



Avdelningen för datavetenskap

Emil Danielsson & Patrik Lundberg

Mediadatabas med webbgränssnitt åt hembygdsförening

Media database with web interface
for local history society

Examensarbete 15 hp
Datavetenskap

Datum/Termin: 09-06-03
Handledare: Katarina Asplund
Examinator: Martin Blom
Löpnummer: C2009 : 05

**Mediadatabas med webbgränssnitt
för en hembygdsförening**

Emil Danielsson & Patrik Lundberg

Denna rapport är skriven som en del av det arbete som krävs för att erhålla en kandidatexamen i datavetenskap. Allt material i denna rapport, vilket inte är mitt eget, har blivit tydligt identifierat och inget material är inkluderat som tidigare använts för erhållande av annan examen.

Emil Danielsson

Patrik Lundberg

Godkänd, 09-06-03

Handledare: Katarina Asplund

Examinator: Martin Blom

Sammanfattning

Under årens lopp har Sveriges hembygdsföreningar samlat på sig en stor mängd material i form av bland annat foton, dokument och föremål. I takt med att mer och mer i samhället har datoriserats har behovet ökat att digitalisera även dessa samlingar. Detta är ett sätt att göra sin media lättare tillgänglig för allmänheten, men även att på ett smidigt sätt kunna säkerhetskopiera exempelvis foton och filmer.

Denna uppsats handlar om det arbete vi har utfört åt Nyeds hembygdsförening i Molkom. Uppgiften bestod i att utveckla en webbplats åt föreningen bestående av två delar, en databas och ett tillhörande webbgränssnitt för administrering. På webbplatsen kan personer med administratörsrättigheter lägga upp nya filer med tillhörande information som allmänheten sedan kan titta på och söka bland. Det som kan lagras på webbplatsen är bilder, videoklipp, ljud och dokument. Uppsatsen beskriver hur vi har gått tillväga i vårt arbete, och avslutas med att beskriva hur väl resultatet stämmer överrens med det mål som vi och uppdragsgivaren hade med projektet.

Media database with web interface

for local history society

Abstract

Throughout the years the Swedish local history societies have collected a large quantity of material, consisting of photos, documents and items, among others. As the society has become more computerized, the need has increased for these collections to become digitalized. This is a way to make their media more easily accessible to the general public, but also to easily be able to archive and preserve for example photos and videos electronically.

This thesis concerns itself with the work we have done for the local history society of Nyed in Molkom. The task set before us consisted of creating both a website for the society, and a reliable database from which the website draws its information. This project allows anyone with administrator privileges to upload new files and any additional information. Beyond that, general users can then search through and view these files. The following is an explanation of our methodology, concluding with an assessment of our performance based on the standards set by the requestor.

Innehållsförteckning

1	Inledning	1
2	Bakgrund	3
2.1	Dagens situation för hembygdsföreningar	3
2.1.1	Picasa	3
2.1.2	Coppermine	4
2.1.3	Sockenbilder	4
2.2	Förutsättningar och krav	5
2.3	Tekniker	9
2.3.1	MySQL	9
2.3.2	MySQL och XML	10
2.3.3	MyISAM och InnoDB	10
2.3.4	HTML	11
2.3.5	Cookies	11
2.3.6	JavaScript	11
2.3.7	CSS	11
2.3.8	AJAX	11
2.3.9	PHP	12
2.3.10	Begränsningar från webbhotellet	13
2.4	Sammanställning	13
3	Databasdesign	15
3.1	E/R-diagram (Entity/Relation)	15
3.2	Mappning	15
3.3	Tabeller	16
3.3.1	Media	16
3.3.2	Album	17
3.3.3	Användare	17
3.3.4	Sök	18
3.3.5	Motiv	18
3.3.6	Upphovsman	18
3.3.7	Landskap	19
3.3.8	Socken	19
3.3.9	Ort	19
3.4	Effektivisering	20
3.4.1	Indexering	20
3.4.2	Normalisering	20
3.5	Sammanställning	22
4	Användargränssnitt	23
4.1	Sidhuvud	23
4.2	Sidfot	24
4.3	Beskrivning av varje sida	25
4.3.1	Startsida	25

4.3.2	Visa album	25
4.3.3	Visa fil i en samling	26
4.3.4	Sökresultat.....	27
4.3.5	Visa media	27
4.3.6	Utskrift	28
4.3.7	Kommentar	28
4.3.8	Sökning	30
4.3.9	Logga in	30
4.3.10	Sökning för administratörer	31
4.3.11	Sökresultat för administratörer.....	31
4.3.12	Ny användare	32
4.3.13	Ny plats	32
4.3.14	Nytt album	33
4.3.15	Ny upphovsman	34
4.3.16	Ladda upp fil	34
4.4	Sammanställning	35
5	Implementation.....	37
5.1	Ny användare	37
5.2	Inloggning	38
5.3	Utloggning	38
5.4	Ändra lösenord	38
5.5	Uppladdning av filer.....	39
5.6	Redigering av fil.....	39
5.7	Ta bort fil	40
5.8	Sökning	40
5.9	Språkhantering	41
5.10	Modulering.....	42
5.11	Captcha	42
5.12	Sammanställning	43
6	Testning.....	45
6.1	Användarvänlighet	45
6.2	Navigering.....	46
6.3	Design	47
6.4	Sammanställning	47
7	Slutsatser.....	49
7.1	Framgångar med projektet	49
7.2	Kraven.....	49
7.3	Rekommendationer för framtida utveckling	50
7.4	Avslutande kommentarer	50

8 Referenser	51
Bilaga A.....	53
A1. Hårdvarurekommendationer för användare.....	53
A2. Stödda format på mediafiler	53

Figurförteckning

Figur 1: Översikt	2
Figur 2: E/R-diagram.....	15
Figur 3: Indexering i databasen.....	20
Figur 4: Sidhuvud.....	23
Figur 5: Sidfot	24
Figur 6: Startside	25
Figur 7: Mediafiler i ett album	25
Figur 8: Information om en mediafil	26
Figur 9: Visa mediafil i ett album	26
Figur 10: Sökresultat.....	27
Figur 11: Visa den valda mediafilen.....	28
Figur 12: Utskriftsvänligt format	28
Figur 13: Skicka kommentar till administratören.....	29
Figur 14: Sökalternativ	30
Figur 15: Administratörsinloggning samt byta lösenord	30
Figur 16: Sökalternativ för administratörer	31
Figur 17: Administratörs sökresultat	31
Figur 18: Skapa en ny användare	32
Figur 19: Lägg till en ny plats	33
Figur 20: Lägg till ett nytt album	33
Figur 21: Lägg till en ny upphovsman.....	34
Figur 22: Ladda upp en mediafil	34
Figur 23: Exempel på hur symboler kan användas för captcha.	43

Tabellförteckning

Tabell 1: Media	16
Tabell 2: Album.....	17
Tabell 3: Användare	17
Tabell 4: Sök	18
Tabell 5: Motiv.....	18
Tabell 6: Upphovsman.....	18
Tabell 7: Landskap	19
Tabell 8: Socken.....	19
Tabell 9: Ort	19
Tabell 10: Fyra olika situationer på geografisk information.....	22

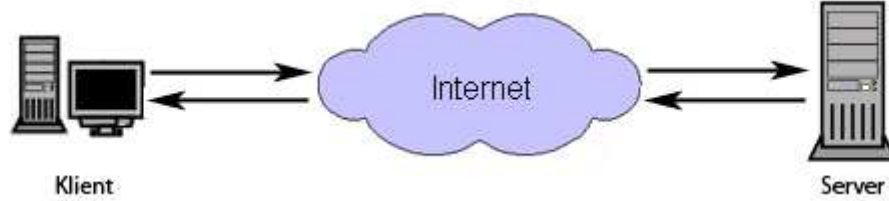
1 Inledning

År 2005 fick Kulturrådet [2] i uppgift av regeringen att mellan åren 2006 och 2009 genomföra stora sysselsättningsåtgärder inom kulturområdet, och denna satsning går under namnet Accessprojektet [3]. För att finansiera detta tilldelades ungefär 643 miljoner kronor av regeringen. Pengarna användes främst för att ekonomiskt hjälpa organisationer och institutioner att bevara och vårda men även för allmänheten tillgängliggöra offentliga samlingar, som till exempel museiföremål och gamla fotoarkiv. När summan av de ansökta pengarna kraftigt översteg den tillgängliga mängden pengar valde man att fokusera på tillgängliggörandet av samlingar.

Dessa pengar gav ideella föreningar, som hembygdsföreningar, möjligheten att utveckla lösningar för att kunna publicera den stora mängd media som de samlat på sig under åren.

Ett stort problem som uppstod var att lösningarna, som erbjöds de lokala hembygdsföreningarna, främst kom ifrån kommersiella företag som erbjöd sina tjänster mot betalning. Kritik har dock kommit mot detta, och många anser att kulturella samlingar är något som ska vara tillgängligt för allmänheten och inte ska drivas av vinstintresse. Som ett resultat av kritiken har alternativa gratistjänster utvecklats, men dessa håller inte tillräckligt hög kvalitet på grund av bristande funktionalitet och design. Orsaken till att Nyeds hembygdsförening [1] i Molkom kontaktade oss var att de behövde hjälp med att utveckla en egen tjänst som uppfyller deras krav på funktionalitet där de samtidigt kan vara med i utvecklingsprocessen.

Målet för projektet var att skapa en icke-kommersiell webbplats där ideella föreningar ska kunna lagra stora mängder media i olika former, till exempel bilder, ljudklipp, dokument och filmer. Portalen ska ha ett snyggt och stilrent gränssnitt för att locka yngre användare, men även vara lättanvänt eftersom det publicerade materialet främst kommer vara riktat till en generation där datorvanan inte är så hög. Figur 1 visar kopplingen mellan användaren och webbplatsen.



Figur 1: Översikt

Kapitel 2 i denna rapport tar upp bakgrunden till projektet, befintliga lösningar som redan används av hembygdsföreningar, de tekniker som vi har använt oss av under projektets gång samt uppdragsgivarens kravspecifikation och diskussion kring denna. I kapitel 3 presenteras designen av databasen med dess tabeller och olika sätt att effektivisera databaser. Rapportens fjärde kapitel tar upp och beskriver webbplatsens gränssnitt. Kapitel 5 presenterar en teknisk beskrivning av några utvalda funktioner. För att hitta eventuella buggar, upptäcka misstag och få synpunkter på designen gjordes en testning av webbplatsen. Resultatet och kommentarer från detta presenteras i kapitel 6. I det sista kapitlet, nummer 7, så presenteras en kort utvärdering av projektet och ifall alla krav har uppfyllts. Även förslag på förbättringar och vad som kunde ha gjorts annorlunda tas upp.

2 Bakgrund

Detta kapitel går igenom bakgrunden till att projektet har uppstått. Avsnitt 2.1 presenterar dagens situation för hembygdsföreningar där olika system används och vilka problem de medför. Avsnitt 2.2 listar de förutsättningar och krav vi mottagit från kunden och i avsnitt 2.3 finns en kortare beskrivning av några av de tekniker som används vid implementationen.

2.1 Dagens situation för hembygdsföreningar

I nuläget använder Nyeds hembygdsförening tre olika tjänster. Anledningen är att de inte har varit nöjda med den funktionalitet som erbjudits och har därför letat efter andra alternativa tjänster för att spara och publicera media. De tre tillgängliga tjänsterna beskrivs nedan.

2.1.1 Picasa

Picasa [4] är en gratistjänst från Google [5] för att publicera bilder på Internet där allmänheten kan utföra sökningar för att hitta rätt typ av motiv. Tjänsten har ett enkelt gränssnitt då man lätt kan välja att justera storleken på tumnagelbilderna genom att dra i ett skjutreglage (slider), sätta namnetiketter och utföra enklare bildredigering. Det går att skapa ett bildspel för varje album och skicka ett e-postmeddelande med bifogad bild, om man har ett medlemskap på Google Mail [6]. På en karta visas var kortet är taget och närliggande referenspunkter.

Det är dock svårt att få ut bra resultat från en bildsökning då det finns väldigt många användare som själva får sätta sökord på sina bilder. Det gör att massor av irrelevanta bilder visas endast för att den person som lagt upp bilden vill få många sökträffar. Gratisversionen av tjänsten har en utrymmesbegränsning på 1GB men detta utrymme går att uppgradera mot en årlig avgift.

