

Bruk av IKT som verktøy for etter- og videreutdanning av lærere

Morten Gabrielsen

Master i informatikk
Oppgaven levert: Juni 2008
Hovedveileder: Terje Rydland, IDI

Forord

Denne Masteroppgaven er et resultat av et forskningsstudie i emnet "IT3900 Masteroppgaven, Informatikkstudiet" ved NTNU fra høsten 2006 til våren 2008. Oppgaven er definert av min veileder, Terje Rydland, ved IDI.

Jeg ønsker å takke Terje Rydland for gode råd og fine samtaler ellers i løpet av tiden jeg jobbet med oppgaven. Det har vært både lærerikt og interessant, og jeg håper resultatet vil komme til nytte både for meg og for flere andre.

Jeg ønsker også å gi en takk til Hildegunn Otnes ved Program for Lærerutdanning, for nyttig informasjon om PLU og om behovet for nettbaserte verktøy ellers. En takk også til studentgruppa i IT2901, og for hjelpen som en representant fra den gruppa har bistått meg i løpet av prosjektet.

Til slutt også en takk til min familie og den støtte de har gitt gjennom studietiden.

Trondheim 30. juni 2008

Morten Gabrielsen

Oppgavetekst

Oppgaven går ut på å foreta en undersøkelse av nettbaserte ressurser for etter- og videreutdanning av lærere. Det skal også ses på design og evaluering av digital kommunikasjon og IKT-verktøy for bruk til dette formålet. Som et case studie skal prosjektet "PLU-veven" studeres.

Sammendrag

Det har vært et behov for en undersøkelse av hvilke ressurser som finnes i etter- og videreutdanning av lærere, og videre en undersøkelse hva det faktisk er behov for av nettbaserte IKT-ressurser i denne konteksten. Denne oppgaven har undersøkt dette ved å foreta intervjuer og spørreundersøkelser rundt eksisterende nettbaserte tjenester for bruk i etter- og videreutdanning av lærere ved NTNU. Det har også vært et behov for å designe og evaluere nettbaserte ressurser for bruk i denne konteksten. Studiet har derfor også undersøkt utvalgte nettbaserte tjenester og verktøy og evaluert dem i kontekst av etter- og videreutdanning av lærere, i tillegg til å gi forslag til design av slike ressurser. Som et case studie følges et påbegynt prosjekt ved Program for Lærerutdanning (PLU) ved NTNU, kalt "PLU-veven." Resultatene av studiet viser at det systemet (Windows SharePoint Services 3.0) som offisielt ble brukt i første omgang for PLU-veven hadde endel svakheter, og oppgaven konkluderer med at PLU burde gå videre med en annen løsning, og foreslår alternativer og forbedringer til dette systemet.

Innhold

1	Introduksjon	8
1.1	Bakgrunn og problemdefinisjon	8
1.1.1	Bakgrunn	8
1.1.2	Problemdefinisjon	9
1.2	Roller og aktører	9
1.2.1	Forfatteren	9
1.2.2	Program for Lærerutdanning	9
1.2.3	Studentgruppa	10
1.2.4	Brukere	10
1.2.5	Verktøyadministratorer	10
1.3	Begreper og definisjoner	11
1.4	Rapportoversikt	13
2	Forskningsagenda og forskningsmetoder	14
2.1	Forskningsvinkel	14
2.2	Forskningsmetoder	15
2.2.1	Case studie	15
2.2.2	Aksjonsforskning	16
2.2.3	Evaluering av CMS og LMS	16
2.3	Tidsforløp og hendelser	18
3	Forstudie	19
3.1	CMS og LMS	20
3.1.1	Teknisk struktur og oppbygning	20
3.1.2	Kategorier	21
3.1.3	Kostnader	23
3.1.4	Bruksområder	23
3.1.5	Motstand mot CMS	24
3.2	Plu-veven	25
3.2.1	Plu-veven på kort sikt	25
3.2.2	Plu-veven på lang sikt	27
3.2.3	Kravspesifikasjon	28
3.2.4	Studentgruppas arbeid	28
3.2.4.1	Kunnskapsnivå	28
3.2.4.2	Samarbeid	28
3.2.4.3	Utviklingsmetode	29
3.2.4.4	Arbeidsmål og krav	29
3.2.4.5	Dokumentasjon	30
3.2.4.6	Utfordringer	30
3.2.4.7	Teknisk plattform	30
3.2.4.8	Resultater	32
3.2.4.9	Test av løsningen	33
3.2.4.10	Konklusjon	40
3.3	Utdanning og nettbaserte verktøy	40
3.3.1	Innenfor NTNU	41
4	Kravspesifikasjon og teori	42
4.1	PLUs kravspesifikasjon	42
4.1.1	Ikke-funksjonelle krav	43
4.1.2	Funksjonelle krav	44
4.1.3	Evaluering av kravspesifikasjonen	44

4.2	Nettbaserte verktøy.....	50
4.2.1	Artikler.....	50
4.2.2	Forum.....	51
4.2.3	Blogg.....	53
4.2.4	IM.....	56
4.2.5	RSS.....	57
4.2.6	Wiki.....	58
4.3	Designteori.....	60
4.3.1	NTNUs designmal for websider.....	60
4.3.2	Tilpasset funksjonshemmede.....	62
4.3.3	Tilpasset mobilt bruk.....	62
4.4	Utviklingsteori.....	65
5	Undersøkelse og evaluering av eksisterende nettbaserte verktøy.....	66
5.1	CMS.....	66
5.1.1	Microsoft Windows SharePoint Services 3.0 (WSS 3.0).....	67
5.1.1.1	Evalueringen av SharePoint Services.....	68
5.1.2	Joomla!.....	77
5.1.2.1	Evalueringen av Joomla!.....	78
5.1.3	It's Learning.....	90
5.1.4	WSS-utgave av plu-veven.....	93
5.1.4.1	Administrators erfaring.....	94
5.1.4.2	Test av WSS-utgave av PLU-veven.....	99
5.2	Teknologi.....	100
5.3	Spørreskjema om nettbaserte tjenester.....	101
5.3.1	Analyse av dataene.....	103
6	Design og utvikling av nettbaserte verktøy.....	104
6.1	Design av nettbaserte verktøy.....	104
6.1.1	Egenutviklet løsning.....	104
6.1.1.1	Mål.....	105
6.1.1.2	Brukerbehov.....	106
6.1.1.3	Brukbarhet.....	106
6.1.1.4	Funksjoner.....	107
6.1.1.5	Grensesnitt og interaksjon.....	114
6.1.1.6	Sikkerhet.....	118
6.1.1.7	Ytelse.....	119
6.1.1.8	Kostnad.....	120
6.1.1.9	Vedlikehold og administrasjon.....	120
6.1.1.10	Teknologi.....	121
6.1.2	Tilpasning av ferdig løsning.....	121
6.1.2.1	Tilpasning av grensesnitt.....	122
6.1.2.2	Tilpasning av navigasjon.....	130
6.1.2.3	Tilpasning av informasjonsorganisering og informasjonsflyt.....	134
6.1.2.4	Tilpasning av funksjonalitet.....	137
6.2	Utviklingsprosess.....	137
6.2.1	Egenutviklet løsning.....	138
6.2.1.1	Mål.....	138
6.2.1.2	Brukerbehov.....	138
6.2.1.3	Brukbarhet.....	139
6.2.1.4	Funksjoner.....	139
6.2.1.5	Grensesnitt og interaksjon.....	139

6.2.1.6 Sikkerhet.....	140
6.2.1.7 Ytelse.....	140
6.2.1.8 Kostnad.....	140
6.2.1.9 Vedlikehold og administrasjon.....	140
6.2.1.10 Teknologi.....	142
6.2.2 Ferdig løsning.....	142
6.3 Den organisatoriske siden.....	143
7 Diskusjon.....	148
7.1 Sammenligning og relevans av ferdige løsninger.....	148
7.1.1 SharePoint Services eller Joomla?.....	148
7.1.2 It's Learning et alternativ?.....	154
7.2 Vurdering og relevans av egenutviklet løsning.....	155
7.2.1 Studentgruppas løsning.....	155
7.2.2 Egenutviklet løsning generelt.....	157
7.3 Ferdig løsning kontra egenutviklet.....	159
8 Konklusjon og videre arbeid.....	159
8.1 Undersøkelse av nettbaserte verktøy.....	159
8.2 Design og evaluering av nettbaserte verktøy.....	160
8.2.1 PLUs valg og veien videre.....	160
9 Referanseliste.....	161
10 Appendix.....	165
10.1 Spørreskjema.....	165
10.2 Webutviklingsmetode eksempel 1.....	177
10.3 James Robertsons evalueringsskjema for CMS.....	180

Figuroversikt

Figur 1: Framsida av studentgruppas løsning.....	34
Figur 2: Svar på forumpost i studentgruppas løsning.....	38
Figur 3: Administrasjonsverktøyet for studentgruppas løsning.....	39
Figur 4: Hvordan en RSS-leser kan framstå. I dette tilfelle er informasjon matet inn fra Yahoos RSS-feed over nyhetsoverskrifter.....	57
Figur 5: Skjerm bilde fra den kjente wiki'en Wikipedia (www.wikipedia.org).....	58
Figur 6: Sidelayout fra NTNUs designmal for websider under NTNUs nettsider.....	61
Figur 7: Grensesnittdesignet i mobil utgave av Windows Live.....	64
Figur 8: Forumet i WSS med flat visning.....	75
Figur 9: Forumet i WSS med trådet visning.....	75
Figur 10: Forumkomponenten Simplest Forum i Joomla, med sin flate struktur.....	86
Figur 11: Bildet viser utdrag fra et phpBB-forum.....	88
Figur 12: Framside – Januar-08-utgave av PLU-veven bygd på WSS.....	94
Figur 13: Diskusjonsforum - Januar-08-utgave av PLU-veven bygd på WSS.....	94
Figur 14: PowerPoint-filer vises som ikoner.....	96
Figur 15: Framsida i et smalt nettleservindu. Teksten flyter og endrer sin posisjon.....	98
Figur 16: Framsida i et bredt nettleservindu.....	98
Figur 17: Figuren viser hvordan framsida i PLU-veven kan framstå som en enkel inngang for de enkelte målgruppene (PLU, lærere, lærerstudenter og praksisveiledere).....	115
Figur 18: En mulig prototype på et forumgrensesnitt integrert i et tematisert grensesnitt som i studentgruppas løsning.....	116
Figur 19: Framside i nyinstallert Joomla! (standard template).....	122
Figur 20: Framside Joomla! med endret template (Avaorus Metrobusiness).....	123
Figur 21: Framside med template "cah_ndeso".....	126
Figur 22: Innloggingsboksen er flyttet høyere opp.....	128
Figur 23: Redigering av CSS-kode for en template.....	128
Figur 24: Template-CSS endret til å inkludere ny logo.....	130
Figur 25: Fra "Module Manager" kan menyer og andre moduler publiseres eller deaktiveres.....	131
Figur 26: Redigering av hovedmenyen.....	132
Figur 27: Redigering av et menyelement.....	133
Figur 28: Hovedmenyen tilpasset PLU-veven.....	134
Figur 29: Framsida tilpasset PLU-veven.....	136
Figur 30: Arbeidsflyt ved publisering av side i et CMS (Mehring, 2005).....	147
Figur 31: Standard tema for en SharePoint-side.....	150
Figur 32: En SharePoint-side tilpasset ved bare å endre tema. (http://www.heathersolomon.com/resume/strada.aspx).....	150

1 Introduksjon

Det kreves stadig mer effektivitet i arbeid innenfor offentlig sektor, og det foregår derfor for tiden storstilte IT-investeringer i denne sektoren for å få dette til [1]. Man ønsker seg mer oversikt, bedre kommunikasjon og forbedring av arbeidsflyten. I en tid da internett anvendes til flere og flere sider av vårt dagligliv, ligger det et stort potensiale i å kombinere informasjonsteknologi med internett i såkalte nettbaserte verktøy eller ressurser. Disse nettbaserte verktøyene kan forhåpentligvis bidra til effektivisering og kunnskapsdeling. Man bør velge de nettbaserte verktøyene som passer ens faktiske behov, og dette er ikke gjort i en håndvending. For noen kan et allerede utviklet verktøy være passende. For andre må det utvikles nye verktøy fra bunnen av, skreddersydd for organisasjonens behov.

1.1 Bakgrunn og problemdefinisjon

1.1.1 Bakgrunn

Bakgrunnen for denne oppgaven var at Program for Læreutdanning (PLU) ved NTNU ønsket å styrke sin satsning på IKT i etter- og videreutdanning av lærere, samt at de så et behov for å styrke praksisveiledernes kompetanse innen forskning og utvikling (FoU). Praksisveilederne er utplassert ved PLUs partnerskoler rundt omkring i Trøndelag. Disse har i oppgave å veilede lærerstudentene som er utplassert ved disse skolene. Siden lærerstudentene starter opp med forsknings- og utviklingsprosjekt i år, ble det et umiddelbart behov for å styrke veiledernes kompetanse innen feltet. PLU så for seg at nettbaserte verktøy kunne bidra til denne kompetansehevingen, gjennom f.eks. en webportal der relevant informasjon om forskning og utvikling og annen etterutdanningsinformasjon blir lett tilgjengelig for veilederne. I et mer langsiktig perspektiv ser PLU for seg at nettstedet kan bli en landsdekkende ressurs innen lærerutdanning. Hvordan prosjektet og nettstedet vil etterhvert ta form er likevel usikkert, spesielt i det langsiktige perspektivet. PLU kan derfor sies å følge prinsippet “veien blir til mens man går” i dette prosjektet, og håpet er at denne oppgaven kan hjelpe noe på å peile ut den riktige veien for dem.

PLU bestemte seg for å la studenter ved universitetet starte utviklingen av webportalen, som ble døpt “PLU-veven”. En prototype av webportalen ble utviklet av en studentgruppe i faget IT2901 Informatikk Prosjektarbeid 2 ved IDI/NTNU, våren 2007. Prototypen ble fullført mai 2007.

1.1.2 Problemdefinisjon

Oppgaven går ut på å undersøke hvilke nettbaserte verktøy som finnes eller som vil være hensiktsmessig for bruk i etter- og videreutdanning av lærere. I dette ligger å undersøke type verktøy, funksjonalitet, brukbarhet, design, grensesnitt, kompatibilitet, vedlikehold, administrasjon, kostnader o.l. for slike verktøy. Ferdige (“out of the box”) verktøy såvel som egenutviklede og tilpasningsdyktige verktøy blir evaluert. Som et case studie følges den påbegynte webportalen (PLU-veven), for å evaluere den og sammenligne med alternative og idéelle løsninger. Dette skal kunne gi gode råd for videreutviklingen av dette prosjektet, og hjelpe PLU å gå den beste veien.

Oppgaven kan derfor sies å være todelt, der den ene delen tar for seg undersøkelsen av nettbaserte verktøy innen etter- og videreutdanning av lærere, mens den andre delen er et case studie som fokuserer på PLU-veven.

1.2 Roller og aktører

1.2.1 Forfatteren

Jeg, som forfatter av denne rapporten, spiller rollen som forsker omkring nettbaserte ressurser, og på plu-veven spesielt. I tillegg har jeg gitt råd og innspill under utvikling av studentgruppas prototype, såvel som for bruk av SharePoint-løsning. Jeg er deltagende i problemfeltet på den måten at denne rapporten vil kunne styre senere valg gjort av PLU eller andre utviklere. De innspill og råd som er gitt underveis gjør meg også deltagende i problemområdet. Jeg har hatt kontakt med PLU, studentgruppa og de potensielle brukerne av plu-veven. Kommunikasjon med de to førstnevnte har skjedd muntlig og gjennom epost, mens de potensielle brukerne (bl.a. praksisveiledere utplassert på praksisskolene) har skjedd gjennom spørreskjemaer.

1.2.2 Program for Lærerutdanning

PLU er de som har igangsatt prosjektet med å ta i bruk nettbaserte ressurser for bruk i etter- og videreutdanning av lærere, dels etter pådriv fra høyere politisk hold, dels ut fra egne behov. PLU kan sies å være kunden som løsningen skal utvikles for, enten de ansetter interne eller eksterne utviklere eller systemadministratorer. PLU fastsetter og har siste ordet i kravspesifiseringen av løsningen, selv om de lytter til andre ved utarbeidelse av løsningens kravspesifikasjon (bl.a.

studentgruppa og trolig også denne rapporten). PLU kjenner de potensielle brukerne best, og bør derfor formidle dette detaljert til systemets utviklere eller evaluerende (bl.a. studentgruppa og forfatteren av denne rapporten), siden disse har begrenset tilgang til observasjon av brukerbehov over tid.

1.2.3 Studentgruppa

Studentgruppa i faget IT2901 spilte rollen som innledende utviklere av en prototype av “pluveven.” De hjalp PLU å utarbeide en kravspesifikasjon for portalen. Deres arbeid var på sett og vis vanskelig gitt den store usikkerheten på hva PLU ønsket seg av verktøyet. Studentgruppa var i kommunikasjon med PLU, sine egne veiledere for prosjektfaget IT2901, og forfatteren av denne rapporten.

1.2.4 Brukere

Brukerne er naturlig nok de som skal ta i bruk systemet. Dette er i første omgang praksisveiledere ved partnerskolene, lærerstudenter og PLU-ansatte. På lengre sikt vil også forelesere og lærere komme mer inn, i tillegg til almenheten, hvis portalen blir et senter for etter- og videreutdanning. Brukerne har først og fremst rollen som forbrukere av systemet, og deltar ikke i utviklingen, administreringen og vedlikeholdet av systemet, unntatt å gi tilbakemeldinger til utviklere og administratorer på hvordan systemet er i bruk, eller ønsker om forbedret design eller funksjonalitet og lignende.

1.2.5 Verktøyadministratorer

Verktøyadministratorer er folk som har i oppgave å vedlikeholde og administrere portalen når den er i drift. Disse kan ha i oppgave å poste og vurdere informasjon som legges ut, sette rettigheter på informasjon på siden, og ellers utføre de oppgavene som er nødvendig for drift av portalen. Disse må gis opplæring på bruk av systemet, og vil komme til å trenge å kommunisere med andre administratorer og vedlikeholdsfolk for å koordinere oppgaver. De må ha tilgang på brukermanual for systemet, så de kjenner dets muligheter og begrensninger, og kan gi noe support til brukerne av systemet. Verktøyadministratorer vil kommunisere med PLU, brukerne og gjerne utviklere.

1.3 Begreper og definisjoner

Avatar: Datagenerert visuell representasjon av en person [4].

Nettbaserte ressurser: I et vidt perspektiv er nettbaserte ressurser alle ressurser som opererer over nett. Det være seg alle slags program eller verktøy som behandler data eller tilbyr interaksjon og kommunikasjon over et nett osv. Listen er lang over mulige typer nettbaserte ressurser. I denne oppgavens kontekst begrenses begrepet til å omfatte program eller tjenester som tilbyr kommunikasjon, deling av kunnskap og data, interaksjon og samarbeid og læring over nett. Med nett menes både Internett og intranett.

En videre avgrensning av området, er at disse nettbaserte ressursene skal kunne brukes på vanlige, moderne arbeidsstasjoner, på de vanligste operativsystemene som brukes i hjemmet og på jobben, og i de vanligste nettleserne (for nettleserbaserte tjenester).

PLU: Program for Lærerutdanning (ved Norges Teknisk Naturvitenskapelige Universitet).

Portal: En nettside som kan betegnes som en “inngangsport” til et større system. Ofte holder det at man logger seg inn én gang for alle (i løpet av en sesjon) på portalsiden slik at man slipper å logge seg på igjen på andre deler av systemet.

SCORM: Shareable Content Object Reference Model. Hensikten med SCORM er å etablere en standard protokoll for kursprogramvaren å kunne prate med dens underliggende LMS [11].

CMS: Content Management System. Et websystem hvis hovedoppgave er å administrere innhold (tekst, dokumenter, filer, mediefiler o.l.) og presentere dette i et webgrensesnitt. Det er vanskelig med en entydig definisjon av CMS siden det finnes så mange forskjellige typer systemer med forskjellige egenskaper. Wikipedias definisjon er:

A content management system (CMS) is a computer software system used to assist its users in the process of content management. A CMS facilitates the organization, control, and publication of a large body of documents and other content, such as images and multimedia resources.

WCMS: Web Content Management System. Uttrykket brukes ofte for å spesifisere at man har å gjøre med et web-/nettbasert CMS.

LMS: Learning Management System. Et LMS er et programvaresystem som integrerer nettbasert

trening, klasseromsbaserte kurs, nettbaserte kurs og system for håndtering av menneskelige ressurser [2]. Et LMS forenkler prosessen med å administrere utdanning og trening. Et LMS gir ledere, administratorer, instruktører og den som lærer en komplett e-lærings-infrastruktur som muliggjør skapelse, lagring og leveranse av unike læringsaktiviteter, såvel som administrering av studentvaktlister og studentevalueringer [2].

LCMS: Learning Content Management System. I tillegg til å behandle og administrere funksjoner for nettbasert læring, gir disse systemene verktøy for å levere og behandle instruktør-ledet, synkronisert og asynkronisert nettbasert trening basert på læreobjekt-metodologi. Et LCMS tilbyr verktøy for å skape og gjenbruke innhold (muterte læringsobjekter) såvel som virtuelle rom for læreinteraksjon (som f.eks. diskusjonsforum og chat-rom). Forskjellen på et LCMS og LMS kan vi si er først og fremst at et LMS er mest opptatt av å administrere, forbedre, planlegge og levere læringshendelser, mens et LCMS' fokus er å lære *innhold*, og gir forfattere, designere og eksperter muligheten til å *lage* gjenbrukbart læringsinnhold effektivt. På tross av denne forskjellen brukes ofte termen LMS for å referere til både LMS og LCMS.

CMS Life Cycle: Syklusen innhold i et CMS går gjennom, fra først å skapes, gjennom redigering og godkjenning, til kategorisering og levering til sluttbrukerne.

WSS: Windows SharePoint Services.

IIS: Internet Information Services (webserver).

Permalink: En permalink er en URL som peker til en spesifikk blogg- eller forumpost etter at den har blitt fjernet fra framsida og befinner seg i arkivene. En permalink har mindre sjanse for å bli irrelevant eller død med tiden enn vanlige lenker.

Podcast: En multimediafil distribuert over internett v.h.j.a. syndikeringsfeeds. Den kan så bli avspilt på datamaskiner og mobile enheter. Podcaster kan sammenlignes med opptak av radio- eller tv-serier [10]. Termen "podcast" er en sammenføring av begrepene "iPod" og "broadcast". Apples iPod var den merkevaren som de første podcast-skriptene ble skrevet for, derav navnet "podcast".

Webtjenester: Webapplikasjoner uten et brukergrensesnitt, som tillater en å tilby tjenester til andre nettsteder eller applikasjoner [9].

1.4 Rapportoversikt

Kapittel **1.0 Introduksjon** tar for seg bakgrunnen og problemstillingen for oppgaven. Kapitlet er viktig for å forstå oppgavens kontekst, og hva som vil bli undersøkt og forsøkt besvart i rapporten. Aktørene og deres roller rundt oppgaven blir forklart, i tillegg til viktige begreper og definisjoner brukt i rapporten.

Kapittel **2.0 Forskningsagenda og forskningsmetoder** tar for seg forskningsmetoder og vitenskapelig framgangsmåte ved utførelse av oppgaven. Kapitlet begynner med en forklaring av det dypere forskningssynet oppgaven grunner på, før forskningsmetodene forklares. Kapitlet avsluttes med en oversikt over hendelser og tidsforløp for oppgaven.

Kapittel **3.0 Forstudie** framstår som et forstudie for selve oppgaven. Konsepter som CMS og LMS blir gjennomgått på en generell måte, siden disse systemtypene er sentrale i oppgaven. Et detaljert studie av prosjektet “plu-veven”, hva som hadde blitt gjort og hva som sto klart ved oppstart av denne oppgaven blir gjennomgått. En introduksjon til nettbaserte verktøy i en utdanningskontekst spesielt blir også introdusert i dette kapitlet.

Kapittel **4.0 Kravspesifikasjon og teori** presenterer PLUs kravspesifikasjon for “plu-veven”. Teori rundt de ulike sidene av nettbaserte verktøy og forskjellige typer av disse blir gjennomgått. Teori omkring design og teknologi for nettbaserte verktøy blir til slutt forklart.

Kapittel **5.0 Undersøkelse og evaluering av eksisterende nettbaserte verktøy** undersøker spesielt utvalgte og aktuelle nettbaserte verktøy (CMS-baserte eller andre typer) for bruk i etter- og videreutdanning av lærere. En oversikt over teknologi brukt og tilgjengelig for bruk i slike verktøy blir også presentert.

Kapittel **6.0 Design og utvikling av nettbaserte verktøy** greier ut om case studiet som omfatter design og utvikling av et nettbasert verktøy. Både egenutviklet løsning og tilpasning av ferdige løsninger blir gransket.

Kapittel **7.0 Diskusjon** diskuterer hvor relevant og aktuelt forskjellige nettbaserte verktøy er, sammenlignet med hverandre, i kontekst av etter- og videreutdanning av lærere. Både ferdige og egenutviklede løsninger blir tatt for seg.

Kapittel **8.0 Konklusjon og videre arbeid** gir konklusjon på oppgavens to delspørsmål; undersøkelse av nettbaserte verktøy for etter- og videreutdanning av lærere, og design og evaluering

av nettbaserte verktøy til dette formålet. Teksten avsluttes med råd om veien videre for "plu-veven".

2 Forskningsagenda og forskningsmetoder

2.1 Forskningsvinkel

Debatten har gått, innenfor vitenskapen generelt, og også innenfor informasjonsvitenskap, rundt metoder for "korrekt" forskning. De debatterendes standpunkter grunner i deres idéelle syn på hvordan forskning burde gjennomføres, som igjen kommer av syn på forskningsargumentasjon og virkelighetsforståelse. Spesielt to sider har stått mot hverandre; positivisme og interpretativisme [3]. Positivisme grunner mer i naturvitenskapelige forskningsmetoder, og ser på virkeligheten som objektiv, med "harde" realiteter og fakta. Tall er akseptert som spesielt viktig og kan brukes for å bevise hypoteser. Positivisme har eksistert lenge innen vitenskapen (hundreder av år). Interpretativisme (fortolkende), på den andre siden, ser mer subjektivt på virkeligheten. Man har ikke nødvendigvis harde, objektive fakta man kan derivere ut fra naturen rundt oss, men ethvert subjekt opplever sin egen virkelighet. "Virkeligheten" kan således varieres fra person til person. Interpretativisme har blitt mer anvendt (spesielt på 90-tallet) i sosiale vitenskaper. Selv om den kan virke ny, har dette forskningssynet også eksistert tidligere, vel og merke ikke så fremtredende i vitenskapelige kretser som positivisme. Interpretativisme bruker mer ord enn tall for å beskrive virkeligheten.

Denne oppgaven tar for seg et bredt spekter av sider rundt nettbaserte verktøy for bruk i utdanning, og prøver å se på helheten for på den måten å vurdere den beste løsningen og nytten av den. Positivistiske argumenter og metoder brukes på aspekter av verktøyene som kan undersøkes på denne måten. Det gjelder bl.a. ytelse (responstid, belastning etc.), størrelse (mengde kode, ressurser påkrevd etc.), tekniske krav (OS, plattform, maskinytelse etc.), designkrav (skjermopløsning, fargebruk, fonter etc.), sikkerhet (identifisering av feil i koden/systemet som tillater angrep), datalagring (lagring av gamle artikler og brukerbidrag på nett) og kostnad (total kostnad, kostnader for enkeltkomponenter, vedlikeholdskostnader, organisasjonskostnader etc.). Interpretativistiske argumenter og metoder brukes også på noen av de samme aspektene ved verktøyene, og på aspekter som brukbarhet (intuitivitet, logiskhet, læringskurve etc.), interaktivitet (grensesnitt, funksjoner, arbeidsprosesser etc.), kostnader (organisasjonskostnader, personressurser etc.), design (visuelt grensesnitt, layout, stil etc.), intervjuer og spørreskjema (tolkning og analyse) osv. Oppgaven kan derfor ses på som stående et sted midt i mellom ytterpunktene mellom positivisme og

interpretativisme.

2.2 **Forskningsmetoder**

Gjennom hele oppgaven ble det benyttet en empirisk framgangsmåte, der fenomenene ble observert og testet i den virkelige verden. Teoretisk forskning ble gjort meget begrenset, og kun rundt utviklingsprosesser. Den delen av oppgaven som konsentrerte seg om “plu-veven” ble utført som et **case studie**, der det spesifikke tilfellet av design, utvikling og innføring av nettbaserte ressurser, sammenfattet i et websystem (“plu-veven”), i en institusjon (PLU) ble undersøkt.

Siden forfatteren av denne rapporten selv hadde mulighet til å påvirke både produkt og prosess gjennom sitt studie, hadde angrepsmåten også et snev av **aksjonsforskning**. Han deltok i noen møter om “plu-veven” med studentgruppa, og man kan ikke se bort fra at utviklingen av dette systemet i mindre grad ble påvirket av dette. Forfatteren har også blitt spurt av PLU om han kunne videreutvikle noe på systemet, så vel som hva hans mening var i forskjellige situasjoner. Noe av grunnen til dette var at systemet tidlig skulle tas i bruk, og det var ønskelig med et litt annet system enn det man hadde foran seg. Dessuten følte PLU de trengte å høre andres meninger før de tok sine beslutninger. Det ble gitt hjelp etter beste evne, ut i fra den kunnskap og erfaring man til da hadde bygd seg opp rundt problemstillingen. Det er usikkert i hvor stor grad dette har påvirket PLUs beslutninger.

2.2.1 **Case studie**

Et case studie er et dybdestudie av en situasjon. Mitt studie av PLU-veven er et slikt case studie. De fleste case studier følger en situasjon over tid. Så også med studiet av PLU-veven. Prosjektet ble fulgt og studert over hele oppgaveperioden, så langt det var mulig. Et case studie kan ses på som et innledende studie til et emne, som andre siden kan bygge teorier rundt, handle etter eller hente ut kunnskap fra. Slik kan også andre bygge videre på PLU-veven ut fra konklusjonene til dette studiet. Walsham [5] mener at man kan få et stort utbytte av tolkende case studier, når det gjelder å få en dypere forståelse av informasjonssystemer. Han er heller ikke spesielt bekymret over problemene med å generalisere fra slike case studier, fordi han argumenterer med at gyldigheten for en ekstrapolering fra et individuelt case eller flere case studier ikke avhenger av den statistiske representasjonen av dem, men av plausibiliteten og overbevisningen til det logiske resonnementet brukt i studiene for å forklare resultatene og slutte konklusjoner fra dem.

Begrensninger med case studier er vanskelighetene med å kontrollere variabler, og finne årsakssammenhenger. Dette kan bekreftes med studiet av PLU-veven. Erfaringen med studiet er at det involverer mange variabler og sider, og på grunn av dette er det vanskelig å gå for mye i dybden på ett spesielt område, la oss si funksjonalitet, av studiet. Man må forsøke å gå i dybden på alle berørte sider av saken, og der lå utfordringen i studiet av PLU-veven.

2.2.2 Aksjonsforskning

Action research, eller aksjonsforskning på norsk, er en framgangsmåte der forskeren ikke kun observerer hendelser men til en viss grad deltar selv i problemsituasjonen sammen med subjektene [3]. Innenfor forskning på informasjonssystemer og datavitenskap kan det tenkes forskeren deltar på f.eks. design, programmering, kravspesifisering eller testing av datasystemer. Forskeren kan bruke sine teoretiske kunnskaper til å forme aktivitetene han/hun deltar i. I tillegg kan forskeren relatere hendelser til før disse teoretiske kunnskapene ble anvendt. Fordelen med aksjonsforskning er at det kan sette forskeren nærmere problemområdet i en situasjon i den virkelige verden. Man kan se ting gjennom subjektens øyne, og fokusere på hva de faktisk *gjør*, og ikke bare *sier* de gjør. Faren er at forskeren blir for deltagende og sneversynt i sin forskning, og derfor mister det hele og fulle bildet av forskningstemaet.

Noe av arbeidet med denne oppgaven har tatt form av aksjonsforskning, siden forfatteren av denne oppgaven har selv samlet inn noe kravspesifikasjon for systemet, gjort forslag til design og utvikling, og testet initiale såvel som alternative systemer. Likevel er det forsøkt å holde vedlike totalbildet av forskningstemaet, og begrunnelsen for deltagelse i problemområdet er begrunnet med både ønsker fra subjekter såvel som med at det ville gi det grundigste arbeidet og beste utgangspunktet for videre arbeid med temaet.

2.2.3 Evaluering av CMS og LMS

For å evaluere CMS, og avgjøre om det passer til organisasjonens forretningsmål, er James Robertsons [6] vurderingskriterier for CMS i hovedsak brukt. Evalueringsskjemaet med detaljerte kriterier er vedlagt Appendix 10.3. Grovt sett går metoden ut på å ta for seg forretningsmål og strategier (for bedriften som ønsker å implementere et CMS) og krav til systemet. Ut i fra dette vurderes CMS'et med hensyn på innholdsskaping, innholdsbehandling, publisering, presentasjon og kontrakter og forretning.

Verdt å ta i betraktning er at Robsertson utarbeidet retningslinjene ut fra disse antagelsene:

Antagelser om organisasjonen som skal ta i bruk systemet:

- Medium til stor organisasjon
- Nåværende publiseringsystemer blir erstattet av nytt CMS
- CMS'et vil behandle både intranett- og internettwebsidene
- CMS'et vil være organisasjonsbrett

Antagelser om typisk innhold som vil bli publisert gjennom CMS'et:

- Enkle sider
- Komplekse sider med spesifikk layout og presentasjon
- Dynamisk informasjon fra databaser etc.
- Treningsmateriale
- Online manualer (politikk og prosedyrer etc.)
- Generelle forretningsdokumenter
- Utbredt lenking mellom sider
- Et volum på tusener av sider totalt

Ikke alle disse antagelsene stemmer med PLUs visjoner for PLU-veven. Antagelser som er **uaktuelle** for PLU-veven er

- Nåværende publiseringsystemer blir erstattet av nytt CMS (PLUs internettsider vil fremdeles være de som allerede er utviklet og er en del av NTNU-portalen)
- CMS'et vil behandle både intranett- og internettwebsidene (Innsida vil fremdeles være i bruk som intranettsider)
- Generelle forretningsdokumenter
- Et volum på tusener av sider totalt (på lang sikt kan dette være relevant)

Man bør derfor ikke forvente at retningslinjene nedenfor passer helt for PLU-veven, men de kan gi en god pekepinn på hvilke CMS som passer til PLUs formål.

Metoder og kriterier for å evaluere et LMS er framsatt av Yildirim et al [2]. Evalueringsskjemaet er

ikke gjengitt i besvarelsen siden LMS ligger litt på siden av hvilke typer nettbaserte verktøy oppgaven fokuserer på. Skjemaet bruker en avkryssingsskala som går fra 0 til 4, der 0 = ikke oppnådd (ikke-eksisterende eller dårlig), 1 = delvis oppnådd (lavt nivå), 2 = moderat oppnådd (gjennomsnittlig grunnleggende nivå), 3 = for det meste oppnådd (godt) og 4 = fullt oppnådd (overgår vanlig forventet standard). Med litt tilpasning kan dette avkryssingsskjemaet brukes for å evaluere systemer mot kravspesifikasjonen til PLU-veven, eller til å evaluere CMS. Dette fordi LMS og CMS ikke er så forskjellige (se kap. 1.3). Vice versa kan metoder for å evaluere CMS (bl.a. metoden nevnt ovenfor) med litt tilpasning brukes til å evaluere LMS.

2.3 Tidsforløp og hendelser

Hendelser som utmerket seg mer enn andre er iløpet av prosjekttiden er summert opp under.

Vår 2007:

- Mastergradsoppgaven ble påbegynt ved å sette seg inn i noe bakgrunns litteratur omkring læring via nettet, og e-lærings- og innholdsadministrasjonssystemer.
- Siden studentgruppa og deres arbeid skulle følges, ble det nødvendig å sette seg noe inn i programmering i C#, og .NET-teknologien.
- Jeg måtte regelmessig oppdatere meg på studentgruppas arbeid for å følge med utviklingsprosessen i “plu-veven” case studiet.

Høst 2007:

- Etter at studentgruppa var ferdig med prosjektet sitt, var en grunnmur av PLU-veven ferdigstilt. Testing av webportalens funksjonalitet og brukergrensesnitt mot kravspesifikasjonen kunne da tas fatt på av meg selv.
- Det ble opprettet kontakt mellom meg og den personen ved PLU som nå hadde tatt over hovedansvaret for PLU-vev-prosjektet. Vi hadde et møte hvor vi samtalte om positive og negative sider ved studentgruppas løsning. Epost ble også ofte brukt som kontaktmiddel mellom oss.
- Jeg sendte ut innledende spørreskjema til potensielle brukere av PLU-veven.
- I utgangspunktet var det tiltenkt at jeg skulle gjøre en del videreutvikling og design på studentgruppas system. Men på grunn av usikkerheten rundt hvorvidt PLU ville satse videre på systemet, avgjorde jeg, i samråd med veileder, at jeg istedet fokuserte mer på litteratur og

teori enn selve videreutviklingen av systemet. Det skulle også senere vise seg at PLU forlot studentgruppas løsning.

- FoU-kurs for praksisveiledere, som PLU-veven skulle være et hjelpemiddel for, var planlagt å starte i oktober måned. Dette ble utsatt på ubestemt tid. Dette forsinket min observasjon av brukernes bruk av webportalen. Brukertesting måtte utsettes, såvel som annen datainnsamling relatert til brukernes reaksjoner på systemet.
- SVT-Drift kom mer og mer i dialog med PLUs prosjektgruppe for PLU-veven. De foreslo å integrere et out-of-the-box forum (phpBB) til studentgruppas løsning, og vurderte i tillegg om det fantes alternativer til studentgruppas løsning.
- PLU var lenge i tenkeboksen på hvorvidt de skulle satse på studentgruppas løsning, eller gå for en ny løsning. Etter å ha samrådd seg med driftavdelingen på SVT-fakultet, bestemte de seg for å gå for en ny løsning med SharePoint Services i grunnen. Avgjørelsen så ut til å falle etter et møte mellom SVT-drift og PLUs prosjektgruppe for PLU-veven, der SharePoint som CMS ble demonstrert for prosjektgruppa. Begrunnelsen for avgjørelsen er omtalt i denne rapporten. Avgjørelsen medførte endringer for mitt studie, da jeg nå måtte inkludere SharePoint-systemet i evalueringen. Studiet dreide seg mer i retning av et breddestudie, og hovedfokuset ble flyttet fra studentgruppas løsning mot en evaluering og sammenligning av disse to systemene, i tillegg til andre, alternative systemer.
- Etterhvert begynte jeg innsamling av data for å undersøke behovet for nettbaserte IKT-ressurser i etter- og videreutdanning av lærere ved NTNU. Flere spørreskjema ble sendt ut og svarene samlet inn. Analyser av svarene ble utført senere.
- Jeg forberedte intervjuer og utførte dem med noen med sentrale personer ved PLU, spesielt prosjektgruppa for PLU-veven.
- Jeg ferdigstilte en prototype på en alternativ løsning i open source systemet Joomla! Denne har bare blitt brukt til å få innblikk i tilpasning av et ferdig system til PLUs kravspesifikasjon.
- Jeg begynte på testing og evaluering av SharePoint Services.

Vår 2008:

- Uttesting av SharePoint Designer, et program for å utvide og designe websider laget i SharePoint Services.

3 Forstudie

Hensikten med forstudiet er å orientere seg og rekognisere seg i problemstillingen. Spørsmål man

ønsker besvart er: finnes det eksisterende studier om nettbaserte verktøy i bruk i etter- og videreutdanning eller utdanning generelt? Hvilke eksisterende nettbaserte verktøy finnes? Hva kjennetegner CMS og nettbaserte verktøy? Forstudiet har også tatt for seg studentgruppas løsning og evaluert denne.

3.1 CMS og LMS

Content Management Systemer (CMS) er nettbaserte verktøy som er brukt i en rekke organisasjoner, alt fra offentlige til private bedrifter, som bedriftens hjemmesider og konsernets intranett, til nettsider for spillsamfunn og frivillige organisasjoner. CMS gjør at man kan sette opp alt fra enkle nettsider til større webportaler som er integrert med eksisterende systemer. Et CMS innehar ofte en rekke mindre verktøy eller funksjonaliteter, som fungerer som en helhet i CMS'et. Det samme gjelder for et LMS (Learning Management System). For fem år siden ble det skrevet i The Gilbane Report [7], en rapport som kommer ut flere ganger i året og som er forfattet av Gilbane Group¹, at selv en fulltids ansatt markedsanalytiker betalt for å være en innholdsadministrasjonsekspert ikke vil kunne ha full oversikt over alle forskjellige produkter og egenskaper som eksisterer for behandling av innhold. Derfor fokuseres det i dette kapitlet (og rapporten forøvrig) på det som antas å være det viktigste.

3.1.1 Teknisk struktur og oppbygning

Open source CMS har ofte en teknisk struktur vi kan kalle AMP (eller LAMP). Dette er et akronym som står for Apache, MySQL og PHP. Apache er webserveren som benyttes, MySQL er databasespråket og PHP er et server-skriptspråk. Akronymet LAMP skiller seg bare ved at L står for Linux. AMP vil være mest riktig i denne sammenhengen fordi Apache, MySQL og PHP (og dermed hele CMS'et) kan også kjøres på andre OS enn Linux.

En annen teknologi som benyttes (spesielt i de senere år) for utvikling av CMS (og websider generelt) er AJAX. AJAX står for Asynchronous JavaScript and XML (Ajax Tutorial, W3 Schools). Dette språket er relativt nytt for bruk i CMS-sammenheng. Teknologien er bl.a. kjent for å muliggjøre raskere lasting av websider, ved at man bare oppdaterer deler av nettsiden (i stedet for hele nettsiden) om bare en liten del av siden trenger å oppdateres. Nettsiden vil derfor ligne mer på en skrivebordsapplikasjon i grensesnittets oppførsel.

¹ www.gilbane.com. The Gilbane Group beskriver seg som analytikere med 20 års erfaring av anvendelse av innhold og informasjonsteknologi for bedriftsapplikasjoner.

Vi har også de websidene (og CMS) som bygger på .NET og ASP.NET. Ofte benyttes da også C#. Et eksempel på et CMS utviklet i C# er DotNetNuke². ASP.NET er spesielt godt egnet til datadrevne interaktive websider som kommuniserer med en database på serveren [9]. Et eksempel på CMS som bygger på .NET og ASP.NET er Microsoft SharePoint Services.

CMS Review³ inkluderer følgende punkter for å summere opp den tekniske siden av et CMS (tatt fra deres forslag til CMS-taksonomi):

Lisens: Open source eller proprietært.

Type: Generelt CMS, rammeverk, frontend (UI), nyhetsportal, blogg, Wiki etc.

Plattform: Plattform som systemet kjører på (Windows, Linux, Mac, etc.).

Webserver: Webserver som benyttes for systemet, f.eks. IIS eller Apache.

Applikasjonsrammeverk: Kan f.eks. være Perl, Python, .NET, J2EE, PHP eller Cold Fusion etc.

CMS-Rammeverk: Kan f.eks. være Axkit, Cocoon, Midgard eller Zope etc.

Språk: Kan f.eks. være Perl, VB, Java, PHP, Python etc.

Databaser: Kan f.eks. være Oracle, SQL Server, MySQL, PostgreSQL, ODBC, etc.

API: API kan gjøres tilgjengelig for at andre kan lage utvidelser.

3.1.2 Kategorier

The Gilbane Report [7] forteller om en innsats for å lage et offentlig tilgjengelig klassifikasjonssystem for CMS-produkter og -egenskaper. Dette beskrivelsessystemet skal være basert på et språk kalt CMSML som er basert på XML (CMSML er et samarbeidsprosjekt mellom OSCOM softwareML project, University of Washington iSchool CMS Evaluation Lab, og CMS Review⁴, og skal være tilgjengelig på www.gilbane.com. Det finnes også andre store rapporter og evalueringer av CMS. CMS Watch beskriver seg selv som en forretning som evaluerer innholdsorienterte teknologier og publiserer side-ved-side-evalueringer av ledende løsninger (www.cmswatch.com). De publiserer omfattende og oppdaterte rapporter, men pga. høy pris for å

² <http://csharp-source.net/open-source/content-managment-systems/dotnetnuke>

³ www.cmsreview.com

⁴ Beskrevet på www.cmsml.org

anskaffe en slik rapport, vil denne oppgaven ikke kunne trekke mye informasjon derfra.

“Content Management” har blitt et “alt-inkluderende” moteord for alt som har med dokumentbehandling, informasjonsbehandling og kunnskapsbehandling å gjøre [12]. “Content management” omfatter derfor en stor mengde funksjoner og tjenester, noe som gjør det dess verre å kategorisere.

Som The Gilbane Report sier, bør man forstå hva et CMS *gjør* for å kunne evaluere det, og ikke minst hva nettstedet vil ha gjort (målene et CMS skal tilfredsstille). Deres egenskapsliste for CMS er inndelt i 5 store områder:

- Produktoversikt (beskrivelse, status, markedsføring, installasjon og støtte)
- Innholdsskaping (tilegnelse, aggregering, forfatterskap)
- Content Management Proper (arbeidsflyt, redigering, godkjenninger, oppstilling, oppbevaring osv.)
- Innholdsleveranse (live server, publisering, syndikering)
- Forbedringer i innholds-livssyklus (sikkerhet, forretningsregler, integrasjon, metadatabehandling, transformasjoner, assosiasjoner, analyse, søk og lokalisering)

De forskjellige CMS kan på den måten klassifiseres ut fra egenskapene opplistet ovenfor. Det følger at flere CMS har like resultater på mange av punktene, men veldig sjelden (om i det hele tatt) at systemer er lik på alle egenskaper. Metoden hjelper å skille de forskjellige CMS og type systemer fra hverandre, og få fram aktuelle CMS som tilfredsstiller betingelser og krav man måtte ha til systemet (f.eks. søk på alle systemer som innehar en spesiell type oppbevaring av innhold, bruker oppegående (live) server for leveranse av innhold, og kan integreres på den og den måten). Man må være oppmerksom på at de forskjellige CMS ikke kan sammenlignes egenskap for egenskap uten videre, siden det hele bildet er mer komplekst enn som så. Som Gilbane Report også sier, så må klassifiseringsmetoden ses på som en måte å få et utgangspunkt i sammenligning og evaluering av CMS på, ikke som en komplett metode. Noen av uttrykkene og beskrivelsene i metoden har med vilje eller ved uhell blitt gjort tvetydige, slik at de kan tolkes forskjellig av de forskjellige leverandører. Dette er delvis gjort for at man skal arbeide videre sammen med kvalifisert personell for å velge det riktige CMS. I en jungel av forskjellige systemer ble denne klassifiseringsmetoden delvis lagd for å lage et kontrollert vokabular for CMS-industrien - delte konsepter som industrien kan enes om [7].

3.1.3 Kostnader

Kostnader for et CMS kan variere mye. Vi har proprietære systemer som kan komme opp i flere titalls eller til og med hundretusenere av kroner, til gratis open source systemer (men som kan ha store kostnader tilknyttet seg når det gjelder innføring og tilpasning). Man kommer stort sett ikke unna at en investering i og drift av et CMS blir kostbart. Dette må veies opp mot fordelene man oppnår ved innføring av et CMS eller en nettbasert ressurs.

Det er mange typer kostnader tilknyttet et CMS, som i sum utgjør *totalkostnaden* ved systemet. En av delkostnadene er *komponentkostnader*, som er de kostnadene tilleggskomponenter krever, enten i form av rent innkjøp av disse, eller utvikling og tilpasning av dem. Man har også *drifts- og vedlikeholdskostnader* ved et CMS, som er de kostnadene tilknyttet oppetid, oppdateringer, feilrettinger, ytelsesforbedringer, integrering med eksisterende systemer og lignende. Disse kan bli svært høye selv om innkjøpskostnadene for systemet er lavt. Her er det også av betydning at systemet har et godt støtteapparat som kan være behjelpelig med feilretting, svar på spørsmål, oppdateringer og integrasjon, slik at man slipper å være sitt eget støtteapparat. Man har også *organisasjonskostnader* tilknyttet et CMS. Disse kostnadene er knyttet til administrasjon av systemet, som kan bestå av skaping, godkjenning og behandling av innhold [13], utseendemessig design av websider, og andre organisatoriske sysler som planlegging av systemets utviklingsforløp og forretningsmål systemet skal tilfredsstille. Kostnader kan være både i form av penger eller ressursbruk.

3.1.4 Bruksområder

Som påpekt i seksjon 3.1.2 så kommer mange forskjellige typer CMS under paraplyen “content management”. Det finnes systemer som per definisjon kommer under begrepet CMS, men som har endt opp under begrepet ECMS (Enterprise CMS). Eksempler på slike systemer er systemer som spesialiserer seg på [14]:

- Planlegging av bedriftens ressurser. Disse kalles Enterprise Resource Planning Systems (ERP).
- Behandling av kundeforhold. Disse kalles Customer Relationship Management Systems (CRM).
- Dokumentbehandling. Disse kalles Document Management Systems (DMS).

