

HOVEDPROSJEKT

ITWEB

**INTERAKTIV WEB
FOR IT-TJENESTEN, HØGSKOLEN I GJØVIK**

FORFATTER:

Anette Sæther

Dato: 21. mai 2006

Sammendrag

Tittel:	Interaktiv web for IT-tjenesten, Høgskolen i Gjøvik
Dato:	
Forfatter:	Anette Sæther
Veileder:	Øivind Kolloen
Oppdragsgiver:	IT-tjenesten Høgskolen i Gjøvik
Kontaktperson:	Anette Sæther
Nøkkelord:	Internett og Intranet, eZ publish, CMS-system, IT-tjenesten Høgskolen i Gjøvik
Antall sider:	
Antall vedlegg:	
Tilgjengelighet:	Åpen
Abstract:	Med eZ publish som rammeverk lage en web portal med fokus på brukervennlighet og hjelp-til-selvhelp. Opprette et intranet hvor IT-tjenestens ansette kan samle intern dokumentasjon, logg og statusmeldinger Fokus på enkelhet i publisering og oppdatering.

Summary

Titel: Interaktiv web for IT-tjenesten, Gjøvik University College

Date:

Author: Anette Sæther

Supervisor: Øivind Kolloen

Employer: IT-tjenesten, Gjøvik University College

Contact person: Anette Sæther

Key words: Internett og Intranet, eZ publish, CMS-system, IT-tjenesten Gjøvik University College

Pages:

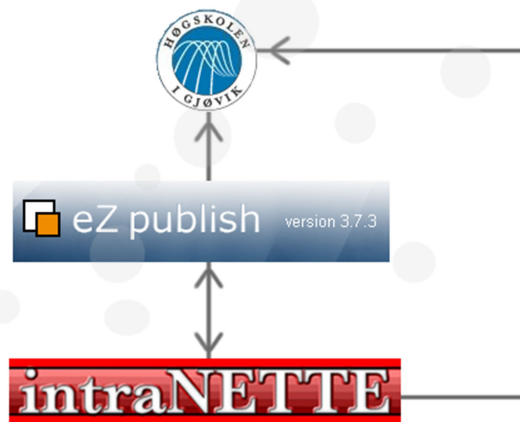
Attachments:

Availability: Open

Abstract: With eZ publish as framework, make an internetportal with focus on useability and help-to-self-help. Make an intranet where IT-services can gather internal documentation, log and statusmessages. Focus on simplicity in publishing and updating ability.

ITweb

- dynamisk web for IT-tjenesten ved HIG



Anette Sæther
03HDMUAA
2006



Forord

Rapporten er utarbeidet som en del av hovedprosjektet ved Høgskolen i Gjøvik (HIG), Institutt for informatikk og medieteknikk. Hovedprosjektet er obligatorisk i siste året i studiet og teller 20 studiepoeng. Studiet er en åpen bachelorgrad med fordypning i multimedieteknikk.

Hensikten med oppgaven har vært å få praktisert fagkunnskapen som jeg har tilegnet meg i studietiden.

I prosjektet har jeg selv valgt å arbeide alene. Dette fordi jeg ønsket å strekke prosjektperioden utover 2 semestre. Jeg har også arbeidet tett innpå IT-tjenesten og deltatt på kurs og brukergruppe møte brukergruppemøte i ett ord innen temaet som representant for dem.

Rapporten tar for seg hovedprosjektet, men er også ment som en informasjon omkring eZ publish og prosessen med å utvikle websiden og intranett med eZ som verktøy.

Ønsker å rette en stor takk til:

- IT-tjenesten, Høgskolen i Gjøvik
- Stian Husemoen, Anders Wiehe og Jon Langseth, IT-tjenesten Høgskolen i Gjøvik
- Veileder Øyvind Kolloen
- IT-avdelingen ved Høgskolen i Lillehammer ved Pål Messenlien
- Pål Kristiansen og Rune Valle

Gjøvik, 9. mai 2006

.....
Anette Sæther

Innhold

1 Innledning.....	7
1.1 Organisering av rapporten.....	7
1.2 Bakgrunn.....	7
1.3 Oppgavebeskrivelse	8
1.4 Prosjekt mål.....	9
1.5 Målgruppe	10
1.5 Faglig bakgrunn.....	10
1.6 Arbeidsformer	11
1.7 Teorier og fagemner.....	12
1.8 Standarder.....	13
2 Design og utstyr	14
2.1 Administrasjon.....	14
2.2 IT-tjenestens websiden.....	14
2.3 intraNETTE	15
2.4 Utstyr.....	15
3 eZ publish	18
3.1 eZ publish nøkkelfunksjoner.....	18
3.2 eZ i praksis.....	20
4 eZ prosessbeskrivelse.....	25
4.1 EZ publish prosessen for utvikler rammeverk.....	26
4.2 Overstyring og maler.....	34
4.7 Usergroup, User, Roller og tilgangsregler.....	47
5 Produksjon.....	51
5.1 Kode hentet fra andre.....	51
5.2 Testing.....	52
5.3 Drøfting.....	53
5.4 Realisering.....	54
6 Konklusjon.....	55
7 Litteraturliste.....	57
7.1 Skriftlig litteratur.....	57
7.2 Fra internett.....	58
Vedlegg.....	59

1 Innledning

1.1 Organisering av rapporten

Kapittel 1 – Innledning

Redegjørelse av bakgrunn, tema, målgrupper, faglig bakgrunn, arbeidsformer, terminologibruk og standarder.

Kapittel 2 – Design og Utstyr

Gir en forståelse av design, brukergrensesnitt og utstyr.

Kapittel 3 – eZ publish

Forklaring og utredning omkring eZ publish.

Kapittel 4 – eZ publish prosessbeskrivelse

Forklarer gangen i å bruke eZ publish som rammeverk. Hvordan hovedprosjekt oppgaven er løst med eZ som verktøy.

Kapittel 5 – Produksjon

Bruk av utviklingsmiljø og verktøy. Hvordan hovedprosjektet er løst.

Kapittel 6 – Konklusjon

Oppsummering av prosjektgangen.

Kapittel 7 – Litteratur

Kilder som er blitt benyttet i hovedprosjektet.

1.2 Bakgrunn

I forbindelse med sammenslåing mellom Høgskolen i Gjøvik og Høgskolen i

Lillehammer ble det behov for et nytt felles webpubliseringsverktøy. Høgskolen i Lillehammer har allerede gått over til eZ publish og det har på Gjøvik også blitt vedtatt at eZ publish skal bli det nye verktøyet for publisering.

IT-tjenesten har i dag en webside som blir lite benyttet av ansatte og brukere. Siden er lite brukervennlig og IT-tjenesten er på jakt etter nye funksjoner og en mer brukervennlig webside som samtidig skal være enkel å oppdatere. Det er også et ønske om et intranett for intern kommunikasjon mellom de ansatte.

1.3 Oppgavebeskrivelse

Forbedre hvordan hjemmesiden til IT-tjenesten bistår sine brukere. Samt lette kommunikasjonsflyten mellom ansatte og brukere ved hjelp av både intranett og webside.

1. Portere websiden fra nåværende system til eZ publish.
2. Lage en brukervennlig og funksjonell webportal.
3. Lage en webportal som representerer IT-tjenesten på en positiv og informativ måte.
4. Lage en intern webportal for ansatte på IT-tjenesten. Formålet her er å skape samhold, informere om interne nyheter og ytringer ansatte i mellom.
1. Knytte driftssystem og webside sammen, gjøre det enkelt å publisere driftsmeldinger og dermed informere brukerne bedre.

For å komme frem til hva som er en brukervennlig og funksjonell webside er det tatt utgangspunkt i tidligere skrevet rapporter om brukervennlighet på HIGs websider.

Mellom IT-tjenestens driftssystem og websiden skal det bli laget et enkelt system for publisering av driftsmeldinger. I dag benyttes RT system for utveksling av informasjon, brukerhjelp via email og noe intern kommunikasjon - dette er ønsket knyttet direkte opp mot websiden for lettere å informere brukerne.

1.4 Prosjektmål

1.4.1 Hovedmål

Lage en ny webside for IT-tjenesten, med mer fokus på funksjoner som er brukervennlig for brukerne. Skal også lette kommunikasjon mellom ansatte på IT-tjenesten og sluttbrukere ved hjelp av både websiden og intranett.

1.4.1.1 Delmål

1. portere nåværende webside til eZ publish
2. utforme ny design for websiden
3. omstrukturere og forbedre strukturen på siden, brukerorientert
4. lage et system for publisering av driftsmeldinger, som henger sammen med helpdesk systemet
5. forbedre og utvikle system for hjelp-til-selvhelp
6. lage en intuitiv og brukervennlig admin. -side for oppdatering/ endring av dynamisk innhold
7. utvikle et intranett for de ansatte hvor beskjeder og informasjon kan bli publisert enkelt med bruker autentisering

1.4.1.2 Prosessmål

1. lære å utvikle og forbedre et prosjekt i alle faser
2. få en dypere innsikt
3. holde tidsrammer som er satt
4. arbeide og strukturere et stort prosjekt
5. eZ publish rammeverk og utvikling
6. tabellfritt design hvor alt styres fra CSS
7. xhtml som standard
8. webutvikling generelt

1.4.1.3 Resultat mål

1. bedre navigering på websiden
2. enkel publisering av driftsmeldinger
3. system for hjelp-til-selvhelp
4. intranett med mulighet for intern kommunikasjon som er knyttet opp mot webportalen med mulighet for publisering

5. webportalen skal representere IT-tjenesten på en positiv og informativ måte til brukerne av siden
6. brukervennlig og funksjonell webportal
7. eZ gir en mer brukervennlig administrasjonsside, gjør også at søkemotorer vil finne siden, og URL-er

1.5 Målgruppe

Målgruppen for produktet er både de ansatte ved IT-tjenesten og brukerne (studenter og ansatte ved HIG).

1.5 Faglig bakgrunn

Grafikk

Til grafisk utforming av designet er det benyttet Adobe Photoshop og Adobe Illustrator. Har kjennskap til Photoshop fra faget «Multimedieteknikk 1» og IGMs kurs «Photoshop trinn 1»¹ tatt via Høgskolen i Gjøvik. Kunnskap om Illustrator er tilegnet under prosjektperioden.

HTML og CSS

I faget «Publisering for web» fikk man en grunnleggende innføring i HTML og CSS.

Javascript

Hadde ingen forkunnskaper om Javascript før prosjektarbeidet. Kunnskap er tilegnet under prosjektperioden.

XML

Var temaet i faget «Elektronisk publisering med XML», fikk her en grunnleggende forståelse, men uten særlig praktisk kunnskap om XML.

Programmering generelt

Hadde litt basiskunnskaper innen C++ fra faget «Grunnleggende datakunnskap og programmering». Hadde ingen forkunnskaper om klient/server-side programmering slik som med PHP. Hadde litt kunnskap om MySQL fra faget «Databaser 1», dette er heller ikke noe som er blitt kodet

¹ IGM betyr Institutt for Grafiske Medier. Holder blant annet forskjellige grafiske kurs. Mer informasjon finnes på <http://www.igm.no>

direkte mot, men ser på det som en liten bit av eZ publish kodespråket.

Ergonomi

I faget «Ergonomi i digitale medier» lærte man prinsipper bak ergonomi på internett, deriblant valg av fonter, skriftstørrelse og fargebruk.

EZ publish

Hadde ingen kunnskap om dette som programvare eller rammeverk før prosjektperioden. Innledningsvis i prosjektet deltok jeg på et «Technical eZ publish Course» på Høgskolen i Lillehammer som representant for IT-tjenesten ved Høgskolen i Gjøvik. Midt i prosjektperioden ble jeg også ansatt inn av Høgskolen for å lage nye Studiesider og internasjonalt kontors sider. Dette har ført til at mer kunnskap er tilegnet og at resultatet på IT-tjenestens websider og intraNETTE² er mer solid og brukervennlig, da spesielt for de ansatte som skal publisere.