2.1.2 Coppermine

Coppermine [7] är en tjänst med öppen källkod som använder sig av MySQL [11] för att lagra media. Tjänsten är lätt administrerad och har enkla funktioner för att skapa album, betygsätta bilder och visa hur många gånger en bild visats. Om webbläsaren har cookies aktiverat går det att skapa egna album som kan laddas ner. Det grafiska gränssnittet för att visa album och bilder har en väldigt enkel design vilket gör sidan lättnavigerad. En nackdel är att det inte går att välja hur många bilder som ska visas per sida. Dessutom har förstasidan mycket information om varje album och känns rörig då man måste skrolla ner på sidan en bra bit innan man når sidfoten.

2.1.3 Sockenbilder

Sockenbilder [8] har all önskad funktionalitet men är implementerat på ett bedrövligt sätt. Om man gör en sökning får man alla bilder i fullstorlek och en massa information om bilden som känns överflödigt. Tyvärr finns inte möjligheten att välja hur många sökresultat som ska visas per sida, vilket gör att man får skrolla ner på sidan för att kunna få en överblick av sökresultaten. Sökningen sker på både motivkategori och sökord vilket är bra, men dock görs ingen kontroll på stora och små bokstäver vilket leder till irritation. När man vill administrera databasen visas de 11 första tuplerna och endast en möjlighet att visa de nästa 11 tuplerna. En önskad funktion skulle vara att kunna välja vilken sida man vill se eller kunna söka sig fram till den tupel som ska redigeras.

Webbplatsen som vi ska utveckla kommer vara direkt anpassad för att visa och lagra media för hembygdsföreningen i Molkom. Deras önskemål om funktionalitet kommer att finnas med när vi tagit hänsyn till de olika lösningarna som redan finns. Framför allt kommer vår webbplats ha möjlighet att visa olika typer av media, vilket inte är möjligt i de andra lösningarna. Det finns ett flertal olika fält för att kunna fylla i information om mediafilen samt möjlighet att se ett bildspel från varje album. Tillsammans med ett lättadministrerat grafiskt gränssnitt kommer användare med begränsad datorvana kunna sköta och underhålla hela systemet. Då vi har en färdig produkt ska den lätt gå att vidareutveckla med exempelvis sökning i andra databaser. Möjligheten att importera data från andra databaser genom ett insticksprogram.

2.2 Förutsättningar och krav

För att strukturera projektet fick vi av uppdragsgivare en generell kravspecifikation som var uppdelad i följande delar:

- Måste ha
- Bör ha
- Bra att ha

Listan såg ut enligt följande:

Måste ha

- Applikationen skall vara helt Internetbaserad.
- Applikationen skall vara hierarkisk med en gemensam ingångsportal, t.ex. Värmland. Varje förening har sedan en egen portal.
- Mediafiler och all fältinformation skall matas in med hjälp av webb-formulär och sparas i en XML-fil.
- Inmatning av informationen skall göras av en eller flera personer med administratörsrättigheter.
- Vissa informationsfält skall konstrueras som förvalslistor, där den som registrerar väljer ett alternativ ur en befintlig lista.
- Applikationen skall ha en sökfunktion för att användare lättare ska hitta den information som önskas.
- Möjlighet skall finnas för besökare att skriva ut bilder med information, skicka in kommentarer och göra bildbeställningar.
- Webbsidan skall ha en engelsk språkversion.

Bör ha

- Bilderna bör finnas i åtminstone 2 olika upplösningar.
- Batch- och offlineinmatning av XML-data bör kunna göras.
- Applikationen bör innehålla en kartfunktion där mediafilers ursprung kan markeras.

Bra att ha

- Med hjälp av XML-data samt bildfiler skapa offline-album på CD/DVD.
- Datautbyte med programmet Sofie.
- Standalone-applikation för hembygdsförening som vill ha full kontroll.

Nästa steg i planeringen blev att gå igenom listan för att upptäcka eventuella oklarheter, men även för att se hur tidsåtgången för de olika punkterna förhöll sig till den tiden vi hade att lägga ned på projektet. Resultatet blev att vi konstruerade en egen kravspecifikation baserad på den första, och efter kontakt med kunden så kunde en version utarbetas som både vi och kunden var nöjda med.

Efter kontakt med vår uppdragsgivare så stod det mycket tidigt klart att vi inte skulle ha tid att implementera alla punkterna i listan, vilket betydde att vi var tvungna att prioritera. Vi valde då att lägga högsta prioritet på det som kunde klassas som grundfunktionalitet. Motiveringen till detta var att eftersom detta ska vara ett utvecklingsprojekt som går att arbeta vidare på efter vårt arbete är över, så är det viktigaste att lägga en stabil grund för webbplatsen genom att implementera grundfunktionaliteten och ett enkelt gränssnitt. Det som vi har ansett vara grundfunktionalitet är att användare ska kunna söka fram mediafiler och visa dessa filer med tillhörande information, och för administratörer att kunna ladda upp nya filer och spara information om dessa, samt att även kunna redigera denna information. Lägre prioritet gavs funktionalitet som inte låg inom dessa områden. En tidig tanke från uppdragsgivaren var att flera hembygdsföreningar skulle kunna ha var sin kopia av webbapplikationen och att dessa sedan skulle kunna kopplas samman genom en gemensam ingångssida. Det blev dock snart klart för oss att detta inte skulle hinnas med, och vi valde därför att utveckla en produkt endast för Nyeds hembygdsförening.

Detta innebar att punkt två under ”Måste ha” uteblev. Även alla punkter i ”Bra att ha” ströks, eftersom de inte kunde ligga inom tidsramen för projektet. Istället för att använda XML som databas så valde vi MySQL, och resonemanget kring det valet presenteras i stycke MySQL och XML i kapitel 2.3.2.

Nedan presenteras den modifierade kravspecifikationen som vi arbetade fram och sedan utgick ifrån när vi utförde vårt arbete:

Måste ha

- Applikationen skall vara helt Internetbaserad.
- Uppladdning av mediafiler och inmatning av all fältinformation ska ske med ett webbformulär, där en del sker med textfält och en del med rullista innehållande fördefinierade alternativ.
- Inmatning av informationen görs av en eller flera personer med administratörsrättigheter.
- Varje fil ska kunna kopplas till en viss motivkategori, vilket främst kommer vara för bildfiler. Därför finns också alternativet att lämna denna blank. Ett användningsområde för motivkategori är ifall en användare vill söka fram bilder med en viss typ av motiv. De motivkategorier som ska finnas är Verksamhet, Föremål, Landskap, Stadsbild, Byggnad, Interiör, Porträtt, Grupporträtt, Djur, Växter, Transportmedel.
- En sökfunktion ska finnas där användaren kan välja att söka på fritextfälten men även använda flera sökalternativ för att ytterligare precisera sökningen och begränsa sökresultaten. Användare ska kunna söka på album, landskap, socken, ort, upphovsman, årtal, sökord, titel, mediatext och motivkategori.
- Möjlighet ska finnas för användare att få fram utskriftsformat av bild och information.
- Möjlighet ska finnas för användare att kunna skicka in kommentarer till en administratör om en viss mediafil. Dessa kommentarer ska sedan kunna användas av administratören för att berika det beskrivande fritextfältet för en mediafil.
- Möjlighet ska finnas för användaren att se webbplatsen på språken svenska och engelska.
- Bilderna ska lagras i minst två olika storlekar, en tumnagel och en fullstorlek.
- Användaren ska kunna visa en fil och kunna få fram all dess information.
- Filer i kategorierna film, ljud, bild och dokument ska kunna lagras.

Bör ha

- Uppladdning av flera filer och offline-inmatning av bildinformation bör kunna göras, som senare kan skickas via FTP [46] direkt till webbhotellet.
- Informationen i databasen bör kunna visas i administratörsläge för att kunna redigeras.
- Användaren bör kunna visa ett album, vilket betyder att albumets bilder visas som tumnaglar som fungerar som länkar för att visa information om varje fil.

Till dessa krav fanns även önskemål om vilka uppgifter som i databasen skulle lagras för varje mediafil. En lista över dessa presenteras här nedan:

- *Filtyp* visar om filen är bild, ljud, film eller dokument.
- *Unikt id* är ett heltal för att enkelt kunna identifiera en specifik mediafil.
- *Filnamn* är helt enkelt filens namn som används för att skapa sökvägen till filen.
- *Landskap* är landskapet där exempelvis bilden är tagen.
- *Socken* är socknen där exempelvis bilden är tagen.
- *Ort* är orten där exempelvis bilden är tagen.
- *Upphovsman* är namnet på personen som har skapat median som filen innehåller.
- *Registrerad av* är namnet på användaren som har laddat upp bilden och informationen.
- *Tidpunkt för registrering* är tidpunkten när uppladdningen skedde.
- *Senast ändrad av* är användaren som senast ändrade informationen om en viss fil.
- *Tidpunkt för senaste ändring* är tidpunkten då den senaste ändringen av en fil skedde.
- *Titel* är ett eller ett fåtal ord som beskriver filen.
- *Anmärkning* är en större fritext som beskriver filen mer utförligt än *titel*.
- *Motivkategori* är ett ord som beskriver vilken kategori en bild tillhör.
- *Sökord* är ett antal ord som kan skrivas in för att sedan sökas på.
- *Publiceras-flagga* är ett booleskt värde för ifall användare ska kunna visa filen.
- *Säljes-flagga* är ett booleskt värde för ifall användare ska kunna beställa filen.
- *År* är det årtal som median skapades, exempelvis när fotot togs.
- *Årsprecision* visar hur precist år är genom värdena *exakt*, *cirka*, *före* och *efter*.
- *Album* är namnet på det album som mediafilen tillhör.
- *Förening* är namnet på den förening som har publicerat filen.

2.3 Tekniker

Detta avsnitt beskriver de tekniker som har används i skapandet av detta projekt.

2.3.1 MySQL

MySQL är ett relationsdatabashanteringssystem (RDBMS, Relationship DataBase Management System) [10], som har mer än 11 miljoner installationer. Programmet används på serversidan och har möjlighet att simultant ge flera klienter tillgång till de databaser som körs. MySQL ägs och sponsras av ett enda vinstdrivande företag, det svenska företaget MySQL AB, vilket numera är ett dotterbolag till Sun Microsystems [14] som innehar upphovsrätten till den största delen av kodbasen. Projektets källkod är tillgänglig under GNU, General Public License version 2, liksom under en rad egenutvecklade avtal. Första versionen av programmet släpptes i maj 1995 och har sedan dess släppts i sju olika versioner där den senaste är version 5.1.

MySQL finns i två olika versioner som är anpassade för olika användningsområden. Den ena är gratis att använda för icke-kommersiellt bruk men har då vissa begränsningar på support, underhåll och övervakningsverktyg. Den andra är Enterprise versionen som är en betalversion och har stöd för större databaser för kommersiellt bruk. Det finns då tillgång till mer avancerade verktyg och dygnet runt support.

MySQL är populärt på webben och fungerar på många olika systemplattformar så som Microsoft Windows [21], Mac OSX [20] och de flesta Linux/Unix-versioner [19]. Ett flertal bibliotek finns tillgängliga för olika programspråk för att kunna koppla upp sig mot en MySQL-databas, däribland C-språken, Java, PHP och Basic. Den stora bredden på tillämpningsområden har gjort MySQL till en av favoriterna för databasadministration runt om i världen. Dess popularitet för användning med webbapplikationer är nära knuten till populariteten för PHP [12], som ofta kombineras med MySQL.

Flera högtrafik-webbplatser använder MySQL för datalagring och loggning av data. Bland annat Wikipedia [27] som körs på MediaWiki programvara [28], vilken är skriven i PHP och använder en MySQL-databas. Andra stora företag som använder MySQL är Google [5], Nokia [29], Facebook [30], YouTube [31] och Flickr [32].

För att administrera MySQL-databaser kan man använda de medföljande kommandorads-verktygen. För användare som inte gillar att arbeta med kommandon finns det många GUI-verktyg, exempelvis Navicat [33] och phpMyAdmin [34].

