

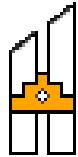
**Arild Sellevold Soltvedt  
Jesro Christoffer Cena  
Åshild Nerhus**

# **Hovudprosjekt**

**EasyPublish**

**10. juni 2005**

**Mediesenteret**



Hovudprosjekt ved Høgskolen i Bergen,  
Avdeling for ingeniørutdanning,  
Linje for data, studieretning dатateknikk

## Forord

---

I sjette og siste semester av den treårige dataingeniørutdanninga ved Høgskolen i Bergen skal alle studentar arbeide med eit hovudprosjekt. Oppgåvene kjem frå eksterne aktørar i næringslivet eller frå høgskulen, og gir oss ei moglegheit til å oppleve korleis livet vil bli etter endt utdanning.

Oppgåva vår er gitt av Mediesenteret og har gått ut på å lage ei enkel publisering løysing som dei skal kunne bruke som ein del av sine framtidige webprosjekt. Vi har hatt arbeidsplass hos Mediesenteret og har arbeidd med oppgåva kvar dag etter påske.

Takk til kontaktpersonane våre ved Mediesenteret, Jan Tore Ynnesdal og Morten Fahlvik, for alltid å vere tilgjengeleg for spørsmål. Ein takk går òg til Mauricio Pavez for grafikk til omslagsarket på rapportane våre og ikkje minst til Hanne Opdal for omfattande testing og arbeid med demosider. Vi må òg takke alle som har vore med å teste systemet vårt, særleg gjengen i Inkubatoren. Til slutt vil vi takke vår kontaktperson ved Høgskolen i Bergen, Svein-Ivar Lillehaug, for konstruktive tilbakemeldingar under prosjektarbeidet.

I samband med prosjektet vårt har vi oppretta ei nettside. Den finn du på:  
<http://www.hovudprosjekt2005.net>

Bergen 10. juni 2005

---

Jesro Christoffer Cena

---

Åshild Nerhus

---

Arild Sellevold Soltvedt

## Innhold

---

Forord.....	
1 Innleiing.....	4
1.1 Organisering av rapporten.....	4
1.2 Oppdragsgjevar.....	4
1.3 Problemstilling.....	4
1.4 Hovuidé for løysingsforslag.....	5
2 Definisjon av oppgåva.....	6
2.1 Kravspesifikasjon.....	6
3 Analyse av problemet.....	8
3.1 Brukstilfelle.....	8
3.2 Databasen.....	9
3.3 Grensesnittet i EasyPublish.....	10
4 Design av moglege løysingar.....	11
4.1 Lage alt frå botnen av.....	11
4.2 Tilpassa ei eksisterande løysing.....	11
4.3 Drøfting av løysingsalternativa.....	11
4.4 Framtidsretta løysing.....	13
5 Vurdering og val av verktøy.....	14
6 Implementering.....	15
6.1 Databasen.....	15
6.2 Klassembibliotek.....	16
6.3 Systemet.....	17
6.4 Webgrensesnittet.....	18
6.5 Sikkerheit.....	18
7 Testing.....	19
7.1 Test av heile systemet.....	19
7.1.1 Test med erfarne brukarar.....	19
7.1.2 Test med ikkje-røynde brukarar.....	20
8 Forklaring av utvikla system i bruk.....	21
8.1 Brukardokumentasjon.....	21
8.1.1 Grensesnitt «Brukar».....	21
8.1.2 Grensesnitt «Administrator».....	23
8.2 Drifts- og vedlikeholdsdocumentasjon.....	24
9 Oppsummering, diskusjon og konklusjonar.....	25
9.1 Evaluering av prosjektarbeidet.....	25
9.1.1 Vanskar vi har møtt.....	25
9.2 Evaluering av systemet.....	26
9.2.1 Kjende svakheiter.....	26
9.2.2 Framtidige forbeteringar.....	26
9.2.3 Kva kunne vi gjort annleis.....	27
Vedlegg.....	

# 1 Innleiing

---

## 1.1 Organisering av rapporten

Vi byrjar rapporten med ein presentasjon av oppdragsgjevar og oppgåva vi har fått tildelt.

Deretter følgjer ei analyse av problemstillinga vi står ovanfor og skisserer så moglege løysingar. Vi presentererer òg løysinga vi valgte til slutt.

Den neste delen omhandlar dei verktya vi har nytta i prosjektarbeidet og måten vi har brukt desse på for å implementere systemet. Etter ein gjennomgang av dei ulike delane i systemet, har vi teke føre oss korleis vi har gjennomført testinga av dette.

Siste del av rapporten tek føre seg korleis det endelige systemet fungerer, før vi rundar av med ei oppsummering av heile prosjektet samt ei evaluering av systemet.

## 1.2 Oppdragsgjevar

Mediesenteret er ein del av Høgskolen i Bergen og dei driv hovudsakleg med medieutvikling og produksjon innan digitale media. Med eit tverrfagleg team, som består av mellom anna designarar, lærarar og systemutviklarar, tilbyr dei design, innhald og metodikk for internett, video og CD-ROM. Særleg ulike former for publisering på internett, webdesign og oppbygging av dynamiske webstadar for ulike typar nett har vore i fokus dei siste åra.

Senteret har òg utvikla og driv ei rekke nye nettbaserte studie som vert tilbodne av Høgskolen i Bergen: Kreativ webdesign, Multimedia for web, Nettpublisering, Kommunikasjonsarbeid og Vidaregåande webdesign.

Mediesenteret held til i Nygårdsgaten 114 og har ti fast tilsette og for tida fire innleigde prosjektleiarar. Sjå <http://www.hib.no/mediesenteret/> for meir informasjon.

## 1.3 Problemstilling

Som nemnd er utvikling av nettstader med dynamisk innhald eit av områda som Mediesenteret har fokusert på dei siste åra. Problemet er at dei i dag har ei publisering løysing som må tilpassast manuelt til kvart prosjekt, noko som omfattar ein del programmering og tek tid.

Det oppdragsgjevarane ser for seg er eit enkelt publiseringstøy for bruk i ulike web-prosjekt, som skal vere enkelt å setje i drift og endre utsjånaden til (vha. CSS). Ein typisk brukar er ein «ikkje-teknisk» person, det vil sei ein person med lita erfaring med data-maskiner. Dette set krav til både utforming, funksjonalitet og ikkje minst språkbruk.

Oppdragsgjevarane har ikkje funne eit system som passar til krava dei har sett for eit slikt webbasert publiseringssystem. Det som eksisterer i dag er enten for avansert, for lite funksjonelt eller kostar pengar. Av dei eksisterande løysingane i dag er det Blogger som ligg næraast det som kan fungere, men det har òg manglar, og brukar eit noko teknisk språk. Løysinga kan byggje på Blogger-prinsippet, men skal altså vere eit fullverdig publiseringssystem og ikkje ein blogg. Eit system som heiter Plone vart òg nemnd, men sidan denne er basert på ein lite utbreidd plattform, Zope, vart dette forkasta.

I dag har Mediesenteret eit bloggsystem tilpassa elevar ved grunnskulen, eLogg, som dei er svært nøgd med, med tanke på tilpassing til ikkje-tekniske brukarar. Oppdragsgjevarane ønskjer at vi vidarefører denne brukarvennlegheita til vårt prosjekt.

## **1.4 Hovudidé for løysingsforslag**

På vårt første møte med Mediesenteret kom det fram at dei hadde ein klar idé for korleis prosjektet vårt skulle sjå ut. Etter møtet fekk vi tilsendt ein kravspesifikasjon der krava til prosjektet var dokumentert. Denne dokumentasjonen har gjort arbeidet vårt mykje enklare, ettersom vi veit kva oppdragsgjevarane vil ha.

Mediesenteret ville ha ei open publiseringsløysing som skulle vere enkel å bruke for ikkje-tekniske personar. Dei sette klare krav for kva som skal kunne gjerast på ulike brukarnivå og har heile vegen vore tilgjengeleg ved spørsmål. Løysinga vår skal ikkje omfatte utsjånaden på publiseringsløysinga, dette skal Mediesenteret lage sjølv.

## 2 Definisjon av oppgåva

---

### 2.1 Kravspesifikasjon

EasyPublish skal vere eit enkelt webpubliseringssystem med eit breitt bruksområde. Målet er at terskelen for å kunne skrive og publisere innhald på nettet skal vere så låg som mogleg med omsyn til «ikkje-tekniske» brukarar. Dette oppnår ein med intuitiv navigering og lett forståeleg språkbruk. Administrasjonen av systemet skal òg vere lettvint.

Andre krav som må oppfyllast er at systemet skal vere basert på gjeldande standardar for HTML/XHTML og CSS. I tillegg skal systemet vere rimeleg og enkelt å setje i drift.

#### Ved hjelp av publiseringssystemet skal ein kunne

- vedlikehalde ein nettstad med dynamiske og statiske sider
- endre utsjånaden berre vha. CSS, utan å endre på HTML-en som systemet genererer
- leggje til mediefiler av ulike slag. Oppdragsgjevarane ønskjer at desse skal bli lagra med unike namn, slik at ein skal kunne laste opp filer utan å tenkje på å endre namna på desse.

