

HOVEDPROSJEKT:

TITTEL

mPetre.Community

FORFATTER(E):

ERLEND LANS PEDERSEN

KJETIL ORMESTAD EID

TOM IVAR ØVERLIE

Dato: 22 mai 2002

SAMMENDRAG AV HOVEDPROSJEKT

Tittel:	<u>mPetre.community</u>	Nr. :	
		Dato :	22.05.02
Deltaker(e):	<u>Erlend Lans Pedersen</u> <u>Kjetil Ormestad Eid</u> <u>Tom Ivar Øverlie</u>		
Veileder(e):	<u>Pål Børre Sveum</u>		
Oppdragsgiver:	<u>NRK Upunkt</u>		
Kontaktperson:	<u>Nina Nordbø og Geir Inge Løvstad</u>		
Stikkord (4 stk)	<u>Hva spilles nå? Community, Portal, Chat</u>		
Antall sider: 58	Antall bilag: 29	Tilgjengelighet (åpen/konfidensiell): Åpen	
Kort beskrivelse av hovedprosjektet:			
<p>Vår hovedprosjekt har fått navnet mPetre Community. Det har i korthet gått ut på å lage et samfunn for radiokalen mPetre på internett. Her skal brukere kunne registrere seg for å medlem av dette samfunnet, og på den måte treffe andre som også gjør det samme. De har mulighet til å chatte med hverandre, delta på konkurranser og søke etter sanger og artister. Vi ville også øke kommunikasjonen mellom programledere på kanalen og medlemmer av samfunnet. En annen viktig del av oppgaven var å lage en "hva spilles nå"-funksjon. Dette er en flash-animasjon som hele tiden står og viser hva som til et hvert tidspunkt spilles på mPetre. Dette lagres i en database, slik at brukere skal kunne søke i denne etter sanger man leter etter og/eller vil vite mer om. Det ble også vår oppgave å designe hele looken til dette samfunnet. Designet de hadde var veldig mørkt og dystert, og de ville gjøre om dette slik at det passet bedre til målgruppen for kanalens lyttere. Vi lagde også et adminsystem til de som skal legge inn nyheter, konkurranser, spillelister o.l. Systemet de hadde for å legge inn spillelister fra uke til uke var veldig tungvint, slik at en forbedring av dette var ønskelig.</p>			
URL: www6.nrk.no/magasin/upunkt/mpetre			

NRK mPE3RE.
COMMUNITY

HOVEDPROSJEKT VÅREN
2002

Forord

Denne rapporten er utarbeidet som et avsluttende hovedprosjekt for en treårig grafisk ingeniørutdanning. Prosjektet vekter 60 % av siste semester, og er obligatorisk for alle avgangsstudentene ved høyskolen i Gjøvik. Hensikten med oppgaven har vært å få praktisert fagkunnskaper vi har tilegnet oss i løpet av studietiden.

Bakgrunnen for valget av dette prosjektet er at vi ønsket å lære mer om å utvikle multimedia-løsninger på internett. Vi bestemte oss tidlig i prosessen for å lage et multimedia produkt og henvendte oss til bedrifter og institusjoner som driver med dette. Første ledd i prosessen med å skaffe et spennende prosjekt, var at vi utviklet en hjemmeside hvor vi presanerte oss selv, og det vi kunne tenke oss å jobbe med. Deretter skrev vi en e-post til de aktuelle bedriftene hvor vi henviste til denne hjemmesiden (<http://artific.net>). Vi fikk mange positive tilbakemeldinger, men et samarbeid med NRK virket ekstra spennende.

Vi ønsker å takke:

Vår oppdragsgiver oppdragsgiver NRK Upunkt. Spesielt Geir Inge Løvstad, som vi har hatt tett kontakt med gjennom hele prosessen.

Redaksjonen i mPetre

Veileder Pål Børre Sveum

Byte AS, for lån av serverplass.

Andre støttespillere i NRK

Selve arbeidet med produktet har funnet sted i perioden fra 1.januar til 23.mai 2002.

Gjøvik 22.05.02

Kjetil O. Eid
Pedersen

Tom Ivar Øverlie

Erlend Lans

Innholdsfortegnelse

1. Innledning.....	4
1.1 Organisering av rapporten.....	4
1.2 Oppgavedefinisjon.....	5
1.2.1 Problemstilling.....	5
1.2.2 Resultatmål.....	5
1.2.3 Gruppas effektmål.....	5
1.2.4 mPetres effektmål.....	5
1.2.5 Oppgavebeskrivelse.....	5
1.2.6 Avgrensning.....	6
1.3 Målgruppe for rapporten.....	6
1.4 Målgruppe for siden.....	6
1.5 Brukere av siden.....	7
1.6 Arbeidsformer.....	7
1.7 Terminologi.....	8
2. Bakgrunn for prosjektet.....	9
2.1 Gruppas bakgrunn.....	9
2.2 Oppdragsgivers bakgrunn.....	9
2.3 Studentenes faglige bakgrunn.....	9
3. Prosjektgjennomføring.....	11
3.1 Praktisk framgangsmåte.....	11
3.1.1 Oppstartsfasen.....	11
3.1.2 Forprosjekt.....	11
3.1.3 Egenlæring.....	11
3.1.4 Utviklingsfasen.....	11
3.1.5 Sammenstilling av rapport.....	12
3.1.6 Planlegging av framføring.....	12
3.1.7 Organisering av kvalitetssikring.....	12
3.2 Kommunikasjon.....	12
3.2.1 Framdriftsplan.....	13
4. Prinsipper og teori.....	14
4.1 Innføring i HTML.....	14
4.1.1 Innføring i Javascript.....	14
4.1.2 Innføring i CSS.....	14
4.2 Innføring i nettlesere.....	15
4.3 Innføring i XML.....	16

4.4	Innføring i databaser.....	17
4.4.1	Hva er en database.....	17
4.4.2	Databaseteori.....	17
4.5	Innføring i SQL.....	17
4.6	Innføring i PHP.....	18
4.7	Innføring i Macromedia Flash.....	18
4.7.1	Innføring i Flash templates.....	18
4.8	Innføring i Swift-tools.....	18
4.9	Innføring i Imagemagick.....	18
4.10	Innføring i webservere.....	18
5.	Utstyr.....	19
5.1	Datamaskiner.....	19
5.2	Diverse.....	19
5.3	Utviklingsprogramvare.....	20
5.3.1	Adobe Photoshop.....	20
5.3.2	Macromedia Flash 5.0.....	21
5.3.3	Macromedia Dreamweaver 4.0.....	22
5.3.4	WsFTP.....	22
5.3.5	Putty.....	23
5.4	Utviklingsarkitektur.....	23
6.	Kravspesifikasjon.....	24
6.1	Generelle krav.....	24
6.2	mPetre community.....	24
6.2.1	Database.....	24
6.2.2	Dalet.....	24
6.2.3	Community.....	24
6.2.4	Søk i Dalet data.....	24
6.2.5	A og B lister.....	25
6.2.6	Chat.....	25
6.2.7	Nyheter.....	25
6.2.8	Artister.....	25
6.2.9	Generell informasjon.....	25
6.2.10	Postkort.....	25
6.2.11	Flash-player.....	25
6.2.12	Administrasjonssystem.....	25
6.2.13	Kompatibilitet.....	25
6.2.14	Ytelse.....	25
7.	Design.....	26
7.1	Teknisk design.....	26
7.1.1	Databasedesign.....	26
7.1.2	Design av PHP-struktur.....	26
7.1.3	Design av Flashstruktur.....	27
7.2	Visuelt design.....	27
7.2.1	Utviklede forslag.....	29
8.	Utvikling.....	32
8.1	Sidekart.....	32
8.2	ER-modell av database.....	33
8.2.1	Oppretting av av database.....	33
8.2.2	Valg av datatyper.....	33
8.2.3	Opprettelse av tabeller.....	34
8.3	Utvikling av design.....	39
8.4	Utvikling av PHP-dokumentene.....	39

8.4.1	Utvikling av sitefil	39
8.4.2	Innlesning av bilder	40
8.4.3	Visning av bilder	41
8.4.4	Utvikling av import av XML i fra Dalet.	41
8.4.5	Utvikling av brukerdelen	43
8.4.6	Utvikling av innloggede brukere	43
8.4.7	Glemt passord og brukernavn.....	43
8.4.8	Utvikling av chat	44
8.4.9	Utvikling av poskort.....	44
8.4.10	Utvikling av flashplayer	45
8.4.11	Utvikling av administrasjonsmodul	45
8.4.12	Preloader i Flash	46
9.	Testing.....	47
9.1	Planlegging.....	47
9.2	Gjennomføring.....	47
10.	Resultater	48
11.	Konklusjon.....	49
12.	Videreutvikling.....	50
13.	Egenvurdering	51
13.1	Erfaringer fra gjennomføringen.....	51
14.	Litteraturliste	53
14.1	Bøker	53
14.2	Nettsider	53
14.3	Figurliste	534
15.	Ordforklaringer.....	555
16.	Index.....	566
17.	Vedlegg.....	588

1. Innledning

1.1 Organisering av rapporten

I arbeidet med rapporten har vi fulgt HIG sine retningslinjer for hvordan rapporten skal være organisert. I de tre første kapitlene har vi gått gjennom bakgrunnen og målene for prosjektet. Vi har også dokumentert hvordan vi har organisert og gjennomført oppgaven. Kapittel 4 - 9 tar for seg selve utviklingen av siden, og den tekniske biten av oppgaven. I slutten av rapporten kommer konklusjon og en egen vurdering på hvordan vi har gjennomført oppgaven. Vi har laget en site, som på mange måter bare er grunnmuren på en lang prosess med endringer og forbedringer. Derfor føler vi at det er naturlig å ha med noen tips til videreutvikling, og dette er også noe NRK Upunkt har snakket om. Vi har tatt med ting både de og vi har foreslått. Bakerst i rapporten finnes index og en ordliste med forklaringer av begreper som er benyttet i rapporten. I tillegg følger all kildekode med på cd-rom og annen dokumentasjon med som vedlegg.

Måten vi har bygd opp selve sidene i rapporten på, er at vi har valgt å ha 4 forskjellige typer overskrifter, alle i fonten Arial. Hovedoverskriftene for de ulike kapitlene har størrelsen 20 pt, og er i fargen grå. Denne fargen går også igjen i logoen for prosjektet, som er plassert oppe i høyre hjørne på alle sidene. Dette er også fargen på sidetallene og teksten "Hovedprosjekt 2002" som går igjen nederst på alle sidene. Neste underoverskrift har størrelsen 12 pt, og har fått fargen blå. Denne fargen har vi valgt fordi det skal være lett å skille den fra 3. underoverskrift, som har størrelsen 11 pt og har svart farge. Bokstavmellomrommet er også mindre på 3. underoverskrift.

De tre overskriftene som er nevnt nå har løpende nummerering ved siden av seg, slik at det skal bli oversiktlig å se hvor man er om man slår opp på et tilfeldig sted i rapporten. Ett-tallet følger alle underoverskrifter under 1. kapittel, og om du er på underoverskrift 2 kommer det til et to-tall. Og så kommer det til et tall til ettersom du ser på de forskjellige delene under der igjen. Slik er det gjennom hele rapporten.

I tillegg har vi en fjerde overskriftstype som blir brukt der det skulle trenges. Denne har også større bokstavmellomrom, slik at den skal skille seg ekstra godt fra den løpende teksten. Størrelsen på denne er 9 pt. Fonttypen vi har brukt til den løpende teksten er Garamond. Dette er en fin fonttype med seriffer, som egner seg godt i trykte medier. Denne er i størrelsen 10 pt.

1.2 Oppgavedefinisjon

1.2.1 Problemstilling

Lag en site som skal fungere som et community for radiokanalen mPetre, der brukere må registrere seg for bli medlem. Vil et mer målrettet design og funksjonalitet kunne øke populariteten til radiokanalen sine nettsider blant målgruppen?

1.2.2 Resultatmål

Resultatet av hovedprosjektet skal bli en multimedieportal for radiokanalen mPetre. Det skal lages en helt ny site med et eget adminsystem for de som skal oppdatere de forskjellige sidene innenfor siden. Portalen skal ha høy teknisk kvalitet i tillegg til et meget godt design som er tilpasset målgruppen.

1.2.3 Gruppas effektmål

Gruppen ønsker å øke sine kunnskaper innenfor de emner hovedoppgaven favner over. Dette vil bli områder innenfor design, og da spesielt rettet mot web, flash og klient-/serversideprogrammering. I tillegg vil vi lære mer om det å jobbe på prosjektoppgaver, planlegging av disse og hvordan dokumentere prosessen. Vi må også klare å forholde oss til flere forskjellige instanser, slik som veileder, kontaktpersoner og oppdragsgiver. Dette er noe vi kan dra nytte av til siden.

1.2.4 mPetres effektmål

mPetre ønsker å øke sin popularitet blant bruken av sine websider.

1.2.5 Oppgavebeskrivelse

Gruppen skal produsere en multimedia-portal for mPetre. Denne skal fungere som en community hvor brukere må registrere seg for å kunne benytte seg av. Denne portalen skal bestå av:

Hva spilles nå funksjon.

mPetre har en server som spiller all musikk på kanalen. På denne serveren er den en software som heter Dalet. Denne softwaren genererer en ny xml fil hver gang en ny låt spilles. Vi skal med grunnlag i denne xml filen generere en tekststreng i Flash hvor låtens tittel og varighet vises. Dataene fra xml filen skal også legges i en database slik at det kan søkes etter hvilke låter som er spilt, antall spillinger med mere.

mPetre.community

Gruppen skal lage et community (samfunn) for brukere av portalen. Basis for denne portalen er databasen med spillelister. Brukere logger på med et passord og brukernavn. Nå de er innlogget kan de lage egne spillelister, basert på innhold i database, som for eksempel kan sendes til programleiderne for avspilling. Medlemmer av samfunnet kan melde seg av og på mailinglister, egen mPetre-chat og i tillegg få mulighet til å tipse venner om denne tjenesten.

Målgrupperelatert design og funksjonalitet

Gruppen skal i samarbeid med mPetre og Upunkt redaksjonen lage ett nytt uttrykk for mPetre på internett. Designet skal ha god brukervennlighet og i tillegg skal det appellere til målgruppen innenfor NRK sine vedtatte krav om design og brukervennlighet.

Administrasjonssystem

mPetre ønsker å legge ut nyheter og konkurranser på sine websider. For å gjøre dette mest mulig effektivt så vil det bli laget et administrasjonssystem på internett. Der kan en eller flere brukere som har tilgang legge inn, endre og slette nyheter og konkurranser. Disse vil umiddelbart bli synlige for brukere av mPetre sine websider.

Spillelister og artistleksikon

mPetre ønsker å opparbeide seg et artistleksikon. Dette skal gjøres med basis i spillelistene. Ukentlig legger mPetre ut spillelister av hvilke låter som spiles oftest. De har også mulighet til å legge ut et utdrag av sangene på internett. Men per i dag blir ikke disse enheten knyttet sammen. Det vi skal utvikle er et administrasjonssystem hvor Upunkt redaksjonen kan legge inn artister og knytte disse sammen med spillelistene. De kan også linke til låtene artisten har slik at brukere kan lytte på disse.

Chat

mPetre hadde en chat fra før, denne blir i hovedsak benyttet når programlederne er i studio. Et stort problem i forbindelse med denne chatten er at brukere later som om de er Christine eller Howard. Derfor var det ønskelig at de i den nye chatten skulle skille seg ut blant de andre brukerne.

Postkort og flashspill

mPetre har savnet en måte for at allerede eksisterende brukere/tilhengere kan tipse venner og bekjente om deres unike tilbud. De hadde et ønske om postkort som brukere selv skulle kunne sette sammen av ulike elementer, for så å sende disse til ønsket e-mail adresse.

De hadde også ønske om andre aktiviteter som skulle kunne trekke brukere til deres nettsider. Her var det fremme flere forslag, men hovedsakelig dreide det seg om ulike spill i flash. Dette var absolutt ikke noe nødvendighet, men noe vi skulle prøve oss på hvis tiden strakk til.

1.2.6 Avgrensning

En slik site som vi her har arbeidet med, blir en så og si aldri ferdig med. Det er alltid muligheter for oppdateringer, endringer og videreutvikling. mPetre og Upunkt har under hele prosessen kommet med innspill på ting de ville ha endret på, flere ting de ville ha med osv. Vi har hele tiden kjørt en hard politikk på at vi ikke skulle ta på oss for mye arbeid, slik at kvantitet gikk ut over kvalitet. Vi følte det var viktig å gjøre de tingene som vi hadde sagt at vi skulle gjøre i forprosjektrapporten skikkelig først.

1.3 Målgruppe for rapporten

Hensikten med prosjektrapporten er å dokumentere det vi har gjort under utviklingen av produktet. Dokumentasjonen av fremgangsmåten er viktig for at andre skal kunne gjenta dette og da kanskje unngå noen av de problemene vi har hatt.