2.3.2 MySQL och XML

XML [38], som står för eXtensible Markup Language, är ett märkspråk som lagrar data mellan ”taggar” på samma sätt som ibland annat HTML. Hur tagg-hierarkin får vara strukturerad finns definierat i ett dokument som fungerar som mall för XML-dokument. Nedan följer ett kort exempel på hur ett XML-dokument kan vara uppbyggt:

```
<bok sprak="engelska">
  <titel>XSLT Cookbook</titel>
  <forfattare>Sal Mangano</forfattare>
</bok>
```

Att använda XML var inget krav från uppdragsgivaren, men ett önskemål. Både MySQL och XML har sina för- och nackdelar, men valet föll till slut på MySQL. Till skillnad från XML så är MySQL en fullfjädrad databas med allt vad det innebär, och även om XML i många fall kan liknas vid en vanlig relationsdatabas så går det inte i slutändan att komma ifrån att det faktiskt är en vanlig textfil, och allt vad det innebär i form av till exempel sortering och säkerhet. Eftersom median som ska lagras på servern ska vara lätt att hitta för användaren, så är sökning i databasen en mycket viktig del av projektet. När det kommer till sökning så är MySQL mer effektivt, bland annat på grund av mer kompakt lagring av data i en relationsdatabas, och möjligheten att indexera olika fält för att lättare komma åt dem.

2.3.3 MyISAM och InnoDB

I MySQL-databaser går det att välja olika sätt att lagra data. Stöd för detta finns genom att man kan använda olika så kallade lagringsmotorer. Två av de vanligaste lagringsmotorerna heter MyISAM [35] och InnoDB [36], där båda motorerna har sina för- och nackdelar. Det som främst kännetecknar MyISAM är dess relativa enkelhet, vilket gör att det i många fall är det snabbare alternativet. MyISAM har också en funktionalitet som inte InnoDB har, vilket är indexering av kolumner med datatypen text. Detta ökar avsevärt hastigheten för sökningar av långa texter. Det som gör att InnoDB ibland är lite extra krävande är att den har stöd för transaktionshantering [16] och främmandenycklar [37]. Främmandenycklar gör att det på ett säkert sätt går att referera till andra tabeller, och transaktionshantering gör att en mängd instruktioner kan grupperas och sedan garanteras att alla körs eller ingen alls. Alla sådana här extra funktioner gör att lite extra kontroller måste utföras, men ofta är tiden dessa tar obetydlig. Även utrymmesmässigt är InnoDB mer krävande eftersom varje rad i databasen måste innehålla extra data för transaktionshantering.

2.3.4 HTML

HTML [13] står för Hyper Text Markup Language, och är inte ett programmeringsspråk utan ett märkspråk. Detta innebär att texten som finns i ett HTML-dokument inte är den som syns när dokumentet presenteras för användaren. HTML är istället instruktioner för hur text och grafiska element ska visas i ett program som kan tolka HTML-instruktioner. HTML består av en mängd olika start- och slut-taggar där ett elementets egenskaper bestäms.

2.3.5 Cookies

En cookie [44] är en textsträng som sparas på en användares dator av en webbläsare. De vanligaste egenskaperna som en cookie har är namn, textsträng och ett datum då den upphör att gälla. Cookies används t.ex. för att spara användarnamn och inställningar.

2.3.6 JavaScript

JavaScript [39] är ett programmeringsspråk som främst används för skript på klient-sidan vid utvecklingen av webbsidor. Koden kan ligga inbäddad i HTML-dokument. Språket är imperativt, och interpreterat, vilket innebär att det innan körning inte behöver kompileras. Vanliga användningsområden för JavaScript är validering av data, händelsehantering, cookie-hantering och webbläsardetektering.

2.3.7 CSS

CSS [18] står för Cascading Style Sheets, och innebär en möjlighet att skapa mallar där egenskaper för olika HTML-element kan sättas vilket bestämmer hur dessa ska visas för användare. Egenskaperna vars värden som kan sättas är bl.a. textstorlek, färger, bredd och höjd. En stor fördel med CSS-mallar är att man kan använda samma CSS för flera olika sidor, vilket gör det mycket lättare att utföra ändringar på hela webbplatser som har samma design. Man kan då ändra många sidors design genom att bara ändra i en fil.

2.3.8 AJAX

Ajax [25] står för Asynchronous JavaScript and XML, och är en tillämpning av JavaScript för att exempelvis öka interaktiviteten på webbsidor. Ett exempel på detta kan vara att

användaren har skrivit något i ett sökfält och då visas föreslagna sökord i en lista innan sökningen genomförs. För att detta ska fungera används objektet XMLHttpRequest [26] i JavaScript som anropar en webbserver och hämtar den nya datan till användarsidan. Detta kan användas för att uppdatera en sidas XHTML (eller HTML) och CSS, utan att behöva ladda om allt innehåll på sidan, utan bara det som har förändrats.

2.3.9 PHP

PHP som står för PHP: Hypertext Preprocessor, är ett skriptspråk som skapades 1995 av den danske programmeraren Rasmus Lerdorf [41]. Den senaste stabila versionen som har släppts är PHP5 men en nyare version, PHP6, håller på att utvecklas.

Huvuduppgiften för PHP är att skapa dynamiska sidor åt webbanvändare, istället för de statiska sidor som ges om enbart HTML används. PHP körs på en vanlig webbserver, och den tar textfiler innehållande PHP-kod som indata, och som utdata ges en webbsida som kan visas av en webbläsare, oftast då i form av HTML.

Det som PHP främst används till är skript på serversidan på webben, men nämnas bör även att det kan användas till ytterligare två saker, nämligen skrivande av desktop-applikationer och skript som är skrivna genom kommandoradsgränssnitt.

PHP kan användas på alla de vanliga operativsystemen, vilket inkluderar Unix, Windows och Mac OS X. Skriptspråket stöder även alla de vanliga databasvarianterna, bland annat MySQL, Oracle Database [39] och Microsoft SQL Server [40]. Detta i kombination med att det är gratis med öppen källkod har gjort att det har blivit ett mycket populärt sätt att använda för utvecklingen av dynamiska webbsidor. PHP används på över 20 miljoner webbsidor och är installerat på över 1 miljon webbservrar.

För att märka ut vilken text i en fil som är PHP-kod så skrivs denna inom avgränsarna `<?php` och `?>`. Det mesta av den grundläggande syntaxen är samma eller mycket lika sådana som återfinns i programmeringsspråk som Java och C#. Exempel på nyckelord är *if*, *while*, *for* och *return*.

Till PHP finns en stor mängd implementerade funktioner som kan användas för att sköta kopplingen till bl.a. databaser och olika protokoll. Ett exempel är ett bibliotek bestående av många funktioner som programmerare kan använda för att kommunicera med MySQL-databaser för att hämta och manipulera data.

2.3.10 Begränsningar från webbhotellet

Uppdragsgivaren är sedan tidigare kund hos webbhotellet One.com [22] där deras nuvarande webbsida administrerades. Bindningstiden med webbhotellet är ett år och några månader kvarstår fortfarande innan kontraktet behöver skrivas om. Enligt uppdragsgivaren har inga problem uppstått med den levererade tjänsten fram till projektets början och eftersom allt pappersarbete redan är klart bestämde vi oss för att fortsätta använda webbhotellets tjänster. Webbhotellet är en av de lågprisaktörer som finns på marknaden vilket ofta innebär vissa begränsningar i möjligheterna att själva göra inställningar för databasen.

En av begränsningarna vi inte var medvetna om när vi började utvecklingen av databasen var att endast MySQL:s standardlagringsmotor MyISAM kunde användas. Detta innebar huvudsakligen saknad för stöd av främmandenycklar och transaktionshantering, mer om detta i stycke 3.3.2.

En annan begränsning är att webbhotellet inte gav oss rättigheter att konfigurera inställningarna för den MySQL-server som hanterar databasen. Det innebär att när en text skall sökas igenom efter ett nyckelord måste nyckelordet vara minst fyra bokstäver långt.

En annan begränsning är att webbhotellet endast ger tillgång till 24 MB i RAM-minne. Det innebär en begränsning av filstorleken till cirka 2 MB för att en bildfil för att kunna skapa en tumnagelbild. Därför har vi satt en begränsning i koden på den maximala storleken på bilder till 1,8 MB för att inte riskera att en tumnagelbild inte ska genereras. Den maximala filstorleken som kan laddas upp till webbhotellet är begränsad till 12 MB.

Ytterligare en begränsning som försvårade arbete med databasen är att stöd saknas för att kunna arbeta genom externa program som ansluter till databasen. Istället utfördes arbetet genom ett webbaserat gränssnitt, phpMyAdmin, som ligger på webbhotellets servrar.

2.4 Sammanställning

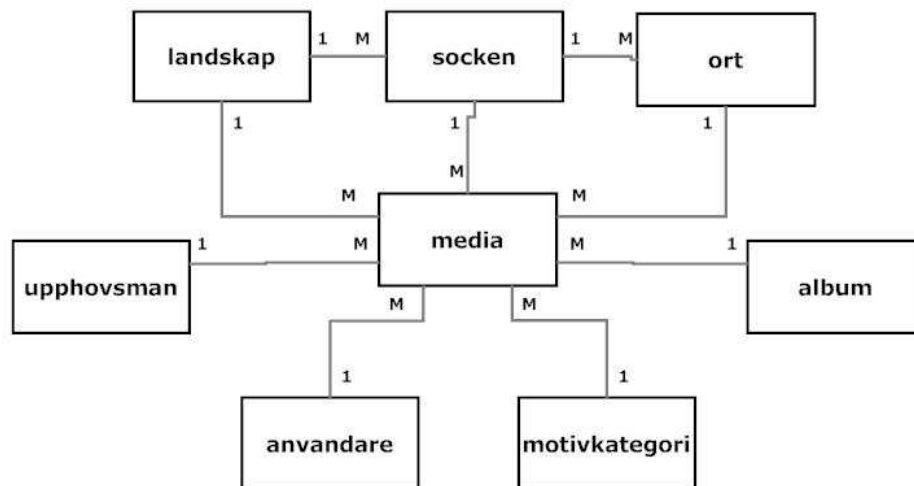
I detta kapitel har en kort introduktion getts om de befintliga lösningar som uppdragsgivaren i nuläget använder sig av för att lagra media. Dessa har många bra funktioner men har alla också vissa nackdelar. Uppdragsgivaren har sedan utgått från de bra egenskaperna för att skapa den ursprungliga kravspecifikationen. Efter diskussion kring denna kravspecifikation kunde en ny konstrueras med en del omprioriteringar, och fokus lades på grundläggande funktionalitet. Kapitlet har även tagit upp de olika tekniker som har använts i projektet och de viktigaste har varit PHP, för att skapa dynamiska webbsidor, och MySQL för att lagra data.

3 Databasdesign

Kapitel 3 går igenom designen av databasen där avsnitt 3.1 visar vår databasdesign i ett E/R-diagram. Avsnitt 3.2 handlar om mappningen från E/R-diagram till databas och avsnitt 3.3 beskriver vad varje tabell innehåller. Avslutningsvis handlar avsnitt 3.4 om effektivisering i form av indexering och normalisering.

3.1 E/R-diagram (Entity/Relation)

Figur 2 visas designen av databasen i form av ett E/R-diagram där media är den entitet som innehåller den viktigaste informationen. De transitiva kopplingarna som kan observeras mellan landskap, socken, ort och media beskrivs i avsnittet om normalisering, 3.4.2.



Figur 2: E/R-diagram

3.2 Mappning

Efter att ett E/R-diagram hade gjorts användes detta för att skapa databasstrukturen där alla entiteter i diagrammet blev tabeller i databasen. Den enda ändring som behövde göras var när det gällde entiteten *medias* attribut *sökord* som kan innehålla flera värden. Därför behövs en egen tabell för *sökord* vilket fick namnet *sok*.

3.3 Tabeller

Detta avsnitt behandlar de tabeller som används i databasen.

3.3.1 Media

Den utan tvekan viktigaste tabellen i databasen är *media*, se Tabell 1. Det är i denna som den mesta informationen om de uppladdade filerna lagras. Varje mediafil kan refereras till med ett unikt id, och varje fil har också i tabellen sitt filnamn lagrat så att filen går att komma åt i sin specifika mapp. Ytterligare sparad data om varje fil är titel, beskrivningstext, sälj- och publiceringsflagga, vilken typ av media filen är (bild, dokument, film, ljud) och år som median är skapad. Tabellen innehåller även ett antal kolumner för heltal, där varje siffra refererar till ett id i en annan tabell. Dessa kolumner är för upphovsman, album, motivkategori, landskap, socken och ort.