- Behandling av menneskelige ressurser. Disse kalles Human Resource Management Systems (HRM).

Bare av ovennevnte finnes en mengde systemer tilgjengelig på markedet. Web CMS er CMS som kjører på en web server, og kan aksesseres fra en nettleser. I lys av denne oppgaven er det mest relevant å fokusere på Web CMS, siden det kommer under oppgavens forskningsspørsmål; undersøkelse av *nettbaserte* ressurser eller verktøy. Bruksområder for slike systemer kan f.eks. være:

- Diskusjoner i online fora
- Nettsamfunn
- Presentasjoner og behandling av blogger
- Behandling og presentasjon av multimedia
- Behandling og presentasjon av online artikler
- Deling av kunnskap over nett
- Sanntidskommunikasjon
- Versjonskontroll
- Auksjoner og netthandel
- Hjemmesider

Dette er naturlig nok bare et utvalg av bruksområdene for Web CMS.

3.1.5 Motstand mot CMS

Siden oppgaven handler om å se på bl.a. et case studie av innføring av nettbaserte verktøy i en organisasjon, vil det være naturlig å se litt på hvordan erfaringen er med innføring av slike systemer. Det kan ofte være lærerikt å se på vanlig motstand mot bruk av CMS [15]. Selv om et CMS definisjonsmessig ikke helt tilsvarer et WCMS, som PLU-veven forventes å være, så overlapper disse to systemtypene mye.

Utviklere

Disse vil ikke være komfortable med et system som ikke er minst like raskt som eksisterende system i bedriften. Dette gjelder både systemets båndbredde og tiden det tar å publisere i systemet.

CMS som ikke har fleksibilitet til å støtte CGI, Java, JavaScript, Flash, XML og andre webteknologier vil ikke ha like stor støtte fra utviklere. De fleste utviklere vil tro at mange av dem vil bli overflødige ved innføring av nye, mer effektive CMS. Sannheten er at det vil fortsatt være behov for dem, til dynamisk skripting og design.

Innholdsskapere

Innholdsskapere vil ofte ikke se fordelene av et nytt Web CMS før systemet er oppe og kjører. De vil forstå at det medfører endringer, og argumenterer mot innføringen med at det blir mer å gjøre for en innholdsskaper i et Web CMS. Realiteten er at det sannsynligvis blir et nytt ledd i prosessen for å publisere innhold; godkjenning av innhold. Prosessen må kartlegges på forhånd, for å unngå forvirring. Et CMS vil gi dem mer direkte tilgang til publiseringen, og gi dem et større ansvarsområde, og det blir en læringskurve å komme over i starten.

Ledere

Ledere ser vanligvis verdien av prosessen, publiseringsgodkjenningen og ledelsesaspektene introdusert i et CMS, men de har større problemer med å akseptere kostnaden ved et CMS. Det lønner seg å regne ut hvor mye man forventer å tjene på å introdusere CMS'et, satt opp mot alle kostnadene ved det. Dette kan lettere gi ledere oversikt over den korrekte lønnsomheten med et CMS.

3.2 Plu-veven

“Plu-veven” var navnet PLU sitt nylige påbegynte prosjekt for nettbaserte ressurser innen etter- og videreutdanning av lærere fikk. Studentgruppa (se kap. 1.2.3) påbegynte med en prototype-løsning av en cms-lignende webportal, laget fra skratsj. Forfatter av denne rapporten fulgte deres arbeid, leste rapporten de hadde produsert og evaluerte løsningen de hadde utviklet.

3.2.1 Plu-veven på kort sikt

På kort sikt skulle plu-veven dekke et nærliggende behov for å øke kompetansen på praksisveilederne på partnerskolene (se kap. 1.1.1). PLU bestemte seg for å la en studentgruppe i faget IT2901 Informatikk prosjektarbeid 2 påbegynne utviklingen av systemet. På spørsmål hvorfor studenter fikk oppgaven med å starte utviklinga av PLU-veven, svarte et intervjuobjekt slik:

”I utgangspunktet fordi de er innenfor et universitet. Det er ikke nødvendigvis slik at interne utviklere er best, men vi ville få inn litt nye idéer. Også det at studenter fikk en konkret oppgave med en tidsfrist. PLU har kontakter på IDI, og IDI har prøvd å få til et samarbeid med PLU mange ganger, uten at de har hatt noe konkret.”

Teksten for studentoppgaven var som følger:

”Program for lærerutdanning har et partnerskap med 36 skoler (grunn og vgs.) som hvert år får utplassert lærerstudenter slik at studentene får praktisk erfaring i det yrket de utdanner seg til. Fra og med i vår skal eksamen for disse studentene inkludere et såkalt FoU arbeid, et forsknings og utviklingsarbeid, som skal ta utgangspunkt i denne praksiserfaringen som de får fra skolen de oppholder seg på. Dette gir oss en ekstra utfordring fordi dem som skal veilede disse studentene i skolen (koordinatorene) ikke har den nødvendige veilederkompetansen som studentene trenger for å gjennomføre arbeidet. Derfor ønsker vi å tilrettelegge for at veilederne kan tilegne seg denne kompetansen via Internett. Vi ser for oss en webside som inneholder fagmoduler, tilrettelagt av våre egne forelesere og forelesere utenfra, som sørger for at disse koordinatorene får den kompetansen de trenger. Fagmodulene består av ulike ressurser; alt fra enkle PowerPoint og artikler til litt mer ”avansert” innhold som videostreaming og Podcasting. I tillegg vil lærerstudentenes eget FoU arbeid være å finne her, slik at koordinatorene kan se konkrete eksempler på FoU arbeid som er gjennomført i skolen. Vi ønsker at dette skal være et åpent nettsted hvor vi tilgjengeliggjør nødvendig informasjon, såkalte ”kurspakker” eller ”moduler”. I tillegg til dette mer el. mindre statiske innholdet er det nødvendig med en el annen form for interaktivitet. Det er viktig at koordinatorene kan stille spørsmål rundt FoU praksisen og sin egen veiledning av studentene. Kanskje kan dette gjøres ved bruk av Chat verktøy eller en blogg? Nettstedet må være oversiktlig og tiltalende slik at det blir tatt i bruk. Får vi til denne type ”veiledningstjeneste” ser vi for oss en fremtidig utvidelse som også kan omfatte et regionalt evt. nasjonal nettverk for og av lærere og lærerutdannere.”

3.2.2 Plu-veven på lang sikt

På lengre sikt ønsker PLU at plu-veven skal om mulig bli en nasjonal arena for etter- og videreutdanning, og lærerutdanning generelt. PLU innrømmer at dette er et ambisiøst mål, men argumenterer bl.a. med at prosjektet har mange år til rådighet for å nå dette målet. Dette målet er dessuten ikke så fastspikret, siden PLU selv ikke er helt sikre på hvilken vei utviklingen av prosjektet vil ta i framtiden. Et intervjuobjekt ved PLU formulerte det som at det viktigste var at det gikk framover, og at webportalen måtte få fotfeste i organisasjonen. Et annet intervjuobjekt ved PLU, når konfrontert med spørsmålet om hvilke forventninger vedkommende hadde til plu-veven på kort og lang sikt, formulerer svaret slik:

"Jeg har realistiske forventninger. Det snakkes ofte med store ord, men det avhenger av menneskene. Prosjektet krever strikt organisering, men jeg tror vi skal få det til. Jeg tror at det vil gå sakte fram. Innholdet må være variert, med forskjellige digitale verktøy innenfor portalen. Vi har en forestilling om at alle lærere i Norge skal diskutere der."

Et annet intervjuobjekt hadde små forventninger til plu-veven:

"Plu-veven har jeg små forventninger til. Jeg tror ikke at det er godt nok forankret i organisasjonen her, og at de ansatte er motivert nok til å bidra. Men jeg tror bare."

En annen uttrykker ønske for en mer databank-tilnærming for den fremtidig plu-veven:

"Jeg tror ikke vi trenger nye webportaler i tradisjonell forstand. Hvis vi derimot hadde fått utviklet en nasjonal "ressursbank" som hadde gode metadata som gjorde mulig kommunikasjon med likesinnede nettsteder internasjonalt, en ressursbank for ulike ressurser, så hadde det vært bra. Jeg tror ikke det eksisterer en slik "bank" ennå. Men det er mange spørsmål knyttet til dette, bl.a. om standarder osv."

Noen intervjuobjekter var litt skeptiske med tanke med ressurskostnaden av plu-veven, som f.eks. uttrykt i dette svaret:

"Spurte de med pluveven hvor mange millioner de hadde tenkt å bruke på dette og for å vedlikeholde det. Jeg føler at man for ofte prøver å gjøre for mye med for lite ressurser ved PLU, med for liten tid også... Skepsisen henger

mye sammen med de ambisiøse målene i starten."

3.2.3 Kravspesifikasjon

Krav til systemet på kort sikt, delvis spesifisert av PLU selv, er bl.a. at systemet må være så brukervennlig at selv datakyndige personer kan ta det i bruk, med minimal opplæring. Digitale ressurser, som presentasjonsfiler (a la PowerPoint), artikler, dokumenter og videofiler skal være tilgjengelige via nettstedet. I tillegg bør funksjonaliteter som søk, kategorisering/tematisering, medlemsregistrering (for å få tilgang til beskyttede ressurser) og forum være ferdig implementert. Navigasjonen må være enkel og intuitiv, og en brukerveiledning må følge med, spesielt for de som skal administrere systemet. Kravspesifikasjonen er nærmere studert i kap. 4.1

3.2.4 Studentgruppas arbeid

Studentgruppa i it2901 ble fulgt og observert gjennom deres arbeid med prosjektet våren 2007. Møtereferater, statusrapporter (selv om det var få av dem), diagrammer, dokumentasjon og kode ble studert, og deltakelse på noen av gruppas møter var nødvendig for å få et innblikk i utviklingsprosessen og utviklinga av systemet.

3.2.4.1 Kunnskapsnivå

Kun én person på gruppa hadde tidligere erfaring med programmeringsspråket (C#) og .NET rammeverket. De andre måtte lære seg dette i løpet av prosjektiden. Dette resulterte i at det ofte var den erfarne som sto for det meste av programmeringa. De andre programmerte også, men han ble den som ofte måtte hjelpe de andre. Til gjengjeld var han ingen nybegynner i teknologien. Vedkommende var også prosjektleder for gruppa, selv om gruppa hadde noenlunde flat organisatorisk struktur. Det var også noen på gruppa som var i arbeid og således hadde arbeidserfaring, noe som sikkert kom godt med i en prosjektsammenheng.

3.2.4.2 Samarbeid

Én person var prosjektleder, men gruppa hadde likevel en noenlunde flat organisatorisk struktur. Prosjektlederens oppgave var å se til at ting ble gjort til riktig tid og at gruppa hadde progresjon. En annen fikk oppgaven med å ta hånd om dokumenthåndtering og mindre organisatoriske deler av prosjektet. En annen person fikk ansvaret for møteinnkalling og agendaer. En annen hadde ansvaret

for møtereferater, mens en tok seg av risikoplaner og risikohåndtering. Gruppeleder skulle fungere som kontaktperson mellom gruppa og PLU og veileder. Man hadde faste møter med veileder for prosjektfaget, og kunden (PLU).

Gruppa satset på en løs kommunikasjonsmodell og en del solojobbing. Siden det var noenlunde flat struktur, kunne alle ta i et tak der det var behov. Man betegnet det som frihet under ansvar. Dette var en følge av gruppas spredning både geografisk og fagmessig. Bl.a. var en gruppedeltaker ofte på jobb i Nord Trøndelag. Man fikk på plass en del faste rutiner for kommunikasjon og dokumentasjonshåndtering. Email skulle brukes som hovedkommunikasjonsmiddel. En mailingliste ble opprettet for å enklere kunne sende informasjon til hele gruppa. Man brukte også MSN til direkte, synkron kommunikasjon. På den måten kunne man holde møter når det ikke var mulig å samles fysisk. Telefon ble også brukt som et hurtig kommunikasjonsmiddel. Man hadde et fast møtetidspunkt fredager kl. 14.00, der alle forsøkte å samles fysisk.

3.2.4.3 Utviklingsmetode

Studentgruppa beskrev sin utviklingsmetode som delvis iterativ med agile programmering⁵ som inspirasjon. Dette besto i følge dem av å først gjøre unna all prosjektering og planlegging i første del av prosjektet, før de gikk i gang med hurtige iterasjoner i programmeringsdelen. Modultester ble gjort underveis, mens brukertester og akseptansetester skulle gjøres til slutt. Begrunnelsen for dette valget var at det forenklet systemtestingen [16].

3.2.4.4 Arbeidsmål og krav

PLUs innledende krav til systemet ble presentert i oppgaveteksten for prosjektet i faget IT2901 Informatikk Prosjektarbeid 2 (se kap. 3.2.1). Fagledelsen krevde at produktet skulle være ferdig 10. mai 2007. Endelig frist for dokumentasjon og rapport ble satt til 25. mai [16]. Siden PLUs prosjekt er omfattende og går over flere år, skulle bare en begrenset del av systemet utvikles av studentgruppa, som en start på det hele, og som et utgangspunkt i noe man trengte, som et intervjuobjekt nevner:

"Man tok utgangspunkt i noe man trengte, nemlig et nettsted for kompetanseheving av praksisveiledere."

Det var ønskelig at systemet kunne tas i bruk på FoU-konferansen 2007, som ble holdt 15. mai,

⁵ The Agile Alliance - <http://www.agilealliance.org/>

men systemet hadde ikke blitt nok utviklet til å kunne tas i bruk da konferansen skulle holdes.

3.2.4.5 Dokumentasjon

Gruppen opprettet en side i Microsoft SharePoint der man la ut alle dokumentene man produserte. Denne hadde også veileder i faget, og PLU tilgang til. Alle nye dokumenter skulle i utgangspunktet opp på SharePoint-siden. Der skulle alle gruppe-medlemmene kommentere det som var gjort gjennom mail eller lignende (MSN). Forslag til endringer og saklig kritikk var en viktig del av prosessen før et ferdig dokument kunne godkjennes av dokumentansvarlig på gruppa. Man hadde en løs struktur, men det krevdes av alle at man var oppmerksomme på nye dokumenter, og leste dem. Source Gear Vault (Source Gear LLC) fungerte på samme måte for håndtering av kildekode (revisjonskontroll). Alle endringer skulle kommenteres og godkjennes av programmeringsansvarlig.

Gruppen laget en prosjektrapport som var av interesse for eventuelle framtidige brukere av prototypen, og eventuelle videreutviklere som trengte mer teknisk informasjon. Rapporten summerte opp bakgrunnen for prosjektet, prosjektplanlegging med risikoestimering, kravspesifikasjon, konstruksjon, implementering, testdokumentasjon, teknisk vurdering, brukermanual, administrasjonsmanual, evaluering av prosjektet, og vedlegg med bl.a. testdokumentasjon og diagrammer.

3.2.4.6 Utfordringer

Det var dessverre ikke alltid at nye og oppdaterte dokumenter ble lagt ut på SharePoint-sida når de skulle. Dette skapte noe forsinkelse i min egen overvåking av gruppas utviklingsarbeid. For eksempel var statusrapporter etterlengtet. I tillegg var PLU oppmerksom på risikoen ved å legge prosjektet sitt ut som et studentprosjekt. Samarbeidet med gruppa ble tatt opp i et intervju med en PLU-ansatt, som fortalte følgende:

"Man hadde ikke noen voldsomme forventninger. Det hadde ikke vært gjort noe lignende før. Man visste dette medførte risikopunkter. Overholdelse av frister fra studentgruppas side ble litt slapt etter hvert. Kanskje ble de gitt for mye frihet, og kanskje var viljen større enn evnen. Gruppen selv fikk også samarbeidsproblemer internt, noe som fikk konsekvenser."

3.2.4.7 Teknisk plattform

Gruppen vurderte i hovedsak to former for teknologisk basis for systemet. Disse var hvorvidt

systemet skulle ha åpen eller lukket kildekode. Med et system basert på åpen kildekode mente gruppa at man ville ha full åpenhet rundt kildekoden, og kunne redigere koden fritt for å tilpasse funksjonalitet og utseende etter eget ønske, men gruppa mente at slike systemer hadde den ulempen at de var dårlig dokumenterte, noe som ville gjøre det vanskelig for etterfølgende utviklere. Systemer som baserte seg på lukket kildekode, mente gruppa besto for det meste av “out of the box” løsninger, som var mer brukervennlige i sin utforming med tanke på utseende. Disse “ferdige” pakkene kunne man bruke til å lage nye produkter av uten å vite hvordan et program eller en funksjon fungerte.

Videre argumenterte gruppa for å lagre data på serveren istedenfor lokalt på brukernes datamaskiner, og gjøre funksjonalitet tilgjengelig for brukerne gjennom webtjenester. Man pekte på Google, Xcerion og Microsoft som forbilder, og tok fram som eksempler tjenester som Google Docs og Spreadsheet og Microsoft live. Man argumenterte for at brukerne da slapp å installere noe lokalt, ta backup lokalt eller laste ned nye oppdateringer, samt at man slapp unødvendige kostnader på lisenser på programvare.

Man så også på mobilitet som en viktig faktor; at systemet kunne brukes på håndholdte enheter med internettaksess. Man falt derfor til slutt på Microsofts .NET teknologi, siden denne skulle gi god fleksibilitet i forhold til bl.a. rettighetsbasert tilgang og var lett å lage modulær.

Applikasjonen styrte mot en 3-lagsarkitektur som besto av henholdsvis datatilgangslaget i bunn, forretningslogikklaget oppå der, og presentasjonslaget på toppen. Lagene ble organisert i en 3-tier struktur der lagene er synlige kun fra toppen og ned. Dvs. at et ovenforliggende lag kan hente data fra et underforliggende lag, men ikke omvendt. Dette innebar at presentasjonslaget kommuniserte med henholdsvis forretningslogikklaget og datatilgangslaget etter behov. Forretningslogikklaget kommuniserte videre med datatilgangslaget, og datatilgangslaget kommuniserte med databasene.

Kommunikasjon skulle være i generiske object collections, i form av for eksempel `List<Type T>`. Dette skulle gjøre at man fikk en ren objektorientert kommunikasjonsform. GUI ville være adskilt fra de andre lagene, og kommunikasjonen gå gjennom en `WebService`. Dette innebar bruk av XML/SOAP (Simple Object Access Protocol), og medførte at selve websiden ikke trengte å ligge på samme fysiske webtjener som resten. Dette skulle imøtekomme framtidig skalerbarhet, og derfor gjøre det lettere for framtidig utvikling ved at man kunne dele applikasjonen i flere moduler og da videreutvikle disse uten at det ville påvirke den allerede fungerende løsningen [16].

Selve implementasjonen ble gjort i rammeverket Microsoft .NET 2.0. Dette fordi man mente at

denne teknologien var framtidsrettet og tidsbesparende (gjennom dra- og slippmetoder). Brukerhåndteringen ble gjort gjennom ASP.NET MembershipProvider, en funksjonalitet som skulle fungere out-of-the-box. Databasen som ble brukt var Microsoft SQL Server 2005. Gruppen mente denne var en ekstremt kjapp og robust databaseserver som lå foran konkurrenter hva skalerbarhet, stabilitet og sikkerhet angikk. Databasen ble bygd på normaliseringsreglene, og er for øyeblikket på 3NF.

3.2.4.8 Resultater

Grappa vektla at PLUs prosjekt med denne websida skulle gå over flere år, og at det sannsynligvis ville bli et mye større system senere, med mange flere funksjoner. Grappa valgte derfor å fokusere mye på de underliggende funksjonene og mindre på GUI. De prøvde å legge grunnen for et system som kunne videreutvikles med mange funksjoner senere. Dette resulterte i at man ikke fikk et helt fungerende system (etter PLUs ønske) opp i tide. PLU hadde ønsket seg et veldig enkelt system som gjorde mindre, men som fungerte (kap 9.1 i [16]). Dette, pluss faktorer som for eksempel arbeidsfordeling på grappa, resulterte i at systemet ikke ble ferdig til tidsfristen PLU hadde ønsket seg en prototype av systemet ferdig (15.mai). I stedet greide grappa å få til en kjørende versjon til 25. mai (kap. 2.6.1 i [16]).

Grappa mente at videreutviklere ikke trenger å bruke den teknologien dem selv har brukt (kap. 9.1.4 i [16]). Videre mener de at systemet er konstruert slik at de forskjellige lagene i applikasjonene kommuniserer med hverandre på en måte som gjør at ting kan byttes ut etter hvert som prosjektet skrider frem. Produktet skal være fleksibelt. Grappa mener at dokumentasjonen skal være tilpasset for, og gjøre det lett for videre utvikling, og at GUI er noe som er viktig å se på for videreutviklere. De delene av applikasjonen som har med datalogikk og bedriftslogikk å gjøre, er grundig testet, og skal etter sigende oppføre seg omtrent plettfritt, og holde en høy standard. En del av GUIet skal fungere bra. For eksempel har visning av forskjellig materiale fungert, men man har ikke fått til innlegging av informasjon fra webgrensesnittet. Grappa løste dette ved i stedet å lage et selvstendig program for opplastning, som bare skal være en midlertidig løsning. Dette må altså løses av etterfølgende utviklere om systemet skal oppfylle PLUs krav til det. Grappa mener at neste del av websideprosjektet bør fokusere på implementering og testing. Videre mener man at bare småting gjenstår før man har et fikst ferdig produkt. Også testing av brukernes oppfattelse av systemet må gjøres, siden grappa ikke fikk gjort dette [16].

3.2.4.9 Test av løsningen

Forfatter av denne rapporten gjennomførte test av prototypen som studentgruppa hadde laget. Testene ble utført i nettleserne Firefox og IE (7) på OS'ene Microsoft Windows Vista og Windows XP. Her ble det testet hva slags funksjonalitet som var blitt implementert, og hvordan denne eventuelt tilfredstilte kravspesifikasjonen. Mangler i funksjonaliteten etter kravspesifikasjonen ble notert. Brukbarhet og grafisk grensesnitt ble også testet i lys av kravspesifikasjonen, i tillegg til ytelse. Det bemerkes at her testes kun etter kravspesifikasjonen til systemet, slik den var definert av studentgruppa og PLU ved oppstart av denne oppgaven. En evaluering av denne kravspesifikasjonen, og forslag til endringer på denne, er diskutert i kap. 4.1. Under er summert opp resultatet av testen. Et detaljert testsammendrag er *ikke* vedlagt rapporten, da dette har blitt langt. En større integrasjons- eller systemtest ble ikke foretatt.

Det er vektlagt en ganske kritisk innfallsvinkel på testene, slik at det som ikke er bra kommer ganske tydelig fram. En totalevaluering av prototypen (inkludert forutsetninger og utviklingspotensiale) er foretatt under kapittel 7.0 (Diskusjon).

Ikke-funksjonelle krav

Det grafiske grensesnittet er ikke dekorerende, men framhever portalens viktigste funksjoner av det som er implementert etter 1. fase (navigasjon mellom temaer og filer og videoer assosiert med temaene). Dvs. at det ikke er brukt dekorerende grafikk som overdøver navigasjonen mellom temaer og arkiv. Farge på tekst og bakgrunnsfarge gir god kontrast for lesning (sort og blå skrift mot hvit og lyseblå bakgrunn). Menyenes framtrepende plassering på venstre side er lett å få øye på siden brukere oftest starter med å se på venstre side (i Norge er vi som kjent vant til å lese fra venstre til høyre). Fontstørrelsen er passelig for lesing og klikking. En brødsmulesti som holder oversikten over hvor man befinner seg på nettstedet (sekvensiell oversikt over navigasjonsstrukturen) er plassert strategisk øverst til venstre under logoen. Terminologien på nettsidene er ikke avansert, noe som passer for noviser og middels datakyndige brukere.



Figur 1: Framsida av studentgruppas løsning.

Framsida (figur 1) skiller seg ganske klart fra de underliggende sidene, med sitt eget innhold som kun kan aksesseres ved å trykke på "Hovedsiden" i brødsmulestien eller ved å taste inn URL til portalen. Der vises ingen menyer for fil- og videoressurser eller forum, og den skiller seg derfor ut fra de underliggende temaside. Innholdet på framsida er umiddelbart tilgjengelig.

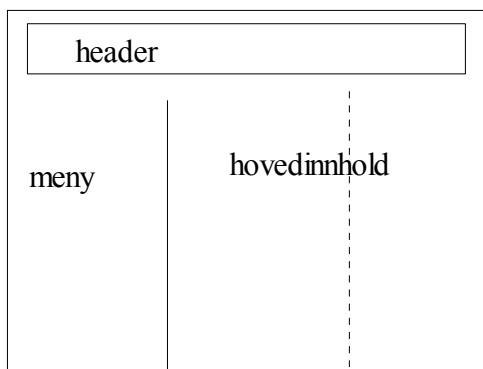
Websidene har ingen spesiell logo som identifiserer at websidene er del av plu-veven. Det burde lages en egen logo som identifiserer websidene som plu-veven, spesielt om prosjektet skal vokse og skille seg ut. Selv om NTNUs retningslinjer for webdesign [17] forbyr å legge egen logo i headeren så lenge PLU-veven forblir under NTNUs nettsider, kan man legge den under headeren på framsida.

Lite eller ingen grafikk er brukt i brukergrensesnittet. Dette gjør det ryddig, men kan samtidig skape mangel på identitet og gjøre sidene mindre attraktive. Illustrasjoner kan dessuten forbedre navigasjon.

For layout så er det brukt fast layout. De mest kritiske elementene på framsida er synlige over sidas fold (synlig uten vertikal rulling) (se figur 1). Generelt for nettstedet er menyene i venstre marg gruppert i logiske grupper. Menyalternativene er uniforme i font, farge og størrelse.

Navigasjonsstrukturen opptar under 20 % av plassen på websidene, noe som er et krav for god layout i følge [18]. Dog vil denne andelen stige til nærmere 1/3 hvis man ruller siden nedover (for

lange sider), fordi headeren da forsvinner og horisontalt opptar da menyene 1/3 av siden. Hovedinnholdet på sida er alltid i en fast blokk som opptar midtre og høyre kolonne (høyre kolonne går i ett med midtre kolonne), som vist i figuren under.



Man ser av figuren at man kan dele skjermen opp i 3 kolonner; venstre, midtre og høyre kolonne. I venstre kolonne er navigasjonsmenyen. I midtre og høyre kolonne presenteres sidas hovedinnhold. Dette er ikke anbefalt i følge NTNUs retningslinjer [17] for design av websider under NTNUs eksterne websider.

Innholdet i portalen er først og fremst gruppert i temaer, dvs. at du må velge et tema for å komme til filer og andre ressurser. Dette kan ses i venstre kolonne i (se figur 1). Når man laster opp ressurser fra adminverktøyet må man også velge under hvilket tema man skal legge det. Innholdet er lett og kvikt tilgjengelig for brukere (ikke mer enn 3 klikk ned til substansielt innhold). Det er opp til innholdsadministratorer å ikke duplisere informasjon som postes på websidene. Fra portalens side så presenteres informasjon bra uten å duplisere innholdet, om det ikke var administrators hensikt.

Det er ingen duplisering av lenker eller menyalternativer. Menyoverskriftene skiller seg godt fra fontene på tekst og lenker ved at de er litt større og fetere. Lenkefontene (menyalternativer og forumposter) har alle samme font og størrelse. God konsistens. Ingen lenker har fått generiske navn som f.eks. "klikk her...".

Det er ikke enkelt ut fra brødsmulestien å forstå hvilke "smuler" som er et tema eller en annen side, men slik det er nå, vil alltid temaet være "smule" nr. 2 fra venstre. Dette vil ikke nødvendigvis være riktig ved eventuelle utvidelser av systemet der man kanskje kan ha undertemaer og andre sider.

Portalen unngår *ikke* horisontal rulling ved standard oppløsning (1024 x 768), men det er ikke mye

om å gjøre. I praksis er det mer et forstyrrende element. Det burde dog unngås helt, slik at “rullestangen” ble borte.

Dato og tid er vist for postinger i forum og elementer i fil- og videoarkivene, men det er ingen oversiktlig måte å se siste oppdatering på webportalen. Tall er brukt som datoformat, noe som ikke var ønskelig i følge kravspesifikasjonen.

Det går ikke helt klart fram av navigasjonen eller layouten at hvert tema har sitt forum. Man kan lett tro at det er kun ett felles forum. Det er heller ingen indikasjoner på hvilket foruminnlegg man til enhver tid ser på, verken i menyen til venstre eller andre steder. Forumverktøyet er likevel enkelt og lettforståelig hva gjelder posting og gjennomlesning.

Nettsidene bruker akseptabel tid på å lastes inn.

Funksjonelle krav

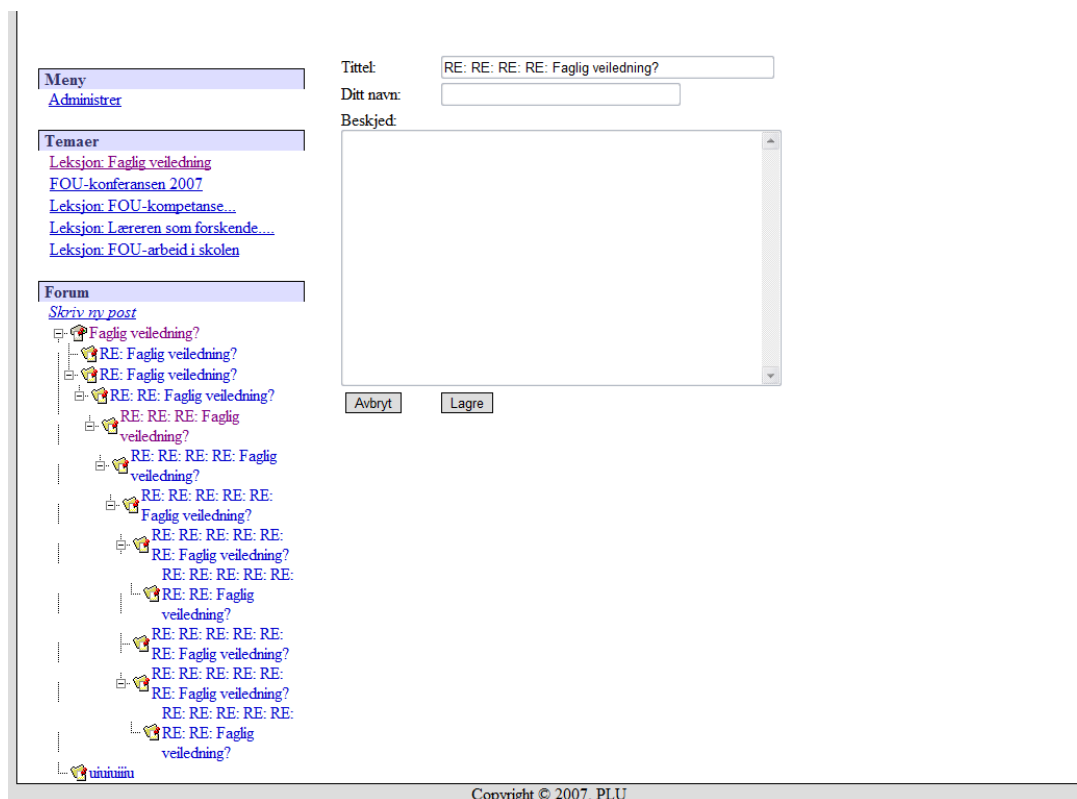
Brukermanual er ikke tilgjengelig på siden. Dette krever at sidene er så logiske og lettforståelige at brukerne ikke trenger en brukerveiledning. Dette er ikke tilfelle, fordi navigasjonen ikke er helt selvforklarende (se ikke-funksjonelle krav).

PowerPoint-filer, dokumenter, podcasts kan lastes direkte opp i løsningen via administrasjonsverktøyet (figur 3). Videostreaming ser ikke ut til å støttes (kun i eksternt program), men man kan laste opp videofiler og laste dem ned fra løsningen og spille dem av offline. Man kan laste ned lydfiler fra løsningen, eller spille av direkte i et eksternt lydprogram. Man kan ikke spille av direkte fra løsningen.

Alle som skal laste opp ressurser til portalen må ha administratortilgang til portalen. Man logger inn med sitt brukernavn og passord.

Det er ikke implementert søk på portalens innhold, men det skal være lagt et grunnlag for denne funksjonen i koden, ifølge studentgruppa.

Forum er implementert og fungerer bra som et enkelt forum med basisegenskaper som posting av nye innlegg og svar på innlegg (figur 2). Man kan kun skrive ren tekst, uten formateringer. Å sette fora til å være begrenset fungerer ikke - om man setter dem til begrenset fra administrasjonsverktøyet så forsvinner de fra GUI'et.



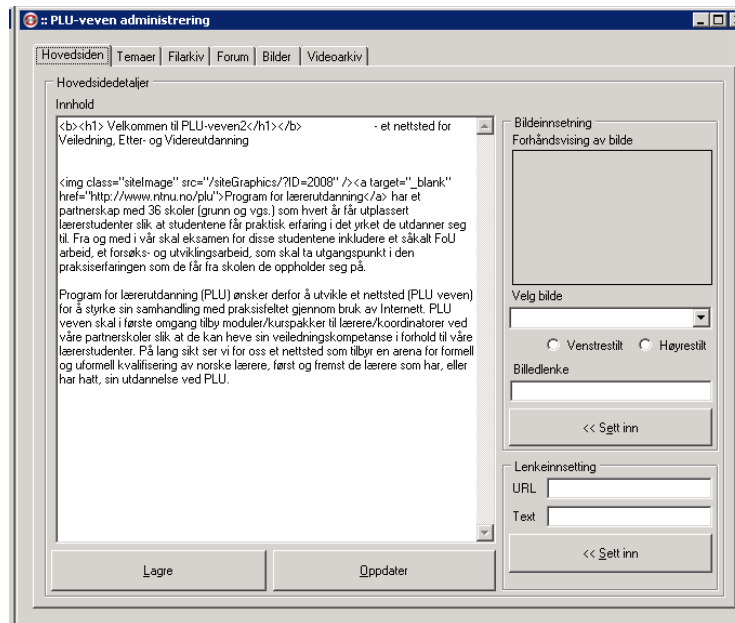
Figur 2: Svar på forumpost i studentgruppas løsning.

Chat-rom, Wiki, RSS feeds og søk i portalen er ikke implementert. Noe av dette ser ut til å være påbegynt i koden.

Åpne ressurser fungerer knirkefritt. Om man derimot begrenser et tema med tilhørende ressurser, så skjules dette helt i GUI'et.

Det er lagt til rette for tagging av filer og videoer med metadata. Dette gjøres i administrasjonsverktøyet.

Administrasjonsverktøyet



Figur 3: Administrasjonsverktøyet for studentgruppas løsning.

Figur 3 viser administrasjonsverktøyet ved oppstart. For å bruke administrasjonsverktøyet må man være tilkoblet NTNUs nett enten direkte eller via VPN. Administrasjonsverktøyet har grei brukervennlighet (mer utdypet lenger ned), men kan være litt komplisert for noviser. Verktøyet har heller ingen spesiell stil eller profil som knytter det naturlig opp mot plu-veven og websidene selv. Verktøyets funksjonalitet burde erstattes med et webgrensesnitt, som ønskelig ut fra kravspesifikasjonen.

Det er ingen hjelpeknapp tilgjengelig fra administrasjonsverktøyet, ei heller noen knapp eller kontaktinformasjon for eventuelle tilbakemeldinger eller spørsmål. Dette kunne vært på sin plass, hvis prototypen var ment å skulle tas i bruk, og ettersom verktøyet har mange forskjellige funksjoner og er i startfasen.

Verktøyets layout er designet med godt logisk grupperte kontroller, tekstfelter og knapper som ligger under sine tilhørende "tema" eller funksjonalitet. Temaene igjen er gruppert under sine adskilte bokser. Det er noe ulogisk fanerekkefølge der faner for "bilder" og "forum" kommer mellom fanene for "filarkiv" og "videoarkiv". Fil- og videoarkivene er dessuten underliggende temaside, så det virker ulogisk at både "tema", "filarkiv" og "videoarkiv" er representert som faner på lik linje med hverandre (se figur 3).

Verktøyet har ingen redundant informasjon, men det som kanskje kan forbedres er at man kunne

sammenføyd knappene for "Legg til" under bildeinnsetting og lenkeinnsetting til én felles knapp for å legge til.

Det kan ta svært lang tid å starte administrasjonsverktøyet om man sitter over VPN og har mindre rask internettoppkobling. Ingen statusindikatorer på hvor lang tid som gjenstår av innlastning før verktøyet kommer opp på skjermen.

3.2.4.10 Konklusjon

Studentgruppas løsning har et enkelt og rent design, men mangler noe på brukbarhet hva angår lettforståelig navigasjon, lett tilgjengelige hjelpefunksjoner og grafikk og layout. Løsningen har oppfylt kravet om tematisert gruppering av informasjon og har god konsistens i tekst, fontbruk og lesekontrast. Innhold er lett tilgjengelig.

Funksjoner implementert er skriving av artikler, inn- og utlogging, forum og filopplastning og -nedlastning. Bilder og lenker kan settes inn i teksten på sidene, men man må bruke HTML-koder i teksten for å formatere den.

NTNUs retningslinjer for design av websider under NTNU påpeker at en gridmal med kun én spalte ikke er å anbefale for presentasjon av tekst, men fungerer best for tabeller. Dette kunne derfor blitt forbedret i eventuelle reviderte utgaver av studentgruppas produkt.

For en diskusjon rundt relevansen av studentgruppas løsning se kapittel 7.2.1. For en konklusjon på relevansen av løsningen i forhold til PLU-veven, se kap. 8.1.

3.3 Utdanning og nettbaserte verktøy

I utdanningssammenheng er ofte Learning Management Systemer (LMS) brukt. Systemer som er kapable til å levere kurs på nett blir også ofte brukt i utdanningssammenheng. Disse blir ofte kalt Course Management Systems, som oversatt til norsk betyr "kursleveringssystemer". LMS og Course Management Systems er systemer som ligner på et CMS (se kap. 1.3), og kan være nettbaserte. Eksempler på nettbaserte kursleveringssystemer og LMS er Moodle og It's Learning. I norsk grunnskole benyttes mest Fronter og It's Learning, mens PedIT og Moodle brukes noe mindre

[19].

Fronter flagger seg selv som en “plattform for nettbasert læring og samarbeid, utviklet spesifikt for utdanningssektoren” (www.fronter.no). Fronter fokuserer på å være et hjelpemiddel i organisering av læringsaktiviteter, informasjon, produksjon og bruk av elektroniske læremidler. Ifølge [20] er Fronter et nettbasert læringssystem og konferansesystem, der læreren kan lage en virtuell bygning som er delt inn i rom der lærere og elever kan samarbeide i forskjellige fag ved hjelp av forskjellige verktøy. Fronter er et administrativt system som lar læreren åpne og låse rom eller mapper der elever leverer inn sine oppgaver. Vanlige verktøy i Fronter er meldinger, nyheter, epost, forum, kalender og delte dokumenter. Et testgenererings-verktøy gjør at man selv kan lage e-læringskurs, øvinger og prøver som kan evalueres digitalt. Elever kan få hele sin læringsbiografi logget i Fronter. For en ekstra kostnad kan verktøy som føring av fravær, karakterer og orden, digital eksamen, designverktøy, formeeditor og plagiatkontroll m.m. legges til systemet.

Moodle tilbyr verktøy for å lage online kurs som kan inneholde moduler som nettpat, fleralternativ-spørsmål, kunnskapsbase, forum, glossar, spørrekonkurranse, LAMS (visuelt miljø for å lage sekvenser av læringsaktiviteter) osv. Systemet støtter bl.a. SCORM-standarden (se kap. 1.3 for en definisjon av SCORM).

It's Learning er mye brukt innenfor NTNU. Se kap. 3.3.1 og 5.1.3 for en beskrivelse av It's Learning som nettbasert verktøy, samt kap. 7.1.2 for en diskusjon av relevansen av systemet for PLU-veven.

For at et LMS skal dekke behovet for e-læring, bør systemet være dynamisk (dvs. aktivt, fleksibelt og tilpasningsegnet). Det bør gi tilgang til folk og læringsressurser, og tilby interaksjon og samarbeid 24 timer i døgnet [21]. [21] vektlegger også at systemet bør oppfylle de fire viktige sidene i e-læring, som han definerer som (1) Fagkrets-utvikling, (2) Klasse-, kurs- og innholdsadministrering og -leveranse, (3) Kompetansevurdering og (4) Samarbeid. Disse burde utføres effektivt i et LMS ved at man designer effektive og lettbrukelige verktøy for læreobjektene, administratorene og instruktørene. De tekniske egenskapene for et LMS burde etterkomme kravene for nødvendig utstyr, sikkerhet og budsjettbegrensninger.

3.3.1 Innenfor NTNU

It's Learning er mye brukt innenfor NTNU. It's Learning tilbyr emneoversikt for studenter som er

semesterregistrert ved universitetet. Alle emner studenten tar det semesteret har sitt eget område på It's Learning, der det kan legges ut bl.a. oppslag, oppkommende aktiviteter (kalender) og pekere for emnet. Man kan også få en oversikt over registrerte deltagere til emnet. Prosjektområder kan opprettes uavhengig av fag, og kan administreres av studenter. Andre studenter kan inviteres til å delta i prosjektet, og rettigheter og roller til deltagerne kan settes. Systemet tilbyr i tillegg et meldingssystem mellom alle studenter som bruker It's Learning, og en kontaktliste. Hver student har sin egen eportefølje der man kan presentere seg selv med bilde, lage sin egen blogg og laste opp filer til et område som er åpent for andre. Lærer kan legge ut felles aktiviteter i kalenderen for et emne og legge ut elementer for et emne. Lærere har i tillegg verktøy som status og oppfølging (rapporter om framgang i et emne), gruppeinndeling av studenter i et emne, faginnstillinger og nettmøte. Innholds- og interaksjonsbaserte elementer i emner kan også legges til av lærere. Disse legges inn i en tremenystruktur for faget, og inkluderer mapper, filer, notater, pekere og bilde med beskrivelse, diskusjoner, konferanser, oppgaver, dokumenter, tester og forklaringssekvenser og lignende⁶. Se kap. 5.1.3 for en fyldigere vurdering av It's Learning.

Et CMS som brukes ved NTNU er Escenic⁷. Dette brukes for å lage mange av NTNUs offisielle nettsider, inkludert nettsidene til Program for Lærerutdanning. Systemet blir ikke undersøkt utfyllende i denne rapporten, men det er et kraftfullt kommersielt verktøy som kjører i et Java-miljø, og som er noe rettet mot nettbaserte aviser, nyheter og journalistikk (www.escenic.com).

Moodle er også brukt ved NTNU. Det finnes sikkert flere CMS og LMS i bruk ved NTNU, men det er ikke denne oppgavens hensikt å få en oversikt over alle slike systemer ved NTNU.

4 Kravspesifikasjon og teori

I dette kapitlet gjennomgås PLUs kravspesifikasjon for plu-veven, som definert sammen med studentgruppa i faget IT2901. Teori rundt nettbaserte ressurser blir presentert og gjennomgått. Uten et viktig teoretisk grunnlag kan man gå i blinde ved valg av nettbaserte verktøy.

⁶ Brukerveiledning i It's Learning for lærere ved NTNU

⁷ www.escenic.com

4.1 PLUs kravspesifikasjon

Dette er utdrag fra kravspesifikasjonen til plu-veven, som ble utviklet av studentgruppa sammen med PLU. For en detaljert kravspesifikasjon, se [16]. Lenger ned vil en evaluering av og forslag til endringer i kravspesifikasjonen gis.

4.1.1 Ikke-funksjonelle krav

Om ikke-funksjonelle krav kan kravspesifikasjonen i [16] oppsummeres som følger:

Sidens viktigste funksjoner må fremheves, slik at brukerne har en klar indikasjon om hvor de skal navigere. Disse funksjonene bør ha en fremtredende plassering, eks øverst og sentrert, og de bør ikke ha mange visuelle konkurrenter. Plassér navigasjonsfunksjonalitet i et område brukerne lett legger merke til. Unngå multiple navigasjonsstrukturer for samme type lenker. Det skjermbilde som til enhver tid er synlig bør inneholde en sti som orienterer brukeren om hvor han/hun befinner seg.

Informasjon må grupperes i logiske grupper, og plasseres i et adskilt område. Redundant informasjon burde unngås. Startsidene skal designes slik at den skiller seg nok ut fra de underliggende sidene til å bli gjenkjent som startside. Bruk animasjon på startsidene så lite som mulig. Løsningen bør ikke ha mer enn 3 klikk ned til substansielt innhold. Bruk layoutstandarder konsistent, og bruk flytende layout. Horisontal rulling på standard oppløsning (1024x768) bør unngås.

Uthev og presisér lenker. Gi dem informative navn, og vektlegg særlig det første ordet, siden brukere ofte skanner det første eller de to første ordene i lenker for å skille dem fra hverandre. Ikke gi lenker generiske navn. La lenkene skifte farge ved besøk (for hver innlogging).

Kunstig terminologi for kategorisering av navigeringsvalg vil forvirre brukerne. Ord som webside, hjemmeside, intranett etc. bør ikke forekomme på siden.

Grafikk brukes for å fremheve innhold, ikke kun for å dekorere. Grafikk eller bilder bør gis merkelapper dersom deres funksjon ikke er intuitiv. Løsningen bør prioritere en tiltalende profil. Akkompagnér søkeboksen med en søkeknapp. Kategorier og lister unngås dersom de har kun ett element.

Sidene bør ha et begrenset sett tekstformateringer, og teksten bør stå i skarp kontrast til

bakgrunnen.

4.1.2 Funksjonelle krav

Om funksjonelle krav kan kravspesifikasjonen i [16] oppsummeres som følger:

Løsningen bør følges av en lettforståelig brukerveiledning som er tilgjengelig på siden, og en lenke som viser til nøkkelpersoner med epostadresser vedrørende løsningen. Tilby en søkeboks på startsidene, som er minst 25 tegn lang, i tillegg til en lenke til avansert søk. Søk skal søke i hele nettstedet. Inkluder en tilbakemeldingslenke, og presiser hvem det leses av. Siste oppdatering av siden skal være synlig.

PowerPoint, Artikler og dokumenter, studentprosjekter og podcasting skal være tilgjengelig på nettstedet, via PLUs egen server. Videostreaming skal være tilgjengelig via lenker til Multimediesenterets server. Ressurser skal kunne lastes opp av brukere tilknyttet PLU og fagpersoner ved partnerskolene. Både åpne og lukkede innholdsressurser skal samles i moduler/temapakker (lukkede ressurser vil kreve innlogging), men åpne ressurser skal samtidig kunne tematiseres på alternative måter. Diskusjonsfora (både lukkede og åpne), chatrom (åpent for alle), wiki og rss feeds skal implementeres (noe av dette implementeres i senere faser).

4.1.3 Evaluering av kravspesifikasjonen

Det var tidlig klart at PLU var usikre på hvilket system de faktisk ville ha. Kravspesifikasjonen reflekterer dette, siden den ikke går for detaljert inn på hvordan systemet skal være, spesielt i et langsiktig perspektiv. Dette trenger ikke være så galt for et utviklingsprosjekt, siden det ikke er hensiktsmessig å definere alle kravene i startfasen [22]. Ofte må en prototype til, som prøves ut av brukere. Brukertestene gir tilbakemeldinger som igjen utvider og detaljerer kravspesifikasjonen. Mer om hvordan prosjektets utviklingsprosess bør skisseres opp står i kap. 6.2. Vi kan betrakte den konkrete kravspesifikasjonen som er forfattet som å være et resultat av innledende planlegging og modellering, og noen få programmeringsiterasjoner med modultesting (se kap. 3.2.4). At studentgruppa ikke kunne brukerteste systemet så mye i løpet av prosjektiden henger noe sammen med at de potensielle brukerne av systemet var veldig spredt. Sett i lys av dette har kanskje ikke kravspesifikasjonen utviklet seg helt på ønskelig måte, dvs. drevet ut fra de best treffende brukerbehov. En driftsutgave av plu-veven ble ikke satt i drift før ved årsskiftet 2007/2008, så tiden har for min del vært knapp, og brukertestene har ikke blitt utført (kun observasjon av en sentral

administratorbruker). Bare noe initiell undersøkning om bruk og ønsker for plu-veven er foretatt via spørreskjemaer og intervjuer. Det er derfor hensiktsmessig at eventuelle videreutviklere av plu-veven foretar grundig brukertesting.

Tilbakemeldingene på spørreskjemaene, samt intervjuer med personer ved PLU, og undersøkelser av eksisterende systemer og teori rundt nettbaserte ressurser har gitt et visst grunnlag for å detaljere kravspesifikasjonen mer. Noe interessant er det at i spørreskjema om hvorvidt de allerede eksisterende nettbaserte tjenester i bruk i arbeidet til bl.a. koordinatorene er dekkende for deres behov, så svarte de fleste at de var allerede dekkende. Likevel, basert på en annen spørreundersøkelse om hva brukernes hensikt med plu-veven ville være gitt at de tok den i bruk, kom bl.a. disse svarene fram:

“Informasjon og oppdatering/videreutvikling”

“...publisere materiale”

“...skaffe relevant informasjon på en lettvinnt måte”

“Vet for lite om plu-veven til å kunne gi svar”

“Tips og idéer...”