Mottakeren av rapporten er først og fremst oppdragsgiveren vår og de som skal evaluere produktet. Rapporten skal kunne gi dem innsikt i hvordan vi har arbeidet og hva slags funksjonalitet produktet har. Den er også viktig for de som eventuelt skal videreutvikle siden, slik at de skal kunne se hva vi har tenkt, og videreføringen kan gå enklere.

1.4 Målgruppe for siden

Målgruppen for siden er mPetres lyttere. Kanalen kan høres av ca. 1,432,500 personer i noen av de største byene i landet. Men målgruppen for mPetre er ungdom i alderen 9 til 19 år. De er opptatt av musikk, musikknyheter og å treffe andre med samme interesser, og er altså spredt ut over hele landet.

1.5 Brukere av siden

Brukere av siden er, som over, mPetres lyttere og også mPetres redaksjon. De skal bruke adminsiden til å oppdatere siden, og å fylle stoff på de forskjellige sidene. I tillegg skal de også til tider logge seg inn der, og kommunisere med brukere aktivt gjennom chatten. Dette gjelder særlig programleiderne Håvard og Christine.

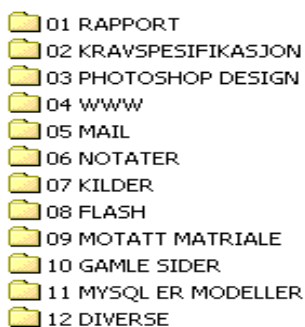
1.6 Arbeidsformer

Vi begynte arbeidet med prosjektet med å lage prosjektsider på internett. Der skrev vi litt om oss selv, hvilken utdanning vi tok, og ønsker i forbindelse med hovedprosjektet. Deretter sendte vi ut mail til de aktuelle bedriftene, og henviste til denne. Når vi hadde fått oss oppgave, endret vi litt på innholdet på denne siden, slik at det ble prosjektets hjemmeside.



Figur 1-1 Prosjekthjemmesiden.

Gruppen fikk utdelt et grupperom i "Mummidalen" på B-bygget. Dette ble hovedbasen for oppgaven, og vi møttes der hver dag i arbeidsperioden for å arbeide på prosjektet. Vi flyttet våre private datamaskiner til dette rommet, da vi synes skolens maskiner er litt trege. Vi søkte allikevel, og fikk tildelt en maskin fra skolen, og denne brukte vi i hovedsak til rapportskrivning og lokal testserver.



Figur 1-2 Mapestruktur på filserver

Ansvarsforhold

Upunkt.nrk.no og mPetre:

- dekker utgifter til prosjektet som i hovedsak består av reiser.
- står for programvarelisenser.
- stiller til rådighet kunnskap og materiell som er nødvendig for å gjennomføre prosjektet.
- server med nødvendig programvare.

Gruppen:

- skal følge tidsplan.
- skal jobbe mot målet som er satt for prosjektet.
- skal rapportere til veileder og oppdragsgiver ved eventuelle endringer.

Intern ansvarsfordeling:

Vi har valgt å ha en flat struktur innad i gruppen, der alle har ansvar for at vi kommer vel i havn med oppgaven. Likevel har vi fordelt noen hovedoppgaver mellom oss;

Kjetil: ansvar for programmering

Tom: ansvar for design

Erlend: ansvar for dialogen med NRK Upunkt

Selv vi har delt opp på denne måten, kommer alle til å delta på alle oppgavene, og for eksempel komme med designforslag.

A d m i n i s t r e r i n g ;

Vi planla også hvor lang tid vi skulle bruke på de forskjellige delene av prosjektet. Dette la vi inn i et gantt-skjema. Se vedlegg G.

Kostnader vi har i forbindelse med hovedprosjektet skal NRK betale. Vi satte opp et budsjett i forprosjektet som de godkjente. I og med at antall møter ikke ble som antatt, ble dette noe mindre enn først antatt. Se vedlegg F for faktiske kostnader vi hadde. I samarbeid med veileder hadde vi i utgangspunktet kommet fram til at vi skulle ha statusmøter med denne hver 14 dag. Dette gikk vi bort i fra tidlig i prosessen. Vi fant ut at det ikke var nødvendig å dra inn veileder så ofte, men at vi heller kunne ta kontakt med denne ved behov, når det var noe vi lurte på. Dette har fungert bra. Vi fastsatte også at vi skulle ha møter jevnlig med NRK. I forprosjektet antydte vi 10 møter. Det faktiske antallet ble 6 møter og en videokonferanse. Se vedlegg J. Vi har i tillegg hatt en løpende dialog med NRK via mail og telefonsamtaler.

1.7 Terminologi

Det har blitt lagt vekt på å benytte et lettfattelig språk i rapporten. Til tider har det vært vanskelig å finne gode norske uttrykk når terminologien innenfor data hovedsaklig baserer seg på engelsk.

2. Bakgrunn for prosjektet

2.1 Gruppas bakgrunn

Gruppens medlemmer kjenner hverandre godt fra før, og siden vi alle hadde noenlunde samme ønsker angående hva vi ville arbeide med, fant vi tidlig ut at vi ville være på samme hovedprosjektgruppe. Vi ønsket alle å jobbe med ting som webdesign, flash og noe programmering, og å øke vår kunnskap og kompetanse på disse områdene. Etter å ha fått positiv respons fra flere bedrifter, fant vi ut at det var NRK som var det mest interessante alternativet.

2.2 Oppdragsgivers bakgrunn

Innad i NRK er det mangler på folk som kan gjøre de forskjellige tingene vi kunne tenke oss å arbeide med på hovedprosjektet. De har blant annet i lengre tid snakket om å få stablet en "hva spilles nå?" funksjon på beina, men det hadde, til vi kom inn i bildet, bare blitt med praten. Omtrent samtidig som vi kom i kontakt med de, hadde de også hatt en evaluering av Upunkt sine mange sider. Der var det flere ting som hadde kommet fram i lyset som de ønsket å gjøre noe med. En av disse tingene var nettopp mangelen på et community, der brukere kunne registrere seg og bli en del av et samfunn.

2.3 Studentenes faglige bakgrunn

Gruppens deltagere går siste året på Grafisk Ingeniør ved Høgskolen i Gjøvik. Kjetil og Tom går linjen for Digital Medieteknikk, mens Erlend går Produksjonsteknikk. Relevante fag vi har hatt i løpet av utdanningen:

Grafisk produksjonsteknologi I

Studentene skal kjenne til de ulike mediers virkemidler og særtrekk. De skal ha innsikt i hensiktsmessig bruk av de ulike medier for å nå definerte målgrupper. Undervisningen skal gi studentene spesiell kjennskap til den grafiske produksjonsprosessen innen førtrykk

Grafisk kommunikasjon og typografisk design

Å etablere kunnskap om grafisk kommunikasjon, med vektlegging på typografisk design

Elektronisk Publisering

Å gi studentene en forståelse av internett som publiseringsmedium.

Databasemodellering

Studentene skal etter kurset kunne:

Kjenne til ulike databasemodeller med hovedvekt på relasjonsmodellen

Prosjektstyring

Etter gjennomført kurs skal studentene ha kunnskaper om den grunnleggende prosjektstyringen ved elektronisk publisering og multimedia

Ergonomi i digitale medier

Faget skal gi studentene bevissthet og kunnskaper om menneskelige faktorer ved design av grafiske brukergrensesnitt i digitale medier

Digital Produksjonsteknikk

Etter gjennomført kurs skal studentene ha innsikt i de ulike verktøy og arbeidsoppgaver som inngår i multimedieproduksjoner

Programmering mot WWW

Studentene skal etter endt kurs ha god kunnskap om Java teknologien og ha gode ferdigheter i Java programmering. Studentene skal beherske utvikling av både Java applikasjoner og Java applets. Kurset gir videre kunnskap om utvikling og implementering av avanserte, interaktive web-sider

Klient- og serversideprogrammering for WWW

Faget gir en innføring i de vanligste formene for serversideprogrammering, ASP/PHP3/Java servlets samt hvordan disse kan brukes sammen med ulike teknologier på klientsiden, HTML/Javascript/CSS for å skape avanserte klient-server løsninger for WWW. Etter endt kurs skal studentene ha kunnskaper og ferdigheter tilstrekkelig til å utvikle løsninger i alle de tidligere nevnte serverside baserte programmerings-teknologiene. Studentene skal etter endt kurs være i stand til å foreslå teknologi og implementere løsninger ut i fra oppdragsgivers behov og eksisterende teknologi

Informasjonen om fagene er hentet fra studihåndboken.

Relevant software vi har brukt tidligere:

Adobe Photoshop, Adobe ImageReady, Adobe Illustrator, Macromedia Flash, Macromedia Dreamweaver, Edit Plus, phpMyAdmin.

Annet

Vi har gjennom utdanningen hatt flere ulike prosjekter som inneholder noe av den samme funksjonaliteten som denne oppgaven har hatt. I tillegg jobber Kjetil i Byte AS ved siden av studiene.

Hvor fant vi kunnskap?

Vi har funnet det meste av informasjonen vi manglet ved hjelp av internett og på skolens bibliotek. Personalet ved biblioteket har vært behjelpelig når det vært bøker vi har trengt, men som de ikke hadde fra før. Bøker ble bestilt, og det tok ikke altfor lang tid før vi fikk ta de i bruk.

Resultat av research

Macormedia generator og Swift generator er begge programer som kan benyttes for å generere swf-filer utafra en swt-fil. En swt-fil er en template hvor en har ulike felter til for eksempel tekst og bilder. Ved hjelp av generator kan en automatisk få generert en swt-fil med det innholdet brukeren ønsker. Dette vil gjøre at en kan få en dynamisk løsning.

3. Prosjektgjennomføring

3.1 Praktisk framgangsmåte

3.1.1 Opstartsfasen

Planleggingsarbeidet av hovedprosjektet ble påbegynt i november med det samme gruppa hadde en oppdragsgiver. Det var i begynnelsen ikke helt bestemt akkurat hva vi skulle arbeide med, og hva vi skulle knytte det opp mot, slik at det var dette vi prøvde å bli enige med både oppdragsgiver og oss selve om. Gruppas medlemmer hadde samarbeidet tidligere, slik at vi alle var klar over hver enkelts styrke og svakheter. Denne perioden ble derfor også benyttet til å snakke om og diskutere hvordan jobbingen med prosjektet skulle utføres, og hvordan ansvarsfordelingen skulle være.

3.1.2 Forprosjekt

Januar gikk med til å gjennomføre forprosjektet for selve hovedprosjektet, se vedlegg. Den dokumenterte hvordan vi planla å gjennomføre hovedprosjektet, med tanke på ansvars- og arbeidsfordeling, hvilke rammer vi hadde på prosjektet ulike deler (økonomi, tid og ressurser) og hvilken arbeidsmetode som skulle brukes. Det ble definert hva målet for prosjektoppgaven var, og hva som gikk inn under dette. Se vedlegg K.

3.1.3 Egenlæring

Det vi på gruppen hadde minst kompetanse innenfor var Flash. De fleste andre emnene vi kom borti under hovedprosjektet hadde vi vært borti via fag ved skolen tidligere. Det førte til at vi måtte lære oss en god del på egenhånd. Til det formålet lånte vi en del bøker som omhandlet dette temaet på biblioteket ved skolen. I tillegg fant vi mye nyttig på internett. Vi brukte også noe tid i begynnelsen der vi bare lekte oss i flash, og lærte en del på den måten.

3.1.4 Utviklingsfasen

Selve utviklingen av siden begynte vi tidlig på. Siden flash bare endte opp med å bli en liten del av den totale oppgaven, var det ikke avhengig for oss å kunne det for å kunne utvikle sidene. Vi utviklet en kravspesifikasjon for oppgaven, og fortsatte med å lage en databasestruktur på grunnlag av denne. Deretter var det bare å begynne å kode. Parallelt med dette begynte vi også tidlig å tenke nytt design og ny funksjonalitet.

Men dette kom allikevel litt i andre rekke i begynnelsen. Mot slutten av prosjektet, da vi hadde fått sydd sammen funksjonaliteten og designet, gikk vi inn i en periode med testing og forbedring av produktet.

3.1.5 Sammenstilling av rapport

Det å sette sammen rapporten ble en tidkrevende prosess. Vi var desverre litt for dårlige til å skrive på rapporten i begynnelsen av perioden, men da var statusrapportene gode å ha, slik at vi kunne gå tilbake og se hva vi egentlig hadde gjort. Det var også selvfølgelig endringer i rapporten underveis, temaer som ikke var så relevante forsvant og nye kom til. Til slutt endte vi opp med en oversikt over det vi ville ha med, som vi var fornøyde med. Vi ville lage en enhetlig rapport, uten gjentakelser mellom avsnittene og med stoff som til en viss grad vil interessere leseren. Vi måtte også prøve å skrive rapporten i et ikke alt for teknisk og tørt språk der dette kunne unngås, slik at folk som ikke er så inn i slike termer også kan ha nytte av den.

3.1.6 Planlegging av framføring

Planleggingen av framføringen tok vi på et sent tidspunkt. Det var ikke noe vi ville bekymre oss for tidlig i prosessen. Vi anså det som uproblematisk å få vise fram produktet i en forelesningssal, siden de fleste salene har datamaskiner som er koblet til internett, og vårt produkt ligger tilgjengelig ute på en server på NRK. Vi delte opp hva vi skulle ha med i framføringen mellom oss, slik at alle skulle ha noe å snakke om.

3.1.7 Organisering av kvalitetssikring

Under hele prosessen gjennomførte vi brukervennlighetstesting. Her ble medarbeidere i NRK benyttet som testgruppe. Under utviklingsprosessen har vi også brukt oss selv veldig mye som testpersoner for hverandre. Har en programmert, har de andre testet det, og kommet med synspunkter til endringer og forbedringer. Vi la siden tidlig ut på en server, slik at de som sitter i de forskjellige redaksjonene på NRK kunne gå inn der for å se, og komme med sine meninger. De skulle også kontrollere og godkjenne de forskjellige delene som vi arbeider med, for å passe på at de holdt høy nok standard til å bli publisert senere.

I tillegg ligger siden mye tryggere inne på NRK, enn om den skulle ligge på en maskin her på skolen. Det er jo veldig viktig at denne er opp 24 timer i døgnet, siden den har hva spilles nå?- funksjonen, som til en hver tid skal vise hva som spilles, og lagre dette i databaser. Vi prøvde å få så få mellomledd mellom Dalet som står på en server på Tyholt i Trondheim, og webserveren der siden vår ligger, inne på Marienlyst. Endte opp med ett mellomledd, men ideelt sett burde det ikke være noen.

Vi har også brukt et felles mailsystem for kontakten med alle involverte parter, der all mail har blitt lagret. Dette har vært veldig viktig, da mye av kommunikasjonen har foregått via mail.

3.2 Kommunikasjon

Oss – veileder

Kommunikasjonen med veileder har gått veldig greit. Vi har fått de møtene vi har hatt behov for, og ellers har kommunikasjonen skjedd pr mail.

Oss – NRK Upunkt

Denne kommunikasjonen har i hovedsak foregått pr mail og telefon, siden NRK Upunkt holder til i Oslo og Trondheim, og gruppen har holdt til på Gjøvik. Alle har tatt like stor del i denne kommunikasjonen. Vi har vært fem ganger i Oslo, og to av møtene var videokonferanse med avdelingen i Trondheim. Vi har også vært på et møte i Trondheim,

hvor Oslo-avdelingen også var representert. Kommunikasjonen med upunkt.nrk.no og mPetre har vært veldig bra. Vi har fått raske tilbakemeldinger, og føler at Upunkt har tatt oss seriøst og vært veldig åpne og imøtekommende ovenfor våre ideer.

Inn i gruppen.

Kommunikasjon internt i gruppen har vært veldig bra. Det ikke vært noen konflikter, til tross for at vi har sittet tett på et grupperom med dårlig ventilasjon. Grunn til at gruppen har fungert så bra som den har gjort, kommer nok av at gruppesammensetning er som den er. Vi er tre personer som kjenner hverandre godt, og fungerer fint som en enhet. Alle har bidratt med sitt og det har vært en veldig positiv holdning til prosjektet gjennom hele prosessen.

3.2.1 Framdriftsplan

Hvordan prosjektet skulle bli utført med hensyn på tidsforbruk, ble gjort i forprosjektrapporten. Tidsbruken på de ulike delene ble beregnet, og alle milepæler fastsatt. Vi kom fram til at vi hadde seks forskjellige milepæler. Alt dette satte vi oversiktlig inn i et ganttskjema. I tillegg pekte vi på kritiske punkter i arbeidsperioden, punkter som det var meget viktig at gikk i orden.

Vi passet hele tiden på at vi, så langt vi kunne, skulle holde tidsfristene vi hadde satt opp. Dette gikk veldig bra, og vi lå ajour så og si hele tiden. På slutten av perioden satte vi opp et ganttskjema med oversikt over tiden vi faktisk hadde brukt. Se vedlegg H.

4. Prinsipper og teori

Produktet skal bli en ferdig webside på internett. Denne websiden skal være dynamisk. Dermed må websiden utvikles med et verktøy som gjør siden lett å oppdatere. Vi har da valg en MySQL database til lagring av informasjon og PHP for kommunikasjon mellom databasen og websiden.