I tabellen finns också kolumner som visar när och av vem en fil blivit uppladdad och ändrad. Av vem en fil blivit uppladdad eller ändrad av visas med ett id som går att matcha mot ett id i tabellen för användare.

Fältnamn	Datatyp	Constraint
Id	int	PK, Auto increment
Titel	varchar	
Datum	date	
Ar	year	
Saljes	tinyint	
Publiceras	tinyint	
Mediatyp	varchar	
registreradav	int	
Andradav	int	
Andraddatum	date	
album_id	int	
Filnamn	varchar	NOT NULL
upphov_id	int	
motiv_id	int	
Anmarkning	text	
landskap_id	int	NOT NULL
socken_id	int	NOT NULL
ort_id	int	NOT NULL

Tabell 1: Media

3.3.2 Album

Tabellen *album*, se Tabell 2, innehåller information om alla album som finns. Varje album har ett unikt id som kan refereras till, ett namn och en beskrivande text. I tabellen finns också en sökväg som visar vart alla filer som tillhör det aktuella albumet ligger lagrade.

Tillsammans med namnet på filen går det att komma åt specifika filer i albumet.

Fältnamn	Datotyp	Constraint
Id	int	PK, Auto increment
Namn	varchar	
Text	text	
albumplats	varchar	NOT NULL

Tabell 2: Album

3.3.3 Användare

Tabellen *anvandare*, se Tabell 3, innehåller information om alla administratörskonton som finns. Varje användare har en unik id som används i "media" för att referera till en viss användare. Ytterligare information som finns i tabellen är användarnamn, ett hashat lösenord, en e-postadress och en siffra som avgör vilka rättigheter en användare har. E-postadressen i tabellen används för att skicka ut användaruppgifter till användare. Det finns två användartyper, och giltiga värden i kolumnen är 0 och 1. För att öka säkerheten så lagras i tabellen istället för det riktiga lösenordet en hashsumma, vilket innebär en 32 tecken lång sträng. Hashfunktionen som används har namnet MD5 [23].

Fältnamn	Datotyp	Constraint
id	int	PK, Auto increment
namn	varchar	NOT NULL
losen	varchar	NOT NULL
niva	tinyint	NOT NULL
email	varchar	NOT NULL

Tabell 3: Användare

3.3.4 Sök

Tabellen *sök*, se Tabell 4, innehåller tupler med attributen id och tagg. Attributet tagg är ett sökord som är kopplat till en viss mediafil genom id, som är en siffra som även finns i tabellen *media*. Eftersom en mediafil kan ha flera sökord så bildar id och tagg tillsammans primärnyckel för tabellen. Varje mediafil kan alltså inte ha samma sökord lagrat i databasen flera gånger.

Fältnamn	Datotyp	Constraint
id	int	PK
tagg	varchar	PK

Tabell 4: Sök

3.3.5 Motiv

Tabellen *motiv*, se Tabell 5, innehåller olika motivkategorier som också har varsitt unikt id. Varje mediafil kan genom detta id ange vilken motivkategori filen tillhör.

Fältnamn	Datotyp	Constraint
Id	int	PK, Auto increment
Typ	varchar	NOT NULL

Tabell 5: Motiv

3.3.6 Upphovsman

Tabellen *upphovsman*, se Tabell 6, innehåller namn på olika upphovsmän som också har varsitt unikt id. Varje mediafil kan genom detta id ange vilken upphovsman som har gett upphov till filen.

Fältnamn	Datotyp	Constraint
Id	int	PK, Auto increment
namn	varchar	NOT NULL

Tabell 6: Upphovsman

3.3.7 Landskap

Tabellen *landskap*, se Tabell 7, innehåller namn på olika landskap och id för att kunna refereras till.

Fältnamn	Datotyp	Constraint
Id	int	PK, Auto increment
Namn	varchar	NOT NULL

Tabell 7: Landskap

3.3.8 Socken

Tabellen *socken*, se Tabell 8, innehåller namn på olika socknar och id för att kunna refereras till. Varje tupel innehåller också ett id för att kunna visa vilket landskap en socken ligger i.

Fältnamn	Datotyp	Constraint
Id	int	PK, Auto increment
Namn	varchar	NOT NULL
landskap_id	int	NOT NULL

Tabell 8: Socken

3.3.9 Ort

Tabellen *ort*, se Tabell 9, innehåller namn på olika orter och id för att i media kunna referera till en specifik tupel. Varje rad innehåller också ett id som används för att referera till en viss socken.

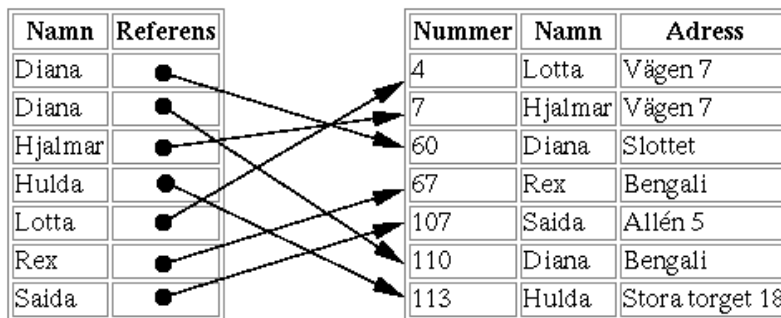
Fältnamn	Datotyp	Constraint
id	int	PK, Auto increment
namn	varchar	NOT NULL
socken_id	int	NOT NULL

Tabell 9: Ort

3.4 Effektivisering

3.4.1 Indexering

Eftersom sökning är en mycket stor och viktig del i projektet så har de textfält som används i sökning indexerats. När ett fält i en tabell indexerats skapas en tabell med det fältet och ytterligare ett fält som används för att referera till den rad som det första fältet tillhör i den ursprungliga tabellen. Detta ökar sökhastigheten eftersom databasen inte behöver stega igenom den ursprungliga tabellens alla fält och för att indextabellen kan vara sorterad på det attribut som sökningen sker på. Ett exempel på indexering visas i figuren nedan, se Figur 3, där index har skapats på Namn vilket har skapat en tabell sorterad efter Namn och med referenspekare till namnens rader i tabellen.



Figur 3: Indexering i databasen

I detta projekt har indexering använts för följande fält:

- media.titel
- media.anmärkning
- sok.tagg

3.4.2 Normalisering

När man designar en relationsdatabas är målet att skapa en korrekt representation av data, dess relationer och begränsningar. För att uppnå detta mål måste man identifiera lämpliga relationer. En teknik som kan hjälpa till att hitta dessa är normalisering som utvecklades av Edgar Frank Codd (1972) [42]. Normalisering utförs ofta genom en mängd olika tester på en relation för att bestämma om kraven uppnås eller bryter mot en given normalform.

Nedan följer ett antal definitioner av viktiga begrepp.

Definitionen av funktionellt beroende är; (egen översättning från engelska [43])

"Beskriver relationen mellan attributen i en relation. Till exempel, om A och B är attribut i en relation R, och B är funktionellt beroende av A (vilket betecknas $A \rightarrow B$), om varje värde av A är associerat med exakt ett värde på B. (A och B kan både bestå av en eller flera attribut)."

Fullständigt funktionellt beroende definieras som; (egen översättning från engelska [43])

"om A och B är attribut i en relation, B är fullt funktionellt beroende av A, om B är funktionellt beroende av A, men inte av en delmängd av A."

Transitivt beroende definieras som, (egen översättning från engelska [43])

"Ett tillstånd där A, B och C är attribut av en relation så att om $A \rightarrow B$ and $B \rightarrow C$, så är C transitivt beroende av A via B (förutsatt att A inte är funktionelltberoende av B eller C)."

Definitionen för första normalformen, 1NF (egen översättning från engelska [43])

"En relation i vilket korsningen mellan varje rad och kolumn innehåller ett och endast ett värde."

Definitionen av andra normalform, 2NF (egen översättning från engelska [43])

"En relation som uppnår första normalform och där varje icke primärnyckels attribut är fullständigt funktionellt beroende av primärnyckeln."

Definitionen av tredje normalformen, 3NF (egen översättning från engelska [43])

"En relation som är i första och andra normalform, och där ingen icke primärnyckels attribut är transitivt beroende av primärnyckeln."

Enligt önskemål från kunden ska man kunna sätta värdet "Saknas" på landskap, socken och ort vilket indikerar att man inte vet var median härstammar ifrån. Detta är en viktig funktionalitet som vi har valt att implementera, men som dock ibland leder till transitiva beroenden i databasen.

De olika situationer som kan uppstå finns nedan i tabellen, se Tabell 10.

Situation	Landskap	Socken	Ort
1	X	X	X
2	X	X	
3	X		
4			

Tabell 10: Fyra olika situationer på geografisk information.

X visar att informationen är känd.

På grund av de olika situationerna uppnår tabellen *media* endast andra normalform medan resterande tabeller uppnår tredje normalform. Om till exempel fältet ort är känt så kan man genom den informationen få fram landskap och socken utan att behöva lagra dessa i *media*. Detta eftersom varje ort är kopplad till en socken och en socken till ett landskap. Om antalet kända fält av de tre är fler än ett uppstår det alltså transitiva beroenden i tupeln.

3.5 Sammanställning

Kapitel 3 har gått igenom designen av databasen där E/R-diagrammet beskrivs efter de situationer som kan uppstå beroende på den kända geografiska informationen. Detta leder till att databasen är i andra normalform då transitiva beroenden skapas. För att ytterligare effektivisera databasen har indexering använts för att ge snabbare resultat på sökningar i textfält.

4 Användargränssnitt

Kapitel 4 går igenom användargränssnittet av webbplatsen och börjar med avsnitt 4.1 som beskriver sidhuvudet. Avsnitt 4.2 beskriver sidfoten och i avsnitt 4.3 beskrivs varje enskild sida. Kapitel avslutas med en teknisk beskrivning av användarhanteringen i avsnitt 4.4.

4.1 Sidhuvud

På samtliga sidor för vanliga användare visas högst upp ett sidhuvud, se Figur 4. Detta består av ett antal olika länkar och ett formulär där användaren kan välja olika sökalternativ. En bild med Nyeds hembygdsförenings logotyp är en länk till föreningens hemsida. Två bilder högst upp i högra hörnet med den brittiska och den svenska flaggan är länkar till mediaportalens startsida och byter då till språket för respektive land, alltså engelska eller svenska. Sökformuläret innehåller några olika delar. En av dessa är en vallista där användaren kan välja hur många sökresultat som ska visas per sida. Som standard visas endast ett textfält för fritextsökning och en knapp, "Sök", för att utföra sökningen. Under dessa fält för sökning finns en panel med fler sökalternativ för att användare lättare ska kunna hitta det de söker efter genom att precisera sin sökning. Denna panel är gömd som standard och visas genom att man klickar på "Fler sökalternativ". Vid tryck på sökknappen skickas all sökinformation som har valts och skrivits in med till sidan *sokresultat.php* där resultatet av sökningen visas.



The screenshot shows the header of the Nyeds Hembyggsförening website. It includes the organization's logo and name, a search bar with a 'Sök' button and a dropdown for 'Antal sökresultat per sida' (set to 15), and a panel titled 'Fler sökalternativ' containing several search filters: Motivkategori, Album, Upphovsman, Landskap, Secken, Ort, Från år, Till år, and Fyllyp. There are also flags for Swedish and English in the top right corner.

Figur 4: Sidhuvud

4.2 Sidfot

På samtliga sidor för vanliga användare visas längst ned på sidan en sidfot. Innehållet i den delen av sidan är mycket litet och består endast av fyra länkar. Av dessa länkar är det en som leder tillbaka till startsidan för portalen, en för att öppna ett kontaktfönster som kan användas om man vill kontakta hembygdsföreningen, en länk som öppnar ett nytt fönster med användarbeskrivning, och en sista länk för att behöriga användare ta sig till administratörsdelen.

På samtliga sidor för administratörer finns det också en sidfot, se Figur 5. Denna följer samma koncept som användardelens sidfot, och innehåller ett fåtal länkar. Här finns det en länk för att få fram användarbeskrivning i ett nytt fönster, en länk för att logga ut och ta sig tillbaka till startsidan, och en länk för att ta sig till startsidan utan att logga ut.