“Finne gode artikler og undervisningsopplegg”

“Til etterutdanning. Bruk av faglig diskusjonsforum”

“Hente ut informasjon og linker til faglitteratur og fagstoff”

“Finne dokumenter og holde orden på alle sider av min jobb. Kanskje kan jeg finne alt på ett sted?”

“Diskusjon”

På spørsmål om hvordan man ønsket at nettstedet skulle hjelpe en å oppnå ens hensikt, var noen av svarene som følger:

“Oversiktlige sider med tydelig merking. Evt. lenker må virke. Informasjon og annet ordnes i skuffer som er entydig merket. Kalender over det som skjer. Kommunikasjonsforum. Alt må være lettlest i svart på lys, ensfarget bakgrunn...”

“Jeg trenger tilgang til Studio 10 eller 11, skanner, video, mikrofon. Gjerne får plu-veven være multimodal med mulighet for bilde- og lydspor, blogg, diskusjonsforum og lenkesamling.”

“Jeg vil vite om nye verktøy og idéer til praktisk bruk”

“Blogg, ressurser og diskusjoner”

“It’s learning fungerer bra. Diskusjonsforumet er bra.”

“Skjermbildet må være enkelt, oversiktlig og informativt og lett tilgjengelig.”

“WYSIWYG - jeg ønsker at alt skal være logisk og selyforklarende.”

“Skal dette virke, må inngangsterskelen være lav. Behøver man passord f.eks.? Diskusjonsforum eller blogg vil være bra.”

“Tja, si det!”

På spørsmål om hvilke kriterier man satte til systemet for at det skulle gjøre en tilfreds, var noen av svarene som følger:

“Lett navigasjon og god lesbarhet”

“Det skal være lett å komme inn. Det skal ikke bare være gammel info. Det skal være en bevegelse, slik at det blir en oppdatert dialog.”

“Oversiktlig, strukturert og lett vint”

“Mac-vennlig og open source.”

“At det er enkelt å administrere og at det legges ut interessante tilbud og artikler m.m.”

“Har tilgjengelighet til Orakeltjenesten”

“Brukervennlig, ryddig og lett å få tak i det du er på jakt etter.”

“Oversiktlig med tema i venstre marg og gjerne vindusbasert med rullegardiner med mer informasjon.”

“Det må være enkelt å komme inn. Når inngangsterskelen er høy, blir det til at man ikke gidder. Samtidig må vi ha mange brukere. Skal man make dette, må stedet eksempelvis ha en informasjonsfunksjon slik at man må inn for å følge med. Det må ikke bli frivillig å være der.”

“Oversiktlig navigasjon og stadig oppdatering av innhold som er aktuelt for målgruppen”

“Lett tilgjengelig. Det må ikke forutsette at du er tilgjengelig eller på nett hele tiden. Bestemte tidspunkt der det er spesiell aktivitet og brukerne er tilgjengelige?”

“Nettstedet må være logisk og enkelt oppbygd”

Et svar nevnte *kalender*, og dette kan være en interessant funksjon. Kanskje kan dette være en sentralt organisert kalender med viktige begivenheter, tillegg til at brukere kan føye på sine egne

avtaler i tillegg? Kanskje kan den holde orden på bestemte tidspunkt da veven oppdateres eller nettmøter eller diskusjoner foregår? Når det gjelder innhold, bør det finnes så viktig innhold at folk går inn i nettstedet for å “følge med”, og det bør være oppdatert og ikke utdatert. Flere svar nevner viktigheten av hvor enkelt det er å komme inn i portalen. Jo enklere, desto flere brukere tar seg bryet med å bruke portalen. Et krav her bør være at man bare behøver å skrive inn *ett* passord, og at man slipper å skrive inn dette igjen før man har logget ut eller eventuelt blitt tidsbestemt logget av. Verktøyet bør være enkelt å administrere. Dette betyr at den siden av portalen som har med administrasjon å gjøre, ikke skal unntas fra kravet om enkelhet og brukbarhet.

Det bør også settes krav til at det digitale verktøyet skal fungere tilfredsstillende i de viktigste nettleserne og plattformer som brukerne bruker.

Det ble foretatt intervjuer av sentrale personer ved PLU, og noen sitater fra disse angående forventet eller ønskelig funksjonalitet i plu-veven er som følger. Det er lagt vekt på sitater som går utover den offisielle kravspesifikasjonen:

“Mulighet for samarbeid og samspill (Klara Klok som svarer på spørsmål). Kanskje åpent diskusjonsforum (periodevis). Men det er mye arbeid med å vedlikeholde, derfor periodevis. Diskutert mye med deltagerne, hva de skal fylle portalen med. Skal de legge inn stoff selv direkte, eller gå via de på PLU?”

“Lyd er også veldig enkelt. Det er en kilde som er litt mer personlig. Det sosiale aspektet trår inn litt her. Multimediasenteret sitt opplegg lar folk velge å se enten slide eller høre talen, eller begge deler. Sikkert mange som hører på radio ute i skolen. Kunne godt brukt mer lyd. Lydkvaliteten er sikkert viktig her.”

“Skype er et veldig greit verktøy om man ikke sitter i samme rom.”

“Programmet for å redigere websidene bør være webbasert og enkelt.”

“Eportefølge, et verktøy for de med digitale porteføljer. Studenten kan lage sine egne digitale læringsobjekter (de har selv kontroll på hva de legger ut). V.h.j.a. web 2.0-teknologi kunne man kommentere hverandres objekter. Man bør kunne kommentere videoene; f.eks. gi dem en score. Studentene kunne hjulpet hverandre på den måten, og det hadde vært nyttig feedback også for lærerne.”

“En kollektiv hypertekst som studentene selv lager.”

“Videoer er ønskelig, men de bør ikke være lenger enn et kvarter. Lydopptak er veldig aktuelt som opptak av foreldresamtaler.”

“Man har behov for å koordinere og planlegge arbeid online.”

Sitater fra intervjuobjekter angående ikke-funksjonelle forventninger til plu-veven (med vekt på sitater som går utover den offisielle kravspesifikasjonen):

“Et prinsipp at sidene skal følge NTNU-malene, med samme kolonnebredde og stilark. Hvorfor ikke bare bruke dette? Portalen bør kvalitetssikres av webdesignere. Den bør være brukervennlig, enkel og logisk organisert.”

“Brukervennlighet - lett å bruke for nybegynnere.”

“Funksjonaliteten er selve webportalen. Den må være lett å lære. Man er bare interessert i å få jobben gjort.”

“Viktig at den er dynamisk og oppdateres jevnlig. Interaktiv innen visse grenser. Bør utnytte det positive potensialet i pekere og lenker. Det bør ikke være for lange tekstbiter. Ideelt sett bør det ikke være mer enn ett skjermbilde med informasjon av gangen.”

“Websidene skal være intuitive - like enkle i bruk som å ta en telefon.”

“Ideelt sett burde man bygge opp plu sine nettsider i stedet for å lage en ny portal.”

“Portalen burde være selvinstruerende og selvinnlysende.”

Sitater fra intervjuobjekter angående framtidig innhold i plu-veven (med vekt på sitater som går utover den offisielle kravspesifikasjonen):

“Oppgaver fra tidligere studenter. Ressurser for studentene selv. Etter- og videreutdanning.”

“Man har ikke behov for å holde oversikten online over det andre aktører innen etter- og videreutdanning gjør. Portalen skal først og fremst synliggjøre PLU-kompetanse.”

På spørsmål om hvorvidt den virkelige identiteten til brukerne av webportalen skulle vises i

portalen, svarte intervjuobjektene bl.a. som følger:

“Kanskje ikke. Fagpersonene bør dog synliggjøres. Det handler litt om kvalitetssikring. De andre deltagerne kan være mer anonyme. Det handler litt om taushetsplikt.”

“Ja. Dette går på synet på læring. Læring kan ses på som en kollektiv ting. Hvordan ville det være i en forelesningssal der folk har en hette over hodet og fordreid stemme?”

“Foreløpig ikke. Vil ha noe åpent, som fremdeles ser ut som et nettsted utad.”

“Ja. Synes at den virkelige identiteten skal vises. Man burde ikke skjule seg bak en anonym identitet. Man kunne for eksempel brukt bilder av seg selv.”

På spørsmål hvilket innhold plu-veven skal ha, svarte noen intervjuobjekter:

“Det burde være ressurser knyttet til FoU-arbeid, metode, observasjon, datainnsamling, kildekritikk, litteratursøking og det å veilede osv.”

“Alt av interesse for en lærers hverdag. Vi har valgt å begynne i det små, og har knyttet det til lærerens FoU-kompetanse.”

“Man burde tilby informasjon om FoU-virksomheten som ansatte ved PLU driver, slik at man kunne knyttet studentenes virksomhet til hva de driver med på PLU.”

Noe sitater mer generelt rettet til behov innenfor nettbasert tjenester (ikke nødvendigvis plu-veven):

“Der det butter i mot er på skriving av matematiske formler og symboler. Til nå er Moodle det som er best, men langt fra optimalt. Det burde finnes noe som er enkelt og greit for studentene.”

“En god teksteditor er ønskelig. Også et møteverktøy hadde vært gunstig for studentene.”

“Blogg gir noe annet enn det andre. Det er en blanding av diskusjonsforum og hjemmesider. Det enkle grensesnittet for blogg motiverer dem, at det er enkelt å lage og at de selv har sin egen blogg. - Se bloggen min!”

“Alt som bidrar til å bedre kommunikasjonen mellom lærerutdanning og skoleverden er av det gode.”

Som man ser av utdragene fra intervjuene, så har man noe mer detaljerte kommentarer til plu-veven hva angår krav og muligheter for veven, selv om disse ikke er risset inn i stein.

4.2 Nettbaserte verktøy

Teori rundt nettbaserte verktøy angår f.eks. hva som finnes av nettbaserte verktøy eller funksjoner, og hvordan de burde tilfredsstillte anerkjente standarder innen brukbarhet og brukervennlighet, hvor utbyggbar de er, hvor lett de integreres med eksisterende systemer osv. Dette kapitlet vil ta for seg teorier rundt generelle krav til nettbaserte verktøy. Samtidig vil noe bli relatert til PLUs behov, og hva som kreves av de nettbaserte ressursene i lys av deres behov. Det som kommer fram av konklusjoner i dette kapitlet kan benyttes i *design* av nettbaserte verktøy, som er tatt for seg i kapittel 6.

Siden det finnes for eksempel mange funksjoner og typer nettbaserte verktøy, vil først og fremst de som kan være relevant for plu-veven bli tatt for seg. Teori om verktøy som kommer langt utenfor PLUs uttalte område når det gjelder plu-veven blir ikke omtalt.

4.2.1 Artikler

Artikler er tekster som presenteres direkte i websidene, oftest i HTML-format. Dette er teksten som utgjør selve innholdet på nettsidene (ikke navigasjon, lenker eller bilder o.l.). De fleste nettbaserte verktøy har en slik funksjon. Noen vil kanskje skille mellom statiske websider og dynamiske websider, og argumenterer for at statiske websider bare består av artikler, mens dynamiske ikke nødvendigvis trenger å presentere artikler (men kan istedet presentere lenker, forum, menyer, interaktive sider o.l.). Men et motargument mot denne oppfatningen er at statiske sider ikke alltid er det samme som artikler per definisjon, fordi man kan definere artikler som en tekst på nett som har et innhold og mening i seg selv, og ikke representerer koordinering, avtaler, logger e.l., som jo en statisk webside også kan presentere.

Artikler blir ofte også omtalt som *innhold (content)* i CMS. Ofte er det mer navnet på en funksjon i verktøyet, der man kan også inkludere bilder, video, lenker m.m. integrert i teksten. Eksempler på CMS som har en slik funksjon, er Joomla! (undersøkt senere i rapporten), bcoos m/flere. Noen CMS har en lignende funksjon, men kalles der for *page* eller *story* (f.eks. eZ Publish og Drupal). I noen CMS er artikler-funksjonen også brukt for nyheter på websidene.

Artikler på nett er svært relevant for plu-veven. Mye informasjon vil naturlig nok forekomme i direkte tekst på nettsidene. Dette kan variere fra en informasjonsside om formålet med plu-veven, til oppslag og faglige artikler og nyheter. I studentgruppas løsning forekommer ikke uttrykket artikler. Istedet har man brukt konseptet *tema*. På samme måte som med artikler kan man med tema skrive inn innhold i HTML-format, som vil presenteres på en webside. Denne funksjonaliteten kan ikke helt sammenlignes med artikkel-funksjonen i de andre verktøyene (Joomla!, bcoos o.l.) fordi i studentgruppas løsning er det til hvert tema knyttet et fil- og videoarkiv, og et forum, noe som gjør tema-funksjonaliteten noe bundet til annen funksjonalitet (arkiv og forum). Denne løsningen kan gjøre det enklere om behovet i første omgang er kun å legge opp sidene etter tema-inndeling, men på sikt blir dette lite modulært og vedlikeholdsvennlig. Et eksempel er om man ønsker seg en enkel webside som ikke er en framside og som ikke skal ha knyttet et eget forum eller arkiv til seg, f.eks. en side for kontaktinformasjon. En artikkel-funksjonalitet kan løse dette enkelt.

4.2.2 Forum

Diskusjonsfora på nett har eksistert noen år, og er vel kjent for de fleste internettbrukere. Disse blir også kalt elektroniske oppslagstavler (bulletin boards). Der kan man poste innlegg og svare på innlegg, og ofte også starte diskusjonstråder med et eget tema. Mange verktøy kommer med ferdiglagde forum som man bare trenger å integrere med sitt eget system. Et eksempel er den utbredte og populære open source forumløsningen phpBB (<http://www.phpbb.com>), som er basert på PHP. Online forum kan variere stort i kompleksitet, fra løsninger som tilbyr en rekke tilleggsfunksjonaliteter; bildeinnsetting i innlegg, profiler og avatarer (bilder eller symboler som representerer brukeren), rangeringer av brukere, kalenderfunksjonalitet etc., til enkle løsninger som tilbyr bare det mest grunnleggende; posting av og svar på innlegg i rent tekstformat, med en flat struktur.

Forum kan være trådede eller flate. I trådede fora spesifiseres det hvilke innlegg man svarer på, noe som tillater flere forskjellige tråder å utvikle seg fra det opprinnelige start-innlegget. I en flat struktur legges bare siste innlegg i diskusjonen til bakerst i en rekke med tidligere innlegg, som om strukturen tilsvarer et enkelt rom der alle innlegg uttrykkes i plenum (se figur 8 og 9 i kap. 5.1.1.1 for et eksempel på flate og trådede forumløsninger).

For å redusere såkalt “forurensing” i nettbaserte fora er en interessant funksjon introdusert i bl.a.

Google Groups. Forurensing i fora forekommer når brukere ikke bryr seg om å lete etter eldre diskusjoner med samme tema, og istedet poster en ny tråd med samme tema. Dette kan skape problemer for moderatorer, i følge Wikipedia. Google Groups ønsker å redusere dette ved å la vise eldre diskusjonstråder med samme temanavn i det en bruker prøver å lage en ny diskusjonstråd med temanavnet. Brukeren vil sannsynligvis på den måten lettere finne svar på sitt spørsmål i en allerede eksisterende diskusjonstråd, og en opprettelse av en ny diskusjonstråd med samme tema kan dermed unngås.

Vanlig funksjonalitet i nettbaserte forum er som følger:

- Noen fora lar administrator eller brukerne selv tilpasse utseendet for forumet.
- En kalender-funksjon tilbys i noen fora, der man kan lagre hendelser og avtaler.
- De fleste forum støtter opplastning av bilder assosiert med en post
- De fleste fora holder oversikten over meldinger brukeren ikke har lest. Noen gjør dette gjennom brukers *sesjon*, der alle poster som er postet etter brukers siste “visitt” på forumet merkes som ulest. Andre ivaretar historie over leste poster utover en sesjon slik at det til alle tider vil vises hvilke meldinger brukeren faktisk har lest eller ikke, i tillegg til at brukeren ofte har mulighet til selv å merke leste poster som ulest.

Forum på nett er diskutert i forskningslitteratur bl.a. av [13], der de diskuterer lærere og skoleledelsens bruk (ved Monash University) av et online diskusjonsfora som studentene bruker flittig. De diskuterer også hvorvidt online diskusjonsfora har noen læringsverdi for studentene.

Eksempler på *gratis* forum basert på PHP er: AEF (<http://www.anelectron.com/>), BabbleBoard (<http://www.babbleboard.co.uk/>), bbPress (<http://bbpress.org/>), phpBB3 (<http://www.phpbb.com/>) og Deluxe Portal (<http://deluxeportal.net/>). Eksempler på *proprietære* forum basert på PHP er: Burning Board (<http://www.wolflab.de/>), FusionBB (<http://www.fusionbb.com/>), vBulletin (<http://www.vbulletin.com/>) og WowBB (<http://www.wowbb.com/>). De fleste av disse tilbyr kun flat struktur og støtter valgfrie temaer. Nesten alle PHP-baserte fora bruker MySQL som database der data lagres.

Eksempler på *gratis* forum basert på ASP, ASP.NET eller variasjoner av dette: bttlxeForum (<http://www.bttlxe.com/bttlxeForum/Index.aspx>), dnfBB (<http://sourceforge.net/projects/dnfbb>), GoBoardz (<http://goboardz.com/>) og Yet Another Forum (<http://www.yetanotherforum.net/>).

Eksempler på *proprietære* forum basert på ASP, ASP.NET eller variasjoner av dette: Active Forums

(<http://www.activemodules.com/>), Discussion Board (<http://www.activedataonline.com.au/discussionboard.html>), ITCN Enterprise Forum (<http://forum.itcn.com/>) og Web Wiz Forum (<http://www.webwizguide.info/>). Andelen av disse som i tillegg til flat struktur tilbyr trådet struktur er høyere enn for de PHP-baserte forumene, men andelen som tilbyr full oversikt over uleste meldinger i forumet er noe mindre enn for PHP-baserte forum. Ellers har de omtrent samme funksjonalitet i det store og hele som de PHP-baserte forumene.

Eksempler på forum basert på andre språk enn PHP og ASP/ASP.NET er: E-Blah (<http://www.eblah.com/>), Groupee Forums (<http://www.groupee.com/>), Ikonboard (<http://www.ikonboard.com/>) og Kareha (<http://wakaba.c3.cx/>). Disse har omtrent samme funksjonalitet som de PHP- og ASP/ASP.NET-baserte forumene⁸.

Relevans for plu-veven

Bruk av online fora, spesielt komplekse fora med mange funksjoner, krever noe datakjennskap av brukeren, i tillegg til nødvendig tid for å bli kjent med applikasjonen. Plu-veven forventes å ha brukere med middels til lave datakunnskaper. Det er derfor en fordel om forum-komponenten holdes på et enkelt nivå. Det er fullt mulig å kombinere dette med stor bredde i funksjonalitet, men da må denne kompleksiteten holdes skjult som standard, slik at nybegynnere ikke forvirres eller skremmes vekk. For de som vil grave i verktøyet og bruke en rekke funksjoner, bør dette være tilgjengelig et sted i verktøyet, men det bør ikke framstå som noe krav at brukeren skal beherske mer enn de mest grunnleggende forum-funksjonene.

4.2.3 Blogg

Ordet “blogg” er en vanlig forkortelse av ordet “web log”. Blogg som verktøy oppstod ca. i 1997 da denne skrive- og presentasjonsformen på web ble begynt brukt [23]. En blogg kan sies å være en nettbasert *journal*. Vi kan definere blogg som en serie arkiverte internett-poster, typisk karakterisert ved korte tekster skrevet i motsatt kronologisk rekkefølge og som generelt inneholder hypertekstlenker til andre sider. Blogger er vanligvis subjektive framstillinger; uttrykk sett fra forfatterens vinkel, med en klar fornemmelse av tilskuere. Bloggingens historiske forgjengere kan sies å være papirjournaler, dagbøker og årbøker.

I en blogg vil forfatteren poste sine tekster i en serie, der siste post som oftest står øverst [23].

⁸ http://en.wikipedia.org/wiki/Comparison_of_Internet_forum_software

Postene har dato og tidsstempel, og kan ofte kommenteres av andre som leser bloggen. En blogg er jo en slags nettside, men forskjellen mellom en blogg og en vanlig nettside er bloggens frekvente, rytmiske (ofte korte) poster i kronologisk rekkefølge. En blogg kan også oppfattes som mer verbal og som om den har en stemme, mye på grunn av bloggens verbale form og personlige stil. Blogging innebærer i tillegg *samfunnsbygging (community)*, siden man søker å tiltrekke lesere av bloggen sin, og får tilbakemeldinger av disse gjennom kommentarer. Man vil også lenke til andre bloggeres blogger i sin egen blogg, eller til bestemte nettsider.

Det er varierte grunner til at folk blogger. Noen ønsker bare å “blogge livet”, ved at de blogger sine daglige aktiviteter og tanker, for å “bekrefte sin egen eksistens”. Noen blogger for å holde orden på sine aktiviteter. Noen blogger for å få sagt noe som ligger dem på hjertet - sine meninger og kommentarer, mens andre blogger for å holde et nettbasert samfunn oppdatert om f.eks. utviklingen av en tjeneste eller type programvare [23]. Blogging er også brukt innenfor utdanning. Et eksempel er studentenes bruk av blogger i PLU-studiet ved NTNU. To av bloggene i studiet av Nardi et al var også relatert til støtte av utdanningssamfunn. Nardi et al forventet at andelen utdanningsrelatert blogging ville øke i årene etter studiet (2004) og at blogging vil bli en viktig kommunikasjonsform i arbeidsområdet.

Funksjonalitet i bloggverktøy kan inkludere redigering av utseende (ofte enkelt), kommentering og redigering av poster, passordbeskytting av blogger, gjøre bloggen søkbar i søkemotorer, opplasting og posting av bilder og andre mediefiler, og arkivering m.m. Ofte er det også mulig å få sett hvem som besøker bloggen din, eller hvor stor trafikk den har.

Eksempler på “out of the box” bloggverktøy (gratis) er Drupal, simplePHPBlog, Slash og WordPress. Eksempler på proprietære bloggverktøy er Community Server, ExpressionEngine og Windows Live Writer. Verktøy som kjøres på utvikler-vert (trengs ikke installeres på lokal maskin) er f.eks. Blogger, MySpace og WordPress.com.

Relevans for plu-veven

Kommentarer i blogger er et viktig aspekt å ta med i vurderingen av verktøyet, siden en bloggs kommentarer vil påvirke hvordan man oppfatter bloggposten, i følge konklusjonen på et brukerstudie utført i [24]. Av dette kan man vurdere nærmere om det er hensiktsmessig å stimulere til kommentering av blogger i plu-veven eller ikke. Hvis det nettbaserte samfunnet er som ønskelig, kan kommentarer i blogger bidra i positiv retning med tanke på det læringsmessige utbyttet av bloggene. En annen faktor som bør diskuteres av PLU er hvorvidt bloggene skal være tilgjengelige

for allment besøkende, eller om de kun skal være tilgjengelige for registrerte brukere. I det første tilfellet vil dette føre til at rammene for samfunnet i plu-veven utvides noe utover registrerte brukere, om vi tar [24] og [23] til etterretning.

Når det gjelder funksjonalitet, kan det være en fordel å holde kompleksiteten på bloggverktøyet på et minimum. Dette bunner i at folk ikke ønsker å bli forvirret av web'en, som påpekt i [25], og som rapporten går mer inn på i kap. 4.3. Som med flere av de andre verktøyene, kan flere funksjoner implementeres, men burde skjules for umiddelbar innsyn av noviser innen blogging, for å unngå følelsen av at man ikke mestrer verktøyet. Funksjoner som med fordel kan implementeres i plu-veven, foruten basisfunksjoner som posting og redigering av poster, er:

- Kommentering av bloggposter (er med på å bygge et nettbasert samfunn ved hjelp av tilbakemeldinger)
- Begrenset tilpasning av en bloggs utseende (hvis bloggene skal ligge under NTNUs nettsider bør man forhøre seg med NTNU hvor stor frihet man har til å avvike fra et NTNU-design). Man kan kanskje tilby et sett med NTNU-temaer. Blir plu-veven landsdekkende står man mer fritt i valg av utseende, bare begrenset av nettstedets eget image.
- Fra bloggen kan det gjerne tilbys en link til forfatterens profil på plu-veven, så leserne vet mer om den som har skrevet den.
- Leserne kan tilbys RSS feeds av hver blogg
- Innskriving av poster burde ha kjente tekstbehandlingsfunksjoner, som font, fontstørrelse, tekstformatering, tekstjustering o.l.
- Bilder og andre mediafiler burde kunne lastes opp i bloggen, siden forskjellige medier kan bety mye i en utdanningssammenheng.
- Bloggforfatter burde kunne velge om bloggen skal kommenteres eller ikke, og om mulig velge hvilke kommentarer som skal slettes. Eventuelt også godkjenning av kommentarer før de postes. Tanken bak dette er at folk ikke skal henges ut av upassende kommentarer på bloggen sin.
- Mulighet for å lage postmal (forhåndslager tekst og kode) som vises hver gang man oppretter en ny post
- For å utvide kommunikasjonen og debatten mellom forskjellige blogger, kan det valgritt

inkluderes *tilbakelenker* i bloggene. Funksjonen gjør at bloggen kan vise hvilke andre nettsider eller blogger som linker til bloggen.

- Mer avanserte funksjoner for å tillate eksterne tjenester for innmating (a la Feedburner)
- Om ønskelig, en funksjon som tillater flere forfattere på én blogg

Begrunnelsen for disse funksjonene er at de ivaretar de mest grunnleggende funksjonene samtidig som noen avanserte funksjoner tillates.

4.2.4 IM

Det finnes flere typer nettbasert IM eller chat (nettprat). De fleste er relativt like, ved at brukeren spesifiserer et navn som sin representasjon av seg selv (avatar) gjennom pratesesjonen. Ved siden av chatvinduet står vanligvis alle brukere som deltar i pratesesjonen eller i “praterommet” listet opp. Alle disse kan skrive inn det de vil si, og meldingen vil vises umiddelbart i chatvinduet så alle brukere kan se det. Brukere kan også invitere andre brukere til spesielle “praterom” [26]. Internet Relay Chat (IRC) er en av de mest kjente chat-programmer, og MUD (Multi User Dungeons) har benyttet chat i mange år. Ofte vil brukere som ikke kjenner hverandre i det virkelige liv prate med hverandre over chat, spesielt i IRC og MUD [26].

Vi kan skille mellom selvstendige chat-applikasjoner og chat som foregår direkte i nettleseren (webgrensesnitt).

Relevans for plu-veven

[27] oppdaget hovedsaklig to bruksstiler angående IM (instant messaging):

- Arbeide sammen
- Koordineringer

Selv om tidligere studier hadde fastslått at IM i hovedsak ble brukt til raske spørsmål, koordinering og sosial interaksjon, så fant [27] ut at IM ble vel så mye brukt i arbeidssammenheng på en arbeidsplass, til å diskutere arbeidsrelaterte spørsmål. Sett i lys av dette kan IM være av spesiell interesse for plu-veven, i tillegg til å være en funksjon for sosial interaksjon. IM er en synkron (eller nesten-synkron) kommunikasjonsform som kan tilfredsstillende behov for umiddelbar kontakt mellom brukere av plu-veven, hvis telefon eller andre medier ikke er å foretrekke.

Hvis plu-veven implementerer IM, burde dette initieres i og fungere i et webgrensesnitt, slik at brukerne ikke trenger å installere eksterne IM-program på lokal maskin. Om installasjon er nødvendig, bør dette skje noenlunde automatisk.

4.2.5 RSS

RSS står for Really Simple Syndication eller Rich Site Summary⁹. RSS er basert på XML, et vidt brukt format for tekstlig informasjonsutveksling mellom internettapplikasjoner. Med RSS kan man lage feeds (matinger) fra websider og blogger og lignende, som man ved hjelp av en RSS-leser kan samle og lese. I en typisk RSS-leser vil man få listet opp de nyeste oppdateringene i informasjonsstrømmen (feed), hva man har lest og ikke lest, osv.



Figur 4: Hvordan en RSS-leser kan framstå. I dette tilfelle er informasjon matet inn fra Yahoos RSS-feed over nyhetsoverskrifter.

⁹ <http://www.google.com/support/feedburner/bin/answer.py?answer=79408#howtoread>

RSS er listet i kravspesifikasjonen for plu-veven, og er derfor å betrakte som en ønskelig funksjon i denne. Begrunnelsen for å ha med denne funksjonen er for å holde seg oppdatert på nyheter/innhold på nettstedet (plu-veven). Dette kan være veldig nyttig for plu-veven, siden folk da kan se når nye oppdateringer har forekommet på nettstedet. Funksjonen forutsetter dog at brukerne vet hvordan man benytter seg av funksjonaliteten. Plu-veven vil bare tilby en feed som går ut fra siden. Brukerne må selv konfigurere sin RSS-leser til å motta feed fra plu-veven. Kanskje kan en enkel steg-for-steg bruksanvisning for dette legges ut på plu-veven?

4.2.6 Wiki

En Wiki er en nettbasert applikasjon for samhandlingsorientert kunnskapsbehandling [28]. Ordet “wiki” er et Hawaiiansk uttrykk som betyr “rask” eller “å haste”. Dette symboliserer de raske endringene i redigeringsprosessene i en wiki. En wiki er en samling sammenlenkede HTML-sider som har krysslenker mellom interne sider. Disse sidene kan redigeres. Endringene kan også reverseres til opprinnelig form. Brukere trenger ikke ha teknisk eller webrelatert kompetanse for å legge til kunnskap i en wiki.



Figur 5: Skjermbilde fra den kjente wiki'en Wikipedia (www.wikipedia.org).

En organisasjon som vil overleve og vokse i det globale konkurrerende markedet trenger å lære seg *organisasjonsmessig læring* [28]. En wiki tar fordel av den samhandlende innsatsen fra alle organisasjonens medlemmer i å lage en lett tilgjengelig kunnskapsdatabase. Sammenlignet med tradisjonell kunnskapsbehandlingssystemer legger wiki mindre vekt på sentralisert kontroll, streng disiplin og utbredt overvåking av systemer.

Potensielle “farer” ved bruk av wiki er at “ukorrekt” informasjon kan redigeres inn i wiki’en (kan løses v.h.j.a. revisjonskontroll der forfatteren identifiseres), og at metoder for kvalitetskontroll er veldig vanskelig å gjennomføre. Vedlikeholdsbehovene av wikisystemet kan øke etterhvert som wiki-databasen øker [28]. Det vil også være vanskelig å bestemme den virkelige kilden til stoffet siden det er mange bidrag til sider på en wiki.

Eksempler på forskjellige wiki som tillater forskjellig innhold er f.eks. SnipSnap som tillater brukere å legge inn tekstrepresentasjoner av mind maps, organigrammer og UML-diagrammer sammen med teksten. OpenWiki tillater brukere å berike sidene med matematiske former skrevet i MathML. TWiki-brukere kan inkludere LaTeX markup kommandoer gjennom en plugin. Endel wiki’er kommer med intern stavekontroll. Dette er vanligvis en frivillig funksjon. Man har også mal-funksjonalitet, f.eks. Wikipedia. PurpleWiki benytter lilla nummer som identifiserer paragrafer, lister, overskrifter osv. Dette fungerer som en lenkemekanisme [29].

Miki er kodenavnet for en pågående (2006) satsning for å lage en infrastruktur for samhandlende skriving av formalisert matematikk på nett [29]. Med formalisert matematikk menes at matematiske konsepter som definisjoner, lemma og teoremer blir kodet i en formalisme som muliggjør mekanisk undersøkelse av gyldigheten av bevis. Eksisterende systemer for dette er ofte sentraliserte og behandlet av utviklerne av en gitt teorem-beviser eller bevis-assistent. Miki forsøker å gjøre skriving av slikt språk mer desentralisert og “wiki-lignende”.

Relevans for plu-veven

Wiki er nevnt i den opprinnelige kravspesifikasjonen som et verktøy som skal opp til diskusjon i en senere fase i utviklingen av plu-veven. Det som taler til fordel for en wiki som en nettbasert tjeneste er den lett tilgjengelige kollektive kunnskapsbasen den gir for brukerne. Dette pluss PLU-ansattes egne bidrag til basen kan bidra til *organisasjonsmessig læring* for PLU som organisasjon. Det som kanskje taler i mot er at det tar tid å bygge en slik database, og det forutsetter mer eller mindre frivillig deltagelse fra mange parter. Hvis man ser langsiktig på dette, og begynner med å bygge rundt relevant og aktuell lokal kunnskap, kan dette ha noe for seg. Enten vil kunnskapsområdet og

dybden i wiki-verktøyet utvide seg, eller forbli på samme nivå. I begge tilfeller kan det være av nytte for brukere av webportalen. Vedlikeholdskostnadene, og det faktum at noviser ikke vil være så aktive i byggingen av kunnskapen, må dog spille inn. Kvalitetssikring er et annet aspekt, og her bør det kanskje sikres at kun fagmessig anerkjente personer får tilgang til å redigere innholdet i wiki-verktøy.

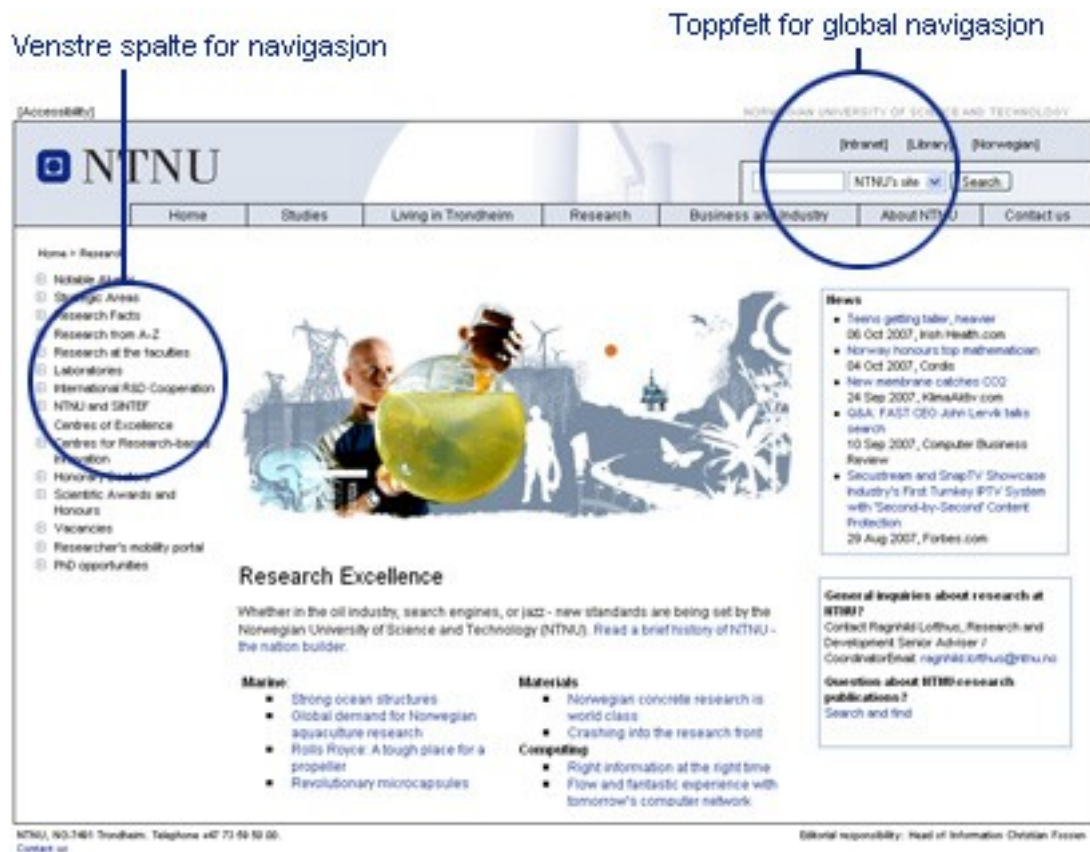
Kilder i PLU etterlyste muligheter for å skrive matematiske uttrykk på nett. Her kan kanskje en plugin eller funksjon som tillater dette, a la Miki, være av interesse.

4.3 *Designteori*

Dette kapitlet tar for seg designteori relatert til nettbaserte tjenester. NTNUs designprinsipper for deres eksterne sider vil bli tatt for seg, i tillegg til teori rundt design tilpasset funksjonshemmede, og design tilpasset mobilt bruk.

4.3.1 NTNUs designmal for websider

Det er et krav til PLU-veven, ihverfall på kort sikt og så lenge portalen befinner seg under NTNUs nettsider, at sidene oppfyller NTNUs designkrav til websider [17]. Et skjermbilde av typisk layout for slike sider er vist i figur 6.



Figur 6: Sidelayout fra NTNUs designmal for websider under NTNUs nettsider.

Følgende punkter er tatt fra NTNUs malverk for websider under NTNU:

- I toppen skal det ligge et bilde som er felles for hele NTNU.
- Gridmal med midt- og høyre spalte: Her anbefaler man å bruke midtspalten til det websida først og fremst skal formidle, og høyre spalte til f.eks. kontaktinformasjon, lenkesamling, multimedia osv.

Gridmal med kun én spalte: Denne anbefales først og fremst om man skal presentere tabeller. Den egner seg ikke spesielt godt til ren tekst siden tekstspalten da blir veldig bred, og det blir vanskelig å lese.

- Hver enhet (her: PLU-veven) styrer sine egne lenker i venstre spalte.
- NTNUs eksternweb skal være inkluderende i forhold til brukere med ulike funksjonsevner. Stikkord er tabellfri layout, alternative tekster for bilder, beskrivende lenker o.l. Overskrifter

for innhold bør også presenteres hierarkisk ved hjelp av header-elementene i HTML (altså ikke bruk f.eks. et H3-element for en overskrift som står over innhold som har en h2-header etc.).

- Man skal ikke benytte (uetiske) triks for å "lure" søkemotorene og få bedre rangering i disse. Et godt råd for søkemotoroptimalisering er heller å skrive godt for brukerne; da skriver man godt for søkemotorene også.

4.3.2 Tilpasset funksjonshemmede

Grensesnittet burde vært tilpasset brukere med funksjonshemming. Man kan anvende Neumans følgende råd for å tilfredsstille dette [31]:

- Presenter prosedyrer, direksjoner og verbalt innhold på nivåer og formater som gjør dem tilgjengelige for folk med lesevansker
- Sikre at responskrav ikke lar brukere fullføre programmer uten å engasjere seg i målkonseptene
- Design tilbakemeldingssekvenser som forklarer årsakene til brukeres feil og som leder dem gjennom prosessene nødvendig for å respondere korrekt

Det er samtidig viktig at man ikke designer systemet slik at det hemmer andres læring eller utvikling bare for å få det likt for alle.

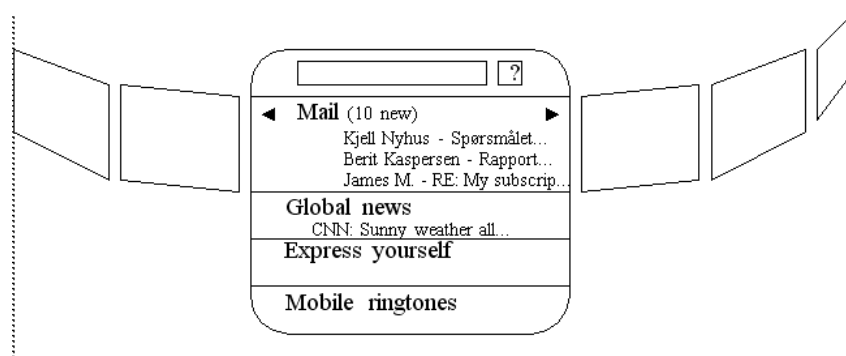
Denne nylig publiserte artikkelen¹⁰ på Operas hjemmesider advarer mot utbredt bruk av AJAX før man har gode nok standarder som gjør det mulig for hjelpemidler (f.eks. skjermlesere) å tolke og presentere informasjonen best mulig for brukeren. Poenget kan være at hvis noe kan implementeres uten bruk av AJAX eller andre webteknologier som hjelpeteknologier har problemer med å tolke, så burde man heller bruke det som hjelpeverktøyene støtter.

¹⁰ <http://dev.opera.com/articles/view/stop-using-ajax/>

4.3.3 Tilpasset mobilt bruk

Bruken av mobile enheter har blitt mye utbredt både for personlig og forretningsmessig bruk [30]. Framkomsten av mobile enheter og den raske utviklingen av trådløs kommunikasjon har også åpnet opp alternative måter for høyere utdanning å ta i bruk mobil læring for å overføre kunnskap til den lærende [39]. En fordel med mobil læring i forhold til e-læring er muligheten for mobilitet. En webportal som ønsker stor utbredelse kan også dra nytten av utvidelsen til mobile enheter. For å få til dette må informasjonen som presenteres på mobile enheter tilpasses de små skjermene på disse enhetene. Noen oppgaver, som å surfe og søke på web, er ineffektivt på små skjermer [38]. Brukere foretrekker mer “kompakt” presentasjon av relatert informasjon på mobile enheter slik at de ikke trenger å navigere gjennom komplekse menystrukturer [37]. Andelen vertikal rulling burde begrenses. Vertikal rulling fungerer likevel bra på visse mobile enheter med litt større skjerm, og nyere enheter med raskere hardware og smartere programvare, som håndterer rullingene raskere og mer sømløst tidligere enheter.

[37] foreslår i sitt studie et grensesnitt-design (tilpasset små mobile skjermer) der skjermen viser et begrenset antall “hjul” som ligger oppå hverandre og består av “ruter”. Hvert hjul kan ha en forskjellig type innhold eller funksjonalitet som brukeren kan rullere gjennom, og hvert hjul har til enhver tid en “rute” som er i fokus, der et spesifikt innhold eller en funksjon presenteres. Hjulene kan dreies mot venstre eller høyre ettersom brukeren trykker på høyre piltast eller venstre (eller det som tilsvarer høyre og venstre på en mobiltelefon). Brukeren kan så velge underliggende eller ovenforliggende hjul og dreie dem. Grensesnittet ble funnet å tilfredsstille brukerne som deltok i testingen av applikasjonen.



Figur 7: Grensesnittdesignet i mobil utgave av Windows Live.

Figur 7 illustrerer hvordan applikasjonen Windows Live ble testet i dette grensesnittet. Skjermen med 5 “hjul” er vist, der hjul nr. 2 fra oven er valgt. Rutene til høyre og venstre for skjermen er bare vist for å illustrere at brukeren kan rullere fokuset til aktuelt “hjul” til høyre eller venstre.

Designet i [37] kan tenkes å anvendes på plu-veven på følgende måte. Det blir tilbydd et antall funksjoner på den mobile utgaven av plu-veven, med informasjon fra webportalen plu-veven. De viktigste funksjonene for brukeren blir lagt i øverste “hjul”. Dette kan f.eks. være informasjon og innhold som artikler, pdf-filer og presentasjonsfiler. De forskjellige rutene kan inneholde henholdsvis dette:

- 1. rute:** Liste over de nyeste artiklene lagt til webportalen. Når brukeren trykker på “aktiveringsknappen” eller “OK” med denne ruten i fokus, tas han/hun til et fullskjerm bilde med liste over alle artikler. Brukeren kan velge den artikkelen han/hun vil se på, og denne vil da vises i fullskjerm.
- 2. rute:** Samme som i 1. rute, bortsett fra at her vises en liste over pdf-artikler som brukeren kan velge å se på (gitt at mobilen støtter formatet).
- 3. rute:** Samme som de forrige rutene, bortsett fra at her vises en liste over presentasjonsfiler, osv. osv.

Neste “hjul” kan inneholde funksjoner som “Se min profil”, “Se diskusjonsforum” og “Se personlige meldinger”, gitt at disse funksjonene finnes i portalen. Neste “hjul” under der igjen kan inneholde ruter som “nyheter”, “søk” osv. Hvordan dette organiseres, kommer an på hva som blir å finne i selve webportalen av plu-veven, og hvilken retning prosjektet tar.

Andre viktige brukbarhets- og designaspekter som bør fokuseres på ved en mobil utgave, er:

- Bruk grafiske metaforer som nøkkel og lås for sikkerhetsrelaterte funksjoner [38] som f.eks. innlogging.
- Innloggings- og sikkerhetsaspektene må designes slik at de ikke er for vanskelige eller tar for lang tid. Dette resulterer i at den mobile utgaven ikke blir tatt i bruk [38].
- Bruk en tiltrekkelig stor fontstørrelse, slik at lesbarhet på små skjermer sikres.
- En tilbakemeldingsmekanisme bør la brukeren vite når en webside har fullført innlastning [38].
- Test den mobile utgaven over treg forbindelse [38].

- La den viktigste informasjonen være på den første websiden. Unngå at brukerne må rulle horisontalt eller vertikalt [38].
- Den bruker-orienterte dialogen må sikte på god mapping mellom den mobile informasjonsvisningen og brukerens konseptuelle modell av informasjonen [39].
- Den mobile applikasjonen bør hjelpe brukere å gjenkjenne, diagnosere og gjenopprette fra feil (programmessige eller brukerinnførte) [39].
- Den mobile applikasjonen bør visualisere status for brukeren [39], f.eks. ved bruk av framdriftsindikatorer.
- Pga. menneskers begrensede korttidshukommelse, bør informasjonsstrukturen være i små og homogene informasjonsbiter, som passer inn i ett skjermbilde av gangen [39].

Ellers gjelder mye av de samme designprinsippene som for vanlige websider på en stasjonær eller bærbar PC.

4.4 Utviklingsteori

En utvikling av en nettbasert tjeneste krever en metode, mens en tilpasning av en “out of the box”-tjeneste ikke krever et veldig rigid rammeverk i form av en utviklingsmetode. Generelt, for et verktøy man utvikler fra bunnen, bør man ta i bruk klare utviklingsmetoder.

Det finnes mange utviklingsmetoder for utvikling av web-applikasjoner. Fortrinnsvis er agile og iterative metoder benyttet i den senere tid for å takle de organisasjonsmessige utfordringene. Dette inkluderer metoder som bl.a. Rational Unified Process (RUP), eXtreme Programming (XP), Adaptive Software Development, Cockburn’s Crystal Family, Scrum, Feature Driven Development, Dynamic System Development Method, Lean Development og Macromedia Fusebox Lifecycle Process (FLiP) [40]. [40] presenterer en webutviklingsmetode som støtter typisk kommersiell webutviklingspraksis og samtidig tar forbehold for de karakteristikkene websystemer har. Man har modifisert FLiP slik at man er uavhengig av å benytte spesielle støtteverktøy i prosessen. Behovet for abstrakt designmodellering er ivarettatt gjennom en tilpasset versjon av Web Application Architecture Framework (WAAF). Metoden består av 3 hoveddeler: prototyping, informasjonsmodellering og systemarkitektur. Man starter med prototypingen, som formulerer frontenden. Både prototypens oppførsel og struktur blir vurdert i dette steget. I informasjonsmodelleringssteget modelleres prototypen i en informasjonsarkitektur (IA). Steget kan

deles opp i *informasjonsstruktur* og *informasjonsoppførsel*. I systemarkitektur-steget konstrueres moduler som tilsvarere designet, og koden modulene skal bestå av. *Systemarkitektur-struktur* og *systemarkitektur-oppførsel* er aktiviteter i dette steget.

Den videre utviklingen av plu-veven kan tenkes å ta i bruk denne utviklingsmetoden. En skisse av hvordan de initielle stegene for plu-veven vil være i følge denne utviklingsmetoden er lagt ved i Appendix 10.2.

5 Undersøkelse og evaluering av eksisterende nettbaserte verktøy

Dette kapitlet tar for seg undersøkelse og evaluering av eksisterende nettbaserte verktøy. Dette er verktøy som finnes ute på markedet i dag, enten open source eller proprietære, eller en kombinasjon av disse. 3-4 utvalgte større løsninger blir evaluert, mens noen mindre enkeltstående verktøy blir evaluert.

5.1 CMS

Det finnes et stort antall CMS eller WCMS ute på markedet. Det er ikke lett å orientere seg hva som finnes av forskjellige løsninger pga. antallet forskjellige løsninger som varierer ganske betydelig fra hverandre. Følgende systemer blir sett nærmere på:

- Microsoft SharePoint
- Joomla!
- It's Learning

Begrunnelsen for å evaluere Microsoft SharePoint er fordi PLU valgte å ta i bruk dette systemet for plu-veven høsten 2007. En evaluering av dette vil kunne gi råd hvorvidt en løsning med SharePoint i grunnen er en god løsning for plu-veven. Begrunnelsen for å evaluere Joomla! er fordi det er et kjent open source produkt som innehar typisk WCMS-funksjonalitet og med et vell av utvidelser, i tillegg til at jeg hadde noe erfaring med systemet på forhånd. Begrunnelsen for å evaluere It's Learning ligger i ønsket om å finne ut om man simplelthen kunne brukt It's Learning istedet for å utvikle eller velge en ny løsning (It's Learning brukes allerede ved NTNU). I tillegg vil en evaluering av dette klargjøre hva eventuelt It's Learning ikke har som det er behov for i plu-veven.