Publiseringssystemet skal handtere minst to brukarnivå, ein brukar og ein administrator.

#### Ein brukar skal kunne

- leggje til innlegg og mediefiler
- endre innlegg, statiske sider og personleg informasjon
- slette innlegg
- logge inn eller få tilsendt passord
- søkje på nettsida

#### Administratoren skal i tillegg kunne

- endre innstillingane til sjølve nettstaden
- vise oversikt over brukarar, statiske sider, kategoriar, innlegg, mediefiler
- leggje til statiske sider, kategoriar og brukarar
- endre ein brukar og kategori
- slette mediefil, statisk side, kategori og brukar

For ein vanleg brukar skal systemet vere enkelt å bruke. Det skal vere nær kopling mellom det å lese ei nettside og å endre denne. Dette vil sei at brukaren skal kunne navigere rundt i nettstaden og enkelt endre på ei side ved å trykkje på ei lenkje. Det skal nyttast ein WYSIWYG-editor slik at brukaren ikkje skal måtte hanskast med HTML. Denne editoren skal ikkje vere avhengig av innpluggingsprogram (plugin) av noko slag.

### Tilleggsfunksjonalitet

I tillegg ønskjer oppdragsgjevarane modular som kan leggjast til ved behov. Dei har nemnd funksjonalitet som:

- Kalender
- Diskusjonsforum
- Kommentarar
- Nyheitsabonnement (RSS)

Dette er funksjonar som dei ønskjer å kunne velje frå administrasjonsvindaugen.

Av tilleggsfunksjonane som er nemnde over har vi fått tid til å implementere kommentarar og nyheitsabonnement. Dersom der er ønskeleg kan administrator gjere desse funksjonane tilgjengelege for sin nettstad. Alle som les innlegg på nettstaden skal då kunne kommentere desse, og dersom ein leser vel å abonnere på nyheter vil ein få alle dei nyaste innlegga. Ei oversikt over all funksjonalitet inngår som vedlegg C.

## 3 Analyse av problemet

### 3.1 Brukstilfelle

På figur 3.1 ser vi eksempel på brukstilfelle som inngår i systemet. Desse brukstilfella vil gjelde for ein person som er innlogga som ein brukar. Figuren brukar fargekodar for å vise prioriteten til kvart brukstilfelle. For ein fullstendig oversikt over brukstilfella viser vi til kravspesifikasjonen (vedlegg D).

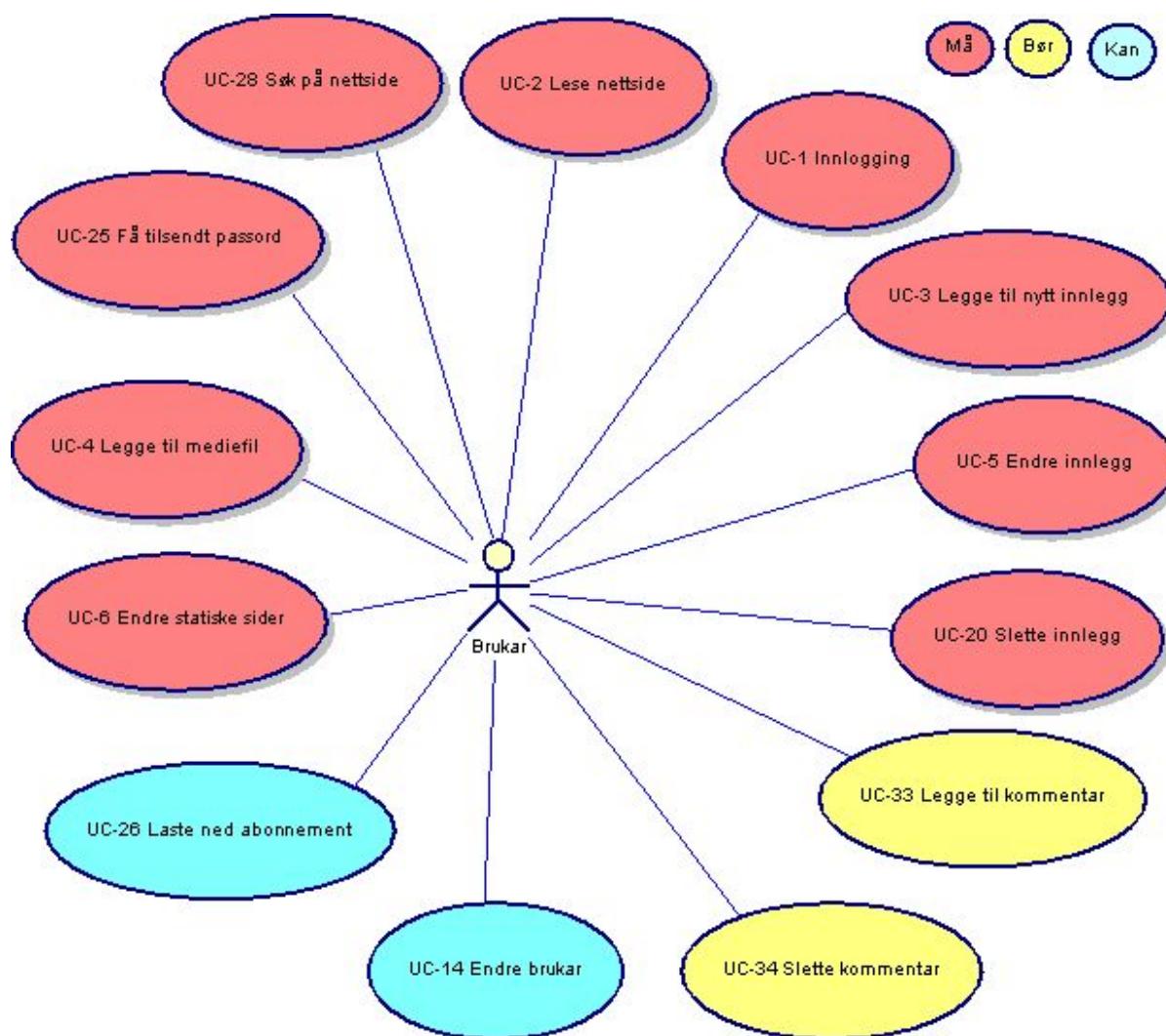


Fig. 3.1: Brukstilfelle for ein brukar

### 3.2 Databasen

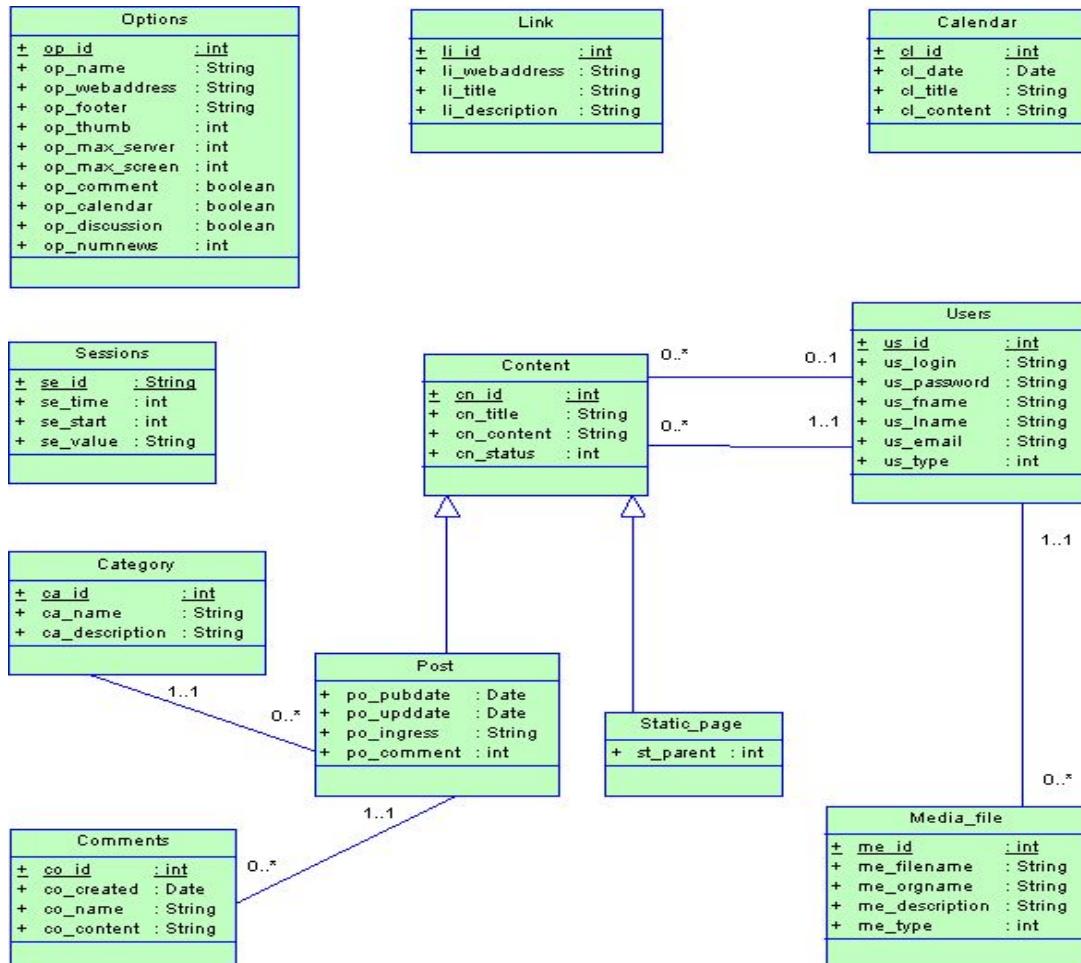


Fig. 3.2: Klassediagram

Databasen i systemet tok utgangspunkt i klassediagrammet vi utforma (sjå fig. 3.2). Denne databasen må kunne handtere hyppig lesing av informasjon, då alt innhaldet i nettstaden vil vere generert ut frå den. EasyPublish vil handtere ein del personlege opplysningar, dette stiller krav til sikkerheita. Dessutan må data behandlast effektivt når ein brukar loggar inn på systemet.

