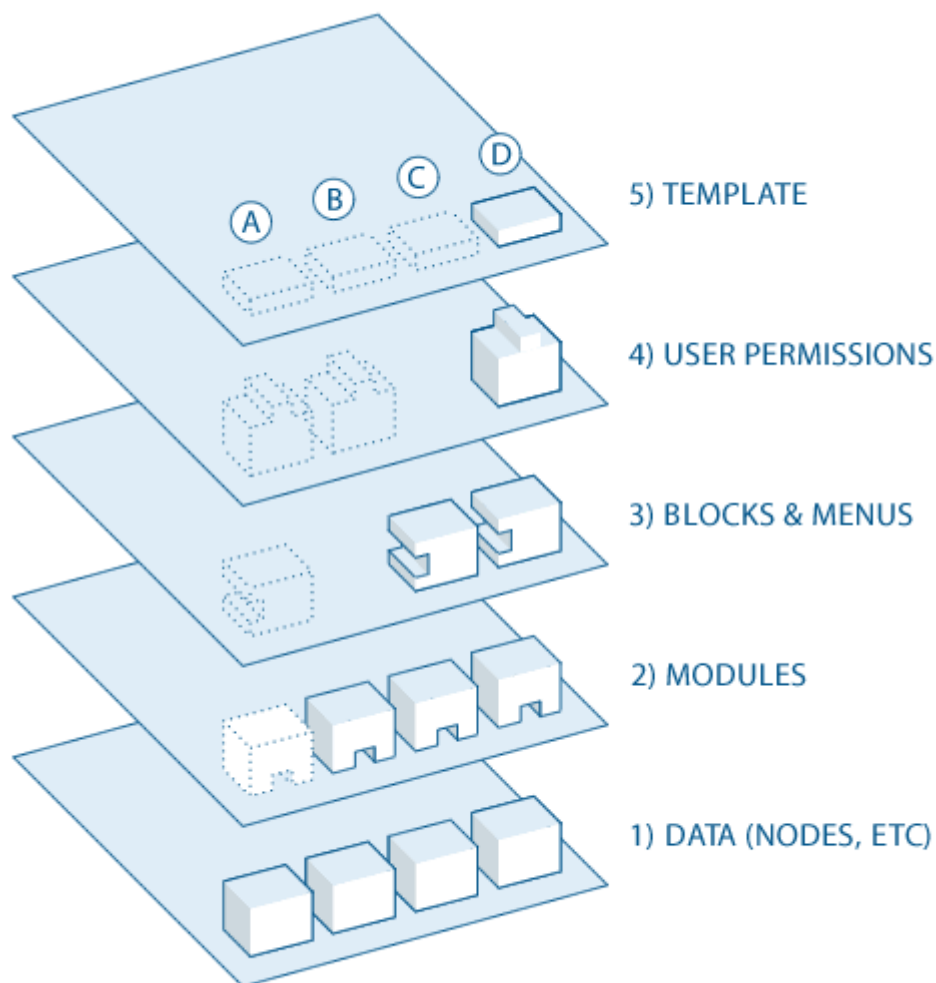




Figur 2. Screenshot av admins profilsida på webbcommunityn

## 7 Intern Design

För att få en klar bild av hur Drupal's interna design och därmed också Karisgossarna.fi:s interna design är uppbyggd och fungerar tillsammans så bör vi ta en titt på de olika nivåerna i Drupal's struktur. Figur 2 kan användas som ett visuellt hjälpmedel till sammanfattningen av Drupal's system.



Figur 3. En visuell skildring av de olika nivåerna i Drupal's interna struktur, Drupal (2008)

[https://drupal.org/files/drupal\\_flow\\_0.gif](https://drupal.org/files/drupal_flow_0.gif) (hämtad 12.3.2011)

I den allra lägsta nivån ligger databasen, här sparas alla data som sparas på sidan. Efter denna nivå följer som syns i figur 2 en nivå som hämtar data från databasen, som noder exempelvis.

Efter datanivån kommer modulnivån. Dessa moduler levereras antingen med Drupal, eller så kan man ladda ner dem från Drupal's community. Communityn innehåller en massa moduler som antingen frivilliga individer eller företag har utvecklat. Moduler är utökningar av Drupal's funktionalitet, vilka gör det möjligt att skräddarsy webbsidan till den sort av webbsida man försöker utveckla, tex. en webbutik eller sociala nätverk.

Nästa nivå är block och menyer. Block kan skräddarsys i hög grad, ofta görs detta genom att hämta output från moduler. Man kan välja var de ska visas på webbsidan och på vilka sidor de skall visas, eller på vilka de inte skall visas, ifall det är lättare så.

Härefter följer nivån för användarrättigheter. Denna nivå definierar vilka delar av olika moduler som användare har rätt att se och ändra innehållet på. Rättigheterna definieras av olika användarroller som användare indelas i.

På den sista nivån av Drupal kommer temat, den externa designen av sidan. Denna del definierar vad som visas var och hur. Som figur 2 även skildrar (under punkt 5, template där bokstäverna står för de olika teman) så är endast ett tema i användning åt gången. (Hunter 2012)

## **7.1 Filstrukturen**

Här följer en kort beskrivning av kataloger noterbara från webbutvecklarens synvinkel i Drupal och därmed också i Karisgossarna.fi:s katalogstruktur.

### **7.1.1 Root-katalogen**

I den så kallade 'root'-katalogen, som från början heter drupal, finns .htaccess, index.php, cron.php, update.php samt install.php. Filen .htaccess är en konfigurationsfil som möjliggör olika konfigurationer för olika kataloger. Denna konfigurationsfil gäller för katalogen den är i, men också för alla underliggande kataloger i katalogstrukturen som inte innehåller en egen sådan konfigurationsfil. cron.php är ett skript som kör cron, schemalagda uppgifter i Drupal. Update.php uppdaterar databasen efter vilka moduler som är aktiverade och bör alltid köras efter att man har tagit i bruk en ny modul. Install.php används vid Drupals installation och bör raderas efter att webbsidan är klart installerad, detta görs av säkerhetsskäl.

### **7.1.2 Modules**

I katalogen "Modules" direkt under rotkatalogen bör endast "core-moduler" finnas, dvs. de som följer med installationen av Drupal. Nya moduler placeras i stället under katalogen "sites". Den mest nämnvärda katalogen i "modules" under rotkatalogen är "system". Här finns en del av de väsentligaste så kallade "core"-filer som exempelvis

"system.admin.inc". Här finns också "page.tpl.php", men det är temats egen "page.tpl.php" som är i användning av sidan.

Det är god praxis att inte modifiera dessa filer så långt det går och i stället lösa problemen på något annat vis. Ifall något skall uppdateras kan annars något sluta fungera eller ändringar försvinna. Det kunde också vara en säkerhetsrisk att modifiera dessa filer.