Har også deltatt som representant for IT-tjenesten på eZ brukergruppemøte i mars. Her ble det vist web-løsninger fra forskjellige høyskoler med bruk av eZ som verktøy.

Prosjektarbeid

Har tidligere erfaring fra prosjektarbeid, men når det gjelder prosjektarbeid på denne størrelsen og med en reell oppdragsgiver har ingen erfaring. Har tidligere skrevet små og forenklede rapporter, men aldri med slik størrelse og omfang.

1.6 Arbeidsformer

Deltok på et Technical eZ publish course i oktober 2005. Dette la grunnlaget for arbeidet med eZ publish som rammeverk.

Baserte strukturen på IT-tjenestens websider på tidligere rapporter om HIG`s webside. Enkelte valg er gjort ut fra IT-tjenestens ønsker, deriblant innhold og noe oppbygning på siden. Dette på grunn av videre utvikling etter at selve websiden er gått live.

Manualen som er laget omkring publisering på IT-tjenestens websider og intraNETTE er testet mot de ansatte på IT-tjenesten med stor suksess: ingen feil eller logiske brister ble oppdaget.

Hadde arbeidsstasjon hos IT-tjenesten, arbeidet tett oppimot målgruppe og kunne ta avgjørelser direkte, uten møter og omfattende testing.

² IntraNETTE er navnet gitt intranet løsningen til IT-tjenesten.

1.6.1 Terminologibruk

eZ som kodespråk kan defineres som en blanding av C++, mysql, php, xml og html. Samtidig kan man knytte inn andre kodespråk som Javascript i koden. Dette gjør at eZ er et kraftig kodespråk med en bratt læringskurve.

1.6.2 Implementering

1.6.2.1 Verktøy

Valg av eZ publish som nytt publiseringsverktøy på Høgskolen i Gjøvik ble gjort tidlig i prosjektperioden. Tilfeldigheter omkring hovedprosjekt for IT-tjenesten og ny webside gjorde at eZ publish måtte læres.

Selve installasjonen av eZ publish, med oppsett av databasen og apache er gjort av IT-tjenesten, prosjektet tok for seg siste del av installasjonen samt utviklingen.

1.6.2.2 Utviklingsmiljø

Fikk tildelt et eget kontor hos IT-tjenesten på Høgskolen i Gjøvik. Satt der i en periode, men den siste tiden er det arbeidet hjemmefra grunnet begrensninger i IT-tjenestens åpningstider. Alt utstyr ble lånt av IT-tjenesten, de bisto også med mye hjelp hvis behov for det. Har også benyttet aktivt internett med forskjellige chatteprogrammer³ for å få hjelp hvis problemer oppsto, og da av Pål Messenlien på Høgskolen i Lillehammer. Da ingen på Høgskolen i Gjøvik har noe kunnskap om eZ publish var dette kritisk for å komme i mål med prosjektet.

1.7 Teorier og fagemner

I hovedprosjektperioden er det blitt lært mye omkring det å arbeide på større prosjekter, loggføring, møtereferater, statusrapporter og ikke minst sluttrapport.

Har også lært mer inngående CSS og å lage et tabellfritt design. I tillegg har det blitt en dypere forståelse for programmering og drift av større systemer. Det er også positivt å ha lært eZ publish som verktøy. Det å planlegge og utvikle et større system har vært en stor utfordring, gøy og veldig lærerikt.

³ Program som kjører over internett hvor brukere kan snakke seg imellom fra pc til pc.

1.8 Standarder

Kodespråket benyttet for sidene og malene er med HTML i bunn, eZ publish koden er deretter flettet inn i denne. Designet er et tabell fritt CSS design.⁴

Kommentarer er i koden skrevet på denne måten:

<code>{* Kommentar *}</code>	For maler og mot eZ
<code>/* Kommentar*/</code>	I php filer i denne rapporten
<code><!-- Kommentar --></code>	Pagelayout filen

Det meste av arbeidet i prosjektet har vært i å programmere maler for visning. Et eksempel fra kodespråket til eZ er den som ofte er benyttet for å hente ut innhold fra en plass og fortelle hva det skal vises med:

```
{*Henter ut høyrefeltet til veiledninger, vise med linje visningen veiledninger*}
{let children=fetch( content, list, hash( parent_node_id, $node.node_id,
                                         sort_by, $node.sort_array,
                                         class_filter_type, include,
                                         class_filter_array, array( 'hoeyrefelt' ) ) )}
    {* Loop for hvert child av noden*}
    {section name=Child loop=$children}
    {* Vise innhold med line view *}
    {node_view_gui view=veiledninger
content_node=$Child:item}
    {* Slutt loop *}
    {/section}
{* Slutt namespace *}
{/let}<br/>
```

Denne koden kan varieres på en rekke forskjellige måter utifra hvilke steder man vil hente ut data fra, hvordan det skal sorteres og hva som skal hentes ut.

I rapporten har admin, administrasjonsgrensesnitt og administrasjon samme betydning. De refererer alle til administrasjonens siden til IT-tjenestens webside.

Templater og maler er betegnelser ofte brukt i rapporten. De har begge samme betydning. Det samme gjelder override og overstyring. Folder og mappe er også det samme.

⁴ Mange websider styrer designet via tabeller. Denne siden styrer alt ut i fra CSS og ikke tabeller.

2 Design og utstyr

2.1 Administrasjon

Webgrensesnittet for administrasjon er gitt av eZ publish men man kan sette mange begrensninger på rolle, brukergruppe og brukernivå. Dertil hvilke siteaccess⁵, klasse og objekt brukeren får lov til å redigere på.



2.2 IT-tjenestens websiden

Den største utfordringen gjennom hele prosjektet var å lage websiden. Dette var fordi det var et krav å ha det samme design som tidligere, men uten tabeller. Løsningen som er benyttet med Eway⁶ har en omfattende CSS og baserer hele designet på bruk av tabeller.

Det nye designet skulle være helt uten tabeller med all design i CSS. Det tok lang tid å sette opp, men resultatet ble veldig bra. Fordelen med å benytte et tabellfritt design er følgende:

1. bruker kan deaktivere CSS og fortsatt lese designet på sidene
2. svaksynte kan forstørre tekst uten at det påvirker innholdet
3. blinde/svaksynte kan bruke taleoppleser for å få fullstendige setninger

⁵ Siteaccess – det er en for hver webside publisert. HIG_it-tjenesten, HIG_admin, intraNETTE og intraNETTE_admin.

⁶ Eway er en portal for samarbeid i en organisasjon og/eller med organisasjonens partnere og kunder. Mer informasjon finnes på <http://www.eway.no>

4. bruker kan benytte eget CSS



2.3 intraNETTE

Designet på intraNETTE er gjort veldig enkelt. Websiden er designet for enkel navigering og for å kunne innholde store mengder data. Jeg Sto også her fritt til å gjøre hva som helst med designet da den ikke behøvde ha noen visuell sammenheng med IT-tjenestens webside.

Som utvikler ville jeg sette et humoristisk og personlig preg (Anette-intraNETTE) på siden, da IT-tjenesten, som er de eneste som kommer til å bruke denne siden, er en uhøytidelig gjeng som satte stor pris på det.

Denne siden har ikke mye ekstra funksjonalitet da den skal benyttes som arkiv og intern kommunikasjon og er derfor gjort så enkel som mulig, men bunn solid med gode navigasjons muligheter.



IntraNETTE er XHTML 1.0 Transitional Validert. ⁷

2.4 Utstyr

2.4.1 Datateknisk utstyr

Fordelen med eZ publish er at man ikke behøver noen fast arbeidsstasjon, så sant man utvikler mot direkte mot serveren. I begynnelsen av prosjektet ble

⁷ Validert via <http://validator.w3.org/>. Når XHTML er validert vil det si at den er godkjent i følge standarden for XHTML.

det arbeidet på en lokal installasjon av eZ på hovedarbeidsstasjonen. Senere i prosjektet, når arbeidet gikk rett mot serveren, ble ulike stasjoner benyttet.

Hovedarbeidsstasjon:

Dell Optiplex GX260
Microsoft XP Service Pack 2
Intel Pentium(R) 4 CPU 1.80GHz
512MB RAM

Biarbeidsstasjon:

iBook G4 Mac OS X
Versjon 10.4.5
1.33Ghz PowerPC
1GB RAM

2.4.2 Programvare

En rekke programmer ble benyttet under utviklingen, designet og dokumentasjons prosessen.

- Firefox med Developer ekstensjon; Testing, CSS og Design.
- Notepad ++; Koding av templater, CSS og HTML.
- Open Office; Rapportskrivning og dokumentasjon.
- Adobe Photoshop; Bilderredigering.
- Adobe Illustrator; intraNETTE banner.
- Macromedia Fireworks; Design testing av intraNETTE samt enkelte Javascipt koder
- Internett Exploter 6.0; testing av websiden
- Opera: testing av websiden
- eZ publish: lokal installasjonspakke, utvikling og testing
- winSCP: oppkobling til server

2.3.4 Server

Selve serveren og oppsettet omkring den, har IT-tjenesten selv stått for da de ønsket å ta seg av driftingen.

- eZ publish versjon 3.7.3
 - Online Editor ekstensjon⁸ (ezdhtml)
 - Shuffle ekstensjon
 - ezlocations ekstensjon
 - ezrss ekstensjon

⁸ Ekstensjon – en liten pakke man laster ned fra internett og legger til på serveren. På ez.no er det mange som deler det de har laget gratis. Noe koster også penger slik som Online Editoren for tekstformatering.

- PHP versjon 4.3.9
- Webserver Apache/2.0.52 (CentOS)
- Webserver maskinvare Intel(R) Xeon(TM) CPU 3.20GHz 3192.512 Mhz, 3.96GB minne
- Database mysql, karactersett iso-8859-1 (Intern)

3 eZ publish

eZ publish er et stort webpubliseringsverktøy som baserer seg på åpen kildekode og ikke minst deling. Omkring eZ publish er det et stort miljø, på deres websider kan man blant annet finne et veldig stort forum, omfattende dokumentasjon og elementer for deling.

Det er registrerte brukere av eZ i 118 land. En rekke andre høgskoler har allerede implementert eZ publish som sitt webpubliseringsverktøy. Disse kjører også regelmessige samlinger kaldt eZ edu meeting⁹, det ble sist holdt på Høgskolen i Lillehammer i april 2006. Her var jeg deltaker som representant for Høgskolen i Gjøvik, IT-avdelingen.

Grunnen til at mange velger å benytte eZ publish:

- benytter gratis åpen kildekode, er et stort miljø
- kan putte dokumenter raskt online
- statistikker blir automatisk generert
- enkelt integrere eksisterende webtjenester
- utvikle nye applikasjoner som arbeider innenfor eZ systems

eZ er i tillegg en meget brukervennlig løsning, har et uniformt design, samtidig som man sparer penger ved bruk av åpen kildekode. Det leverer dynamiske html-sider og kan kode designet inn i eZ-templater. Hver side kan enkelt ha sitt eget utseende/tema. EZ publish kan enkelt generere internettfunksjonalitet og kan administrere rettigheter på alle nivåer på en brukerrettet måte. Systemet har også innebygd funksjonalitet for oversetting mellom flere språk på en intuitiv måte.

3.1 eZ publish nøkkelfunksjoner

eZ er et åpent kildekode, innholds-, styringssystem og utvikler rammeverk med funksjonalitet for webpublisering, intranett, e-handel og mye mer.

⁹ EZ education meeting er brukergruppe møte for de som er medlem av gruppen for utdanning. Det blir holdt møter regelmessig gjennom året. Her vises det frem arbeid, deler erfaringer og deler informasjon.

3.1.1 eZ publish definisjoner

Klasser – for å definere strukturen av innholdet. Som en instruksjon på hvordan en skal lagre et spesielt type innhold. Eksempel artikkel som inneholder en introduksjon av innhold og bilde. Disse elementene av klassen blir kaldt attributter.

Objekter – representerer en del av innholdet på en side. Hvert objekt er basert på en type klasse i systemet. Eksempelvis kan man lage ett dokument basert på artikkel klasse definisjonen.