Når ein brukar vert oppretta vil denne få innlagt namn og e-post, samt eit automatisk generert passord. Seinare kan brukaren sjølv endre både passord og e-postadressa, dette har vi lagt til rette for i databasen.

### **3.3 Grensesnittet i EasyPublish**

Ein brukar som er innlogga vil kunna sjå eit ekstra felt på sida der han kan klikke på sitt eige namn og få tilgang til å endre opplysningar om e-post og passord.

Oppbygginga av grensesnittet skal gjerast så enkelt som mogleg for ein ikkje-teknisk person, det vil sei at nokon som har liten erfaring med nettet skal enkelt kunne navigere seg rundt på sida. Dette blir spesielt vist når ein skal leggje til ei mediefil i eit innlegg, dette blir gjort på ein enkel og forklarande måte.

## 4 Design av moglege løysingar

---

Sjølv med dei rammene sett av oppdragsgjevarane, så var det fleire ulike måtar å løyse denne oppgåva på. Vi såg i byrjinga for oss to ulike løysingsalternativ for problemstillinga vi vart presentert for:

1. Å lage alt frå botnen av.
2. Ta utgangspunkt i ei eksisterande webpublisering løysing og tilpasse denne.

### 4.1 Lage alt frå botnen av

Det å utvikle eit system frå grunnen av gir moglegheiter for å kunne lære seg eit programmeringsspråk grundig og implementere løysingar på vår eigen måte. Men det å lage eit system frå grunnen av kan vere svært tidkrevjande og tid er ikkje det vi har mest av i arbeidet med hovudprosjektet.

### 4.2 Tilpasse ei eksisterande løysing

I dag finnест det ei rekke webpubliseringssystem på marknaden. Ein del av desse er utvikla gjennom prinsippet for open kjeldekode, som gjer det mogleg for alle å ta del i vidareutvikling eller vedlikehald av dette. Dermed kan ein vidareutvikle eller tilpasse programvare til eige bruk utan å måtte betale for det. Det einaste kravet er at systemet ein tilpassar òg blir levert som open kjeldekode og dermed tilgjengeleg for andre.

Det å arbeide med eit eksisterande system krev at ein set seg inn i den ferdigutvikla koden for å kunne jobbe vidare med denne, noko som kan vere tidkrevjande. Fordelen er at ein kan jobbe med ferdig, gjennomtesta kode.

### 4.3 Drøfting av løysingsalternativa

Desse to løysingsalternativa representerer to vidt forskjellige måtar å gjennomføre oppgåva på. Begge metodane er tidkrevjande på sine måtar, sjølv om begge har sine fordelar.

Vi fann til slutt ut at det ideelle ville vere å velje eit kompromiss, å finne ein gylden middel-

veg mellom å lage alt frå botnen av og tilpasse eit eksisterande system. Vår løysing vart å lage enkelte delar av systemet frå botnen av og ta andre modular frå open kjeldekode og tilpasse dette til vårt system. Dermed kunne vi dra fordelar av begge løysingsalternativa. Vi har brukt mindre tid på å byggje systemet frå grunnen av, fordi vi har nytta enkelte ferdige modular og vi har brukt mindre tid på å setje oss inn i eksisterande kode fordi det er færre modular å setje seg inn i.

Sjølv om PHP enno ikkje er eit fullverdig objektorientert programmeringsspråk, valde vi å byggje heile systemet ved hjelp av klasser. Med PHP 5 på veg, så meiner vi at eit objektorientert system vil vere meir framtidsretta.

På internett finnест det ei rekke open kjeldekoderessursar for PHP-klasser og dette har vi drege god nytte av. Desse klassene var enkle å implementere og kunne lett brukast som baseklasser for våre eigne klasser. I tillegg har vi nytta eit ferdig malsystem, Smarty, som har vore til stor hjelp i høve til å skilje logikk frå presentasjon. Smarty er eit omfattande malsystem som fint vil kunne henge med i ei vidareutvikling av webpubliseringssystemet.

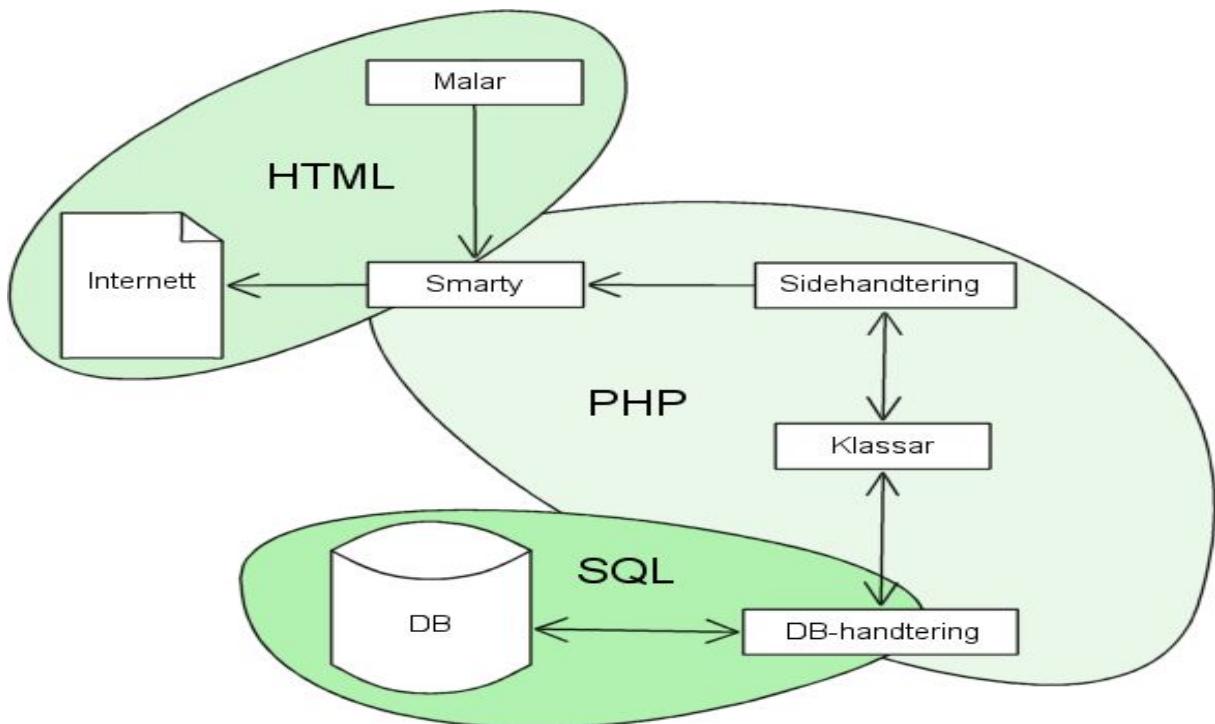


Fig. 4.1: Arkitektur

Figur 4.1 viser korleis alt heng saman i prosjektet vårt. Vi prøver å lage systemet uavhengig av databasetype. Ved å lage ein databaseklasse som handterer all kommunikasjon med databasen, treng ikkje resten av systemet å vite kva databasetype som er i bruk. Denne

databaseklassen pakkar inn data i PHP-objekt, som resten av systemet nyttar. Til slutt blir informasjonen lagt inn i Smarty og malsystemet nyttar passande mal for å generere HTML-sider.

## 4.4 Framtidsretta løysing

Sjølv om det ikkje var eit krav frå oppdragsgjevarane, så har vi gjennom fleire enkle grep prøvd å leggje til rette for ei framtidssretta løysing. Nokre grep har vi allereie nemnd, som bruk av objektorientert programmering, abstraksjon av data og val av malsystem. I tillegg har vi opna for omsetjing av malar frå bokmål til andre språk, ved å samle alle malane i ei eiga mappe.