### **7.1.3 Includes**

I katalogen "includes" finns inkluderade filer som används av Drupal, jag har dock inte behövt bekanta mig desto mer med dem under detta utvecklingsprojekt.

### **7.1.4 Themes**

I katalogen "themes" finns alla teman man har för webbsidan. "Karisgossarna.fi" är en tämligen omfattande modifierad version av temat "Garland". Nämnvärt i denna katalog är "garland.info", denna fil innehåller information om temat och definierar vilka .css-filer som skall användas. Page.tpl.php är en annan väsentlig fil som definierar sidinnehåll, node.tpl.php definierar innehållet för noder, block.tpl.php definierar innehållet för blocks, comment.tpl.php definierar innehållet för kommentarer och user-profile.tpl.php definierar innehållet för användarprofiler. Temat ändras genom Drupal administrationsverktyg på webbsidan där det också går att göra vissa mindre temaändringar.

### **7.1.5 Sites**

Under katalogen "sites" och "default" finns "settings.php" som innehåller inställningar för sidan. Under "files" och "color" finns en katalog för temat, som för närvarande är i användning i formen av exempelvis "garland-f3574734". I katalogen finns style.css och andra väsentliga filer för den externa designen av webbsidan. Anmärkningsvärt för detta område är att katalogen är en tillfälligt genererad katalog av Drupal, ifall temaändringar görs bör de även göras i katalogen "themes" och "garland". Ifall man konfigurerar temat på nytt med Drupals integrerade verktyg och sparar temat kommer Drupal att göra en ny temakatalog som tas i användning. Denna katalog kommer att sakna filer som är väsentliga för den externa designen. Det är alltid kritiskt att ha en backup av dessa kataloger. Man kunde också ha flyttat de inbyggda temana till en

egen temakatalog under "sites" och "all" och "themes" genom vissa modifikationer i temafilerna vilket vore bättre vid uppdateringar av Drupal, men detta lärde jag mig först i efterskott.

I katalogen "modules" under "sites" och "all" placeras alla nya moduler man laddat ner från Drupals community och vill utnyttja på webbsidan i fråga. Efter att ha packat upp modulen bör en administratör aktivera den i Drupal innan den tas i användning av webbsidan.

I katalogen "libraries" under "sites" och "all" finns filer som hör till modulen "bad behaviour", konfigurerat enligt readme-anvisningar som följer med modulen. För mer information angående "bad behaviour" se rubrik 13.4. SPAM.

I katalogen "photos" under "sites" och "default files" laddar gallerimodulen upp foton i originalstorlek i en skild katalog för bildgalleriet i fråga. Dessutom skapas en "thumb"-katalog inne i galleriets katalog med ett nummer efter i namnet där modulen har "cropt" och skalat bilden till en thumbnail i storleken 640x480 pixel, 240x180 pixel samt en thumbnail i storleken 100x75 pixel. Detta styrs av modulen "photos" och storleken och andra inställningar för detta kan konfigureras efter behag i Drupals administrationsinställningar.

I katalogen "temp" under "sites", "default files" laddas filer temporärt upp och i katalogen "pictures" under "sites", "default files" laddas resterande bilder upp, exempelvis användares avatar.

## 7.2 Databasen

Alla data som sparas och hämtas på sidan sparas i en SQL-databas. Denna har innehåll som levereras med Drupal, tilläggsmoduler samt input på webbsidan som sparas. Då man aktiverar en ny modul är det väsentligt att köra update.php för att databasen skall uppdateras med fält etc. som behövs av de nya modulerna.

Det väsentligaste att nämna då det gäller databasen, är tabellen för medlemsinfo. Denna tabell skapade jag med fälten för användar-id, födelsedatum, hemadress, hemtelefon, mobiltelefon, epostadress, facebook, google+ och tidpunkten då informationen sparats i databasen. Alla fält utom det för användar-id är konfigurerade att tillåta värden för varchar med olika längder, beroende på hur långa jag ansåg att

värdena maximalt kunde tänkas bli, fältet för användar-id tillåter integer-värden. Denna tabell innehåller medlemsinfon som användare av sidan matar in i olika fält efter att de klickat på "Medlemsinformation"-länken. Delar av funktionen som sköter om att inskrivna data sätts in i databasen kan ses i Kodexempel 2. Mer om denna funktion kan läsas under rubrik "11.1 Spara i databas".

"Medlemsinformation"-länken är en länk till en pageroute som hjälper användaren igenom några sidor där information som medlemsinformationen, vad som står på användarens profilsida samt användarkontots inställningar, som exempelvis användarnamn lösenord samt avatar konfigureras.

### **7.3 Om webbcommunityns moduler**

De väsentligaste modulerna som används för att utveckla en community-baserad webbsida med Drupal som bas är Advanced Profile Kit, Facebook-style Statuses (Microblog), Guestbook och Advanced forum. Resterande moduler används för att få allting att fungera så nära det sättet man vill ha det på som möjligt. Nedan följer en kort skildring av vad de olika modulerna används till för webbcommunityns funktion.

Utöver de moduler som skildras i underrubrikerna till detta stycke finns det ytterligare ett flertal som är grundval för en installation av Drupal, inte spelar någon större roll för sidans funktion eller är undermoduler för någon av de andra modulerna. Deras funktioner finns till för att få sidan rätt konfigurerad eller är krav för att någon annan modul skall fungera. Dessa nämns inte i detta sammanhang då deras antal är rätt stort.

#### **7.3.1.1 Användarprofilen**

På Karisgossarna.fi används Advanced profile kit och Content Profile så att alla användare skall få en egen användarprofil-sida.

#### **7.3.1.2 Användarprofilers design**

Chaos tools (ctools) används av Advanced profile kit som ett utmärkt verktyg för att ändra på layouten och innehållet på medlemmarnas profilsidor till önskat utseende. Profilsidan är till stor del uppbyggd enligt Advanced profile kits-dokumentation (Cox 2012), profilsidornas uppbyggnad görs genom att administrera den sidans uppbyggnad. Chaos tools var även ett krav för några andra moduler.

### **7.3.1.3 Väggen och den egna streamen**

Facebook-style Statuses (Microblog) och undermodulen comments används till att få en facebookliknande vägg eller "Wall" där man skriver statusuppdateringar. Den gör också så att användare får en egen "stream" på sin profilsida, där användare kan kommunicera på liknande sätt som i Facebook.

### **7.3.1.4 Väggens, den egna streamen och profilsidors design**

Views använde jag till att konfigurera profilsidorna, väggen med statusuppdateringar och den egna streamen på användarprofilen så som jag tyckte att de såg bäst ut med html, css och "substitution patterns" som exempelvis [message].