5.1.1 Microsoft Windows SharePoint Services 3.0 (WSS 3.0)

Hele evalueringen av WSS i dette kapitlet har hentet kildemateriale fra WSSs integrerte hjelp, Microsofts sider om produktet og ved egne undersøkelser.

Følgende lister er et utvalg av egenskaper for Microsoft Sharepoint Services 3.0, som er det systemet plu-veven i tok i bruk ved årsskiftet 2007/2008, og som også er det bestående systemet i skrivende stund. Spesielt interessante funksjoner med tanke på PLU-vevens behov er vektlagt.

Dokumenter (delte): Man kan legge til mapper for dokumenter, legge til og laste opp nye dokumenter etc. Man har også mulighet for mer avanserte handlinger, som synkronisering med Outlook 2007 og eksportering til regneark.

Lister: Eksempler er *kalender* og *oppgaver*. Også avanserte handlinger som synkronisering med Outlook 07, eksportering til regneark og RSS.

- Man kan håndtere sin egen kalender for å legge inn avtaler og hendelser, og dele denne med de andre brukerne. Visning av kalender.
- Man kan lage og håndtere oppgaver og bestemme når og hvem som skal gjøre dem. Visning av oppgaver.

Diskusjoner: Her kan man legge til nye diskusjoner, eller delta i diskusjoner.

Nettsteder (sites): Man kan lage et sett med individuelle nettsteder innenfor det totale systemet, enten som helt uavhengige nettsteder, eller som del-nettsteder under et overordnet nettsted.

Folk og grupper: Gir en oversikt over nettstedets medlemmer, besøkende til nettstedet, og nettstedets eier(e).

Noen av mulighetene:

- Spesifisere rettigheter til brukere og brukergrupper på nettstedet.
- Ordne brukere i grupper
- Kontakte brukere
- Legge til brukere eller brukergrupper o.l.

Nettstedshandlinger: Under denne menyen har man bl.a. mulighet til å legge til såkalte *webdeler* (*webparts*), som kan være f.eks. *wikibibliotek*, *dokument-bibliotek*, diverse *webdeler* for kommunikasjon, *webdeler* for oppfølging og monitorering, og *webdeler* for tilpassede *lister*, og *websider*.

Andre muligheter eller informasjon om systemet:

- Bruk av mapper for å organisere informasjon
- Microsoft søk: søk i nettstedet og alle underliggende nettsteder
- Nettstedsmaler og listemaler kan lages i Visual Studio 2005 og benyttes i SharePoint Services 3.0
- “Quick Launch”-panelet (hovedmenyen på venstre side) og navigasjonspanelet på toppen kan enkelt reorganiseres
- SharePoint Services nyter påliteligheten, skalerbarheten og funksjonaliteten integrert i ASP.NET 2.0. Dette inkluderer støtte for modellen for web-deler i ASP.NET 2.0.
- Innholdstyper er et kjernekonsept i SharePoint Services. Disse brukes for å hjelpe brukere å organisere innholdet. En innholdstype er en gjenbrukbar samling av innstillinger som kan anvendes på forskjellige innholdskategorier. Dette gjør en i stand til å sentralt behandle metadata og oppførsel for en dokument- eller enhetstype.

5.1.1.1 Evalueringen av SharePoint Services

For å evaluere og sammenligne de utvalgte CMS, og avgjøre om de passer til PLUs forretningsmål, er James Robertsons metode for å evaluere et Content Management system benyttet her. Se kap. 2.2.3 for en bakgrunn rundt Robertsons evalueringsmetode, og Appendix 10.3 for selve evalueringsskjemaet. Det er viktig å merke seg at evalueringen ikke inkluderer alle mulige tilleggsutvidelser for systemet, som kan eller kan ikke ha innvirkning på resultatene i evalueringen.

Evalueringen av SharePoint Services etter Robertsons kriterier grupperes i følgende bolker:

- Innholdsskaping
- Innholdsbehandling
- Publisering
- Presentasjon
- Kontrakter og forretning

Innholdsskaping

Innholdsskaping i WSS (Windows SharePoint Services) gjøres f.eks. ved å opprette nye Word-dokumenter i et dokumentbibliotek på sidene, skriving av annonser eller poster i det integrerte forumet. De to siste gjøres i et grensesnitt for tekstinput som inneholder standard verktøy for

skrivning og redigering av tekst.

Ved oppretting av interne websider er det ingen separasjon mellom innhold og presentasjon, noe som vanskeliggjør publisering til forskjellige formater. Innholdet lagres automatisk i en .NET content file med ekstensjon .aspx. Internt i grensesnittet har man heller ikke muligheten for å velge presentasjonsformat, men man kan forøvrig skrive tekstinnhold i rent HTML-format, for så å redigere stilarket for dette i et program som er kompatibelt med WSS.

Flerbruker-skriving, dvs. at flere brukere samtidig redigerer innhold, er ivaretatt ved at flere brukere kan være innlogget samtidig, og redigere innhold på portalen. Brukere kan også redigere på samme fil, men da med fare for at man overskriver hverandres arbeid. For å hindre dette har man *check in* og *check out* funksjoner som sikrer at brukere ikke kan redigere samtidig på filer.

Som en nettstedseier kan man lage nye innholdstyper for nettstedet, dvs. en gjenbrukbar gruppe med innstillinger for en kategori av innhold. Man kan så legge disse innholdstypene til lister og biblioteker på nettstedet eller i underliggende nettsteder. Innholdstyper kan gjenbrukes i multiple dokumentbiblioteker. Metadata kan legges til innholdstyper.

Krysslenker mellom sider ser ut til å være stabile mot omstrukturering av nettstedet, men manuelt opprettede lenker må ettergås og endres manuelt.

Det er ikke nødvendig for tekstskapere å kunne HTML eller annen teknisk kunnskap.

Løsningen er lettbrukelig ved at man f.eks. enkelt kan opprette dokumenter direkte i løsningen (i Microsoft Word format), og websider i .aspx-format. Noe uvanlig navigasjon og meny plasseringer, men er effektiv når man vender seg til det. Det kan være noe vanskelig å forstå konseptet med lister og innholdstyper.

Innholdsbehandling

Det er valgfritt å versjonstilknytte filer. WSS tilbyr versjoner på listeelementer, filer og dokumenter. Metadata blir lagret for hver versjon av et dokument. Man kan dessuten spesifisere om det skal være en større eller mindre versjonsutgave av et dokument.

Hvis organisasjonen bestemmer seg for å gå bort fra Microsoft-systemer som Microsoft Office, så kan dette by på problemer med integrasjon med SharePoint Services. Man kan implementere spesifikke forretningsprosesser på dokumenter og enheter på en side, for å la folk samarbeide og

konsentrere seg om sin oppgave. I tillegg kan brukere få tildelt forskjellige rettighetsnivåer, som kan organisere innholdsadministrasjonen bedre.

Systemet tilbyr autentisering for å bedre sikkerheten. Brukere med konto kan autentiseres med autentiseringsmetoden brukt på webserveren (kun IIS). Brukere kan videre tildeles rettigheter på elementer på nettsidene.

WSS er spesielt utviklet for integrasjon med Microsoft Office, og er tett integrert med dette. Det tilbys offline synkronisering med Microsoft Office Outlook 2007 for å behandle dokumentbibliotek, lister, kalendere kontakter o.l. Man kan også lagre et Word-dokument fra Word direkte inn i dokumentbibliotek på webportalen.

Systemet har funksjoner for bruksstatistikk og rapportering. Videre kan alarmer informere brukere når nye kritiske elementer har kommet opp på websida.

Publisering

Visning av innhold kontrolleres av stilark, i tillegg til sidemaler og nettstedsmaler. Sluttresultatet av visningen er altså bestemt av disse tre.

Systemet støtter en prosess for kontinuerlig forbedring av brukergrensesnitt-designet ved at sidemaler og nettstedsmaler kan tilpasses og designes i et program som er kompatibelt med SharePoint Services. Stilark kan også endres i et slikt program. Masterpages kan editeres i et program som er kompatibelt med SharePoint Services (f.eks. SharePoint Designer). Her kan layout, menyer og logo tilpasses for flere sider av gangen.

Systemet støtter bl.a. publisering til .NET content file-format, Word-format og Excel-format.

Personalisering tilfredsstilles ved at forskjellige brukere kan ha tilgang til forskjellig informasjon. Bruk av innholdstyper gjør at instillinger for en innholdstype avgjør dens organisering og oppførsel.

Bruksstatistikk kan logges ved at systemet kan vise antallet treff på en side, antallet unike brukere, informasjon om browser og OS, og refererende domener og ULR'er. Dette kan gi en indikasjon på hvilke sider som er mest populære.

Presentasjon

Hvorvidt de publiserte sidene oppfyller standarder for lettbrukelighet, læringskurve og effektivitet

kommer noe an på den som skaper sidene. Men visse brukergrensesnitt-elementer er uforanderlige. Ut fra hva dette studiet har rukket å teste, er enkle funksjoner lette å forstå og lære seg, mens mer avanserte funksjoner og større kompleksitet gjør straks brukeren mer forvirret.

Når det gjelder aksesserbarhet for funksjonshemmede gjennom støtteverktøy som skjermlesere og lignende, er det forsøkt lagt til rette for dette gjennom Microsoft Active Accessibility (MSAA). MSAA tilgjengeliggjør innhold for folk med funksjonshemninger gjennom bruk av støtteverktøy som skjermlesere og lignende. Ikke alle kontroller i SharePoint Services er kompatible med disse. Man kan aktivere "More Accessible Mode" for å få systemet til å tegne opp kontroller som tilsvarende standard HTML-kontroller som er mer tilgjengelige for skjermlesere. Den dårlige nyheten er at disse funksjonene bare gjelder for lisensierte brukere i USA. Brukere utenfor USA må undersøke med sin forhandler hvorvidt disse funksjonene er tilgjengelige i deres område.

De publiserte sidene i WSS har dårlig kompatibilitet med andre nettlesere enn Microsoft Internet Explorer. Forskjeller mellom IE (7) og Firefox (2.0 Bokmål): IE viser thumbnail previews av bilder i bildegalleri, noe Firefox ikke gjør. Når et Word-dokument åpnes fra dokumentbiblioteket, spør IE om det skal åpnes editierbar eller skrivebeskyttet versjon, noe Firefox ikke gjør. Menyfunksjonen "New – New document" i dokumentbiblioteket lager et nytt, tomt Word-dokument fra IE-nettleseren. I Firefox kommer bare en beskjed om at en SharePoint Services-kompatibel applikasjon ikke er installert, og at IE 6.0 eller høyere kreves for denne funksjonen. Firefox-brukere må istedet laste opp et nytt dokument. Under menyen "Upload" fra dokumentbiblioteket tilbyr Firefox bare alternativet "Upload document", mens IE også tilbyr "Upload Multiple Documents". Videre under menyen "Actions" i dokumentbiblioteket er det store forskjeller i antallet og innholdet i menyalternativene, med Firefox som den som tilbyr færreste menyfunksjoner. Disse eksemplene er nok for å stadfeste at funksjonaliteten ikke er optimal i Firefox (2.0 Bokmål).

For Opera (9.27) er situasjonen enda verre. Der fungerer nesten ingen av de ovennevnte funksjonene. Når det gjelder støtte for web-deler i SharePoint Services er det kun IE som tilbyr den optimale visuelle og funksjonelle ytelsen. Andre nettlesere har begrenset ytelse eller fungerer ikke i det hele tatt hva web-deler angår.

Det kreves ikke klientside-teknologier som Flash eller Java. Det er usikkert om JavaScript kreves.

Sidestørrelsene forventes ikke å by på særlig høye lastetider for brukerne, gitt en ADSL-oppkobling og normal nettsurfing.

Det er usikkert om sidene oppfyller nåværende HTML-spesifikasjon, men WSS bygger i stor grad på ASP.NET 2.0, som oppfyller XHTML-spesifikasjonen veldig bra. Men WSS' kompatibilitet med andre nettlesere og OS enn IE og Windows er likevel ikke veldig høy.

Når det gjelder navigasjonen, så er den konsistent, men kan virke noe overflødig i enkelte tilfeller. Ett eksempel er at det er 3 linker til framsida som standard. Ellers er navigasjonen lettforståelig, men kan virke noe kompleks for datanoviser. Noe av navigasjonen er mulig å tilpasse. Det gjelder Quick Launch (venstremenyen) og Top Link Bar (toppmenyen). Andre navigasjonselementer er ikke mulig å tilpasse.

For metadata og indeksering så kan man lage dokumentkolonner med Dublin Core metadatasett.

Kontrakter og forretning

Treningsmateriale for CMS'et finnes bl.a. på <http://office.microsoft.com/en-us/training/CR102146081033.aspx>. I tillegg tilbys det kurs, ifølge SVT-Drifts personale.

CMS'et har integrert dokumentasjon, for brukere og administratorer. Det finnes også online forum av nytte for utviklere og administratorer, bl.a. her: <http://frontpages-web-hosting.net/forums/>.

For vedlikeholdsavtaler så kan man tegne en supportavtale med Microsoft, ifølge SVT-Drifts personale.

Påkrevde maskinvare-, programvare- og operativsystemressurser for systemet er som følger:

- OS: Windows Server 2003 SP1 eller Windows Server 2003 x64 eller Windows Small Business Server.
- Ressurser: .NET Framework 3.0, Internet Information Services (IIS) 6.0 with common files, SMTP Service, WWW Service.
- Maskinvare: Server med minimum 2,5 Ghz prosessor, min. 1 GB RAM, min. 2 GB harddiskplass for installasjon og min. 5 GB harddiskplass for data.

For en organisasjon vil systemet kreve ferdigheter innen Microsoft-produkter, webutviklingsverktøy kompatible med SharePoint Services 3.0, generell datakompetanse for administrasjon av innhold, og ressurser for planlegging og koordinering av arbeid med verktøyet. For mer inngående redigering kan det være aktuelt med kompetanse innen ASP.NET, blant annet for å redigere web-deler.

Det kreves også kompetanse innen serverspesifikk drift og konfigurering av systemet (dette vil en typisk driftsavdeling ta seg av).

WSS 3.0 er gratis, dvs. inkludert med Windows server 2003/2008. Pris avhenger videre av avtaletype. NTNU har Campus eller selectavtale, og får det til skolepris [33].

SharePoint Services 3.0s oppnår skalerbarhet gjennom installasjon av systemet i et server-farm-miljø. Tusener av websider kan da gjøres tilgjengelig for titusener av brukere¹¹. Systemkravene for en slik installasjon er som med single box, i tillegg til at man trenger SQL Server 2000 SP3 (eller senere) eller SQL 2005. Systemet må ha dual prosessorer på 2.5 Ghz og minimum 2 GB RAM.

En omfattende undersøkelse rundt eksisterende maskin- eller programvare som systemet må grense mot, eller kjøre på, f.eks. operativsystemer, databaser eller webservere er ikke foretatt. Inntil videre må det forventes at systemet må kunne grense knirkefritt mot maskiner med Windows-, Mac- og Linux- OS (klienter), Windows- eller Linux-tjenere, IIS eller Apache (eller annen webserver) webservere. Foruten dette vil systemet kjøre noenlunde uavhengig av eksisterende systemer innenfor PLU. Systemet vil ha trøbbel med å grense mot Mac- og Linux-klienter hvis disse f.eks. bruker andre nettlesere enn IE.

Referanser til bruk av systemet i organisasjoner som ligner noe på PLU i størrelse er oppført nedenfor (vel og merke vellykkede implementeringer).

- EUC Nord Fredrikshavn er det største forretnings- og tekniske universitetet i det nordlige Danmark, med 500 ansatte og over 3000 studenter fordelt på fire stedslokasjoner. Universitetet ønsket at administratorer i universitetsdepartementer selv skulle kunne behandle innhold de produserte og var ansvarlige for, ved hjelp av kjente skrivebordsverktøy, istedet for at IT-ansatte hadde ansvaret for dette. De bestemte seg for å implementere Microsoft Office SharePoint Portal Server 2003 og Microsoft WSS, og fikk en tredjepart til å utvikle og implementere løsningen. Websiden ble dog bygd bl.a. ved bruk av disse teknologiene: Microsoft Visual Basic, .NET 2003 development system, stilark, Microsoft Office Professional 2003, SharePoint Portal Server og Microsoft SQL Server 2000. Resultatet ble bl.a. at menneskelige ressurser ble mer effektivt utnyttet, kurskataloger mer oppdatert og at den nye løsningen var mer skalerbar (Microsoft Learning Gateway, www.mlg.no).
- Kingston University, et universitet spredd over flere steder i sør-vest London, implementerte

¹¹ http://www.megasolutions.net/Sharepoint/Microsoft_Windows_SharePoint_Services_3_0.aspx

en løsning som inkluderte bl.a. SharePoint Portal Server 2003, Microsoft .NET-rammeverk, Microsoft ASP.NET og Microsoft Windows SharePoint Services. Denne løsningen ble valgt fordi den enkelt kunne integreres med universitetets eksisterende IT-ressurser, inkludert Blackboard. Fordelene rapportert på www.mlg.no innebar bl.a. at driftskostnadene ble holdt på et minimumsnivå, man fikk forfatter- og versjonskontroll for en enkelt kilde av nøkkeldokumentasjonen, og man fikk et komplett samhandlingsmiljø.

Under evalueres SharePoint Services innenfor et utvalg sentrale nettbaserte funksjoner og ut fra krav som kan være aktuelle for plu-veven.

Artikler

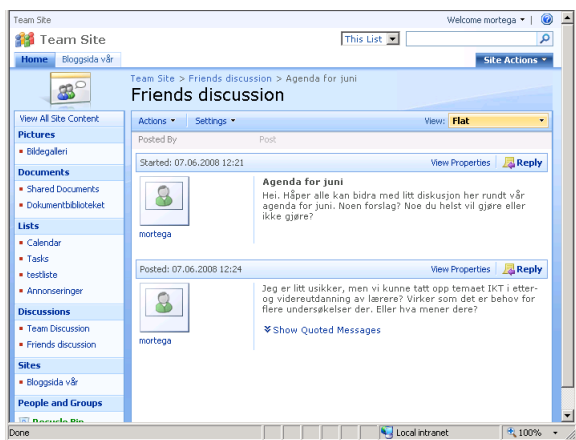
Artikler er utdypet i kapittel 4.2.1. Ut fra denne beskrivelsen av funksjonen er det som ligner mest på artikler og rent innhold i SharePoint Services en Basic Webpage. Dette er en statisk webside som lages ved å redigere innholdet i den integrerte tekst- og HTML-editoren. Man kan også redigere innholdet mer detaljert (og f.eks. legge på CSS-stil) i webredigeringsprogrammet SharePoint Services hvis man har dette installert.

I tillegg til de statiske websidene som lages i funksjonen Basic Webpage, er det mulig å lage dynamiske websider i funksjonen Web Part Page. På en slik side kan man inkludere webdeler som dynamisk kan vise innhold fra bl.a. lister og tidsbestemt innhold. Man kan velge forhåndsbestemte layouts for fordeling av webdelene på siden, eller lage sin egen layout. En Web Part Page kan, i likhet med en Basic Webpage, bare opprettes i et dokumentbibliotek, og bare aksesseres derfra. Det betyr at brukeren må navigere til et dokumentbibliotek før han/hun kan se siden. Dette begrenser noe gjenbruket av slikt innhold. Innholdet lagres i .aspx-filer, som trolig kan gjenbrukes i andre .NET-applikasjoner, men ikke i generelle websider uten bruk av .NET-rammeverket.

Forum

SharePoint Services har et integrert forum som er innebygget i løsningen. Når man lager et nytt forum, blir det automatisk lagt under mappen "Discussions". Man kan ikke velge hvor forumet skal være eller hvor det skal presenteres i løsningen som standard. Man kan endre en rekke innstillinger i forumet, bl.a. hvorvidt postene må godkjennes før de postes, tilgangsrettigheter for brukergrupper, vedleggsinnstillinger, hvorvidt forumet skal være søkbart etc. Man kan også legge til en arbeidsflyt til diskusjonen. Man kan velge flat visning av postene, eller trådet visning. Figurene (8 og 9)

nedenfor viser forumet i WSS i henholdsvis flat og trådet visning.



Figur 8: Forumet i WSS med flat visning.



Figur 9: Forumet i WSS med trådet visning.

Man bruker den integrerte teksteditoren for å skrive forumpostene. Her kan man også sette inn bilder og mediafiler, og endre fontvalg og størrelse samt redigere HTML-koden m.m.

Blogg

Det er mulig å opprette blogger i systemet. Man må lage disse som et underliggende nettsted (site), fordi bloggfunksjonen er en del av en såkalt forhåndsdefinert side for samarbeid, en forhåndslaget mal for blogging. Men en blogg kan også opprettes og tilpasses i et webredigeringsprogram kompatibel med WSS, som f.eks. SharePoint Designer.

Bloggposter kan kategoriseres, f.eks. som “hobby” eller “spesielt prosjekt” eller lignende, alt etter hva brukeren ønsker. Det er mulig å poste til en blogg gjennom en epostmelding som sendes til systemet. Som standard kreves det godkjenning før en bloggpost kan publiseres.

IM

Instant Messaging (IM) eller chat tilbys ikke i WSS.

RSS

Nyhter, blogger og annet innhold kan gjøres tilgjengelig for RSS-abonnenter. En nettstedseier kan aktivere RSS for følgende innhold:

- Nettstedssamlinger (site collections)
- Toppnivånettsteder og undernetsteder
- Webapplikasjoner
- Lister

Lister kan være alt fra dokumentbibliotek, bildebibliotek, form-bibliotek og annonseringer til kalenderlister, blogger, undersøkelser og diskusjonsfora.

Wiki

WSS tilbyr å lage et wiki-bibliotek, som kan inneholde bilder, tabeller, hyperlenker og interne lenker. Man bruker den integrerte teksteditoren for å redigere innholdet i wikibiblioteket.

Innholdsorganisering og -behandling

WSS benytter innholdstyper (content types) for å organisere og behandle innhold på en konsekvent måte på nettstedet (eller samlingen av nettsteder). En innholdstype er en gruppe av gjenbrukbare innstillinger som beskriver den delte oppførselen for en spesifikk type innhold. Innholdstyper kan defineres for et hvilket som helst type element, inkludert dokumenter, listeelementer og mapper. Bl.a. definerer innholdstypen hvilke typer metadata som kan tilskrives et element. Dokument-maler ved opprettelse av nye dokumenter av en dokument-innholdstype kan defineres. De tilgjengelige arbeidsflytene for en innholdstype kan også defineres. Innholdstypene blir selv organisert i et hierarki, noe som tillater arv av egenskaper mellom innholdstyper.

WSS 3.0 bruker arbeidsflyter for å hjelpe folk å samhandle på dokumenter og å behandle prosjektoppgaver ved å implementere spesifikke forretningsprosesser på dokumenter og elementer i et nettsted. Tanken er å la folk som arbeider innenfor disse prosessene konsentrere seg om arbeidet i stedet for å arbeide med arbeidsflyten. Innen WSS er en arbeidsflyt definert som den automatiske forflytningen av dokumenter og elementer gjennom en spesifikk sekvens av handlinger eller oppgaver som er relatert til en forretningsprosess.

I GUI'en organiseres innholdet i mapper, lister og biblioteker, der f.eks. dokumenter og bilder befinner seg i henholdsvis dokumentbiblioteket og bildebiblioteket, kalendere, oppgaver og annonseringer befinner seg i sine egne lister, og diskusjoner i egne mapper osv. Ved å trykke "View All Site Content" fra framsida, får man en oversikt over alt innholdet på nettstedet man befinner seg på. Her kan man se at alt innholdet er gruppert og kategorisert etter en forhåndsdefinert oppdeling.

Man får presentert overskrifter med titlene “Lists”, “Document Libraries”, “Discussion Boards” o.l., der de forskjellige dokumentbibliotek, lister og diskusjonsfora o.l. grupperes. Oppdelingen er fast, og det er uklart om denne oppdelingen kan endres på eller tilpasses. På nettsidas framside kan man la Quick Launch-menyen (hovedmenyen på venstre side) ha menylenker til de forskjellige dokumentbibliotekene, listene og diskusjonsforaene etc.

Basert på denne evalueringa av Windows SharePoint Services 3.0 så kan vi kort oppsummert konkludere at løsningen er først og fremst en samhandlingsportal for å la et team av mennesker samarbeide med å løse spesifikke oppgaver innenfor spesifikke forretningsprosesser. Det er mindre et webutviklingsverktøy, og har mindre funksjonalitet hva gjelder fri utvikling og tilpasning av en webportal. For en mer omfattende tilpasning kreves et webredigeringsprogram som f.eks. SharePoint Designer.

5.1.2 Joomla!

Joomla! er et open source prosjekt som springer ut fra det populære Mambo CMS. Årsaken til at Joomla!-utviklerne skilte lag med Mambo-utviklerne i 2005 og laget sitt eget CMS skyldes uenigheter i retningen CMS-prosjektet (Mambo) skulle ta [14]. Noen egenskaper ved Joomla CMS er listet opp under:

- Fri kildekode
- Et stort og ivrig samfunn av brukere og utviklere
- Enkelt arbeidsflyt-system
- Publiseringssystem for innhold
- Filbehandler for å laste opp og administrere filer
- Multispråkfunksjon
- Enkel installasjon av utvidelser
- Bruksstatistikk
- Rangerings- og stemmesystem
- WYSIWYG-editor
- Innhold i RSS-format
- Kraftfullt malsystem (HTML, CSS, PHP)

- Hierarkiske brukergrupper
- Administrasjonsgrensesnitt separert fra websidene

Eksempler på utvidelser er:

- Forum
- Nedlastningsbehandling
- Bildegalleri
- Dokumentbehandling

5.1.2.1 Evalueringen av Joomla!

En evaluering av Joomla! etter James Robertsons metode blir her foretatt. Se kap. 2.2.3 for en bakgrunn rundt Robertsons evalueringsmetode, og Appendix 10.3 for selve evalueringsskjemaet. Det er viktig å merke seg at evalueringen ikke inkluderer alle mulige tilleggsutvidelser for systemet, som kan eller kan ikke ha innvirkning på resultatene i evalueringen.

Evalueringen av Joomla! etter Robertsons kriterier grupperes, som i evalueringen av de andre systemene, i følgende bolker:

- Innholdsskaping
- Innholdsbehandling
- Publisering
- Presentasjon
- Kontrakter og forretning

Innholdsskaping

Det integrerte forfattermiljøet har en kraftfull WYSIWYG-editor (TinyMCE) for innskriving og redigering av tekst og innhold. I tillegg til vanlige tekstredigeringsfunksjoner har den bl.a. redigering av HTML-kildekode, fullskjermvisning og redigering av CSS.

For separering av innhold og presentasjon er det mulig å anvende stiler på tekstinhold, direkte i den integrerte teksteditoren. Tekstlig innhold kan også konverteres til pdf eller sendes som epost som default. En funksjon for visning av tilpasset utskrift er tilgjengelig. Det er mulig å skrive tekst i rent HTML-format (for så å anvende stilark).

Flere brukere kan redigere samme innhold på samme tid, men kan ikke se hverandres endringer. Den som lagrer sist vil få sine endringer beholdt. Ingen kjente automatiske låsemekanismer finnes i siste versjon.

Innhold som skrives inn, lagres som artikler i brukeropprettede kategorier og seksjoner, eller bare som statisk innhold. Alt dette kan brukes og anvendes nesten hvor som helst i portalen, eller flyttes på og innlemmes i andre seksjoner og kategorier.

Alt innhold kan tilknyttes metadata, som f.eks. beskrivelse, nøkkelord og forfatter.

Artikler og innhold kan lenkes til gjennom menyer. Det er mulig å flytte på menyene uten at lenkene til innholdet de peker mot, ødelegges. På den måten er systemet stabilt mot omstruktureringer. Lenker som lages integrert i innhold, og som lenker til andre artikler, må dog følges opp manuelt ved omstrukturering.

Forfattere av tekster trenger ikke å beherske HTML eller lignende teknisk kunnskap. Det er dessuten et brukervennlig grensesnitt med kjente GUI-elementer som menyer og ikoner.

Det er relativt lett å lage innhold. For å lage innhold velger man enten “Add New Article” eller “Article Manager” fra kontrollpanelet i portalens backend/ administratormeny. For “Add New Article” kommer man rett inn på en side for å skrive inn tittel, kategorisering og selve innholdet for en ny artikkel. Alt foregår i et relativt lettførståelig GUI. Bildeinnsetting var noe innviklet i kjerneutgaven av tidligere utgaver av Joomla, men dette er nå veldig enkelt i de nyeste versjonene. Fra “Article Manager” kan man filtrere visning av innhold etter seksjon, kategori og/eller tilstand. Man kan dessuten plassere en hvilken som helst artikkel på en hvilken som helst side.

Innholdsbehandling

Hver artikkel inneholder informasjon om:

- Artikkel ID
- Tilstand (f.eks. publisert)
- Antall revisjoner
- Tidspunkt elementet ble opprettet
- Siste tidspunkt elementet ble modifisert
- Forfatter

Dette kan imøtekomme krav til versjonskontroll og lovmessige ansvar, backup og katastrofehåndtering. I tillegg kan man spesifisere tilgangsnivå til artikler, samt tidspunkt for publisering, og tidspunkt for avslutning av publisering (fjerning fra websidas frontend). Artikler som er i bruk (men kan være enten publisert eller ikke) blir liggende i artikkellista under “Article Manager”. Eldre artikler, f.eks. utdaterte versjoner, kan legges i et arkiv for dette fra “Article Manager.”

Personer trenger ikke være lokalisert på samme sted for å skape innhold. Innholdsskaping kan fordeles på forskjellige folk med hver sine brukerrettigheter for skriving av innhold, redigering, godkjenning og publisering m.m. Siden alt foregår inne på selve portalen, er ikke arbeidsflyten som definert i Joomla spesielt sårbar for organisasjonsforandringer.

Brukere kan få tildelt forskjellige roller med forskjellige rettighetsnivå og muligheter på portalen (bl.a. registrert, forfatter, redaktør eller utgiver). Rettighetsnivå organiserer innholdsadministrasjonen på en bedre måte.

Sikkerhet ivaretas i Joomla bl.a. ved at innhold kan tilgangsbegrenses til kun enkelte registrerte brukergrupper (public, registered eller special). Ethvert skapt innhold (som artikler) får forfatterens (brukernavnets) stempel på seg. Sikkerhet ved opprettelse/godkjenning av nye brukere kan gjøres ved å la brukere registrere seg fra framsida, men må samtidig oppgi en epostadresse for å motta en aktiveringslink som må klikkes på for å aktivere brukerkontoen på portalen. Dette verifiserer at brukeren eksisterer, og at han/hun har en gyldig epostadresse. Det gir også brukeren mulighet til å lage sitt eget passord ved registrering i tillegg til å gi nettstedadministratorer bedre oversikt over aktiverte/ikke aktiverte kontoer [47].

Et søk etter "Joomla" på securitydot.net ga 40 resultater på sikkerhetssårbarheter (vulnerabilities) for dette CMS'et. Antall sikkerhetsutnyttelser (exploits) lå på 22, der de 10 øverste på lista var rangert som medium (av en aller annen grunn var det ikke mulig å få fram resten av lista). Til sammenligning ga samme søk på CMS'et PHP-Nuke (open source) 76 sårbarheter og 10 utnyttelser. Drupal (open source) fikk henholdvis 38 og 1. Man bør likevel ikke felle sin dom basert på disse resultatene, siden disse kan avhenge av populariteten til et CMS (det kan derfor være flere som finner svakheter ved det). Slike offentlige lister kan være fordelagtige for open source systemer, som er avhengige av “allmenhetens” bidrag til utvikling og testing.

For integrasjon med eksterne systemer har Joomla søtte for XML-RPC. I tillegg kan eksterne komponenter, som f.eks. forumet phpBB, integreres med websidene [14]. Siden Joomla skal

fungere med de fleste store nettleserne (som IE 6+, Firefox 1.5+, Safari og Opera), vil systemet fungere godt sammen med klienter som kjører disse nettleserne. Siden Joomla selv kan kjøre på de vanligste variantene av Windows, Linux og Apple Mac OSX (og mulig med flere plattformer), er det trolig at systemet også uten for mye problemer kan integreres med eksisterende systemer som kjører på disse plattformene, selv om dette ikke er undersøkt ytterligere i oppgaven.

Systemet rapporterer situasjoner som oppstår ved at beskjeder kommer opp når moduler og templates eller lignende har blitt installert, eller når innhold er lagret. Det er også mulig å aktivere debugging i systemkonfigurasjonen. Registrerte brukeres siste visitt på webportalen blir loggført. Innloggede brukere blir oppgitt i backend.

Publisering

CSS blir brukt i portalens valgte template, som også kontrollerer hvordan innhold skal presenteres. Det er mulig å selv redigere CSS-filene etter ønske.

Systemet tilbyr sidelayouter i form av om man vil presentere kun én artikkel eller ett innholdselement på siden, eller flere, samt hvilken layout de skal presenteres i (listeform, full artikkel eller blogg-layout etc.).

Kontinuerlig forbedring av brukergrensesnitt-design kan støttes gjennom endring av CSS-filer og HTML-filer for endring av portalens design. Dessuten kommer open source miljøet ofte med nyutviklede grensesnittelementer som f.eks. nye menysystemer, men oppdateringer/nye versjoner av systemkjernen kan skape problemer med kompatibilitet med litt eldre grensesnittelementer og andre tilleggskomponenter og -moduler.

Systemet gir mulighet til å presentere innhold til:

- HTML (Web): De opprinnelige innholdselementene blir presentert i HTML (med css for utforming og stil) på portalens sider
- PDF: Innhold som vises på websidene kan vises med ikoner brukere kan klikke for å få opp artikkelen i pdf-format eller sende artikkelen på epost.
- Trykte materialer: Innhold som presenteres på websidene kan vises med ikoner brukere kan klikke for å få opp artikkelen i utskriftsformat, som kan lett skrives ut

Forskjellig innhold kan presenteres eller gjøres tilgjengelig for en bruker ut fra nivået av tilgangsrettigheter brukeren har. Innholdselementer kan settes til “offentlig”, “registrert” eller

“spesiell”, som hver for seg avgjør hvorvidt en bruker kan se eller få tilgang til innholdet.

Bruksstatistikk kan logges ved at sidetreff etter dato lagres. Dette gjør det mulig å monitorere mest populære sider og daglig bruk. Søkemotorbruk på nettsidene kan også logges.

Presentasjon

Hva gjelder brukbarhet, læringskurve og effektivitet, kan det trekkes noe for litt uoversiktlighet rundt hva som er menyer og moduler, og hvor man skal gå inn for å redigere hva. Hvor skal man f.eks. redigere tittelen for hovedmenyen? Dette gjøres ikke som man kanskje hadde trodd fra “Menu Manager”, men fra “Modules”-panelet, der man må redigere tittelen på hovedmeny-modulen. Det tar litt tid å komme inn i denne tankegangen, men det er konsekvent og logisk når man først har forstått det. Konklusjon: en litt bratt og kort læringskurve til å begynne med, men effektivt når man først har kommet over denne kneika.

Hva angår tilgjengelighet for funksjonshemmede og oppfyllelse av standarder for dette, så ligger alt til rette for at front end oppfyller WCAG 1.0 Priority 2 og krav fra Section 508¹². Men det er fremdeles opp til template-designere og innholdsbehandlere å sørge for at det de skaper også oppfyller kravene. Oppfyllelse av WCAG for backend (administrasjonsområdet) er planlagt for 2.x-versjoner av systemet. Et eksempel på en “web accessible” nettside laget i Joomla er www.3tc4u.co.uk.

Joomla er regulært testet og brukt i de fleste store nettlesere, som IE 6+, Firefox 1.5+, Safari og Opera. Systemet kan også fungere i andre nettlesere¹³. Likevel ble det registrert problemer med den interne editoren ved bruk i Opera 9.5 (tekstinnhold kan bare skrives inn i HTML-kildekoden). Brukergruppen for PLU-veven forventes ikke å gå utenfor disse nettleserne.

Ingen spesielle klient-side teknologier trengs for å se innhold på webportalen. Enkelte tilleggskomponenter kan ta i bruk andre teknologier, som f.eks. Flash, og da må klienten tilsvarende ha installert Flash.

Sidestørrelsene på websidene i systemet gir en akseptabel nedlastningstid for normal nettsurfing ved f.eks. en ADSL-oppkobling.

Angående gyldig HTML og kompatibilitet på tvers av nettlesere så brukte Joomla 1.5.0 fremdeles XHTML tabell-layout, en programmeringsmåte som gir problemer for nettsidenes tilgjengelighet

¹² <http://help.joomla.org/content/view/805/60/>

¹³ <http://help.joomla.org/content/view/1938/310/>

etter standarder som gir god tilgjengelighet. Nettstedsmaler med tabellfri kode skal være like om hjørnet, i følge [14], noe som eraserer vekk dette problemet. Ellers så brukes det valid XHTML, men det er opp til implementeringspersonell å påse at denne er godt kodet (om man vil gå lenger i tilgjengelighet for funksjonshemmede etc.).

Navigasjonen for backend er konsistent. Man har alltid en tilstedeværende toppmeny som inneholder snarveier til alle sidens funksjonsområder. De samme ikonene er brukt for interaksjon med funksjonalitet og innhold over hele nettstedets backend, og de befinner seg alltid på samme sted. For frontend kommer dette an på hvordan man designer sin webside. Joomla legger til rette for hovedmeny, toppmeny, brukermeny (når en bruker er innlogget), lenker i artikler og "les mer"-lenker etc.

Hver artikkel kan tillegges metadata med beskrivelse, nøkkelord og forfatter. I tillegg kan nøkkelord for robot-program som søker gjennom web'en for katalogisering, angis. Det finnes en nedlastbar ekstensjon som kan gi metadata på Dublin Core-format.

Kontrakter og forretning

Joomlas websider har treningsmateriale og "Beginners guide", bl.a. her: <http://docs.joomla.org/Beginners>. Joomlas offisielle wiki-dokumentasjon finnes her: http://docs.joomla.org/Main_Page. Nettside for utviklere i Joomla: <http://developer.joomla.org/>.

Joomla! utgis under GPL-lisensen og har ingen garanti (<http://www.serr.biz/serrbiz-gpl-joomla.html>)

Konsulent- eller webutviklingssselskap tar eventuelt på seg oppgaven med støtteforetak og oppgraderingsprosesser. En undersøkelse av hvilke slike tredjeparter som eventuelt tilbyr slike tjenester er utenfor forskningsområdet for denne rapporten, men Joomla-kurs tilbys f.eks. her: <http://www.exentra.no/joomlakurs/22.html>.

Joomla! har blitt vellykket testet på mange varianter av de tre vanligste OS (Windows, Linux og Apple Mac OSX). Systemet kan fungere også på andre plattformer, men dette er mindre dokumentert¹⁴.

Ferdigheter påkrevd for PLU for bruk av Joomla er generell datakompetanse for administrasjon av innhold, og ressurser for planlegging og koordinering av arbeid med verktøyet. Det kreves

¹⁴ <http://help.joomla.org/content/view/1938/310/>

kunnskaper innen CSS og HTML om man ønsker å programmere sine egne utseende-maler. Ønsker man å programmere sine egne utvidelser (plugins eller komponenter) bør man i tillegg kunne PHP. Men det finnes en rekke ferdige utseende-maler på markedet, i tillegg til en rekke ferdige utvidelser, så man klarer seg også uten den tekniske kompetansen, dog på bekostning av flere tilpasningsmuligheter.

Det vil også kreves kompetanse innen serverspesifikk drift og konfigurering av systemet (dette kan eventuelt en driftsavdeling ta seg av). Forøvrig er systemet lett å installere.

Joomla er gratis, men kostnader beløper når det gjelder serverressurser og -vedlikehold, noe som blir bestemt av den organisatoriske rammen rundt systemet.

Om Joomla's skalerbarhet er det vanskelig å finne noen konkret informasjon.

Når det gjelder eksisterende maskin- eller programvare som systemet må grense mot eller kjøre på, f.eks. operativsystemer, databaser eller webservere, er en omfattende undersøkelse rundt dette ikke foretatt. Inntil videre må det forventes at systemet må kunne grense knirkefritt mot maskiner med Windows-, Mac- og Linux-OS (klienter), Windows- eller Linux-tjenere, IIS eller Apache (eller annen webserver) webservere. Foruten dette vil systemet kjøre noenlunde uavhengig av eksisterende systemer innenfor PLU. Siden Joomla både kan kjøre på de 3 vanligste OS, såvel som at det støtter de vanligste nettleserne, så er man et stykke på vei når det gjelder integrering med eksisterende systemer.

Referanser til nettsteder som bruker Joomla!:

3tc: <http://www.3tc4u.co.uk/>

www.joomla.org

<http://www.vetclinics.net/>

<http://www.urth.tv/>

<http://www.unric.org/>

En utfyllende referanseliste på web finnes her: http://joomlapowersites.com/component/option,com_alphacontent/section,1/cat,21/Itemid,27/

Under evalueres Joomla innenfor et utvalg sentrale nettbaserte funksjoner og ut fra krav som kan være aktuelle for plu-veven.

Artikler

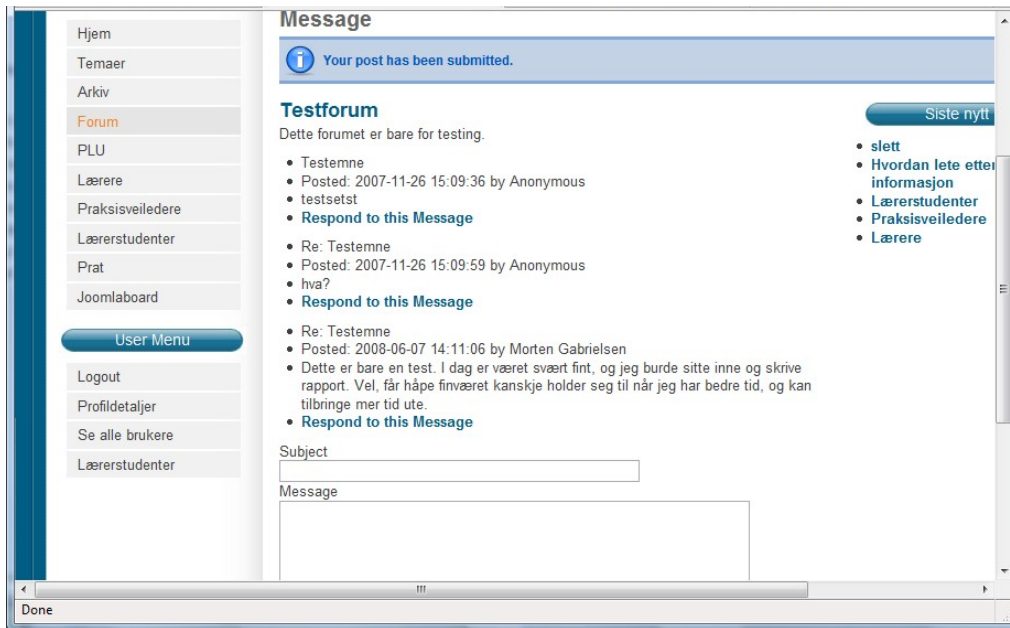
Artikler er en sentral funksjon i Joomla. Stort sett alt innhold på websider som lages med Joomla befinner seg i artikler. Artikler er innholdsbiten som kan opprettes, katalogiseres, endres og slettes alt etter hvilke rettigheter den registrerte brukeren har (eller administratoren). Det er frivillig om man vil katalogisere artikler. Artikler kan stort sett benyttes hvor som helst på webportalen, også på framsida. Man kan velge å presentere kun én artikkel på en side, eller flere, i f.eks. et blogg-format.

Man lager innholdet i en artikkel ved å benytte den integrerte tekst- og HTML-editoren, der man også kan sette inn mediefiler. Man kan forøvrig velge mellom 2-3 forskjellige editorer, noe som gjøres i profilen til brukeren som editerer, eller i de globale instillingene for nettstedet. Man kan la en artikkel være publisert eller ikke publisert, samt velge om den skal fremstå på nettstedets forside eller ei. Hvis innholdet i en artikkel er veldig langt, kan man sette inn side-delere (page breaks) forskjellige steder i artikkelinnholdet. Artikkelen vil da automatisk sideinndeles innholdet, og lenker for å navigere mellom sidene vil automatisk komme opp. En rekke parametre som kan tildeles artikkelen finnes. Disse er bl.a. om tittelen på artikkelen skal vises, om forfatter og dato artikkelen ble modifisert skal vises, o.l. Metadata for artikkelen kan også lages, og det kan spesifiseres i hvilket tidsrom artikkelen skal publiseres.

Forum

Forum er ikke en komponent som er med i standardinstallasjonen av Joomla, men det finnes en rekke nedlastningsbare forumkomponenter fra open source miljøet som man kan laste ned og installere enkelt i løsningen. Typer fora kan variere fra det kjente, avanserte og populære phpBB (<http://forum.joomla.org/viewforum.php?f=216>), til enklere forumkomponenter som f.eks. Simplest Forum (http://extensions.joomla.org/component/option,com_mtree/task,viewlink/link_id,3131/Itemid,35/). Tilgang til forumet kan så settes f.eks. fra et menyalternativ, ved at man lar dette menyalternativet peke mot forumkomponenten. Noen typer forum (som f.eks. Simplest Forum) bruker bare nettstedets eksisterende utseendemal (med tilhørende CSS) for presentasjon av forumpostene, mens andre forum (som f.eks. phpBB) ofte har sitt eget utseende, som må tilpasses for hånd (editere HTML og CSS). Det finnes også tredjepartsselskaper som kan utføre tjenesten å tilpasse forumets utseende til resten av portalens utseende-stil (et eksempel: http://www.pro-it-service.com/en/phpbb_layout_customization.html). De forskjellige forumkomponenter varierer i hvordan brukerne poster og leser postene. Noen fora har kun en flat struktur, mens andre har både flat struktur og trådet struktur. Figurene (10 og 11) nedenfor viser to forskjellige forumkomponenter

i Joomla.



Figur 10: Forumkomponenten Simplest Forum i Joomla, med sin flate struktur.



Figur 11: Bildet viser utdrag fra et phpBB-forum.

Blogg

Det finnes en rekke utvidelser med blogg-funksjonalitet som kan lastes ned og installeres (bla.

herfra: [http://extensions.joomla.org/index.php?](http://extensions.joomla.org/index.php?option=com_mtree&task=listcats&Itemid=35&cat_id=1759)

[option=com_mtree&task=listcats&Itemid=35&cat_id=1759](http://extensions.joomla.org/component/option,com_mtree/task,viewlink/link_id,3374/Itemid,35/)). En av disse komponentene har navnet mojoBlog (http://extensions.joomla.org/component/option,com_mtree/task,viewlink/link_id,3374/Itemid,35/), og er en blogg-komponent som er løst utviklet etter den populære open source bloggeplattformen WordPress. Egenskaper inkluderer integrert kommentering, plugin-støtte for avatarer, permalink støtte (se kap. 1.3), integert RSS med støtte for feedburning (www.feedburner.com) m.m. mojoBlog støtter i tillegg mange tredjeparts-plugins for WordPress.

IM

Funksjoner for chat og instant messaging kan skaffes ved å laste ned utvidelser med denne funksjonaliteten. Det finnes en rekke utvidelser med chat-funksjonalitet, med varierende kvalitet (bl.a. her: http://extensions.joomla.org/component/option,com_mtree/task,search/Itemid,35/searchword,chat/cat_id,0/).

En slik komponent er AVChat, en flash video chat komponent for Joomla som støtter audio- og videokommunikasjon i sanntid. Man kan også bruke text chat med private og offentlige rom, og man kan moderere chattesesjonen med mulighet for å sparke og utestenge brukere. Komponenten integrerer med portalens eksisterende brukerdatabase slik at disse brukernavnene kan benyttes i chat.

RSS

Som standard tilbyr Joomla RSS-feed kun for framsida (<http://www.pathos-seo.com/joomla-optimization/setting-up-joomla-rss-feed.html>). Som standard er den aktivert. Man kan velge mellom RSS-formatene RSS 2.0 og Atom 1.0.

En rekke nedlastbare komponenter for Joomla tilbyr mer RSS-funksjonalitet. F.eks. finnes det en bro mellom Joomla og open source komponenten Gallery 2, som tilbyr RSS. Jom Comment (http://extensions.joomla.org/component/option,com_mtree/task,viewlink/link_id,676/Itemid,35/) er en komponent for et Ajax-basert kommentar-system for webportalen, der RSS-mating for kommentarene støttes etc.

Wiki

Nedlastbare komponenter finnes, som enten tilbyr wiki-funksjonalitet eller integrerer andre wikisystem med Joomla. En komponent som integrerer MediaWiki (www.mediawiki.org) med Joomla er tilgjengelig for nedlastning. MediaWiki er gratis server-basert programvare som er lisensiert under GPL, og var opprinnelig skrevet for WikiPedia. MediaWiki har skalerbar programvare med mange egenskaper, og er basert på PHP og MySQL.