4.1 Innføring i HTML

Hyper text marking language er standarden for utvikling av websider. HTML 4.0 er utviklet i fra SGML for beskrivelse av utseende av websider. Selve HTML-filen (eller HTM-filen) er egentlig et helt enkelt tekstdokument som kun inneholder skrift, tall og enkelte symboler. I HTML-dokumentet skriver man inn forskjellige koder som forteller hvordan siden skal se ut. Det finnes koder for omtrent alt: hvilken skrift man skal bruke, hvilken farge skriften skal ha, hvilken bakgrunn siden skal ha, om det skal være bilder på siden og så videre.

Det er nettleseren som gjør mye av jobben. En nettleser er et program som sørger for å "oversette" eller "tolke" kodene som står i HTML-dokumentet slik at den vises som en "normal" og lesbar side. Nettleseren viser selvfølgelig ikke kodene når den viser siden. Microsoft Internet Explorer og Netscape Navigator er slike nettlesere. Den norskproduserte nettleseren Opera begynner også å bli anerkjent, mye p.g.a. at den tar så liten plass på harddisken.

4.1.1 Innføring i Javascript

Javascript var et språk som ble lansert med Netscape 2.0, og som raskt ble så populært at de andre nettleserne måtte støtte det for å kunne konkurrere med Netscape. Netscape har alltid vært det selskapet som har utviklet Javascript videre, og Netscape 3.0 og 4.0 kom med mange nyttige utvidelser av språket.

Eksempel på javascript

```
<script language="javascript">
  document.writeln("Hei, verden!");
</script>
```

4.1.2 Innføring i CSS

CSS står for Cascading Style Sheets (på norsk stilsett) og er en relativt ny standard som skal utfylle HTML. Sidene på WWW skrives i HTML og de som laget HTML (World Wide Web Consortium eller W3C) ønsker ikke at ting som har med layout å gjøre skal inn i HTML. I stedet ønsker de at HTML-markeringene bare skal beskrive informasjonen i dokumentet, slik at vevleseren kan vise dokumentet på den måten som

passer leseren best. Man må huske på at på WWW kan leseren bruke et hvilket som helst system, kanskje til og med kjøre i tekstmodus eller være blind, og sidene skal være like tilgjengelige uansett. Derfor har W3C jobbet med CSS siden 1994, men forslaget var altså ikke ferdig før september '96, og nettlesere som forstår CSS har ikke blitt vanlig før nå nylig. Likevel har dette hele tiden vært målet til W3C, selv om enkelte produsenter av nettlesere har blitt overivrige og utvidet HTML med ting beregnet på å påvirke presentasjonen. I og med HTML 4.0 er disse tingene på vei ut av HTML igjen. W3C utga i mai '98 CSS level 2 (CSS2), som utvider CSS1 med en del nyttige ting.

Fordelene med CSS

CSS gir forfatteren (og brukeren) mulighet til å angi hvordan HTML-dokumenter skal vises, enten i HTML-dokumentet, eller i en egen CSS-fil. Fordelene med dette er mange, og en av de viktigste er at man slipper å hele tiden skrive FONT FACE, FONT COLOR, BGCOLOR osv over alt i HTML-dokumentene. I stedet kan man ha all layout-informasjonen i noen få CSS-filer, og ved å endre en av disse kan man endre utseendet på mange HTML-dokumenter. Dette gjør også at HTML-markeringene blir enklere å skrive og det blir lettere å vedlikeholde sidene. En siste fordel er at det blir mindre å laste ned, slik at nedlastingen går raskere. (Når CSS begynner å bli vanlig i bruk vil dette faktisk kunne begrense belastningen på nettet litt også.).CSS har også rent grafiske fordeler, i og med at det gir muligheter rent grafisk som ikke er tilgjengelige via HTML. Man har langt større mulighet for å bestemme hvor elementer skal plasseres, avstand mellom elementer, skrift-størrelser, bakgrunnsfarger, innramming (ikke FRAMES) osv. Det er også noen fordeler ved dette for brukeren. Brukeren skal nemlig kunne angi sine egne stilsett som skal gjelde i tillegg til de som er angitt av forfatteren. Slik kan en svaksynt bruker f.eks. angi at all tekst skal vises i dobbel størrelse, mens resten av reglene fra forfatterens stilsett beholdes. Dersom forfatteren har gjort jobben sin skikkelig skal dokumentet tåle det uten å bli uleselig. Den siste, og kanskje viktigste, fordel er at stilsett "degrades gracefully". Det vil si at om man har en nettleser som ikke støtter stilsett så vil sider som bruker stilsett fortsatt være like leselige i den, for siden er skrevet i vanlig HTML. Dette gjør at CSS kan innføres uten at det går ut over nettlesere med gammel programvare som ikke støtter CSS. Det betyr også at bruk av CSS ikke går ut over lesere med ikke-grafiske nettlesere som Lynx. Eksempel på hvordan H1 overskriften skal se ut:

```
: H1 { font-weight: bold; font-size: 26pt; font-family:
"Times New Roman"; text-align: center }
```

4.2 Innføring i nettlesere

Det finnes en rekke ulike produsenter av nettlesere. Den mest populære er Internet Explorer. De siste årene har markedsandelen som bruker denne nettleseren nærmet seg 90%. Den andre store på markedet er Netscape. Denne nettleseren tar i hovedsak for seg de resterende prosentene. Internet Explorer har etter lanseringen av 4.0 versjonen fått bedre og berdere støtte for javascript og css. I Netscape 4.0 har god støtte for javascript, men har en del hull når det gjelder css. I Netscape 6.0 og 6.2 så er dette forbedret kraftig. Netscape er markedsledene når det gjelder Linux OS. Av andre nettlesere så er det norsk firmaet Opera på vei fremover. Denne nettleseren har god støtte for både javascript og css. Dekodingen av html og css ligner på Internet Explorer.

Nettlesere statistikk

Nettlesere versjon	Apr 01	Jul 01	Okt 01	Jan 02	Apr 02
Internet Explorer 6.x			9%	23%	32%
Internet Explorer 5.x	79%	81%	71%	64%	55%
Internet Explorer 4.x	8%	7%	5%	4%	3%
Netscape 4.x	7%	5%	5%	4%	4%
Andre Netscape kompatibel	3%	4%	5%	1%	1%

Skjermopløsning statistikk

Oppløsning	Apr 01	Jul 01	Okt 01	Jan 02	Apr 02
1024x768 eller mer	39%	38%	37%	39%	41%
800x600	51%	53%	54%	52%	51%
640x480	6%	4%	4%	4%	3%
Ukjent	4%	5%	5%	5%	5%

Farge dybde

Farge dybde	Apr 01	Jul 01	Okt 01	Jan 02	Apr 02
32 bit (max color power)	26%	26%	28%	31%	34%
24 bit (16 million farger)	9%	10%	11%	10%	10%
16 bit (65000 farger)	58%	58%	55%	53%	50%
8 bit (256 farger)	5%	4%	4%	4%	4%
Andre eller ukjent	2%	2%	2%	2%	2%

OS statistikk

Operativ system	Apr 01	Jul 01	Okt 01	Jan 02	Apr 02
Windows 98 / ME	80%	74%	70%	65%	65%
Windows 95	8%	8%	7%	6%	5%
Windows NT	5%	5%	5%	4%	5%
Windows 2000	4%	7%	11%	16%	17%
MAC	1%	1%	1%	1%	2%

Javascript støtte

JavaScript	Apr 01	Jul 01	Okt 01	Jan 02	Apr 02
JavaScript 1.2 (eller høyere)	85%	88%	86%	88%	87%
JavaScript 1.1.(eller lavere)	0%	0%	0%	0%	0%
JavaScript slått av	14%	11%	12%	11%	11%
Ukjent	1%	1%	2%	1%	2%

Kilde: <http://thecounter.com>

4.3 Innføring i XML

Extensible Markup Language (XML) er en lett utgave av SGML.

XML er en relativt ny standard (utviklingen startet i 1996), som foreløpig ikke har støtte hos alle nettlesere. Men XML handler mer om at webdokumenter skal kunne leses av flere programmer enn bare nettlesere.

XML er akkurat som HTML på den måten at det bruker tagger (med < foran og > bak) og attributter. Men XML gir mulighet for mye større fleksibilitet rundt tolkningen av taggene. Med XML kan man faktisk definere sitt eget markeringsspråk som formidler informasjon på mer presis måte enn i HTML.

E(X)stensibel Markup Language er et dokument for å beskrive data.

```
<katalog>
  <tlfnummer>
    <navn>Kjetil</navn>
    <tlf>98280066</tlf>
  </tlfnummer>
  <tlfnummer>
    <navn>Tom</navn>
    <tlf>32323232</tlf>
  </tlfnummer>
</katalog>
```

4.4 Innføring i databaser

4.4.1 Hva er en database

En database brukes for å enkelt lagre data. Databasen tar seg av hvordan dataene lagres mens det programmereren må kunne er å hente ut, legge inn og endre dataene.

4.4.2 Databaseteori

MySQL er en av de mest populære databasene som finnes. Databasen er SQL kompatibel. Det vil si at man kan brukes SQL, Structure Query Language, for å legge inn og hente ut informasjon i fra databasen. MySQL er kjent for hurtige og effektive spørringer sett i forhold til andre databaser. Databasen har støtte for lagring av tall, strenger og binære data.

4.5 Innføring i SQL

SQL ble først definert av IBM Research i 1967. I 1979 ble det introdusert kommersielt av Oracle Corporation. En komite ved ANSI (American National Standards Institute) har nå godkjent SQL som standard språket for RDBMS (relational database management systems), relasjonsdatabase systemer. I dag er SQL i bruk over hele verden på alle mulige plattformer fra stormaskiner, minimaskiner, arbeidsstasjoner og til vanlige PC'er.

Hva kan du gjøre med SQL ?

Med SQL har du et komplett sett av funksjoner både for å manipulere data, men også for å definere data.

Du kan:

- lage tabeller i databasen
- lagre informasjon i tabellene
- velge ut den informasjon du ønsker
- endre data og datastrukturer
- kombinere og kalkulere data til ny informasjon

Forskjellen på SQL og andre datamanipuleringsspråk er at SQL ikke er prosedyre-orientert. Det betyr at du kan si hva som skal utføres i stedet for hvordan. Du kan for eksempel med en enkel kommando i SQL oppdatere en rekke tabellrader i databasen uten å tenke på hvor data er lagret eller hvordan de lagres. Disse systemopplysningene blir tatt hånd om av selve databasesystemet og du kan konsentrere deg om dataene.

4.6 Innføring i PHP

PHP (Hypertext PreProcessor) er ett serverside programmeringsspråk. Med serverside mener vi at kilden blir parsert (kjørt) på webserveren. Siden PHP er på serverside, vil man aldri kunne se selve PHP-kode i kilden til den eventuelle siden. PHP blir installert som en modul på webserveren sammen med Apache og blir kjørt når en request kommer. Hvis PHP er installert riktig, vil Apache la PHP-applikasjonen kjøre gjennom filen. Da gjør den alt den skal gjøre, og viser dette til slutt i HTML, som nettleseren din vil takle. PHP er et scriptspråk som er utviklet til å lage websider. Dette scriptspråket har en mengde med innebygde funksjoner, deriblant å kommunisere med MySQL.

4.7 Innføring i Macromedia Flash

Flash er et animasjonensverktøy hovedsakelig beregnet på bruk på websider. Flash bruker i hovedsak vektorgrafikk og et programmeringsspråk for å danne animasjonene. Programmeringsspråket er i "slekt" med javascript og kalles actionscript.

4.7.1 Innføring i Flash templates

Flash har en egen mulighet til å eksportere flash movies i Shockwave template format (.swt). Disse filene er beregnet på bruk mot Macromedia generator. Macromedia generator er en server programvare som lager dynamiske flash sider ut i fra databaser med mer. I en swt-fil så defineres det felter hvor de dynamiske komponentene skal komme inn. Disse feltene defineres ved at man i flash-filen lager et felt betegnet med {felt}. Da vil det dynamisk innholdet bli lagt i {felt} når filen kjøres på server.

4.8 Innføring i Swift-tools

Swift-tools er en programvare som gjør det mulig å generere flash dynamisk. Swift tar i mot en Flash "Template" og ved hjelp av en fil med parametere så dannes det en flash-fil som kan vises til bruker. Swift har mulighet til å bytte ut tekster og bilder i flash-malen ved hjelp av parametere som sendes i en sws fil til Swift-tools. Sws filen har en innhold som sier noe om hvilke felter og bilder som skal endres i swt filen.

Eksempler på innhold i en sws fil:

```
DEFINE tekst = "innhold i tekstfelt"
```

Denne koden vil bytte ut feltet {tekst} i flash malen med "innhold i tekstfelt" når dette kjøres i Swift-tools.

4.9 Innføring i Imagemagick

Dette er en programvarepakke utviklet for Linux plattform for behandling av bilder. Bilder kan konverteres mellom ulike formater og man kan skalere bilder ved hjelp av noen strenger på kommandolinjen. Imagemagick kan også kjøres via PHP ved å gjøre kall på kommandolinje funksjoner i PHP.

Kommando som vil skalere fil.jpg til 200 ganger 200 pixler

```
Mogrify -geometry 200x200 fil.jpg
```

4.10 Innføring i webservere

Webservere brukes for å gi brukeren den informasjonen han er ute etter. I hovedsak brukes det servere på NT og Linux/Unix plattform. NT har et stort marked med servere som kjører asp og MsSQL databaser. Linux/unix kjører i hovedsak MySQL og PHP/Pel/Python løsninger.

Løsningen kjører på en Linux OS. Vi har hatt utviklingsservere som kjører Redhat 7.1 og Mandrake, mens webløsningen skal brukes på en Unix maskin.

5. Utstyr

5.1 Datamaskiner

Arbeidstasjoner

AMD Athlon XP 1700+, 1,47 GHz, 256 MB DDR ram, 40 GB harddisk, Windows XP

AMD Athlon, 1 GHz, 256 MB ram, 40 GB harddisk, Windows 2000

Intel Celeron 700 MHz, 256 MB ram, 60 GB harddisk, Windows 2000

Digital HiNote laptop Pentium 2, 166 MHz, 96 MB ram

Webservere

Intel Pentium 4, 1 GHz, 512 MB ram, RAID 5, Redhat 7.1 linux

Unix (NRKs webserver)

Filserver

Intel pentium 166MHz, Mandrake 7.0

5.2 Diverse

Kamera

Canon D30

Nikon Coolpix 995

Canon EOS 5

Scanner

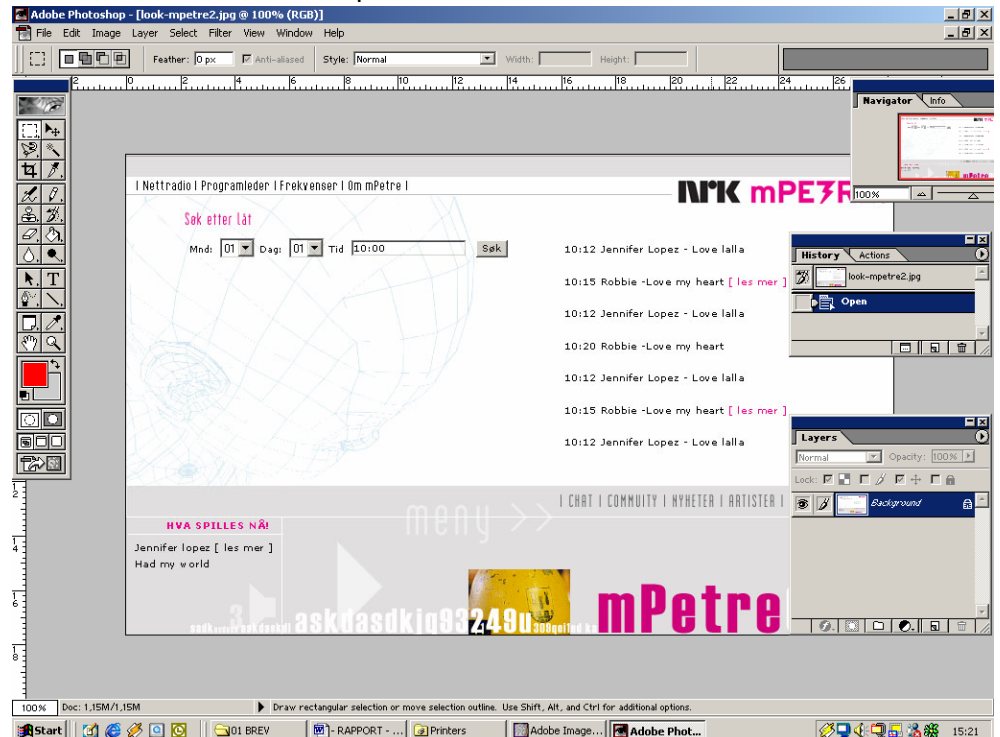
SnapScan e50

Lyd

Trust SoundWave 240 3D

5.3 Utviklingsprogramvare

5.3.1 Adobe Photoshop



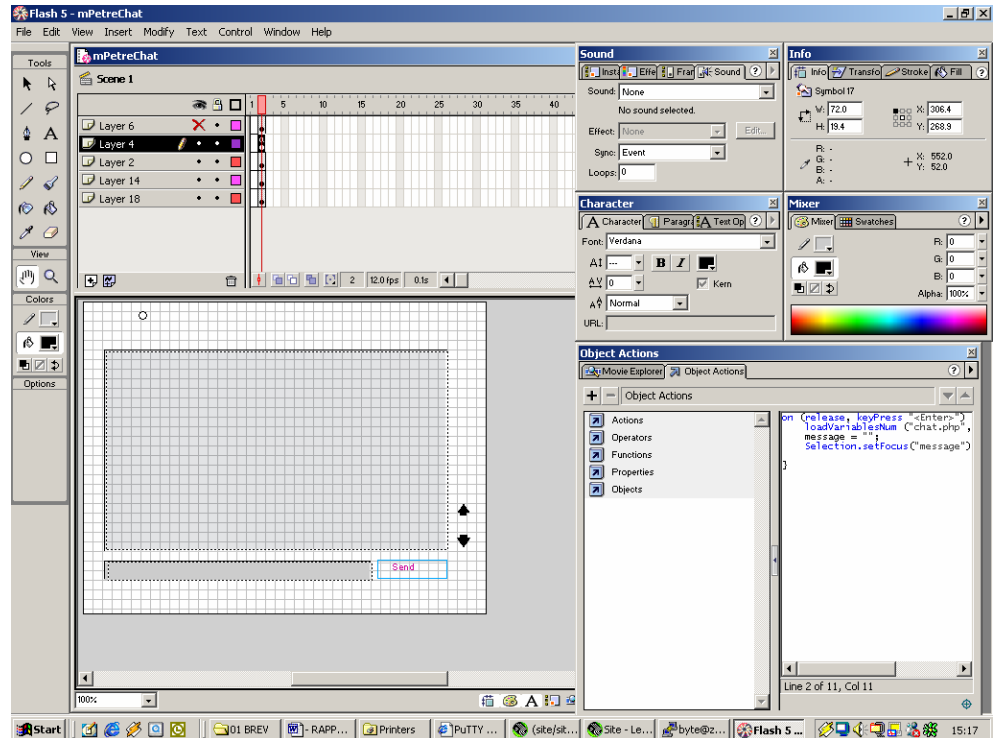
Figur 5-1 Adobe Photoshop i bruk med utvikling av et design forslag.