### **7.3.1.5 Gästböcker**

Guestbook-modulen används för att skapa en gästbok för hela webbsidan där även oinloggade besökare kan skriva vad de vill. Med Guestbook-modulen får alla användare också en egen gästbok på sin profilsida där inloggade användare kan skriva någonting. Detta innebär att det finns två sätt att kommunicera med enskilda användare förutom chatten, så användarna kan välja vilket sätt de tycker bättre om. Guestbook-modulen gör också så att man får en länk till en lista över alla användare till vänster i navigationsmenyn, vilket också är en bra funktion.

### **7.3.1.6 Forumet**

Advanced forum används till att få ett mera dynamiskt forum och forum access behövdes då jag ville skapa ett av forumen öppet för inloggade användare och ett annat forum endast öppet för styrelsemedlemmar. Modulen Forum som hör till Drupals kärna används som bas för Advanced forum.

### **7.3.1.7 Forumets design**

Author Pane används för att hämta information om användaren som skrivit en post i forumet och visar därefter infon om användaren, dvs. användarnamn, "avatar", datum och tidpunkt för posten, onlinestatus och när användaren blivit medlem på sidan i en ruta vänster om själva texten.

### **7.3.1.8 Åtkomsträttigheter**

Access control list (ACL) används i samband med Simple Access och Content Permissions, en undermodul till Content Construction Kit (CCK), för att göra det

lättare att kontrollera vilka sidor och noder som skall synas för och/eller kommas åt av gäster, inloggade användare samt styrelsemedlemmar.

#### **7.3.1.9 Chatten**

QTChat används för att få ett realtids chatsystem, QTChat använder User Relationships-modulen för att räkna vilka "vänner" som är online och begränsa chatten till dem. För att kunna chatta med en annan användare måste den ena av dem, på den andra användarens användarprofil, lägga till den som sin "chatt kaveri" Chatten sker alltså från person till person och är inget chattrum, för det ändamålet finns forumet. QTChat .module-filen har jag gjort några små ändringar i så att det skulle motsvara beställarens krav, ifall modulen uppdateras kommer dessa ändringar att försvinna.

#### **7.3.1.10 Medlemsinfo**

Pageroute och Profile Setup används av Karisgossarna.fi för att hjälpa användarna på traven i att fylla i medlemsinformationen genom att göra processen mera användarvänlig.

#### **7.3.1.11 Galleri**

Album Photos används till bildarkivet och photos swfu (även kallat SWFUpload) används till att göra uppladdningen av bilderna lättare och mer användarvänlig exempelvis genom att göra det möjligt att välja många foton på samma gång vid uppladdningen. Detta görs genom att tillämpa en flashapplikation för uppladdning av bilderna, modulen använder Drupals integrerade Upload modul som bas. Undermodulen dfgallery används till att visa ett flashalbum i form av en slideshow av valt bildgalleri.

#### **7.3.1.12 Editeringsverktyg**

FCKeditor är en WYSIWYG (från What You See Is What You Get) editeringsverktyg som används för att göra webbsidan mer användarvänlig. På detta sätt går det lättare att skapa och uppdatera sidor med information och bilder. Detta krävdes så att Karis Gossarna r.f:s styrelse själva kan uppdatera sina sidor i framtiden.

#### **7.3.1.13 Översättning**

Content translation användes för att få sidan relativt lätt översatt till svenska, väsentliga översättningar som saknades översatte jag fritt själv. Översättningar av



moduler som inte definierats i content translation är oöversatta, slutanvändaren ser ändå inte det mesta av det. Content translation är en del av Drupals kärna och följer med standardinstallationen av Drupal.

#### **7.3.1.14 Sökning**

Search-modulen används för att få en sökfunktion på Karisgossarna.fi. Search är en del av Drupals kärna och följer med standardinstallationen av Drupal.

### **7.4 Användarrättigheter**

I Drupal används olika rättighetsnivåer vilka begränsar vilka delar av webbsidan som olika typer av användare har tillgång till. Användarroller som används av Karisgossarna.fi är "anonymous users", "authenticated users", "Moderator" och slutligen "Administrator". Av dessa följer "anonymous users" och "authenticated users" med från början efter en installation av Drupal.

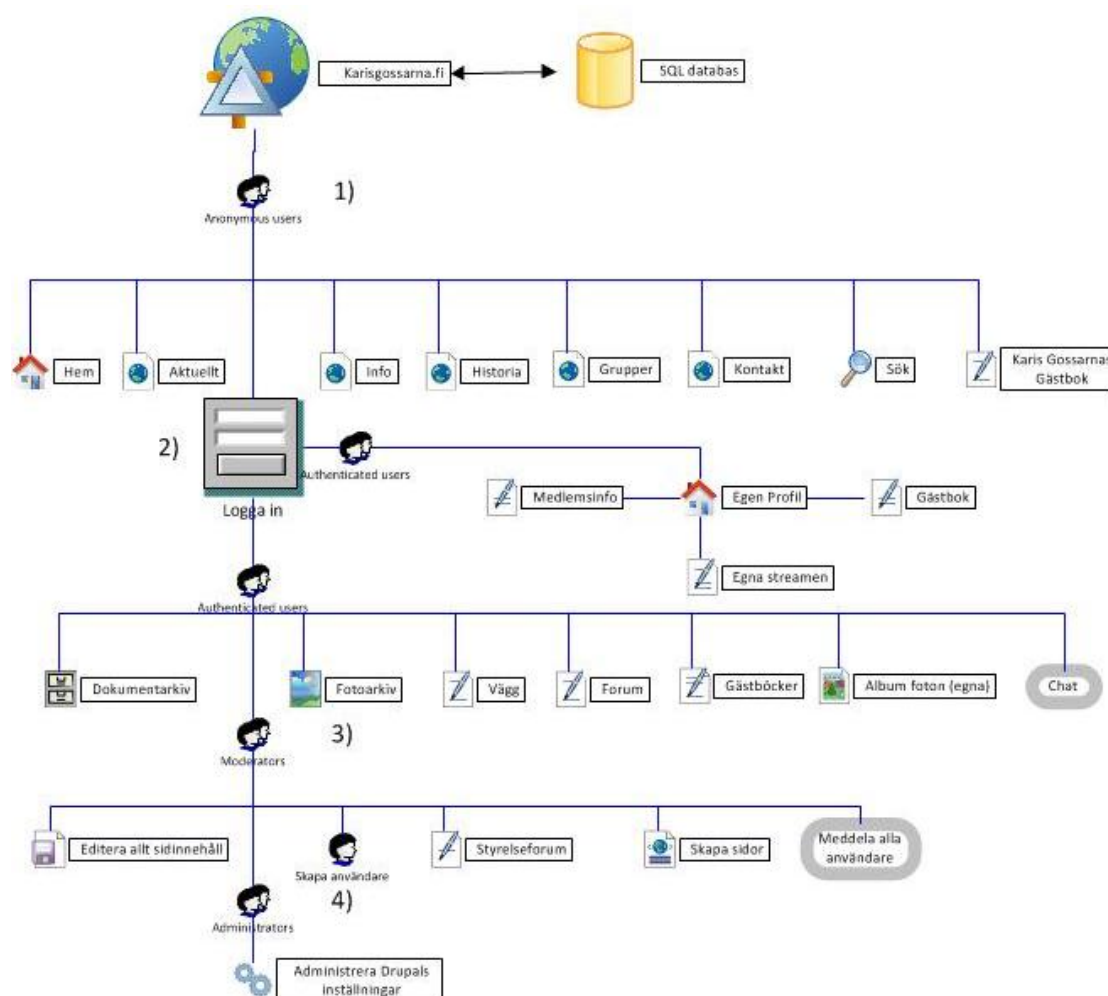
Användarrollen "anonymous users" är besökare som inte loggat in på webbsidan, de har inte tillgång till det interna nätet och kan endast kommentera i Karis Gossarnas gästbok och ser sidinnehåll ämnat för dem. Denna nivå syns i Figur 3 under behörighetsnivå 1.