Tre – innholdstre blir brukt for å organisere data. Hver gang man lager et nytt objekt velger man en plassering for den inne i innholdstreet. Eksempelvis kan man lage en artikkel og plassere den inne i en mappe med navn nyheter. Dette gjør det enkelt å finne og navigere i innholdet.

3.1.2 eZ publish sikkerhet

Auditiv prøving/versjon kontroll – holder oversikt over alle forandringer som er gjort på innholdet. Logges for fremtidig bruk.

Innholdsgodkjennelse – tillater å sette opp systemet slik at innhold eller deler av innhold blir godkjent før publisering.

Email verifisering – eZ kan konfigureres til at man kan registrere nye kontoer i systemet. Det blir da gjort slik at eZ sender en aktiverings email med en aktiveringsnøkkel. Forsikrer at både email adressen er gyldig og at den korrekte brukeren registrerer seg selv.

Privilegier/Rollebasert regelverk – eZ inneholder et avansert privilegie system som tillater å gi privilegier på alle nivåer. Det viktigste privilegiet er på klasse nivå, objekt nivå og separate deler av innholdstreet. LDAP-autentisering¹⁰ – tilbys som plug-in.

Tilkoblingsmuligheter for autentisering – tilbys som plugg-inn. Kan bruke tredje parts database med logg inn informasjon for å autentisere brukere i tillegg til den interne databasen i eZ.

Sandbox – forhåndsvisning hvor man kan se innholdet akkurat slik det vil bli seende ut på brukersiden. Kan eksperimentere uten å ødelegge den offisielle siden.

¹⁰ LDAP autentisering betyr Lightweight Directory Access Protocol å er en nettverks protokoll for spørring og modifisering av directory service som går over TCP/IP.

SSL kompatibilitet – fanger opp SSL¹¹ og man kan velge å kjøre hele siden med SSL.

Versjon – innholds objekt system. Hver gang et objekt er redigert, en ny versjon er laget. Du kan tilbakestille til forrige versjon når du selv måtte ønske.

3.2 eZ i praksis

Innholdet i objektet blir brukt for å generere URL-er. På IT-tjenestens websider er det benyttet noden¹² som streng på URL, dette fordi en artikkel kan ha en lang overskrift, og da kan også URL-en bli veldig lang. Linkene inneholder derimot aldri lange og kryptiske strenger som kan misforstås. En webside publisert med eZ er mer lesbar for mennesker og indekserbar for søkemotorer slik som Google.

Brukere kan enkelt laste opp bilder av hvilke som helst størrelse, eZ skalierer automatisk til ønsket størrelse. Skaleringen kan enkelt forandres for å passe individuelt behov. Dette er gjort på bildene av type «small», for at de skal kunne passe inn i høyre feltet på IT-tjenestens webside.

EZ tilbyr et fleksibelt override system som tillater å brukerdefinere hvordan en klasse og objekter skal håndteres, er dermed optimalisert for opprettelsen av disse typene. Et avansert template språk som tillater å vise innholdet eksakt slik du vil, ved hjelp av state-of-the-art¹³ logikk hvis behov.

For enkelt å sette inn avanserte Office-lignende dokumenter med header, font, forandringer og bilder tilbys en WYSIWYG¹⁴ editor. Ved å bruke denne Online Editoren kan man redigere objekter på lik måte som i større Office verktøy. En av de mange fordelene med denne er at du kan kopiere inn tekst direkte fra for eksempel Word og få formateringen med. Man kan også kopiere fra internett og da få med tabeller og klasser. Editoren kan deaktiveres når man selv måtte ønske og man kan da redigere direkte i den genererte XML-koden. En negativ ting med Online Editoren er at den takler bildeplassering og linjeskift i tabeller dårlig. Dette er derimot et kjent problem og det arbeides med å løse dette problemet. Online Editoren er en modul man må kjøpe på ekstra, koster fra 850,- å oppover avhengig av hvor

¹¹ SSL betyr secure socket layer og er en kryptert kommunikasjons protokoll.

¹² Node er ID`n til objektet. Et objekt har et navn og en node. ID`n blir mye brukt i programmeringen mens navnet blir brukt i for eksempel admin for å finne igjen objektet.

¹³ State of the art betyr det ypperste innen et område.

¹⁴ What you see is what you get – hva du ser er hva du får.

stor siden er og hvordan man ønsker oppgraderingsmulighetene.

I tillegg til caching ¹⁵med HTML kan man cache hvilken som helst intern struktur, som roller og tillatelser, innstillinger og kompilerte templates. Man kan når som helst skjule/vise innhold, eZ behandler begge deler enkelt på et par klasse basiser.

eZ blir styrt totalt fra internett ved hjelp av brukergrensesnittet. Dette er optimalisert for bruk med moderne nettlesere med Javascript. Alle funksjoner skal også fungere med gamle nettlesere eller tekstbaserte nettlesere

Det er også tillatt å ha flere undersider innenfor samme innholdstre. Disse sidene kan være uavhengige med egen menystruktur osv. Er dette som er gjort med IT-tjenestens webside og intraNETTE, den har to forskjellige siteaccesser og dermed to helt separate templates, struktur og ikke minst design. Fordelen her er at administrasjonssiden er den samme og man kan kontrollere innholdet på begge samtidig og på tvers av hverandre. Man kan også ha to helt separate sider på samme eZ publish installasjon.

Når man sletter objekter, blir de lagret i søppel. Man kan gjenopprette objekter fra søppel når man måtte ønske, da med versjonsinformasjonen intakt. Søppelkurven kan også tømmes når man måtte ønske.

Via administrasjonsgrensesnittet tilbys web-basert stil/template styring. Herfra kan man styre templates og stiler i systemet etter behov. For avanserte brukere finner man den samme funksjonaliteten fra filsystemet. Gjennom prosjektperioden har jeg arbeidet på både det web-baserte- og det filbaserte systemet direkte.

Med en funksjon for arbeidsflyt kan man lage såkalte triggere, dette gjør at man kan lage brukerdefinert og genetiske funksjoner. Denne funksjonen blir benyttet når en driftsmelding blir publisert i intraNETTE. Den blir så lagret både i administrasjonssiden til intraNETTE og websiden, den blir da også publisert på websiden for sluttbruker. Her måtte det først legges inn en ekstensjon for å få funksjonene til å fungere.

En kraftig RSS ¹⁶modul følger med. Man kan eksportere innholdet på denne måten og man kan ha så mange RSS som man selv ønsker. eZ supporterer RSS import til hvilken som helst del av innholdstreet, på denne måten kan man synkronisere materiale fra en annen side og bruke den på din egen. UFT-8 support tillater å lage en multilingual¹⁷ side uten forskjellig tegnsett

¹⁵ Cache er lagring av data midlertidig. Brukeren går ikke nødvendigvis inn på original data man kopier av det i cache. Når forandringer er gjort må cache tømmes for at forandringene skal tre i kraft.

¹⁶ RSS betyr Really Simple Syndication. En form for web synkronisering av tekst ofte brukt av nyhetssider og i weblogger..

¹⁷ Multilingual betyr internasjonalisering og plassering. I denne sammenhengen språk på websiden.

for hvert språk.

Med eZ publish kan man enkelt bruke om igjen innhold og objekter på flere forskjellige steder i innholdstreet. Enhver forandring i det originale objektet vil være synlig på hver lokasjon i innholdstreet. På denne måten blir ikke bare gjenbruk enkelt, men også oppdatering.

Innholdsattributter kan lages spesifikt til brukeren slik at de tilfredsstiller et hvert behov. Dette kalles utvidet brukerprofil og er basert på klasser, akkurat som annet innhold i systemet.

Det finnes en rekke innebygde applikasjoner i eZ det opprettes automatisk databaser for disse. Dette er valg man foretar under installasjonen av eZ publish på serveren.

Eksempler er:

- blog
- kontakt
- forum
- fil distribusjon
- skjema
- link
- fotogalleri
- spørreskjemaer
- produktstyring (e-handel)
- søkemotorer
- site map¹⁸
- brukertilbakemelding

Publisering av innhold, dokumenter, filer og multimedia er basisfunksjonalitet i eZ publish.

Administrasjon med eZ blir gjort gjennom et intuitivt webgrensesnitt, ingen installasjon på datamaskinen er nødvendig. Tilgang for admin. er styrt med en kraftig, rollebasert tilgangskontroll som gir lov til å lese, redigere, publisere innhold osv. Systemet har ingen begrensning på mengde av innhold eller antall brukere av systemet. Man kan også sette opp eZ slik at man ikke behøver å gå inn i admin. for å redigere innhold, dette kan gjøres direkte fra brukersiden.

Publiseringsprosessen kan deles inn i tre trinn:

1. navigering – innholdet er organisert i en tre-struktur, brukertilgang avgjør hvilken del av strukturen man kan navigere gjennom og opprette innhold i, flytte eller kopiere innhold og publisere det samme innholdet på forskjellige steder
2. lage og redigere – velg lokasjon for innholdet og lag nytt innhold eller rediger eksisterende. Noen av standard funksjonene for redigering er:

¹⁸ Site map er et kart over oppbygningen av websiden. Typisk hirarkisk oppbygd.

1. versjonkontroll med «roll-back»¹⁹
2. søppelbøtte
3. fremtidig publisering og upubliserte
4. styring av oversettelinger
5. utkast
6. forhåndsvisning
7. relaterte objekter
8. automatisk bilde komprimering, som størrelse
9. snille URL-er²⁰
10. sjekking av døde linker
3. Innholdspublisering
 1. arbeidsflytssystem – eksempel godkjenning av det som skal publiseres med ett-klikk funksjonalitet
 2. template og design – bestemmer hvordan innholdet blir vist på websiden
 3. multimedia og dokumenter. Med eZ som rammeverk kan man utvide systemet for spesifikke behov. Standard som supporteres med pakken er:
 1. innhold – artikler, nyheter, informasjons sider, mapper, produkter, kommentarer og link katalog
 2. multimedia – alle standard formater som JPEG, GIF, video, quick time, MS mediaplayer, flash og audio
 3. dokumenter – MS Office dokumenter som word, excel, powerpoint, pdf, open office osv.

I eZ er det også integrert søkemotorer med avanserte søk, separasjon av innhold og design, support for intra, extra og internett i en løsning.

Kjernemotorer og utvikler rammeverk

Kjernen i eZ inneholder en samling generelle, fleksible søkemotorer og biblioteker. Disse bygger et kompakt, åpent og fleksibelt rammeverk med mulighet for å lage kraftige webapplikasjoner. Det er med dette rammeverket alle eZ funksjonalitet er bygd. eZ distribuerer en integrert løsning som inneholder «Out-of-the-boks»²¹ funksjonalitet og fleksibelt utviklingsrammeverks, som gir et stort rom for utvidelse og brukerdefinerte løsninger og applikasjoner.

Template-systemet er veldig kraftig og fleksibelt. Man kan brukerdefinere designet med templer for å få den eksakte profilen og man kan sette inn visningslogikk som gir omfattende muligheter. Selv om template systemet er fleksibelt, er det ikke dermed lett å lære. Gjennom hele prosjektperioden er

¹⁹ Roll-back betyr at man kan gå en versjon bakover.

²⁰ Snille URL-er er linker mennesker kan lese og forstå, ikke lange og kryptiske datastrenger.

²¹ Out-of-the-boks betyr som levert fra forhandler.

det allikevel bare skrappt på overflaten av dette. Jeg forstår det enda ikke fullt ut da systemet er stort og læringskurven er ekstremt bratt.

eZ er designet for å være et kompakt system med en åpen arkitektur. Man kan ekspandere alle deler av systemet, kan lage egne innholdsklasser, nye datatyper, moduler, templateoperatorer, funksjoner, påminnelseshåndtering, arbeidsflyt og mer.

EZ benytter seg av åpne standarder og har et åpent brukergrensesnitt for all slags integrering. SOAP²² er et av bibliotekene som er med i eZ publish, som tillater eZ å kommunisere med andre systemer via webservice.