Kanskje vil ei vidareutvikling av dette systemet dukke opp som ei ny oppgåve frå Mediesenteret seinare?

## 5 Vurdering og val av verktøy

---

Mediesenteret ønskte eit system som var rimeleg i drift og godt støtta av ulike webhotell. Dei ønskten difor at vi skulle bruke PHP/MySQL i arbeidet med prosjektet. Det kom opp eit forslag om bruk av Plone fordi det er enkelt å bruke og vidareutvikle. Plone byggjer på Zope-plattform, noko som få webhotell tilbyr, difor vart ikkje dette noko av. Vi stod fritt til å skrive koden i kva teksteditor vi ville, valet fall på ein gratis PHP-editor, PHP Designer 2005.

Vi valde å bruke CVS for versjonshandtering av filene våre. For dette har vi nytta TortoiseCVS, ein CVS-klient som integrerer seg med «Utforsker» i Windows. Denne er òg fritt tilgjengeleg via internett.

Open kjeldekode er ein ressurs som vi har nytta oss av. Her har vi funne ferdige løysingar, klassar og malsystem, som er gjennomtesta på førehånd.

For å utvikle databasen valde vi å bruke Sybase PowerDesigner, som vi har gode erfaringar med frå tidlegare kurs. Her teikna vi klassediagram og fekk SQL-koden automatisk generert. Vi har òg brukt PowerDesigner for å teikne brukstilfellediagram.

Vi har brukt OpenOffice i staden for Windows sin Office-pakke, fordi den er fritt tilgjengeleg for alle plattformer og vi kan enkelt generere PDF-dokument til bruk på nettsidene våre.

# 6 Implementering

Dette kapittelet omhandlar implementering av systemet vårt med det verktøyet vi har omtala i kap. 5. Etter å ha skrive kravspesifikasjonen og fått denne godkjent av oppdragsgjevarane vart dette grunnlaget for vår implementasjon av webpubliseringssystemet.

## 6.1 Databasen

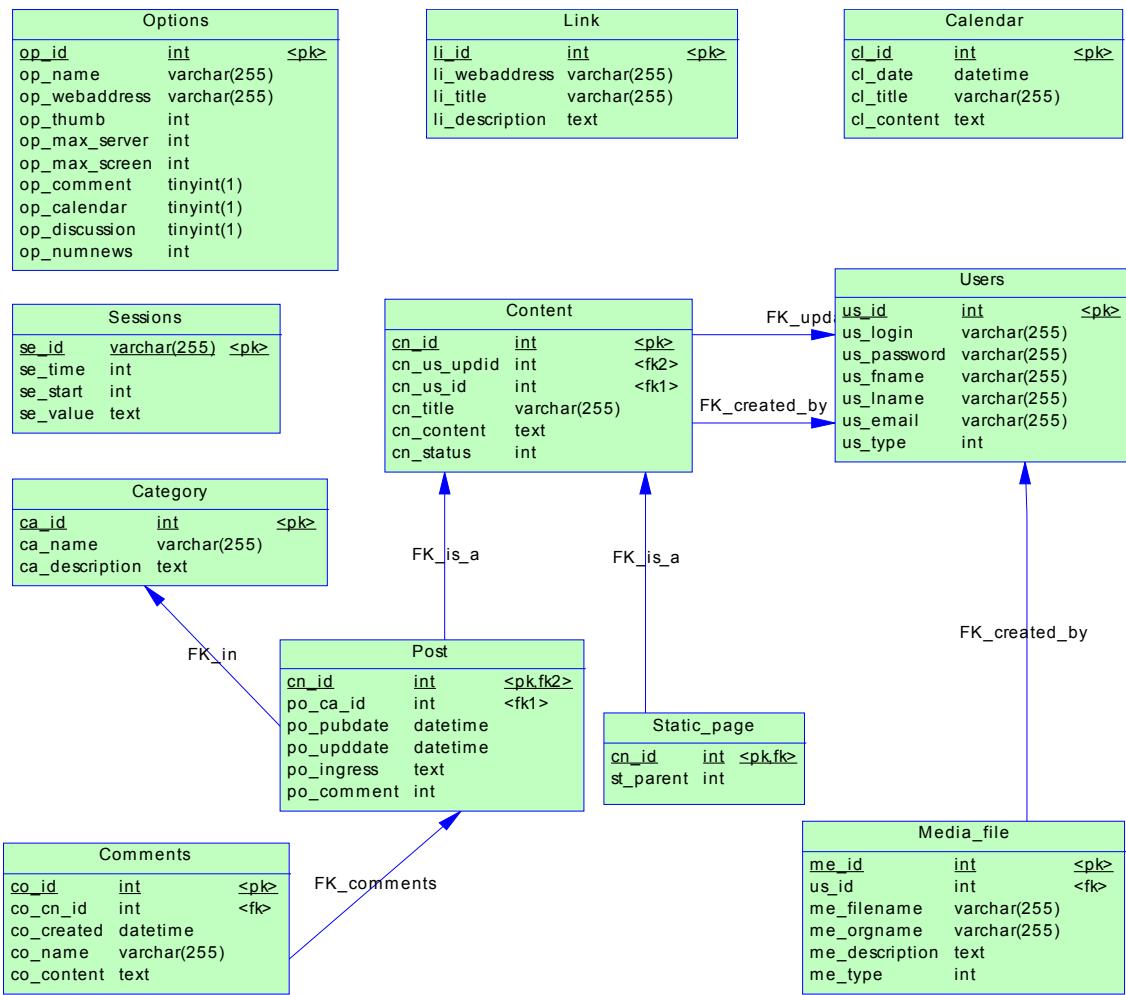


Fig. 6.1: Fysisk datamodell

Databasen var grunnmuren i systemet vårt og det var difor naturleg å byrja prosjektarbeidet med å planlegge denne. Vi nytta PowerDesigner for å lage ein domenemodell som visualiserte dei ulike klassene vi meinte vi fekk bruk for i følgje kravspesifikasjonen. Denne modellen vart omgjort til ein fysisk datamodell, som vi igjen nytta for å generere SQL-kode for MySQL.

Sjølv om alt dette arbeidet vart gjort automatisk i PowerDesigner, måtte vi gjere enkelte endringar manuelt.

Databasen inneheld elleve tabellar som kvar inneheld dei data som objekta inneheld i EasyPublish. Her finst oversikter over kategoriar, statiske sider, brukarar, innlegg, mediefiler og innstillingar. All kommunikasjon med databasen skjer via SQL-spørringar i ein eigen databaseklasse.

Gjennom heile prosjektet har databasen vore gjennom små endringar. Dette fordi vi etter kvart i prosessen har oppdagat nye moglegheiter og behov, og for å gjere systemet meir framtidssrett. Eit døme på ei endring som har kome i ettermiddag er lagring av sesjonsinformasjon i databasen. Den endelege fysiske datamodellen for databasen, som vist på figur 6.1, viser mellom anna korleis brukarinformasjonen heng saman med innhaldet. Denne datamodellen inneheld òg delar som ikkje er blitt implementert i systemet. Dette gjeld kalenderfunksjonen og lenkjeadministrasjon, som er blitt nedprioritert av omsyn til tid.

## 6.2 Klassebibliotek

Etter å ha laga det første utkastet til databasen, byrja vi så smått å lage eit klassebibliotek med utgangspunkt i domenemodellen, klassar som omfattar brukarar, innlegg, statiske sider, innstillingar osb. I tillegg byrja vi å lage ein databaseklasse for å handtere spørringar og leggje resultatet av spørringane inn i objekt. Utanfor databaseklassen eksisterer det ikkje SQL-kode. Vi har òg unngått bruk av MySQL-spesifikke funksjonar, som den innebygde passordkrypteringa for å sikre at systemet er databaseuavhengig. Denne funksjonaliteten er blitt erstatta av ein krypteringsfunksjon i PHP.

I denne fasen såg vi òg etter ferdige klasser i open kjeldekodemiljøet. Vi fann snart ut at nettsida PHP Classes hadde eit omfattande bibliotek av PHP-klasser (Lemos 2005). Av klasser vi har nytta herifrå er:

- Simple DateTime Object

Ein klasse for å handtere datoar. Denne klassen kan mellom anna konvertere mellom fleire datoformat, generere select-boksar i HTML og er skriven av Christian Hansel.

- Resize Class

Denne klassen kan ta bilete av ulike filformat og endre storleiken til desse. Klassen er skriven av Shiege Iseng.

- File Upload Class

Ein klasse skrive for å kontrollere filer og opplasting av desse. Den sjekkar mellom anna filformat, storlek osb. og er skriven av Olaf Lederer.

- Random Password Generating Class

Ein klasse som genererer eit tilfeldig passord med gitt antal bokstavar og tal. Klassen er skriven av Gobinath.

Alle klassene var utgjevne med GNU General Public License (GPL) og kunne difor fritt nyttast og tilpassast til vårt prosjekt. Denne lisensen seier at ein fritt kan endre kjeldekoden til eige bruk, men at, dersom ein ønskjer å utgi denne koden, så må den òg vere underlagd GPL (GPL 1991).