"Authenticated users" är medlemmar som loggat in, men inte har befogenheter att göra några ändringar i webbsidans innehåll, förutom att dela en status på väggen, posta på andra medlemmars "stream" och i gästboken, posta i forumet för icke-styrelsemedlemmar och lägga upp bilder i ett eget galleri. Denna nivå kan ses i Figur 3 under behörighetsnivå 2.

"Moderators" har, utöver "authenticated users" användarrättigheter, även rättigheter till att editera andra användares forumposter, statusuppdateringar samt kommentarer. Ytterligare har de rättigheter att ändra på själva sidoinnehållet och kan därmed relativt enkelt med hjälp av modulen fckeditor fylla i sidoinnehåll för olika sidor i formen av text samt bilder. Denna nivå kan ses i Figur 3 under behörighetsnivå 3.

Utöver "Moderators"-rättigheter har "Administrators" även tillgång till "Administer" genvägen och genom den möjlighet att ändra på alla konfigurationsinställningar som finns tillgängliga i Drupal. Dessa användare har fått de flesta rättigheter som huvudanvändaren "admin" har från början. Den för närvarande enda användaren av

denna typ är användaren ”administrator”. Denna nivå kan ses i Figur 3 under behörighetsnivå 4. Orsaken till att administrators bör användas och inte själva huvudanvändaren ”admin” är givetvis en säkerhetsåtgärd. Ifall lösenordet byts och glöms bort eller ännu värre företeelser som dataintrång inträffar är det bättre på detta vis. Huvudanvändaren 'admin' är således endast tänkt att användas i "nödfall", vid större uppdateringar eller ändringar av sidans konstruktion.



Figur 4. Översikt av användaråtkomst och rättigheter för hela webbsidan

## 8 Medlemsinformationen

### 8.1 Spara i databas

Jag tänkte i ett skede skapa en hel modul i stället för att skriva in php-koden i en nod, vilket jag sist och slutligen ändå valde. Det är möjligt att använda php-kod inne i noder och block om man aktiverar modulen ”PHP filter” som hör till Drupals kärna.

Den största skillnaden är att koden sparas i databasen om man skriver koden i noder i stället för i en egen fil. Dock bör jag nämna att det oftast rekommenderas av den interna communityn på drupal.org att man skapar moduler och inte kod inne i noder. Orsaken är troligtvis ”best practice”. Det blir klarare för andra webb utvecklare, man får mindre databasstorlek, bättre möjlighet för underhåll av modulen av communityn samt mer integrerad kod med Drupals funktionalitet utan att behöva inkludera en massa filer. Många klagar också på att det är jobbigt att editera då man inte kan använda ett ordentligt editeringsverktyg, men det kommer man runt genom att kopiera och klistra in koden först då den är färdigskriven i webbläsaren.

Medlemsinformationen fylls in i fält på en nod på webbsidan och då man fyllt klart trycker man på en ”submit”-knapp som efter en snabb kontroll att värdena inte är NULL skickar iväg ifyllda data till databasen. Efter webbformulären, som är skrivna i html, kommer min php-kod som körs då man trycker på submit-knappen, därefter skriver koden om datan i webbformulären till variabler som exempelvis \$Fornamn och körs sedan igenom funktionerna stripslashes() och mysql\_real\_escape\_string() för att förhindra kritiska säkerhetshål som ”sql-injection”. Detta kan ses i Kodexempel 2, som jag kommer att referera till nedan. Variabelnamnen, fältnamn och tabellnamn är utbytta som en extra säkerhetsåtgärd.

Följande väsentliga kod att notera i Kodexempel 2 är variabeln \$uppdaterat, den sparar dagens datum, vilket behövs för att hålla reda på när medlemsinformationen uppdaterats. Detta behövs för att kunna urskilja vilka som sparats under den senaste månaden för epostrapporteringen.

Därefter i kodexempel 2 kommer \$q som uppdaterar tabellen info i databasen genom att definiera fälten som skall ha '%s' som värde, dvs. värden i form av strängar och vilka som skall ha '%d' som värden, dvs. skall ha integer-värden. Slutligen kollar koden om användaren redan har fyllt i medlemsinformationen i databasen i variabeln \$exist och om det finns, så skriver koden över informationen med den nyinskrivna datan. Ifall användaren inte skrivit in någon medlemsinformation ännu i databasen så måste sql-satsen för variabeln \$q skrivas om från ”update” till ”insert into” innan det fungerar att köra queryn.

## Kodexempel 2. Kod för uppdatering av medlemsinfo i databasen

```

if (isset($_POST['submit'])) {
...
$foramn = $_POST['Fornamn'];
    $foramn = stripslashes($Fornamn);
    $foramn = mysql_real_escape_string($Fornamn);
...
    $facebook = ($facebook ? $facebook : 'off');
...
$updaterat = mktime(0,0,0,date("m"),date("d"),date("Y"));
...
$q=      "UPDATE {info} SET mobiltele = '%s', hadress = '%s', hemtele = '%s', epostadr = '%s',
facebook = '%s', googleplus = '%s', fodelsedatum = '%s', updaterat = '%s', username = '%s' WHERE
userid = %d";

$exist = db_result(db_query("SELECT COUNT(userid) as count FROM info WHERE userid = %d",
$userid));

        if( $exist )
        {
            $x = db_query($q, $mobitel, $hemadress, $Hemtel,
                $epostadress, $Facebook, $Googleplus, $Fodelsedag,
                $updaterat, $user, $userid);
        }

        else
        {
            $q="INSERT INTO {info}
            (userid, fodelsedatum, hadress, hemtele, mobiltele, epostadr,
            facebook, googleplus, updaterat, username)
            VALUES (%d, '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s')";

            $x = db_query($q, $userid, $fodelsedag, $hemadress, $hemtel ,
                $mobitel, $epostadress, $facebook, $googleplus, $updaterat,
                $user);
        }
}

```

Slutligen skrev jag en logg över gjorda ändringar i medlemsinfon, för felsökningssyften, till en textfil i en mapp med begränsade rättigheter på webbservern. Detta gjordes med php-kommandot fwrite som skriver samma data som skrivs till databasen även i en ".txt"-fil.