²² SOAP er en protokoll for å dele XML-baserte beskjeder over et nettverk, typisk med bruk av HTTP.

4 eZ prosessbeskrivelse

eZ publish er en stor og omfattende webpubliseringsløsning. Læringskurven er bratt og det tar lang tid å lære seg template og override systemet. For å avklare eventuelle uklarheter og omfang er det ønskelig å gå gjennom selve prosessen for å benytte eZ som utvikler rammeverk for å lage en webside.

La oss benytte intraNETTE som eksempel da det er den enklest oppbygde siden. Utviklingen er gjort parallelt mot både admin og direkte på server, da det ble lettere å holde oversikt over hvilke filer, klasser og seksjoner som blir overstyrt.

Siden er bygd opp med header med bakgrunnsbilde, brødsmule, tre-meny på venstre side og bunntekst nederst.

hig intraNETTE

Dokumentasjon
Driftslog
Driftsmeldinger
Statusmeldinger
Søk
Vedlikeholdsskjema

RT
Open : 189
Reply : 1d, 4:15
Resolve : 2d, 9:7

Velkommen. Du er logget inn som Administrator User

[Logg ut](#)

Driftsmeldinger

Av: AS, Publisert: 04.05.2006
Ombygginger av nettverk for studenthjem

Det kan i dag forventes noe utfall i internetforbindelse ved SOPP sine studenthjem på Gjøvik, i forbindelse med ombygging av nettverkets struktur.

Av: AS, Publisert: 04.05.2006
Problemer med intertelefon mellom HiL og HiG

Det er for tiden problemer med å ringe internt mellom HiG og HiL.

Det skal være mulig å ringe med internummERMellom institusjoner i Uninett, men frem til problemet Umoe IKT arbeider med å løse er ute av verden, må vi se oss nødt til å legge til 611/612..

Av: AS, Publisert: 04.05.2006
Oppdateringer av servere

4.1 EZ publish prosessen for utvikler rammeverk

Installasjon på server

Før man kan gjøre noe som helst må man installere eZ publish på serveren. Dette tok de ansatte på IT-tjenesten seg selv av. Det er installert eZ publish 3.7.3²³ på serveren, den krever PHP 4.4 mens det på servere er PHP 4.3. Man trenger også apache og en mysql database.

Når selve installasjonen er kjørt på serveren kan man gå inn i en veiviser for installasjonen, denne går videre med siste del av eZ installasjonen. Blant annet oppsett av email, database, mal type, passord og brukernavn og ekstra funksjonalitet siden skal ha og URL håndtering (port eller URL). Den har også en Finetune funksjon hvor man kan se om det er noe feil i for eksempel en php fil eller om noe mangler.

Funksjonalitet via veiviseren har lite å si da man allikevel overstyrer det meste av dette og oppretter ny funksjonalitet. Det er allikevel greit å ta med mye funksjonalitet da dette kan benyttes som mal i opprettelsen av nye funksjoner.

Etter installasjonen kan man gå direkte til websiden og admin. for å se om det fungerer. Man vil da se en standard mal som man senere skal overstyre for å få rett design og funksjonalitet etter ønske.

Siteaccesser og fil innstillinger

Før man kan starte med selve utviklingen må man gjøre visse innstillinger, kopiere/opprette mapper og skrive om litt i diverse filer.

IT-tjenestens webside og intraNETTE har to forskjellige design og funksjonaliteter, så denne prosessen er kjørt to ganger.

Går inn i /var/ katalogen, oppretter her en mappe kaldt /intraNETTE/. Her lagres all midlertidig data og når Cache skal tømmes er det denne som blir tømt.

Navigerer seg så ut igjen å inn i /design/ oppretter her katalogen /intraNETTE/. Under den opprettes katalogene /override/, /override/templates/, /templates/, /images/ og /stylesheets/. Det er her alle malene, designet og bildene til designet kommer til å ligge.

Kobler til filsystemet på serveren, går inn i settings/siteaccess/, kopierer her /Standard/ og gir den nytt navn: /intraNETTE/. Går så inn i

²³ Siste eZ publish versjon er nå 3.8.

settings/siteaccess/intraNETTE/site.ini.append.php. Her får man informasjon om siden, her må også mye forandres.

settings/siteaccess/intraNETTE/site.ini.append.php

```
<?php /* #?ini charset="iso-8859-1"?

[DatabaseSettings]                               /*Databaseinformasjon*/
DatabaseImplementation=ezmysql
Server=mysql.hig.no
User=*****
Password=*****
Database=plain
Charset=
Socket=disabled
SQLOutput=disabled

[SiteSettings]                                   /*Her setter man site navn, url*/
SiteName=intraNETTE                             /* og index side.*/
SiteURL=ez.hig.no/index.php/intraNETTE
LoginPage=custom
IndexPath=/user/login

[SiteAccessSettings]                            /*Om man må ha innlogging og at */
RequireUserLogin=false                          /*skjulte noder skal gjemmes.*/
ShowHiddenNodes=false

[DesignSettings]                                /*Her må man sette at designet */
SiteDesign=intraNETTE                           /*skal være intraNETTE og ikke */
AdditionalSiteDesignList[]=base                 /*Standard. Har ett design i*/

                                                /*reserve kaldt base.*/

[RegionalSettings]                              /*Språk innstillinger.*/
Locale=nor-NO
ContentObjectLocale=nor-NO
TextTranslation=enabled

[FileSettings]                                  /*Dette må settes til intraNETTE*/
VarDir=var/intraNETTE                          /*og ikke base.*/

[ContentSettings]                               /*Oversetter ingenting, så denne*/
TranslationList=                                /*står tom.*/

[DebugSettings]                                 /*Om vi vil se tid, brukte */
DebugOutput=enabled                             /*maler. Noe justering*/ på hastighet
DebugRedirection=disabled                       /*går også herfra, går raskere uten*/
                                                /*debug.*/

[TemplateSettings]
Debug=disabled
ShowXHTMLCode=enabled
ShowUsedTemplates=enabled

[FormProcessSettings]                           /*Denne må fylles inn hvis man vil*/
Module=enabled                                  /*skjemaer på websiden.*/
*/?>
```

Kopierer deretter siteaccess/Standard_admin/ og kaller den intraNETTE_admin. Her må det også gjøres forandringer i /site.ini.append.php filen.

settings/siteaccess/intraNETTE_admin/site.ini.append.php

```
<?php /* #?ini charset="iso-8859-1"?

[DatabaseSettings]                                /*Samme som for intraNETTE*/
DatabaseImplementation=ezmysql
Server=mysql.hig.no
User=publish
Password=Aghx3ki6
Database=plain
Charset=
Socket=disabled

[SiteSettings]                                    /*Side navn og index side, login*/
SiteName=intraNETTE                               /*siden står som standard.*/
SiteURL=ez.hig.no/index.php/intraNETTE_admin
LoginPage=custom

[SiteAccessSettings]                             /*Må ha innlogging og her kan vi se*/
RequireUserLogin=true                             /*skjulte noder.*/
ShowHiddenNodes=true

[DesignSettings]                                 /*Designet skal være admin.*/
SiteDesign=admin

[RegionalSettings]
Locale=nor-NO
ContentObjectLocale=nor-NO
TextTranslation=enabled

[FileSettings]                                   /*Denne må forandres fra standard*/
VarDir=var/intraNETTE                             /*til intraNETTE.*/

[ContentSettings]                                 /*Innholdsinnstillinger.*/
CachedViewPreferences[full]=admin_navigation_content=0;admin_navigation_details=0;admin_navigation_languages=0;admin_navigation_locations=0;admin_navigation_relations=0;admin_navigation_roles=0;admin_navigation_policies=0;admin_navigation_content=0;admin_navigation_translations=0;admin_children_viewmode=list;admin_list_limit=1;admin_edit_show_locations=0;admin_leftmenu_width=10;admin_url_list_limit=10;admin_url_view_limit=10;admin_section_list_limit=1;admin_orderlist_sortfield=user_name;admin_orderlist_sortorder=desc;admin_search_stats_limit=1;admin_tre_emenu=1;admin_bookmarkmenu=1;admin_left_menu_width=13
TranslationList=
*/?>
```

Da er siteaccessen for både intraNETTE og intraNETTE_admin satt opp. For at det skal fungere er det fortsatt noen innstillinger som må defineres. Gå inn i settings/override/ og åpne filen site.ini.append.php. Alle innstillingene her overstyrer standardinnstillingene.

Settings/override/site.ini.append.php

```
<?php /* #?ini charset="iso-8859-1"?

[Session]
SessionNameHandler=custom
```

```

[SiteSettings]                                /*Da vi har to sider må begge to*/
DefaultAccess=HIG                             /*settes her. HIG er standarden.*/
SiteList[]=HIG
SiteList[]=intraNETTE

[SiteAccessSettings]                          /*Står denne til true starter */
CheckValidity=false                           /*veiviser installasjonen igjen.*/
AvailableSiteAccessList[]=HIG                 /*Må sette hvilke siteaccesser som */
AvailableSiteAccessList[]=intraNETTE         /*er tilgjengelige – både alle fire*/
AvailableSiteAccessList[]=HIG_admin          /*må stå her. */
AvailableSiteAccessList[]=intraNETTE_admin
MatchOrder=uri
HostMatchMapItems[]

[MailSettings]                                /*Mailoppsett,dette står ferdig her*/
Transport=SMTP                                /*når vi åpner fila.*/
TransportServer=studenter.hig.no
TransportUser=*****
TransportPassword=*****
AdminEmail=
EmailSender=

[ExtensionSettings]                           /*Etter installasjon av ekstensjoner*/
ActiveExtensions[]=ezdhtml                    /*må det sjekkes at*/ disse står her,*/
ActiveExtensions[]=ezlocations                /*ellers fungerer de ikke.*/
ActiveExtensions[]=shuffle
ActiveExtensions[]=ezrss

[TemplateSettings]                            /*Ekstra innstilling for shuffle*/
ExtensionAutoloadPath[]=shuffle              /*ekstensjonen brukt på HIG siden.*/
*/ ?>

```

Da skal alt være klart til utvikling. Det er her en fordel å teste siden, her vil man få en feilmelding på intraNETTE siden kernel(1), det er fordi det enda ikke er satt rettigheter og roller til siden.

Design

Går inn i design/intraNETTE/templates/ og oppretter filen /pagelayout.tpl denne filen er hovedfilen og er den som styrer mye av designet og visningen.