## 6.3 Systemet

All systemlogikk vart delt opp i klassar etter oppgåvetype. All administrasjon vart lagt i ei eiga fil, medan brukaroppgåver og generell lesertilgang vart lagt i ei. Dette gjorde det enklare å skilje mellom rollene og oppgåvene dei skulle utføre.

Utviklinga av sjølve maskineriet byrja med administrasjon av brukarar, for å kunne opprette, slette og vise brukarar, i tillegg til innlogging av desse. På slutten av denne utviklingsfasen testa vi gjennom heile brukaradministrasjonen for å vere sikre på at denne fungerte ordentleg.

Etter kvart som brukaradministrasjonen byrja bli ferdig, tok utviklinga av innhaltsadministrasjon til. Vi byrja først med innlegga eller nyheitene, sidan desse var meir omfattande med meir data å hanskast med, med kategoriar og datoar med meir. Dei statiske sidene kunne ein sjå på som ei forenkla utgåve av innlegg og var dermed enklare å implementere så snart innlegga var i boks.

I løpet av denne fasen, måtte vi òg implementere ein WYSIWYG-editor for redigering av innhaldet. Kravet for WYSIWYG-editoren var at den skulle kunne fungere utan innstikksprogram. Valet fall på ei Javascript-basert løysing og med vår avgrensa kunnskap om

Javascript, byrja vi tidleg å sjå etter ferdige open kjeldekode-løysingar. Oppdragsgjevarane våre hadde tidlegare erfaringar med ein editor, men då vi oppdaga at vedlikehaldet av denne var avslutta, fall valet på TinyMCE. Ein del undersøkingar og omtalar rundt på veven, og det faktum av det framleis var aktivitet (vedlikehald og vidareutvikling) rundt prosjektet, var utslagsgjevande for valet.

Då all innhaldsadministrasjon var implementert kunne vi byrja å leggje inn funksjonalitet knytta til mediefiler. Her byrja den vanskelege jobben, nemleg å få Javascript og PHP til å utveksle data med kvarandre. Sidan Javascript er eit skriptspråk som blir køyrd på klientsida, medan PHP blir køyrd på serversida, var ikkje dette noko som gjekk av seg sjølv. Løysinga vart å køre eit Javascript på klienten som kalla ei PHP-fil på server. Denne PHP-fila utførte oppgåvene og genererte Javascript-kall på serversida som sidan vart køyrd automatisk på klienten. Implementeringa av mediefilhandteringa utgjorde siste del av kjernefunksjonane.

Då all kjernefunksjonaliteten var ferdig, tok vi til på funksjonane som var kategorisert som «bør ha» og «kan ha». Dei funksjonane vi tok oss tid til å implementere var kommentarar, RSS-abonnement.

## 6.4 Webgrensesnittet

For å skilje systemlogikk frå presentasjon, har vi som nemnd i kapittel 4.3, valgt å bruke malsystemet Smarty. Dette har vore til stor hjelp ved at vi lagar ferdige malar i HTML, med datafelt som seinare blir fylt med data. Vi har nytta ein standard som anbefala av W3C, noko som er viktig for at CSS skal fungere ordentleg. Vi valgte å bruke XHTML 1.0 Transitional pga. måten vi har implementert Javascript og fordi TinyMCE genererer XHTML. Mot slutten av prosjektarbeidet, sette vi oss ned saman med ei av dei tilsette ved Mediesenteret for å planlegge kva endringar som måtte til i malane for at rammeverket skulle kunne vere så fleksibelt som mogleg for CSS.

## 6.5 Sikkerheit

Gjennom heile prosjektet har det vore naudsint å tenkje på sikkerheita i systemet. Internett kan gjere deg svært tilgjengeleg, òg for personar som kan gjere skade på systemet. Vi har difor lagt inn enkelte sikkerheitsfunksjonar, som ei eiga klasse som tek seg av handtering av all data lagt inn av brukarar og lagring av sesjonsdata i databasen. Vi har òg lagt inn ein unik iden-

tifikasjonsstreng i alle skjema som blir samanlikna med ein sesjonsvariabel for å redusere misbruk av skjemaskripta våre, til dømes for kommentar-spam.

## 7 Testing

---

Vi har gjennom heile utviklinga testa nye modular som vi har lagt til. Dette har vi gjort for å unngå mykje ekstra arbeid på slutten av prosjektet, då det meste av tida må brukast til rapportskriving. Når vi har testa systemet sjølv er det lett å sjå seg blind på ting fordi ein veit korleis det fungerer. Difor var det viktig for oss å få inn eksterne testpersonar som kunne gå gjennom systemet vårt med nye øye. Samtidig har oppdragsgjevarane våre fått eigne brukarar i systemet slik at dei har hatt moglegheiter til å komme med kommentarar og forslag til endringar.

### 7.1 Test av heile systemet

Som testpersonar har vi både nytta datakyndige og personar som ikkje er like gode i data. Dei datakyndige rapporterte sjølv tilbake til oss, medan vi «overvaka» ikkje-erfarne brukarar samstundes som dei navigerte rundt på sidene våre.

#### 7.1.1 Test med erfarne brukarar

Vi gjorde tidleg ein avtale med ei anna hovudprosjektgruppe om testing av kvarandre sine system. Datakyndige som testar veit kvar eventuelle feil kan ligge og dei vil teste om vi har fått med alle nødvendige sjekkar på data som blir fylt inn i forskjellige skjema. Dei vil òg vite kva som er enkelt å gjere av eventuelle utbetringar og foreslå slike.

#### Testgruppe

Svein Rune Engebakken, Kenneth Korneliussen og Jan-Terje Bræin Sørensen

#### Tid på testing

1,5 timer

#### Forarbeid

Før testinga oppretta vi ein brukar for kvar av testpersonane. Dermed fekk dei tilsendt ein e-post med brukarnamn og passord, samt adresse til innloggingssida. Dessutan fekk testgruppa utdelt kvar sitt skjema (sjå vedlegg E) med oppgåver dei skulle teste.

## Prosess

Testpersonane startar med å lese e-posten systemet vårt har sendt, der brukarnamn og passord står. Deretter loggar dei inn på sida for å endre passord. Til slutt går dei gjennom alle oppgåvane vi har gitt dei, og for kvar oppgåve prøver dei alle alternative vegen for å sjå om det er noko vi har gløymt å ta omsyn til.

## Tilbakemeldingar

Etter testen fekk vi tilbake skjemaet som testpersonane hadde fylt ut, slik at vi kunne utbetre dei feila som oppstod. Testgruppa hadde kome opp med ein god del konstruktiv kritikk, og vi fekk retta opp i det meste. Vi ser at det var svært nyttig med ein slik test utført av utanforståande.

### 7.1.2 Test med ikkje-røynde brukarar

For å teste om systemet er så enkelt som oppdragsgjevarane vil ha det har vi òg testa det på personar som ikkje har så mykje erfaring med bruk av datamaskiner. Dette er jo ei viktig gruppe å teste systemet med, sidan systemet skal vere laga for denne typen brukarar.

For å gjennomføre testar med ikkje-erfarne brukarar, jobba vi individuelt saman med testpersonane. Vi sat i bakgrunnen og gav testpersonen enkle oppgåvar i forma:

- «Prøv å logge deg inn.»
- «No skal du prøve å skrive eit innlegg»
- osb.

Etter kvart som personen navigerte rundt, kunne vi sjå korleis dei brukte systemet og om det var ting som verka vanskeleg å forstå. Då kunne vi spørje kvifor dei ikkje skjønte korleis ting skulle gjerast. Dette gav oss nyttige opplysningar om korleis slike brukarar tenkjer og kva som kunne betrast. Mellom dei forbetingane som kom fram under testane med desse brukararane, var at vi måtte skrive betre hjelpetekstar, samt leggje til hjelpetekst på ein del lenkjer.

## 8 Forklaring av utvikla system i bruk

### 8.1 Brukardokumentasjon

EasyPublish har to webgrensesnitt for å vise informasjon på sidene. Det er eitt grensesnitt for ein vanleg brukar, og eitt for administrator. Ein brukar har avgrensa moglegheiter i systemet, medan ein administrator har moglegheiten til å vise og gjere endringar i alt innhold på nettstaden. Dette blir styrt via eit eige administrasjonsvindauge.

#### 8.1.1 Grensesnitt «Brukar»

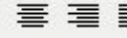
Tittel:

Innhold:

Her kan ein skrive nett det ein vil.

- Lage lister
- Lage [lenkjer](#) og *formatere tekst*.
- Og legge inn bilde



**F K U ABC |**  **-- Stiler --**  Brødtekst 

          **HTML**

Status:  

Plasseres i:  

**Lagre side**

Fig. 8.1: Grensesnitt for å skrive ei nettside

Brukarane skal som nemnd ha eit enkelt grensesnitt. Som vist i figur 8.1, har vi lagt til ein WYSIWYG-editor, for å gjere formatering av innhald lettare. Symbola vil vere kjende for dei som har nytta tekstbehandlingsprogram.