## 8.2 Cron.php och automatiska e-postfunktionen, skriva till fil

Cron.php körs enligt schemat definierat med crontab via Nettihotellis webbaserade hanteringspanel i enlighet med instruktioner som finns tillgängliga på deras webbsida (Nettihotelli Internet Oy. 2012). Cron.php är inställd att köras en gång per timme då serverns minutantal är 35. Cron.php gör databasuprensningar och indexerar webbsidan för sökningsfunktionen som finns med på webbsidan.

Jag modifierade cron.php att inkludera en egen ".inc"-fil som skickar e-post om uppdateringar i medlemsinformationen till ett mailkonto, vilket är inställt att vidarebefordra eposten till den styrelsemedlem som ansvarar för att hålla medlemsregistret uppdaterat. På detta sätt behöver man bara ändra på adressen konfigurerad för vidarebefordran ifall ansvarsposten byter till en annan medlem och inte ändra i själva webbsidans källkod. Delar av denna ".inc"-fil finns för påseende i kodexempel 3 nedan.

Det första som bör noteras då cron.php inte körs från själva webbsidan är att då måste man inkludera bootstrap.inc, database.inc och database.mysql.inc och köra conf\_init(); för att ladda väsentlig konfiguration för Drupals funktionalitet och för att kunna kommunicera med databasen utan att skilt definiera anslutningsinställningar till databasen.

För att få reda vem som uppdaterat sin medlemsinformation och data de skrivit in under den senaste månaden måste en kontroll mellan tidpunkten då cron.php körs och tidpunkten då medlemsinfon skrevs in i databasen göras (se variabeln \$uppdaterat i kodexempel 2). I kodexempel 3 tar variabeln \$forst värdet av dagantalet av dagens datum och variabeln \$tim timantalet, detta på grund av att if-satsen efteråt kollar om det är den första dagen i månaden och om timantalet är 10. Denna kontroll räcker för att uppfyllas endast en enda gång per månad då cron.php är inställd att köras en gång per timme. Variabeln \$smanad har -2 för att det i variabeln \$qry står "WHERE uppdaterat > ' . \$smanad", således går det bara en månad bakåt från dagens datum vilket borde vara den första dagen i månaden då if-satsen uppfylls.

Efter att ha definierat vilka fält i databasen som är aktuella i kodexempel 3 skriver jag in informationen i info.xls i funktionen myarray(). Egentligen blir formatet inte i äkta Microsoft Excel-format, men filen blir mer överskådlig då man separerar värdena med tabbar och Microsoft Excel ändå klarar av att öppna filen. I variabeln \$data skriver jag in namnen på fälten och i \$data2 kommer inputen från databasen som fås från variabeln \$qry via variabeln \$result, variabeln \$tab lägger till tabbar mellan varje värde från databasen så de kommer i egna celler i exceldokumentet och slutligen så gör variabeln \$br ett radbyte till nästa gång något skall läggas till.

Kodexempel 3. Hämtar medlemsinfo från databas och skriver den till en fil

```

include('./includes/bootstrap.inc');
include('./includes/database.inc');
include('./includes/database.mysql.inc');

conf_init();
drupal_bootstrap(DRUPAL_BOOTSTRAP_CONFIGURATION);
drupal_bootstrap(DRUPAL_BOOTSTRAP_DATABASE);

$forst=date('j');
$stim=date('H');

$smnad = mktime(0, 0, 0, date("m")-2, date("d"), date("Y"));

if ($forst==1 & $stim==10)
{
  $qry= 'SELECT username, fodelsedag, hemadress, hemtel, mobiltel, epostadress, fcebook, gplus
FROM info WHERE uppdaterat > ' . $smnad;

  $result = db_query($qry) or die(mysql_error());

function myarray($result) {

  $filnamn='./noaccess/info.xls';
  $fil = fopen($filnamn,"w");
  $tab = "\t";
  $br = "\n";

  $data = "Anvandarnamn" . $tab . "Fodelsedatum" . $tab . "Adress" . $tab .
    "Hemtelefon" . $tab . "Mobil" . $tab . "E-post" . $tab . "Facebook" . $tab .
    "Google+" . $br;

  fwrite($fil,$data);

  while($row = mysql_fetch_assoc($result))
  {

    $nname = $row['uname'];
    ...

    $data2 = $nname . $tab . ...$br;

    fwrite($fil,$data2);
  }
}

fclose($fil);

```

Efter att ha skrivit medlemsinfon från databasen till en “.xls”-fil skall den skickas per epost till en adress som är inställd att vidarebefordra meddelandet till den styrelsemedlemmen som ansvarar för att hålla ordning på medlemsregistret. Denna funktion hade jag svårigheter med att få att fungera korrekt. Slutligen hittade jag en

webbsida som i stort sett hade lösningen på mitt problem (WebCheatSheet 2012). Den största utmaningen var därefter att komma på att göra ändringen till "Content-Type: application/vnd.ms-excel; name=info.xls", de flesta exemplen på nätet var antingen ".txt"-filer eller ".zip"-filer.

### **8.3 Vid användarregistreringen**

Om kryssrutan för att informera om nya användare per epost vid kontoregistreringen är ikryssad då en administratör skapar ett användarkonto åt en medlem i Karis Gossarna r.f. får medlemmen automatiskt ett e-post-meddelande innehållande inloggningsuppgifter och instruktioner för hur de skall gå till väga. Användaren tillbeds ändra sitt lösenord och skriva in sin medlemsinformation via en pageroute jag skapat för att underlätta processen. Pagerouten är en förhandsdefinierad följd av noder där det bara är att trycka på "framåt"-knappen för att gå vidare till nästa nod.

Då användaren har ändrat sina kontoinställningar till vad som önskas och sparat dem kommer användaren till följande steg av "pagerouten", nämligen att ändra medlemsinformationen. Medlemsinformationen består av födelsedatum, hemadress, hemtelefonnummer, mobiltelefonnummer, e-post-adress och om användaren använder Facebook eller Google+ och kan kontaktas via det.