Design/intraNETTE/templates/pagelayout.tpl

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <style type="text/css">
      @import url({'stylesheets/stil.css'|ezdesign});
    </style>
    {include uri='design:page_head.tpl'}
  </head>
  <body>
    <div id="titel"><strong>intraNETTE</strong></div>
    <div id="frame">
      {*Logo*}
      <div id="header"></div>

```

```

{*Sti*}
<div id="path">{include uri="design:path.tpl"}</div>

{*Venstrefelt*}
<div id="menu">

    {*Søkfelt*}
    <form action="{/content/search/}|ezurl" method="get">
    <input class="searchbox" type="text" size="8"
    name="SearchText" id="Search"
    value="" />
    <input class="searchbutton" name="SearchButton"
    type="submit" value="Søk" />
    <input type="hidden" name="SubTreeArray[]" value="522" />
    <input type="hidden" name="SubTreeArray[]" value="511" />
    <input type="hidden" name="SubTreeArray[]" value="510" />
    <input type="hidden" name="SubTreeArray[]" value="506" />
    <input type="hidden" name="SubTreeArray[]" value="505" />
    <input type="hidden" name="SubTreeArray[]" value="504" />
    <input type="hidden" name="SubTreeArray[]" value="577" />
    </form><br/>
    {*Slutt søkfelt*}

    {*Tremeny*}
    {* Top kategori *}
    {let top_cat=503
    used_node=false()}
    {section show=is_set($DesignKeys:used.node)}
    {set used_node=$DesignKeys:used.node}
    {section-else}
    {set used_node=$top_cat}
    {/section}
    {let
    node_obj=fetch(content,node,hash(node_id,$used_node))}
    {* Nivå 1 *}
    {section
    loop=fetch(content,list,hash(parent_node_id,$top_cat,
    class_filter_type, "include", class_filter_array,
    array(1),sort_by,array(array('name', true())))}
    <a class="path1" href="{concat("/content/view/full/"
    ,$item.node_id,"/")}|ezurl}>{$:item.name}</a><br />
    {* Nivå 2 *}
    {section show=$node_obj.path_array|
    contains($:item.node_id)
    loop=fetch(content,list,hash(parent_node_id,$:item.node_id,
    class_filter_type, "include", class_filter_array,
    array(1),sort_by,array(array('name', true())))}
    <a class="path2" href="{concat("/content/view/full/"
    ,$:item.node_id,"/")}|ezurl}>{$:item.name}</a><br />
    {/section}
    {/section}
    {/let}
    {/let}<br/><br/>
    {*Slutt meny*}

    {*RT boks*}
    <b>RT</b><br/>
    <iframe src="http://dilbert.hig.no/rtstat3.php" width="140"
    height="80" align="right" scrolling="auto" frameborder="0" >
    [Nettleseren din støtter ikke frames]</iframe>
    {*Slutt RT boks*}
    <br/><br/><br/><br/>

```

```

        {*Hvem innlogget og utlogging*}
        {let thisuser=fetch('user','current_user')}
        {section show=$thisuser.is_logged_in}
        <p>Velkommen. Du er logget inn som
        {$thisuser.contentobject.name}</p>
        <p><a href="{/user/logout"|ezurl}>Logg ut</a></p>
        {/section}
        {/let}
        {*Slutt innlogget/utlogging*}
        <br/><br/><br/><br/><br/><br/>
        <br/><br/><br/><br/><br/>
        {*Slutt venstrefelt*}
        </div>

        {*Innholdet kommer her*}
        {$module_result.content}
        <br clear="all" />

        {*Slutt frame*}
        </div>

        {*Bunntekst*}
        <div id="footer">Av Anette Sæther 20©06 Høgskolen i Gjøvik</div>
    </body>
</html>

```

Pagelayout.tpl henter inn en fil kaldt path.tpl, den ligger også i design/intraNETTE/templates. Denne blir brukt for å få frem en sti på siden.

Design/intraNETTE/templates/path.tpl

```

{*Sti*}
<div class="path">
{section name=Path loop=$module_result.path}
    {section show=$Path:item.url}
        {section show=is_set($Path:item.url_alias)}
            <a href="{ $Path:item.url_alias|ezurl}>{$Path:item.text|shorten(20)}</a> >
        {section-else}
            <a href="{ $Path:item.url|ezurl} >{$Path:item.text|shorten(20)}</a> >
        {/section}
    {section-else}
        {$Path:item.text|wash}
    {/section}
{section-else}
{/section}</div>
{*Slutt sti*}

```

CSS filen skal ligge i katalogen design/intraNETTE/stylesheets/.

Design/intraNETTE/styleheets/stil.css

```

body {
    color: #000000;
    font-size: 12px;
    font-family: Arial, Verdana, Helvetica, sans-serif;
}

```

```
        line-height: 150%;
        background-color: black;
        text-align: left;
    }

    p {
        padding-right: 6px;
        padding-bottom: 6px;
        padding-left: 6px;
    }

    td {}

    #titel {
        visibility: hidden;
        position: absolute;
    }

    #frame {
        background-color: #ffffff;
        margin-right: auto;
        margin-left: auto;
        width: 820px;
    }

    #header {
        background-color: #ff0000;
        text-align: center;
        height: 105px;
    }

    #path {
        color: #000000;
        font-size: 10px;
        font-family: Arial, Helvetica, SunSans-Regular, sans-serif;
        background-color: #000000;
        background-repeat: no-repeat;
        background-position: 0 0;
        text-align: left;
        text-indent: 160px;
        height: 16px;
    }

    #menu {
        background-color: #666666;
        text-align: left;
        padding: 10px 10px 100%;
        width: 140px;
        margin-right: 1em;
        float: left;
    }

    #content {
        background-color: #ffffff;
        text-align: left;
        margin-top: 8px;
        width: 640px;
        float: left }

    #one {
        border-bottom: 1px solid #666666;
    }

    li{
```

```
        margin-left: 1em;
    }

    #vedlikehold {
        font-size: 10px;
    }

    #footer {
        color: #666666;
        font-size: 10px;
        font-family: Arial, Helvetica, SunSans-Regular, sans-serif;
        text-align: center;
        background-position: 0 0;
        margin-right: auto;
        margin-left: auto;
        padding-top: 22px;
        width: 780px;
        height: 20px;
    }

    a.path1:link {
        color: #000000;
        font-size: 11px;
        font-family: Arial, Verdana, Helvetica, sans-serif;
        font-weight: bold;
        line-height: 150%;
        text-decoration: none;
    }

    a.path1:hover {
        color: #000000;
        text-decoration: underline;
    }

    a.path1:visited {
        color: #000000;
        text-decoration: none;
    }

    a.path2:link {
        color: #999999;
        text-decoration: none;
    }

    a.path2:hover {
        color: #999999;
        text-decoration: underline;
    }

    a.path2:visited {
        color: #999999;
        text-decoration: none;
    }

    a:link {
        color: #ff0000;
        font-size: 11px;
        font-family: Arial, Verdana, Helvetica, sans-serif;
        font-weight: bold;
        line-height: 150%;
        text-decoration: none;
    }

    a:hover {
```

```
        color: #ff0000;
        font-size: 11px;
        font-family: Arial, Verdana, Helvetica, sans-serif;
        font-weight: bold;
        line-height: 150%;
        text-decoration: underline;
    }

    a:visited {
    color: #ff0000;
    font-size: 11px;
    font-family: Arial, Verdana, Helvetica, sans-serif;
    font-weight: bold;
    line-height: 150%;
    text-decoration: none;
    }

    h1 { padding: 6px }
    h2 { padding: 6px }
    h3 { padding: 6px }

    .imagecenter {
        padding: 5px;
        display: block;
        text-align: center;
        clear: both;
    }

    .imageleft {
        padding: 5px;
        display: block;
        clear: both;
    }

    .imageright {
        padding: 5px;
        display: block;
        clear: both;
    }