Figur 5: Sidfot

4.3 Beskrivning av varje sida

4.3.1 Startside

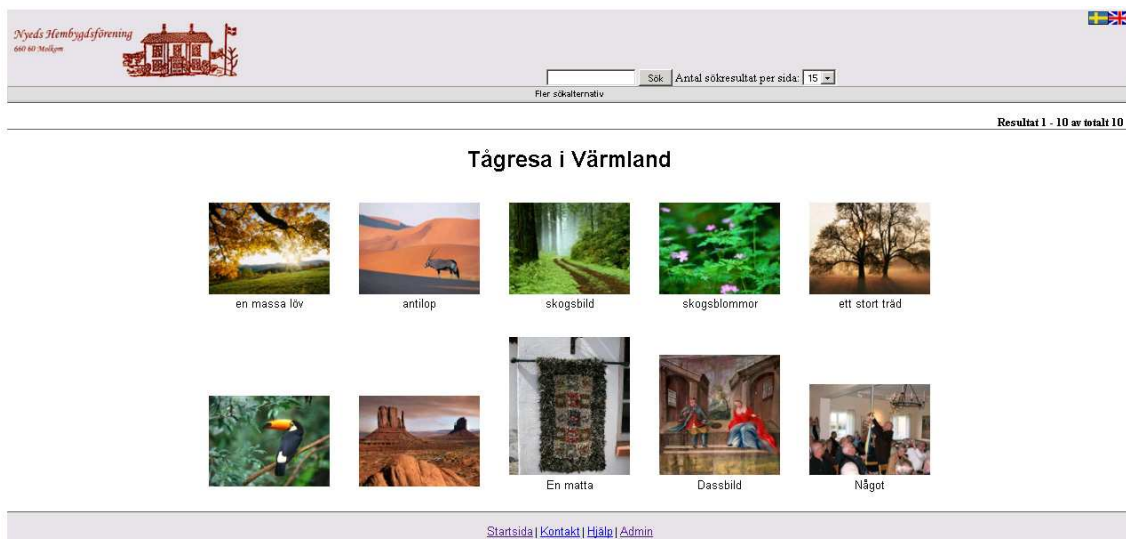
På startsidan, se Figur 6, visas de olika album som finns lagrade i databasen. Under bilden hittar man namnet på albumet. För att se innehållet i ett album klickar man på tumnagelbilden som laddar en ny sida och visar samtliga bilder i albumet.



Figur 6: Startside

4.3.2 Visa album

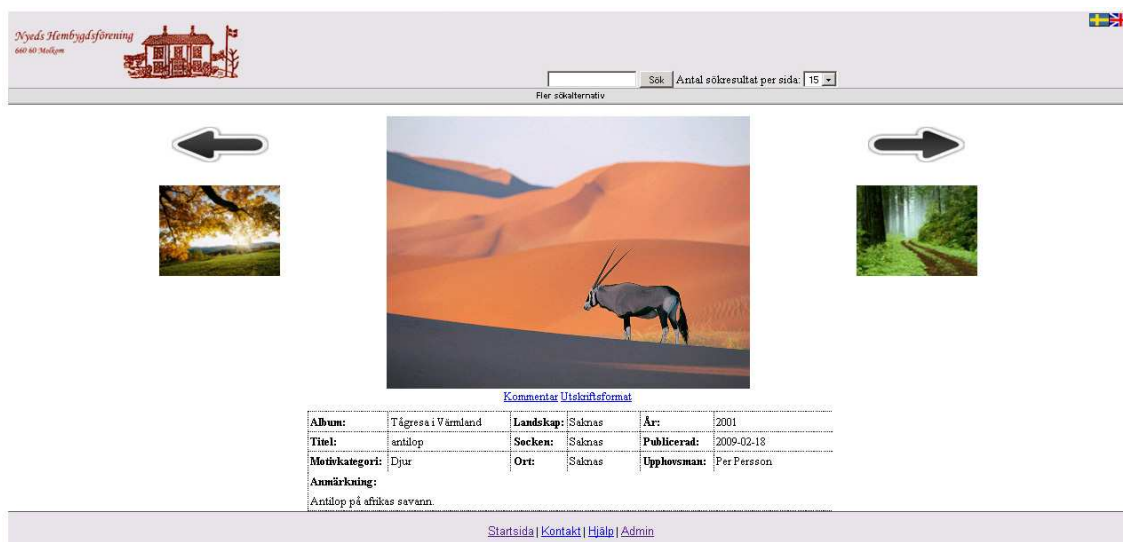
På albumsidan, se Figur 7, visas tumnagelbilder av alla mediafiler som finns i det valda albumet, under bilden står titeln på mediet. Som standard visas 15 bilder per sida.



Figur 7: Mediafiler i ett album

4.3.3 Visa fil i en samling

När man klickar på en tumnagel, se Figur 8, visas en bild i en mellanstorlek. För att visa mediet i fullstorlek kan man klicka på bilden som öppnas i ett nytt fönster. Under mediet finns det en länk till ett e-postformulär formulär för att skicka in en kommentar till administratören. I kommentaren skickas automatiskt filens id med för att administratören lätt ska kunna identifiera vilken fil kommentaren syftar på. Om man vill skriva ut bilden med tillhörande information kan man klicka på ”Utskriftsformat” som öppnar en ny sida med lämpligt storlek på bilden. På höger och vänster sida finns det tumnagelbilder som visar nästa respektive föregående media i albumet. Dessa går att klicka på för att bläddra mellan alla filer i albumet. Om mediet är först i serien kommer ingen tumnagel visas på vänster sida om huvudbilden eller om bilden är sist i serien visas ingen tumnagel på höger sida om huvudbilden.



The screenshot shows a web gallery interface for 'Nyeds Hembyggsförening'. At the top, there is a search bar and a dropdown menu for 'Antal sökresultat per sida' set to '15'. Below the search bar, there are three images: a small thumbnail on the left, a large central image of an antelope in a desert landscape, and another small thumbnail on the right. Below the central image, there are links for 'Kommentar' and 'Utskriftsformat'. A metadata table is displayed below the links, and at the bottom, there are links for 'Startsida', 'Kontakt', 'Hjälp', and 'Admin'.

Album:	Tågresa i Värmland	Landskap:	Saknas	År:	2001
Titel:	antilop	Socken:	Saknas	Publicerad:	2009-02-18
Motivkategori:	Djur	Ort:	Saknas	Upphovsman:	Per Persson
Anmärkning:	Antilop på afrikas savann.				

Figur 8: Information om en mediafil

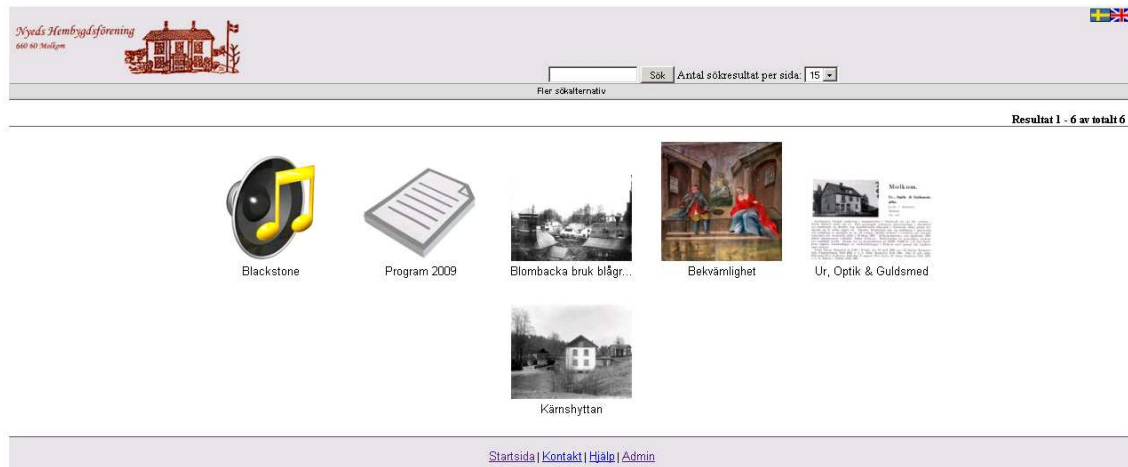
Under länkarna i mitten på sidan skrivs informationen ut om den aktuella bilden, se figur 9.

Album:	Julbord	Landskap:	Värmland	År:	1907
Titel:	Nyedporträtt	Socken:	Nyed	Publicerad:	2009-04-25
Motivkategori:	Porträtt	Ort:	Norum	Upphovsman:	Saknas
Anmärkning:	Vid avflyttning från Norums bruk år 1907 erhöil bruksförvaltaren John Herlitz och hans maka Julia ett album av minnesgoda Nyedsbor. Gåva av Erik Herlitz 1964.				

Figur 9: Visa mediafil i ett album

4.3.4 Sökresultat

Figur 10 visar sidan sökresultat där resultaten av en användares sökning visas. När en sökning har utförts kommer max 15 sökresultat visas som standard per sida. Det finns även möjlighet att visa 30 eller 50 resultat om så önskas. Resultaten visas som en tumnagelbild med respektive titel. För att se mer information om mediet klickar man på tumnagelbilden och kommer då till en ny sida.



Figur 10: Sökresultat


4.3.5 Visa media

Här visas en bild i en mellanstorlek, se Figur 11. För att visa mediet i fullstorlek kan man klicka på bilden som öppnas i ett nytt fönster. Under mediet finns det en länk till ett e-postformulär för att skicka in en kommentar till administratören. I kommentaren skickas automatiskt filens id med för att administratören lätt ska kunna identifiera vilken fil kommentaren syftar på. Om man vill skriva ut bilden med tillhörande information kan man klicka på "Utskriftsformat" som öppnar en ny sida med lämpligt storlek på bilden. Under länkarna i mitten på sidan skrivs informationen ut om den aktuella bilden, se figur 11 för exempel.

Nyeds Hembyggsförening
600 90 Molkom

Sök Antal sökresultat per sida: 15

Fler sökalternativ



[Kommentar](#) [Utskriftsformat](#)

Album:	Tågresa i Värmland	Landskap:	Värmland	År:	1908
Titel:	en massa löv	Socken:	Nyed	Publicerad:	2009-02-18
Motivkategori:	Djur	Ort:	Molkom	Upphovsman:	Per Persson
Anmärkning:	Hur går det med rediframåtningen?				


[Startsida](#) | [Kontakt](#) | [Hjälp](#) | [Admin](#)

Figur 11: Visa den valda mediafilen

4.3.6 Utskrift

Bilden visas i ett anpassat format för utskrift tillsammans med tillhörande information, se Figur 12.

http://nyed.se/exjobb/printformat.php?id=32



Id:	32	Landskap:	Saknas	Publicerad:	2009-02-18
Album:	Tågresa i Värmland	Socken:	Saknas	Publicerad av:	ernil
Titel:	antilop	Ort:	Saknas	Senast ändrad:	2009-05-04
Motivkategori:	Djur	Upphovsman:	Per Persson	Ändrad av:	putte
Filnamn:	Oryx Antelope.jpg	År:	2001		
Anmärkning:	Antilop på afrikas savann.				

[Skriv ut](#)

Figur 12: Utskriftsvänligt format

4.3.7 Kommentar

För att skicka en kommentar om en mediafil eller kontakta administratören med övriga synpunkter fyller man i sitt namn, e-postadress samt kommentar i fälten, se Figur 13. För att

kunna skicka in din kommentar måste man skriva in de fem bokstäver som finns i bilden. Det görs i textfältet precis under bilden. Om det av någon anledning inte skulle fungera med den teckensekvens som visas går det att klicka på ”Uppdatera bild” för att få en annan teckensekvens.

Denna funktion gör att endast människor kan skicka in kommentarer då en maskin inte kan läsa vad det står i rutan och därmed undviks risken för oönskade meddelanden (spam). Mer information finns om man klickar på länken ”Hjälp” i kommentarsfönstret.