Då användaren sparat sin medlemsinformation kommer den sista delen av "pagerouten". I denna del fyller användaren i Info som visas så att alla kommer att se det som besöker användarens användarprofil. Exempelvis intressen och information om användaren är en del av informationen. Till sist, då användaren sparar informationen, styr "pagerouten" användaren till en egen profilsida. Då är det färdigt att börja använda webbcommunityn.

## **9 Testprocessen**

### **9.1 Buggar i sidan och lösningar**

Under testfasen påträffades naturligtvis också ett antal problem och så kallade "buggar" i systemet. De största av dessa "buggar" är uppräknade nedan.

Vanliga inloggade användare "authenticated users" hade rätt att skriva i Styrelseforumet där endast styrelsens medlemmar skall ha rätt att skriva forumposter. Detta kunde inte åtgärdas med Drupals vanliga rättighetssystem då medlemmarna måste kunna posta i det andra forumet, i "advanced forum"-modulen fanns det inte heller några möjligheter att begränsa rättigheterna till specifika forum. Problemet åtgärdades med hjälp av några tilläggsmoduler, "Access control list", "Simple Access" och speciellt "Forum Access" genom att definiera specifika rättigheter för olika forum samt 'topics'.

Endel div-boxar på en del sidor dras ut längre än containern, detta händer ifall man skriver väldigt långa sammanhängande ord eller långa webbadresser. Detta problem åtgärdade jag med 'word-break: break-all;' för <p>-taggen, det borde inte påverka andra element som exempelvis webbadresser (så länge de får <a> taggen) som ofta är långa. Däremot får inte andra element som exempelvis <h1> och andra dylika taggar i sidan användas. Används någon sådan, via exempelvis fckeditor, så kan texten ännu vara delvis utanför div-boxarna vid speciellt långa sammanhängande textsträngar. Dock finns det, mig veterligen, inga ord i det svenska, engelska eller finska språket som skulle innehålla tillräckligt med tecken för detta.

"Authenticated users", dvs. vanliga användare, hade vid olika tidpunkter inte rättigheter att skriva på väggen, eller ens på sin egen stream. De hade inte heller rättigheter att editera sin egen profil. Dessa rättighetsproblem uppstod i samband med uppdateringen av vissa moduler som exempelvis Facebook-style Statuses (Microblog). Den egna profilsidans editering kan det dock hända att jag glömt att ge rättigheter till från första början.

Ifall man laddade upp en bild till ett galleri och bildnamnet innehöll ett å, ä eller ö så visades inte bilden. Det kan också hända att problemet berörde andra specialtecken. Problemet löstes genom att modifiera inställningarna för uppladdning av bilder till galleriet att ändra på bildnamnet vid uppladdningen till ett slumpvalt genererat bildnamn.

Chatten fungerade inte. Detta var på grund av att det behövdes ett så kallat förhållande medlemmar emellan så att QTChat kunde räkna vilka användare som var online. Jag hade inte testat denna funktion före uppladdningen till Nettihotelli då det i praktiken är svårt att chatta med någon annan på 'localhost'. Jag kunde dock ha gjort



några konfigurationsändringar i MoWeS Portable II och på så sätt tillåtit trafik till webbsidan inom det lokala nätverket ifall jag kommit på det i det skedet.

## 10 Problem och lösningar under utvecklingen

Under utvecklingen av Karisgossarna.fi påträffades naturligtvis ett flertal problem. En del av dessa problem var lätta att lösa, som exempelvis att komma ihåg att starta MoWeS Portable II innan man försöker komma åt webbsidan, andra var mer omfattande att lösa.

### 10.1 Påträffade i lokal miljö

Ett av de värsta problemen man kan stöta på då man utvecklar webbsidor är att hela sidan blir vit. Detta kallas ofta för "White screen of death" (WSOD) och sker oftast i Drupal pga. modulproblem. Det som egentligen oftast förorsakar detta problem är ett kritiskt fel i PHP-koden, ofta är dock visning av felmeddelanden avstängt från början så i stället för att felmeddelandet visas så blir sidan helt tom. Detta hände sammanlagt två gånger under utvecklingen av Karisgossarna.fi. (jwuk 2012)

Det kanske lättaste sättet att lösa problemet är att återställa en säkerhetskopia av påverkade filer eller databas före ändringen som förorsakade problemet. Detta gjorde jag båda gångerna jag påträffade problemet då jag som tur var hade nyligen gjorda säkerhetskopieringar. Ifall man inte har nyligen gjorda säkerhetskopior måste man lösa problemet manuellt och då bör man först sätta på felrapporteringen (och sidan migreras till en testmiljö först om webbsidan finns på leverantörens server) och läsa på vilken rad felmeddelandet kommer och lösa problemet därifrån. Om man däremot ändrar på koden i någon modul så blir det troligtvis problem då man skall uppdatera modulerna i framtiden.

Ett annat problem jag stötte på var ett problem vid uppdateringen av databasen. "Fatal error: Maximum execution time of 60 seconds exceeded in H:\Skola\Web Development\mowes\_portable\www\drupal\includes\database.mysql.inc on line 108"

Lösningen till problemet är att ändra på vissa värden i *php.ini* i *mowes\php5\php.ini*. Värdet som behövde ändras var *max\_execution\_time*, som jag ändrade till *480* i stället för *60*. Dock löste detta inte problemet så jag ändrade på Development value,

production value, och *max\_input\_time* till 480 sekunder också. Därefter fungerade uppdateringen av databasen. Man kan också prova att utföra kommandot man försöker göra flera gånger, eller uppdatera sidan, men är databasen stor så kommer kommandot inte att lyckas i vilket fall som helst, vilket var fallet vid båda mina problem.

Jag stötte också på meddelandet *"Site off-line, The site is currently not available due to technical problems"* flera gången under utvecklingsprocessen. Jag provade flera sätt att fixa detta som jag hittade. Jag kollade att *settings.php* hade rätt inställningar (här var problemet då jag uppladdade sidan till den riktiga servern), försökte logga in via olika urler för inloggning som jag hittade på nätet, "http://localhost/drupal/?q=user/login", "http://localhost/drupal/user", "http://localhost/drupal/?q=user", men inget tycktes fungera. Jag misstänker att detta inte fungerar på grund av att webbsidan är inställd att flytta användaren till sin egen profilsida då man loggat in. Därefter provade jag en annan potentiell lösning som jag hittade, vilket var att använda SQL-satsen i kodexempel 4.

Tyvärr hjälpte inte ens denna manipulering. Slutligen blev jag tvungen att använda en backup som var en vecka gammal, efter detta så såg jag till att göra frekventa och regelbundna säkerhetskopieringar åtminstone innan jag gjorde stora ändringar.