Adobe Photoshop har lenge vært dominerende når det gjelder bildebehandling. De siste versjonene har også fått meget god støtte til utvikling av websider. Adobe Photoshop ble brukt til å utvikle design av siden. Grafikkelementer ble bearbeidet og justert slik at det ble tilpasset en webside. Når vi ble ferdig med et designutkast ble filen lagret og åpnet i ImageReady.

ImageReady

Imageready er en programvare som følger med Adobe Photoshop 6.0 og høyere. Dette programmet er spesialisert for tilpasning og utvikling av websider og webdesign. Programmet har funksjonalitet for å lage onmouseover javascript. Det vil si at når en bruker tar musen over et bilde så skiftes dette ut med et annet. ImageReady har også en «slice» funksjon. Denne funksjonene gjør at du kan ta et stort bilde å dele det opp i deler som plasseres i en tabell. Dette gjør at utseende blir helt likt originalen i psd format, men du har nå konvertert bildene til gif og jpg format samt at det automatisk har blitt generert en HTML fil.

5.3.2 Macromedia Flash 5.0



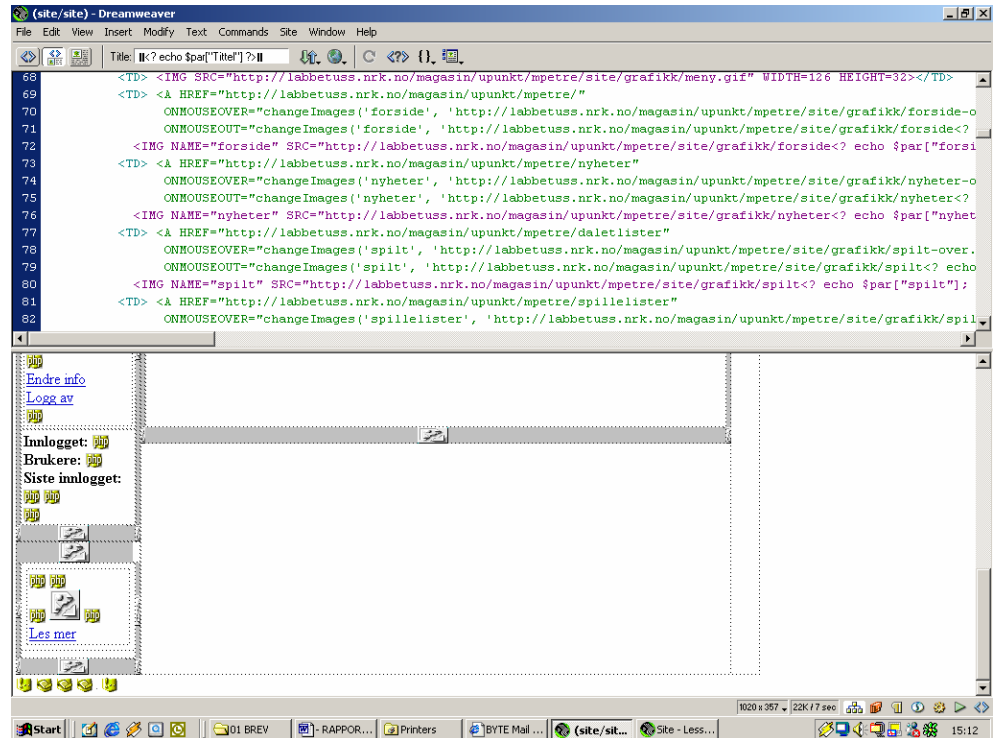
Figur 5-2 Macromedia Flash 5.0 i bruk ved utvikling av chat.

Flash 5.0 er laget av Macromedia. Macromedia er en av de største programvareprodusentene når det gjelder verktøy til webutvikling.

Grensesnitt

Flash har flere vinduer inne i programmet. En tidslinje, en stage og flere andre vinduer for modifikasjon av ulike parametere i flash-movie. Tidslinjen gir en kontroll over hvilke elementer som skal vises når. Hvert enkelt lag(layer) har sin egen tidslinje. I denne tidslinjen er det mulig å sette inn actionscript. Stage gir et visuelt forhåndsinntrykk av hvordan den ferdige flash-filen vil se ut. Her plasseres grafikk, lyd og andre elementer. Actionscriptvinduet gir tilgang til å endre actionscript på tidslinjen eller script som er koblet til elementer i flash. Her vil man ha tilgang til funksjonene på venstre side, mens koden ligger i høyre vindu.

5.3.3 Macromedia Dreamweaver 4.0



Figur 5-3 Macromedia Dreamwaver i arbeid med vår sitefil.

Macromedia Dreamweaver 4.0 er en markedsledende HTML editor på det profesjonelle markedet. Den har gode muligheter for både automatisk generert kode og manuelt skrevet HTML. Dreamweaver har en innebygget ftp software til kommunikasjon med server. For editering av HTML ser grensesnittet ut som på Figur 5-3

5.3.4 WsFTP

WsFTP er en ftp software for å laste opp/ned filer via ftp protokollen. Vi brukte denne programvaren til å laste opp filer på våre testservere. WsFTP har også støtte for å sette rettigheter på filer. Dette ble også brukt.

5.3.5 Putty

```

byte@zuno.byte.no: /home/byte
login as: byte
Sent username "byte"
byte@byte.no's password:
Last login: Wed May 15 15:33:08 2002 from 213-145-170-194.dd.nextgentel.com
(byte@zuno byte)$ ls
core      NAV5.zip      public_html  trondheim.zip
Desktop  NeroS540.zip  thomas.zip   www
(byte@zuno byte)$

```

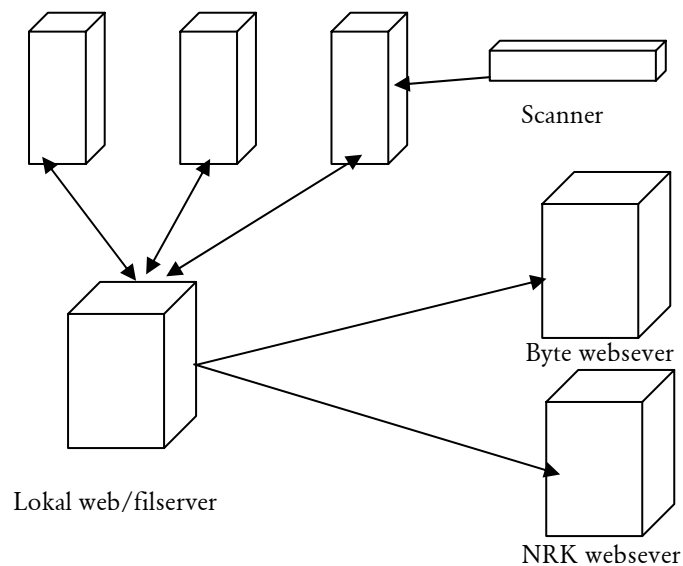
Figur 5-4 Putty software.

Putty er en software for fjernstyring av servere. Dette programmet ble brukt under hele prosjektet. Først for konfigurering av egen webserver, så for å tilpasse NRK sin webserver våre krav til software. Ved hjelp av denne klienten kan man utføre operasjoner på en server uten å ha fysisk tilgang til denne.

5.4 Utviklingsarkitektur

Under er en oversikt over utviklingsarkitekturen som har blitt brukt i prosjektet.

Arbeidstasjon 1 Arbeidstasjon 2 Arbeidstasjon 3



6. Kravspesifikasjon

6.1 Generelle krav

Sidene skal kunne lestes ned med et ISDN modem uten problemer. Større Flash-filer skal ha preloaderer som viser hvor lang tid det er igjen. Fargene som benyttes skal i være i henhold til mPetre sin grafiske profil. Det skal benyttes en lett leselig font på websidene. Fonten skal være grotesk da disse er mest leselige på skjerm. Brukere skal få en indikasjon på hvilken side de er på ved hjelp av et tydelig grafiskelement. Systemet skal ha innebygde sikkerhetssjekker slik at bruker får tilbakemelding hvis han gjør en feil.

6.2 mPetre community

6.2.1 Database

Databasestrukturen skal følge reglene om normalform. Databasen må være rask og effektiv. Databasen må være tilpasset mange poster/rader.

6.2.2 Dalet

Dalet har en mulighet til å eksportere XML-filer med hva som blir spilt på mPetre til en hver tid. Disse XML-filene skal mottas av systemet for behandling. Dataene i denne XML filen skal vises realtime i en flash-fil i tillegg til at de skal legges i en database for lagring. Dataene i databasen skal brukes som grunnlag for resten av innholdet på websiden.

6.2.3 Community

Brukere av mPetre på nett skal få muligheten til å bli medlem av mPetre.community. Bruker skal selv kunne opprette en brukerkonto. De skal også ha tilgang til å endre informasjon om seg selv i ettertid. Opplysninger som skal registeres er brukernavn, navn, passord, sted og fødselsår. Brukernavn og passord brukes til å logge seg på systemet.

6.2.4 Søk i Dalet data

Bruker skal ha tilgang til å søke i hva som spilt på mPetre. De skal kunne søke etter når en artist ble spilt og når hvilke artister eller låter ble spilt.

6.2.5 A og B lister

Brukere skal kunne se på A og B listen på mPetre. De skal få informasjon om artist, låt og eventuelt høre et lydklipp i fra låten.

6.2.6 Chat

Brukere skal kunne prate med hverandre ved hjelp av en flash-chat. Chatten skal ha en enkel formgivning og fungere effektivt.

6.2.7 Nyheter

Brukere skal få tilgang til nyheter i fra mPetres redaksjon. Nyhetene skal inneholde tittel, ingress og en tekst. Det kan også være bilder og linker tilknyttet en nyhet.

6.2.8 Artister

Bruker vil få informasjon om artister. Artistene beskrives med tekst og bilder. Det vil også være knyttet linker til artisten.

6.2.9 Generell informasjon

Generell informasjon om mPetre skal presenteres på en ryddig og oversiktlig måte.

6.2.10 Postkort

Brukere kan lage sine egen postkort og sende til venner. Bruker kan legge inn tekster og bilder som skal vises i postkortet. Dette ble lagd i flash.

6.2.11 Flash-player

Det skal utvikles en flash fil som realtime henter informasjon om hva som spilles på mPetre. Denne skal vises på alle mPetre sine undersider.

6.2.12 Administrasjons system

Det skal utvikles et enkelt administrasjonssystem for å styre bruker, nyheter, A og B lister og artist informasjon. Dette skal være enkelt å bruke, men også meget kraftig.

6.2.13 Kompatibilitet

Dataene som lagres må kunne brukes i andre medier med enkle grep. Det vil for eksempel være aktuelt å utvikle en wap-tjeneste med samme data.

6.2.14 Ytelse

Systemet skal utvikles med tanke på mange brukere . Derfor må systemet være effektivt.

7. Design

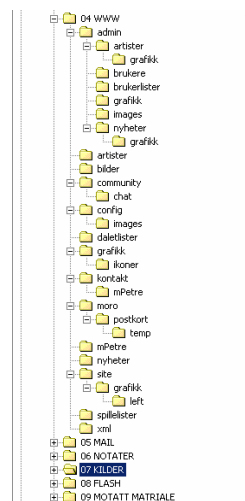
7.1 Teknisk design

7.1.1 Databasesdesign

Under oppbyggingen av databasen, måtte vi hele tiden tenke på at det skulle være enkelt å oppdaterer den, samt det skulle gå hurtig å søke i den. Dette krevde at linkingene mellom de forskjellige tabellene måtte være gjort på en mest mulig korrekt måte, og at tabellene måtte være godt gjennomtenkte.

7.1.2 Design av PHP struktur

Mange av PHP dokumentene i vår løsning er avhengig av en del faste komponenter. I stedet for å lage mange versjoner av samme komponent, har vi steden utviklet en del filer og funksjoner som inkluderes i de ulike sidene.



Figur 7-1 Mapestuktur på websever

D a t a b a s e k o b l i n g

Det ble utviklet en fil db.php som kobler seg opp mot MySQL databasen. Denne filen er så og si inkludert i alle filene i systemet.

Sitefil

Alle filene i systemet er knyttet opp mot en design fil. Denne filen styrer layout på hele siden. Det er to ulike site-filer, en for brukersidene og en for adminsiden.

Funksjoner

Dette er en fil med innebygde funksjoner for omformatering av timestamp til en standard dato og klokkeslettstreng.

7.1.3 Design av Flashstruktur

Alle flash filer har en preloader som sier hvor lang tid det er igjen før filen er ferdig lastet.

7.2 Visuelt design

mPetre og upunkt redaksjonen hadde et ønske om ett helt nytt uttrykk for mPetre på internett. De hadde lenge hatt i tankene å få laget et nytt design på Upunkt sine sider. Stikkord i forbindelse med dette var følgende: lyse sider, friske farger og mere luft. Dette kom nok mye av at de syntes at dagens sider er for mørke og ikke formidlet det rette uttrykket for Upunkt.

Brukersidene

Da vi begynte på designet til de nye mPetre-sidene hadde vi hele tiden disse stikkordene som utgangspunkt. Vi fikk fri tøyler, bortsett fra at NRK og mPetre-logoen skulle bestå. mPetre sin logo skiller seg også ut med sin rosa farge, og denne fargen ble dermed utgangspunktet for videre fargevalg. Vi utviklet flere designforslag, som vi fikk positiv tilbakemelding på. De følte at vi hadde oppfatt deres ønsker og at vi beveget oss i riktig retning. Et annet viktig moment i designprosessen var at vi måtte ta hensyn til det veldig varierende innholdet på de ulike sidene, i tillegg til de faste elementer som hva spilles player, ukens artist og innloggings/pålogget-elemente.

Det endelig designet er bygget opp med en grå ramme rundt de ulike elementene. På denne måten kan vi styre størrelsen på elementene utafra innholdet. Det var spesielt viktig i forhold til hovedrammen som skal ramme inn det varierende innholdet på de ulike sidene.

På toppen av siden er det laget et grafiskelement som inneholder logoen til NRK og mPetre og med tilleggsteksten community. Under topelementet er hovedmeny elementet plassert. Det er disse to elementene som har fast størrelse, og dermed angir bredden på siden.

Resten av siden er delt inn i to kolonner. På venstre side er de faste elementet som hva spilles nå?-player, innlogging/pålogget, ny bruker og ukens artist plassert, men til høyre er hovedrammen med innholdet plassert. Bredden på de ulike elementet er fast, mens høyden styres dynamisk utafra innholdet til en hver tid.

Hovedmenyen er bygget opp ved hjelp av ulike felter som kan sammenliknes med ulike skilleark. Teksten i de ulike felten er duset ned, når brukeren fører musepekeren over de ulike elementene vil teksten bli tydeligere. Da brukeren klikker på et av menyvalgene vil teksten bli tydeligere og det vil komme frem et rosa felt under menyvalget. Dette er gjort for at brukeren lett skal oppfatte hvilken side han befinner seg på.