<b>Kva vil du gjere</b>	<b>Korleis få det til</b>
Lese innlegg	Trykk på overskrifta til innlegget
Lese ei statisk side	Trykk på sidenamnet i menyen
Lese gamle innlegg	Trykk på «arkiv» i menyen
Lese innlegg etter kategori	Trykk på kategori i innlegget
Logge inn	Trykk på «Logg inn» i menyen
Er eg innlogga?	Du er innlogga viss du får opp eit ekstra felt med ditt eige namn øvst på sida.
Endre passord	Trykk på namnet ditt ifeltet øvst på sida og vel deretter «endre passord»
Endre e-postadresse	Trykk på namnet ditt ifeltet øvst på sida og eit skjema med dine personalia dukkar opp.
Legg til kommentar	Trykk på overskrifta til innlegget du vil kommentere, om det er mogleg å kommentere innlegget blir det vist eit kommentarfelt nedst på sida.
Slette kommentar	For kvar kommentar finst ein sletteknapp/ikon, den blir vist dersom du er innlogga.
Tillate kommentarer til innlegget mitt	Når du opprettar eit innlegg vert det vist ein sjekkboks nedst for å tillate kommentarar til innlegget, merk av denne boksen.
Legge til eit innlegg	Gå til hovudsida og vel «nytt innlegg».
Gå til hovudsida	Trykk på ikon/lenkje med nettstadens namn.
Slette innlegg	Om du er innlogga kan du slette eit innlegg, trykk på overskrifta til innlegget, og vel deretter å endre innlegget. I botnen blir det no vist ein sletteknapp/ikon.
Endre statisk side	Vel den sida du vil endre på frå meny/framsida og vel deretter «endre»
Gløymt passord	Gå til hovudsida og vel «logg inn» frå menyen, deretter vel du «gløymt passord»

## 8.1.2 Grensesnitt «Administrator»

The screenshot shows the administrator interface for 'EasyPublish - Demo'. At the top, a black bar displays the message 'Du er logget inn som Herr Administrator'. Below this, the title 'EasyPublish - Demo' is shown in green. A sub-header states 'Demonstrasjon av et enkelt webpubliseringssystem.' On the left, a section titled 'Innstillinger' (Settings) contains several input fields: 'Nettsidenavn' (Website name) set to 'EasyPublish - Demo'; 'Slagord' (Keywords) set to 'Demonstrasjon av et enkelt webpubliseringssystem'; 'Nettadresse' (Website address) set to 'http://www.hovudprosjekt2005.net/easypubli...'; 'Antall nyheter på framsida' (Number of news items on the front page) set to '10'; and 'Størrelse, miniatyrbilde' (Size, thumbnail) set to '100'. To the right of these fields is a search bar with a 'Søk' (Search) button. Further down the right side is a vertical navigation menu under the heading 'Administrasjon' (Administration), listing 'Nyheter', 'Kategorier', 'Nettsider', 'Mediefiler', 'Innstillinger', and 'Brukere'. At the bottom of the interface is a link 'Arkiv' (Archive).

Fig. 8.2: Innstillinger i administrasjonsgrensesnittet

Ein administrator vil i tillegg til oppgåvane ein brukar kan utføre, få tilgang til ei rekke administrative funksjonar. Figur 8.2 viser eit utsnitt av innstillingsvindaugen samst administrasjonsmenyen. Administratoren kan utføre følgjande:

Kva vil du gjere	Korleis få det til
Er eg innlogga?	Du er innlogga viss du har same felt øvst på sida som ein vanleg brukar, og ein ekstra administrasjonsmeny
Endre innstillingar for nettstaden	I administrasjonsmenyen vel du «innstillingar», og ein oversikt over alle innstillingane du kan endre dukkar opp
Slette ein brukar	Vel «brukarar» frå administrasjonsmenyen og trykk slett ved sida av den brukaren du vil slette
Endre ein brukar	Vel «brukarar» frå administrasjonsmenyen og trykk endre ved sida av den brukaren du vil endre informasjonen om

Kva vil du gjere	Korleis få det til
Opprette ein ny brukar	Vel «brukarar» frå administrasjonsmenyen og trykk «ny brukar» øvst i oversikten over brukarar
Opprette ein ny kategori	Vel «kategoriar» frå administrasjonsmenyen og trykk «ny kategori» øvst i oversikten over kategoriar
Slette ein kategori	Vel kategoriar frå administrasjonsmenyen og trykk slett ved sida av den kategorien du vil slette
Opprette ei ny statisk side	Vel «statiske sider» frå administrasjonsmenyen og trykk «ny side» øvst i oversikten over statiske sider
Slette statisk side	Vel «statiske sider» frå administrasjonsmenyen og trykk «slett» ved sida av den statiske sida du vil slette
Slette mediefil	Vel «mediefiler» frå administrasjonsmenyen og trykk «slett» ved sida av den mediefila du vil slette
Endre kategori	Vel «kategoriar» frå administrasjonsmenyen og trykk «endre» ved sida av den kategorien du vil endre namn på
Søkje etter innhald	Gå til hovudsida og skriv inn søkjeord i søkerfeltet

## 8.2 Drifts- og vedlikeholds dokumentasjon

Etter kvart som fleire brukarar kjem til i systemet og legg inn fleire innlegg som igjen inneholder ei rekkje mediefiler blir det raskt mykje data i databasen. Det hender jo også at ein brukar blir deaktivert, då ligg informasjonen om dei framleis i databasen og tar opp plass. Rutinar for å vedlikehalde databasen ligg ikkje innanfor definisjonsområdet til prosjektet og det vil vere naturleg at systemansvarleg har oversikt over informasjonen i databasen.

# 9 Oppsummering, diskusjon og konklusjonar

---

## 9.1 Evaluering av prosjektarbeidet

Vi starta prosjektarbeidet med nøye planlegging av heile perioden og systemet som skulle implementerast. Mellom anna skreiv vi ein detaljert kravspesifikasjon som tok utgangspunkt i kravspesifikasjonen som er lagt ved. Denne har seinare vore eit godt hjelpemiddel for modultesting. På slutten av kvar veke har vi sett av tid for å planlegge heile neste veke, basert på Gantt-diagrammet vårt (vedlegg A). Denne planen har vi stort sett klart å halde oss til.

Vi føler at vi har klart å planlegge implementeringa av webpubliseringssystemet på ein fornuftig måte, og at gjennomføringa har gått rimeleg greitt. Frå starten av har vi hatt ein god struktur på filer og mapper som har vore tilnærma uendra gjennom heile prosjektperioden. Vi har i store periodar vore framfor tidskjema i høve til planane, men vi visste at vi trengte all den tid vi kunne opparbeide oss etter kvart som eksamensperioden nærma seg.

Mot slutten har vi fått lagt til ein del av tilleggsfunksjonaliteten, i tillegg til eit par endringar som oppdragsgjevar har bede om i siste augneblink.

Arbeidet med dette prosjektet har vore svært lærerikt og har gitt oss erfaringar som vanskeleg kunne blitt gitt oss gjennom forelesningar. Det å gå gjennom heile prosessen frå å møte oppdragsgjevarar via planlegging og implementering av prosjektet til å presentere eit ferdig produkt for dei som har gitt oss oppgåva, er noko vi trur vi vil få god nytte av når vi seinare skal ut i arbeidslivet.

### 9.1.1 Vanskar vi har møtt

Før prosjektet var kome i gong sette vi opp ei risikoliste (vedlegg B) over ting som kunne forseinkje arbeidet eller i verste fall velte heile prosjektet. Nokre av desse risikoane slo til og vi måtte omstrukturere prosjektet litt. Vi har mellom anna hatt sjukdom blant medlemmane i gruppa, både over kortare og lengre periodar, noko som har ført til ei omstrukturering av framdriftsplanane. I tillegg har testserveren vår vore ute av drift i periodar. Dette har vi løyst ved å bruke ein reserveserver. Ein halv dag stod vi utan både testserver og reserveserver, og vi var nøydd til å utføre modultesting lokalt på ei av arbeidsmaskinene, noko som seinka

arbeidstempoet betrakteleg, då berre to av oss kunne jobbe effektivt. Det ordna seg imidlertid etter nokre timer med reserveservaren, slik at vi fekk fortsett arbeidet vårt, litt lenger tid skulle det gå med testserveren som kom opp igjen etter nokre dagar.

## **9.2 Evaluering av systemet**

Ut frå den kravspesifikasjonen vi fekk frå Mediesenteret har vi implementert den funksjonaliteten som utgjorde minstekravet til systemet, altså kjernefunksjonaliteten. I tillegg har vi fått oppfylt enkelte av krava til tilleggsfunksjonalitet. Oppdragsgjevarane våre har, saman med resten av Mediesenteret, fått ein demonstrasjon av kva ein kan gjere med systemet, og vi har kun fått positive tilbakemeldingar hittil. Det vi står med, mot slutten av hovudprosjekt-perioden, er eit fullt fungerande publiseringssystem som allereie er planlagt brukt i oppdrag som Mediesenteret skal utføre.