*Kodexempel 4. SQL kommando för att manuellt ta webbsidan ur "offline mode"*

```
UPDATE variable SET value = 's:1:"0";' WHERE name= 'site_offline';  
DELETE FROM cache WHERE cid = 'variables';
```

Ett aningen genant fel, som jag påträffade flera gånger i början innan jag gjorde en ominstallation av MoWeS Portable II, var att utöver att Drupal klagade på *"off-line, The site is currently not available due to technical problems"* så klagade phpmyadmin att *" Kan inte starta session utan fel. Kontrollera felmeddelanden i din PHP:s och/eller webbservers loggfil och konfigurera din PHP-installation korrekt."* Jag kom slutligen fram till att detta berodde på att jag ursprungligen installerat MoWeS Portable II till mitt USB-minne och fastän jag körde filerna från C: på datorn så var installationen kopplad till minnestickan och inget fungerade då den inte var i. En temporär lösning var att sätta i USB-minnet, men jag gjorde senare om MoWeS Portable II installationen på C:, vilket löste problemet.

Ett ganska kritiskt problem som jag stötte på under utvecklingsstadiet var att det ibland då sidor laddades uppstod något fel som resulterade i att det på sidan skrevs en stor del av webbsidans källkod med vit bakgrund. Troligtvis var det något fel i någon moduls källkod som orsakade detta. Det verkade som om felet skulle ha legat i "Photos"-modulen, eftersom det första som stod på sidan oftast var ".tar/photos/js/jquery.jeditable.pack.js?r"></script>". Det märkliga var dock att ifall man uppdaterade sidan så fungerade det igen, detta fel återskapades kanske en gång per femton sidladdningar. Problemet löste sig av sig självt efter ett tag, innan jag hunnit börja reda ut det ordentligt. Antagligen löste det sig efter att jag hade laddat ner alla nyaste versioner av alla moduler, av vilka en troligen var "photos"-modulen.

Andra relativt små problem jag stötte på var "*apache server has stopped working*" och att script inte svarade i Firefox och Internet Explorer. Lösningen till det Apache-relaterade problemet var att i MoWeS Portable II trycka "end mowes" och därefter starta om MoWeS Portable II, för att restarta servicen fungerade tydligen inte. Lösningen till att scripten inte svarade var helt enkelt att stänga av webbläsaren fullständigt och starta den på nytt. Det fungerade av någon anledning inte att sluta köra script på sidan i Internet Explorer, och inte heller att stoppa scripten manuellt i Firefox.

## 10.2 Påträffade vid uppladdning och i Nettihotellis miljö

Vid uppladdningen av sidan till den egentliga servern på nettihotelli.fi var det första problemet jag stötte på "*Site off-line, The site is currently not available due to technical problems*". "Inte igen", tänkte jag och förväntade mig ett stort problem. Den här gången var det dock pga. att inställningarna i *settings.php* var felaktiga och jag fick det fixat relativt lätt.

Ytterligare så slutade "header" att fungera, kan inte säga riktigt vad detta berodde på. I samband med detta märkte jag att det också uppstått problem med filuppladdningar till webbsidan. Detta visade sig bero på att det inte gick att skapa kataloger på nivån ovanf /httpdocs (katalog på servern motsvarande för www\Drupal) i filstrukturen pga. leverantörens standard-inställningar för filstrukturen. Detta innebar alltså att katalogen inte skapats i samband med överföringen till servern och att filuppladdningskatalogen för bilder på servern var "sönder". På 'localhost' hade jag bytt temporary upload-katalogen till katalogen www\upload\temp medan Drupal var i

katalogen `www\Drupal`. Detta var en extra säkerhetsåtgärd som egentligen inte är nödvändig. Slutligen bytte jag temporary upload-katalogen på servern tillbaka till `sites/default/files/temp` i stället.

Ett vanligt felmeddelande som påträffas då man försöker logga in i Nettihotellis phpmyadmin är "Cannot start session without errors, please check errors given in your PHP and/or webserver log file and configure your PHP installation properly.". Det här händer alltid då och då med Nettihotelli, troligtvis pga. hög trafik. Lösningen har i dessa fall varit relativt enkel, vänta en liten stund och uppdatera sidan på nytt.

Ett annat problem som jag haft med Nettihotellis phpmyadmin är att exportfunktionen av databasen inte lyckas, varken i Internet Explorer eller i Firefox. Enligt Nettihotelli är det troligtvis för att databasen är för stor, vilket kan vara fallet. Det spelar i alla fall ingen roll då jag fick SSH-tillgång till databasen och får databasen exporterad med "mysqldump".

### 10.3 Övriga utvecklingsutmaningar

Jag stötte på ett par lite knepigare utmaningar under utvecklingen av koden för att skriva ändringar i medlemsinformationen i databasen. Dessa var att checkboxarna jag använde för att medlemmarna skall kunna meddela om de kan och får kontaktas via Facebook eller Google+ inte fungerade på grund av att de returnerade *NULL* ifall de inte var ikryssade. Jag fick slutligen hjälp med detta problem via Drupals community, det löste sig genom omskrivning med koden "\$facebook = (\$facebook ? \$facebook : 'off');". *NULL*-värdet skrivs alltså om till 'off', vilket jag ansåg var förnuftigast då ccheckboxarna returnerar 'on' då de är ikryssade. Detta kan ses i kodexempel 2.

### 10.4 SPAM

Under testprocessen uppdagades problem med spambottar på Karis Gossarnas gästbok, vilken är öppen för kommentarer för 'anonymous users'. Kommentarer kom till ca en per minut med reklam. Problemet orsakade att databasen blev över fyra gånger sin vanliga storlek, se figur 4 för antal poster. Genast då jag märkte problemet satte jag sidan i offline-läge och installerade modulerna "Botcha" och "Bad Behavior".

Botcha lägger till fält som är osynliga för vanliga användare, men som spambotar tror är ett vanligt fält som skall fyllas i. Detta sätt är väldigt effektivt då de flesta spambotar alltid fyller i alla fält. Bad behavior har smart blockering av kända mönster för spambots redan i skedet för 4xx-felmeddelanden, detta sparar på dyrbar bandbredd. Jag blockerade också de vanligaste ip-adresserna som genom att läsa loggarna visade sig vara spammers, målet med detta var att spambotarna skulle generera mindre trafik och kräva mindre av bandbredden som Karis Gossarna r.f. betalar för.

"Bad Behavior" är inställd att köra under strict mode med BL access key och honeypot quicklinks i Blocks som visas på alla sidor. Bad Behavior lär sig intelligent hitta spambotar och blockerar i första hand, BOTCHA tar bort resten, det som Bad Behavior missar innan den lärt sig.



Figur 5. Antal spamkommentarer i den allmänna gästboken på webbcommunityn

För att ta bort spaminläggen men hålla kvar de riktiga inläggen från gästboken valde jag att från databasen ta bort alla inlägg som hade både epostadress- och webbsida-fälten ifyllt, ingen verkade ha postat med båda ifyllda.