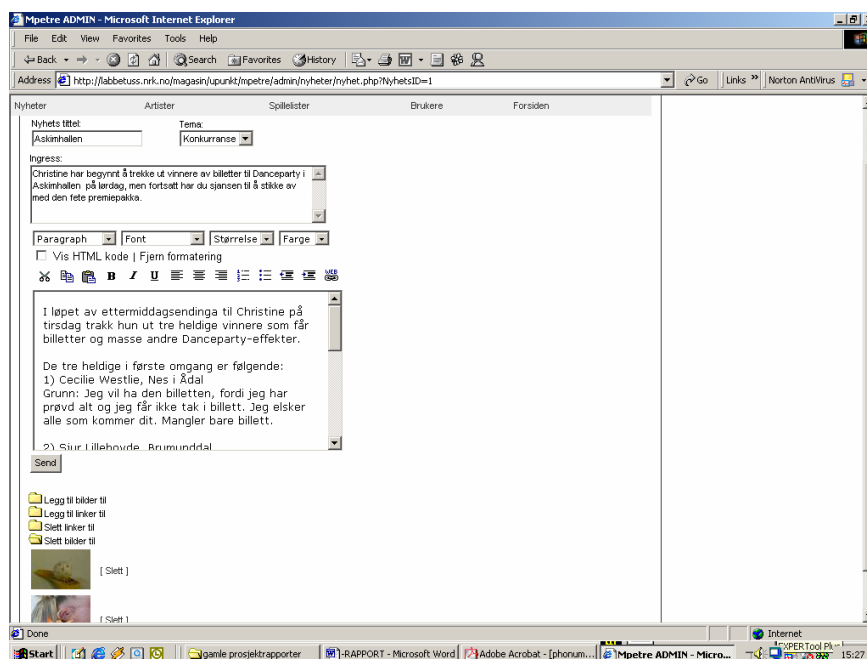
Designet på bildegalleriet er identisk med de ulike elementet fra hovedsiden. Det hele blir rammet inn av en grå ramme. Denne rammen er statisk, da størrelsen på de ulike bildene blir skalert ved opplasting. Galleriet kommer frem som et nytt vindu, vi har derfor valgt å ha med NRK og mPetre logoen også her.

Vi har også laget diverse ikoner og symboler som skal illustrere forskjellige valg. Disse er i hovedsak rosa, og er dermed lett gjenkjennelig. Et eksempel på dette er et kamera som kommer frem hvis det finnes flere bilder til en nyhet eller artist.

Designet på hva spilles playeren er utviklet av Upunkt, og tilpasset Upunkt sitt nye design.

Adminsidene

På disse sidene har vi lagt mest vekt på at de skal være funksjonelle og lett å benytte. Designet er veldig enkelt med et menyelement på toppen, som består av fem ulike hovedvalg. Hvis en klikker på et av valgene vil det komme frem en nedtrekksmeny, hvor en finner de ulike valgene som er. Under meny elementet er det en ramme som styres dynamisk ut i fra innholdet på de ulike sidene. Her vil all funksjonalitet og aktivitet skje.

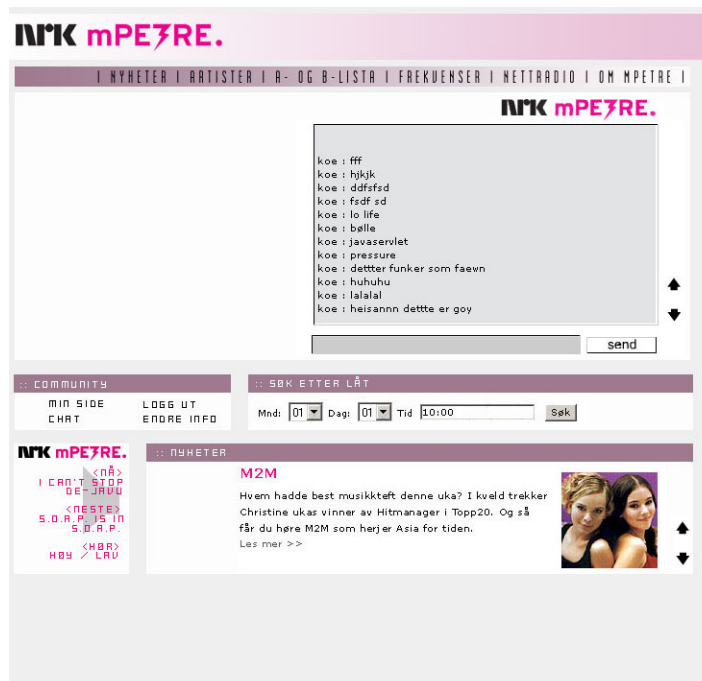


Figur 7-2 Adminsidene er enkle og oversiktlige

7.2.1 Utviklede forslag

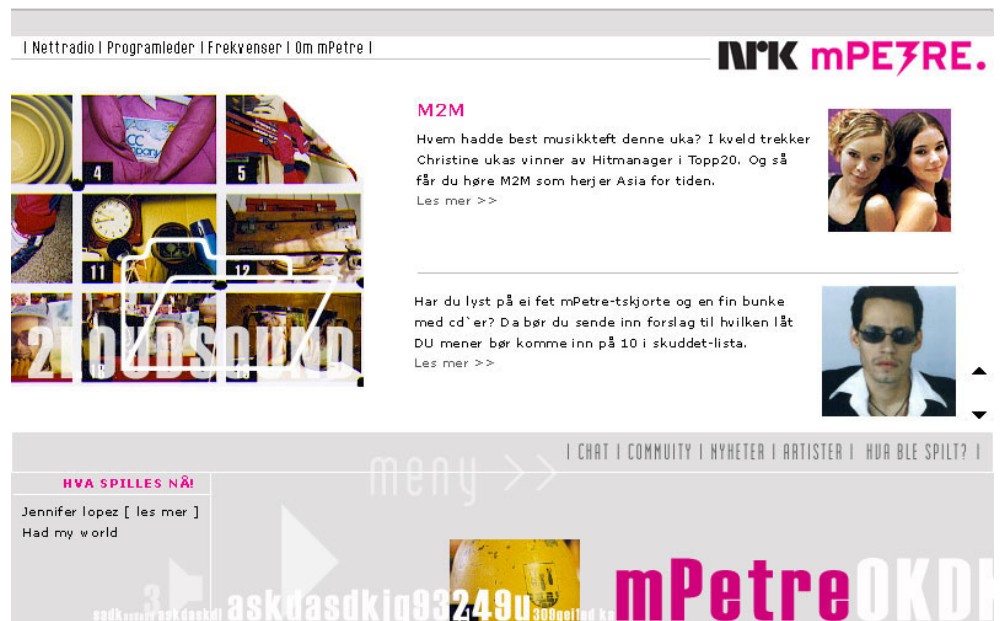


Figur 7-3 Første utkast.

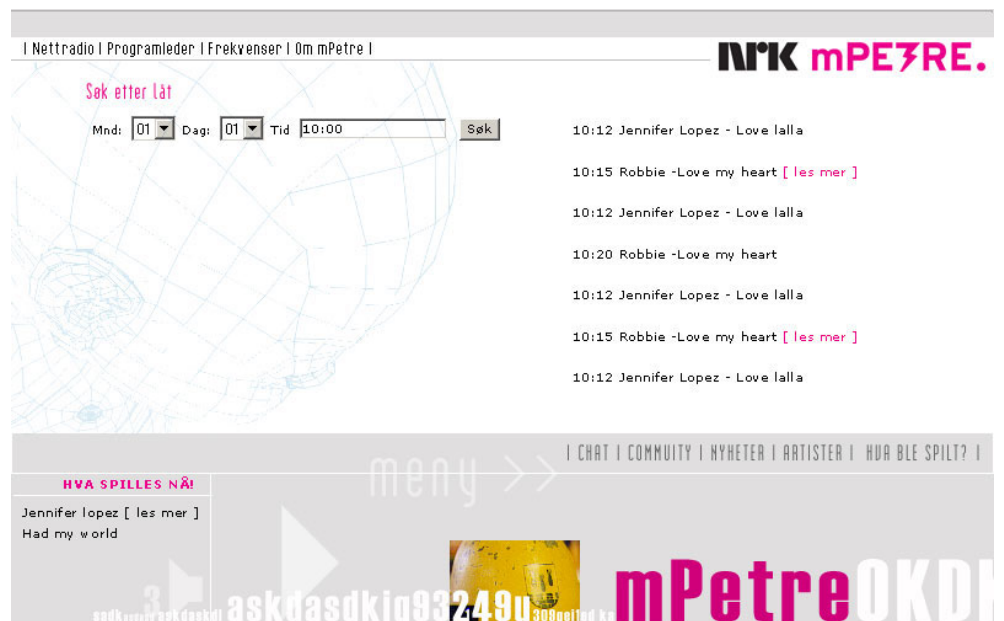


Figur 7-4 Samme utkast men litt annet innhold.

Forslag 2

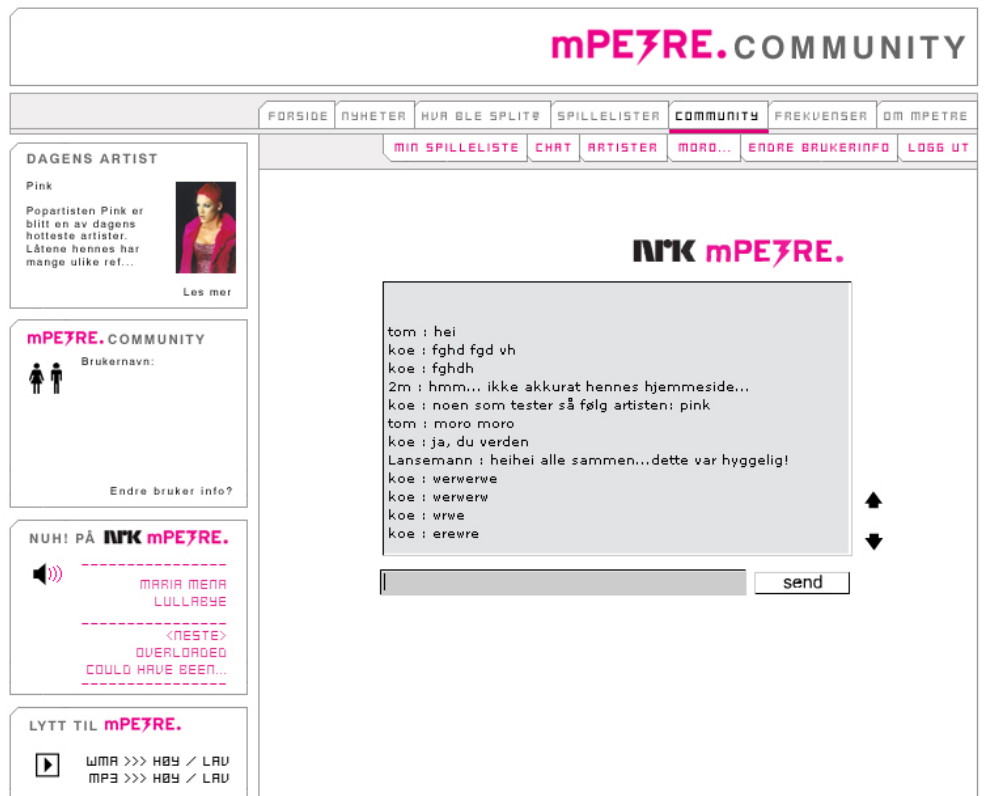


Figur 7-5 Andre utkast.



Figur 7-6 Andre utkast med søking etter låter.

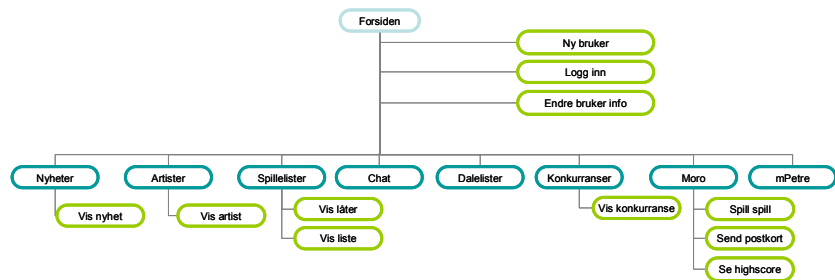
Forslag 3



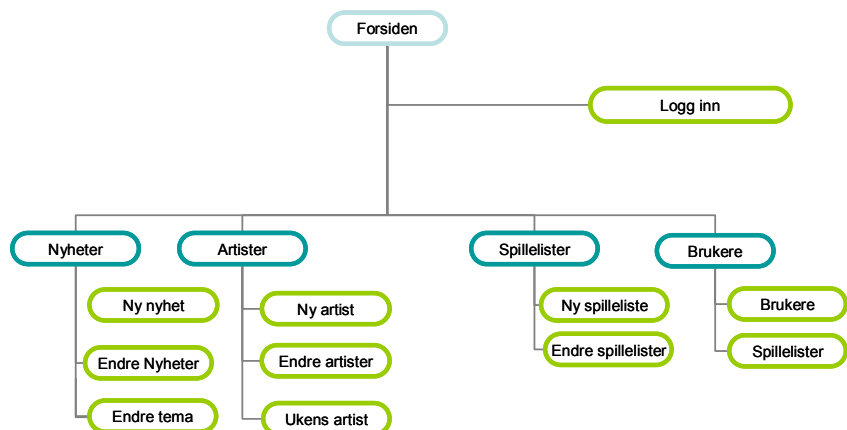
Figur 7-7 Tredje forslag.

8. Utvikling

8.1 Sidekart



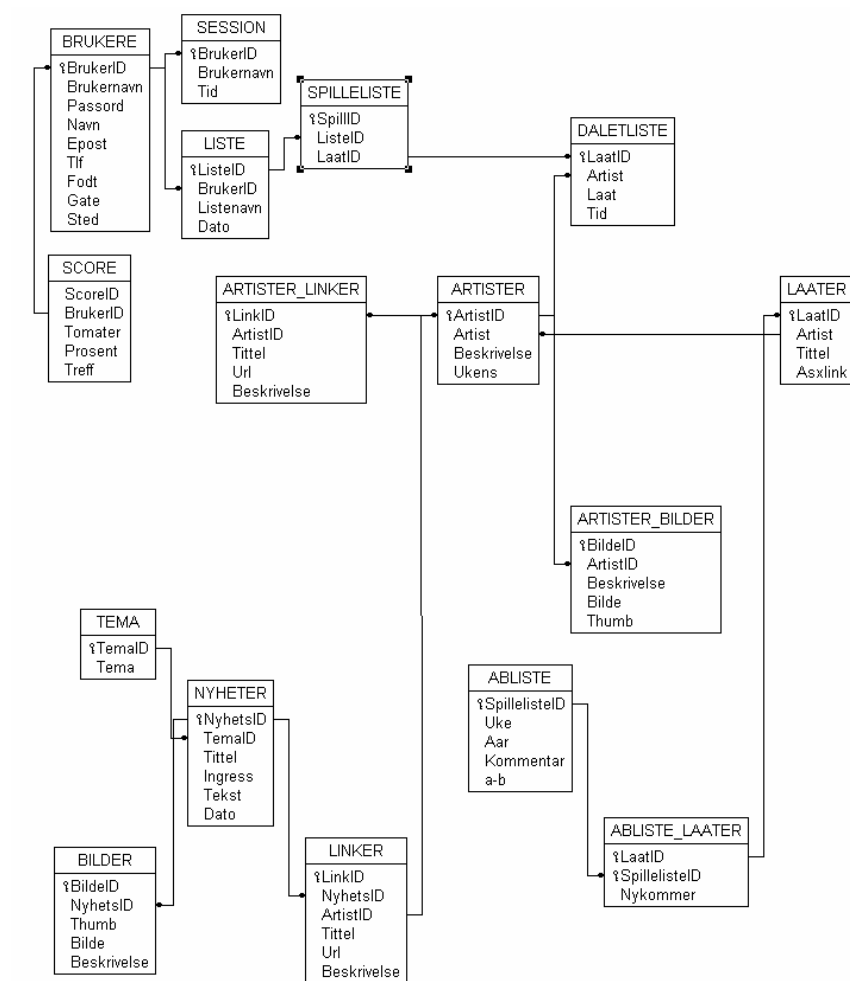
Figur 8-1 Sidekart for brukersidene.



Figur 8-2 Sidekart for administrasjonsmodulen.

8.2 ER-modell av database

Det ble totalt opprettet 16 tabeller for mPetre.Community.



Figur 8-3 ER-Modell for MySQL databasen.

8.2.1 Oppretting av av database

Databasens tabeller ble opprettet via et webgrensesnitt for administrasjon av MySQL databaser. Denne sotwaren som er utviklet i PHP kalles phpMyAdmin. Her kan man utføre alle operasjoner mot en MySQL database i et grafisk grensesnitt.

8.2.2 Valg av datatyper

En Mysql-database har en rekke ulike datatyper for lagring av ulike data.