### **9.2.1 Kjende svakheiter**

Sjølv om vi er svært nøgde med det arbeidet vi har utført til no, finst det ting som vi gjennom testar har oppdaga ikkje fungerer som dei skal.

Handtering av mediefilene fungerer ikkje heilt som det skal for enkelte biletformat. Dette gjeld bilete som bmp-filer, animerte og transparente gif-filer. Dette er fordi vi nyttar grafikkbiblioteket GD for bilethandsaming. Vi kunne nytta ImageMagick, som er eit kraftigare verkty for biletmanipulering, men dette er, av sikkerheitsgrunnar, ikkje like tilgjengeleg som GD i webhotell som leverer løysingar på delte vertar. Dessutan har vi ikkje laga automatisk generering av HTML for å vise Flash-filer som ein del av innhaldet.

Automatisk utlogging er ikkje heilt konsekvent, noko som kjem av måten sesjonsvariablane blir handsama i databasen.

WYSIWYG-editoren fungerer ikkje heilt som ønskja. Mellom anna har vi opplevd enkelte småproblem i høve til visning av innhald i editoren. Den fungerer ikkje i nettlesarar som ikkje støttar Javascript og editoren hindrar oss òg i å servere XHTML-sidene som anbefala av standardane.

### **9.2.2 Framtidige forbetringar**

Resultatet må vi sei oss nøgde med sjølv om det er moglegheiter for forbetringar og utvidingar i framtida. Det kan nemnast at vi ikkje vart heilt ferdige med «kjekt-å-ha» funksjonane som vi

skulle implementere dersom tida strekte til. Dette er funksjonar som det å opprette eit diskusjonsforum, lage ein kalender med dagens hendingar og lenkjeadministrasjon.

Funksjonalitet som vi gjerne skulle hatt med i prosjektet vårt er kalender og diskusjonsforum. Dette hadde i våre auge gjort webpubliseringssystemet meir komplett.

### **9.2.3 Kva kunne vi gjort annleis**

No som vi ser tilbake på arbeidet vi har gjort, har vi merka oss ting vi kunne gjort annleis.

Organisering av koden kunne vore noko annleis, ettersom enkelte klasser er blitt ganske store. Dette gjeld spesielt klassen som styrer logikken for ein lesar og ein innlogga brukar. Denne kunne med fordel blitt delt opp for å betre oversikta.

Vi skulle gjerne brukt meir tid på å optimalisere spørringane til databasen. Det er potensiale for å få ein mykje meir effektiv datahandtering ved å gå gjennom spørringane.

Testing er noko ein alltid kan gjere meir av. Særleg det å setje oss ned saman med fleire ikkje-erfarne brukarar for å sjå korleis dei arbeider med systemet viste seg å vere svært nyttig og vi skulle gjerne ha gjort testar med fleire personar.

## Litteraturliste

---

**Ambler, Scott W. og Larry L. Constantine (2000)**

*The Unified Process Inception Phase: Best Practices in Implementing the UP*  
**Lawrence(CMP Books)**

**DuBois, Paul, Stefan Hinz, Mike Hillyer, Jon Stephens og Russell Dyer**

*MySQL Manual*  
<http://dev.mysql.com/doc/mysql/en/index.html>

**Flanagan, David (1998)**

*Javascript, The Definitive Guide, 3<sup>rd</sup> edition*  
**O'Reilly**

**Free Software Foundation (1991)**

*GNU General Public License*  
<http://www.gnu.org/copyleft/gpl.html>

**Hojtsy, Gabor (redaktør) (2005)**

*PHP Manual*  
<http://no.php.net/manual/en/index.php>

**Lemos, Manuel (2005)**

*PHP Classes Repository*  
<http://www.phpclasses.org/>

**Lunn, Ken (2003)**

*Software Development with UML*  
**New York (Palgrave Macmillan)**

**Mediesenteret**

*Om Mediesenteret*  
<http://www.hib.no/mediesenter/>

**Ohrt, Monte og Andrei Zmievsk (2005)**

*Smarty Manual*

<http://smarty.php.net/manual/en/>

**Shiflett, Chris (2004)**

*PHP Security Workbook*

<http://shiflett.org/php-security.pdf>

**Welling, Luke og Laura Thomson (2003)**

*PHP and MySQL WebDevelopment.*

**Indianapolis(Sams Publishing)**

## Stikkordregister

---

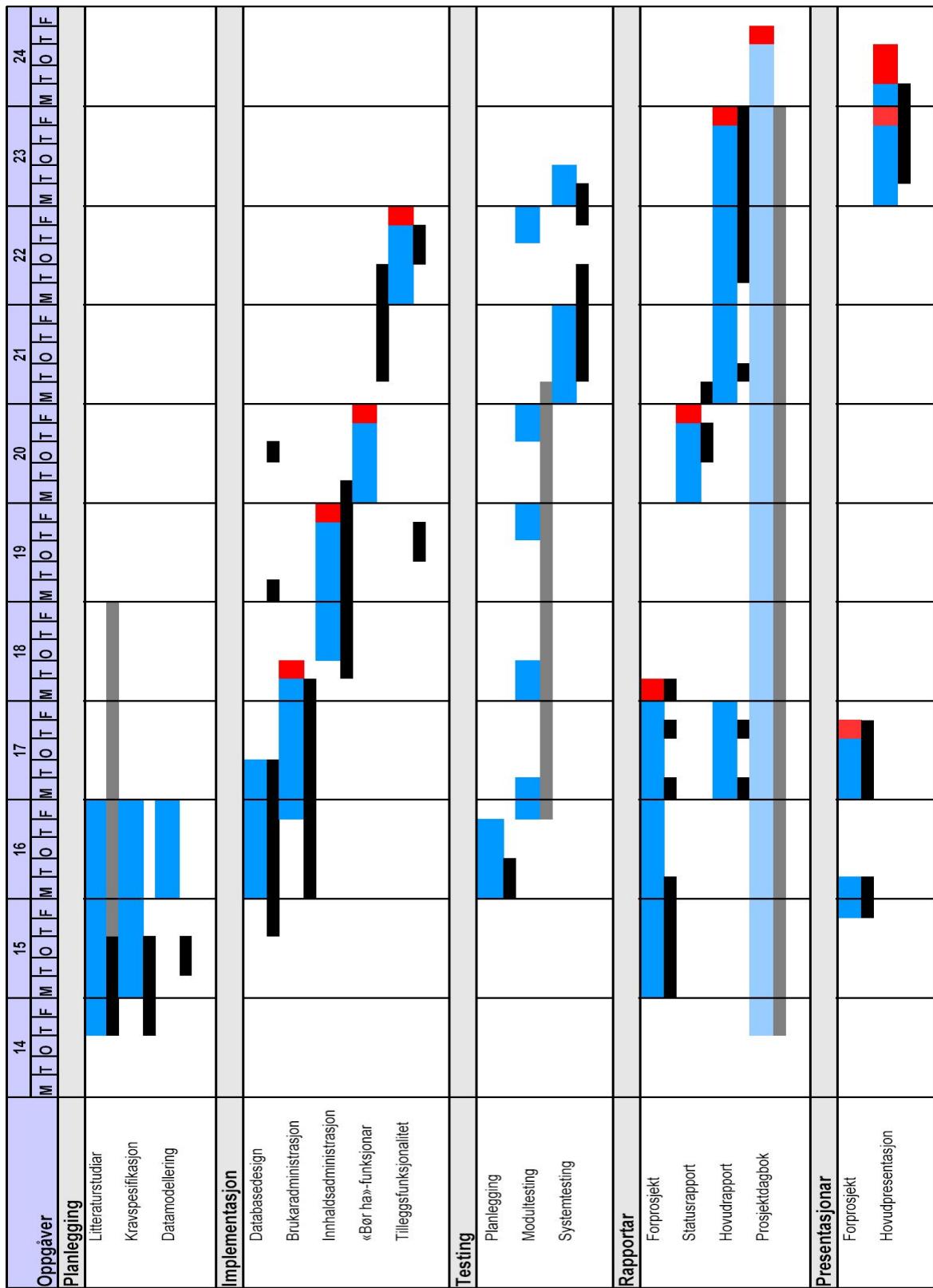
administrator	<b>6</b>
arkitektur	<b>12</b>
brukar	<b>6</b>
brukstilfelle	<b>8</b>
database	<b>9</b>
EasyPublish	<b>6</b>
GD	<b>25</b>
General Public License	<b>17</b>
grensesnitt	<b>10, 21</b>
ImageMagick	<b>25</b>
innhaltsadministrasjon	<b>18</b>
klassebibliotek	<b>16</b>
løysingsalternativ	<b>11</b>
Mediesenteret	<b>4</b>
open kjeldekode	<b>14</b>
Smarty	<b>18</b>
systemlogikk	<b>17</b>
tilleggsfunksjon	<b>7</b>
TinyMCE	<b>18</b>
webgrensesnitt	<b>21</b>
WYSIWYG	<b>17</b>