    .attribute-image {
        padding: 3px;
        margin: 0;
    }
```

I katalogen design/intraNETTE/images legges alle bilder til designet. Bilder man publiserer blir liggende og lagret via admin i mediebiblioteket.

Selve mal-filene vi bruker for å overstyre standarddesignet ligger i design/intraNETTE/override/templates. Man er nå klar til å starte selve utviklingen på websiden.

4.2 Overstyring og maler

Under utviklingen er det blitt arbeidet både via administrasjons

grensesnittet og direkte på filsystemet på serveren. I admin ser man lettere hvilke mal-filer man har og hvilke man overskriver. Her setter man også seksjoner, oppretter nye klasser, setter brukerrettigheter og oppretter nye brukere.

4.2.1 Opprette mapper i admin

Slett først alle mappene i innholdsstrukturen i admin. For å ha noe konkret å arbeide mot, og for å opprette menyen senere på websiden opprettes mapper for punktene i menyen. Innholdet på websiden legges senere under disse igjen.

For å opprette mappe går man på Innholdsstruktur, klikker på ønsket toppnivå (eZ publish). Kryss av på alle mappene under denne og klikk Fjern valgte og bekreft dette.

Gå deretter ned i nedtrekksvinduet i bunnen, velg Folder og Lag her.

Navn	Synlighet	Type	Endret av	Endringstidspunkt	Seksjon	Prioritet
<input type="checkbox"/> kopisentralen	Synlig	Folder	Administrator User	15.02.2006 11:34	Kopisentralen	0
<input type="checkbox"/> startsidene	Synlig	Article	Administrator User	29.12.2005 20:55	Ukjent	0
<input type="checkbox"/> e-tjenesten	Synlig	Folder	Administrator User	17.04.2006 16:23	IT forside	0
<input type="checkbox"/> studier	Synlig	Folder	Maria Lillemoen	24.04.2006 13:48	Studier seksjon	0
<input type="checkbox"/> intraNETTE	Synlig	Folder	Administrator User	26.04.2006 18:46	intranette	0

Når man er inne opprettelsen av Folder fyll inn Name og Short name. Klikk deretter på Send til publisering. Kan når som helst avbryte publiseringen med å klikke Forkast utkast.²⁴

Opprett en mappe for hvert av meny punktene:

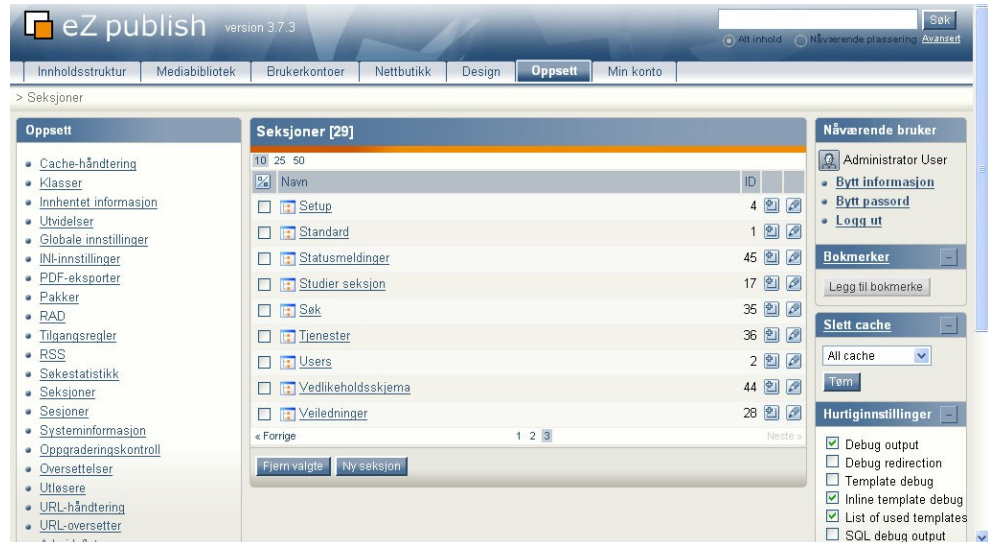
intraNETTE
 Søk
 Dokumentasjon
 Vedlikeholdsskjema
 Driftslogg
 Driftsmeldinger
 Statusmeldinger

Kan så gå videre til opprettelsen av seksjoner og videre utvikling.

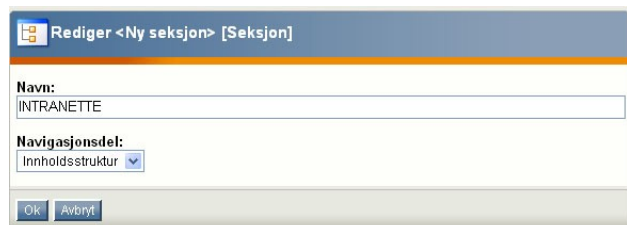
²⁴ Publisering blir gått nøyere gjennom i Vedlegg B og C for hver av websidene.

4.2.2 Forsiden

Starter inne i administrasjonsgrensesnittet med å opprette en seksjon for hele intraNETTE. Går da på Oppsett -> Seksjoner og Ny seksjon. Seksjonsnavn og ID er kun ment som illustrasjon.



Kaller seksjonen INTRANETTE og setter Navigasjonsdel til Innholds struktur. Klikker så ok.



Kommer da tilbake til forrige skjermbilde, hvor man navnet på Seksjonen og hvilke ID (48) det har. Klikker så på den lille plusstegnet bak INTRANETTE, da man må legge Folder til i seksjonen for at den skal kunne brukes til noe.



Da kommer man til et vindu hvor man setter folder til tilhørende seksjon. Navigerer seg frem til rett plass, som i dette tilfelle er folderen intraNETTE. Krysser av i boksen og klikker så ok.



Prosessen med seksjoner gjøres for hver av under mappene til intraNETTE som ble opprettet via Innholds strukturen.

4.2.3 Informasjonsfeltet til høyre

Her ser man til en hver tid hvem som er innlogget og Legg til bokmerke funksjon. Noe av det viktigste her er funksjonen for Tøm cache.

Når noe er gjort via admin er det viktig å tømme cache, ellers vil ikke resultatet komme frem på websiden. Man kan velge hva man vil tømme i Cache, men kan like greit tømme All Cache.

I feltet hurtiginstillinger kan man velge om man vil se debug på siden og hvilke maler som er brukt. Siteaccess er nødt til å stå på den samme som du jobber mot, ellers vil ingenting skje eller gå feil. Står siteaccessen på Global vil standard malen fra eZ kjøre og man vil ikke se noe av det man har gjort.



4.2.4 Opprettelsen av klasser

Man kan bruke de innebygde klassene i eZ. Men hvis man vil ha litt mer spesielle eller brukerdefinerte klasser kan man også opprette nye. Da får man full kontroll over feltene og blir lettere å bruke for de som senere skal publisere.

Går inn i Oppsett -> Klasser, oppretter først en ny Klassegruppe.

The screenshot shows the 'Oppsett' (Settings) menu with the following items:

- Cache-håndtering
- Klasser
- Innholdet informasjon
- Utvidelser
- Globale innstillinger
- INI-innstillinger
- PDF-eksporter
- Pakker
- RAD
- Tilpasningsreoler
- RSS
- Søkestatistikk
- Seksjoner
- Sesjoner

The 'Klassegrupper [7]' section contains the following table:

Navn	Endret av	Endringstidspunkt
Content	Administrator User	06.10.2002 18:35
Users	Administrator User	06.10.2002 18:35
Media	Administrator User	06.10.2002 18:35
Setup	Administrator User	13.04.2004 14:07
Studier	Administrator User	13.01.2006 14:09
Kopisentraen	Administrator User	01.02.2006 11:31
IT	Administrator User	30.03.2006 18:18

Kaller denne IT og trykker Ok.

The dialog box shows the following fields and buttons:

- Rediger <Ny gruppe> [Klassegruppe]
- Navn: IT
- Buttons: Ok, Avbryt

Går så inn i klassegruppen IT, klikker deretter Ny klasse.

The details page for the 'IT' group shows the following information:

- Sist endret: 30.03.2006 18:18, Administrator User
- ID: 8
- Navn: IT
- Buttons: Rediger, Fjern

The 'Klasser i <IT> [7]' section contains the following table:

Navn	ID	Identifikator	Endret av	Endringstidspunkt	Objekter
Ansatt	42	ansatt	Administrator User	20.04.2006 19:49	19
Driftslog	44	driftslog	Administrator User	23.04.2006 22:28	5
Driftsmelding	46	driftsmelding	Administrator User	27.04.2006 16:54	8
Høyrefelt	40	hoeyrefelt	Administrator User	30.03.2006 18:19	5
Ord	41	ord	Administrator User	05.04.2006 19:37	12
RT	48	rt	Administrator User	28.04.2006 21:39	1
Vedlikeholdsskjema	47	vedlikeholdsskjema	Administrator User	28.04.2006 20:42	3

Her setter man Navn, Identifikator, Objektnavnmønster og om den er en beholder før man starter å legge til attributter.

Klassen Driftslogg er den klassen med minst attributter. Under ser man en oversikt over innstillinger og attributter.

Navn: Driftslog
 Identifikator: driftslog
 Objektnavn-mønster: <paa>
 Beholder: Nei
 Antall objekter: 5
 Attributter

1. Utført på [Tekstlinje] (id:347)

Navn: Utført på	Identifikator: paa
--------------------	-----------------------

Standardverdi: Ingen verdi Maksimum tekstlengde: 0 Karakterer	Flagg: Er påkrevet Er ikke søkbar Innhenter informasjon Kan oversettes
--	--

2. Utført arbeid [Tekstblokk] (id:344)

Navn: Utført arbeid	Identifikator: arbeid	
		Flagg: Er påkrevet Er søkbar Innhenter informasjon Kan oversettes
Foretrukket antall rader: 10		

3. Dato [Dato] (id:346)

Navn: Dato	Identifikator: dato	
		Flagg: Er ikke påkrevet Er søkbar Innhenter ikke informasjon Kan oversettes
Standardverdi: Nåværende dato		

4. Utført av [Tekstlinje] (id:349)

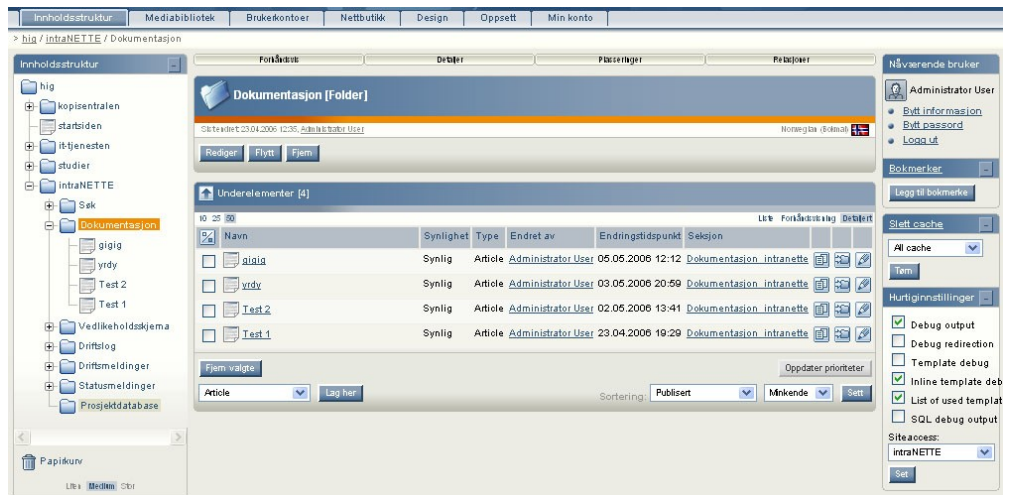
Navn: Utført av	Identifikator: author	
		Flagg: Er påkrevet Er ikke søkbar Innhenter ikke informasjon Kan oversettes
Standardverdi: Ingen verdi Maksimum tekstlengde: 0 Karakterer		

Det er opprettet egne klasser på denne måten på både intraNETTE og IT-tjenestens webside for klassene Ansatt, Driftslog, Driftsmelding, Høyrefelt, Ord, RT og Vedlikeholdsskjema.

Når klassene er på plass kan man benytte å overstyre og skrive nye maler.

4.2.5 Legge til innhold

For å ha noe konkret å teste mot må man legge til innhold i mappene. Gå på Innholds struktur og klikk på Dokumentasjon under intraNETTE. Helt nederst kommer det et rullgardinvinde hvor det velges Artikkel og klikk Lag her.



Da kommer man til en ny side med en rekke felter. Fyll ut alle, spesielt de hvor det står Påkrevet. Klikk deretter på Publiser helt nederst. Tøm deretter Cache.

4.2.6 Overstyring og maler

Når man skriver maler må man begynne helt i bunnen, å hente ut innhold før man setter visningen av det. Begge websidene henter her opp innholdet via full/folder standardmalen. Forklarer fremgangen med bruk av seksjonen Dokumentasjon_intraNETTE. Da denne siden er veldig greit satt opp og prosessen enklere å forklare.

Gå inn i Design. Man må her passe på at siteaccessen står på intraNETTE og ikke f.eks HIG eller intraNETTE_admin. Gå deretter inn på Maler.



Da ser man alle overstyringsfilene i systemet. Navigere ned på siden, under

Vanligste maler der node/view/full.tpl står. Her ser man også hvilke Desingressurs denne filen bruker. Klikk på full.tpl.

<input type="checkbox"/>	forum_topic_full	design/base/override/templates/full/forum_topic.tpl	class_identifier : forum_topic	20	
<input type="checkbox"/>	forum_reply_full	design/base/override/templates/full/forum_reply.tpl	class_identifier : forum_reply	21	
<input type="checkbox"/>	weblog_full	design/base/override/templates/full/weblog.tpl	class_identifier : weblog	22	
<input type="checkbox"/>	product_full	design/base/override/templates/full/product.tpl	class_identifier : product	23	
<input type="checkbox"/>	gallery_full	design/base/override/templates/full/gallery.tpl	class_identifier : gallery	24	
<input type="checkbox"/>	poll_full	design/base/override/templates/full/poll.tpl	class_identifier : poll	25	
<input type="checkbox"/>	person_full	design/base/override/templates/full/person.tpl	class_identifier : person	26	
<input type="checkbox"/>	company_full	design/base/override/templates/full/company.tpl	class_identifier : company	27	
<input type="checkbox"/>	feedback_form_full	design/base/override/templates/full/feedback_form.tpl	class_identifier : feedback_form	28	

Fjern valgte Ny overstyring Oppdater prioriteter

Her ser man hvilke filer som overstyret full.tpl. Sortert etter prioriteter. Helt til høyre ser man hvilke filer de enkelte malene kjører på. Eksempel Node, Seksjon eller Klasse.

For å sette ny prioriteter skriver man inn aktuell prioriteter i tall feltet og klikker Oppdater prioriteter. Malene man utvikler må ligge over de som er standard for at det skal fungere som det skal.

Bak tallet for prioriteter er det et lite ikon med en blyant, her trykker man når filen skal redigeres.

Helt nederst på siden går man inn på Ny overstyring.

Fyller her inn i feltene og setter hva man vil kjøre overstyringen på. Kaller filen dokumentasjon_intrafullfolder, .tpl blir her lagt på automatisk. På klasse velger man i dette tilfelle Folder, siden vi skal hente ut innholdet i folderen. Setter Seksjon til Dokumentasjon_intranette. Baserer malen på Tom fil. Etter at alt er ferdig utfylt klikker man bare Ok.

Lag ny overstyrende mal for <node/view/full.tpl>

Den nye malfilen vil bli plassert i design/intraNETTE/override/templates/.

Filnavn:
.tpl

Nøkler for overstyring:

Klasse:

Seksjon:

Node-ID:

Baser malen på:

Tom fil

Kopi av standardmalen

Beholder (med barn)

Visning (uten barn)

Ok Avbryt

Kommer så tilbake til forrige skjermbildet, der vil man finne igjen filen man nettopp lagde, og trykker så på redigeringsikonet.

Da er man kommet til et blankt felt, det er her selve koden skal ligge.

```

Rediger malen <design/intraNETTE/override/templates/dokumentasjon_intrafullfolder.tpl>

<div class="one">
  [*Navn på folder - Dokumentasjon*]
  <h2>{*node.name}</h2><br/>

  [*Description på folder - hvis det er noe*]
  {$node.object.data_map.description.content.output.output_text}

  [*Knapp for å lage ny*]
  <form method="post"
    action="{content/action/"|ezurl}>
    <input class="button"
      type="submit"
      name="NewButton"
      value="Ny dokumentasjon" />
  [*Hvile klasseID man skal skrive til (article=2)*]
  <input type="hidden"
    name="ClassID"
    value="2" />
  [*Node man er i*]
  <input type="hidden"
    name="NodeID"
    value="{*node.node_id}" />
  </form><br/>
  [*Slutt knapp*]

  [*Henter ut alle artikler fra noden vi står i = dokumentasjon*]
  {let doku=fetch(content, list, hash( parent_node_id, $node.node_id,
    sort_by.array(published, false),
    class_filter_type, include, class_filter_array, array(article)))}
  {section name=dok loop=$doku}
  {node_view_gui view=line content_node=$dokitem}
  {/section}
  {/let}
</div>

Aktiver endringer  Tilbake til maloverstyringen
  
```

Koden her begynner med å lage en knapp for å lage ny dokumentasjon. Etter det henter den ut alle artikler i folderen Dokumentasjon. Deretter sier den at visningen skal være med line view.²⁵

Klasser til CSS er også satt inn i denne koden som sikrer at innholdet blir vist rett på siden.

Design/intraNETTE/override/templates/dokumentasjon_intrafullfolder.tpl

```

<div class="one">
  {*Navn på folder - Dokumentasjon*}
  <h2>{*node.name}</h2><br/>

  [*Description på folder - hvis det er noe*]
  {$node.object.data_map.description.content.output.output_text}

  [*Knapp for å lage ny*]
  <form method="post"
    action="{content/action/"|ezurl}>
    <input class="button"
      type="submit"
      name="NewButton"
      value="Ny dokumentasjon" />
  [*Hvile klasseID man skal skrive til (article=2)*]
  <input type="hidden"
    name="ClassID"
    value="2" />
  [*Node man er i*]
  <input type="hidden"
    name="NodeID"
    value="{*node.node_id}" />

  </form><br/>
  [*Slutt knapp*]
  
```

²⁵ Line view er det samme som linjevisning. Ofte med link til visning av en hel artikkel.