The image shows a web browser window titled "Skicka kommentar - Mozilla Firefox". The address bar shows the URL "http://nyed.se/lejobb/kommentar.php?id=32". The form contains the following elements:

- Input fields for "Namn:" and "Email:".
- A large text area for "Kommentar:".
- A "Verifiering" section with the instruction "Skriv in texten från bilden:" and a CAPTCHA image showing the letters "F I C E O".
- A link for "Uppdatera bild" and a link for "Hjälp" below the CAPTCHA.
- An input field for the verification code.
- A "Sticka" button at the bottom right.

Figur 13: Skicka kommentar till administratören

4.3.8 Sökning

För att utföra en sökning kan man skriva in ett eller flera sökord i sökfältet som finns i sidhuvudet, se Figur 14. Det går även att precisera sin sökning om man klickar på den rullgardinspanel (collapsible panel) där det står ”Fler sökalternativ”. Resultatet av sökningen visas i ett nytt fönster.

Fler sökalternativ					
Motivkategori:	<input type="text"/>	Landskap:	<input type="text" value="Värmland"/>	Från år:	<input type="text"/>
Album:	<input type="text" value="Tågresa i Värmland"/>	Socken:	<input type="text" value="Nyed"/>	Till år:	<input type="text"/>
Upphovsman:	<input type="text" value="Per Person"/>	Ort:	<input type="text" value="Molkom"/>	Filtyp:	<input type="text" value="Bild"/>

Figur 14: Sökalternativ

4.3.9 Logga in

Inloggningen är endast till för administratörer, se Figur 15. För att logga in skriver man in sitt användarnamn och lösenord och klickar på knappen ”Logga in”. Valet finns för administratören att klicka i ”Kom ihåg mig” vilket innebär att man fortsätter att vara inloggad även om webbläsaren stängs.

Om man har glömt sitt lösenord går det att få ett nytt skickat till den e-postadress som man använde vid registrering. Av säkerhetsskäl går det inte att byta till ett valfritt lösenord.

Användarnamn:

Lösenord:

Kom ihåg mig

Logga in

Användarnamn:

Skapa nytt lösenord

[Tillbaka](#)

Figur 15: Administratörsinloggning samt byta lösenord

4.3.10 Sökning för administratörer

För att underlätta administratörens arbete går det att söka sig fram till den mediafil som man vill redigera eller ta bort, se Figur 16. Det går att söka direkt efter det id-nummer som varje mediafil har i databasen eller göra en generell sökning efter mediafilens information.

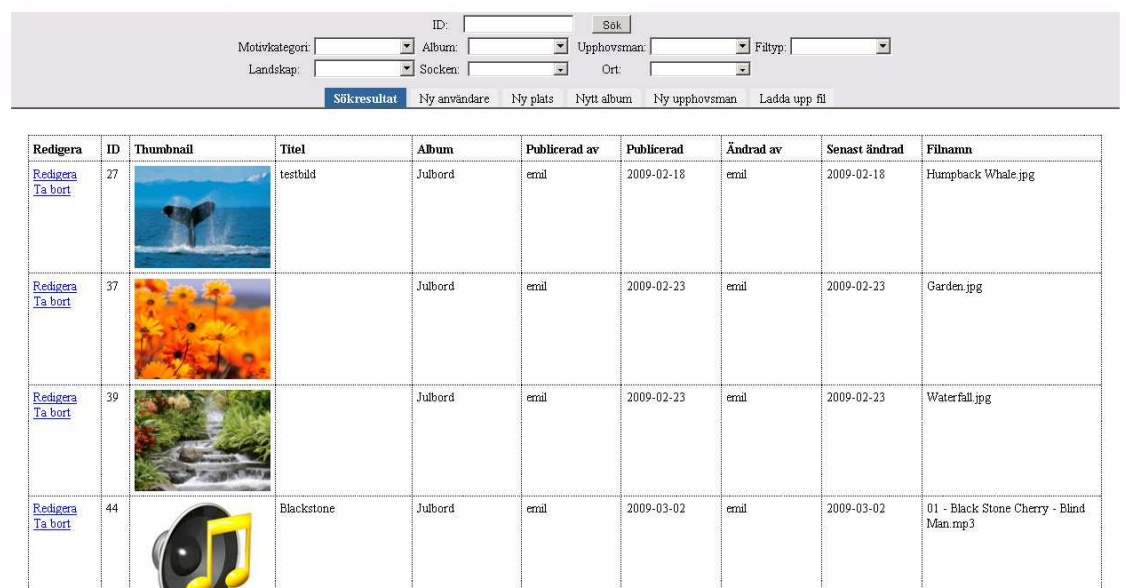






Figur 16: Sökalternativ för administratörer

4.3.11 Sökresultat för administratörer

För att redigera informationen om en mediafil söker man sig fram till den och klickar på ”Redigera”, se Figur 17. En ny sida öppnas där man kan ändra den information som finns om den valda filen. För att göra ändringarna permanenta klickar man på knappen ”Spara”.

För att radera en mediafil samt dess information klickar man på knappen ”Ta bort”. Filen med tillhörande tumnagel samt all information i databasen tas då bort från webbhotellet.



Redigera	ID	Thumbnail	Titel	Album	Publicerad av	Publicerad	Ändrad av	Senast ändrad	Filnamn
Redigera Ta bort	27		testbild	Julbord	emil	2009-02-18	emil	2009-02-18	Humpback Whale.jpg
Redigera Ta bort	37			Julbord	emil	2009-02-23	emil	2009-02-23	Garden.jpg
Redigera Ta bort	39			Julbord	emil	2009-02-23	emil	2009-02-23	Waterfall.jpg
Redigera Ta bort	44		Blackstone	Julbord	emil	2009-03-02	emil	2009-03-02	01 - Black Stone Cherry - Blind Man.mpeg

Figur 17: Administratörs sökresultat

4.3.12 Ny användare

För att lägga till en ny användare skriver man in ett önskat användarnamn och sin e-postadress, se Figur 18. Dessa måste vara unika, det vill säga det får inte finnas två användare med samma användarnamn eller e-postadress.

The screenshot shows a web interface for creating a new user. At the top, there is a search bar with an 'ID:' label and a 'Sök' button. Below this are several dropdown menus: 'Motivkategori:', 'Album:', 'Upphovsman:', 'Fälttyp:', 'Landskap:', 'Socken:', and 'Ort:'. A navigation bar below the dropdowns contains buttons for 'Sökresultat', 'Ny användare' (highlighted in blue), 'Ny plats', 'Nytt album', 'Ny upphovsman', and 'Ladda upp fil'. The main form area contains three input fields: 'Användarnamn:', 'Användarens e-mail:', and a 'Skapa användare' button. At the bottom of the page, there is a footer with links for 'Hjälp', 'Logga ut', and 'Till startsidan'.

Figur 18: Skapa en ny användare

4.3.13 Ny plats

För att lägga till ett landskap skriver man in ett landskapsnamn och klickar på knappen ”Lägg till landskap”, se Figur 19. För att lägga till en socken måste man först välja vilket landskap, i vallistan, som socknen tillhör och sedan skriva in ett sockennamn och klicka på knappen ”Lägg till socken”. För att lägga till en ny ort måste man först välja vilket landskap, i rullistan, som orten ligger i och i vilken socken, i rullistan, som orten tillhör. Ett ortnamn kan sedan skrivas in och läggas till genom att klicka på knappen ”Lägg till ort”.

ID: Sök

Motivkategori: Album: Upphovsman: Fälyp:

Landskap: Socken: Ort:

Sökresultat Ny användare **Ny plats** Nytt album Ny upphovsman Ladda upp fil

Nytt landskap Ny socken Ny ort

Lägg till landskap Lägg till socken Lägg till ort

[Hjälp](#) | [Logga ut](#) | [Till startsidan](#)

Figur 19: Lägg till en ny plats

4.3.14 Nytt album

För att skapa ett nytt album måste man skriva in ett nytt albumnamn, se Figur 20. Albumnamnet kommer att visas i panelen för sökalternativ och på startsidan för webbplatsen. Administratören måste skriva in ett namn på den mapp som kommer att skapas för att lagra albumets mediafiler i.

ID: Sök

Motivkategori: Album: Upphovsman: Fälyp:

Landskap: Socken: Ort:

Sökresultat Ny användare Ny plats **Nytt album** Ny upphovsman Ladda upp fil

Albumnamn:

Mappnamn:

Albumbeskrivning:

Skapa album

[Hjälp](#) | [Logga ut](#) | [Till startsidan](#)

Figur 20: Lägg till ett nytt album

4.3.15 Ny upphovsman

För att lägga till en ny upphovsman skriver man in namnet på den person som äger rättigheterna till det material som ska publiceras, se Figur 21. Dessa visas i en lista vid sökning.

The screenshot shows a web interface for adding a new author. At the top, there is a search bar with an 'ID' field and a 'Sök' button. Below this are several dropdown menus for 'Motivkategori', 'Album', 'Upphovsman', 'Fälyp', 'Landskap', 'Socken', and 'Ort'. A row of navigation buttons includes 'Sökresultat', 'Ny användare', 'Ny plats', 'Nytt album', 'Ny upphovsman' (which is highlighted in blue), and 'Ladda upp fil'. Below the navigation buttons is a 'Namn:' label followed by a text input field and a 'Lägg till upphovsman' button. At the bottom of the page, there is a footer with links for 'Hjälp', 'Logga ut', and 'Till startsidan'.

Figur 21: Lägg till en ny upphovsman

4.3.16 Ladda upp fil

För att ladda upp en fil till databasen klickar man på "Bläddra" och får då välja vilken fil som man vill skicka upp. Det enda som är obligatoriskt är att välja vilket album som filen ska tillhöra. För att lättare kunna hitta filen vid en sökning rekommenderas att fylla i mer information om exempelvis vilken kategori mediet tillhör.

The screenshot shows a web interface for uploading a file. At the top, there is a search bar with an 'ID' field and a 'Sök' button. Below this are several dropdown menus for 'Motivkategori', 'Album', 'Upphovsman', 'Fälyp', 'Landskap', 'Socken', and 'Ort'. A row of navigation buttons includes 'Sökresultat', 'Ny användare', 'Ny plats', 'Nytt album', 'Ny upphovsman', and 'Ladda upp fil' (which is highlighted in blue). Below the navigation buttons is a 'Välj fil:' label followed by a text input field and a 'Browse...' button. To the right of this is an 'Album:' dropdown menu with 'Julbord' selected. Below these are several other dropdown menus for 'Titel', 'Sökord', 'Årtal', 'Publiceras:', 'Säljes:', 'Motivkategori', 'Upphovsman', 'Landskap', 'Socken', and 'Ort'. The 'Publiceras:' and 'Säljes:' dropdowns have 'Ja' selected. Below these is an 'Anmärkning:' label followed by a large text area. At the bottom of the form is a 'Spara' button. At the bottom of the page, there is a footer with links for 'Hjälp', 'Logga ut', and 'Till startsidan'.

Figur 22: Ladda upp en mediafil

4.4 Sammanställning

Kapitel 4 har gått igenom det grafiska användargränssnittet där sidhuvudet har en dold panel med avancerade sökalternativ och språkinställningar som alltid finns tillgängligt. Sidfoten innehåller länkar till bland annat inloggning och hjälptexter. Både sidhuvud och sidfot är sparade i separata filer för att lättare kunna utföra förändringar på hela webbplatsen. En kort användarbeskrivning för varje sida på webbplatsen har sedan presenterats.

5 Implementation

Kapitel 5 går tekniskt igenom vissa delar av implementationen där avsnitt 5.1 beskriver hur en ny användare skapas. Avsnitt 5.2 och 5.3 går igenom hur in- och utloggning fungerar och avsnitt 5.4 beskriver hur lösenord genereras. Avsnitt 5.5 till 5.7 presenterar hur uppladdning, redigering och borttagning av filer sker på webbplatsen. I avsnitt 5.8 beskrivs hur sökning genomförs och avsnitt 5.9 behandlar hur språkfunktionerna fungerar. Kapitel avslutas med en presentation om moduleringen, avsnitt 5.10, och hur captcha förebygger spam-attacker i avsnitt 5.11.

5.1 Ny användare

Administratörer har möjligheten att skapa nya användarkonton och får då skriva in det nya kontots användarnamn och den e-postadress som ska vara kopplad till kontot. Dessa uppgifter hanteras sedan av ett PHP-skript för att registrera ny användare. De kontroller som utförs innan det nya kontot skapas är om användarnamnet är av godkänd längd, vilket är mellan 6 och 25 tecken, och att varken användarnamnet eller e-postadressen sedan tidigare finns registrerade. Om dessa krav uppfylls så registreras en ny användare i databasen. Ett e-postmeddelande skickas sedan till den registrerade adressen med användarnamn och lösenord till kontot.