Innholdsorganisering- og behandling

Innholdet i Joomla organiseres i artikler, som kan være publisert, upublisert eller arkivert. Artiklene kan gjenbrukes stort sett hvor som helst på portalen man konstruerer. Videre kan hver enkelt artikkel kategoriseres i en kategori, som igjen plasseres i en seksjon. Man kan opprette, endre og slette kategorier og seksjoner. Innhold opprettes ved å opprette artikler. Disse kan også endres hvis brukeren har rettigheter til dette. Man har også et kontrollpanel for å behandle medieinnhold (Media Manager). Her kan man organisere, laste opp og navigere gjennom medieinnhold (som bilder, bannere, smilefjes o.l.) på serveren. Innhold knyttes til menyer for visning på portalen.

5.1.3 It's Learning

Kildematerialet i denne evalueringen er først og fremst tatt fra test av demo og It's Learnings intrerte hjelpesystem.

Evalueringen av LMS'et It's learning ble gjort ved å sammenligne systemet mot en kriterieliste framsatt av [2] for evaluering av LMS. Her gjenngås en oppsummering av denne evalueringen.

Brukbarhet

LMS'et har en konsekvent navigasjon som er lett å forstå. Navigasjonen ligner mye på et filsystem, med dets hierarki og mappestruktur. Systemet har også gjennomgående norsk språk. Dette gjør systemet lett å bruke, spesielt for de med kjennskap til filsystem. Det er også av den grunn lett å bli kjent med verktøyet. Det benyttes toppfaner for å skille nettstedets hovedområder fra hverandre, mens det er kontekstbestemte menyer eller trestrukturer på venstre side. Layouten minner mye om de mange websidene eller applikasjonene som finnes med dette formatet, og skulle derfor for de fleste brukerne være kjent.

Noen innholdselementer kan lett importeres og eksporteres fra verktøyet, f.eks. eportefølje,

eksamensbesvarelser, karakterbok, læringsplan og innholdspakker i IMS/ SCORM.

Standarder

Verktøyet følger standarder som AICC HACP, IMS/SCORM og XML-format (XML-eksport/-import av læreplanen). Lisensalternativene for systemet fokuserer på institusjonen; det er en lisens per bruker. I tillegg må det betales for leie av serverplass i IT Solutions' driftsmiljø.

Organisasjonsbehov

Systemet vil ikke fungere helt etter PLUs behov, siden de først og fremst har etterlyst et såkalt åpent system som ligner mer på eksterne websider. Her kommer It's learning noe til kort, selv om den har en rekke verktøy som tilfredsstiller PLUs behov.

Design

Navigasjonen er lettvinnt og konsekvent. I tillegg til toppfaner og trestrukturer er det en hendig brødsmulesti som gir brukeren opplysninger om hvor han/hun befinner seg på nettstedet.

Grensesnittet er enkelt å bruke siden det er konsekvent og lettforståelig. Teksten er godt synlig, og tekstinnholdet er kortfattet og beskrivende. Tekstformatet har god leselighet grunnet lettlesefont og passende størrelse for lesing ved middels oppløsning. Animasjon brukes ikke, unntatt som en mulighet i biblioteket dersom forfatter av siden inkluderer denne modulen på nettstedet. Da er det opp til innholdsskaper å påse at animasjon og andre elementer er tilpasset de visuelle designprinsippene. Lyd kan tas opp og spilles av i den interne editoren, samt spilles av når man mottar interne meldinger. Fargevalget gir god kontrast (vanligvis lys bakgrunn og mørk skriftfarge/forgrunnsfarge).

Grafikk brukes bl.a. i ikoner for menyer og lenkevalg. Nettstedets grafikk fremhever navigasjon og hjelper å gruppere innhold på en god måte, først og fremst med at det er enkelt, og det som brukes, overdøper ikke innholdet, og er dessuten konsekvent brukt.

Designet er fleksibelt ved at visning av kurs/fag kan tilpasses etter hvorvidt brukeren er en lærer, student eller en gjest. I tilleggsmodule "Bibliotek" kan man lage websider med tilpasset mal og grafikk for visning av fag. Man velger der mellom et utvalg maler, eller man kan redigere sin egen (i HTML og/eller CSS).

Kontekst

It's Learning kan legge til rette for interaksjonsbasert innhold som f.eks. diskusjon, konferanse, oppgaver, prosessorienterte dokumenter, tester, undersøkelser, forklaringssekvenser o.l.

Systemet tilbyr bl.a. egenskaper for å presentere og gruppere informasjon, lage nettbaserte prøver og kurs, forklaringssekvenser og diskusjon. Systemet har egenskaper passende til en utdanningskontekst, og det er kanskje ikke helt hva PLU ser etter i plu-veven.

Læringsstøtte

Det tilbys fri toveis-kommunikasjon i diskusjons- og konferansefunksjonen og det interne meldingssystemet. Systemet tilbyr en mappevurderingsfunksjon, som gir studenten mulighet for selv vurdering. Alle brukere i systemet tildeles en profil med ulike tilgangsrettigheter basert på profiltipe. Brukere kan lage sin egen eportefølge.

Støtte for instruktører

Systemet lar instruktører og lærere observere og administrere både kurset og de som lærer, gjennom forskjellige funksjoner. Instruktører kan koordinere kunnskapsdelingen i systemet ved at de kan bestemme *hva* som kan legges ut *når* på nettsidene. Systemet lar instruktører gi umiddelbare tilbakemeldinger om nødvendig. Dette kan gjøres direkte i grensesnittet, eller f.eks. via det interne meldingssystemet. IT Solutions står for backup av kursinnholdet, men fagadministratorene selv kan sikre noe innhold mot tap ved å eksportere innholdspakker, eksamensbesvarelser, læreplaner og karakterbøker ut fra systemet på en relativt lettvinnt måte.

Støtte for kurslevering

Multimedia kan legges inn i oppslag o.l. av administratorer. Epost støttes ikke, men brukerne kan få varsel på epost om nye meldinger i det interne meldingssystemet. Nyheter kan legges ut på en oppslagstavle. Chat er tilbydd gjennom konferansefunksjonen. Diskusjonsforum kan legges til. Konferanse-funksjonen har en integrert tavle man kan tegne på sammen i sanntid. Systemet støtter konferanse med webkamera, og opptak av lyd for utlegging i editoren.

Systemet tilbyr en integrert hjelpefunksjon. Som støtte til instruktøren/læreren gir systemet læreren mulighet til å overvåke studentprosjekter. Ellers kan instruktøren/læreren benytte seg av den integrerte hjelpen og community-funksjonen. V.h.j.a. community-funksjonen kan instruktøren aksessere og dele kurs og fag blant alle it's learning-brukere.

Hver bruker har sitt filområde som kan deles med andre brukere. Informasjon kan deles i f.eks. eporteføljen eller biblioteket. Fag kan deles med alle brukere i It's learning gjennom community-funksjonen.

Systemet tilbyr konferanser og diskusjonsforum som verktøy for samarbeid.

Serverkrav

IT Solutions drifter serveren, og systemet leveres som en tjeneste til kundene. It Solutions fronter med en oppetid på 99,9 %, og søker å holde løsningen oppe 24 timer i døgnet, 365 dager i året.

Bruk og tilgang

Løsningen gir lett tilgang til systemet via innlogging via web.

Profiler eller tilgangsrettigheter kan bestemme hvorvidt brukeren har tilgang til et kursverktøy eller ikke. Brukernes privilegier kan settes og endres etter behov. Brukerne kan gis lave eller høye privilegier.

Sikkerhet

Systemet støtter oppretting av profiler med brukernavn og passord. Brukernes passord er kryptert. Innlogging skjer via SSL. Systemet foretar tidsbehandling på innlogging og sesjon. For personvern kan man la systemet advare når Adobe Flash Player forsøker å bruke mikrofonen og webkameraet tilknyttet brukerens datamaskin i forbindelse med opptak av lyd og video.

Filer opprettet eller lastet opp i løsningen låses under redigering. Det er også mulig å gi forskjellige brukere forskjellige tilganger til filer og elementer (f.eks. lese-, skrive- eller ingen tilgang).

Det ser ikke ut til å være noen integrert virusbeskyttelse, men lagringsområdet drives av IT Solutions, og det er sannsynlig at de har sine egne antivirusiltak. Som en forhåndsregel (ikke fra IT Solutions men fra forfatter av denne rapporten) bør brukere likevel skanne filer fra ukjent opphav når de lastes ned lokalt på egen maskin.

5.1.4 WSS-utgave av plu-veven

I denne seksjonen gjennomgås systemet SVT-Drift hjalp PLU å implementere, og som ble brukt i

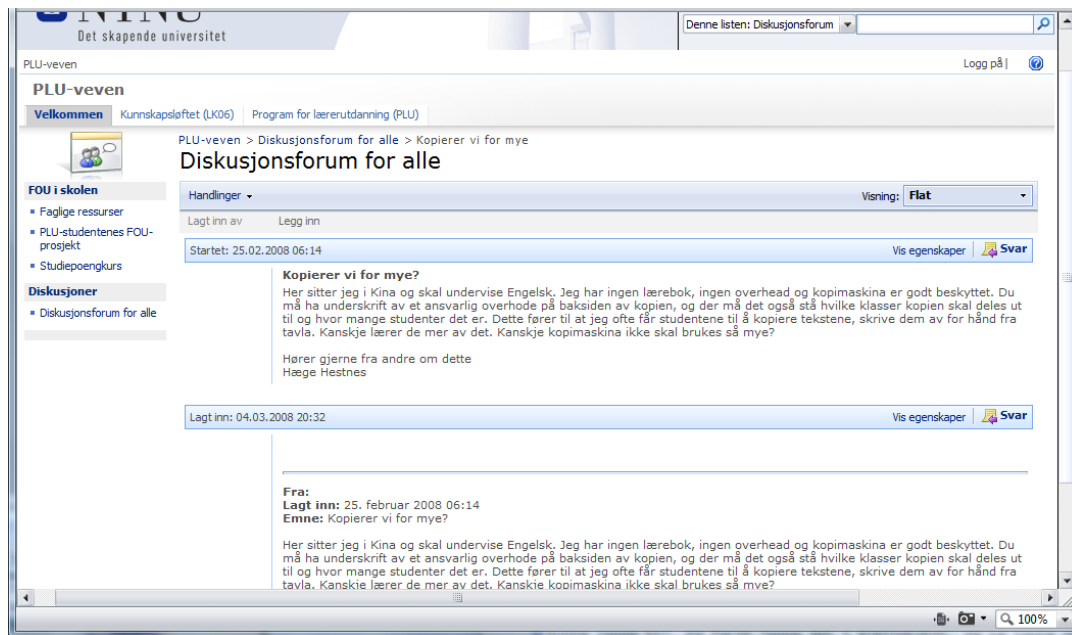
første omgang for plu-veven istedet for studentgruppas løsning. Erfaringer fra administratorer av systemet, brukernes erfaringer, og en evaluering av løsningen blir gjennomgått. En evaluering av systemet som ligger i bunn av denne utgaven av PLU-veven (Windows SharePoint Services 3.0) er gjennomgått i kap. 5.1.1.

PLU-veven ble offisielt tatt i bruk for første gang i januar 2008. Da ble det i første omgang brukt som et nettsted som supplerte et kurs om FoU. Systemet bygde på Windows SharePoint Services 3.0. Man hadde også installert Microsoft Office SharePoint Server 2007 på toppen av dette, men dette var lite brukt til PLU-veven. PLU valgte denne løsningen delvis fordi SVT-Drift hadde sagt seg villig til å bistå mer med en slik løsning enn hva de kunne bistå med for en løsning basert på studentgruppas system. Med WSS kunne de f.eks. hjelpe til med oppsett og drift av løsningen. Noe utseendetilpasning ble også gjort av personer ved SVT-Drift, bl.a. ved hjelp av SharePoint Designer.

En person ved PLU fikk ansvaret for å administrere PLU-veven, legge inn innhold og organisere dette.



Figur 12: Framside – Januar-08-utgave av PLU-veven bygd på WSS



Figur 13: Diskusjonsforum - Januar-08-utgave av PLU-veven bygd på WSS

5.1.4.1 Administrators erfaring

Administrators erfaring med plu-veven slik den var når den ble satt i drift i januar 2008 var blandet. Enkelte ting var lett og forståelig, mens andre ting var komplisert og uforståelig, ifølge hun selv. I en observasjonssesjon av administrators bruk av systemet, kom følgende fram som punkter administrator var misfornøyd med:

- Få lagt ut PowerPoint-filer på en annen måte enn som et filikon eller en liste over filikoner. Som hun sa det: "ikke som i It's Learning".
- Hun ønsket å "ta vekk navnet sitt i en kolonne for å få større bredde i en annen kolonne", men hun kunne ikke huske hvordan hun fikk det til (hun hadde gjort det en gang før).
- Forskjellen på "Nytt dokument" og "Last opp dokument" var vanskelig å forstå.
- Det var ikke WYSIWYG i editoren som editerte innhold for midtkolonnen på websida. Dette gjorde at hun måtte gå ut og inn av editoren flere ganger for å sjekke det reelle resultatet av posten på websida.
- Det var stor forvirring – også delvis for support på SVT-Drift – om hvordan tilgangsrettigheter på portalen skulle konfigureres slik at det ga det ønskelige resultat.

- Det var komplisert å få opprettet lenker fra undersider til nye undersider under disse igjen. En person fra SVT-Drift måtte foreløpig ordne slike ting for henne.

Av positive bemerkninger om systemet fra administrator var bl.a. dette:

- Systemet (WSS) hadde mange muligheter (som hun antok hun ville forstå mer av etterhvert)
- Det var lett å legge til kunngjøringer.

Ellers er det verdt å merke seg at observasjonen la vekt på at hun identifiserte *problemer* hun hadde med systemet og grensesnittet. Dette forklarer noe av den skjeve fordelingen av positive og negative bemerkninger. En slik observasjonssesjon kan gi en pekepinn på hvor bra et system og grensesnitt tilfredsstillende en brukers behov og hensikter, ved å identifisere og undersøke nærmere de problemer som oppstår under observasjon av brukerens bruk av systemet. I dette tilfellet var det endel problemer. Men alt var ikke nødvendigvis problemer med selve systemet, men kan ha sin årsak i manglende kunnskap om og opplæring i systemet, eller feil fordeling av oppgaver mellom driftere og administratorer. Under foretar jeg en drøfting av hva årsakene til de identifiserte problemene kan ha vært.

- Få lagt ut PowerPoint-filer på en annen måte enn som et filikon eller en liste over filikoner (se figur 14).

The screenshot shows the NTNU WSS interface. The main content area displays the course description for "Beskrivelse av kurset 'FoU i skolen'". The sidebar on the right, titled "Kursdokumenter", contains a table of documents:

Type	Navn	Endret
	Gruppearbeid som lærings situasjon (MBP)	04.04.2008 10:55
	Rammer for presentasjon av FoU -arbeid (MBP)	04.04.2008 10:56
	Teoriens rolle - analyse (MBP)	04.04.2008 10:54

Below the table, there is a section for "Interne lenker" with a link to "Kursoppgaver". A red arrow points from the text "PowerPoint-filer som ikoner" to the document icons in the table.

Figur 14: PowerPoint-filer vises som ikoner

Mulig årsak og løsning: Det ligger ingen funksjon i WSS som tillater PowerPoint-filer å presentere sitt innhold direkte i nettsidene. En løsning som kan virke mer tiltalende enn

visning av filikoner/overskrifter er å ta et skjermbilde av første slide i presentasjon, og legge dette ut som en klikkbar thumbnail-representasjon av presentasjonen sammen med en oversikt og en beskrivelse av presentasjonen. Når brukeren klikker bildet, kan han/hun velge å enten lagre presentasjonen eller åpne den.

- Hun ønsket å "ta vekk navnet sitt i en kolonne for å få større bredde i en annen kolonne", men hun kunne ikke huske hvordan hun fikk det til (hun hadde gjort det en gang før).

Mulig årsak og løsning: Årsaken til problemet kan være todelt: 1) Årsaken kan ligge i manglende opplæring av brukeren. Som hun sier så hadde hun "gjort det før", men husker ikke hvordan hun gjorde det. Dette viser at systemet innehar funksjonen som skal til. 2) Årsaken til problemet ligger i systemets grensesnitt. Siden systemet innehar funksjonen, men brukeren likevel sliter med å finne den, kan årsaken ligge i dårlig design av brukergrensesnittet. Brukergrensesnittet kan f.eks. være lite konsekvent eller lite logisk, noe som forvirrer brukeren.

- Forskjellen på "Nytt dokument" og "Last opp dokument" var vanskelig å forstå.

Mulig årsak og løsning: Årsaken til forvirringen kan først og fremst ligge i systemets grensesnitt. En bruker kan ikke øyeblikkelig forstå at funksjonen "Nytt dokument" betyr "nytt dokument i denne mappen", (selv om det står i beskrivelsen for funksjonen) fordi det er dette som skjer når man velger funksjonen. Det opprettes et nytt (Word-) dokument i den mappen (dokumentbibliotek) brukeren befinner seg i. Funksjonen "Last opp dokument" kan dessuten lett oppfattes som ekvivalent med å opprette et "nytt dokument i denne mappen," fordi det er delvis dette som skjer når funksjonen velges. Man henter et "nytt" dokument fra en plassering lokalt på maskinen, til dokumentbiblioteket på websida (serveren). Heldigvis kan forskjellen mellom disse to funksjonene lett og raskt prøves ut slik at brukeren lærer forskjellen. Et forslag til forbedring av grensesnittet for å unngå forvirringen kan være som følger: Menyalternativet "Nytt dokument" kan erstattes med "Forfatt et nytt dokument", eller "Åpne og rediger nytt dokument". Dette spesifiserer nærmere at man åpner et nytt dokument for redigering, og ikke har noe å gjøre med å laste opp.

- Det var ikke WYSIWYG i editoren som editerte innhold for midtkolonnen på websida. Dette gjorde at hun måtte gå ut og inn av editoren flere ganger for å sjekke det reelle resultatet av posten på websida.

Mulig årsak og løsning: Årsaken til problemet kan være todelt. 1) Systemets manglende WYSIWYG-funksjonalitet for editoren, kombinert med brukerens forventninger om

WYSIWYG-funksjonalitet for editoren. Løsning: Implementere WYSIWYG-funksjonalitet i editoren siden dette har blitt mer og mer standard for en rekke editorer i disse dager.

Tekstforfattere i dag er ofte vant med å bruke tekstbehandlingsprogrammer med denne funksjonaliteten. Et alternativ er å implementere en klarere separasjon mellom innhold (tekst) og presentasjon (stil). Brukeren kan f.eks. først få et grensesnitt for å skrive tekstinnholdet, for deretter få et separat grensesnitt for å sette stiler, fonter og utseende på innholdet, hvis sluttresultat på websida kan ses umiddelbart. Dette bidrar i det minste til å ta vekk brukerens forventning om WYSIWYG-funksjonalitet slik at forvirring ikke oppstår.

2) PLU-veven ble designet med flytende (fluid) bredde som layout. Dette gjør at tekst kan "flytte på seg" når brukeren endrer størrelsen (bredden) på nettleservinduet (se figur 15 og 16). F.eks. kan starten på nye tekstlinjer forflytte seg lenger opp eller lenger ned alt etter hvordan nettleservinduet endres i størrelse. Løsning: Designe PLU-veven med fast (fixed) bredde som layout.



Figur 16: Framsida i et bredt nettleservindu



Figur 15: Framsida i et smalt nettleservindu. Teksten flyter og endrer sin posisjon.

- Det var stor forvirring – også delvis for supportpersonell fra SVT-Drift – om hvordan tilgangsrettigheter på portalen skulle konfigureres slik at det ga det ønskelige resultat.

Mulig årsak og løsning: *Årsaken er foreløpig ikke kjent. Dette kan f.eks. ha noe med serverinnstillinger, brukerprofiler, konfigurering av brukergrupper, manglende kunnskap om hvordan tilgangsrettighetene fungerer, eller hvordan grensesnittet er designet å gjøre. Det er flere innganger til redigeringer og oversikter over tilgangsrettighetene i grensesnittet, noe som gjør det hele noe komplisert.*

- Det var komplisert å få opprettet lenker fra undersider til nye undersider under disse igjen. En person fra SVT-Drift måtte foreløpig ordne slike ting for henne.

Mulig årsak og løsning: Årsaken til dette er at beholderen lenkene ville lages i, var en såkalt web page (HTML-side), og innsetting av lenker derfra kan bare spesifiseres med navn på lenken og adressen lenken skal peke mot (man kan ikke velge i grensesnittet blant interne sider man ønsker å lenke mot). Er det ønskelig at lenken skal peke mot en annen intern side, eller et dokument, må denne lages og plasseres eksplisitt et sted på portalen, og adressen til denne må limes inn i lenkens adressefelt.

Dette er noe administrator etterhvert kan lære seg, men hvis det er få slik lenker, kan det være greit at personer fra drift foretar denne jobben, inntil administrator får mer erfaring med systemet.

Det kan også løses bedre ved å lage såkalte wikisider, der man kan lenke enkelt til andre interne sider. Usikkerheten her er hvorvidt andre brukere som i realiteten bare er ment å skulle lese innhold, blir i stand til å redigere i wiki'en.

Ellers kan man merke seg følgende kommentarer fra administrator om systemet:

- Hun ønsker å lage flere "typiske nettsider" (HTML-sider) i framtiden (i motsetning til å bare legge ut filer).

5.1.4.2 Test av WSS-utgave av PLU-veven

Det ble foretatt testing av WSS-utgaven av PLU-veven, i nettleserne Internet Explorer (IE) 7, Opera 9.5 og Firefox 2.0.0.14 (Firefox 1.5.0.10 på SuSE Linux) på Windows Vista, Windows XP og SuSE Linux. Systemet er ikke blitt testet i andre versjoner av henholdsvis IE, Firefox eller Opera, men det er grunn til å tro at resultatet ikke vil avvike særlig, siden disse versjonene stort sett er de nyeste i skrivende stund. Testen avdekket bl.a. disse kritikkverdige punktene:

- Den integrerte teksteditoren for skriving av forumposter kommer ikke fram hverken i Opera eller Firefox. Brukerne av disse nettleserne må ty til ren HTML-koding når de poster.
- Innholdet og oppførsel på meny over diskusjonsforumet varierer mellom IE, Firefox og Opera.
- Visuell layout er noe uryddig ved at det f.eks. er varierende hvorvidt (og i hvor stor grad) headeren har en fast eller flytende bredde mellom de 3 nettlesertypene IE, Firefox og Opera. I IE er headeren flytende til en bestemt minimumsbredde. Er vinduet smalere enn dette så forblir headeren fast på minimumsbredden, mens kroppen på sida fremdeles er flytende.

Dette skaper en skjev og uryddig layout. I Opera og Firefox har headeren flytende bredde uansett bredde på nettleservinduet, bortsett fra søkeboksen som befinner seg i headeren. Man risikerer at hele søkeboksen blir liggende langt utenfor headeren og sidas ramme (border). Dette skaper et uryddig og uprofesjonelt utseende.

- Brødsmulestien fungerer ikke. Stien er blank på portalens framside, mens den heller ikke viser hvilken side du er på for de andre sidene du navigerer til, siden det da kun står "PLU-veven", og ikke noe mer. Unntaket fra dette er sidene under diskusjonsforumet. Her oppdateres brødsmulestien korrekt. Manglende brødsmulesti gjør at brukeren ikke har skikkelig informasjon om hvor han/hun til enhver tid befinner seg på nettstedet.
- Selv om PLU har gått for at man skal unngå horisontal rulling ved en oppløsning på 1024 x 768, unngår portalen det akkurat *ikke*, selv om det ikke er mye om å gjøre. Scrollbaren er til stede i alle de testede nettleserne, og ved smalt nettleservindu kan man rulle langt utenfor sidas høyre ramme.
- Når man søker fra søkeboksen får man opp en innloggingsboks, noe som tyder på at man må logge inn for å søke. Dette er ikke særlig ønskelig om portalen også skal være åpen for folk uten medlemskonto på den.

Portalen har ikke blitt testet for påloggede brukere.

5.2 Teknologi

Eksisterende nettbaserte verktøy kan gjerne bygge på en rekke teknologier. Under er nevnt punktvis noen typiske teknologier med typiske anvendelsesområder eller nettbaserte verktøy de er egnet for å konstruere.

HTML: Artikler og statisk innhold.

XHTML: Definert som å være etterfølgeren av HTML. Skaperne av XHTML hadde målet å skape en renere separasjon mellom dokumentstruktur og presentasjon, og å reformulere HTML som en applikasjon av XML (<http://msdn.microsoft.com/en-us/library/aa479043.aspx>). XHTML skal følgelig gi bedre tilgjengelighet for funksjonshemmede.

PHP: Brukes hovedsakelig for server side scripting, og støtter en rekke databasetyper

(<http://no.php.net/manual/en/intro-whatcando.php>). Dynamiske websider, Web-basert chat, WCMS, etc.

PHP støtter å snakke med andre tjenester i protokoller som LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (på Windows) m. flere.

ASP.NET: Bygge XHTML-kompatible websteder og interaktive webapplikasjoner. Muligheter for å dra og slippe kontroller i GUI-grensesnittet (når man kombinerer ASP.NET med Visual Studio). Kombinerer websteder med webtjenester. ASP.NET har klasser som støtter opprettelse av standardtilfredsstilte webtjenester og webklienter. Teknologien egner seg for å lage nettsteder som er dynamisk og datadrevne, som skalerer godt og som virker bra på en rekke nettlesere uten tilpasset kode [9]. Teknologien er tett integrert med .NET-rammeverket.

Ajax: Asynchronous JavaScript and XML. Ajax er en nettleserteknologi som er uavhengig av webtjenerprogramvaren. Teknikken gjør det mulig å la JavaScript kommunisere direkte med webtjeneren ved bruk av et JavaScript XMLHttpRequest objekt. Med dette objektet kan JavaScript utveksle data med en webtjener uten å laste om sida. Ajax bruker asynkron dataoverføring (HTTP-forespørsler) mellom nettleseren og webtjeneren, og tillater derfor websider å spørre etter mindre informasjonsbiter fra tjeneren i stedet for hele sider. Ajax-teknikken gjør internettapplikasjoner mindre, raskere og mer brukervennlige [8].

Webstandarder (utviklet av W3C): Aaccessibility, HTML 5, XML 1.0, XHTML/ XHTML2, CSS, DOM, CDF, Internasjonalisering, OWL, etc.

Disse webstandardene søker å være standarder for god design, brukbarhet, kompatibilitet, uniformitet og tilgjengelighet o.l. for websider og webapplikasjoner.

.NET: .NET-rammeverket sitter på toppen av operativsystemet (de nyeste versjonene av Windows). Rammeverket består av:

- Kompilatorer for fem offisielle språk (C#, Visual Basic, C++, J# og Jscript).
- Klassebibliotek kjent som Framework Class Library (FCL), som har støtte for Windows og webapplikasjoner, datatilgang, webtjenester m.m.
- Common Language Runtime (CLR), en objekt-orientert motor i hjertet av rammeverket som oversetter mellomliggende (intermediate) kode generert av språkkompilatorene til native kode nødvendig for å eksekvere applikasjonen.

JavaScript: Et skriptspråk som kjører på klienten, men som også kan brukes på serveren. Språket kan brukes i de fleste kjente nettlesere. JavaScript kan enkelt integreres med HTML-elementer.

Med JavaScript kan designere uten stor kodeerfaring kode en klientside-applikasjon uten store problemer pga. språkets enkelhet. Man kan lage dynamiske sider. JavaScript brukes ofte til å validere tekstfelt ([http:// www.topxml.com/javascript/](http://www.topxml.com/javascript/)).

Java: Java tilbyr to typer web-applikasjoner: 1) Presentasjonsorienterte webapplikasjoner som genererer interaktive websider som inneholder forskjellig typer markup-språk (HTML, XML, osv.) og dynamisk innhold som respons til forespørsler. 2) Tjenesteorienterte webapplikasjoner som implementerer slutt punktet i en webtjeneste (web service) ([http://java.sun.com/j2ee/1.4/docs/tutorial/doc/ WebApp.html](http://java.sun.com/j2ee/1.4/docs/tutorial/doc/WebApp.html))

Javateknologier og -rammeverk inkluderer bl.a. JSP, Java Servlets, Webtjenester og Java Web Application Architecture (JWAA).

5.3 Spørreskjema om nettbaserte tjenester

Det ble sendt ut 2 spørreskjemaer. Ett rundt bruk og behov for nettbaserte tjenester i etter- og videreutdanning av lærere ved NTNU, og et annet som skulle kartlegge behov ytterligere, samt samle en profildata om praksisveilederne. Det siste spørreskjemaet er stort sett i sin helhet gjengitt i kap. 4.1.3 (Evaluering av kravspesifikasjonen). Det første spørreskjemaet er vedlagt i Appendix 10.1. En analyse av svarene på undersøkelsen er gjennomgått i denne seksjonen. Analysen kan, i tillegg til å belyse hvilke nettbaserte tjenester som benyttes av de involverte partene, også bidra til å belyse hvorvidt det virkelig er behov for nye nettbaserte ressurser.

En oppsummering av svarene på spørsmålene i skjemaet i Appendix 10.1 følger. Det er viktig å poengtere at verktøyene/tjenestene omtalt i spørreskjemaet ikke innbefatter PLU-veven, siden skjemaet ble sendt ut *før* PLU-veven hadde blitt tatt i bruk.

Alle (100 %) bruker et eller annet nettbasert verktøy. Epost brukes av de fleste (78 %). Å surfe fritt rundt på Internett etter informasjon brukes av svært mange (89 %). Svært lite foregår gjennom nettpat eller nettmøter (0 %). Noen få bruker blogg som verktøy (11 %). Noen få bruker video som media for informasjons- og kunnskapsoverføring (22 %). Svært få ser ut til å bruke ren lyd som medieformat (0 %). Diskusjonsfora på nett brukes av noen (33 %). Noen få bruker andre verktøy (11 %).

Når det gjelder spørsmålet om de nåværende tjenestene man bruker, er dekkende for folks behov, svarte folk slik: Ingen mente tjenestene var lite dekkende (0 %). Svært få mente tjenestene var litt dekkende (11 %). Over halvparten mente tjenestene var mye dekkende (56 %). Ingen mente tjenestene dekket alle behov (0 %), men noen mente tjenestene dekket mer enn man har behov for (22 %).

Når det gjelder spørsmål om hvilke tjenester man syntes man trenger i sitt arbeid med etter- og videreutdanning av lærere, svarte respondentene slik: Det store flertallet var samstemte i at epost og nettsider generelt var tjenester man trengte (epost 78 % og nettsider 100 %). Nettprat/nettmøter, blogger, video og lyd var alle nødvendig for 22 % av respondentene. Det samme gjaldt andre verktøy og tjenester, som delbar whiteboard for håndskrift, og wiki (22 %). Diskusjonsforum på nett var etterspurt av 33 % av respondentene. Kun 11 % var usikre på hva de hadde behov for.

Når det gjaldt spørsmålet om hvilken nettbasert tjeneste som var viktigst i respondentenes arbeid, var de fleste enige (67 %) om at nettsider generelt var viktigst. Epost (22 %) og andre (11 %) var også viktig for noen. Når det gjelder hvilke tjenester som kom som nest viktigst i respondentenes arbeid, havnet epost og nettsider (begge 44 %) høyest på lista, mens 11 % hadde diskusjonsforum på nett som nest viktigst. Som tredje viktigste nettressurs pekte over flertallet (67 %) av respondentene ut diskusjonsforum på nett. Noen (33 %) pekte ut video og it's learning som tredje viktigste nettressurs i sitt arbeid. Når det gjelder spørsmålet om hvilken nettbasert ressurs som var minst viktig i respondentenes arbeid, svarte litt over halvparten (56 %) blogger. Lydopptak (22 %) og nettprat (11 %) ble også nevnt av noen som minst viktige nettressurser.

5.3.1 Analyse av dataene

Det var totalt 9 respondenter til spørreundersøkelsen, hvorav 6 var koordinatore/veiledere for lærerstudenter, av i alt 36 slike koordinatore som finnes rundt på de forskjellige praksisskolene.

Det vi først og fremst kan konkludere av resultatene, er at epost og nettsider generelt, er, som forventet, de mest brukte nettbaserte ressursene, og også de som betraktes som de viktigste i arbeidet. Grunnen til dette er nok at epost er blitt veldig utbredt i løpet av de siste 15-20 årene, og er blitt et offisielt kommunikasjonsmedium mellom arbeidskolleger og bedrifter. Samtidig ligger veldig mye informasjon, for eksempel om bedrifter og institusjoner, ute på nettsider. Dette er nok noe av forklaringen på at disse nett-tjenestene scorer høyt på respondentenes bruks- og behovsliste.

Diskusjonsforum var også en viktig nett-tjeneste, etter epost og internett.

Videre virket det som over flertallet var fornøyd med de nettbaserte ressursene man hadde til rådighet på det tidspunktet spørreundersøkelsen ble gjennomført, fordi 56 % mente de daværende ressursene var mye dekkende. Det var også flere (22 %) som mente tjenestene dekket mer enn man hadde behov for, enn de (11 %) som mente de dekket mindre enn det man hadde behov for. Det ser også ut til at noen ønsket seg nye nettbaserte verktøy, som nettprat/nettmøter og lyd. Disse var på det daværende tidspunkt ikke i bruk hos noen av respondentene, men var etterspurt. Også en litt spesiell teknologisk løsning, en *delbar whiteboard* for håndskrift, var ønskelig.

Et moment å være klar over er at det kan ha vært tvil om alternativet ”nettsider” også innbefattet nettbaserte verktøy som et CMS eller LMS. Det ville også være naturlig, siden mange websider er oppbygd ved hjelp av et CMS (LMS).

6 Design og utvikling av nettbaserte verktøy

I dette kapitlet blir det sett på hvordan man kan designe og utvikle forskjellige nettbaserte verktøy, og gjengangstonen ved utvikling av slike verktøy. Det blir først og fremst vinklet mot PLUs behov for nettbaserte verktøy.

Design er svært vanskelig når lite kontakt, tilbakemeldinger og kravinnsamling fra brukerne foreligger. Dette studiet har hatt minimalt med tid og ressurser til dette, så designet må derfor ses på som en rettesnor. Utviklere som ansettes av PLU burde gjennomgå hele eller deler av prosessen selv.

6.1 Design av nettbaserte verktøy

6.1.1 Egenutviklet løsning

For å kunne konstruere et nettbasert verktøy fra bunnen, må man anvende systemutviklingsmetoder som omfatter hele utviklingsprosessen. Som et forslag kan man se på hele løsningen som et lagdelt system der hvert lag er en viktig del eller side av løsningen. Forskjellige lag kan være:

- Brukbarheten for systemet

- Funksjonalitet i systemet
- Sikkerhet i systemet
- osv.

Utviklerne bør, i samråd med PLU, velge hvilke sider/lag av systemet som skal prioriteres høyest, der det er mulig å velge dette, og sette dette opp i en lagdelt liste for systemet som helhet. Skal f.eks. sikkerheten vektlegges over grensesnittet? Er ytelse og skalerbarhet viktigere enn kostnader, osv. De høyest prioriterte lagene burde tas fatt på noe tidligere enn de lavere, og spesifiseres ganske utfyllende før spesifiseringen av de lavere lagene tar til. Dette gjelder for design og utvikling av systemet. I selve implementeringsprosessen vil kodingen av de forskjellige lagene i systemet være sammenflettet og nokså sammenfallende i tid. F.eks. må man tenke på sikkerhet etterhvert som man koder funksjonalitetet etc.

De høyeste lagene har virkning og konsekvenser for alle underliggende lag. Ethvert lag må føye seg etter alle lagene som befinner seg ovenfor dette på lista. På den måten kan man sette målet med systemet som øverste lag, og gjerne spesifisere uviktigere sider ved systemet i et lavere prioritert lag. Med tanke på at det får store konsekvenser for et utviklingsprosjekt i form av omlegginger og omgjøring av arbeid når krav til forskjellige sider av systemet endrer seg i løpet av prosjektiden, så bør man være varsom med å ha en høy frekvens på testing og oppdateringer på de høyeste lagene for systemet, spesielt i starten. Dette fordi endringer i disse lagene vil ha størst konsekvenser hvis det må gjøres om på disse senere i utviklingsprosessen (de har flere underliggende lag som det følgelig også må endres på). En teoretisk guide under disse høyfrekvente iterasjonene kan være syklusen bestående av Awareness – Understanding – Action, utdypet i [45]. "Awareness" sikter til at man er oppmerksom på seg selv og verden rundt (her: systemet og verden utenfor systemet), "understanding" sikter til at man oppbygger seg forståelse for hva man må gjøre for å tilpasse seg verden rundt (her: hvilke endringer som må gjøres i systemet), mens "action" sikter til at man gjør det som må til for å tilpasse seg (her: gjør endringer i systemet).

Forslaget under er utarbeidet ut fra teori, PLUs behov, og forfatterens erfaring med IT-systemer. Forslaget er spesielt rettet mot den langsiktige versjonen av PLU-veven, men kan også benyttes som et utgangspunkt for PLU-veven på kort sikt.

Datainnsamlingsmetode for hvert punktvis lag presenteres i kap. 6.2.1.

Prioriterte "lag" av systemet:

1. Mål
2. Brukberbehov
3. Brukbarhet
4. Funksjoner
5. Grensesnitt og interaksjon
6. Sikkerhet
7. Ytelse
8. Kostnad
9. Vedlikehold og administrasjon
10. Teknologi

Design for hvert lag er utdypet nedenfor.

6.1.1.1 Mål

- Målene for de nettbaserte verktøyene er å hjelpe PLU å tilfredsstille forretningsmål som de har satt seg.

6.1.1.2 Brukerbehov

Her er listet opp forskjellige behov brukerne har, og som PLU-veven er tenkt å skulle oppfylle. Dette er behov som er ekstrahert fra møtoreferater og intervjuer o.l., og som kommer i tillegg til kravspesifikasjonen (kap. 4.1):

- Verktøyet skal hjelpe til med å tilegne seg informasjon og kunnskap relatert til det som trengs for å bli videreutdannet som lærer innen ett eller flere fagfelt.
- Høyne praksisveiledernes kompetanse innen FoU.
- Verktøyet skal være en sosial læringsarena, der brukerne kan bli en del av et samfunn av andre etterutdannende lærere, forelesere, og studenter.
- Verktøyet må kunne brukes av folk med lite IKT-kompetanse.
- Verktøyet skal være en samlende faktor på PLU som organisasjon.
- Innhold må være oppdatert og aktuelt
- Lærere skal kunne publisere innhold
- Lettvint
- Presentere faglitteratur og fagstoff

- Så mye som mulig tilrettelagt for funksjonshemmede.
- Selvinstruerende
- Inneha de funksjoner ønsket av de potensielle brukerne (identifisert i intervjuer og spørreskjema etc.)

6.1.1.3 Brukbarhet

Webportalen bør være logisk, oppfylle webstandarder, ha god brukbarhet og være lett å navigere i for at det skal være enkelt å tilegne seg informasjon og kunnskap. Grensesnittet må optimalisere og fremheve funksjonaliteten. Interaktiviteten bør gjenspeile utførelsen slik mennesker er vant til å utføre oppgaver i den fysiske verden, og oppfylle kjente, utbredte grensesnittstandarder. Det bør legges til rette for funksjonshemmede. Siden PLU-veven blir liggende under NTNUs nettsider, må de følge NTNUs designstandard for websider.

Tekst bør være slik at de kan leses godt av forskjellige typer brukere. Teksten bør kunne leses godt på skjerm, og skrives lett ut til skriver i et format tilpasset utskrift. Ha gjerne en knapp for å få fram artikkelen i et format klart for utskrift. Det bør ikke presenteres for mye tekst på én side. Lange artikler leses sjelden av nettbrukere [18].

De forskjellige funksjonene bør være lettforståelig og oversiktlig. Her må man ta hensyn til nybegynnere innen IKT. Er det for komplisert, kan man ikke regne med så stor deltagelse [25]. Samtidig kan erfarne webbrukere føle det som en nedgradering å benytte webbaserte verktøy med mindre tilleggfunksjonalitet enn det de er vant med i andre nettressurser. Fortrinnsvis bør kanskje de erfarne brukerne lokkes til å bruke et enklere verktøy ved å legge ut mye nyttig og interessant stoff, presentert med et image som passer PLU-veven.

Verktøyet bør ha gjennomgående norsk og godt beskrivende språk. Nettstedet kan på sikt også tilbys på engelsk.

6.1.1.4 Funksjoner

Med utgangspunkt i ovenforliggende punkter/sider ved systemet, defineres følgende funksjoner som ønskelige i PLU-veven. Behov for funksjoner kommer spesielt fram under prototyping i punktet

ovenfor, og følgende liste må derfor bare ses på som en mulig rettesnor:

Artikler/innhold

- Skrivning og redigering av nye artikler eller nytt innhold. Disse må kanskje godkjennes av en innholdsadministrator med høyere brukerrettigheter før de kan bli postet på websida.
- Programsetting av innhold: Innhold skal kunne programmeres til å postes på websida innenfor et visst tidsrom. Dette kan forebygge utdatert innhold, og kan koordinere publisering av aktualiteter i riktige tidsrom.
- Forhåndsvisning før publisering. Forebygger at man må slette artikler/innhold gjentatte ganger før man er fornøyd. Ofte kan da sletting glemmes, og en uferdig artikkel vil befinne seg på websida.
- Kategoriseres innenfor et tema eller en kategori. Dette skaper orden i innholdet på portalen. Spesielt gjelder dette etterhvert som innholdsmengden i systemet vokser. Kategorier burde kunne endres uten at innholdet tilknyttet dem forsvinner.
- Skrivning og redigering av innhold skal ha de viktigste funksjonene for tekstbehandling. Dette skaper motivasjon hos enkelte innholdsskapere.
- Må kunne sette inn medieelementer som lyd, bilde, video og animasjon. Medieelementer kan vinkle læremessig stoff fra nye sider.
- Gi innholdet en tittel/identifikasjon. Dette er en identifikasjon av innholdet, og er separat fra en overskrift for artikkelen (som kan være forskjellig fra tittelen).
- Redigering av metadata tilknyttet artikkelen/innholdet. Etterhvert som innhold endres, eller konteksten for innholdet endres, kan det være et behov for å endre metadata.

Diskusjonsforum

- Poste innlegg og svare på innlegg. Viktig for deltagelse og engasjement i temaer.
- Se andres innlegg og diskusjoner. Viktig for å holde seg oppdatert på stoff og øke sin kunnskap.
- Skrivning av innlegg må følge tekstbehandlingsstandarder, og være WYSIWYG. Unngå HTML-koding. Dette er viktig for å opprettholde motivasjon for enkelte web-brukere (noviser).
- Muligheter for en begrenset mengde smilefjes. Dette skaper engasjement hos enkelte web-brukere, og større muligheter for å uttrykke seg i et diskusjonsforum.

- Må kunne sette inn medieelementer som lyd, bilde og video og animasjon. Medieelementer kan vinkle læremessig stoff fra nye sider.
- Opprette og tilpasse sin egen “avatar” eller profilinnstillinger, uten å overdrive dette. Kanskje nok med bare et registrert brukernavn. Avatarer og profiler kan bidra til større engasjement hos enkelte web-brukere, siden de får mulighet til å utvikle sin egen originale identitet på portalen.
- Innlegg må kunne modereres, slik at upassende og useriøse innlegg ikke får florere.
- Systemet bør foreslå eksisterende tråder med samme tema når en bruker forsøker å opprette et nytt diskusjonstema (som i Google Groups). Dette forhindrer for mange tråder om samme tema.

Fil- og dokumentarkiv

- Administratorer og de med rettigheter: opplastning av filer. Dette gjør det mulig for flere å bidra med filer og dokumenter som kan ha nyttig informasjon for portalens formål.
- Nedlastning av filer slik at brukerne kan dra nytte av de opplastede filene.
- Diverse statistikk og oversikt (over nedlastning og lignende), slik at administratorer kan finne ut hva som er populært og hvorvidt brukerne laster ned de utlagte filene/ dokumentene.
- Mulighet for kommentering av filer og dokumenter i web-grensesnittet. Dette kan gjøre det mulig for brukerne å utveksle meninger og synspunkter om en fil eller et dokument, slik at man slipper å opprette eget diskusjonsforum for dette (som mange brukere ikke vil finne fram til).

Podcasting(lydfiler)

For en forklaring på begrepet “podcast” se kap. 1.3

- Podcasts kan abonneres på gjennom RSS-funksjonaliteten.
- Administratorer og de med rettigheter: opplastning av filer. Dette gjør det mulig for flere å bidra med podcasts som kan ha nyttig informasjon for portalens formål.
- Om mulig en egen modul som tar seg av opplastning, kategorisering, presentasjon og navigering blant podcasts på websida. Dette kan være hensiktsmessig for å skille ut podcasts fra vanlige filer og dokumenter. Mer om hvordan dette tenkes gjort, under punktet for Grensesnitt og interaksjon.
- Mulig for andre brukere å kommentere en podcast. Forfatter av podcast kan aktivere og

deaktivere kommentarer av podcasten.

- Nedlastning av podcasts og avspilling direkte i nettleseren. Podcasts burde kunne lastes ned for de som ønsker å ta vare på dem (og spilles av i et lokalt program, en såkalt *podcatcher*). De burde kunne spilles av direkte i nettleseren for de som ikke bryr seg om dem om de må lastes ned først.
- Lydkontroll ved avspilling direkte i nettleseren.

Videostreaming

- Gjerne en egen modul som tar seg av opplastning, kategorisering, presentasjon og navigering i videoer på websida. Denne kan gjerne integreres med podcasts-modulen som foreslått under podcasts.
- Videoer som har blitt lastet opp og integrert i forumposter og artikler skal kunne spilles av direkte i artikkelen eller forumposten på websida. Dette er standard i mange nettsider om dagen, og krevde man nedlastning kunne man risikert liten interesse fra brukerne. Tilgang til slike videoer avhenger av tilgangen til artikkelen det er postet i.
- Videoene kan også lastes ned ved å høyreklikke på dem og lagre dem. For de som ønsker å ta vare på videoene, burde dette være mulig. Videoforfatteren kan bestemme hvorvidt en video er nedlastbar eller bare kan streames over nettsiden.
- Spill av, pause avspilling, restart, lydkontroll og lysstyrkekontroll o.l. Fant man en avspiller som kunne spilt av baklengs og spille av i forskjellige hastigheter så kunne dette vært nyttige funksjoner for (avansert) lærestoff.
- Mulig for andre brukere å kommentere en video. Forfatter av video kan aktivere og deaktivere kommentarer av videoen.

RSS

- Oppdatering av enkelte artikler/innhold innen enkelte temaer, nytt stoff og nyheter. Abonnerter på RSS-mating fra portalen burde kunne holde seg oppdatert på nyheter og nytt stoff.
- RSS kan brukes som en metode for å levere kursinnhold, se [32]. Det er mulig å integrere denne lille applikasjonen med eksisterende systemer, også kommersielle.

- Podcasts (eventuelt også videosnutter) burde tilbys i RSS-feed.
- *Nyhetsaggregatorer:* Ved bruk av nyhetsaggregatorer skal man kunne hente ut informasjon fra RSS feeds basert på nøkkelord som man oppgir. Dette kan fungere som en innsnevring av informasjonen for brukerne av plu-veven. Et verktøy for å gjøre dette er å finne på <http://www.bloglines.com>. Slik omtaler nettstedet verktøyet: *“Bloglines is a free online service for searching, subscribing, creating and sharing news feeds, blogs and rich web content.”*
- Begrenset innhold bør ikke vises i RSS-feed. Om mulig kan det lages egen RSS-feed for kun begrenset innhold.
- Egne RSS-feeds for blogger
- En brukerveiledning på portalen for hvordan bruke RSS-feeds (spesielt rettet mot noviser).

Søk

- SEO (Search Engine Optimization) kan benyttes for å få internettsøkemotorer til å lettere finne portalen, og innhold i portalen. Dette kan gjøre det mulig å lettere dra utenforstående til portalen. Begrenset innhold skal ikke være søkbart fra utenfor portalen.
- Innhold på portalen skal kunne søkes etter fra en søkeboks på portalen som er lett tilgjengelig fra framsida. Alt ubegrenset innhold skal kunne være søkbart. Begrenset innhold skal ikke være søkbart på portalen om den som søker ikke har rettigheter til å ha tilgang til begrenset innhold (dette vil da gjelde enkelte brukere og alle avloggede brukere). De som har tilgang til begrenset innhold skal kunne søke etter det på portalen etter å ha logget seg inn.
- Enkelt søk på framsida (eller i headeren på hver side), pluss en lenke til avansert søk.
- Det kan legges til rette for at bruker skal kunne krysse av hvilket innhold søkeordet skal gjelde for, f.eks. artikler, diskusjoner, filer og dokumenter, podcasts, blogg o.l.
- Man skal kunne søke på brukere, f.eks. fornavn, etternavn og/eller kallenavn/profilnavn.

Blogg

- Brukere, eller kun enkelte brukergrupper, skal kunne ha sin egen blogg på portalen.
- Bloggpostene skal kunne kommenteres av andre brukere, for å skape mer kommunikasjon.

Kommentarer skal likevel kunne deaktiveres av bloggforfatter om ønskelig, for å unngå uønskede kommentarer.