Int

Int er valgt til primærnøkler og fremmednøkler i databasen. Alle primærnøkler og fremmednøkler er opprettet med en int lengde på 10. De er også satt til å være Unsigned. Dette gjør at dataene(tallene) i feltet kun går i fra 0 og oppover, det vil si at feltet ikke kan være negativt.

V a r c h a r

Denne datatypen er brukt til alle felt med tekstlig innhold som ikke skal overstige 255 tegn. Brukes i stedet for Text datatypen p.g.a. at den er raskere å søke i.

T e x t

Text brukes der det skal legges inn lange tekster. Det vil si i nyheter og i artister.

T i m e s t a m p

Timestamp brukes for å sette inn ett eksakt tidspunkt.

M e d i u m b l o b

Brukers der det skal legges binær data inn i et felt. Vi legger inn bilder i disse feletene. Vi bruker mediumblob i stedet for blob for det denne er mere plassbesparende.

E n u m

Enum feltet brukes kun i A-/B-listetabellen. Dette feltet vil bare lagre et gitt sett med verdier.

Y e a r

Brukes til å lagre årstall.

D e c i m a l

Denne datatypen brukes til å lagre desimaltall. Her defineres antall desimaler i opprettelse av feltet.

8.2.3 Opprettelse av tabeller

A B L I S T E

Denne tabellen tar vare på informasjon om mPetre sine spillelister. Denne tabellen lagrer informasjon om hvilken uke og hvilket år spillelisten gjelder. Den lagrer også en kommentar samt om det er en A eller B liste. Primærnøkkelen er SpillelisteID som er en relasjonsnøkkel som brukes i ABLISTE_LAATER

```
# Tabell-struktur for tabell `ABLISTE`
CREATE TABLE ABLISTE (
  SpillelisteID int(10) unsigned NOT NULL auto_increment,
  Uke char(2) NOT NULL default '',
  Aar year(4) NOT NULL default '0000',
  Kommentar text NOT NULL,
  ab enum('a','b') NOT NULL default 'a',
  PRIMARY KEY (SpillelisteID)
) TYPE=MyISAM;
```

ABLIS TE _ LAATER

Denne tabellen lagrer info om hvilken spilleliste som har hvilke låter. Den lagrer derfor SpillelisteID, LaatID samt at den lagrer om låten er en nykommer. Tabellen har en delt primærnøkkel mellom SpillelisteID og LaatID

```
# Tabell-struktur for tabell `ABLIS TE _ LAATER`
#

CREATE TABLE ABLIS TE LAATER (
  SpillelisteID int(10) unsigned NOT NULL default '0',
  LaatID int(10) unsigned NOT NULL default '0',
  Nykommer tinyint(2) NOT NULL default '0',
  PRIMARY KEY (SpillelisteID,LaatID)
) TYPE=MyISAM;
```

LAATER

Denne tabellen lagrer låter på spillelistene. Den lagrer informasjon om artist, låtens navn og en eventuell link til lydfil. LaatID er primærnøkkel i denne tabellen.

```
# Tabell-struktur for tabell `LAATER`
#

CREATE TABLE LAATER (
  LaatID int(10) unsigned NOT NULL auto increment,
  Laat varchar(80) NOT NULL default '',
  Artist varchar(80) NOT NULL default '',
  ASXlink varchar(10) NOT NULL default '',
  PRIMARY KEY (LaatID)
) TYPE=MyISAM;
```

ARTISTER

Denne tabellen lagrer informasjon om artisten lagt inn via adminmodulen. Det lagres informasjon om artisten navn og en beskrivelse av artisten. Artisten kan også settes til å være ukens artist og dermed bli plassert på forsiden av mPetre.community. Primærnøkkelen er ArtistID.

```
# Tabell-struktur for tabell `ARTISTER`
#

CREATE TABLE ARTISTER (
  ArtistID int(10) unsigned NOT NULL auto_increment,
  Artist varchar(100) NOT NULL default '',
  Beskrivelse text NOT NULL,
  Ukens enum('0','1') NOT NULL default '0',
  PRIMARY KEY (ArtistID)
) TYPE=MyISAM;
```

ARTISTER _ BILDER

Denne tabellen lagrer et stort og et lite bilde av en artist. Samt at det er mulig å skrive inn en beskrivelse til bildet. Bildet relateres til ARTISTER ved hjelp av ArtistID. Primærnøkkelen i tabellen er BildeID.

```
# Tabell-struktur for tabell `ARTISTER BILDER`
#

CREATE TABLE ARTISTER BILDER (
  BildeID int(10) unsigned NOT NULL auto_increment,
  ArtistID int(10) unsigned NOT NULL default '0',
  Bilde blob NOT NULL,
  Thumb blob NOT NULL,
  Beskrivelse varchar(150) NOT NULL default '',
  PRIMARY KEY (BildeID)
) TYPE=MyISAM;
```

ARTISTER_LINKER

Tabellen lagrer informasjon om linker til en artist. Innholdet linkes til en artist ved hjelp av ArtistID. Det lagres informasjon om url, tittel på url og en beskrivelse. Primærnøkkelen er LinkID.

```
# Tabell-struktur for tabell `ARTISTER_LINKER`

CREATE TABLE ARTISTER_LINKER (
  LinkID int(10) unsigned NOT NULL auto_increment,
  ArtistID int(10) unsigned NOT NULL default '0',
  Tittel varchar(80) NOT NULL default '',
  Url varchar(80) NOT NULL default '',
  Beskrivelse text NOT NULL,
  PRIMARY KEY (LinkID)
) TYPE=MyISAM;
```

BRUKERE

Denne tabellen lagrer den personlige informasjonen om en bruker. Informasjon som lagres er brukernavn, passord, navn, epost, tlf, fødselsår og sted. Brukernavn og Epost må være unike. Det vil si at det ikke kan være to rader med samme brukernavn eller epost. Primærnøkkelen er BrukerID.

```
# Tabell-struktur for tabell `BRUKERE`

CREATE TABLE BRUKERE (
  BrukerID int(10) unsigned NOT NULL auto increment,
  Brukernavn varchar(20) NOT NULL default '',
  Passord varchar(20) NOT NULL default '',
  Navn varchar(100) NOT NULL default '',
  Epost varchar(80) NOT NULL default '',
  Fodt year(4) NOT NULL default '0000',
  Tlf varchar(20) NOT NULL default '',
  Gate varchar(100) NOT NULL default '',
  Sted varchar(80) NOT NULL default '',
  PRIMARY KEY (BrukerID),
  UNIQUE KEY Brukernavn (Brukernavn),
  UNIQUE KEY Epost (Epost)
) TYPE=MyISAM;
```

DALETLISTE

Denne tabellen lagerer informasjonen som kommer i fra Dalet sitt XML-dokument. Tabellen lagrer artist, låt og tid. I tillegg, for testing, så blir hele rå xml filen lagt inn i databasen. Det legges også inn tidspunkt når innleggingen skjedde i databasen. Primærnøkkel er LaatID.

```
# Tabell-struktur for tabell `DALETLISTE`

CREATE TABLE DALETLISTE (
  LaatID int(10) unsigned NOT NULL auto increment,
  Artist varchar(80) NOT NULL default '',
  Laat varchar(80) NOT NULL default '',
  Tid varchar(80) NOT NULL default '',
  XML text NOT NULL,
  Innlegging timestamp(14) NOT NULL,
  PRIMARY KEY (LaatID)
) TYPE=MyISAM;
```


LISTE

Denne tabellen lagrer informasjonen om en bruker sin liste. Det blir lagret hvilken bruker som har opprettet listen, ved bruk av fremmednøkkelen BrukerID. Det blir også lagret et navn på listen samt hvilken dato listen er opprettet.

```
# Tabell-struktur for tabell `LISTE`

CREATE TABLE LISTE (
  ListeID int(10) unsigned NOT NULL auto increment,
  BrukerID int(10) unsigned NOT NULL default '0',
  Listenavn varchar(80) NOT NULL default '',
  Dato timestamp(14) NOT NULL,
  PRIMARY KEY (ListeID)
) TYPE=MyISAM;
```

SPILLELISTE

Denne tabellen lagrer hvilke låter en bruker har i sin spilleliste. Informasjon som lagres er ListeID og LaatID. LaatID hentes i fra DALELISTE mens ListeID hentes i fra LISTE.

```
# Tabell-struktur for tabell `SPILLELISTE`

CREATE TABLE SPILLELISTE (
  SpillID int(10) unsigned NOT NULL auto increment,
  ListeID int(10) unsigned NOT NULL default '0',
  LaatID int(10) unsigned NOT NULL default '0',
  PRIMARY KEY (SpillID)
) TYPE=MyISAM;
```

NYHETER

Denne tabellen styres vi adminmodulen. Her lagres dataene om nyheter i databasen. Informasjon som registreres er tittel, ingress, tekst, tema og dato. Tema hentes fra TEMA tabellen. Primærnøkkel er NyhetsID.

```
# Tabell-struktur for tabell `NYHETER`

CREATE TABLE NYHETER (
  NyhetsID int(10) unsigned NOT NULL auto increment,
  TemaID int(10) unsigned NOT NULL default '0',
  Tittel varchar(80) NOT NULL default '',
  Ingress text NOT NULL,
  Tekst text NOT NULL,
  Dato timestamp(14) NOT NULL,
  PRIMARY KEY (NyhetsID)
) TYPE=MyISAM;
```

TEMA

Denne tabellen inneholder de ulike teamene en nyhet kan være. Primærnøkkel er TemaID. Tema er en tekststreng på maks 80 tegn.

```
# Tabell-struktur for tabell `TEMA`

CREATE TABLE TEMA (
  TemaID int(10) unsigned NOT NULL auto increment,
  Tema varchar(80) NOT NULL default '',
  PRIMARY KEY (TemaID),
  UNIQUE KEY Tema (Tema)
) TYPE=MyISAM;
```

LINKER

Denne tabellen lagerer informasjon om linker til en nyhet. Tabellen relateres til nyheter ved hjelp av nøkkelen NyhetsID. Det lagres informasjon om url, tittel og en beskrivelse. Primærnøkkelen er LinkID.

```
# Tabell-struktur for tabell `LINKER`

CREATE TABLE LINKER (
  LinkID int(10) unsigned NOT NULL auto increment,
  NyhetsID int(10) unsigned NOT NULL default '0',
  ArtistID int(10) unsigned NOT NULL default '0',
  Tittel varchar(80) NOT NULL default '',
  Url varchar(80) NOT NULL default '',
  Beskrivelse text NOT NULL,
  PRIMARY KEY (LinkID)
) TYPE=MyISAM;
```

BILDER

Denne tabellen lagrer bilder tilknyttet en nyhet. Det lagres en liten og en stor utgave av bildet. Det er også mulighet til å lagre en beskrivelse til bildet. Tabellen relateres til nyheter ved hjelp av NyhetsID.

```
# Tabell-struktur for tabell `BILDER`

CREATE TABLE BILDER (
  BildeID int(10) unsigned NOT NULL auto increment,
  NyhetsID int(10) unsigned NOT NULL default '0',
  Thumb blob NOT NULL,
  Bilde mediumblob NOT NULL,
  Beskrivelse text NOT NULL,
  PRIMARY KEY (BildeID)
) TYPE=MyISAM;
```

SCORE

Denne tabellen lagrer informasjon om hvor høy score en bruker har fått i flash-spillet. Primærnøkkelen er ScoreID. Tabellen relateres til BRUKERE ved hjelp av BrukerID. Det lagres informasjon om hvor mange tomater som er brukt, antall treff og prosentfordelingen.

```
# Tabell-struktur for tabell `SCORE`

CREATE TABLE SCORE (
  ScoreID int(10) unsigned NOT NULL auto_increment,
  BrukerID int(10) unsigned NOT NULL default '0',
  Tomater int(4) NOT NULL default '0',
  Treff int(4) NOT NULL default '0',
  Prosent decimal(4,2) NOT NULL default '0.00',
  PRIMARY KEY (ScoreID)
) TYPE=MyISAM;
```

SESSION

Denne tabellen brukes til å lagre innloggede brukere. Brukere som finnes i denne tabellen betegnes som innloggede. Det som lagres er Brukernavn, BrukerID og Tid.

Tid oppdateres hver gang en bruker laster siden.

```
# Tabell-struktur for tabell `SESSION`

CREATE TABLE SESSION (
  BrukerID int(10) unsigned NOT NULL default '0',
  Brukernavn varchar(80) NOT NULL default '',
  Tid timestamp(14) NOT NULL,
  PRIMARY KEY (BrukerID)
) TYPE=MyISAM;
```

8.3 Utvikling av design

Bildeformater

Vi har valgt to ulike bildeformater, fordi komprimeringsgraden er forskjellig på fotografier og grafikk i de to formatene. Det ene er formatet JPEG (Joint Photographic Experts Group), som brukes til fotografiske bilder. Joint Photographic Experts Group er en underkomite i standardiseringsorganisasjonen ISO. JPEG- definisjonen er en standard for komprimering av stillbilder. Dette er en såkalt «lossy» komprimeringsmetode der man ved komprimeringen mister en del data.. I beste kvalitet vil imidlertid bildet ikke være synbart forringet i forhold til originalen, selv om filstørrelsen er redusert til en brøkdel. Når man åpner en JPEG-fil, blir bildet automatisk dekomprimert. Dette formatet fungerer aller best for å komprimere fotografier. Det andre formatet er GIF (Graphics Interchange Format), et populært filformat for bitmapgrafikk (punktgrafikk). GIF ble utviklet av CompuServe på midten av 1980-tallet. Det inneholder maksimalt 8-bits farger (256 farger) og bruker LWZ-metoden for komprimering. Dersom bildet inneholder skarpe kontraster, som man får ved å legge på tekst eller tegne i bildet, er GIF et velegnet format for Web. Etttersom formatet bare støtter 256 farger, får du ikke lagret bilder i full RGB-modus i GIF -format, men programmene tar selv hånd om den nødvendige fargereduksjonen under lagringen. Det vil si at dette formatet fungerer best til bruk på grafikkelementer.

Fra bilder til html

Designet på sidene ble utviklet i Adobe photoshop. Da vi var ferdig med design-prosessen ble bildene klippet opp i de ulike delene som layouten skulle bestå av. For å tilpasse de ulike delene til bruk på web, benyttet vi funksjonen save for web i photoshop. De ulike delene ble lagret i enten gif eller jpeg. Hovedmenyen ble eksportert til Adobe ImageReady for å animeres. Her ble menyen delt i opp i de ulike stykkene som den skulle bestå av, for så å animere de ulike delene. I første omgang ble det kun laget animasjoner på hva som skulle skje når brukeren tar musepekkeren over de ulike valgene i menyen. Da dette var gjort laget vi en siste utgave av de ulike delene, som skulle benyttes når brukeren har trykket på et av valgene. Her ble det ikke laget noe animasjonen, da denne delen blir gjort ved hjelp av php.

8.4 Utvikling av PHP-dokumentene

8.4.1 Utvikling av site-fil

For å gjøre endringer og oppdateringer av design på websiden enklest mulig ble det utviklet en global fil med hoveddesignet til siden.

Meny	
Logg inn boks Ukens artist Info om antall innloggede	Hovedramme

Figur 8-4 Viser oppbygning av sitefilen.

Hele websiden baserer seg på to ulike maler, en for adminsyste­met og en for brukerdelen. Sidemalen er en PHP fil med to funksjonskall, `topp()` og `bunn()`. I disse funksjonene skrives det ut HTML kode for å generere designet på websiden. Alle filene i systemet har inkludert denne sitefilen. Funksjonene tar også noen parametere. `Topp()` tar en array med parametere. Den tar i mot i sidens tittel og hvilket menyelement som skal være aktivt. `Bunnfunksjonen` tar også i mot en parameter, men den brukes ikke foreløpig. Den er lagt ved for eventuell utvidelse. Dette er et eksempel på bruk av site filen.

```
<?
include "../config/db.php"; //inkludere databasekobling
include "../config/log-in.php"; //inkludere log-in fil
include "../config/tabell.php"; //lager en kant tabell
include "../config/funksjoner.php"; //inkluder funksjoner
$parameter["tittel"] = "Søk i hva som er spilt på mPetre";
//setter tittel
$parameter["spilt"] = "-valgt"; //velger hvilket grafikk element som skal
være aktivt
topp($parameter); //skriver ut toppen av html koden
?>
Her kommer innholdet i hovedrammen
<?
bunn("") //skriver ut HTML i bunnen av filen
?>
```

8.4.2 Innlesning av bilder

Alle bilder som legges inn dynamisk blir lagt binært i databasen. Dette gjøres for å få en effektiv og oversiktlig måte å håndtere bildene på. Bildene blir skalert ved hjelp av `Imagemagick` sin `mogrify` funksjon og deretter blir bildedataen lagt inn i en variabel i PHP. Denne variabelen blir så lagt inn i databasen. Det lages to utgaver av hvert bilde, en liten og en stor.