För att öka säkerheten för användare så använde vi oss av två extra säkerhetsåtgärder för lösenorden. Den ena är att användaren inte själv får skapa sitt lösenord utan tilldelas en framslumpat sträng vilken består av både bokstäver och siffror. Detta görs för att det inte ska finnas lösenord som är för lätta att lista ut.

Den andra åtgärden är att använda krypterade lösenord. I vårt fall betyder det att lösenordet inte ligger sparad i databasen i klartext. Istället hashas lösenordet med hjälp av hashfunktionen MD5 med lösenordet som parameter där formatet är en sträng med 32 tecken.

Anledningen till att hashning används är för att skydda lösenorden från att övriga administratörer ska kunna läsa dem i klartext i databasen. Om någon skulle komma över innehållet i databasen kan de ändå inte använda personers användarkonton eftersom hashsumman i sig är värdelös.

5.2 Inloggning

För att en användare ska kunna logga in måste den ange sitt användarnamn och sitt lösenord. Dessa uppgifter tas emot av ett PHP-skript som kontrollerar ifall det är giltiga uppgifter som har matats in.

Skriptet tar först fram resultatet av hashfunktionen MD5 för lösenordet eftersom det är denna hashsumma av lösenordet som finns lagrad i databasen. Sedan undersöker skriptet ifall det finns en tupel i databasen som innehåller både det angivna användarnamnet och det hashade lösenordet. Om det finns en tupel som uppfyller detta betyder det att en användare har matat in giltiga inloggningsuppgifter.

För att sedan kunna upprätthålla tillståndet att en användare är inloggad så används i vår lösning cookies. Två cookies används och den ena, med namnet *user_name*, har användarnamnet som värde. För att förhindra att personer använder sig av andras användarnamn, vilket lätt hade kunnat göras om det bara fanns en cookie, så används ytterligare en för säkerhet, kallad *id_hash*. Den innehåller resultatet av MD5 för en sträng bestående av användarnamnet och en hemlig sträng som bara finns på servern.

För att kontrollera om en användare är inloggad undersöks först om det finns några cookies satta för *user_name* och *id_hash*. Om så är fallet kontrolleras också ifall resultatet, när värdet för *user_name* och den hemliga strängen hashas, är likadana som det som ligger sparad i *id_hash*. Om de är likadana så betyder det att användaren är inloggad.

5.3 Utloggning

När en användare ska logga ut så används ett PHP-skript som sätter värdena för *user_name* och *id_hash* till tomma strängar.

5.4 Ändra lösenord

Om användaren har glömt sitt lösenord finns det möjlighet att få ett nytt tilldelat sig om man anger sitt användarnamn. Ett nytt lösenord genereras då och skickas till den e-postadress som finns registrerad för det inmatade användarnamnet. Sedan hashas lösenordet och hashsumman sparas till databasen.

5.5 Uppladdning av filer

En av funktionerna i systemet är att kunna ladda upp filer och information. Denna uppladdning sker genom ett webbformulär med ett bakomliggande PHP-skript. I formuläret kan administratören välja vilken fil som ska laddas upp och skriva i eller välja den information som den vill ska sparas. Minimumkravet är att en fil och ett album väljs. Alla andra informationsfält är valfria att fylla i.

Fälten *titel*, *sökord*, *årtal* och *anmärkning* är fritextfält och *publiceras*, *säljes*, *album*, *motivkategori*, *upphovsman*, *landskap*, *socken* och *ort* kan väljas i vallistor.

Efter att administratören har tryckt på knappen ”Spara” och den valda filen har laddats upp så körs ett skript. Detta skript är till för att först kontrollera alla värden och om de blir godkända sedan spara filen på rätt plats, spara informationen samt skapa en tumnagelbild för filen om det är en bild. Platsen som filen sparas på beror på vilket album administratören valde att den skulle tillhöra eftersom varje album har en specifik mapp.

Skapandet av tumnagelbilder sker med en PHP-funktion som tar originalbilden och förminskar den och sänker även dess kvalitet. Denna tumnagelbild sparas sedan i undermappen ”thumbs” till albumets mapp.

5.6 Redigering av fil

Att kunna redigera och lägga till ytterligare information om en mediafil görs i ett ungefär likadant formulär som vid uppladdning av ny fil. Sidan för att ändra information öppnas med ett fil-id som parameter i sin adress och om parametern är giltig så läses all den filens information in från databasen. Varje redigerbart fält och vallista i formuläret fylls sedan med den redan sparade informationen. Administratören kan då ändra informationen i fälten och vallistorna och sedan trycka på ”Spara”. Ett PHP-skript kontrollerar först om något har ändrats och om så är fallet ifall de ändrade värdena är godkända. Ifall båda villkoren uppfylls så uppdateras de ändrade fälten i databasen.

Det som inte går att ändra genom formuläret är den fysiska fil som har laddats upp och vilket album som mediafilen tillhör.

5.7 Ta bort fil

Om en administratör vill ta bort en fil klickar den på länken ”Ta bort” bredvid filen i administratörsdelens sökresultatstabell. Genom en dialogruta får administratören bekräfta borttagningen och om han/hon gör det så används AJAX för att anropa en PHP-sida på servern med filens id som parameter i adressen. Skriptet tar bort huvudfilen, en eventuell tumnagelbild, tupeln för mediafilen i tabellen *media* och eventuella sparade sökord i *sok*. Efter borttagningen har skett så laddas sidan som administratören befinner sig på om.

5.8 Sökning

Sökning används för att användare ska kunna hitta specifika filer som de känner till någon viss information om. För att kunna erbjuda ett så bra sökverktyg som möjligt är det därför viktigt att antalet kategorier och fält som det går att söka på är högt.

För att begränsa sökresultaten kan man som användare välja att precisera sökningen efter motivkategori, album, mediatyp, upphovsman, landskap, socken, ort samt årsintervall för skapandet av mediafilerna.

Då det kan vara svårt för användaren att veta vad som finns för värden sparade i de olika kategorierna så presenteras de i vallistor. Dessa uppdateras varje gång en sida laddas.

Tillsammans med ett fritextfält blir detta ett kraftfullt verktyg för att hitta det man söker efter.

För att göra så effektiva sökningar som möjligt så skapas frågesträngarna mot databasen dynamiskt och beror helt på vilka sökalternativ som användaren har valt att använda sig av. Tabellen som ligger till grund för sökningen är *media*, som är den som innehåller det som unikt kan identifiera en mediafil. För varje sökalternativ som användaren har använt sig av läggs en Inner Join [24] till i söksträngen för tabellen där värdena för det aktuella sökalternativet finns för att tabellerna ska kopplas ihop i sökningen. Till söksträngen läggs också ett villkor till som måste uppfyllas, exempelvis *media.ort = 3* om användaren har valt en ort med id-nummer 3.

Det som användaren skriver in i fritextfältet används för att söka på fälten *titel* och *anmärkning* i tabellen *media* och *tagg* i tabellen *sok*. Fältet *tagg* söks igenom med operatorn *LIKE* tillsammans med *%*. På detta sätt behöver man inte bara matcha hela strängar utan även delsträngar. *%* används som ett sorts ”wildcard” efter ordet som söks på, vilket betyder att

matchning sker även om sökordet endast är en delsträng i början av en *tagg*. För de lite större textfälten *titel* och *anmarkning* så används istället operatorm *MATCH AGAINST* [15]. Denna operator tar fälten som ska sökas på som parameter till *MATCH* och sökordet till *AGAINST*, exempelvis *MATCH(media.titel, media.anmarkning) AGAINST('träd*' IN BOOLEAN MODE)*. * fungerar på samma sätt som % för *LIKE* och *BOOLEAN MODE* är ett speciellt sökläge som i vårt fall endast används för att kunna använda ”wildcard”. För att *MATCH AGAINST* ska fungera så måste fälten operationen utförs på vara indexerade med typen *fulltext*.

5.9 Språkhantering

Ett krav från uppdragsgivaren var att webbplatsen skulle finnas både på svenska och engelska och att det lätt skulle gå att byta emellan de båda språken. Endast användardelen behövde ha stöd för två språk, och endast den statiska texten behövde översättas, och alltså inte det som hämtades från databasen.

Det förmodligen enklaste och ett mycket vanligt sätt att implementera olika språkversioner på är att helt enkelt ha olika sidor för de olika språken. Detta var något som övervägdes, men vi valde dock istället en annan lösning.

Lösningen använder sig av PHP istället för vanliga statiska texter. Varje språk som webbplatsen har stöd för har en egen PHP-fil. Innehållet på de olika sidorna delas upp i fraser och ord som står tillsammans och dessa sparas i vektorer i varje språkfil. Till detta används vektorer vars värden inte tas fram med positioner utan med s.k. nycklar. Ett exempel illustreras nedan där värdet för nyckeln 'welcome' sätts till 'Välkommen':

```
$vektor['welcome'] = 'Välkommen';
```

Varje språkfil har vektorer med samma namn och nycklar, och det enda som skiljer dem åt är värdena för nycklarna. Dessa sätts för respektive fil till översättningen för just det språket. För att detta ska fungera måste en språkfil inkluderas för varje sida. För att ändra språk så trycker användaren på någon av flaggorna som finns högst upp på sidan. Flaggorna är bilder som fungerar som länkar, där länken innehåller en variabel *lang* som sätts till en språkkod. Ett exempel på en sådan adress visas nedan:

http://nyed.se/bildgalleri/index.php?lang=sv

Om en variabel i adressen sätts till ett giltigt värde, i vårt fall *sv* eller *en*, så sätts även en variabel för språk i användarens webbläsarsession till det värdet. Språkvariabeln avgör vilken fil som ska användas för sidan.

Den stora fördelen med detta system är mindre redundans jämfört med att ha olika sidor för olika språk. För att utöka med fler språk räcker det att lägga till en språkkod (exempelvis *sv*) för språket och en språkfil innehållande vektorer med översättningar.

5.10 Modulering

Många av sidorna på webbplatsen innehåller gemensamma delar som inte ändras från sida till sida. Sidorna på användardelen innehåller sidhuvud och sidfot som är lika på alla sidor. Det samma gäller på administratörsdelen, fast det är inte samma sidhuvud och sidfot som på användardelen.

Eftersom webbplatsen innehåller ganska många sidor och det ofta sker ändringar på de gemensamma delarna var det en fördel att ha separata filer för dem. Dessa filer kunde sedan inkluderas i de sidor där de behövdes, och skulle ändringar göras behövde de bara göras i en fil istället för i alla.

5.11 Captcha

En captcha [47] är ett test som används på många hemsidor för att avgöra ifall en besökare är en verklig människa eller bara ett automatiserat skript. Det sker genom att besökaren möts av en bild med en förvrängd text som är något svårläslig men ändå går att tyda för människor, men dock inte av automatiska program och skript. För att visa att man är en verklig person skriver man då in den text som visas på bilden och vid korrekt inmatning så visar man att man är en godkänd användare. Denna teknik används ofta på hemsidor i forum, gästböcker och vid olika sorters registreringar. Detta görs för att förhindra attacker där sidans funktioner missbrukas. Ett exempel finns nedan på hur det kan se ut, se Figur 23.



Figur 23: Exempel på hur symboler kan användas för captcha.

5.12 Sammanställning

Kapitel 5 har gått igenom tekniska aspekter på vissa delar av implementationen, som då en ny användare skapas och lösenord hashas och lagras i databasen. Vid inloggning kontrolleras uppgifterna och två cookies skapas som används för att kontrollera om användaren redan är inloggad. Användaren kan vid lyckad inloggning administrera databasen och ladda upp, redigera samt ta bort filer. För att underlätta administrationen används en sökfunktion som dynamiska skapar frågor. Kapitlet avslutades med att beskriva hur captcha används för att förhindra spam-attacker.

6 Testning

Kapitel 6 tar upp en del av den feedback som mottagits fått från våra testanvändare. Testen utfördes regelbundet under slutet av projektet där testanvändarna fick ett administratörskonto och fick prova på att använda all funktionalitet. De olika områden som feedback lämnades på var hur användarvänligheten, navigeringen och designen upplevdes. Om något behövde förtydligas, funktionaliteten inte motsvarade det som förväntades eller om något saknades. En sammanställning av testanvändarnas kommentarer finns här presenterade med korta svar eventuella åtgärder.