- Bloggen skal kunne kategoriseres i et åpent eller lukket tema (se “Tilgangsbegrensning”).
- Bloggforfattere skal kunne lenke (internt) til andre blogger på portalen såvel som til eksterne blogger og URL'er. Dette burde være mulig siden man ofte søker å lage seg et *samfunn* (community) ved blogg (se kap 4.2.3).
- For mer relevant funksjonalitet, se kap. 4.2.3 (Relevans for plu-veven).

Wiki

- For å la portalens brukere bygge en online kunnskapsbase over etter- og videreutdanning av lærere.
- For matematisk informasjon kan man prøve å utvikle eller integrere en eksisterende infrastruktur for samhandlende skriving av formalisert matematikk på nett i et wikilignende format, som omtalt i [29] i deres satsning på et slikt system (Miki). Miki er omtalt i kap. 4.2.6.
- Flere brukere i portalen gis tilgang til å redigere i wikisystemet, men minst én person bør ha oppgaven med å kvalitetssikre innholdet. Kvalitetssikring kan også innskjerpes ved at brukeridentiteten til de som har editert en side vil vises [28].
- Redigering av sidene i wikidatabasen burde være WYSIWYG, slik at mange vil delta. Helst burde man benytte samme innholdsredigeringseditor som for artikler og innhold, og tillate multimedia.
- Søktbart innhold.

Tilgangsbegrensning

- Alle typer innhold på websida (artikler, videoer, podcasts, filer, dokumenter, blogg o.l.) skal kunne kategoriseres i tema som er begrenset/lukket. Alt innhold som er kategorisert i et begrenset tema, blir selv begrenset. Et begrenset tema kan ha samme navn som et åpent tema, men med sin egen ID som gjør den unik. Man kan når som helst gjøre temaet åpent igjen, men dette gjøres bare av en (super-) administrator. Har et begrenset tema samme navn som en åpent tema, og gjøres åpent, slås innholdet i disse temaene sammen.

- Brukere som har tilgang til begrenset innhold, skal kunne se de begrensede temaene og deres innhold når de logger på.
- Når et innhold lages, kan man velge å legge det til et åpent tema, eller et begrenset/ lukket tema.
- Oppretting, redigering og sletting av åpne og lukkede tema administreres av en superadministrator. Dette for å hindre uaktsom avdekking av begrenset innhold, og for å forbedre sikkerheten.

IM/Chat

- Fortrinnsvis et webgrensesnitt.
- Teknologi (f.eks. PHP, Java, Ajax, JavaScript) uvesentlig, så lenge det tilfredsstillende behovene.
- En liste over innloggede brukere med deres status (f.eks. borte, opptatt etc.) i chat-vinduet eller på websida burde vises.
- Muligheter for moderering (forvisning og utestengning) hvis PLU har ressurser til dette.
- Man kan være innlogget/tilgjengelig i chat som standard når man er innlogget, unntatt når man har eksplisitt gitt uttrykk for at man skal være avlogget. Dette betyr at å logge inn på portalen er det samme som å logge inn i chat. Dette betyr at avloggede brukere ikke skal kunne prate, noe som kan forhindre uønsket trafikk i pratekanalen. Eventuelt skal gjester kunne logge seg på chat fra framsida, men får kun tildelt chattenavn som starter med "gjest".
- Man skal slippe å "lete etter" chat-funksjonen. Den kan være tilgjengelig i footeren på samme måte som f.eks. i Facebook.

Eportefølje

- For kurs i PLU-veven kan kursporteføljer eventuelt benyttes. Kursporteføljer er et sett av dokumenter som fokuserer på utviklingen av et kurs, fra oppstart til sluttresultatet, og inneholder typisk et kurs' læremål, innhold og struktur, samt kursets rolle i et større utdanningsprogram. Porteføljen inneholder også evalueringer av studentens arbeid gjennom terminen [41].

- Eportefølge i samvirke med blogger og verktøy for sosiale nettverk kan skape et personlig læringsrom for brukere i webportalen [42].
- Ved å holde en oppdatert eportefølge kan andre se hvordan brukere utvikler seg (f.eks. praksisveilederne omkring forskning og utvikling) [42].
- Brukere kan lagre og vise fram sine arbeider i eporteføljen (i flere formater).

Personlig profil

- En personlig administrasjonsside hvor den innloggede brukeren kan redigere sin profil, laste opp bilde som representasjon av seg selv, endre sitt passord og sette tilgangsrettigheter for andre brukere til sin blogg og andre elementer, og lignende.

Andre funksjoner

- Klara Klok: Et dedikert sted på portalen der brukere kan spørre om FoU, veiledning og etter- og videreutdanning i form av at de sender inn spørsmål (asynkront eller synkront) til en figur (f.eks. "Klara Klok") som besvarer spørsmålene. Det kan rulleres på hvilke folk som skal "spille" Klara Klok og besvare brukernes spørsmål.
- FAQ: FAQ om bruk av portalen.

Flere mulige funksjoner blir vurdert etter hvert som utviklingen av systemet forløper

6.1.1.5 Grensesnitt og interaksjon

I dette systemlaget designes grensesnittet og hvordan interaksjonen med dette foregår for brukeren. Forslagene er basert på punktene ovenfor, spesielt brukbarhet i det foregående punktet.

- Administrasjonsgrensesnittet: Administrasjonsgrensesnittet bør være et webgrensesnitt som tilbyr:
 - Redigering av artikler/innhold
 - Redigering av brukere og brukertilganger
 - Tilgangsbegrensning på innhold
 - Oversikt over blogger (de enkelte bloggposter og tilhørende kommentarer bør ikke

kunne ses og redigeres unntatt av bloggforfatteren selv).

- Diverse globale instillinger

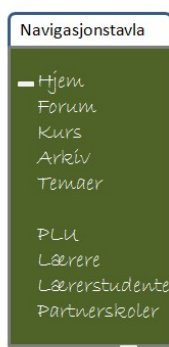
Opplastning og sletting av filer, dokumenter, media, videoer og bilder til et arkiv for dette.

- Redigering av portalens utseende eller utseendemal.
- Konfigurering av RSS eller nyhetsbrev per epost.
- Moderering, låsing, sletting og opprettelse av forumdiskusjoner.
- Konfigurering, aktivering og deaktivering av funksjonalitet og tilleggsutvidelser.
- Etc.

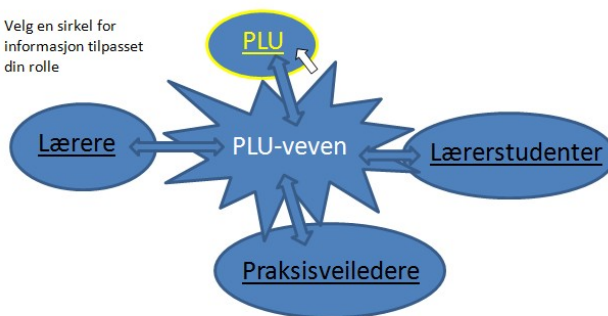
- Det bør være enkelt for de enkelte målgruppene for portalen å finne informasjon relatert til dem. Fortrinnsvis bør det være en separat lenke eller menyalternativ for hver av målgruppene (praksisveiledere, lærerstudenter og PLU-ansatte etc.) fra framsida, som f.eks. på figur 17.



Brukernavn
 Passord
 Søki pluveven



Velg en sirkel for informasjon tilpasset din rolle



Program for Lærerutdanning har et partnerskap med 36 skoler (grunn og vgs.) som hvert år får utplassert lærerstudenter slik at studentene får praktisk erfaring i det yrket de utdanner seg til. Fra og med i vår skal eksamen for disse studentene inkludere et såkalt FoU arbeid, et forsøks- og utviklingsarbeid, som skal ta utgangspunkt i den praksiserfaringen som de får fra skolen de oppholder seg på.

Program for lærerutdanning (PLU) ønsker derfor å utvikle et nettsted (PLU veven) for å styrke sin samhandling med praksisfeltet gjennom bruk av Internett. PLU veven skal i første omgang tilby moduler/kurspakker til lærere/koordinatorer ved våre partnerskoler slik at de kan heve sin veiledningskompetanse i forhold til våre lærerstudenter. På lang sikt ser vi for oss et nettsted som tilbyr en arena for formell og uformell kvalifisering av norske lærere, først og fremst de lærere som har, eller har hatt, sin utdanning ved PLU.

Figur 17: Figuren viser hvordan framsida i PLU-veven kan framstå som en enkel inngang for de enkelte målgruppene (PLU, lærere, lærerstudenter og praksisveiledere).

- **Artiklene:** Som også påpekt i [18], bør tekst som leses på skjerm har store og lesbare fonter, med god kontrast. Bakgrunnsfargen skal ikke krangle med fargen på tekstfonten. Relativ fontstørrelse brukes, slik at brukernes nettlekere kan stille inn tekststørrelsen etter eget ønske. For å unngå for lange tekster på én side brukes hyperlenker for å dele opp innholdet i artikkelen. Paragrafer brukes på lange tekstbolker. Tekst kan skrives lett ut til skriver i et format tilpasset utskrift ved å trykke på en knapp for å få opp teksten i et utskriftsvennlig vindu.
- **Diskusjonsforum:** For mange temaer gir et inntrykk av uprofesjonell drift; man bør finne et passende antall temaer brukerne kan diskutere, og ha med et generelt ("alt-mulig") forum i tillegg. Ta med et kjent antall smilefjes (ikke for mange, da dette blir for overveldende). Se også punktet for tekst lenger ovenfor. Et forslag til forumlayout integrert med en vri av studentgruppas layout av GUI er vist i figur 18.



Figur 18: En mulig prototype på et forumgrensesnitt integrert i et tematisert grensesnitt som i studentgruppas løsning.

- Fil- og dokumentarkivet bør tilby lettforståelig opp- og nedlastning av filer. Oppgi i grensesnittet om det vil være begrensninger på størrelsen på filene. Oppgi også om kun spesielle filtyper tillates opplastet. Opplastning av filer bør kun gjøres av innloggede og autoriserte brukere. Om mulig skal brukere slippe å høyreklikke og velge ”Lagre mål som...”.
- Podcasting kan enten legges i et eget arkiv for dette, eventuelt sammen med videoarkivet, eller sammen med fil- og dokumentarkivet, eller det kan integreres i artikler på nettsiden eller i diskusjonsposter. Om kun spesielle filtyper tillates som podcasts, opplyses det om dette i grensesnittet (under opplastning). Har man en avspiller et sted på websida kan dette skape engasjement rundt bruken av podcasts. Et annet alternativ er at podcasts bare integreres i forum- og artikler på webportalen, som lydfiler man kan legge inn i teksten. Dette kan også være tilgjengelig i tillegg til det første alternativet.
- Videostreaming bør enten finne sted i et eget arkiv for dette, som lenke til en ekstern side for dette, eller integrert i andre sider på portalen. Om man kan spille av direkte fra PLU-

veven, bør det foregå i et spillergrensesnitt som er brukervennlig selv for nybegynnere innen internettbruk.

- For RSS feeds er sannsynligvis bruk av det oransje XML-ikonet det som er mest utbredt og som brukerne kan kjenne igjen som RSS-funksjonalitet. Bruk dette som et klikkbart ikon på webportalen, helst med en beskrivende lenke ved siden av. Når brukerne klikker på disse, kan de abonnere på RSS-feeds. Webinnhold må tilpasses for å kunne brukes på mobile enheter, og dette kan være viktig å se på for en framtidig utgave av PLU-veven (noen vurderinger rundt dette er beskrevet i kap. 4.3.3). Riktig bruk av og anvendelse av teknikker for RSS kan forbedre webvisning på mobile enheter, ved for eksempel å redusere størrelse på innholdsmengden som tas inn for visning i den mobile enheten. For mer informasjon om en teknikk som kan brukes til dette, se [34].
- Søk foregår fra en inputboks på framsiden. Den bør være minst 25 karakterer lang for å støtte standard spørringer [16]. Ha en søk-knapp som må trykkes for å sende søket, eller bare ved å trykke "Enter". Det må i tillegg finnes en lenke til avansert søk.
- Hver registrerte bruker i PLU-veven skal kunne opprette sin egen blogg, og offentliggjøre den på PLU-veven. Siden mange ukyndige databrukere bruker blogg, bør grensesnittet for konstruksjon av blogginnlegg og tilpassing av utseende for bloggen være enkelt og logisk, og følge kjente standarder for blogg. Det er opp til PLU hvordan de vil at bloggen skal være, og hva slags tilleggsfunksjonaliteter som skal finnes i den. Men det er grunn til å forvente at man her bør ha et rikt utvalg av fonter, bakgrunner, fargevalg, rammevalg, smilefjes og innsetting av bilder og videoer i bloggen. Om mulig kan det forhåndslegges et utvalg utseendemaler/templates med et PLU-image som brukerne kan velge mellom.
- IM/Chat Noe av de samme brukergrensesnitt-egenskapene bør gjelde for chat-verktøyet som for diskusjonsverktøyet, nemlig at man ikke bør ha for mange kompliserte funksjoner og muligheter, men holde det på et akseptabelt nivå, slik at uerfarne chatbrukere ikke skremmes vekk. Bruk heller lokkemidler for å tiltrekke erfarne folk som måtte ønske mer.
- Personlig profil: Grensesnittet for personlig profil bør ha en visningsside som andre brukere kan se, og et grensesnitt der brukeren kan redigere sin profil. Profilsiden bør vise:
 - Bilde
 - Kort presentasjon av vedkommende
 - Fagspesialitet
 - En mulig rangering av hvor kunnskapsdelende vedkommende er. Dette er en automatisk

beregnet rangering fra 0 til 10 basert på hvor mye stoff vedkommende har delt/lagt ut på verktøyet.

- Lenke til blogger vedkommende har laget.
- Mulige interne lenker til anbefalt stoff på PLU-veven
- Mulige eksterne lenker til anbefalt eksternt stoff.

6.1.1.6 Sikkerhet

Sikkerhet er alltid viktig å ta vare på for datasystemer som opererer over det åpne internettet. Spesielt hvis det skal legges ut viktige data, som bør avgrenses fra offentlig innsyn, eller kopibeskyttet materiale. PLU bør finne ut hvor høyt sikkerhet skal prioriteres for PLU-veven, men det er en kjent sak at sikkerhet har blitt mer og mer viktig de senere årene, pga. økt aktivitet fra hackere, datasnokere og automatiserte programmer som angriper datasystemer.

Det er foretrukket å følge et risikorammeverk for utviklingen av systemet, for eksempel et rammeverk med 5 steg, der disse stegene består i å

1. Forstå forretningskonteksten
2. Identifisere forretningsrisikoer og tekniske risikoer
3. Syntentisere og rangere risikoene
4. Definere strategi for risikolindring
5. Foreta opprettinger og validering

der punkt 2-5 repeteres et nødvendig antall ganger for å redusere risikoen for produktet som utvikles. Utviklerne bør altså først forstå konteksten for systemet, forretningsmål og prioriteringer for systemet. Deretter har vi en repeterende løkke der man identifiserer forretningsrisikoer som direkte eller inndirekte truer et av forretningsmålene (slik at man ser sannsynligheten for at dette skal skje), prioriterer de risikoene som truer de mest sentrale forretningsmålene eller som kan umiddelbart true et forretningsmål, definerer en strategi som skal lindre og redusere disse risikoene på en måte som PLU som organisasjon kan ha råd til og kan forstå og håndtere skikkelig, og til slutt foretar de nødvendige opprettingene som definert i lindringsstrategien. Mer detaljert informasjon om dette risikorammeverket finnes i [35].

Lukkede innholdsressurser: Innholdsressurser som skal holdes lukket, bør sikres mot nettopp tilgang fra uautoriserte personer. Folk må ikke kunne logge inn uten å ha fått tildelt sitt eget brukernavn og passord. Hvor streng sikkerheten skal være, dikterer hvor strengt passordet må være, og hvor ofte det må endres. Politikken med å tildele brukernavn og passord må vurderes ut fra sikkerhetshensyn.

Utvikling av systemet må ellers bl.a. ta hensyn til følgende punkter:

- Det må ikke forekomme *avslørende kommentarer* i (HTML-) koden, noe som kan hjelpe hackere [36].
- Webapplikasjonen må ikke vise *avslørende feilmeldinger*. Dette kan hjelpe innbruddstyver ved å få kjennskap til instillinger på server eller svakheter i applikasjonen o.l. [36].
- Sikkerheten til tredjepartskomponenter må også sjekkes (dvs. komponenter andre har utviklet) [36].
- Man bør unngå bruk av *hidden fields* i HTML-forms [36].
- Sett *restriksjoner på inndatavalg* (f.eks. v.hj.a. JavaScript), slik at uventede inndata ikke forekommer [36].

Også flere andre sikkerhetstiltak bør innveves i programmet.

6.1.1.7 Ytelse

Systemet som helhet bør være ganske kjapp i responsen, maksimum *ett* sekunds responstid, hvis ikke oppfattes det ikke for brukeren som om det er noen sammenheng mellom bruk og respons. En responstid tregere enn ett sekund vil oppleves for brukeren som at han ikke opererer på dataene, og vil bryte brukerens tankeflyt [18]. Helst bør responstiden være under 0.1 sekund, men det sier seg selv at dette er vanskelig å oppfylle for en applikasjon som arbeider med å sende og motta data over nettet. Man er ofte prisgitt dataoverføringshastigheten i et hvert tilfelle.

- Chat: Brukeren bør ikke måtte vente lenger enn 10 sek på en oppdatering av chat-dialogen. Dette er grensen for å kunne beholde brukerens fokus på dialogen, som Jakob Nielsen påpeker i [18].
- Søk: Det bør ikke ta spesielt lang tid å søke i nettstedets innhold, maks 5 sekunder.

- Nettsider bør ikke bli overlesset med multimedieelementer, noe som forsinket nedlastning av sidene [18].

6.1.1.8 Kostnad

Kostnaden av systemet kan prioriteres inn på dette nivået. Dette avhenger av hvor høyt PLU setter kostnaden som et begrensende kriterium for systemet. Det kan derfor nedprioriteres eller prioriteres høyere, alt etter PLUs IKT-politikk eller tilgjengelige ressurser. En øvre og/eller en nedre kostnadsgrense kan spesifiseres her, som en ramme for underliggende sider (lag) av systemet.

- Utvidelseskostnader: Det påbeløper gjerne høyere kostnader med hver utvidelse av systemet. En øvre og/eller en nedre kostnadsgrense for tredjepartskomponenter og tilleggsmoduler som gir ny funksjonalitet settes her. Dette for å forhindre overskridelser, og for å sette en ramme for underliggende lag som har med utvidelseskostnader å gjøre.
- Vedlikeholdskostnader: En øvre og/eller en nedre kostnadsgrense for tredjepartskomponenter og tilleggsmoduler som gir ny funksjonalitet settes her. Dette for å forhindre overskridelser, og for å sette en ramme for underliggende lag som har med vedlikeholdskostnader å gjøre.
- Driftskostnader: En øvre og/eller en nedre kostnadsgrense for driftskostnader settes her.

6.1.1.9 Vedlikehold og administrasjon

På dette nivået kan man om mulig spesifisere hva man ønsker (ut fra tilgjengelige ressurser) å vedlikeholde ved systemet. En hel mengde sider av systemet er likevel en nødvendighet å vedlikeholde og administrere. Vedlikehold og administrasjon er først og fremst en side av utviklingsprosessen tilknyttet et system, mer enn en designside av systemet. Se derfor "Vedlikehold og administrasjon" (kap. 6.2.1.9) under "Utviklingsprosessen" for viktige områder under vedlikehold og administrasjon.

6.1.1.10 Teknologi

På dette nivået kan det spesifiseres hvilke teknologier systemet skal bygge på. Hvis det er en sterk preferanse for en spesiell teknologi, bør dette spesifiseres på et høyere nivå. Hvis teknologien brukt

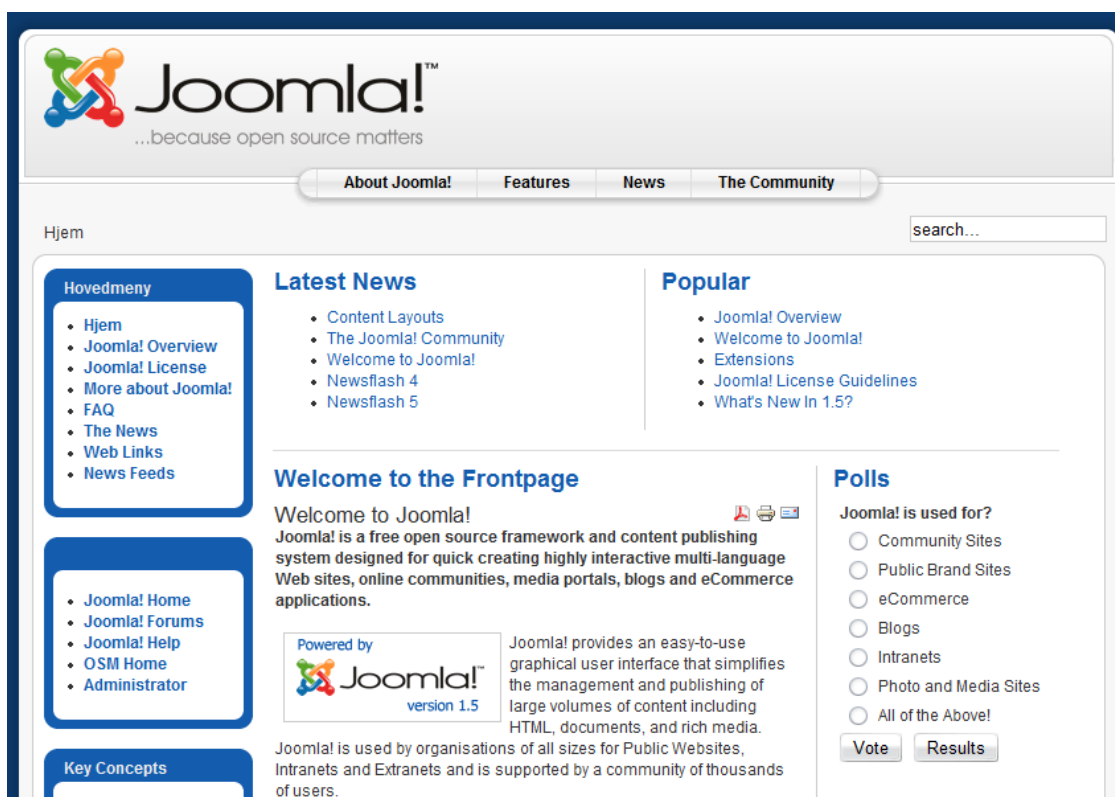
i kjernen av systemet er modulært og integrasjonsvennlig og støtter kjente standarder, kan utvidelser på systemet foregå i forskjellige andre teknologier. Teknologi for disse kan derfor variere.

- **IM/Chat:** Ajax kan benyttes for å forhindre en oppdatering av dialogen på lenger enn 10 sek. Ajax kan muliggjøre raskere lasting av websider ved å bare oppdatere deler av siden. Mer om Ajax i kap. 3.1.1 og 5.3.

6.1.2 Tilpasning av ferdig løsning

Denne seksjonen vil utbrodere forslag til tilpasninger av ferdige (“out-of-the-box”) løsninger for konstruksjon av webportalen PLU-veven. Joomla! har blitt valgt som et eksempel på et ferdig system som tilpasses etter kravspesifikasjonen for PLU-veven, og det vil bli vist en oversikt hvordan dette systemet kan tilpasses.

Før tilpasningen av Joomla! tar fatt i denne oppgaven, er det vist standard skjermbilde av framsida av Joomla! før tilpasninger har blitt gjort (det er bare gjort kosmetiske endringer – f.eks. endret tittel på hovedmeny). Det er valgt å installere eksempeldata i installasjonen, så innholdet og grensesnittlementene som er til stede ved første visning av sida er bestemt av dette. Se figur 19.



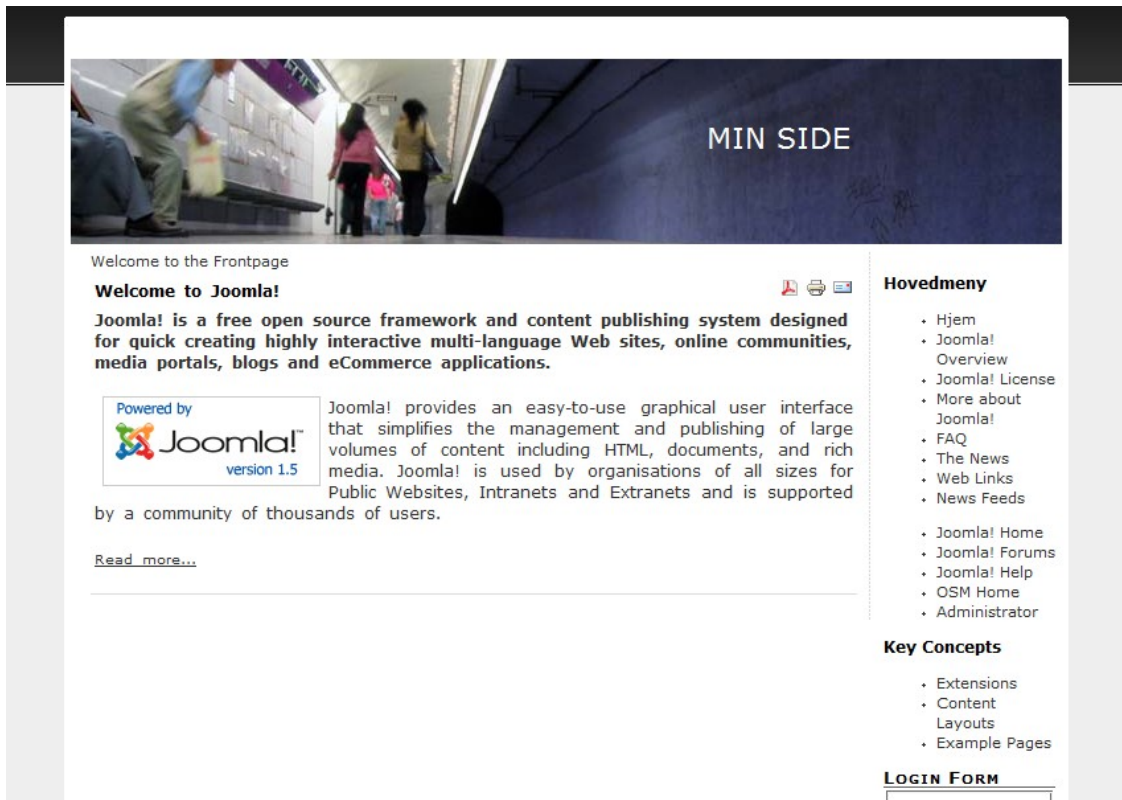
Figur 19: Framside i nyinstallert Joomla! (standard template)

6.1.2.1 Tilpasning av grensesnitt

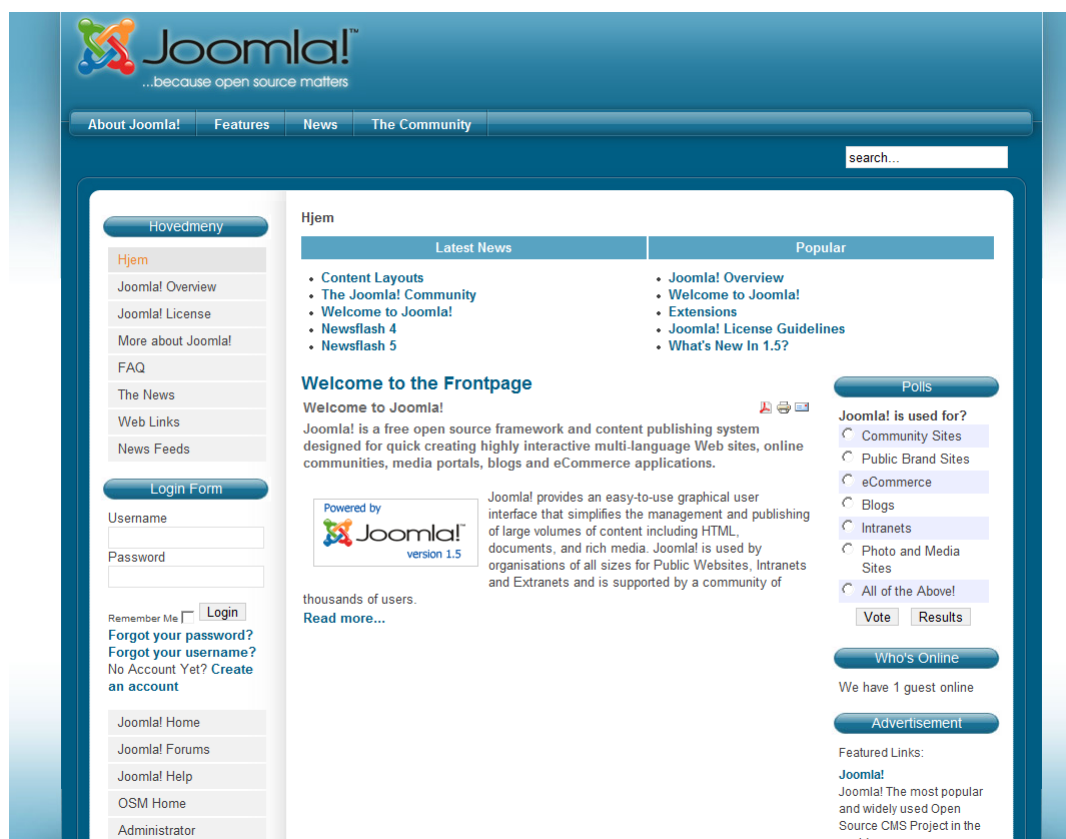
Grensesnittet bør tilpasses ønsket GUI for PLU-veven. Spesielt på kort sikt må nettstedet oppfylle NTNUs maler for nettdesign (se kap. 4.3.1), siden nettstedet skal ligge under NTNUs nettsider. GUI kan tilpasses ved enten å finne passende ferdige utseendemaler for løsningen som helhet, eller ved å tilpasse dette manuelt. En ferdig utseendemat vil stort sett bare endre på hvordan grensesnittets HTML-elementer skal stiles, og inneholder som regel ferdiglagde CSS-filer som lastes inn i systemet. For noen systemer kan utseendemaler også endre noe på HTML-elementene, som gjør at f.eks. menyer endres på i grensesnittet (forsvinner eller kommer fram o.l.). Men kan få utrettet mye ved selv å redigere på CSS-, HTML- eller Master Pages -filer (alt etter hvilken ferdig løsning man har i bunnen). Hvis det er ønskelig for PLU å ofte endre mye og detaljert på grensesnittet, bør man ansette en person med kompetanse på dette, og/eller bruke et verktøy med visuell redigering av grensesnittet. Det finnes neppe ferdige utseendemaler som tilfredsstillter NTNUs designmaler. Da må man benytte noe HTML- og CSS-koding. Men NTNUs sider er ikke spesielt kompliserte å designe, så med et bilde av NTNUs logo å plassere i headeren (NTNU-logoer kan lastes ned fra http://www.ntnu.no/selvhjelpspakken/logo_maler/logo), bruk av riktige fonter og litt smart bruk av rammer (bordere) i CSS rundt nettstedets innhold så burde man være et stykke på vei med å oppfylle designmalene.

Tilpasning av grensesnitt i Joomla!

Grensesnittet i Joomla kan endres enkelt ved å laste inn ferdige templates i løsningen. Man velger "Template Manager" fra administrasjonsmenyen, og kan der velge fra en liste over allerede installerte templates, eller man kan velge "install" fra utvidelsesmenyen, for deretter å velge "upload package file". Man velger da en template-pakke man har liggende lokalt på harddisken, for opplastning til løsningen. Figur 20 og 21 viser framsida når man har valgt andre templates.

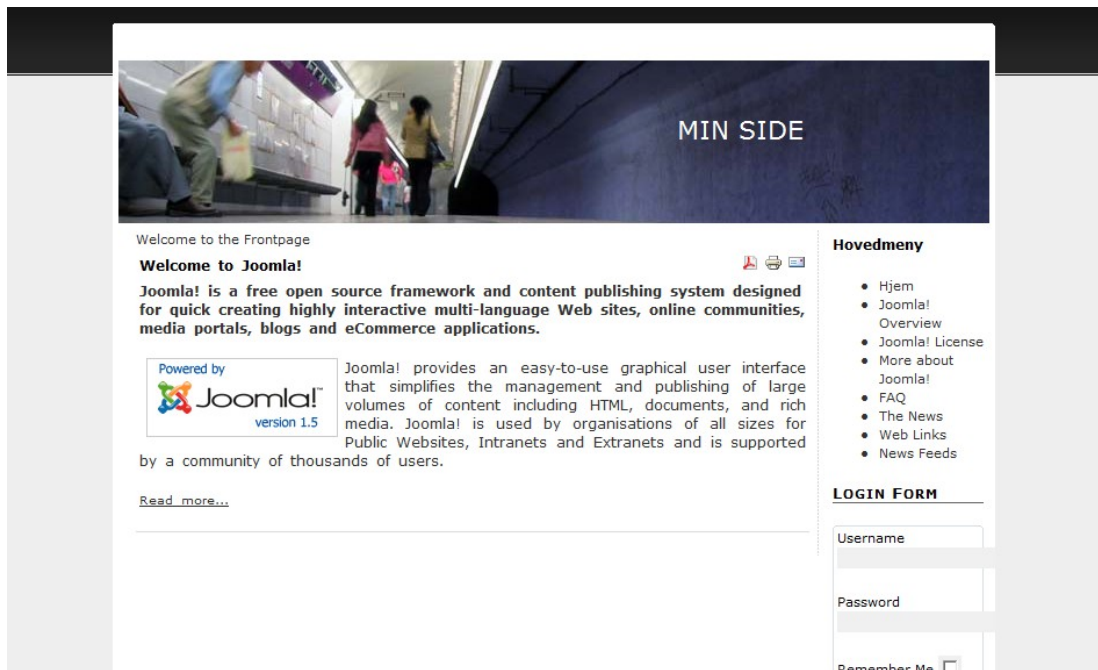


Figur 20: Framside Joomla! med endret template (Avaurus Metrobusiness)



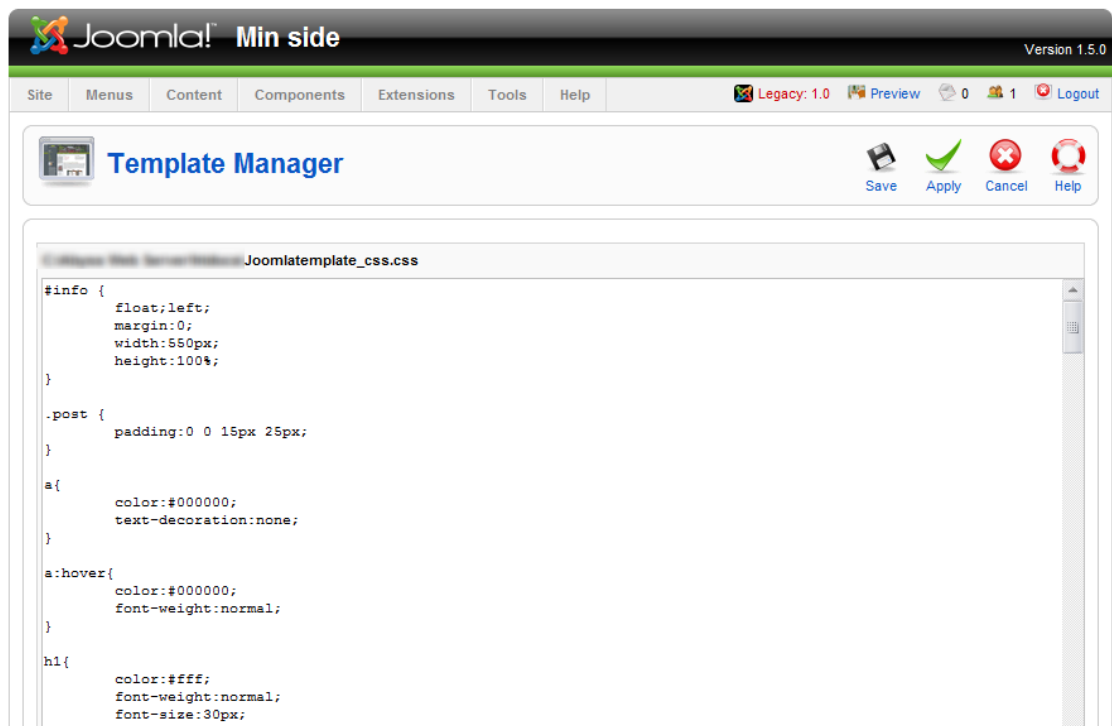
Figur 21: Framside med template "cah_ndeso".

Grensesnittet kan videre tilpasses ved å endre på menyene (mer om dette under "Tilpasning av navigasjon"), eller tildele moduler eller komponenter til fastbestemte plasser i websidas layout (dette vil samtidig berøre navigasjon- og funksjonstilpasning). F.eks. kan man velge om man skal inkludere en innloggingsboks (HTML-form) på en eller flere av sidene, eller flytte på den. Man kan flytte på innloggingsboksen ved å gå på "modules"-menyen, velge "Login Form" fra listen over moduler, og redigere denne til å komme høyere opp blant rekkefølgen av moduler som vises i høyre marg (endrer bare "order"-attributten til et høyere nivå). Figur 22 viser et nytt skjermbilde med template Avaurus Metrobusiness der innloggingsboksen er flyttet høyere opp.



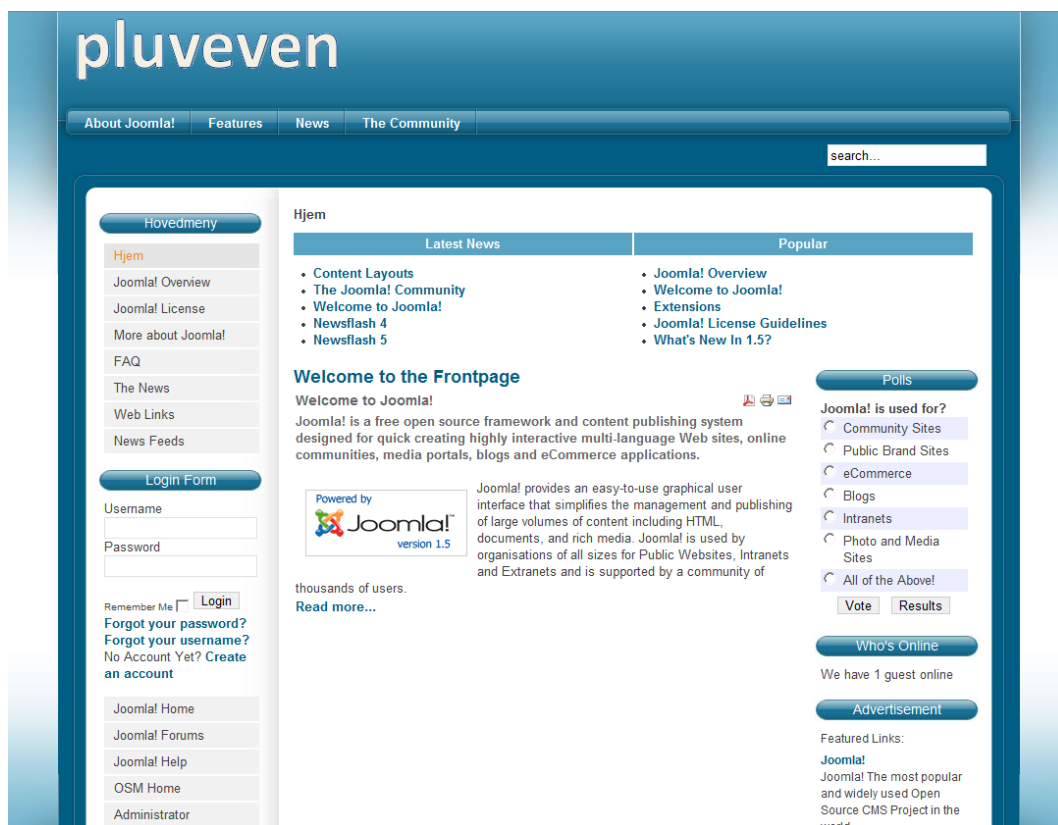
Figur 22: Innloggingsboksen er flyttet høyere opp.

For en mer detaljert tilpasning av grensesnittet, kan man endre selve stilarket og/eller HTML-koden som definerer utseende til den template som er valgt. I Joomla! kan dette gjøres ved å få opp CSS- og HTML-koden i selve administrasjonsgrensesnittet, som vist på figur 23.



Figur 23: Redigering av CSS-kode for en template

Ved manuell redigering av CSS- og HTML-koden kan man tilpasse slike ting som å legge inn lenke til sitt eget logobilde, samt posisjonere dette etter ønske o.l. Man kan også endre bakgrunnsfarger og marger og forskyvninger mellom moduler, bokser eller komponenter på websida, men dette krever noe kunnskap om CSS og HTML. Det kan lønne seg for PLU å først finne en nesten-passende template, for så å beskrive for en kompetent person hvilke tilpasninger som må gjøres for å få det ønskelige resultatet. Dette for å slippe å designe en template helt fra bunnen av (men det er også mulig). Figur 24 viser template cah_ ndeso endret i CSS-koden til å inkludere et nytt logobilde med teksten "pluveven".



Figur 24: Template-CSS endret til å inkludere ny logo.

Følgende linje måtte endres i CSS-koden for å endre logo (endret url til nytt bilde, pluvev_logo.png):

```
#logo{background:url(../images/pluvev_logo.png) no-repeat;width:295px;height:86px;float:left;position:relative;top:0px;left:10px;}
```

6.1.2.2 Tilpasning av navigasjon

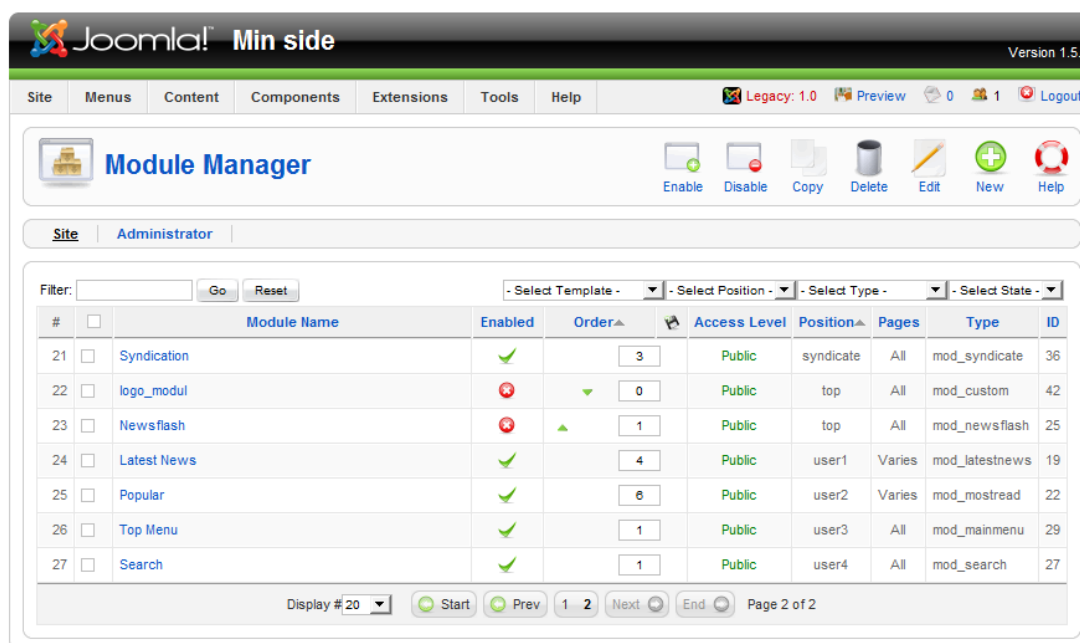
Navigasjon i et ferdig system kan tilpasses ved å endre synligheten av menyer, og hvilke menyalternativer som skal presenteres i dem. Tilgjengeligheten på lenker bidrar også til

navigasjonen, og kan tilpasses ved enten å redigere dem manuelt, eller tilgjengeliggjøre eller skjule de/den (interne) sidene/funksjonen de peker mot.

Som regel finnes det et administrasjonspanel som tar seg av behandling av menyer på portalen. Vanligvis kan man her opprette nye menyer, og redigere og slette dem, samt redigere de elementene/adressene de peker mot. Menyrekkefølger (på menyalternativer) kan vanligvis også endres. Visuelle plasseringer av menyene på portalen kan vanligvis ikke endres, eller bare endres ut i fra et antall forhåndsdefinerte plasseringer. Man må ofte ty til HTML/CSS for å tilpasse meny- og navigasjonsstrukturer til minste detalj rent plasseringsmessig.

Tilpasning av navigasjon i Joomla!

I Joomla! kan navigasjon tilpasses ved å endre på synligheten til menyer, endre på rekkefølgen på menyalternativer, og sette inn manuelle lenker osv. For å synliggjøre en meny, f.eks. toppmenyen, for en side, går man inn på "Modules"-menyen, og velger så om man vil aktivere eller deaktivere en meny, i dette tilfellet aktivere "Top Menu." Denne operasjonen er vist i figur 25.



The screenshot shows the Joomla! Module Manager interface. At the top, there is a navigation bar with tabs for Site, Menus, Content, Components, Extensions, Tools, and Help. Below this is the Module Manager header with a search filter and various action icons (Enable, Disable, Copy, Delete, Edit, New, Help). The main content area displays a table of modules with the following data:

#	Module Name	Enabled	Order	Access Level	Position	Pages	Type	ID
21	Syndication	✓	3	Public	syndicate	All	mod_syndicate	36
22	logo_modul	✗	0	Public	top	All	mod_custom	42
23	Newsflash	✗	1	Public	top	All	mod_newsflash	25
24	Latest News	✓	4	Public	user1	Varies	mod_latestnews	19
25	Popular	✓	6	Public	user2	Varies	mod_mostread	22
26	Top Menu	✓	1	Public	user3	All	mod_mainmenu	29
27	Search	✓	1	Public	user4	All	mod_search	27

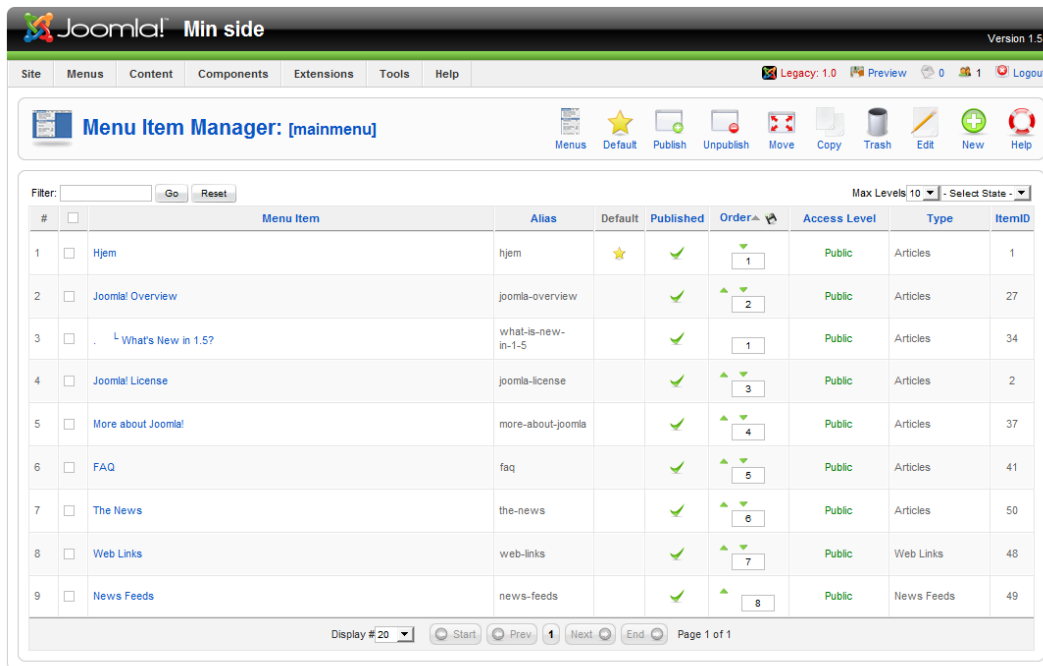
At the bottom of the table, there are pagination controls: Display #20, Start, Prev, 1 2, Next, End, and Page 2 of 2.

Joomla! is Free Software released under the GNU/GPL License.

Figur 25: Fra "Module Manager" kan menyer og andre moduler publiseres eller deaktiveres.