Kode for innlegging av bilde i databasen:

```
if(file_exists($Bilde))
{
    exec("mogrify -geometry 200x200 -quality 50 -density 72x72
$Bilde");
    $Bildedata = addslashes(fread(fopen($Bilde, "r"),
filesize($Bilde)));

    exec("mogrify -geometry 70x70 -quality 50 -density 72x72 $Bilde");
    $Thumbdata = addslashes(fread(fopen($Bilde, "r"),
filesize($Bilde)));

    $sql = "insert into ARTISTER BILDER (ArtistID, Beskrivelse, Bilde,
Thumb) values ('$ArtistID', '$Beskrivelse', '$Bildedata',
'$Thumbdata')";
    $res = mysql_db_query($db, $sql);

    if(!mysql_error())
    print "<script>alert('Bilde er lagt inn');</script>";
}
```

8.4.3 Visning av bilder

Ved at bildene ligger i databasen må det lages et eget script for å vise bildene på websiden. Dette scriptet mottar en parameter, BildeID. Dette gir scriptet informasjon om hvilken rad det skal hente ut. For å få vist bildet på nettsiden så må det sendes ut en header som sier at dette er et bilde. Derfor sendes header("Content-type:image/jpeg");

```
<?
if($BildeID) {
include "../config/db.php";
$query = "select Bilde from ARTISTER BILDER where BildeID = '$BildeID'";
$res = MYSQL DB QUERY($db, $query);
$rad = mysql result($res,0);
Header( "Content-type: image/jpeg");
echo $rad;
}
?>
```

HTML koden for å vise et bilde, hvis denne filen har navn vis_artist.php, blir slik:

```

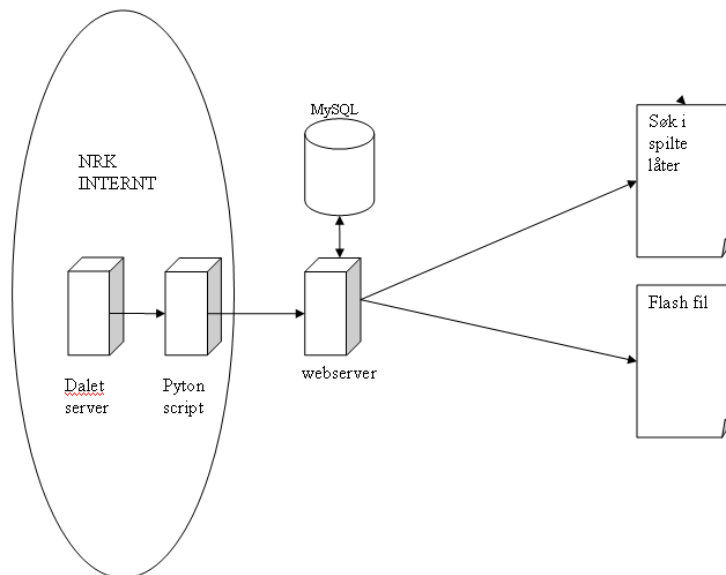
```

Dette ville vist bilde nummer 3 ut på websiden.

8.4.4 Utvikling av import av xml i fra Dalet.

Dalet

Dalet er en programvare som brukes innenfor kjøring av radioprogrammer. Dalet har en database med låter og informasjon om disse. Låten blir kjørt ut av Dalet som lyd, og i tillegg lages det en XML-fil hver gang en ny låt starter.



Figur 8-5 Viser XML-flyt fra Dalet til bruker.

XML fra Dalet

Dalet skriver denne xml-filen hver gang en ny låt spilles:

```
<BroadcastMonitor>
  <updated>2001-08-20T11:55:58</updated>
  <stationName>Unknown</stationName>
  <Current>
    <startTime>2001-08-20T11:55:58</startTime>
    <itemId>1000020674:176887</itemId>
    <titleId>286973</titleId>
    <itemCode>051497CS0002</itemCode>
    <itemReference> </itemReference>
    <titleName>I CAN'T STOP</titleName>
    <artistName>DE-JAVU</artistName>
    <albumName></albumName>
  </Current>
  <Next>
    <startTime>2001-08-20T11:59:55</startTime>
    <itemId>1000020674:176888</itemId>
    <titleId>156071</titleId>
    <itemCode>045779CS0001</itemCode>
    <itemReference> </itemReference>
    <titleName>TITTEL</titleName>
    <artistName>ARTIST</artistName>
    <albumName></albumName>
  </Next>
</BroadcastMonitor>
```

Denne Dalet-filen blir generert hver gang en ny låt eller jingel starter. Denne filen blir lest av et python-script som putter filen til webserver hvis den ikke har blitt sendt før. På webserveren så står det et script som tar i mot denne XML-filen og behandler den. Først hentes kun informasjon som er interessant. Det vil si nåværende låt, artist og tidspunkt den startet, og neste låt og artist. Nåværende låt legges i databasen hvis det er en låt, ikke en jingel. Dette sjekkes ved at itemCode til en låt er en streng på 12 tegn. Alle låter som legges i databasen konverteres slik at de får stor forbokstav i hvert ord. Dette gjøres på grunn av at alle data som kommer inn er i kapiteler og dette forringer leseligheten på websiden. I tillegg til å legge nåværende låt i databasen skrives det også til en tekstfil som flash-playeren leser i fra. Denne filen inneholder 4 variabler: nåværende artist og låt, og neste artist og låt. I denne filen sendes det ut en header til nettleseren slik at Flash playeren skal bruke gammel informasjon. Når tekstfilen skrives så settes dette i headeren til filen:

```
<?
header("Expires: ".gmdate("D, d M Y H:i:s")."GMT");
header("Cache-Control: no-cache, must-revalidate");
header("Pragma: no-cache");
?>
```

Dette gjør at flash-playeren oppdateres umiddelbart når filen endres.

Det hadde vært ønskelig at webserveren hadde lest direkte fra Dalet slik at vi ikke hadde brukt pythonscriptet. Men firewall/sikkerhetsreglene hos NRK er slik at webserver og internettet er adskilt. Slik at den eneste protokollen som er åpen er http på port 80. Derfor leser pythonscriptet xml-filen og videresender den til webserver via en post metode ved hjelp av http-protokollen. Det ble diskutert en ftp-løsning, men det var dessverre ikke støtte for dette i den gjeldende versjonen av Dalet.

8.4.5 Utvikling av brukerdelen

Bruker autorisasjon

Alle brukere som er inne i community må være logget på. Dette skjer ved hjelp av PHP 4.0 sin session-funksjon. Dette gjør at serveren kan ha en oversikt over hvilke brukere som er innlogget. Scriptet som styrer dette heter log-in.php. I dette scriptet gjøres det en sjekk på om bruker allerede har opprettet en session, det vil si at han er logget inn. Ellers så gjøres det en sjekk mot databasen:

```
SELECT BrukerID FROM BRUKERE WHERE Brukernavn = '$Brukernavn' AND
Passord = '$Passord'
```

Hvis denne spørringen returnerer et resultat så blir brukeren logget inn og kan forsette å se på mPetre.community sine sider. Hvis brukeren skriver inn feil passord eller brukernavn så vil session bli slettet og bruker får vist et nytt logg på-vindu.

```
if (mysql_num_rows($result) == 0)
{
    session_unregister("Passord");
    session_unregister("Brukernavn");
}
```

8.4.6 Utvikling av innloggede brukere

Vi ønsket å ha en oversikt over hvilke brukere og hvor mange som er innlogget på løsningen til en hver tid. For å kunne ha denne funksjonaliteten, ble det opprettet en egen tabell (SESSIONS). I denne tabellen legges brukernavnet og tidspunkt inn. Når en bruker har vært inaktiv i mer en 10 minutter så fjernes brukeren, og er dermed ikke innlogget lengre.

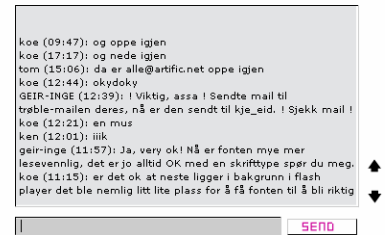
8.4.7 Glemt passord og brukernavn

Det ble laget en funksjon for å sende ut brukernavn og passord på epost hvis brukeren hadde mistet dette. Brukeren kan da velge om han vil fylle ut brukernavn eller epost-adresse. Det gjøres så en sjekk i PHP-scriptet om brukeren finnes, og i såfall sendes informasjon til brukeren ved hjelp av epost. Eksempel viser kode for glemt brukernavn.

```
if($BrukernavnBrukerp != "")
{
    $sql = "select * from BRUKERE Where Brukernavn = '$BrukernavnBruker'";
    $res = mysql_db_query($db, $sql);
    $rad = mysql_fetch_array($res);
    $ant = mysql_num_rows($res);
    $BrukernavnBruker = $rad["Brukernavn"];
    $PassordBruker = $rad["Passord"];
    $Epost = $rad["Epost"];
    if($ant > 0)
    {
        mail($Epost, "Brukernavn og passord på mPetre", "Hei\nHer er dine
opplysninger for å logge inn på mPetre:\nBrukernavn:
$BrukernavnBruker\nPassord: $PassordBruker\n\nmvh mPetre");
        print "<p>Passord er sendt til $Epost</p>";
    }
    else
    {
        print "<p>Dessverre så finnes ikke dette brukernavnet i vår
database. Du kan lage en ny bruke ved å klikke på linken ny bruker</p>";
    }
}
```

8.4.8 Utvikling av chat

Chat er basert på en flash-movie. Denne flash-movien kommuniserer med et PHP-script. Flash-movien laster dette PHP-scriptet med jevnlig mellom rom og ved trykking på send-knappen. Php-scriptet oppdaterer tekstfeltet i flash-filen slik at det inneholder siste informasjon.



Figur 8-6 Chatten i bruk

Flash - fil

Flash-filen består av to tekstfelter , en scrollbar og en send-knapp. Tekstfeltet i senter mottar informasjon i fra php-scriptet. Mens det nederste tekstfeltet sender informasjon til php scriptet ved hjelp av actionscript. Dette skjer først når bruker trykker på send.

Script som sender tekst til PHP:

```
on (release, keyPress "<Enter>")
{
    loadVariablesNum ("chat.php", 0, "POST");
    message = "";
    Selection.setFocus ("message");
}
```

Script som henter teksten inn i tekstfeltet:

```
loadVariablesNum ("chat.php", 0, "GET");
```

PHP - script

PHP-scriptet mottar en variabel i fra flash. Denne variabelen blir behandlet, dvs at HTML-tagger m.m blir fjernet. Så skrives variabelen til en fil. Denne filen inneholder også alle tidligere kommentarer i chatten. Når filen overstiger en viss lengde så blir de eldste kommentarene slettet. Dette gjøres for at nedlastningstiden av filen ikke skal bli for lang.

8.4.9 Utvikling av poskort

Postkort på nett var noe mPetre redaksjonen ønsket. Vi vurderte flere mulige løsninger. Et alternativ var en løsning hvor vi utviklet en flash-fil med en del elementer hvor brukere selv kan skrive inn en tekst som vil vises i det ferdige postkortet. Denne løsningen ville basert seg på en flash-fil og en lagring i f. eks i en database. Det vil da ikke være mulighet for å sette inn bilder fra bruker på grunn av at dette støttes ikke av flash alene. Vårt andre alternativ var å finne en programvare-komponent som kunne sette inn bilder fra brukere inn i flash filen. Vi undersøkte litt rundt om dette, og fant ut at vi i hovedsak hadde to muligheter. Macromedia genarator og Swift-tools. Valget falt på Swift-tools p.g.a. av kompleksitet og pris. Men Swift-tools alene er ikke nok for å generere dynamiske flash-sider. Derfor måtte vi utvikle en kommunikasjon mellom flash og PHP. Denne programvarebiten har vi kalt FlashGen.

FlashGen - class

For å bruke Swift-tools effektivt via PHP så har vi utviklet en egen klasse i PHP(objekt orientert programmering) som kommuniserer med dette programmet. Denne klassen har en del funksjoner som kan kalles på et objekt opprettet av denne klassen.

Funksjonskall til klassen:

```
$Flash = new FlashGen;
Lager et nytt objekt av klassen
$Flash->sett_valg($valg);
Setter en del valg som til objektet. Blant annet hvor swift programmet finnes
$Flash->sett_var("tekst","tekst streng");
Setter variabler i Flash Templatefilen
$Flash->bytt ut("7",$stekstkommandoer);
Bytter ut tekst i Flash Templatefilen
$Flash->ny_linje("DEFINE IMAGE \"robotest.gif\" -quality 30
\"robotest.jpg\");
Setter inn en ny linje i kommando dokumentet til swift-tools. Her byttes et bilde ut i Flash

Skriver flash filen og returnerer dette binært.
```

8.4.10 Utvikling av flash-player

Flash-playeren baserer seg på importen av XML i fra Dalet. I importfunksjonen punkt 8.4.4 så genereres det en tekstfil som legges på server. Denne tekstfilen leser flash-playeren vært 10 sekund. Så hvis det er kommet en ny fil så vil flash playeren vise dette innholdet ut til bruker uavhengig om han oppdaterer websiden. Intervallet ble satt på 10 sekunder slik at belastningen på server ikke skal bli alt for stor, da denne flash-filen vil sende en request til server hvert 10 sekund i fra alle brukere.

8.4.11 Utvikling av administrasjonsmodul

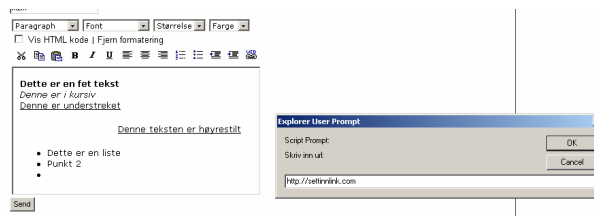
Administrasjons modulen skal styre informasjon og brukere på mPetre.Community.

Passord beskyttelse

Administrasjonsmodulen er passordbeskyttet ved hjelp av en .htaccess fil. Denne filen blir generert ved hjelp av en applikasjon i Apache(htpasswd). Htpasswd-funksjonen tar i mot et brukernavn og et passord som krypteres og legges i en angitt fil. .htaccess-filen inneholder en referanse til denne filen. Hver gang en bruker da prøver å nå en fil i den passordbeskyttede katalogen så må han skrive inn riktig brukernavn og passord.

Nyheter og artister

mPetre redaksjonen ønsket seg et kraftig og enkelt publiseringsverktøy. Kompetansen om HTML i redaksjonen er ujevn i tillegg til at publisering helst skal skje hurtig. Med disse momentene i bakhodet har vi utviklet en WYSIWYG editor til Internet Explorer nettlesere. Den er utviklet på basis av et eksempel på Microsoft developer sine sider.



Figur 8-7 WYSIWYG editoren på adminsiden.

Editoren har mulighet for å sette inn overskrifter, stille skriftstørrelse og farge, velge fet, kursiv og understrek, lime inn fra andre programmer, klippe ut, sette innrykk, lage lister, plassere tekst til høyre, senter og venstre og mye mer. Editoren er utviklet med javascript og visual basic script. Disse scriptene er integrert i en php fil for enklere bruk. Ved hjelp av tre funksjonskall i php, vil det bli generert en slik editor.

Denne editoren fungerer bare i Internet Explorer (5 og høyere). Dette skyldes at noen av komponentene i editoren baserer seg på en del av Internet Explorer sin funksjonalitet.

Brukeradministrasjon

Som administrator kan man se på, endre og slette brukere. Brukere listes etter brukernavn. Ved å klikke på en bruker kan man også se på brukerens spillelister.

8.4.12 Preloader i Flash

Vi utviklet en preloader i flash som viser nedlastningshastighet, hvor lang tid det er igjen og størrelse på Flash-filen. Denne ble kodet i actionscript.

```
onClipEvent(load) {
    //henter total størrelse på flash movie
    filstørrelse = root.getBytesTotal();
    starttid = getTimer();
    startBytes = _root.getBytesLoaded();
    // sett bar til 0
    this.xscale = 0;
}
onClipEvent(enterFrame) {
    // se hvor mye som er lastet
    nedlastet = _root.getBytesLoaded();
    // finn ut hvor mye som er lastet
    totaltlastet = nedlastet/filstørrelse;
    // konverter til prosent
    prosentlastet = int(100*totaltlastet);
    // skaler statusbar
    this._xscale = prosentlastet;
    // få nedlastnings hastighet
    tidsalang = getTimer() - starttid;
    hastighet = nedlastet/tidsalang;
    // tid igjen å laste
    bytesigjen = filstørrelse - nedlastet;
    tidigjen = (bytesigjen/1000)*(hastighet/1000);
    // lag et desimaltall av det
    hastighet = int(10*hastighet)/10;
    // skriv til tekst felter
    _root.bytestekst = int(nedlastet/1000) + "kb/" +
int(filstørrelse/1000) + "kb";
    root.hastighettekst = hastighet + "k/sec";
    root.tidtekst = int(tidigjen) + " sekunder igjen";
    // se om den er ferdig lastet
    if (totaltlastet >= 1.0)
    {
        _root.gotoAndStop(2);
    }
}
```

Denne koden legges på et objekt på stage`n. I tillegg må det være et actionscript i tidslinjen som stopper filmen.