## Vedlegg

---

- A) Framdriftsplan
- B) Risikoliste
- C) «Løkdiagram»
- D) Kravspesifikasjon
- E) Testskjema
- F) Ulike eksempel på grensesnitt

# Framdriftsplan

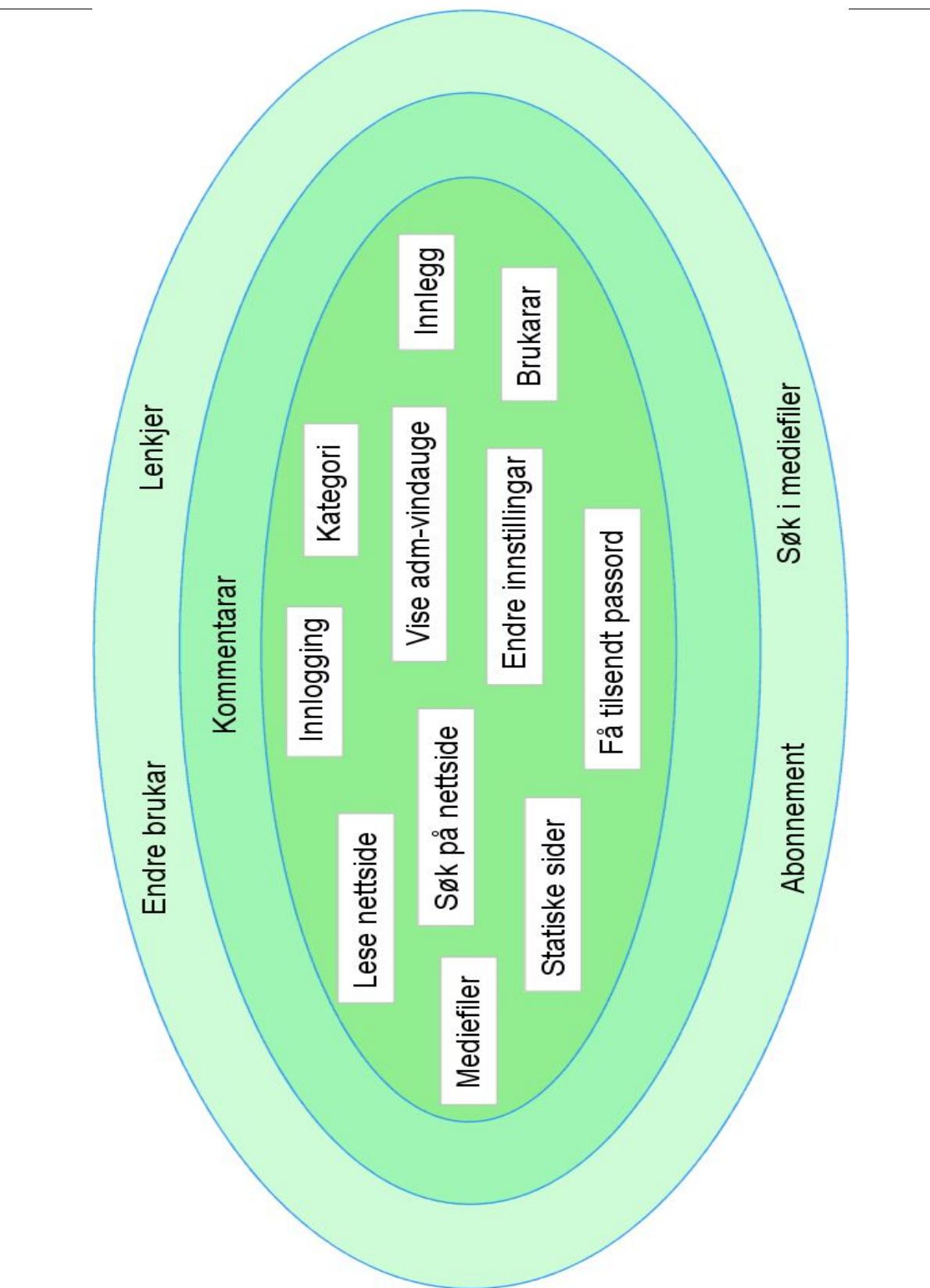


# Risikolist

ID	Risiko	Sannsyn	Alvor	Konsekvens	Tiltak
R-1	Ikkje ferdig i tide	Lav	Høg	Ikkje bestått	<ul style="list-style-type: none"> <li>• Må planlegge godt, og sette av nok tid til alle oppgåver</li> <li>• Eventuelt avgrense ved tidsnød</li> </ul>
R-2	Ikkje ferdig til milepælar	Høg	Middels	Må fortsette arbeidet ut i neste periode	<ul style="list-style-type: none"> <li>• Sette av god nok tid til alle fasar i planlegginga</li> <li>• Arbeide ekstra</li> </ul>
R-3	Produktet fungerer ikkje som forventa	Middels	Høg	Misnøgd oppdragsgjevar	<ul style="list-style-type: none"> <li>• Sette av god tid til <ul style="list-style-type: none"> <li>• Kravspesifikasjon</li> <li>• Testing</li> </ul> </li> </ul>
R-4	Får ikkje testa godt nok	Høg	Middels	Får ikkje luka ut alle feila	<ul style="list-style-type: none"> <li>• Må sette av rikeleg tid til testing</li> <li>• Planlegge testinga godt</li> </ul>
R-5	Stort fråvere	Lav	Høg	På etterskudd med arbeidet	<ul style="list-style-type: none"> <li>• Alle må helde seg friske</li> <li>• Omdisponeringar</li> </ul>
R-6	Konflikt i prosjektgruppa	Middels	Høg	Dårleg arbeidsmiljø og lite effektiv jobbing	<ul style="list-style-type: none"> <li>• Må bruke tid til å diskutere når vi er usamde</li> <li>• Komme fram til demokratiske slutningar</li> </ul>

ID	Risiko	Sannsyn	Alvor	Konsekvens	Tiltak
R-7	Enkeltpersonar «melder seg ut», gjer ikkje arbeidet sitt	Lav	Høg	Kjem på etterskudd med arbeidet	<ul style="list-style-type: none"> <li>● Passe på at alle føler seg viktige</li> </ul>
R-8	Har sett for høge mål	Lav	Middels	Får ikkje gjort alt	<ul style="list-style-type: none"> <li>● Må sette realistiske mål for produktet</li> <li>● Gode tidsestimat</li> </ul>
R-9	Sit fast i programmeringa	Høg	Middels	Kjem ikkje vidare, får ikkje gjort det ein ønskjer	<ul style="list-style-type: none"> <li>● Må lese seg opp i alle felt før programmeringa startar</li> <li>● Spørje om hjelp</li> </ul>
R-10	Tap av data	Lav	Høg	Må gjere ting på nytt	<ul style="list-style-type: none"> <li>● Alltid ha backup av alle data</li> </ul>
R-11	Webserver ikkje tilgjengeleg	Lav	Høg	Får ikkje tilgang til database og får ikkje testa	<ul style="list-style-type: none"> <li>● Kjøre server lokalt på eiga maskin</li> </ul>
R-12	Serveren med all dokumentasjonen krasjer	Lav	Middels	Mykje arbeid blir bortkasta.	<ul style="list-style-type: none"> <li>● Alle har lokale kopiar av dokumentasjonen til ei kvar tid.</li> </ul>

# Funksjonalitet



# Ulike eksempel på grensesnitt

[EasyPublish - Demo](#)

Demonstrasjon av et enkelt webpubliseringssystem.

**Velkommen!**

EasyPublish håper du får et hyggelig opphold. Logg deg gjerne inn for å skrive et innlegg.

[Om sidene](#)  
[Arkiv](#)  
[Logg inn](#)

**Nyheter**

[Enda eit innlegg](#)  
Generell 09.06.2005 23:41  
Dette er innlegget som skal nyttast under demoen på skulen.

[Les mer](#)

**MUSIKKNETT VEST**

**Universiteter og Høgskoler**

- Griegakademiet
- Høgskolen i Bergen
- Universitetet i Bergen
- Høgskolen i Stord og Haugland
- Ole Bull Akademiet

**Musikkarkiver**

- ABS
- FIR
- Sogn
- OBA
- HARD
- Div. Muséer
- Griegskolen?

**Muséer**

- Trolldhaugen
- Siljustol
- Løseen
- Fargein Valen

**FORSKNING OG KULTUR**

**STUDIER**

**TEKST**

**AKTIVITETER**

[Logg inn](#)

**rv 555** rv555 - demoside  
i kampen for å berge liv på vegene...

Trist som faen...  
Wish U T best!  
svare sluttet

**Nyheter**

**Strengere regler for gutter enn jenter?**  
Nyhet 07.06.2005 14:08  
"Politiadvokat Kjetil Kvalvik luftar tanken om at det kanskje burde vere eigne reglar for førerkort for menn mellom 18 og 25 år. Han

fart  
ungdomsulykker  
rus  
gutter og jenter  
savn  
Arkiv  
Logg inn