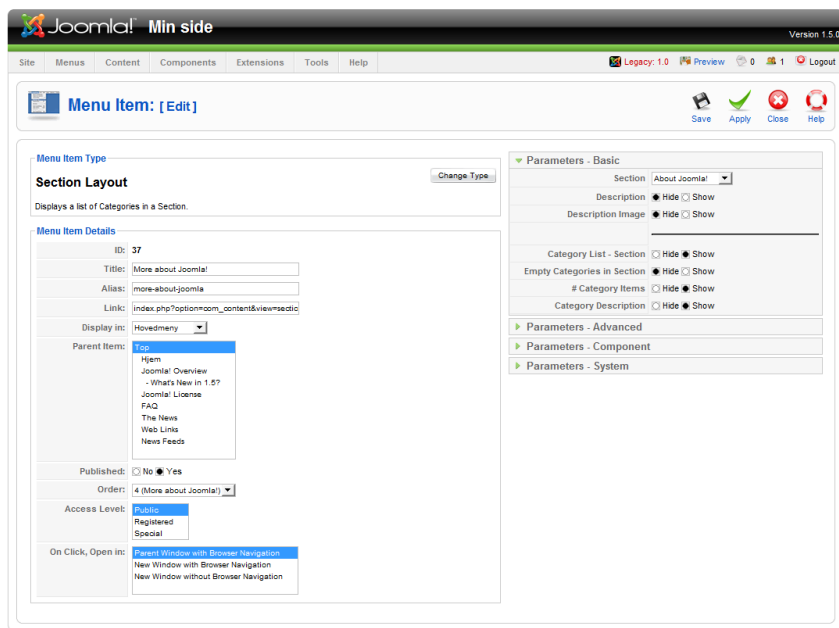
Man kan tilpasse, opprette og slette menyalternativer for menyene i "Menu Manager." På figur 26 er

det vist redigering av hovedmenyen i "Menu Manager."



Figur 26: Redigering av hovedmenyen

Her kan man trykke på ikonene i kolonnen "Published" for å sette menyelementer til publisert (synlig) eller ikke-publisert (usynlig). Man kan også klikke på "Access Level"-kolonnen for menyelementene for å bestemme deres tilgang (bestemme om de skal være tilgjengelige for avloggede, registrerte brukere eller spesielle brukere). Man kan også bestemme rekkefølgen på menyalternativene ved å redigere kolonnen "Order". Klikker man på et menyalternativ, får man opp et skjermbilde for redigering av selve elementet, som vist i figur 27.



Figur 27: Redigering av et menyelement.

Her kan man redigering elementets tittel, hvilken meny det skal befinne seg i, hvorvidt det skal være et underlement av et annet menyelement (trestruktur), pluss en rekke parametre. Man kan også endre hvilket type menyelement det skal være; f.eks. hvorvidt det skal peke mot en artikkel, ekstern lenke eller en seksjon eller kategori etc.

Figur 28 viser hvordan vi har endret hovedmenyen i portalen til å inneholde menyelementer relevante for PLU-veven. Vi har deaktivert standardelementene som fulgte med installasjonen, og i tillegg opprettet nye elementer.

Joomla! Administration Version 1.5.0

Site Menu Content Components Extensions Tools Help Legacy: 1.0 Preview 0 1 Logout

Menu Manager: [mainmenu]

Menus Default Publish Unpublish Move Copy Trash Edit New Help

Filter: [] Go Reset Max Levels: 10 - Select State: []

#	<input type="checkbox"/>	Menu Item	Default	Published	Order	Access	Type	ItemID
1	<input type="checkbox"/>	Hjem	★	✓	3	Public	Articles	1
2	<input type="checkbox"/>	Joomla! Overview		✗	5	Public	Articles	27
3	<input type="checkbox"/>	What's New in 1.5?		✗	1	Public	Articles	34
4	<input type="checkbox"/>	Joomla! License		✗	7	Public	Articles	2
5	<input type="checkbox"/>	More about Joomla!		✗	8	Public	Articles	37
6	<input type="checkbox"/>	FAQ		✗	9	Public	Articles	41
7	<input type="checkbox"/>	The News		✗	10	Public	Articles	50
8	<input type="checkbox"/>	Web Links		✗	11	Public	Web Links	48
9	<input type="checkbox"/>	News Feeds		✗	12	Public	News Feeds	49
10	<input type="checkbox"/>	Temaer		✓	13	Public	Articles	53
11	<input type="checkbox"/>	Arkiv		✓	14	Public	DOCman	56
12	<input type="checkbox"/>	Forum		✓	15	Public	Simplest Forum	59
13	<input type="checkbox"/>	PLU		✓	16	Public	Articles	57
14	<input type="checkbox"/>	Lærere		✓	17	Public	Articles	58
15	<input type="checkbox"/>	Praksisveiledere		✓	18	Public	Articles	60
16	<input type="checkbox"/>	Lærerstudenter		✓	19	Public	Articles	61
17	<input type="checkbox"/>	Prat		✓	20	Public	BlastChat Client 2.2	63
18	<input type="checkbox"/>	Skriv blogg		✓	21	Public	IDoBlog	72

Display # 20 Start Prev 1 Next End Page 1 of 1

Joomla! is free software released under the GNU/GPL License.

Figur 28: Hovedmenyen tilpasset PLU-veven.

6.1.2.3 Tilpasning av informasjonsorganisering og informasjonsflyt

Informasjonsorganiseringen bør planlegges og bestemmes uavhengig av det ferdige systemet man tar i bruk for tilpasning mot PLU-veven. Faktorer man bør tenke på er:

- Volumet av informasjon som skal presenteres eller ligge ute på portalen.
- En grov gruppering av noen få tema man kan dele informasjonen inn i. Antallet av det øverste nivået av temaer bør være fra ca. 3 – 10. Forskning viser at mennesker er begrenset til å huske omlag +- 7 forskjellige elementer i korttidshukommelsen. Dette gjør det lettere og raskere å jobbe med både for innholdsskapere og brukere av portalen. En gruppering mindre enn 3 elementer er nærmest bortkastet (men ikke umulig); man kan heller da la nivået under være høyeste grupperingsnivå.
- Man bør merke seg om informasjonen ikke kan grupperes, eller om man ser behov for en

endring (evolusjon) i portalens informasjonsgruppering i framtida. Dette må formidles videre til de som skal designe informasjonsaspektene ved portalen, slik at de kan ta høyde for dette i det grunnleggende designet.

- Hvor dypt innhold skal organiseres/kategoriseres. Hvor mange undernivåer bør det tas hensyn til?
- Målgruppa for innholdet.

For å organisere informasjonen på webportalen kan man bl.a.

- Tilknytte metadata til informasjonsobjekter/-elementer (dette kan være alt fra artikler, dokumenter, filer, presentasjoner, videoer, blogger, diskusjoner og podcasts osv.) som legges inn, som spesifiserer under hvilket tema de tilhører.
- Underlegge informasjonsobjekter under de aktuelle temaene. Dette kan vanligvis gjøres i et administrasjonsgrensesnitt eller ved å redigere egenskaper for et informasjonsobjekt.
- Et alternativ er å organisere og presentere informasjon med det samme temaet i egne sider eller plasser på portalen, hvis ikke logisk organisering under temaer ikke er mulig i verktøyet. Dette bør gjøres uansett, og man bør ta høyde for senere endringer (utvidelser spesielt).

Informasjonsflyt omhandler hvordan informasjonen flyter gjennom webportalen. Dette angår bl.a.:

- Hvordan prosessen for opprettelse, godkjenning, publisering og arkivering av artikler og innhold i portalen skal foregå.
- Hvor lang tid innhold skal ligge ute som publisert, før det legges i et arkiv (tas av webportalens frontend).
- Hvorvidt artikler og innhold skal godkjennes før de publiseres.
- Etc.

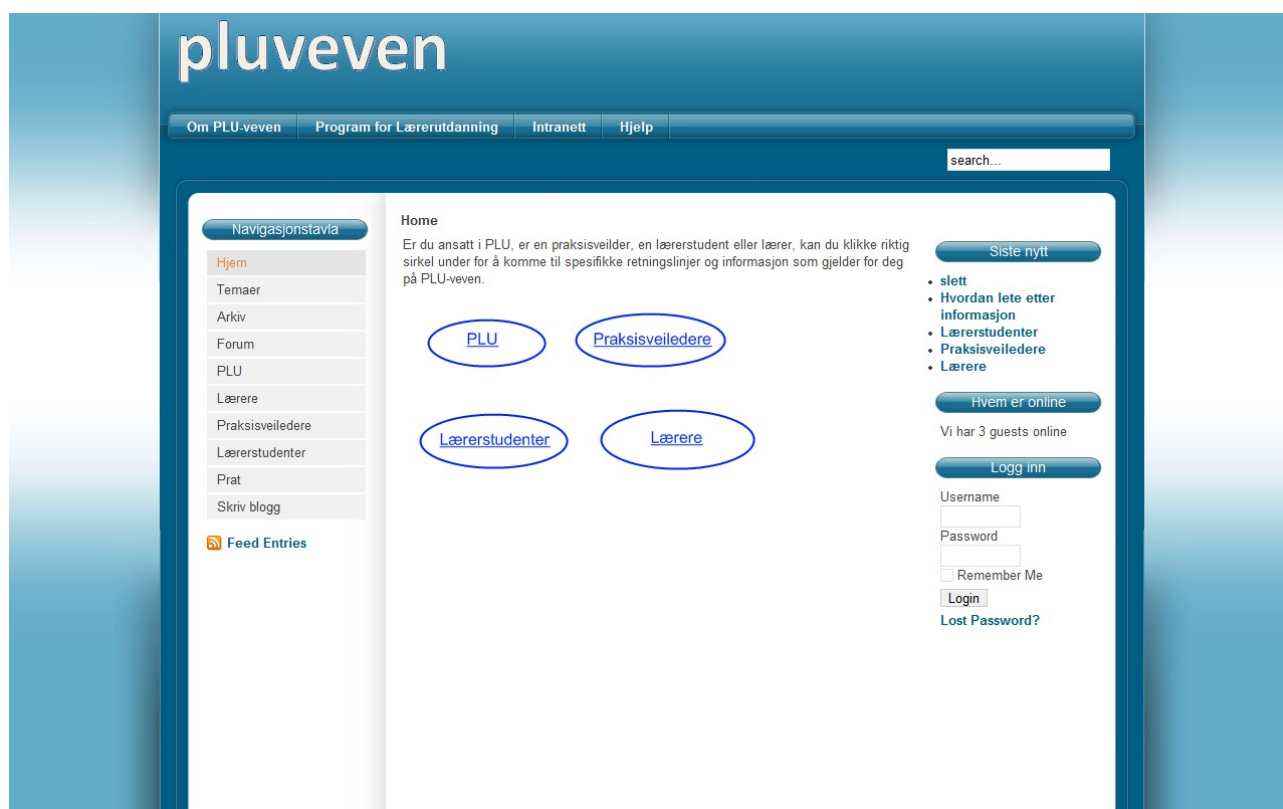
Tilpasning av informasjonsorganisering og informasjonsflyt i Joomla!

Informasjonsorganisering i Joomla! kan gjøres ved bl.a. å kategorisere artikler under egendefinerte seksjoner og kategorier. En seksjon har vanligvis flere kategorier under seg. For PLU-veven kan man la seksjoner være toppnivåtemaene, og kategoriene kan være undertemaer. Alle artikler som

skrives inn i portalen, og som gjerne publiseres, kan kategoriseres innenfor disse temaene. Det er en enkel jobb å opprette seksjoner og kategorier, og gruppere artikler under disse. Så fort man har informasjonsorganiseringen klart, kan disse opprettes.

Informasjonsflyten kan tilpasses bl.a. ved å bestemme, for hvert innhold som publiseres, i hvilket tidsrom det skal være publisert. Innhold blir automatisk publisert i oppgitt tidsrom, og blir automatisk tatt vekk utenfor dette tidsrommet. For artikler og innhold som krever godkjenning før det postes, ser det ikke ut til at Joomla! støtter en slik funksjonalitet direkte, men en bruker med tilgangsnivå "Publisher" kan bestemme hvorvidt innhold og artikler skal være publisert eller ikke [14].

I vårt eksempel har vi laget nytt innhold (artikler) som vi har organisert i seksjoner (pluveven, temaer osv.) og kategorier (nyheter, faglige ressurser osv.). Disse har vi knyttet til menyelementene på hovedmenyen. Toppmenyen er redigert, og vi har også plassert en artikkel for visning på framsida. Vår framside ser nå ut som i figur 29.



Figur 29: Framsida tilpasset PLU-veven.

Brødsmulestien befinner seg over teksten, der det står "Home". Denne kan lett endres til "Hjem". En videre tilpasning av framsida kan være å flytte innloggingsboksen øverst til høyre i headeren, siden det er et tomt og ubrukt område der. Innlogging er også en sentral funksjon i PLU-veven, og

burde være lett synlig.

6.1.2.4 Tilpasning av funksjonalitet

Funksjonalitet tilpasses vanligvis ved å aktivere og tilgjengeliggjøre komponenter, moduler, webdeler og utvidelser på websidene. Ofte finner man at systemet innehar ønskede funksjoner, og det er bare å gjøre små tilpasninger av disse for at det kan anvendes etter behov.

Eksempel 1: Tilpasning av Forum

Forum kan tilpasses ved å installere den utvidelsen, om nødvendig, som gir ønsket forumfunksjonalitet.

Tilpasning av funksjonalitet i Joomla!

Funksjonalitet kan tilpasses i Joomla ved at man tilgjengeliggjør komponenter og moduler for brukerne av portalen. Ønsker man funksjonalitet som ikke finnes i standardinstallasjonen av Joomla!, kan man søke etter det på <http://extensions.joomla.org/>, og laste det ned og installere det i løsningen. Vanligvis tilbyr nye utvidelser egne administrasjonssider under "Component"-menyen.

For tilpasning til PLU-veven kan det lastes ned utvidelser for blogg, f.eks. den ikke-kommersielle JooBlog, tilgjengelig fra extensions.joomla.org, eller kjøpe og laste ned den kommersielle og mer profesjonelle My Blog, også tilgjengelig fra extensions.joomla.org. Det kan også lastes ned forum, f.eks. den ikke-kommersielle forumløsnigen Agora, eller integrere et phpBB-forum i løsningen. Et utvalg flere funksjoner tilgjengelig for Joomla som kan være relevant for PLU-veven er: wiki, chat, community builder (støtte for profiler med bilder o.l.) etc.

Ikke alle utvidelser kan integreres uten diverse problemer som f.eks. kompatibilitetsproblemer med den versjonen av Joomla som er installert, stabilitetsproblemer, manglende teknisk støtte osv. Dette er noe man må leve med ved et open source system.

Flere andre tilpasninger enn de områdene nevnt i dette kapitlet må også tilpasses.

6.2 Utviklingsprosess

Et rammeverk for systemutviklingsprosessen er forklart i begynnelsen av kap. 6.1.1. Design og utvikling går hånd i hånd. Under er presentert forskjellige arbeidsmetoder som kan benyttes på de forskjellige nivåene i systemutviklingen.

6.2.1 Egenutviklet løsning

6.2.1.1 Mål

Målene for PLU-veven er allerede definert i deres kravspesifikasjon for systemet, men disse kan om mulig endres senere som følge av endringer i utviklingsforløp, IKT-politikk, tilgjengelige ressurser eller andre faktorer. Overordnede mål for PLU-veven identifiseres ut fra overordnede samfunnsmessige behov, for eksempel behov for økt kompetanse for praksisveilederne.

6.2.1.2 Brukerbehov

Brukerbehov identifiseres i begynnelsen som de behov man ser kommer fram for å oppnå målet med systemet. Videre kan de spesifiseres ved å intervju og ha møter med de potensielle brukerne, før selve utviklingen av systemet har tatt til. Når man har fått på plass de mest grunnleggende behov, kan man lage en prototype (eller papirprototype) av systemet, som enkelte brukere kan få prøvekjøre eller teste (brukertester). Tilbakemeldinger og observasjoner fra disse testene kan benyttes for å ytterligere samle inn brukerbehov (og redesigne prototypen). Dette er en iterativ prosess, og er beskrevet bl.a. i [30]. Den samme iterative tankegangen for applikasjoner generelt er bl.a. beskrevet i [22].

Prototyping vil, grovt sett, bestå av følgende “lag” hvis prototypen er et eget system (ellers kan man bruke det utviklende systemet selv som prototype):

1. Mål

Få testet brukeropplevelse og få tilbakemeldinger på krav for systemet

2. GUI

Grensesnittet skal representere de funksjonene som brukerkravene tilsier

Ingen implementert funksjonalitet, kun skjermbilder og utseende

Skal etterligne virkelig funksjonalitet og brukbarhet

3. Interaksjon

Brukeren skal kunne bruke systemet om det var implementert grensesnittinteraksjon

Prototypen er begrenset avhengig av hvilken funksjonalitet eller sider av systemet som skal testes i prototypen.

6.2.1.3 Brukbarhet

Data om brukbarhet og brukeropplevelse kan samles inn gjennom intervjuer, både av hvordan brukere på forhånd forventer at systemet bør være, eller deres dårlige og gode erfaringer med bruk av systemet, og deres ønsker om forbedringer av brukbarhet og opplevelse. Utviklerne kan også foreta tester av systemets brukbarhet, for eksempel ved å benytte formelle brukbarhetstester som sikter på å luke ut feil i brukergrensesnittet, eller brukbarhetstester med brukerne som beskrevet i [30].

6.2.1.4 Funksjoner

Identifiseringen av funksjoner oppstår i starten ut fra de brukerbehovene man har samlet inn, eller vil samle inn. Men beslutningen for hvilke funksjoner man går for bestemmes også av faktorer som kostnaden av hver enkelt funksjon, om funksjonene passer sammen og utfyller hverandre eller om det er funksjoner som overlapper hverandre og derfor er overflødige, eller om noen funksjoner nedgraderer ytelsen på systemet, eller gjør systemet mer brukerfiendtlig og vanskeligere å bruke for enkelte brukere, og lignende.

Oppgavene vil bestå i å utvikle funksjoner (eller lete opp og eventuelt kjøpe ferdig tilleggsfunksjonalitet) etter de behov man har identifisert. Hvis type teknologi er spesifisert i et lavere lag for systemet, står man rimelig fritt til å utvikle (eller lete opp ferdige utvidelser) i de teknologiene som måtte ha det grensesnittet som muliggjorde integrering med kjernen i det nettbaserte systemet.

6.2.1.5 Grensesnitt og interaksjon

Grensesnitt for systemet og for hver enkelt funksjonalitet utvikles og testes gjennom bl.a. papirprototyper [30] eller formelle metoder. Det kan hende at ferdige tilleggsutvidelser har begrensede muligheter for å konstruere eget grensesnitt. Grensesnitt designes videre ut fra brukbarhet identifisert (kap. 6.2.1.3), kravspesifikasjon (kap 4.1) og designteori (kap. 4.3).

6.2.1.6 Sikkerhet

Data for å få oversikt over systemets sikkerhetstilstand samles inn ved gjentatt testing av webapplikasjonen. Det er anbefalt at utviklerne går aktivt inn for selv å prøve å utføre de vanligste angrepene på systemet ("Black Hat Testing"), for å teste om det er motstandsdyktig, i tillegg til å legge til rette for prosesser for å utføre forebyggende sikkerhetsarbeid ("White Hat"). Alternativt kan en annen gruppe utføre angrepene på oppfordring fra utviklerne. Det er ofte en fordel at andre utfører grundig testing, fordi egne utviklere kan føle for å beskytte og forsvare sin egen kode og eget produkt, med det resultat at svakheter ikke blir rettet opp [35].

6.2.1.7 Ytelse

Ytelse kan måles gjennom webserverstatistikk og tester av systemet (stresstesting) for å måle dets ytelse både under normal bruk, og ekstremt bruk med mange sideforespørsler.

6.2.1.8 Kostnad

Her vil man regne ut utviklingskostnader, ut fra tidsbruk, ressurser, utstyr og behov for folk og størrelsen på systemet og etter hvor mye av systemet som skal utvikles selv.

Arbeidet består også av utregning av kostnader på eventuelle tilleggsmoduler/-funksjoner som har en prislapp. Utregning av kostnaden ved integrasjon, design, programmering og/eller vedlikehold av tilleggsmoduler/-funksjoner.

Kostnadene av systemet som helhet kan beregnes ved å regne ut Return on Investment (ROI), dvs. det man kan forvente å tjene på systemet, minus innkjøpskostnader og vedlikeholds- og driftskostnader. Kostnadene på innkjøp av et eventuelt kommersielt system kommer fram av prislappen på systemet. I tillegg beregnes vedlikeholds- og driftskostnader av en eller flere personer i PLU eller andre økonomiansvarlige.

6.2.1.9 Vedlikehold og administrasjon

Vedlikehold er en omfattende og flerdimensjonal side av et system. Systemets vedlikehold omfatter ting som innholdsadministrasjon, videreutvikling, testing og påbygninger. Kostnadene ved vedlikehold kan, spesielt for et system man selv har utviklet fra bunnen av, stå for det aller meste av kostnadene ved et system. Det er velkjent at for egenutviklet programvare vil omkring 80 % av

kostnader og ressursbruk for systemet være knyttet til vedlikehold. Med kostnad menes kostnaden som kommer av ressursbruk som tid, folk, kompetanse og penger. For mer eller mindre ferdige systemer er vedlikeholdskostnadene ofte mindre, fordi man har ofte god dokumentasjon og støtte gjennom et støtteapparat i form av støtte fra en bedrift eller frivillige nettsamfunn. For et system som utvikles fra bunnen av, må man ofte være sitt eget støtteapparat, med mindre det åpnes for at andre utviklere får arbeide med systemet, og på den måten danne et samfunn som støtter hverandre i utvikling og vedlikehold. Et slikt samfunn kan likevel ta sin tid å oppnå, og man skal gjerne ha noe spesielt å by på for å tiltrekke seg andre medutviklere, og lykkes med en slik strategi.

Under er presentert viktige områder for vedlikehold og administrasjon av en nettbasert ressurs.

Innhold/artikler: Det trengs personer for å skape, vedlikeholde og oppdatere innholdet i webportalen. Dette er et ganske omfattende arbeid. Et av intervjuobjektene mente man trengte kanskje mer enn en heltidsstilling til dette arbeidet.

Drift og backup: Det kreves en kontinuerlig innsats for å vedlikeholde og drifte OS, HW og server(e) systemet skal kjøre på. Om mulig kan dette "outsources" til en driftsavdeling eller en tredjepart. Innholdet i systemet bør også tas backup av med jevne mellomrom. Her kan en kopi av databasen tas, eller data eksporteres til andre formater for lagring. Det kan være ønskelig å makulere digitale data etter en tid (spesielt forumdiskusjoner o.l.).

Diskusjonsforum: Forumet bør modereres av en eller flere personer, slik at useriøse eller uetiske innlegg blir sensurert. I tillegg bør noen være en pådriver og motivator for forumene. Dette kan kanskje til en viss grad erstattes med gulrøtter for de mest aktive brukerne på forumene, eller de som har fått flest score/poeng fra andre brukere på sine bidrag.

Kvalitetskontroll: Ressurser bør settes av til kontinuerlig kvalitetskontroll av innholdet på webportalen. Dette inkluderer f.eks. kvaliteten på innhold i wikidatabasen, der flere brukere normalt kan redigere inn innhold [28].

Teknisk administrasjon av portalen: Det bør ansettes en person som kan holde orden på selve det nettbaserte verktøyet (portalen). Dette innebærer å sørge for dets installasjon, konfigurering, oppdateringer og drifting. Et nettbasert verktøy må forventes å kontinuerlig oppdateres eller fikses feil på. En slik person bør dessuten sørge for problemfri integrasjon mot andre, eksisterende

systemer.

6.2.1.10 Teknologi

Dette er et designmessig valg, og kommer an på hva utviklerne av systemet finner som den beste teknologiske totalløsningen.

6.2.2 Ferdig løsning

Utviklingsprosessen for en ferdig løsning blir betydelig kortere enn for en egenutviklet løsning, men man må ta høyde for at mye tid går med til tilpasning av systemet. De samme punktene som for utviklingsprosessen i kap. 6.2.1 kan likevel delvis anvendes; det tar bare kortere tid. Spesielle merknader ved de forskjellige punktene i en ferdig løsning er for hvert enkelt punkt:

Mål: Samme som egenutviklet.

Brukerbehov: Samme som egenutviklet

Brukbarhet: Samme som egenutviklet

Funksjoner: Her må man forvente å finne bare tilnærminger til ideelt ønskede funksjoner. Man kan også selv utvikle utvidelser, men det kan tenkes at egenskapene ved kjernesystemet setter begrensninger på hva som er mulig. Hvis kjernesystemet i tillegg er proprietært eller har lukket kildekode, vil det være vanskelig å tilpasse.

Grensesnitt og interaksjon: Her kan man forvente en tilnærming, men normalt vil et godt nettbasert verktøy tillate stor tilpasningsdyktighet hva gjelder grensesnittet.

Sikkerhet: Dette er mest prispågående kjernesystemet, men også utvidelser må ha sin sikkerhet.

Ytelse: Som med egenutviklet, men man står ikke så fritt til å modifisere ytelsen.

Kostnad: Vil være annerledes enn for egenutviklet. Først og fremst faller utviklingskostnader av kjernesystemet bort, og erstattes med en anskaffelseskostnad (for proprietært system).

Vedlikehold og administrasjon: Samme som for egenutviklet, bortsett fra at man ikke trenger å vedlikeholde koden i kjernesystemet.

Teknologi: Man kan ikke endre teknologien brukt i kjernesystemet, og er ofte bundet til bestemte teknologier også i utvidelser.

6.3 Den organisatoriske siden

Den organisatoriske siden av design og utvikling av nettbaserte verktøy er viktig å forstå, siden organisasjonen som skal ta i bruk de nettbaserte ressursene, i case studiet er det PLU, vil måtte ta hensyn til de organisatoriske endringene og tilpasningene nødvendig for en vellykket innlemmelse av nye nettbaserte verktøy. Endel interessante synspunkter hva angår PLU som organisasjon ved innlemmelse av nye nettbaserte ressurser kom fram i intervjuer. Nedenfor er sitert noen av synspunktene.

Det var ganske stor enighet om at det var en stor utfordring å drive et nettbasert forum, og at det krevde involvering fra lærere og fagpersoner. Det var likevel enighet om at dette var viktig å satse på:

- *"Læreren må være veldig på hugget tidlig i semesteret for å motivere til aktivitet (i diskusjonssider på nett)."*
- *"Man kan kanskje ha et åpent diskusjonsforum, men bare periodevis siden det er mye arbeid med å vedlikeholde."*
- *"Det er en absolutt nødvendighet at fagpersonen er en bidragsyter i nettbaserte diskusjonsfora."*
- *"Erfart at det er vanskelig å organisere diskusjonsforum på nett. I tvil om i hvilken grad læreren skal være med. Studentene skal diskutere med hverandre."*

At det ellers ville være ressurskrevende å drive webportalen var det også rimelig stor enighet om:

- *"På lang sikt håper jeg at det blir en skikkelig satsing. Det koster veldig mye ressurser, og må oppdateres kanskje hver dag. Det må skje noen ting hele tiden."*
- *"PLU har undervurdert hvor mye ressurser som må settes inn i et slikt webportal-prosjekt, i form av tid. Man må kanskje ha mer enn en stilling på heltid for å få det til å fungere."*
- *"Nettbaserte tjenester stiller jo kolossale krav til den som skal legge ut ressursene."*
- *"Prosjektet vil kreve strikt organisering."*
- *"Man må finne ut av hvor mange som skal være med å drifte/administrere en slik webportal. Hvor mange må læres opp? Dette er litt uavklart ennå. Hvis noen slutter, må man ha noen i reserve."*

- *"En utfordring er om PLU skjønner at dette krever oppfølging i starten. Utradisjonelt å tenke nettbasert i lærerutdanning."*

Foreløpig arbeidet de i en arbeidsgruppe med PLU-veven, som beskrevet i noen av sitatene fra intervjuene:

- *"Vi arbeider med PLU-veven i en arbeidsgruppe, der vi møtes en gang i uka. Andre involveres når det er behov for det. Én person står for faglig utvikling, og det er også viktige personer som har med FoU å gjøre involvert."*
- *"Ledelsen har nedsatt en arbeidsgruppe med prosjektet. Mindre beslutninger tas i arbeidsgruppa. Når det gjelder penger må vi forholde oss til ledelsen. For det tekniske og faglige innhenter vi ekspertisens råd."*
- *"Vi har ikke samarbeidet med andre lærerutdanningsinstitusjoner, men med flere på NTNU."*

Det ble også uttrykt bekymring om hvorvidt man hadde tid og ressurser til å administrere systemet:

- *"Jeg har små forventninger til webportalen. Jeg tror ikke den er godt nok forankret i organisasjonen her. Jeg tror ikke de ansatte er motiverte nok til å bidra. Men tror bare."*
- *"Jeg er med når jeg har anledning, men det er lite."*
- *"Folk er veldig opptatt. De er engasjerte, men har liten tid."*

Intervjuobjektene tenker om den videre organiseringen av prosjektet:

- *"Man må være litt nøye med å følge opp den sosiale biten av læringsfellesskapet hvis mye legges til nettbaserte verktøy."*
- *"Vi vil ikke ansatte utviklere på det nåværende tidspunkt, ikke så lenge det praktisk lar seg løse på andre måter."*
- *"Mulig med en skiftløsning, der ansatte bytter på å legge ut informasjon og vedlikeholde innholdet."*
- *"Vi har en liten informasjonsbit å gjøre på PLU."*
- *"Det er diskutert mye med deltagerne om hva de skal fylle portalen med. Skal de legge inn stoff selv direkte, eller skal det gå via de på PLU?"*
- *"Det er viktig at webportalen får et fotfeste i organisasjonen. Teknologi og organisasjon bør gå hånd i hånd."*
- *"Prosjektet vil kreve et samarbeid mellom de faglige pedagogene på den ene side og IT-folk*

på den andre siden. Det er ikke bare IT-folk som skal pushe dette frem."

- *"Skeptisk til å sette studenter til å konstruere portalen, fordi disse forsvinner etter en stund. Det mangler kontinuitet. Det bør gjøres av et firma eller av agenter."*
- *"Drift kan ikke videreutvikle fordi de har mye å gjøre."*
- *"Webportalen bør være kvalitetssikret av webdesignere."*

Om feilvalg i organiseringen av prosjektet og nettbaserte ressurser generelt:

- *"Man kunne overlatt de tekniske løsningene mer til teknisk ekspertise i starten av PLU-veven-prosjektet."*
- *"Lærerutdanningen har ikke vært flinke nok til å ta IKT på alvor."*
- *"Prosjektleder ble sittende helt alene med prosjektet etterhvert."*

Hvilke organisasjonsmessige konsekvenser man mente nettbaserte ressurser kunne ha:

- *"En webportal kan bidra til økt kunnskap i forhold til bruk av IKT i etter- og videreutdanning av lærere. Lærere er litt skeptisk til å legge stoff ut åpent, men noe **skal** legges ut åpent."*
- *"Vi har ennå ikke (før PLU-veven) noe prosjekt som kan virke samlende på institusjonen."*
- *"Et webportal-prosjekt kan ha den negative effekten at det kan ta oppmerksomheten bort fra andre ting ved tilrettelegging av etter- og videreutdanning av lærere. Man kan bli for enøyd på IKT."*
- *"IKT gir oss større fleksibilitet og tilgjengelighet, pluss at man på den måten også kan få praksisskolene til å ta i bruk mer IKT."*
- *"En webportal egner seg for kompetanseheving av praksisveilederne fordi man kan utvikle en konkret nettutdanning, som gjør at lærerne kan være i jobb og samtidig studere."*
- *"Håper NTNU-videre forstår at denne typen systemer kan være med å effektivisere."*

Ellers noen aktuelle sitater om organisasjon eller organisering:

- *"Vi tok kontakt med SVT-Drift for å diskutere hva de kunne bidra med. De sa de kunne gjøre noe småtteri. De har bidratt med råd og vært veldig ivrig og behjelpelig og god til å kommunisere. Vi har hatt kontakt med 2-3 personer derfra. De holdt kurs i SharePoint. Drift mente en Escenic-løsning ikke var enkel; man måtte bl.a. ha brukere i BDB."*
- *"Vil ikke si at prosjektet har vært ressurskrevende så langt."*

- *"En utfordring er at initiativtakeren sitter i Tromsø. Litt vanskelig å skape engasjement."*
- *"Det verste som kan skje er at den ikke blir brukt."*

Som man kan observere ut fra intervjuer med folk fra PLU selv, er gjengangstonen at organisasjonen også må involveres i prosjektet, hvis ikke vil de ikke lykkes. Noen er skeptiske, og tror ikke PLU-veven er godt nok forankret i organisasjonen, eller man har undervurdert de organisasjonsmessige ressursene som kreves. Diskusjonsforumet blir spesielt tatt fram som noe som krever en innsats fra PLU å administrere og vedlikeholde. Forslag til tiltak som kan forebygge uheldige organisatoriske overraskelser ved innføring eller utvikling av et IKT-system er bl.a.:

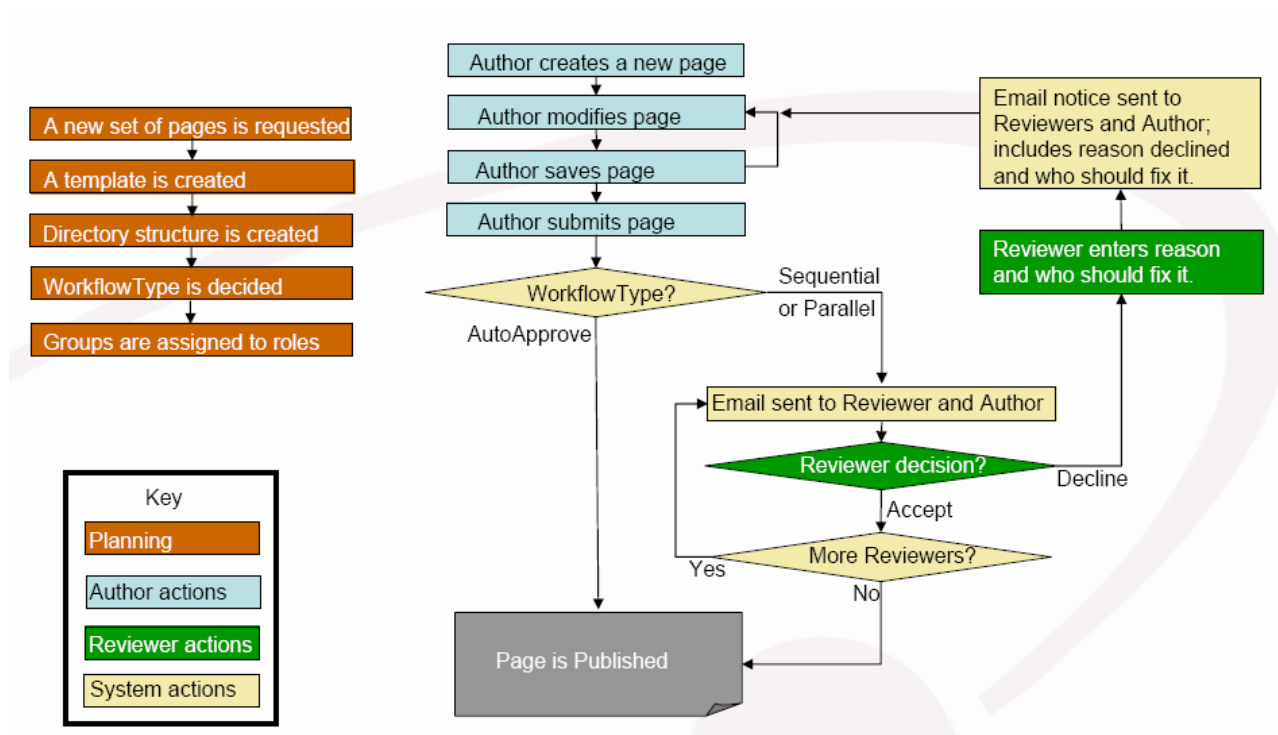
- De behov brukerne og organisasjonen har som systemet er ment å skulle dekke (de forretningsmessige behovene), må spesifiseres klarere, slik at utviklingen av systemet, eller valget av system, ikke preges av vaghet.
- En øvre grense (om mulig) for de ressurser PLU kan sette av for utvikling og/eller vedlikehold av systemet. Dette gir en ramme man kan holde seg innenfor, slik at sannsynligheten for uheldige kostnadsoverskridelser forminskes.
- Bygge på erfaringer fra andre, lignende prosjekter i lignende organisasjoner. Gjerne hold en kommunikasjonslinje med slike.
- Innføre systemet gradvis, slik at man unngår et "stort smell" om det går galt. Prototyping er et stikkord her.
- Lage en prioriteringsliste over hvilke sider av systemet som er viktigst.
- Eventuelt ha noen lokkemidler for bruk av systemet.
- Gi tilstrekkelig informasjon om systemet, hva som er hensikten med det, og en lettforståelig brukerveiledning.

For å bruke et CMS effektivt trengs det i følge [46] et dyktig team bestående av en designer, programmerer og en administrator. Typisk vil da en designer gjøre tilpasninger på utseendemaler og det visuelle (HTML/css), programmereren vil typisk utføre skripting (ASP, JavaScript e.l.), og administratoren vil behandle og organisere innholdet. Ifølge Mehringer vil man ha behov for å dekke følgende roller for et CMS:

- Administratorer
- Moderatorer
- Ressursbehandlere
- Template designere

- Abonnenter
- Forfattere
- Redaktører

Figuren under (figur 30) viser en typisk arbeidsflyt for publisering av en side i et CMS, og inkluderte roller [46].



Figur 30: Arbeidsflyt ved publisering av side i et CMS (Mehringner, 2005).

En forfatter lager en ny side, modifiserer den og lagrer den et antall ganger, før den blir satt som publisert. For noen systemer vil siden da bli direkte publisert (automatisk godkjent), men man ser av figuren at man også kan la anmeldere se over sida før den endelig publiseres, gjennom en sekvensiell eller paralell arbeidsflyt. Her blir anmeldere og forfatter varslet, og anmelder ser over sida og enten aksepterer eller avslår publisering. Ved avslag oppgir anmelder grunn til avslag og foreslår hvem som skal "fikse" sida. Ved aksept spør systemet om flere anmeldere skal se over sida eller om den skal endelig publiseres. Ved publisering av hyppig oppdatert innhold vil trolig en arbeidsflyt med automatisk godkjenning være mest relevant for PLU, mens en arbeidsflyt med godkjenning eventuelt er relevant for mer kritiske sider i systemet.

I kap. 3.1.5 ble vanlig motstand mot innføring av CMS i en organisasjon gjennomgått. Ut fra disse erfaringene burde man være litt føre var i sin bruk av, utvikling og innføring av IKT (spesielt CMS) i en organisasjon. Følgende områder er viktige å se nærmere på for å forhindre større organisasjonsmessige problemer. Tiltakene bygger på (<http://webdesign.about.com/library/weekly/aa022502a.htm>):

- Det nye systemet bør være minst like raskt som de ansattes bruk av andre, lignende systemer i deres organisasjon er. For PLU-veven betyr det like raskt som andre nettbaserte verktøy lærere, PLU-ansatte, praksisveiledere og lærerstuderer benytter seg av.
- (Videre-) utviklere kan lokkes til å arbeide på prosjektet ved å opplyse dem om at det er bruk for dem til f.eks. dynamisk skripting og design, selv om mye av systemet allerede er ferdigprogrammert.
- Prosessen med publisering og godkjenning av innhold som skal publiseres må kartlegges på forhånd, slik at forvirring med hensyn på innholdsskaperes oppgaver unngås.
- Innholdsskaperne må forberedes på et stort ansvarsområde, og at det blir en læringskurve å overkomme i begynnelsen når man tar i bruk systemet.
- Den langsiktige lønnsomheten ved IKT-ressursen, sammenlignet med kostnadene, kan vise ledere hvor mye midler man kan tillate seg å sette av til kjøp eller utvikling av et system.

7 Diskusjon

Diskusjonskapitlet tar først for seg en diskusjon rundt relevansen av de ferdige løsningene som ble evaluert i rapporten i forhold til PLU-veven, for deretter å diskutere relevansen av egenutviklede løsninger i forhold til PLU-veven.

7.1 *Sammenligning og relevans av ferdige løsninger*

I dette kapitlet gjennomgås en sammenligning av ferdige løsninger, og strategier basert på disse i forhold til utviklinga av nettbaserte ressurser (plu-veven) for bruk i etter- og videreutdanning av lærere.

7.1.1 **SharePoint Services eller Joomla?**

I kapittel 5 ble det gjennomgått undersøkelser av eksisterende ("out-of-the-box") nettbaserte

verktøy som var relevante for bruk i etter- og videreutdanning av lærere og som grunnlag for pluveven. To systemer ble evaluert etter Robertsons skjema for å evaluere CMS [6]. I denne seksjonen sammenlignes disse to systemene mot hverandre, og fordeler og ulemper med dem blir diskutert, bedømt ut fra PLUs behov for nettbaserte verktøy. Sammenligningen av systemene vil først og fremst foregå innenfor følgende områder, etter skjemaet til [6]:

- Innholdsskaping
- Innholdsbehandling
- Publisering
- Presentasjon
- Kontrakter og forretning

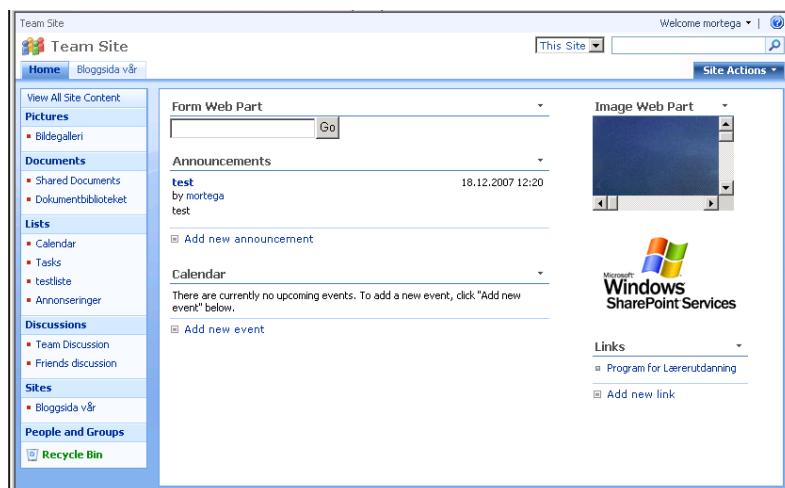
Innholdsskaping

Innholdsskaping er en viktig ressurs for PLU-veven. Skaping av HTML-sider for visning i portalen skjer i WSS gjennom å opprette nye interne websider v.h.j.a. den integrerte editoren. Dette fungerer fint, men det er begrensning på hvor man kan gjenbruke dette innholdet i portalen. Selv om man kan lage sine egne innholdstyper som kan gjenbrukes i portalen, så er disse innholdstypene bare maler for metadata til elementer, maler for dokumenter i en samling, og maler for arbeidsflyt for elementer o.l. Dette kan komme godt med i en mer organisasjonsmessig kontekst der man bruker systemet for å behandle organisasjonens dokumenter, men er ikke særlig aktuelt for PLU-veven. PLU-veven er ikke først og fremst definert for å oppfylle dokument- og informasjonsflyt innenfor PLU, men som et samlingspunkt og sosial arena med mer *presentasjon* og utveksling av kunnskap. Oppretter man innhold i WSS er det f.eks. begrenset hvor det kan presenteres i portalen. Det sentrale innholdselementet i Joomla er artiklene, som skapes i en WYSIWYG-editor. Innholdet kan brukes og presenteres hvor som helst på portalen, noe som passer PLU-veven bedre.

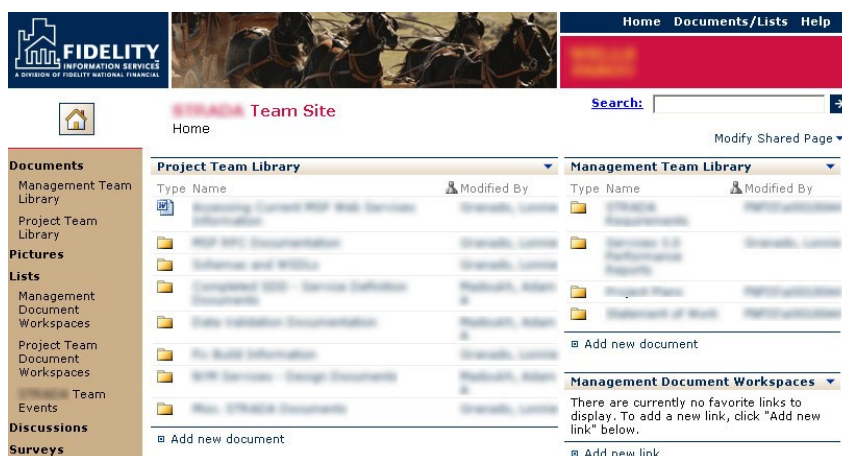
Joomla har ingen kjente låsemekanismer for flerbruker-skriving, mens WSS bruker *check-in* og *check-out*. Igjen viser dette at WSS er mer rettet mot tett samarbeid hva gjelder innholdsskaping og dokumentflyt innen en organisasjon, mens Joomla er et system som er mer rettet mot presentasjon av innhold. Presentasjon av innhold er noe som var ønsket i PLU-veven, og er noe av begrunnelsen bak argumentet om at systemet skulle være "åpent", som en vanlig nettside. Man kan totalt endre utseende og layout (og endre på navigasjonsmenyer osv.) i websider som bygger på WSS ved å benytte kompatible webredigeringsverktøy som SharePoint Designer for å redigere nettstedets

Master Pages [43]. Likevel endrer ikke dette på at behandlingen av innhold i systemet er rettet mot *organisasjon*, og ikke først og fremst presentasjon.

Se figur 31 og 32 for et eksempel på hva bare å endre et tema kan gjøre med en WSS-side.



Figur 31: Standard tema for en SharePoint-side.



Figur 32: En SharePoint-side tilpasset ved bare å endre tema. (<http://www.heathersolomon.com/resume/strada.aspx>)

Begge systemene er lettvinde med tanke på innholdsskaping, og er logiske og konsekvente når man har forstått hvordan de fungerer. Konseptet med lister og innholdstyper i WSS kan være noe

vanskelig å forstå til å begynne med.

Innholdsbehandling

WSS scorer høyt på versjonskontroll i forhold til Joomla, selv om systemene støtter versjonskontroll på noe forskjellige typer innhold. At WSS scorer bra her, illustrerer at WSS vektlegger informasjonsflyt i samarbeidende prosjekter, f.eks. arbeidsflyt i en organisasjon eller samarbeid i et programutviklings-team, noe som krever streng versjonskontroll.

WSS er sårbar for forandringer i organisasjonen hvis denne ønsker å gå bort fra Microsoft-løsninger som f.eks. Microsoft Office. WSS er tett integrert med denne, og fungerer bra med det. Bl.a. bruker man Microsoft Word som dokumentformat. I Joomla foregår hele arbeidsflyten innenfor systemet, og systemet er derfor mindre sårbar mot slike organisasjonsforandringer. WSS krever dessuten å kjøre på spesielle Windows-plattformer, mens Joomla er vellykket testet på de 3 vanligste OS'ene på plattformene Windows, Linux og Mac OSX. Endringer i en organisasjons bruk av plattformer kan få størst konsekvenser for et WSS-basert system.

Tilgangsrettigheter og tilpasning av dette er mer avansert i WSS enn i Joomla, med tanke på at man ikke kan definere sin egen tilpassede brukergruppe i Joomla. Det finnes i stedet 7 forhåndsdefinerte brukergrupper med forskjellige tilgangsrettigheter, men en funksjon for å lage sine egne kan allerede ha blitt implementert (denne opppgaven undersøkte ver. 1.5, men det var planlagt en mulig funksjon for dette i neste versjon av Joomla) [14]. Hvilke systemer som faktisk er mest sikre er vanskelig å si sikkert. Begge systemer har innebygd sikkerhet mot uautorisert tilgang til administrasjonsgrensesnittet. Videre vil også sikkerheten for webserveren, databasen og OS'et man kjører på spille inn. For videre utredninger om sikkerheten på disse henvises det til andre studier.

Publisering

Hovedankepunktet mot WSS når det gjelder publisering er at systemet kun støtter publisering av hele grensesnittet til HTML (web) og spesifikt innhold til MS Office-spesifikke formater som Word- og Excel-filformat. PDF støttes for eksempel ikke. Innhold for presentasjon i websider innenfor systemet er i .NET content file format. Systemet kan fungere fint for PLU, men er da avhengig at man går for en helhetlig Microsoft- og .NET-løsning, og er sånn sett et system som passer best på innsiden av en organisasjon med en fastbestemt HW- og SW-løsning (basert på

Microsoft). Så snart man går ut på nettet, har man straks mange forskjellige plattformer og typer klienter å hankses med.

Joomla støtter publisering av innhold/artikler til web i HTML-format, utskriftsformat og til mer plattformuavhengige formater som PDF.

Både Joomla og WSS må ty til programmering for å totalt forandre en nettsides utseende. Men det finnes en rekke utseendemaler for Joomla som totalt forandrer nettsidenes utseende, så man kan unngå programmering om man finner en passende utseendemat. Det finnes ferdige "themes" for WSS som man kan laste ned uten å programmere, men disse endrer bare layoutens skinn med forskjellige bilder og farger med mulighet for å gjemme enkelte kontroller. Man må selv endre portalens Master Pages for å totalt endre sidenes utseende [43].