Kode på frame en:

```
Stop();
```

9. Testing

9.1 Planlegging

Vi planla å ha kontinuerlig testing av siden underveis i prosessen. Vi følte det var viktig å få høre synspunkter fra både Upunkt- redaksjonen og mPetre- redaksjonen angående brukervennligheten både på brukerdelen og adminsystemet, siden de hadde gitt oss såpass frie tøyler når det gjaldt utforming av site og design. Særlig var det viktig for mPetre å komme med innspill på admin delen, siden det faktisk var de som skulle bruke den senere.

9.2 Gjennomføring

Testingen gikk veldig greit, ettersom vi la ut siden på en server som kunne nåes fra internett, og vi hele tiden jobbet mot denne. Der kunne begge redaksjonene hele tiden logge seg på og sjekke arbeidets gang. Dermed kunne de komme med innspill med en gang de oppdaget noe som kunne forbedres. Dette gjaldt også admin delen, og vi oppfordret de som skulle bruke den senere til å sitte å leke seg med systemet. Vi hadde også et møte med mPetre, der vi satte de ekstra grundig inn i systemet. I tillegg viste vi siden til folk i klassen, og bad disse komme med tilbakemeldinger om ting de følte kunne gjøres bedre.

En av de tingene som ble endret etter en testperiode var det å se hvilken dato en befant seg på når en skulle se på hvilke sanger som hadde blitt spilt. Vi hadde bare ramset opp alle datoene etter hverandre, og merket den datoen du befant deg på med rødt. Dette var vanskelig for brukere å se, slik at vi måtte gjøre en endring på dette. Resultatet ble at vi har datoen for gjeldende søk som overskrift, slik at dette skal fungere som et blikkfang..

Det kom også tilbakemeldinger om at når man skulle velge ut sanger til brukers spillelister, så var dette lite brukervennlig. Man måtte gå inn i lista over sanger, trykke på den sangen man ville ha, og da havnet man på førstesiden igjen. Det betød at for hver sang de ville ha med i spillelisten, måtte man foreta et søk etter nye låter. Dette var en tungvint løsning. Dette endret vi slik at det er avhukningsbokser ut for hver sang. Når man har foretatt et søk, kan man huke av flere sanger om gangen, slik at man slipper å søke på nytt for hver sang.

10. Resultater

Underveis har vi hele tiden prøvd å følge de retningslinjene og planene som vi bestemte i forprosjektrapporten. Der kalte vi oppgaven en multimedia-portal for radiokanalen mPetre. Dette har vi gått litt tilbake på. Under prosjektets gang har det blitt mer og mer klart at det er et community de har ønsket seg, der brukere må registrere seg for å bli en del av dette. Vi følte også at en multimedia-portal skal være fokusert rundt de forskjellige aspektene rundt multimedia, slik som film, bilder, lyd og interaktivitet, og der brukere skal kunne søke etter slike ting på nettet, og legge sine egne arbeider ut til allmenn beskuelse. Slik som resultatet er blitt, har det blitt mer en site fokusert rundt radiokanalen mPetre, og ikke nødvendigvis mot multimedia.

Selv om dette har blitt endret, har vi fortsatt med de forskjellige tingene vi nevnte i forprosjektrapporten. I tillegg har vi en seksjon som heter moro. Her har det kommet inn et lite flashspill som vi har lagd, og muligheten til å sende et postkort. Dette har vi hatt tid til å legge inn, ettersom vi hele tiden har ligget litt foran tidsskjemaet med de mest tidkrevende arbeidsprosessene.

Designet har blitt videreutviklet og endret på under hele prosessen, som den eneste delen som har tatt lengre tid enn planlagt. Utgangspunktet var at dette skulle bli ferdig i begynnelsen av april. Hoveddelen var ferdig til da, men det har hele tiden kommet tilbakemeldinger om endringer fra NRK, angående hvor de forskjellige tekstboksene skulle være og se ut, hvilke farger vi skulle bruke o.s.v..

Vi er fornøyde med administrasjonsdelen for siden, og det er også de som skal jobbe med den senere, mPetres redaksjon. Det er denne de skal bruke for å legge inn tekster og bilder inn på de forskjellige sidene på siden. De har altså ikke mulighet til å legge til flere sider i siden, eller endre layouten fra disse adminsidene. For å gjøre det, må man gå inn i koden som ligger i de forskjellige filene. Men dette var aldri meningen heller, det systemet skal bare styre innholdet på de sidene som er der fra før.

Vi har også hele tiden tenkt på strukturering av koding og oppbygging av site-filene. Dette skal sørge for at videreutvikling av siden skal kunne skje enkelt og at utenforstående skal kunne ta over senere. XML til flash har fungert bra. Noen problemer var det i begynnelsen, men dette var en feil som lå i et script på en maskin på Marienlyst. I tillegg vil det ikke fungere da det er Ti i skuddet-sending. Dette er fordi programet blir avvirket fra en annen dalett-server.

11. Konklusjon

Har vi klart å svare på det vi skulle gjøre i forhold til problemstillingen. Dette er i alle fall et sikkert ja på den første delen. Vi har kommet vel i havn med å lage det samfunnet som vi hadde planlagt å gjøre, og vi føler vi har lyktes med både designet og funksjonaliteten. Men foreløpig er det vanskelig å få svar på om det vil øke populariteten, og få flere besøkende enn dagens site. Dagens nettsider er godt besøkte, men har ikke all den funksjonaliteten som vi har fått til i vår oppgave. Samfunnet er ikke lansert enda, og datoen for dette er heller ikke bestemt. Derfor har vi ingen reelle tall å legge på bordet. Men vi mener allikevel at svaret på dette er et ubetinget ja. Både vi og arbeidsgiver er veldig fornøyde med det resultatet som foreligger.

Hvorfor kommer denne siden til å slå ann blant brukerne?

- De har muligheten til en hver tid se hva som spilles på radiokanalen. Brukerne kan også gå inn der og se hvilke sanger som ble spilt f.eks dagen før. De hørte kanskje en sang rundt kl 15.30 som de likte, og kan gå inn i listene og se hvilke sanger som ble spilt rundt de tider. Kanskje finner man igjen sangen man likte.
- Man kan, med utgangspunkt i sanger som er spilt på kanalen, lage sine egne spillelister. Dette kan brukes i avstemninger, konkurranser og liknende, slik at akkurat din spilleliste kan bli kåret til den beste brukerlisten. Kanskje er det din liste som blir spilt på radioen.
- Man har muligheten til å snakke med de andre brukerne på en chat. Man kan også være heldig å treffe en av programleiderne der.
- Designet har blitt mye lysere, og fått mere luft enn det gamle designet av mPetres hjemmesider. Dette er noe vi tror kommer til å slå godt ann blant brukerne.
- Det skal være en seksjon med spill og moro der. Her skal det komme forskjellige spill. Det er det alltid gøy å prøve seg på. Man har muligheten til å sende postkort til venner med et bilde av deg selv, mikset sammen med et bilde av Christine. Man fyller ut det hun skal si, og sender det til en e-post adresse.
- Bare det at man er en del av noe, der det er mange brukere på din egen alder, og med dine egne interesser, er en ting som kommer til å slå positivt an blant ungdom.

12. Videreutvikling

Her er det store muligheter for videreutvikling, og oppdragsgiver har allerede snakket mye om dette. Og som nevnt tidligere, er det alltid noe å arbeide med på en slik site som dette her, og som skal ha den funksjonen denne skal. Det er alltid rom for utbedringer og endringer, og også å legge til nye elementer. En rekke interessante forslag angående utviklingen videre har kommet på banen, og disse er:

- Knytte låt- og artistlistene opp mot Inpoc sin server for søk etter ringetoner og logoer av favorittartisten. Dette er veldig enkelt å få til, siden vi har artistnavn og eventuelt sangnavn, og bare kan gjøre et søk med disse parameterne i Inpoc sin server.
- Gi brukere mulighet til å laste opp et bilde av seg selv. Dette krever oppfølging, slik at tjenesten ikke blir misbrukt og støtende bilder blir lastet opp.
- Muligheten til å sammenlikne spillelistene med andre brukeres lister. På den måten kan en se om andre brukere har samme musikksmak som deg, eller absolutt ikke har samme smak som deg.
- Om du har sett en bruker med samme interesser/musikksmak som deg selv, kan det være interessant for brukere å ha muligheten til å sende private meldinger til hverandre, uten å måtte gå veien om e-post adressen.
- Legge inn et felt der brukere kan skrive litt om seg selv og interesser de har.
- Personlig gjestebok til brukerne.
- Diskusjonsforum der man kan diskutere ulike tema. Der skal det være mulig å legge inn nye innlegg og svare på innlegg.

13. Egenvurdering

Alt i alt er vi fornøyde med egen innsats. Vi føler at vi har klart å lage det produktet vi satset på å gjøre i forprosjektrapporten, og i forhold til problemstillingen. Vi har lært oss mye på veien mot ferdig produkt, både når det gjelder den tekniske biten, og selve organiseringen av oppgaven. Det er selvfølgelig ting vi kunne gjort bedre når vi nå i ettertid ser tilbake på denne prosessen, men nå har vi lært til neste gang vi skal jobbe med liknende prosjekter. Vi har tross alt lært mye underveis, og satt ikke på all kunnskapen i begynnelsen. Tom og Erlend har blant annet tatt faget Klient- og serversideprogrammering parallellt med hovedprosjektet, og det er jo et fag som går ut på mye av det samme som denne hovedoppgaven har gjort. Kunnskap om flash var noe vi måtte lære på egenhånd. Her var bøker fra biblioteket og internett gode ting å ha.

Vi er også veldig fornøyde med behandlingen og oppfølgingen vi har fått fra NRK, og er veldig glade for at vi har fått jobbet for dem. Dette er noe som forhåpentligvis kommer til å se bra ut på en CV.

13.1 Erfaringer fra gjennomføringen

Vi har gjort oss mange nye lærdommer i prosessen mot ferdig produkt, og disse skal vi ta med oss videre i vår karriere innenfor prosjekter og jobb generelt.

- Samarbeidet i gruppa har gått veldig bra. Vi har kommunisert bra og alle har vært med på å ta avgjørelser. Dette er nok fordi vi har vært samlet under hele prosessen, på grupperommet vårt. Det er lettere å kommunisere ansikt til ansikt enn over telefon og mail.
- Det er veldig lurt å dokumentere arbeidet underveis i prosessen, når du faktisk gjør tingene. Dette kan f.eks. være det å skrive møtereferater. Dette har vi i stor grad gjort på møter med NRK, men kunne ha vært litt flinkere til det på interne møter i gruppa. Det er alltid vanskelig å huske på ting senere. Ta også vare på alle regninger, billetter o.l. som du bruker i forbindelse med prosjektet. Dette er viktig å ha til budsjettet.
- Det er en tidkrevende prosess å sette sammen rapporten. Den er skrevet av tre forskjellige personer som tenker ulikt, og bruker ulikt språk. Det er viktig å få satt sammen denne i fellesskap, og få gjøre endinger slik at det blir en helhetelig rapport med god flyt, og der ikke samme ting står flere steder.

- En god kravspesifikasjon er bra og viktig å ha klar så tidlig som mulig. Denne skal idéelt sett ha vært godkjent av oppdragsgiver før arbeidet med oppgaven blir satt i gang. På dette punktet kunne vi nok ha vært litt bedre. Vi skrev ned i forprosjektet hva oppgaven skulle gå ut på, men burde ha lagt en mer utfyllende kravspek. tidligere. Samtidig er det vanskelig å lage en veldig detaljert kravspesifikasjon tidlig på en slik oppgave som dette. Det kommer hele tiden tilbakemeldinger om forbedringer og endringer underveis i arbeidet, slik at et endelig innhold/oppsett ikke kommer før sent i prosessen.
- Også forprosjektrapporten må det legges ned mye arbeid i. Men det kan være vanskelig å se fram over i tid med tanke på tidsbruk og arbeidsfordeling. Og om det går helt annerledes enn en tenkte, er det lett å kritisere seg selv for ikke å ha lagt ned nok arbeid i forprosjektet. Men det kan være mange årsaker til at slikt skjer, og det er viktig å dokumentere de avvikene som skjer. Et eksempel fra vår oppgave var arbeidet med å få xml til flash. Dette hadde ingen gjort før, og vi visste derfor ikke hvor lang tid dette ville ta. I vårt tilfelle tok det mye kortere tid enn vi først antok.
- Når en arbeider med en oppgave med mye programmering, er det viktig å strukturere denne bra. Også oppbyggingen av de forskjellige site-filene er det viktig å være bevisst på så tidlig som mulig. Dette er noe vi har vært flinke til. Dette skal være åpent for innsyn, og det skal videreutvikles av andre enn oss senere. Er dette gjennomført, blir det lettere for de å sette seg inn i det som allerede er gjort.
- Når en samarbeider med en stor ekstern oppdragsgiver som NRK er det mange å ta hensyn til. Mange skal ha et ord med i spillet. Derfor har det vært en bevisst holdning fra Geir Inge Løvstad i NRK å ikke presentere siden for alt for mange medarbeidere i organisasjonen for tidlig.

14. Litteraturliste

14.1 Bøker

PHP for Flash av Steve Webster (Friendsoft)

Flash 5 av Robert Reinhardt og Jon Waren Lentz(Bible)

Macromedia Flash Actionscript for fun and games av Gary Rosenzweig(QYE)

Foundation Flash 5 av Sham Bhangal, Amanda Farr og Patrick Rey (Friendsoft)

Flash 5 Actionscript studio av Davis Beard, Michaels Bedar, Sham Bhangal, Richard Chu, John Davey, Justin Everett-Churcj, Jamie MacDonald, Jose R. Rodriguez og Adam Wolff

14.2 Nettsider

<http://php.net>

<http://phpbuilder.com>

<http://groups.google.com>

14.3 Figurliste

Figur 1-1 Prosjekthjemmesiden.....	7
Figur 1-2 Mappestruktur på filsever.....	7
Figur 5-1 Adobe Photoshop i bruk med utvikling av et design forslag.....	20
Figur 5-2 Macromedia Flash 5.0 i bruk ved utvikling av chat.....	21
Figur 5-3 Macromedia Dreamwaver i arbeid med vår sitefil.....	22
Figur 5-4 Putty software.....	23
Figur 7-1 Mappestruktur på websever.....	26
Figur 7-2 Adminisidene er enkle og oversiktlige.....	28
Figur 7-3 Første utkast.....	29
Figur 7-4 Samme utkast men litt annet innhold.....	29
Figur 7-5 Andre utkast.....	30
Figur 7-6 Andre utkast med søking etter låter.....	30
Figur 7-7 Tredje forslag.....	31
Figur 8-1 Sidekart for brukersidene.....	32
Figur 8-2 Sidekart for administrasjonsmodulen.....	32
Figur 8-3 ER-Modell for MySQL databasen.....	33
Figur 8-4 Viser oppbygning av sitefilen.....	39
Figur 8-5 Viser XML-flyt fra Dalet til bruker.....	41
Figur 8-6 Chatten i bruk.....	44
Figur 8-7 WYSIWYG editoren på adminsidene.....	45

15. Ordforklaringer

J i n g e l

En kort lydreklame for et program eller arrangement.

C o m m u n i t y

Et samfunn på internett hvor brukere med felles interesser kan kommunisere med hverandre, og eventuelt andre. I dette tilfelle altså programlederne.

W Y S I W Y G

What You See Is What You Get. Dette er en måte å lag et grensesnitt på slik at det brukeren ser, er hva resultatet blir.

M u l t i m e d i a

Flere medier smelter sammen til ett. Det vil si publisering av f. eks lyd og bilde.

P a r a m e t e r

Innenfor programmering menes det en variabel som sier hvordan funksjonen skal utføre en oppgave eller hva funksjoenen skal gjøre.

16. Index

Actionscript.....	21
Administrasjon	45
Adobe Photoshop.....	20
Apache	45
Cascading Style Sheets	<i>Se CSS</i>
Chat	25, 44
Community	24
CSS	14
Dalet.....	24, 41
Databaser.....	17
Datatyper.....	33
Decimal.....	34
Enum	34
ER-modell.....	33
flash player	45
FlashGen	44
Font.....	24
Forprosjekt.....	11
Gruppas effektmål.....	5
HTML.....	14
Hypertext PreProcessor.....	<i>Se PHP</i>
Imagemagick.....	18
ImageReady.....	20
Int	33
Internet Explorer	15, 45, 46
Javascript	14
Kravspesifikasjon.....	24
Kvalitetssikring.....	12
Lynx.....	15
Macromedia	18
Macromedia Dreamweaver	22
Macromedia Flash.....	18, 21
Målgruppe	6
Mediumblob.....	34
MySQL	14, 17, 18, 33
Nestcape.....	15
Nettlesere	15
Opera	15

Oppdragsgiver	9
Oppdragsgivers	9
Oppgavebeskrivelse.....	5
PHP	17
Poskort.....	44
Problemstilling.....	5
Putty	23
Rapport.....	Se rapporten
Rapporten.....	6
Resultatmål.....	5
Sidekart	32
SQL	17
Swift-tools	18
Tabeller	34
Text	34
Timestamp	27, 34
Utstyr.....	19
Utviklingsarkitektur	23
Varchar	34
Visuelt design.....	27
webservere.....	18
WsFTP	22
XML	16
Year	34

17. Vedlegg