6.1 Användarvänlighet

Användarläge

1. Kommentar: *"Sökrutan längst upp söker "bara" i titelfältet, den bör också söka i fältet "Sökord!"*

Svar: Denna funktion finns, dock måste man separera sökorden med ett kommatecken när de läggs in.

Administratörläge

1. Kommentar: *"Hanterar inte felinmatning av för stora bilder. Det går att ladda upp en mycket stor bild (t.ex. 7MB) men det blir ingen klickbar tumnagel. Bilden går att se om man matar in motsvarande ID. "*

Svar: Detta problem beror på en begränsning från webbhotellet då deras inställningar, för PHP5 som endast ges 24 MB i RAM-minne, inte ger tillräckligt mycket resurser för att skapa en tumnagelbild för stora bilder. Mer om begränsningar från webbhotellet finns i stycke 4.2.

2. Kommentar: *"Om man väljer fel album vid uppladdning av en mediafil finns inte möjligheten att korrigera vid editering."*

Svar: Detta var en tänkt funktion, men med låg prioritet, då vi pga tidsbrist fokuserat på viktigare funktioner. En alternativ lösning är att ta bort bilden genom webbgränssnittet och ladda upp den igen till ett annat album.

3. Kommentar: *"Fälten "Filnamn", "Registrerat av", "Registreringstidpunkt", "Ändrat av" och "Ändringstidpunkt" bör finnas med i administratörsgränssnittet samt vid utskrifter, men inte nödvändigt för en besökare då det inte är väldigt relevant."*
Detta ansåg vi vara onödig information, men efter önskemål är de nu tillagda.

6.2 Navigering

1. Kommentar: *"Navigeringen för att bläddra mellan de olika sidorna i sökresultaten är lätt att förstå då sidorna är numrerade."*

Svar: Tanken bakom navigeringen har varit att den ska vara väldigt enkel att förstå, och när man vill öppna en ny sida ska det kännas naturligt vart man ska klicka.

2. Kommentar: *"När man som vanlig användare tittar på en bild i ett album så finns ju föregående och nästa bild som miniatyrer strax bredvid. Utmärkt, men de borde nog kompletteras med pilar vänster-höger så att användaren förstår att man befinner sig i en bläddringsbar serie."*

Svar: Detta var ett utmärkt förslag då det ger en tydligare indikation på att det finns en blädderfunktion. Vi valde därför att lägga till pilar enligt testarnas önskemål.

6.3 Design

1. Kommentar: ”Designen på sidan är väldigt stilren och utan en massa onödiga detaljer som förvirrar användaren. Dock skulle den grå färgen kunna bytas ut mot något lite mer färgglatt.”
2. Kommentar: ”*Det är väldigt smidigt att kunna utföra en avancerad sökning då den alltid finns tillgänglig på varje sida.*”

Svar: Tanken bakom designen på webbplatsen har varit att alla sidor ska använda sig av ett gemensamt enkelt gränssnitt för att användaren ska känna igen sig mellan de olika sidorna. För att inte tappa fokus på syftet med webbplatsen används inga filmer eller animerade ikoner som kan upplevas som störande.

6.4 Sammanställning

Kapitel 6 har gått igenom de kommentarer som testanvändarna lämnat under slutskedet av projektet. Under testningen mottogs många synpunkter på vad användarna ville förbättra i design, användarvänlighet och navigering. De flesta av problemen som synpunkterna beskrev valde vi att åtgärda, men en del lämnades på grund av antingen tidsbrist eller eftersom de var önskemål om ny funktionalitet eller utbyggnad av befintlig. I kapitlet har vi tagit upp ett urval av dessa synpunkter och kommentarer om hur vi hanterade dem.

7 Slutsatser

Syftet med projektet var att skapa en mediadatabas åt hembygdsföreningen i Molkom. I denna uppsats har vi presenterat bakgrunden till arbetet, hur vi har resonerat i valen kring lämpliga tekniker, implementeringen samt har visat resultaten av våra val.

7.1 Framgångar med projektet

Kunden är nöjd med projektet som har gett dem en grundläggande funktionell lösning för en webbplats med en mediadatabas. Alla multimediafiler kan nu samlas på samma ställe vilket underlättar hantering och administration. Vår förhoppning är att någon vill fortsätta att utveckla projektet med exempelvis en gemensam ingångsportal för att samla alla föreningar i ett landskap vilket ytterligare skulle förenkla tillgänglighörandet för allmänheten.

7.2 Kraven

Av den framtagna kravspecifikationen har vi utfört alla punkter förutom en. Det är möjligheten att skapa ett offline-album och senare kunna skicka upp informationen via FTP som vi inte har haft tid att utveckla.

Bland de önskade fälten som skulle lagras har vi implementerat alla utom två. De två vi inte har lagt till är årsprecision och förening. Årsprecision lämnades utan åtgärd på grund av tidsbrist, och förening valde vi att utelämna eftersom den inte ansågs vara nödvändig i en sådan här tidig version av webbplatsen. Fältet skulle användas för att visa vilken förening som laddat upp en viss fil, vilket endast är aktuellt när webbplatsen är gemensam för flera föreningar, vilket i nuläget inte är fallet.

Med undantag för ett fåtal saker har vi alltså gett kunden det som denne efterfrågade.

7.3 Rekommendationer för framtida utveckling

I detta stycke beskrivs nedan några förslag på framtida funktioner som eventuellt skulle kunna ingå i en vidareutveckling.

- För att det ska bli ännu tydligare vart en mediafil härstammar ifrån skulle ytterligare ett fält, plats/gata, kunna skapas för geografisk information.
- För att inte begränsa webbplatsen för endast mediafiler som har skapats i Sverige kan ett nytt fält med land skapas som är kopplat till landskap.
- För att underlätta för administratörer vid inmatning kan en hjälptext (tool tip) om formatering för varje fält visas.
- Sammankoppling med andra hembygdsföreningar för att skapa en större webbplats.
- Ett byte av lagringsmotor för databasen från MyISAM till InnoDB för att i databasen kunna använda sig av främmandenycklar och transaktionshantering är något som skulle öka säkerheten för webbplatsen.

7.4 Avslutande kommentarer

Det har varit ett intensivt och roligt arbete där vi har fått använda oss av de kunskaper som vi har fått under utbildningen. Vi har jobbat med många nya tekniker under projektet och får förhoppningsvis användning av dessa kunskaper när vi kommer ut i arbetslivet. Det har varit spännande att få genomföra ett projekt ända från planeringsstadiet till slutet av implementationen.

8 Referenser

- [1] Hembygdsföreningen, <http://www.hembygd.se>, 2009-04-07
- [2] Kulturrådet, <http://www.kulturradet.se>, 2009-04-27
- [3] Accessprojektet, <http://www.kulturradet.se/default.aspx?id=7&epslanguage=SV>, 2009-04-07
- [4] Picasa, <http://picasa.google.com>, 2009-04-07
- [5] Google, www.google.com, 2009-04-07
- [6] Google Mail, www.gmail.com, 2009-04-07
- [7] Coppermine, <http://coppermine-gallery.net>, 2009-04-07
- [8] Sockenbilder, <http://www.sockenbilder.se>, 2009-04-07
- [9] SOFIE, www.vbm.se/tjanster/sofie.html, 2009-04-07
- [10] RDBMS, <http://en.wikipedia.org/wiki/RDBMS>, 2009-04-07
- [11] MySQL, <http://dev.mysql.com/doc/>, 2009-04-07
- [12] PHP, <http://www.php.net>, 2009-04-07
- [13] HTML, <http://www.w3.org/TR/1998/REC-html40-19980424>, 2009-04-07
- [14] Sun Microsystems, <http://www.sun.com>, 2009-04-07
- [15] Match Against, http://dev.mysql.com/doc/refman/5.1/en/fulltext-search.html#function_match, 2009-03-31
- [16] Transactional analysis, http://en.wikipedia.org/wiki/Transactional_analysis, 2009-04-07
- [17] JavaScript, <http://en.wikipedia.org/wiki/JavaScript>, 2009-04-07
- [18] CSS, <http://www.w3.org/Style/CSS>, 2009-04-07
- [19] UNIX, <http://www.unix.org>, 2009-04-07
- [20] Mac OS X, <http://www.apple.com/macosx/>, 2009-04-07
- [21] Microsoft Windows, <http://www.microsoft.com/WINDOWS>, 2009-04-07
- [22] One.com, www.one.com, 2009-04-07
- [23] MD5, <http://tools.ietf.org/html/rfc1321>, 2009-04-07
- [24] Join-villkor, <http://dev.mysql.com/doc/refman/5.0/en/join.html>, 2009-03-31
- [25] Ajax, <http://sv.wikipedia.org/wiki/AJAX>, 2009-04-23

- [26] XMLHttpRequest, <http://www.w3.org/TR/XMLHttpRequest>, 2009-04-23
- [27] Wikipedia, <http://wikipedia.org>, 2009-04-23
- [28] MediaWiki, <http://www.mediawiki.org>, 2009-04-23
- [29] Nokia, <http://www.nokia.com>, 2009-04-23
- [30] Facebook, <http://www.facebook.com>, 2009-04-23
- [31] Youtube, <http://www.youtube.com>, 2009-04-23
- [32] Flickr, <http://www.flickr.com>, 2009-04-23
- [33] Navicat, <http://www.navicat.com>, 2009-04-23
- [34] PhpMyAdmin, <http://www.phpmyadmin.net>, 2009-04-23
- [35] MyISAM, <http://dev.mysql.com/doc/refman/4.1/en/myisam-storage-engine.html>, 2009-04-27
- [36] InnoDB, <http://www.innodb.com/products/innodb>, 2009-04-27
- [37] Främmandenycklar, http://en.wikipedia.org/wiki/Foreign_key, 2009-04-27
- [38] XML, <http://www.w3schools.com/xml/default.asp>, 2009-04-27
- [39] Oracle databases, <http://www.oracle.com/database/index.html>, 2009-04-27
- [40] Microsoft SQL, <http://www.microsoft.com/sqlserver/2008/en/us/default.aspx>, 2009-04-27
- [41] Rasmus Lerdorfs, <http://blog.360.yahoo.com/rlerdorf>, 2009-04-27
- [42] Edgar Frank Codd, http://en.wikipedia.org/wiki/E.F._Codd, 2009-05-05
- [43] Connolly Begg, Database systems Third edition, Addison Wesley
- [44] Cookies, http://en.wikipedia.org/wiki/HTTP_cookie, 2009-05-05
- [45] Västerbottens museum, <http://www.vbm.se>, 2009-05-07
- [46] RFC 959, <http://www.ietf.org/rfc/rfc959.txt>, 2009-05-18
- [47] CAPTCHA, <http://www.phpcaptcha.org>, 2009-05-19

Bilaga A

A1. Hårdvarurekommendationer för användare

- En webbläsare som stöder CSS som används för att designen ska visas korrekt.
- JavaScript är aktiverat för att öppna nya fönster och för captcha, som används för att skicka in kommentarer. JavaScript används även i AJAX för att kunna välja socken och ort vid avancerad sökning.
- Bildskärmsupplösning på minst 1024 x 768.

A2. Stödda format på mediafiler

De olika format som stöds finns nedan tillsammans med en rekommenderad mjukvara för öppna mediafilerna.

Bild

Om mediet är en bild kan man dubbelklicka på bilden och få fram den i fullstorlek.

Format: jpg, gif, png.

Dokument

Om mediet är ett dokument i pdf-format går det att öppna dokumentet direkt ett nytt fönster, annars kan man spara mediet till sin hårddisk och öppna med valfri lokal dokumenthanterare.

Format: doc, pdf.

Rekommenderad mjukvara: [Open Office](#)

Film

För att se ett filmklipp behöver man ladda hem filen och öppna den med sin mediaspelare.

Format: avi, mpg, mpeg.

Rekommenderad mjukvara: [VLC](#)

Ljud

Om mediet är en ljudfil behöver man ladda hem filen innan man kan lyssna på den i sin mediaspelare.

Stödda format: mp3

Rekommenderad mjukvara: [VLC](#)