```

{*Henter ut alle artikler fra noden vi står i =dokumentasjon*}
{let doku=fetch(content, list, hash( parent_node_id, $node.node_id,
                                   sort_by,array(published,false()),
                                   class_filter_type, include, class_filter_array,
                                   array(article)))}
    {section name=dok loop=$doku}
        {node_view_gui view=line content_node=$dok:item}
    {/section}
{/let}
</div>

```

Etter at denne koden er aktivert, må man oppdatere prioritet. Så må man tømme cache, deretter kan man sjekke siden og se etter om det fungerer som det skal. Man vil da se innholdet i Dokumentasjon seksjonen som navn med etlite ikon, dette fordi det enda ikke er blitt satt noen linjevisning.

For å sette linjevisning, klikk på Maler på venstre side og finn filen node/view/line.tpl i stede for full.tpl, å gå inn på denne. Nederst på siden klikk Ny overstyring.

Fyll inn i feltene:

Filnavn: dokumentasjon_intralinje

Klasse: Article

Seksjon: Dokumentasjon_intranette

Baser filen på: Tom fil

Etter at man har godkjent dette, kommer man tilbake til listen med overstyringer på line.tpl, finner da igjen filen man laget og redigerer denne. Putter inn koden for linje visningen av artikler i folderen Dokumentasjon i seksjonen Dokumentasjon_intranette.

Design/intraNETTE/override/templates/dokumentasjon_intralinje.tpl

```

{*Innhold - linje visning*}
<div class="content">
    {*Dato*}
    {$node.object.published|!ion( 'shortdate' )}

    {*Tittel som link til selve artikkelen*}
    <a href={$node.url|ezurl}>{attribute_view_gui attribute=$node.object.data_map.title}</a>

    {*Forfatter*}
    {attribute_view_gui attribute=$node.object.data_map.author}
    <br/>
</div>

```

Oppdater prioritet og tøm cache. Så kan resultatet sees på websiden. Filen dokumentasjon_intralinje.tpl har en linje til fullvisning av selve artikkelen. Denne er det ikke enda blitt satt noen mal på, så det må gjøres.

Finn igjen filen `node/view/full.tpl` under Maler, gå inn på denne og Lag Ny overstyring. Fyll inn feltene likt som på fullfolder visningen men kjør den på Article og ikke Folder som på folder visningen. Kall filen `dokumentasjon_intraarticle`, klikk deretter Ok.

Da kommer man tilbake til forrige skjermbilde. Rediger så filen og putt inn kode.

Design/intraNETTE/override/tempaltes/dokumentasjon_intraarticle.tpl

```
{*Artikkelvisning av artikler i dokumentasjons mappen*}
<div class="content">
  <div class="one"><br/>
    {*Tittel*}
    <h2>{attribute_view_gui attribute=$node.object.data_map.title}</h2>

    {*Intro*}
    {attribute_view_gui attribute=$node.object.data_map.intro}

    {*Body*}
    {attribute_view_gui attribute=$node.object.data_map.body}<br/>

    {*Forfatter*}
    <i>{attribute_view_gui attribute=$node.object.data_map.author}</i><br/>

    {*Dato*}
    {$node.object.published|!ion( 'shortdate' )}</i><br/>
  </div>
</div>
```

Trykk Aktiver endringer etter at koden er skrevet inn. Oppdater så prioritert og tøm cache. Da skal alt innholdet i dokumentasjonsmappen bli vist korrekt helt fra full folder, via linje og til artikkel.

En ting gjenstår allikevel, det er nå ingen funksjon for å publisere. På linjevisningen opprettets en knapp for å lage ny dokumentasjon. Denne er allerede satt til å ha en funksjon, men designet til denne siden er ikke satt.

Her er man nødt til å redigere direkte på serveren. Opprett filen `design/intraNETTE/override/templates/edit_dokumentasjon_entry.tpl` og sett inn koden:

design/intraNETTE/override/templates/edit_dokumentasjon_entry.tpl

```
{*For å lag ny dokumentasjon*}
<div class="one">
  <h2>Lag ny dokumentasjon</h2>
  {*skjema for selve publiseringen - henter først opp objekt id og lager ny*}
  <form enctype="multipart/form-data"
    method="post"
    action="{concat("/content/edit/",
      $object.id,
      "/",
      $edit_version)}|ezurl">
```

```

{*Tabell for å sikre rett visning*}
<table>
  <tr>
    <td>
      {*Templater for redigering*}
      {include uri="design:content/edit_validation.tpl"}
      {include uri="design:content/edit_attribute.tpl"}

      {*Knapper for publisering*}
      <div class="buttonblock">
        <input class="defaultbutton"
          type="submit"
          name="PublishButton"
          value="Publiser" />
        <input class="button"
          type="submit"
          name="DiscardButton"
          value="Forkast" />
        {*Hovednoden*}
        <input type="hidden"
          name="MainNodeID"
          value="{ $main_node_id }" />
      </div>
    </td>
  </tr>
  {*Slutt tabell*}
</table>
{*Slutt skjema*}
</form>
</div>

```

Her lages det et skjema med en tabell for å lage selve dokumentasjonen. Det er enda ikke satt hvilke klasse og seksjon denne skal overstyre. Gå inn på `settings/siteaccess/intraNETTE/override.ini.append.php` og legg til

`settings/siteaccess/intraNETTE/override.ini.append.php`

```

[edit_dokumentasjon_entry]                                /*hva overstyringen
heter*/
Source=content/edit.tpl                                  /*kilder*/
MatchFile=edit_dokumentasjon_entry.tpl                  /*filnavn*/
Subdir=templates
/*ligger i templer*/
Match[class]=2
/*artikkelklasse er nr 2r*/
Match[section]=43
/*Dokumentasjon_intranetter*/ seksjonen