## 11 Diskussion och resultat

Detta utvecklingsprojekt visade sig vara mer omfattande än jag i början insåg, en bidragande orsak till detta var min ringa erfarenhet i webbutveckling med hjälp av Drupal som CMS-system samt några missuppfattningar beträffande dess grundläggande utvecklingsprinciper. Under examensarbetet har jag stött på en hel del problem, men därigenom har jag även lärt mig massvis. Exempelvis visste jag inte hur man använder sig av modulen "Views", vilket visade sig vara ett användbart verktyg.

Ytterligare hade jag tidigare ingen aning om hur man använder sig av php-kod i Drupal, utan att utveckla hela egna moduler, eller hur man egentligen använder sig av ”blocks”. Php-, html- samt css-kunskaper som jag lärt mig tidigare, men sedan glömt, lärde jag mig på nytt samt utvecklade dem ytterligare under projektets gång. Förekomsten av spambots även på mindre populära webbsidor kom också som något av en kalldusch för mig och var en nyttig lärdom.

Den största begränsningen, som åtminstone till en början var i vägen för detta projekts förverkligande, kan man sammanfatta att var jag själv i form av min okunskap samt oerfarenhet. Jag har endast vid ett tidigare tillfälle under ett projekt under Novias handledning bekantat mig med Drupal. Dock bör erkännas att jag nog inte lärde mig behärska systemet ordentligt vid det tillfället, då jag endast arbetade med några specifika områden på grund av att arbetet utfördes i samarbånd med en projektgrupp. Det intryck som systemet lämnade mig med var att det är ett rätt råddigt administrations-system men med många möjligheter, bara man behärskade dem. Det visade sig att jag hade rätt då man kan säga att inlärningskurvan för Drupal är väldigt hög för en utvecklare. Detta problem kommer dock inte alls att förekomma i samma utsträckning för användarna då de endast har en mycket begränsad nivå av administreringsmöjligheter. För att hjälpa användarna att behärska systemet har jag även skrivit flera guider på forumet för hur man går till väga för att göra de vanligaste samt de mera invecklade företeelserna som jag kom att tänka på beträffande användningen av webbcommunityn.

I stort sett hölls utvecklingsmålen som sattes upp vid början av utvecklingen. Ett önskemål som lämnades bort i förverkligandet av webbsidan var att en fotorapport gärna skulle skickas till styrelsen per epost, så man kunde kolla vem som laddat upp vilka foton ifall man vill lägga till dem till det officiella galleriet. Detta lämnades bort på grund av att det skulle krävas så djup insättning i gallerimodulernas interna funktionalitet samt interaktion med databasen, dessutom tvivlar jag att det skulle bli användbart i praktiken och i stället skulle ha upplevts som SPAM. Utöver det finns på webbcommunityn färdigt länkar för ”latest albums” och ”latest images” som man kan få samma information genom. Ifall det fanns tid över var det önskvärt att det skulle finnas flera olika färgteman att välja mellan för att få webbsidan att se ut efter eget behag. Detta lämnades bort delvis på grund av att jag ansåg att det var väsentligare att

få funktionaliteten ordentligt gjord samt utvecklingen tog mer tid än jag förväntat mig.

Personligen anser jag att jag gjort ett gott val gällande CMS-system och att resultatet av detta slutarbete blev rätt lyckat, trots en massa påträffade problem, samt att arbetsplanen för slutarbetet inte på långt när hölls. Skulle jag idag göra om utvecklingen från början så skulle jag göra en ordentlig utredning över "Buddypress" och Drupal 7s funktionalitet och utvecklingsmöjligheter samt modulutbud, troligtvis skulle jag även välja något av de systemen i stället för Drupal 6.

Webbsidan finns för närvarande uppe på webben och kan besökas på [www.karisgossarna.fi](http://www.karisgossarna.fi). Tyvärr kommer endast medlemmar åt det interna innehållet som är den centrala delen av webbsidan som utgör själva webbcommunityn. Nu är det upp till Karis Gossarna r.f:s styrelse samt föreningens medlemmar att engagera sig och tillsammans göra webbsidan till en lyckad och välanvänd webbcommunity.

## Källförteckning

- Cox, Michelle. (2012) *Installation and basic setup*. Drupal.  
<http://drupal.org/node/517182> (hämtad: 2.11.2010)
- Creative Commons. (2011)  
<http://creativecommons.org/licenses/by-nc-sa/2.5/legalcode> (hämtad: 16.8.2012)
- Creative Commons. (2012)  
<http://creativecommons.org/licenses/by-nc-sa/2.5/deed.en> (hämtad: 16.8.2012)
- Drupal. (2012) *Licence.txt*.  
<http://api.drupal.org/api/drupal/LICENSE.txt/6> (hämtad: 13.3.2011)
- Gluskin, Shai. (2012) *Choosing a Drupal version*. Drupal.  
<http://drupal.org/documentation/version-info> (hämtad 15.8.2012)
- Hunter, Lee (2012) *The Drupal overview*. Drupal  
<http://drupal.org/getting-started/before/overview> (hämtad: 12.3.2011)
- jwuk. (2012) *The White Screen of Death (Completely Blank Page)*. Drupal  
<http://drupal.org/node/158043> (hämtad: 9.10.2011)
- Lokalkarkivet arresten. (2010) *Karis Gossarna r.f.*  
[http://www.arkiv.karis.fi/arkivforteckningar/Karis\\_Gossarna.pdf](http://www.arkiv.karis.fi/arkivforteckningar/Karis_Gossarna.pdf) (hämtad 27.8.2012)
- MozillaZine. (2012) *Problematic extensions*  
[http://kb.mozillazine.org/Problematic\\_extensions](http://kb.mozillazine.org/Problematic_extensions) (hämtad: 10.12.2011)
- Nettihanoteelli Internet Oy. (2012) *Ajastetut tehtävät (crontab)*  
[https://www.nettihanoteelli.fi/?Ohjeet/Ajastetut\\_teht%E4v%E4t](https://www.nettihanoteelli.fi/?Ohjeet/Ajastetut_teht%E4v%E4t) (hämtad: 27.3.2012)
- PhpMyAdmin. (2012) *PhpMyAdmin 4.0.0-dev Documentation*.  
<http://www.phpmyadmin.net/documentation/> (hämtad: 22.11.2010)
- RealWorld Graphics. (2009) How to make a fav-icon?  
<http://www.rw-designer.com/favicon> (hämtad: 12.3.2011)
- WebCheatSheet. (2012) *PHP: Sending Email (Text/HTML/Attachments)*  
[http://webcheatsheet.com/php/send\\_email\\_text\\_html\\_attachment.php#attachment](http://webcheatsheet.com/php/send_email_text_html_attachment.php#attachment),  
(hämtad 28.11.2011)