/*har ID 43r*/

```

Lagre filen og tøm cache via admin. Da skal man kunne trykke på knappen Ny dokumentasjon og komme inn i et område hvor ny artikkel kan lages og publiseres. Link til artikkelen vil komme på linjevisning hvor også informasjon om dato og signatur vises. Inne i visningen av hele artikkelen, vil alt av innhold i klassen artikkel vises.

Dette er fremgangsmåten for å overstyre filer med eZ publish som

rammeverk. Det er ikke brukt fullvisning av klassen overalt, som på artikkel, i dette eksempelet. Da det ikke bestandig er behov for det. Kodens og kompleksitet varierer også mye fra overstyring til overstyring. To visninger er sjelden like, og da det er egendefinerte klasser, henter man ut og viser som oftest annet innhold på sidene.

4.7 Usergroup, User, Roller og tilgangsregler

For at siden skal kunne bli brukt av anonyme brukere er man nødt til å sette roller og opprette brukere.

Gå inn på Brukerkontoer. Nederst i rullgardinvinduet forandre fra User til User group og trykk Lag her.

Kall brukgruppen IT og klikk Send til publisering. Da vil brukergruppen komme opp på venstre side i det første skjermbildet under User.

Klikk så på IT brukergruppen.

Her ser man informasjon om brukergruppen og hvilke roller og tilgangsregler som er satt til den. Klikk Lag her bak User nederst i vinduet. Øverst kan man også sette hvilke informasjon man vil se, eksempelvis roller og tilgangsregler for den brukergruppen man ser på.

IT [User group]

Siste endring: 24.04.2006 14:35, Administrator User

Rediger | Flytt | Fjern

Underelementer [4]

Navn	Synlighet	Type	Endret av	Endringstidspunkt	Seleksjon
<input type="checkbox"/> Jon Langseth	Synlig	User	Administrator User	03.05.2006 11:03	Users Kopier
<input type="checkbox"/> Stian Husemoen	Synlig	User	Administrator User	03.05.2006 11:00	Users Kopier
<input type="checkbox"/> Jørgen Hanssen	Synlig	User	Administrator User	03.05.2006 10:58	Users Kopier
<input type="checkbox"/> ittest it	Synlig	User	Administrator User	29.04.2006 14:57	Users Kopier

Fjern valgte | Oppdater prioriteter

User | Lag her

Sortering: Publisert | Minkende | Sett

Her oppretter man brukere til brukergruppen IT. Fyll ut feltene, spesielt de som er påkrevet. Fornavn, etternavn, brukernavn og passord og email. Resten av feltene er ikke påkrevet. Trykk til slutt på publiser. Tøm så cache. Da er brukeren opprettet og den vil komme opp under brukergruppen på venstre side under IT.

Gå deretter inn på Roller og tilgangsregler. Her finner man en oversikt over alle oppsatte roller. Klikk så på Ny rolle nederst på siden.

Innholdsstruktur | Medialbibliotek | Brukerkontoer | Nettbutikk | Design | Oppsett | Min konto

> Rolleliste

Brukerkontoer

- Users
- Guest accounts
- Administrator users
- Editors
- Studier editorer
- Anonymous Users
- IT

Papirkurv

Tilgangsinstillinger

- Roller og tilgangsregler

Roller [9]

Navn	Synlighet	Type	Endret av	Endringstidspunkt	Seleksjon
<input type="checkbox"/> Administrator	Synlig	User	Administrator User	03.05.2006 11:03	Users Kopier
<input type="checkbox"/> Anonymous	Synlig	User	Administrator User	03.05.2006 11:00	Users Kopier
<input type="checkbox"/> Editor	Synlig	User	Administrator User	03.05.2006 10:58	Users Kopier
<input type="checkbox"/> Forum administrator	Synlig	User	Administrator User	29.04.2006 14:57	Users Kopier
<input type="checkbox"/> Forum user	Synlig	User	Administrator User	03.05.2006 11:03	Users Kopier
<input type="checkbox"/> IntraNETTE	Synlig	User	Administrator User	03.05.2006 11:00	Users Kopier
<input type="checkbox"/> IT	Synlig	User	Administrator User	03.05.2006 11:00	Users Kopier
<input type="checkbox"/> Studier	Synlig	User	Administrator User	03.05.2006 11:00	Users Kopier
<input type="checkbox"/> Weblog user	Synlig	User	Administrator User	03.05.2006 11:00	Users Kopier

Fjern valgte | Ny rolle

Nåværende bruker

- Administrator User
- Bytt informasjon
- Bytt passord
- Løse ut

Bokmerker

Lagg til bokmerke

Slett cache

All cache

Tem

Hurtiginstillinger

- Debug output
- Debug redirection
- Template debug
- Inline template debug

Kall rollen INTRANETTE, klikk deretter Ny tilgangsregel.

Man kommer da til en side med nedtrekksvindu hvor man velger modul, eksempelvis Content. Gå gjennom stegene til resultatet ser slik ut (ref til bilde). Trykk deretter Ok. Da kommer man tilbake til oversikten over rollen, helt nederst er det et felt for å tilordne rollen.

Modul	Funksjon	Begrensning
user	login	SiteAccess(intraNETTE_admin , intraNETTE)
ezdhtml	All funksjoner	Ingen begrensninger
rss	feed	Ingen begrensninger
form	All funksjoner	Ingen begrensninger
setup	managecache	Ingen begrensninger
user	selfedit	Ingen begrensninger
infocollector	read	Ingen begrensninger
content	All funksjoner	Ingen begrensninger

Bruker/gruppe	Begrensning
IT	Ingen begrensninger

Her velges brukergruppen IT og trykker Ok. Tøm Cache og test at brukeren i brukergruppen fungerer med de begrensningene som er satt.

IT-tjenestens webside har flere begrensninger enn intraNETTE, der det kan gjøres mer skade. Her arbeider brukerne direkte mot siden og trenger ikke

admin til noe bortsett fra å eventuelt slette noe for godt.

Når man setter roller og tilgangsregler må man ofte teste frem og tilbake før man får satt det rett.

5 Produksjon

5.1 Kode hentet fra andre

Alt templatekoding med override og utviklingen mot eZ publish er laget på egen hånd. På visse områder kom kunnskapen til kort og fikk derfor hjelp til visse ting av andre på IT-tjenesten. Dette gjaldt javascript for kartet på datalabber og boks med antall åpne helpdesk saker.

I starten ble websiden design laget med CSS som baserte seg på tabeller. Da dette ikke var ønskelig fra IT-tjenestens side fikk jeg litt starthjelp fra IT på koding av CSS uten bruk av tabeller.

eZ er også bygd opp slik at man har et sett standardmaler som man igjen overstyrrer for å få de funksjonene og visningene man selv ønsker. Mye kode ligger derfor internt i systemet, som en slags manual for kodingen. Dette er nyttig ved for eksempel koding av søk og skjemaer for redigering.

Javascript er et kodespråk jeg har liten erfaring med. Fikk derfor hjelp til å kode opp spørsmålsfeltene på kontaktskjemaet til websiden av Jon Langseth. Det samme gjelder javascript koden i pagelayout for å få vist siden riktig i Internett Explorer, dette ble mottatt av Stian Husemoen.

Boksene for åpne helpdesk (RT) saker er hentet ut fra Request Tracker med bruk av PHP. Koden for å hente ut data er skrevet av Anders Wiehe på IT-tjenesten. Mottok en URL på dette og det ble deretter puttet inn i en iFrame for visning på websiden.

Tre-menyen på intraNETTE er hentet fra http://ez.no/products/ez_publish/documentation/customization/tips_tricks/creating_tree_menus den er modifisert en del for å passe behovet på siden.

5.2 Testing

Hele arbeidet foregikk direkte mot serveren. Så det var derfor enkelt å avdekke feil og mangler på et tidlig stadie. Benyttet Firefox som nettleser, ettersom man med en extension kan man redigere på CSS filen mens man ser resultatet direkte. Dette var til stor nytte under arbeidet med designet og ikke minst utover i prosjektet hvor CSS-filen ble større og mer kompleks.

Firefox developer-extension har også funksjon for å utheve bokser og vise aktuelt element, dette kom til stor nytte i utviklingen. Det fremhever CSS bokser og man ser hvilke tag fra CSS man holder musen over.

EZ har en innebygd funksjon hvor man kan sette på brukte templer, debug output og debug template output. Da kan man underveis i arbeidet se hva som går galt og hvilke templer som benyttes. Den sier også hvor lang tid siden bruker på å laste. Hvis noe av koden har generert noe kraftig bruk av prosessorkraft kan dette avdekke det med en gang.

I begynnelsen i prosjektet ble desktop-versjonen av eZ-publish installert, denne fungerte veldig greit, men det går raskere å kjøre det direkte mot serveren. Samtidig er det lettere å dele, vise og problemløse det man har gjort når det ligger på servere.

En av grunnene til at det ble valgt å arbeide direkte mot serveren var å få hjelp til eventuelle problemer. Da ingen på IT-tjenesten og HIG hadde noe kunnskap om eZ publish var det til tider behov til å benytte hjelp utenfra. Det besto i hovedsak av Pål Messenlien på HIL som har veldig god kunnskap om eZ.

Backup-script ble skrevet av de ansatte på IT-tjenesten. Det tar hver natt backup av utviklingen i eZ. Har ved en anledning i arbeidet hatt behov for å benytte meg av denne backupen. Da eZ er så stort og omfattende er det fort gjort å gjøre noe som resulterer i at hele websiden går ned.

5.3 Drøfting

Er veldig godt fornøyd med resultatet generelt, det samsvarer også med ønsker gitt av IT-tjenesten.

Et av kravene fra forprosjektrapporten måtte dessverre settes til side, det var ønsket om å foreta brukertester for å sette god navigasjon og innhold på IT-tjenestens websider. Dette gikk bort på grunn av resten av oppgavens

omfang samt at det tidligere var foretatt lignende tester på HIG`s webside som helhet. Valgte derfor i samråd med de på IT-tjenesten og veileder å gå bort fra dette kravet og heller legge basis i de aktuelle rapportene.²⁶

Det var også ønskelig fra IT-tjenestens side med mer funksjonalitet på websidene, mye av dette ble forkastet på grunn av kompleksitet i forhold til antatt tidsbruk.

På intraNETTE under Statusmeldinger skulle det være en RSS feed på RT helpdesk, denne ble også satt opp men mye tydet på at feil php versjon på serveren gjorde at det ikke fungerte. Installasjonen av apache, php, database og eZ på serveren var noe IT-tjenesten selv tok seg av.

Det var også ønskelig å ha en Prosjektdatabase på intraNETTE. Kravspesifikasjonen på denne ble mottatt veldig sent i prosjektperioden så det ble derfor kuttet ut. Visningen i denne var også noe vel komplekst til at det var gjennomførbart på så kort tid som det var til rådighet før prosjektlevering.

IT-tjenestens webside hvor det er informasjon om datalaber var det et ønske om at kartet også skulle inneholdt linker til informasjon om de forskjellige rommene. Dette ble et mer omfattende arbeid og ble derfor begrenset til kun et oversiktskart. Image map²⁷ tagger ligger fortsatt i koden til kartet, så hvis man ønsker å videreutvikle dette er det fullt mulig ved en senere anledning. De er nå kodet ut og inaktive.

Da arbeidet besto av et tett samarbeid med IT-tjenesten var det mange ønsker og ideer som kom angående både websiden og intraNETTE, mye ble sortert ut og måtte sette en grense for at det ikke skulle utvikle seg til å bli et ekstremt stort prosjekt.

På begge sidene er det en sti, denne fungerer ikke som den skal på websiden. Dette fordi hele websiden ikke ligger i eZ, men noe i eWay. Da vil ikke stien bli riktig når denne blir satt. Derfor vises ikke stien hele veien, men bare deler av den. På intraNETTE derimot fungerer den mye bedre, men først står HIG som link. Dette fordi det ligger i samme administrasjons grensesnitt. Fant ingen god løsning for noen av disse problemene.

Til tross for at det var mye ønsker fra IT-tjenesten så var det også til tider vanskelig å få tak i informasjon. Dette gjaldt i hovedsak innhold som skulle publiseres på websiden. Dette har igjen resultert i at websiden nå er litt innholdsløs. Dette er igjen noe hovedprosjektet i seg selv ikke har ansvar for.

²⁶ Se litteraturlisten.

²⁷ Image map tagger er koordinater i bildet, feltene gitt av koordinatene er igjen i dette tilfelle linker.

5.4 Realisering

Websiden ble overlevert IT-tjenesten med manual 3. mai 2006. Ønsket var stort om å benytte den med en gang, men på grunn av manglende innhold utsatte man å gå live²⁸ noen uker til innholdet kom på plass.

IntraNETTE blir overlevert IT-tjenesten 22. mai. Den blir da også kjørt live.

Alle brukere fikk tildelt eget brukernavn og passord. De fikk også utlevert manual for begge websidene. Publiseringen har gått veldig greit og alle er veldig fornøyd med resultatet.

²⁸ I denne sammenhengen betyr live å begynne å bruke websiden. Kjøre den ut til brukerne.

6 Konklusjon

I begynnelsen av prosjektet var det en viss urolighet omkring kompleksiteten til prosjektet og om det var gjennomførbart i forhold til forkunnskapene. Arbeidet gjennom perioden har gått mye bedre enn forventet og resultatet ble veldig bra resultatet er en god og brukervennlig internett løsning for IT-tjenesten, og da ikke bare brukervennlig for sluttbrukerne men også de ansatte på IT-tjenesten i forhold til publisering. De har fått et enkelt og intuitivt intranett som vil bedre kommunikasjon og forenkle arbeidet internt.

De faglige kunnskapene tilegnet gjennom arbeidet kan benyttes til en rekke ting, blant annet er det å kunne eZ publish bra for videre karriere. Samtidig skal det etter prosjektperioden arbeides videre med både publisering og utvikling med eZ mot Høgskolen.

En rekke faglig kunnskap er tilegnet under prosjektarbeidet:

- arbeide mer praktisk med en konkret oppdragsgiver
- sette teorier i praksis
- tett samarbeid med og inngått i et faktisk arbeidsmiljø
- organisere og styre et stort og omfattende prosjekt
- inngående kunnskap om CSS og webdesign
- html, xhtml og xml kunnskap
- eZ publish som program, utvikler rammeverk og miljø
- programmering generelt

Hvis det var noe som burde vært gjort annerledes vil jeg påpeke at det fra begynnelsen av burde vært benyttet rett php versjon og at jeg burde presset mer på for å få gjennomført dette. Det hadde antagelig ført til at RSS feeden ville ha fungert som den skulle.

I forhold til IT-tjenesten og samarbeidet med dem har det gått veldig bra, dette var også noe det rådte noe usikkerhet omkring i begynnelsen. Jeg har vært en del av de på IT-tjenesten og har alltid hatt noen å spørre der hvis det har vært noe spesielt de har kunnet hjelpe med. Samtidig har de gjort en enestående jobb jeg ikke kunne vært foruten med tanke på servere og backsystemet.

Prosjektet er absolutt noe jeg ville ha gjort igjen, det har vært utfordrende,

lærerikt og jeg har tilegnet meg kunnskap og erfaring jeg aldri ville trodd jeg hadde kommet til å arbeide med.

7 Litteraturliste

7.1 Skriftlig litteratur

Paul Borgermans, Tony Wood, Paul Forsyth, Ben Pirt, Björn Dieding: Learning eZ publish 3: Building Content Management Solutions, Packt Publishing Limited 2004.

Eric A. Meyer, Cascading Style Sheets: The Definitive Guide, Second Edition, O`Reilly, USA 2004.

Eric A. Meyer: CSS Pocket Reference, Second Edition, O`Reilly, USA 2004.

Jennifer Niederst: HTML Pocket Reference, Second Edition, O`Reilly, USA 2002.

Robert Eckstein with Michel Casabianca: XML Pocket Reference, Second Edition, O`Reilly, USA 2001.

Stein Erik Frugård og Øystein Leiknes Nag: Brukerkvalitetstesting av nettstedet til Høgskolen i Gjøvik, Fordypning i brukerkvalitetstesting, Høgskolen i Gjøvik 2005.

Marianne Brattrud, Hans Erik Grøthaug, Maria Sunde Wroldsen og Synne Kvamme Repp: Rapport i brukskvalitetstesting, Test av nettstedet www.hig.no, Høgskolen i Gjøvik 2005.

Trond Viggo Bjerke og Terje Stafseng: Fordypning i brukskvalitetstesting – test av HIG sine nettsider, Høgskolen i Gjøvik 2005.

Håvard Aasem, Anders Haagensen, Finn Nikolai Hondrelis og Jan Thomassen: Hovedprosjekt, Dynamisk internettløsning for Raufoss Industripark, Høgskolen i Gjøvik 2004.

7.2 Fra internett

- <http://ez.no/>
- http://ez.no/doc/ez_publish
- http://ez.no/doc/extensions/online_editor
- http://ez.no/community/contribs/template_plugins/extension_shuffle
- http://ez.no/community/contribs/workflow/locations_workflow_event
- http://ez.no/community/contribs/template_plugins/ezrssfeed
- http://ez.no/products/ez_publish/documentation/customization/tips_tricks/creating_tree_menus
- http://ez.no/products/ez_publish/documentation/building_an_ez_publish_site
- http://ez.no/download/ez_publish
- <http://ez.hig.no/HIG/studier>
- <http://int.hig.no>
- <http://int2.hig.no>
- <http://validator.w3.org/>

8.V

Vedlegg