Presentasjon

Begge systemene scorer ganske bra på brukbarhet og lettforståelig grensesnitt hva gjelder administratorsidene, men brukere av begge systemene må påberegne seg noe tid i starten for å sette seg inn i logikken til systemene. Når man har overkommet dette, er begge relativt enkle å forstå. Begge systemer har likevel avanserte funksjoner og parametre som kan være vanskelig for noviser å forstå. Kanskje gjelder dette spesielt Joomla, som har en rekke avanserte parametre på artikler og innhold f.eks. Når det gjelder de eksterne sidene, har Joomla de største integrerte mulighetene for å tilpasse disse sidene, mens WSS må ty til webredigeringsprogram.

WSS scorer bedre enn Joomla hva gjelder tilgjengelig for funksjonshemmede, og kompatibilitet med visse webstandarder. Men betingelsen er at din forhandler av WSS må tilby denne funksjonaliteten i systemet du får fra dem. At WSS oppfyller webstandarder bygger mye på ASP.NET 2.0, som er et språk som ble designet med den tanke å oppfylle standarder for tilgjengelighet og XHTML-spesifikasjonen [44]. Men også Joomla kan bruke valid XHTML, det kommer bare an på implementeringspersonellet å påse at dette er godt nok kodet. Joomla oppfyller også standarder for god tilgjengelighet, men har foreløpig mye tabelldrevet kode, noe som senker tilgjengeligheten for bruk av f.eks. skjermlesere som støtteverktøy for blinde.

Joomla er langt mer kompatibel med forskjellige typer nettlesere enn det WSS er. WSS er bare fullt ut kompatibel med Internet Explorer, mens Joomla skal være kompatibel med de mest vanlige nettleserne. Skal plu-veven bygge på WSS må følgelig brukerne benytte Internet Explorer for full utnyttelse av sidene. Dette er selvfølgelig avhengig av hva man tilbyr på sidene av funksjonalitet (spesielt webdeler fungerer bare tilfredsstillende i Internet Explorer), og denne inkompatibiliteten gjelder først og fremst administratorgrensesnittet. Men flere personer er tenkt å kunne være

administratorer som bidrar med innhold i PLU-veven. Da må disse bruke Internet Explorer for full utnyttelse av sidene. Joomla lider ikke av samme problemet, og synes på dette området å være bedre stilt for plu-veven enn WSS.

Kontrakter og forretning

Anskaffer man WSS kan man tegne en supportavtale med Microsoft, og er sikret endel støtte for systemet. Joomla har ingen slik støtte, men det finnes tredjeparter som kan påta seg rollen med å tilpasse Joomla-websider, som nevnt i kap. 5.

At WSS har et mer snevert krav for maskin- og programvareressurser enn Joomla, trenger ikke spille så stor rolle for PLU-veven sin del, siden dette kan driftes fra én plass, eller fra én server. PLU har jo også anskaffet en server som oppfyller de fleste kravene (bortsett fra å ha en 2,13 Ghz prosessor mens kravet er en 2,5 Ghz prosessor minimum).

Av inngående programmeringskunnskaper for systemet så vil HTML, CSS og PHP være tilstrekkelig for detaljert og dyp tilpasning av Joomla til å konstruere plu-veven. For en WSS-basert portal vil kunnskaper om HTML, CSS og ASP.NET stort sett være tilstrekkelig for en detaljert og dyp tilpasning av systemet til å konstruere plu-veven. I tillegg kommer administrasjon og drift for begge av systemene.

Kostnader trenger ikke spille så stor en rolle for PLU, siden de får begge systemene nær gratis. Hvis PLU også ønsker å benytte Microsoft Office SharePoint Server (MOSS) for å bygge PLU-veven, påbeløper det lisenskostnader.

Har PLU ønsker om å utvide PLU-veven, har WSS dokumentert god skalerbarhet gjennom å bruke serverfarmer.

Integrasjon mot andre systemer vil være problematisk for et WSS-basert system hvis klientene kjører på andre plattformer enn Windows.

Det finnes en rekke referanser om vellykkede implementeringer av WSS (vel og merke noen løsninger i samvirke med f.eks. Microsoft Office SharePoint Server og SharePoint Designer). Endel av referansene er bruk av denne løsningsstrategien innenfor universiteter. Dette kan være sammenlignbart med plu-veven, men spørsmålet er om det er slik at disse systemene lettere fungerer innenfor en kontrollert organisasjon (der man kan bruke produkter innenfor en spesifikk plattform mer uniformt) enn mellom brukere som er mer geografisk adspredt (ikke bundet sammen i en organisasjon i så sterk grad).

Noen vellykkede referanser for bruk av Joomla inkluderer de offisielle sidene for Joomla!. Dette er sider som ikke befinner seg innenfor en kontrollert organisasjonskontekst, siden folk fra hele verden bruker ofte dette nettstedet for å laste ned Joomla og finne svar på spørsmål om systemet osv. Selve systemet er jo open source, og er ikke begrenset til en fysisk organisasjon i så måte. En annen nettside gjort med Joomla er www.urth.tv. Dette er en sosial samlingsarena for folk over hele verden.

Se kapittel 8 for en konklusjon på evalueringen og sammenligningen av Joomla! og Windows SharePoint Services 3.0.

7.1.2 It's Learning et alternativ?

It's Learning ble evaluert i kapittel 5.1.3, der det ble evaluert etter en kriterieliste framsatt av [2] for evaluering av LMS. Her vil relevansen av It's Learning som et eventuelt grunnlag eller en erstatte for PLU-veven, diskuteres.

It's Learning's konsekvente brukergrensesnitt med norsk språk gir den et godt utgangspunkt som et system som kan brukes for PLU. Ulempen med grensesnittet er at det ikke kan tilpasses i stor grad etter de designønsker PLU måtte ha for websidene. Menyenes plasseringer er fastlåste. Det samme gjelder visuell gruppering av informasjon og funksjoner i grensesnittet. I tilleggsmodule "Bibliotek" kan man presentere faglig innhold på en mer presentasjonsrettet måte, akkurat som i en webside, men man er ikke helt fri i sin utformelse av grafikk, farger og layout. Man velger mellom et sett forhåndsdefinerte grafiske maler, eller man kan lage sin egen forhåndsdefinert mal ved å endre på CSS-kode for en mal.

It's Learning er dessuten et typisk LMS som fokuserer på skolerelaterte funksjoner som eksamensbesvarelser, karakterbøker og læringsplaner. Dette er stort sett uaktelt for bruk i PLU-veven. Da har man også lite nytte av systemets muligheter for å importere og eksportere innhold for bruk i disse funksjonene, samt verktøyets tilfredsstillende av e-læringsstandarder. Dette gjelder også andre, lignende LMS.

Siden PLU har kjøpt seg egen server, kan kanskje det faktum at det må leies serverplass hos IT Solutions være en ulempe akkurat nå. Bortsett fra det så vil driftsbehovet tas hånd om av andre enn PLU selv, og IT Solutions fronter med høy oppetid og godt vedlikehold.

En rekke medielementer av interesse for PLU-veven kan likevel inkluderes og presenteres i systemet, som lyd, bilder, video og animasjon. Og interaksjonsbasert innhold som f.eks. diskusjon,

konferanse, oppgaver, tester og undersøkelser kan også være av interesse for PLU-veven. Eportefølge-funksjonen kan være nyttig å inkludere som en del av profilen til PLU-vevens fremtidige brukere. Av relevanse for PLU-veven er også It's Learnings tilgangsrettighetsfunksjoner, eller profiler, som kan mange tilpasningsmuligheter.

Sett helhetlig på It's Learning så produserer man ikke åpne og tilgjengelige (for allmennheten) nettsider med dette verktøyet. Man må ha brukernavn og passord, og logge inn, for å kunne se innhold som er lagt ut. Bare dette alene kan, som påpekt av intervjuobjekter, være nok til å avskrive It's Learning som en potensiell plattform for PLU-veven.

7.2 Vurdering og relevans av egenutviklet løsning

En vurdering og relevans av egenutviklet løsning vil inkludere en vurdering av både studentgruppas prototype samt mitt eget forslag til egenutviklet løsning.

7.2.1 Studentgruppas løsning

En evaluering og test av studentgruppas løsning ble foretatt i oppgavens forstudie (kap. 3.2.4). Ved avslutning av gruppas prosjekt, var en prototype ferdig. Denne er for begrenset til at den kan tas i bruk som en nettbasert ressurs for PLU eller praksisveilederne. Den fungerer slik som den er, men tilfredsstillende ikke helt kravene for PLU-veven. I tillegg innehar ikke portalen alle de ønskede funksjoner som PLU-veven ville ha.

Studentgruppas løsning har noen fordeler sammenlignet med den løsningen som PLU etterhvert gikk for høsten 2007 (og som ble brukt i et FoU-kurs januar 2008). For det første har den et mer ryddig og stabilt grensesnitt enn WSS-utgaven av PLU-veven har. WSS-utgaven av PLU-veven hadde endel overflødige lenker og navigasjonsvalg. Studentgruppas løsning lider ikke av samme problemet. Noe som er savnet er en funksjon som gir lett og rask oversikt over portalens innhold, f.eks. en *sitemap*. Det kunne også vært ønskelig med en toppmeny på sida. Spesielt gjelder dette hvis sidene skal ligge under NTNUs sider, da disse bruker en standard navigasjonsstruktur der toppmenyen brukes for global navigasjon [17]. En kolonne helt til høyre som kan brukes til tilleggsinformasjon og kontaktinformasjon o.l. kunne også vært ønskelig. Så langt tilbyr ikke studentgruppas løsning mulighet for 3 kolonner, eller noen muligheter for å tilpasse dette etter ens ønsker (uten at man må redigere koden). Administrasjonen av sida foregår dessuten i et eget program som må lastes ned og installeres, og innfylling av innhold i portalen forutsetter noe

kjennskap til HTML.

Studentgruppas løsning står og faller mye på muligheten for videreutviklere å kunne bygge videre på løsningen. Dvs. om den lett lar seg utvide både med tanke på funksjonalitet, brukergrensesnitt og interaktivitet. Spørsmålet da er hvorvidt koden og dokumentasjonen er god nok for at det skal gå problemfritt for etterfølgende utviklere. Savnet i dokumentasjonen var et mer lettlest DFD-diagram, dvs. et DFD-diagram på et høyere nivå. Gruppen laget kun et DFD-diagram på nivå 3, og dette ble nokså stort og uoversiktlig. Dette var kanskje grunnen til at kun et utsnitt av DFD-diagrammet ble presentert i rapporten. En videreutvikler kunne kanskje ønsket seg en mindre og mer oversiktlig versjon av dataflyten, i for eksempel et DFD-diagram på nivå 2 eller 1. Koden er ikke uttømmende evaluert, men den skal være modulær og gi et godt grunnlag for videreutvikling, i følge studentgruppa selv [16]. Bl.a. mener man at videreutviklere ikke trenger å bruke den teknologien dem selv har brukt i portalen. Gruppen har brukt C#, ASP.NET og .NET, og her kan f.eks. videreutviklere bruke Java. Det er også nylig kommet flere valgmuligheter for integrasjon mellom Java og .NET [48]. WebSphere MQ og ActiveMQ transport kan nå brukes for å bygge tverrfunksjonelle webtjenester mellom Java og .NET. Likeledes er pålitelig sending av beskjeder i .NET utvidet. I tillegg til MSMQ kan nå både WMQ og Active MQ brukes som pålitelig transport for .NET-webtjenester. En mer utfyllende vurdering av teknisk integrasjon er ikke foretatt i oppgaven.

Portalens grafiske utseende skulle ligge godt til rette for å kunne endres. Her kan man f.eks. endre Master Pages (som forøvrig er et konsept som muliggjør å endre utseende på flere sider ved kun å endre på én side) og CSS.

Når det gjelder studentgruppas påstand om at systemer med åpen kildekode er dårlig dokumenterte, har de ingen dokumenterte referanser til sin påstand, og den må derfor betraktes som noe løs. Hvor godt et system er dokumentert kommer i stor grad an på hvem som utvikler systemet og skriver dokumentasjon. Hvorvidt systemer med åpen kildekode generelt sett er dårligere dokumenterte enn de med lukket kildekode er et studie i seg selv som ligger utenfor denne rapporten. Ved åpen kildekode har man jo i tillegg tilgang til kildekode med kommentarer for slike systemer, og dette i seg selv er på sett og vis dokumentasjon. Påstanden om at åpen kildekode vil gjøre det vanskeligere for etterfølgende utviklere kan derfor betraktes som veldig tynn, og det skulle i utgangspunktet ikke være noe forskjell fra systemer med lukket kildekode her. Gruppen mente at ferdige ("out of the box") løsninger gjorde det mulig å konstruere produkter uten å vite hvordan programmer og funksjoner i løsningen fungerte. Dette er riktig, men det finnes også svært mange slike systemer som er basert på åpen kildekode (<http://www.la-grange.net/cms>) (<http://www.opensourcecms.com/>).

Valget om hvorvidt den nettbaserte ressursen skal være basert på åpen eller lukket kildekode bør kanskje derfor være basert på andre argumenter, og det vil likevel være opp til PLU eller de som de ansetter til å se på dette og foreta den endelige beslutningen i framtiden.

Studentgruppa argumenterte også for å lagre data på serveren istedet for lokalt på brukernes datamaskiner. Dette er i tråd med de fleste ferdige (out of the box) WCMS (som f.eks. lagrer i en database på serveren). Spørsmålet er også om PLU vil drifte denne serveren selv eller om en tredjepart vil ta seg av serverdriften. Dette kan spare PLU for driftskostnader, men kan frata dem endel tilpasningsmuligheter. I det hele tatt er en løsning med webtjenester en god løsning, siden kommunikasjonen mellom klient og tjener er bygget på spørring og respons, med data som kan leses på mange ulike plattformer¹⁵.

7.2.2 Egenutviklet løsning generelt

Den egenutviklede løsningen som ble presentert i kapittel 6 er relevant for PLU om de anskaffer seg utviklere for å utvikle løsningen. Design- og utviklingsforslaget der kan også brukes for å videreutvikle studentgruppas løsning. Man har allerede deres prototype på plass, og man trenger bare bygge videre på denne. Løsningen presentert i kapittel 6 er basert på teori, tilbakemeldinger fra brukere (intervjuer og spørreskjemaer), kravspesifikasjonen fra PLU og NTNUs designprinsipper (på kort sikt). Løsningen er også bygd på de undersøkelser av nettbaserte verktøy som ble foretatt i mitt studie og skulle derfor gi et godt forslag til en langsiktig løsning. Forslaget burde derfor være ideelt for PLU, selv om det var vanskelig på dette tidspunkt å se hele veien framover (siden PLU ikke har det helt klart for seg hvordan systemet skulle være). Funksjonalitet vil kunne legges til om PLU skulle ønske dette. Man kan også benytte designforslaget som en rettesnor for tilpasning av en ferdig løsning, siden det ikke beskrives *hvordan* et slikt design skal oppnås. Finner man en ferdig og tilpasningsdyktig hylleløsning, det være seg et open source produkt eller et kommersielt produkt, kan denne tilpasses det foreslåtte designet i kapittel 6. Om man går helt for en løsning tilsvarende å programmere systemet fra bunnen av, kan man også benytte utviklingsidéene presentert der.

Det er mulig å bruke løsningsstrategien i kap. 6 også med Microsoft WSS, men dette er ikke anbefalt siden WSS først og fremst er rettet mot kommunikasjon, koordinering og samarbeid innad i en organisasjon (se kap 7.1.1). Noe funksjonalitet vil følgelig ikke samsvare med det som finnes kun i WSS 3.0.

¹⁵ <http://www.digitroll.no/comweb.asp?id=90>

Endel sider av systemet i kapittel 6 overlapper med WSS-utgaven og studentgruppas løsning.

Eksempler på dette er:

- Forumfunksjonalitet (bortsett fra at løsningen i kap. 6 tilbyr mer funksjonalitet, bl.a. WYSIWYG i teksteditoren for innskriving av innhold i løsningen).
- Posting av artikler/innhold.

Eksempler på funksjonalitet i denne løsningen som ikke finnes i hverken WSS-utgaven eller studentgruppas løsning, er bl.a.:

- Eportfølje
- Søk
- IM/Chat
- Personlig profil
- Wiki (Wiki er mulig å opprette i WSS, men er ikke brukt i WSS-utgave av PLU-veven)

Begrunnelsen for inkludering av disse er beskrevet i kapittel 6. For det første ble søk og chat nevnt som mulige verktøy som skulle opp til diskusjon i senere faser av PLU-veven. Søk skal være lagt til rette for i både WSS-utgaven og i studentgruppas løsning, men i WSS-utgaven må man logge inn for å søke, mens implementeringen ikke er fullført i studentgruppas versjon. Eportfølje vil være et viktig bidrag til eventuelle FoU-kurs og mer e-læringsforetak innenfor veven. Personlig profil vil la brukere uttrykke seg mer på veven, og gi mer informasjon om seg selv enn bare et brukernavn. Dette kan være nyttig for personer som ikke kjenner de andre i virkeligheten. Wiki ble også nevnt som et verktøy som skulle opp til diskusjon i senere utgaver av PLU-veven. Verktøyet kan være nyttig for f.eks. å bygge en nettbasert kunnskapsbase rundt FoU og etter- og videreutdanning av lærere, eller erfaringer og kunnskap rundt bruk av IKT i denne konteksten. Funksjonen vil da kreve et stort bidrag fra flere personer.

Designet i kap. 6 legger også ganske mye vekt på sikkerhetsaspektet ved det nettbaserte verktøyet. Dette blir stadig viktigere i en hverdag der tilrøving av digital informasjon og innbrudd er ventet å øke.

7.3 Ferdig løsning kontra egenutviklet

Ferdige løsninger har de fordelene at de trenger bare å tilpasses etter administrators hensikt, men det kan påbeløpe mye ressurser bare for å få dette til, hvis verktøyet er lite tilpasningsvennlig.

Egenutviklede løsninger har de fordelene at de kan tilpasses i sin helhet fra bunnen av. Ulempene med disse er at de krever god programmerings- og utviklingskompetanse. PLU kan spare flest ressurser om de går for en tilpasningsvennlig ferdig løsning, men dette avhenger mye av om de finner en passende ferdig løsning. På den annen side har de studentgruppas løsning som jo allerede er påbegynt. Det krever derfor ikke så mye ressurser å få videreutviklet denne, selv om en del gjenstår (se kap. 7.2.1).

Ferdige løsninger har ofte en rekke utvidelser tilgjengelig, vanligvis for en liten pris. Det er en fordel at disse har åpen kildekode hvis det er ønskelig for PLU å gjøre tilpasninger på disse utvidelsene. Vanligvis er det ikke behov for å redigere kildekode for systemkjernen, så denne kan ha en lukket kildekode. Et annet alternativ er å anskaffe en ferdig løsning med lukket kildekode, for så å selv utvikle utvidelser som kan integreres eller installeres i systemet. Dette krever at den ferdige løsningen har god dokumentasjon på hvordan dette skal gjøres, og/eller tilbyr en API for dette. Hvis systemet f.eks. har et grensesnitt for å kjøre webtjenester, kan man utvikle sine egne tilleggstjenester som kan kjøres på systemet.

8 Konklusjon og videre arbeid

8.1 Undersøkelse av nettbaserte verktøy

I denne oppgaven ble det gjort undersøkelser av eksisterende nettbaserte verktøy, inkludert studentgruppas prototype. Studentgruppas løsning ble funnet å oppfylle grunnleggende funksjonalitet, med et enkelt og rent design, men med noen mangler som ble avgjørende for å betrakte denne løsningen som uferdig for full drift etter PLUs behov. Dette gjaldt det faktum at administrering av websidenes innhold ikke foregikk i et webgrensesnitt, men i et program som måtte installeres hos klienten. Det gjaldt også tungvint redigering av innhold i portalen, og et noe uferdig GUI. Likevel anbefales løsningen som noe man kan bygge videre på dersom man ikke finner et ferdig (out of the box) system som tilfredsstillende alle kravene for PLU-veven, og dersom man heller ikke ønsker å utvikle et nytt systemet fra bunnen av igjen.

Microsoft Windows SharePoint Services 3.0 er en ferdig løsning som ble evaluert etter PLUs behov. Denne ble funnet å ikke helt tilfredsstillende behovene, og denne oppgaven vil ikke anbefale videre bruk av denne løsningen hvis PLU-veven har tenkt å vokse mot de målene som er satt for den på lang sikt. Løsningen kan likevel kanskje benyttes samfungerende med andre løsninger, f.eks. Microsoft Office SharePoint Server (MOSS), Microsoft SQL Server og lignende. Rammeverket Learning Gateway (<http://www.learninggateway.net/mlgbenefits/default.aspx>), som inkluderer WSS sammen med Microsoft SQL Server og Microsoft Exchange Server, kan være en løsning for samhandlingsfunksjonalitet, men denne er ikke undersøkt i denne besvarelsen. WSS kan godt benyttes internt i PLU.

Joomla! er en ferdig open source løsning som ble evaluert i besvarelsen. Løsningen ble funnet å tilfredsstillende de fleste av PLUs krav til PLU-veven, hvis man tar i bruk innstallering av noen utvidelser og små visuelle tilpasninger i grensesnittet. Joomla anbefales for bruk i PLU-veven på kort sikt og også på lang sikt, selv om skalerbarheten til systemet (i størrelsen hundretusener sidetreff eller mer per dag) er noe usikker. Noe kjennskap til PHP og konfigureringer av systemet for å motvirke små feil er anbefalt.

It's Learning er et e-læringsystem som leveres som en tjeneste til kundene. Systemet har først og fremst e-læringsfunksjonalitet, noe som kommer noe utenfor PLUs behov. Løsningen kan benyttes for FoU-kurs, men systemet anbefales ikke for PLU-veven, spesielt pga. at systemet er "lukket" (man må ha brukernavn og passord for å se innholdet) og manglende muligheter for å tilpasse visuell layout og konstruere nettsider.

8.2 Design og evaluering av nettbaserte verktøy

Det har blitt gitt et forslag til design av en helhetlig nettbasert løsning for PLU-veven. Denne ble laget ut fra teori om brukbarhet, grensesnitt, PLUs kravspesifikasjon, intervjuer og spørreskjemaer og NTNUs designprinsipper. Siden PLU ikke helt vet hvordan PLU-veven skal være, må den ses på som en rettesnor. Designet og den tilhørende utviklingsprosessen kan først og fremst benyttes for egenutviklet løsning, men kan også fungere som en sjekklister for videreutvikling på eksisterende løsninger, i det den lister opp viktige sider ved et system som PLU-veven tenker å ta i bruk.

8.2.1 PLUs valg og veien videre

Denne besvarelsen har besvart forskningsspørsmålene:

- Undersøkelse av nettbaserte ressurser for bruk i etter- og videreutdanning av lærere.
- Design og evaluering av nettbaserte verktøy for bruk til dette formålet.

Besvarelsen har sett på PLU-veven som et Case studie. Forhåpentligvis vil dette ha gitt PLU noen svar omkring veien videre for PLU-veven, samt at etter- og videreutdanning av lærere generelt sett kan dra nytte av undersøkelsene utført i denne besvarelsen.

Veien videre for PLU når det gjelder nettbaserte ressurser i etter- og videreutdanning av lærere bør først være å evaluere hvordan erfaringen med den nåværende utgaven av PLU-veven har vært, og hva som kan gjøres bedre. Veien videre avhenger mye av tilgjengelige ressurser i PLU som organisasjon, og tilgangen til utviklere. Det ble f.eks. observert i løpet av studiet at PLU valgte bort studentgruppas løsning fordi det ble gitt et bedre driftstilbud ved NTNU for en annen løsningsstrategi. Årsaken til dette er mest sannsynlig driftsavdelingens manglende kompetanse i bistand på den første løsningen, og tilsvarende større kompetanse på SharePoint-løsninger. Dette er noe PLU må leve med, men de kan trolig komme rundt dette hinderet med ansettelse av personer med den riktige kompetansen.

Det viktigste er at IKT-verktøyet skal hjelpe til å oppfylle PLUs mål for etter- og videreutdanning av lærere, skape mer læring og kunnskapsdeling, sosial utvikling og til syvende og sist gi en større nytte for samfunnet.

9 Referanseliste

- (1) E. Zachariassen, *Handler IT-tjenester for milliardbeløp*, Teknisk Ukeblad nr. 9, mars 2008.
- (2) Yildirim et al, 2004, *What Makes a Good LMS: An Analytical Approach to Assessment of LMSs*.
- (3) Cornford & Smithson, *Project Research in Information Systems, A Student's Guide*, S.E., Palgrave Macmillan, 2006, Kap. 4.
- (4) Takashima, K., Omori, Y., Yoshimoto, Y., Itoh, Y., Kitamura, Y., and Kishino, F. 2008. Effects of avatar's blinking animation on person impressions. In Proceedings of Graphics interface 2008 (Windsor, Ontario, Canada, May 28 - 30, 2008). ACM International Conference Proceeding Series, vol. 322. Canadian Information Processing Society, Toronto, Ont., Canada, 169-176.

- (5) Walsham, G. 1993. IS strategy and implementation: a case study of a building society. SGOIS Bull. 14, 2 (Dec. 1993), 13-16. DOI= <http://doi.acm.org/10.1145/170846.170852>
- (6) Robertson J., *How to evaluate a content management system*, 2002. [Online]. Tilgjengelig: http://www.steptwo.com.au/papers/kmc_evaluate/
- (7) Gilbane Group, nr 2, mars 2003. *The Classification and Evaluation of Content Management Systems*, The Gilbane Report. [Online]. Tilgjengelig: <http://www.cmsreview.com/Resources/GilbaneReport.html>
- (8) Ajax Tutorial, W3 Schools. [Online]. Tilgjengelig: <http://www.w3schools.com/ajax/default.asp>
- (9) Liberty & Hurwitz, 2006, *Programming ASP.NET*, O'Reilly Media Inc.
- (10) Andronache, A., Brust, M. R., and Rothkugel, S. 2007. Hycast- podcast discovery in mobile networks. In Proceedings of the 3rd ACM Workshop on Wireless Multimedia Networking and Performance Modeling (Chania, Crete Island, Greece, October 22 - 22, 2007). WMuNeP '07. ACM, New York, NY, 27-34. DOI= <http://doi.acm.org/10.1145/1298216.1298224>
- (11) Wang, T., Chang, H., Sie, Y., Chan, K., Tzou, M., and Shih, T. K. 2005. Reading SCORM compliant multimedia courses using heterogeneous pervasive devices. In Proceedings of the 13th Annual ACM international Conference on Multimedia (Hilton, Singapore, November 06 - 11, 2005). MULTIMEDIA '05. ACM, New York, NY, 806-807. DOI= <http://doi.acm.org/10.1145/1101149.1101325>
- (12) Gilbane Group, *What is Content Management?*, The Gilbane Report vol. 8, nr. 8, oktober 2000. [Online]. Tilgjengelig: http://gilbane.com/gilbane_report.pl/6/What_is_Content_Management.html
- (13) Sheard et al, *Web-based Discussion Forums: The Staff Perspective*, 2003.
- (14) Graf H., 2007, *Building Websites with Joomla! 1.5 Beta 1*, Packt Publishing.
- (15) Kyrnin J., *Overcoming Resistance to Content Management*. [Online]. Tilgjengelig: <http://webdesign.about.com/library/weekly/aa022502a.htm>

- (16) Studentgruppa, 2007, *PLU-veven, Kundeprosjekt nr. 14*. IT2901.
- (17) NTNUs retningslinjer for webdesign. [Online]. Tilgjengelig:
<http://www.ntnu.no/selvhjelpspakken/web/malverk>
- (18) Nielsen, 2000, *Designing Web Usability*, New Riders Publishing.
- (19) Blogg, 2007, *Hva vet vi om bruk av LMS'er i norsk skole?* [Online]. Tilgjengelig:
<http://tomerik.wordpress.com/2007/12/03/hva-vet-vi-om-bruk-av-lms%C2%B4er-i-norsk-skole/>
- (20) Brodersen, C., Christensen, B. G., Grønbaek, K., Dindler, C., and Sundararajah, B. 2005. eBag: a ubiquitous Web infrastructure for nomadic learning. In Proceedings of the 14th international Conference on World Wide Web (Chiba, Japan, May 10 - 14, 2005). WWW '05. ACM, New York, NY, 298-306. DOI= <http://doi.acm.org/10.1145/1060745.1060791>
- (21) W. D. Graziadei, 2004
- (22) Larman, 2002, *Applying UML and Patterns*, S.E., Prentice Hall PTR.
- (23) Nardi et al, 2004, *"I'm Blogging This" A Closer Look at Why People Blog*.
- (24) Hu, M., Sun, A., and Lim, E. 2007. Comments-oriented blog summarization by sentence extraction. In Proceedings of the Sixteenth ACM Conference on Conference on information and Knowledge Management (Lisbon, Portugal, November 06 - 10, 2007). CIKM '07. ACM, New York, NY, 901-904. DOI= <http://doi.acm.org/10.1145/1321440.1321571>
- (25) Van Doren, juni 2001, *Simplicity: The Cobbles of the Designer's Path*, Digital Web Magazine. [Online]. Tilgjengelig: http://www.digital-web.com/articles/simplicity_the_cobbles_of_the_designers_path/
- (26) Grinter, R. E. and Palen, L. 2002. Instant messaging in teen life. In Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work (New Orleans, Louisiana, USA, November 16 - 20, 2002). CSCW '02. ACM, New York, NY, 21-30. DOI= <http://doi.acm.org/10.1145/587078.587082>
- (27) Isaacs et al, 2002, *The Character, Functions, and Styles of Instant Messaging in the Workspace*.

- (28) Hasan & Pfaff, 2006, *The Wiki: an environment to revolutionise employees' interaction with corporate knowledge*.
- (29) Di Iorio & Zacchiroli, 2006, *Constrained Wiki: an Oxymoron?*
- (30) Shneiderman & Plaisant, *Designing the User Interface*, 2005, Pearson Education Inc., Addison Wesley.
- (31) Neuman, Delia, *Learning disabled students' interactions with commercial courseware: A naturalistic study*, *Educational Technology Research and Development*, 39,1 (1991), 31-49.
- (32) Glotzbach et al, 2007, *RSS as a Course Information Delivery Method*.
- (33) SVT-Drift, NTNU.
- (34) Blekas et al, 2006, *Use of RSS feeds for Content Adaptation in Mobile Web Browsing*.
- (35) G. McGraw, 2006, *Software Security – Building Security In*, Pearson Education Inc., Addison Wesley.
- (36) Andrews & Whittaker, 2006, *How to Break Web Software*, Pearson Education Inc., Addison Wesley.
- (37) Anderson et al, 2008, *Wheels Around the World: Windows Live Mobile Interface Design*.
- (38) Halpert, B. J. 2005. Authentication interface evaluation and design for mobile devices. In Proceedings of the 2nd Annual Conference on information Security Curriculum Development (Kennesaw, Georgia, September 23 - 24, 2005). InfoSecCD '05. ACM, New York, NY, 112-117. DOI= <http://doi.acm.org/10.1145/1107622.1107649>
- (39) Seong, D. S. 2006. Usability guidelines for designing mobile learning portals. In Proceedings of the 3rd international Conference on Mobile Technology, Applications & Systems (Bangkok, Thailand, October 25 - 27, 2006). Mobility '06, vol. 270. ACM, New York, NY, 25. DOI= <http://doi.acm.org/10.1145/1292331.1292359>
- (40) Kong et al, 2005, *Supporting Web User Interface Prototyping through Information Modeling and System Architecting*.
- (41) Tenenberg & Wang, 2005, *Using Course Portfolios to create a Disciplinary Commons*

across Institutions.

- (42) Razavi & Iverson, 2006, *A Grounded Theory of Information Sharing Behavior in a Personal Learning Space.*
- (43) H. Solomon, 2007, *Master Pages vs. Themes: Which Do You Choose?* [Online].
Tilgjengelig: <http://www.heathersolomon.com/blog/archive/2007/05/13/Master-Pages-vs.-Themes-Which-Do-You-Choose.aspx>
- (44) S. Walther, 2008, *Building ASP.NET 2.0 Web Sites Using Web Standards*, ASP.NET Technical Articles, .NET Framework Developer Center. [Online]. Tilgjengelig: <http://msdn.microsoft.com/en-us/library/aa479043.aspx>
- (45) J. Cato, 2001, *User-Centered Web Design*, Pearson Education Limited, Addison Wesley.
- (46) S. Mehringer, 2005, *Sit! Fetch! Roll Over! Tricks You Can Perform with a Content Management System*, [Online]. Tilgjengelig: <http://www.siguccs.org/Conference/Spring2005/presentations/mehringer.pdf>
- (47) M. Bailey, 2006, *Building a Website Using a CMS in 90 Minutes*, Management Books 2000 Ltd.
- (48) B. Lublinsky, 2008, *New Options for .NET-Java Web Services Interoperability*. [Online].
Tilgjengelig: <http://www.infoq.com/news/2008/03/wmq-activemq-interop;jsessionid=7511F76BAF76CBC17304A2D9AF01021E>

10 Appendix

10.1 Spørreskjema

Skjemanavn:

Evaluering av nettbaserte tjenester for bruk i etter- og videreutdanning av lærere ved NTNU

Spørreskjematekst:

Det er et behov for å kartlegge behovet for eller bruken av nettbaserte tjenester (tjenester på nett) i forbindelse med etter- og videreutdanning av lærere ved NTNU. Resultatet av spørreundersøkelsen

skal hjelpe å forstå viktigheten av et pågående prosjekt (webportal) for bruk til dette formålet, samt hjelpe å styre utviklingen av prosjektet eller andre nettbaserte tjenester til dette formålet.

Først og fremst er spørreskjemaet aktuelt for praksisveilederne ved partnerskolene, men alle som er et ledd i etter- og videreutdanning av lærere ved NTNU oppfordres til å svare på undersøkelsen.

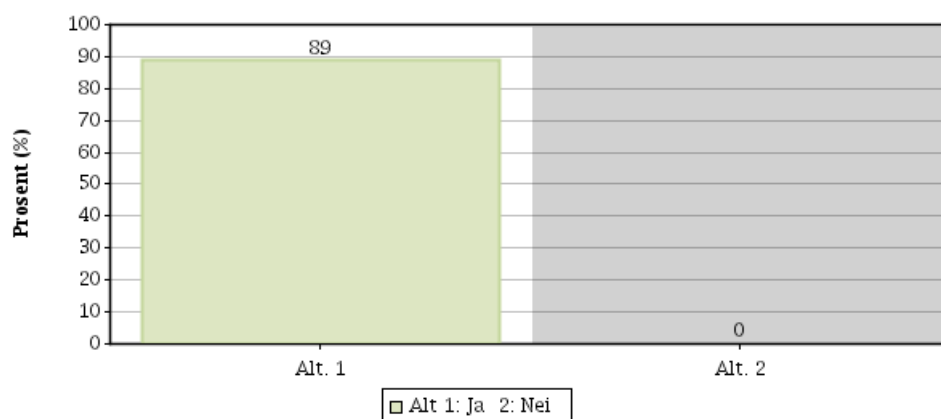
Spørsmål nr: 1

Mener du det finnes nettbaserte ressurser for bruk i etter- og videreutdanning av lærere ved NTNU? Antall svar: 8

Alternativ nr: 1 Ja 8

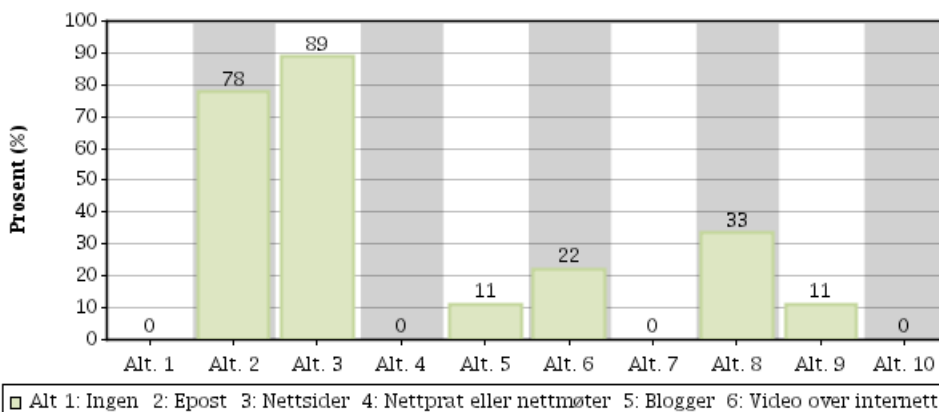
Alternativ nr: 2 Nei 0

Spørsmål: Mener du det finnes nettbaserte ressurser for bruk i etter- og videreutdanning av lærere ved NTNU?

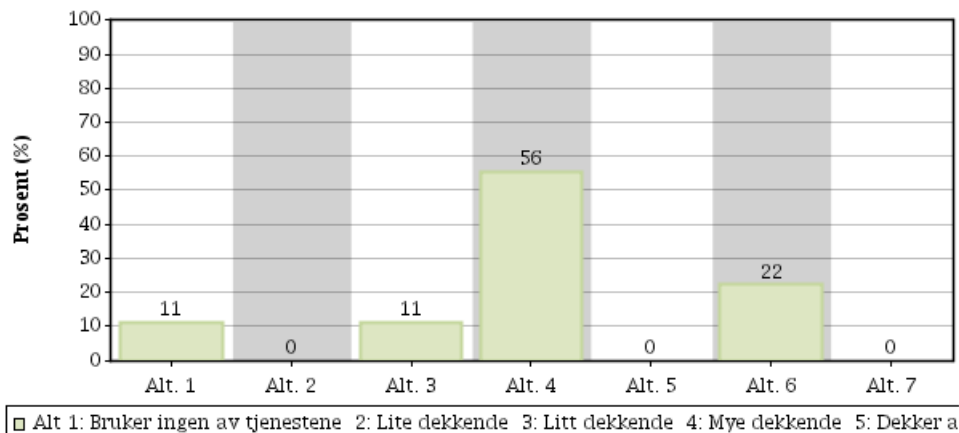


Spørsmål nr: 2**Hvilke nettbaserte tjenester tar du selv i bruk som del i etter- og videreutdanning av lærere?** Antall svar: 22

Alternativ nr: 1 Ingen	0
Alternativ nr: 2 Epost	7
Alternativ nr: 3 Nettsider	8
Alternativ nr: 4 Nettprat eller nettmøter	0
Alternativ nr: 5 Blogger	1
Alternativ nr: 6 Video over internett	2
Alternativ nr: 7 Lydopptak over internett	0
Alternativ nr: 8 Diskusjonsforum på nett	3
Alternativ nr: 9 Andre - Moodle	1
Alternativ nr: 10 Ikke sikker	0

Spørsmål: Hvilke nettbaserte tjenester tar du selv i bruk som del i etter- og videreutdanning av lærere?**Spørsmål nr: 3****Er de tjenestene du bruker i dag, dekkende for ditt behov i ditt arbeid?** Antall svar: 9

Alternativ nr: 1 Bruker ingen av tjenestene	1
Alternativ nr: 2 Lite dekkende	0
Alternativ nr: 3 Litt dekkende	1
Alternativ nr: 4 Mye dekkende	5
Alternativ nr: 5 Dekker alle behov	0
Alternativ nr: 6 Dekker mer enn hva jeg har behov for	2
Alternativ nr: 7 Ikke sikker	0

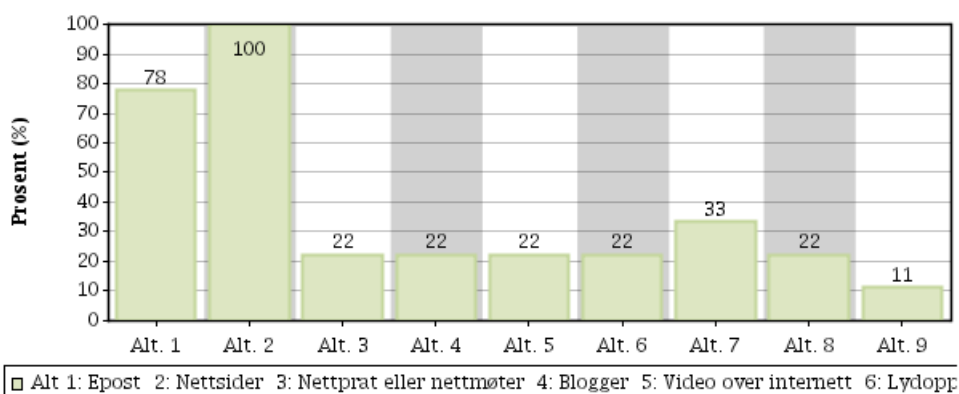
Spørsmål: Er de tjenestene du bruker i dag, dekkende for ditt behov i ditt arbeid?

Spørsmål nr: 4

Hvilke nettbaserte IKT-tjenester synes du at du trenger i ditt arbeid? Sett kryss ved alle tjenester du synes du trenger, uansett om du bruker dem i dag. La alle stå blanke om du synes du ikke trenger noen. Antall svar: 30

Alternativ nr: 1 Epost	7
Alternativ nr: 2 Nettsider	9
Alternativ nr: 3 Nettprat eller nettmøter	2
Alternativ nr: 4 Blogger	2
Alternativ nr: 5 Video over internett	2
Alternativ nr: 6 Lyddoptak over internett	2
Alternativ nr: 7 Diskusjonsforum på nett	3
Alternativ nr: 8 Andre	2
- Delbar Whiteboard for håndskrift	
- wiki	
Alternativ nr: 9 Ikke sikker	1

Spørsmål: Hvilke nettbaserte IKT-tjenester synes du at du trenger i ditt arbeid? Sett kryss ved alle tjenester du synes du trenger, uansett om du bruker dem i dag. La alle stå blanke om du synes du ikke trenger noen.

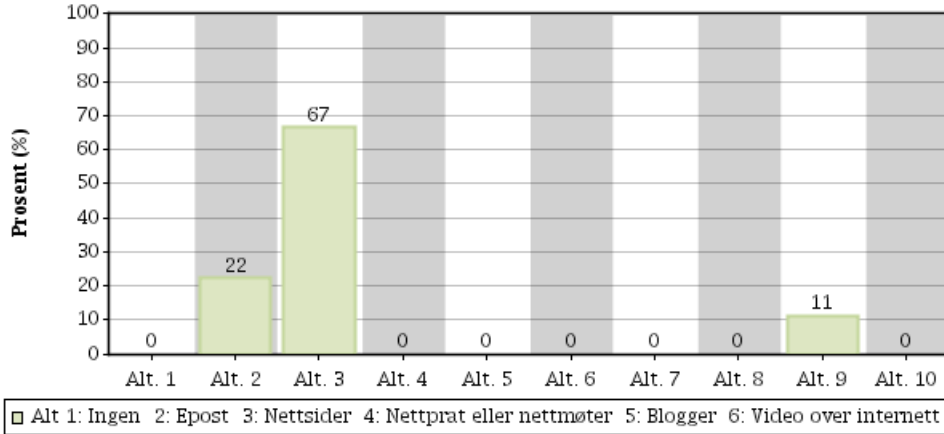


Spørsmål nr: 5

Merk av den nettbaserte tjenesten du synes er viktigst i ditt arbeid. Antall svar: 9

Alternativ nr: 1 Ingen	0
Alternativ nr: 2 Epost	2
Alternativ nr: 3 Nettsider	6
Alternativ nr: 4 Nettpprat eller nettmøter	0
Alternativ nr: 5 Blogger	0
Alternativ nr: 6 Video over internett	0
Alternativ nr: 7 Lydopptak over internett	0
Alternativ nr: 8 Diskusjonsforum på nett	0
Alternativ nr: 9 Andre	1
- Moodle	
Alternativ nr: 10 Ikke sikker	0

Spørsmål: Merk av den nettbaserte tjenesten du synes er viktigst i ditt arbeid.

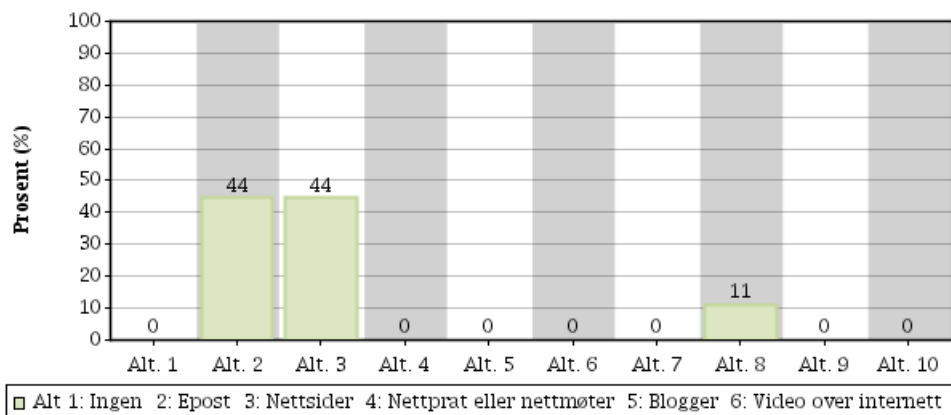


Spørsmål nr: 6

Merk av den nettbaserte tjenesten du synes kommer på en 2. plass i hvor viktig den er i ditt arbeid. Antall svar: 9

Alternativ nr: 1 Ingen	0
Alternativ nr: 2 Epost	4
Alternativ nr: 3 Nettsider	4
Alternativ nr: 4 Nettprat eller nettmøter	0
Alternativ nr: 5 Blogger	0
Alternativ nr: 6 Video over internett	0
Alternativ nr: 7 Lydopptak over internett	0
Alternativ nr: 8 Diskusjonsforum på nett	1
Alternativ nr: 9 Andre	0
Alternativ nr: 10 Ikke sikker	0

Spørsmål: Merk av den nettbaserte tjenesten du synes kommer på en 2. plass i hvor viktig den er i ditt arbeid.

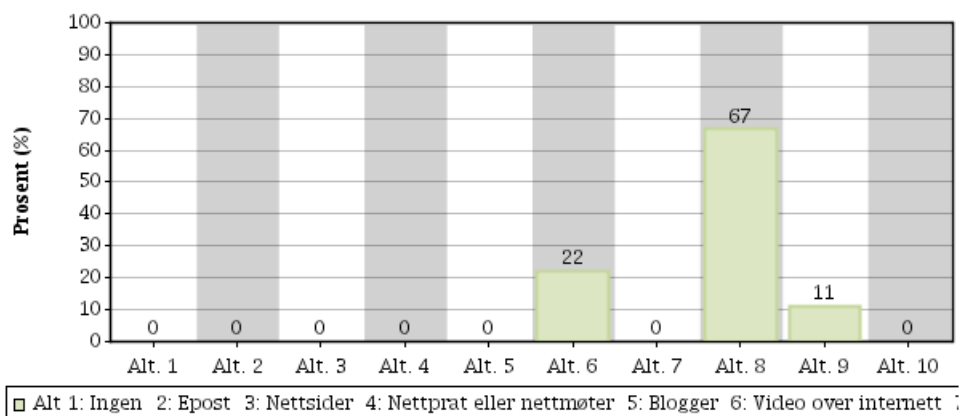


Spørsmål nr: 7

Merk av den nettbaserte tjenesten du synes kommer på en 3. plass i hvor viktig den er i ditt arbeid. Antall svar: 9

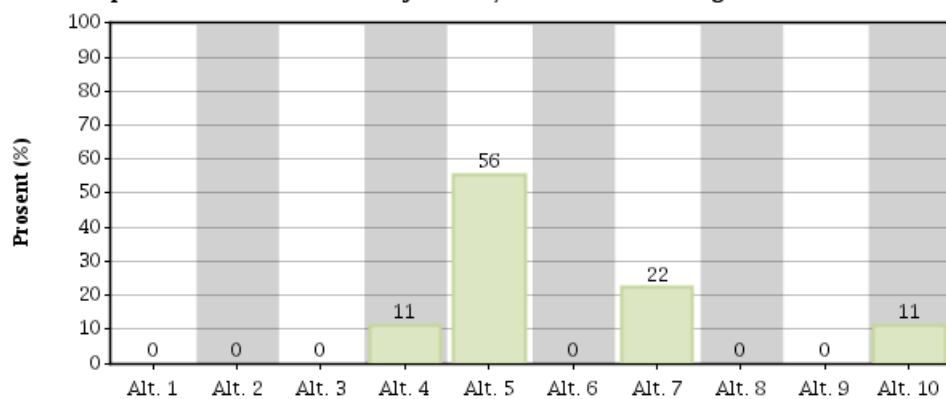
Alternativ nr: 1 Ingen	0
Alternativ nr: 2 Epost	0
Alternativ nr: 3 Nettsider	0
Alternativ nr: 4 Nettprat eller nettmøter	0
Alternativ nr: 5 Blogger	0
Alternativ nr: 6 Video over internett	2
Alternativ nr: 7 Lydopptak over internett	0
Alternativ nr: 8 Diskusjonsforum på nett	6
Alternativ nr: 9 Andre	1
- bruk av it*s learning går vel under "nettsider"	
Alternativ nr: 10 Ikke sikker	0

Spørsmål: Merk av den nettbaserte tjenesten du synes kommer på en 3. plass i hvor viktig den er i ditt arbeid.



Spørsmål nr: 8**Hvilken nettbasert tjeneste synes du er minst viktig i ditt arbeid?** Antall svar: 9

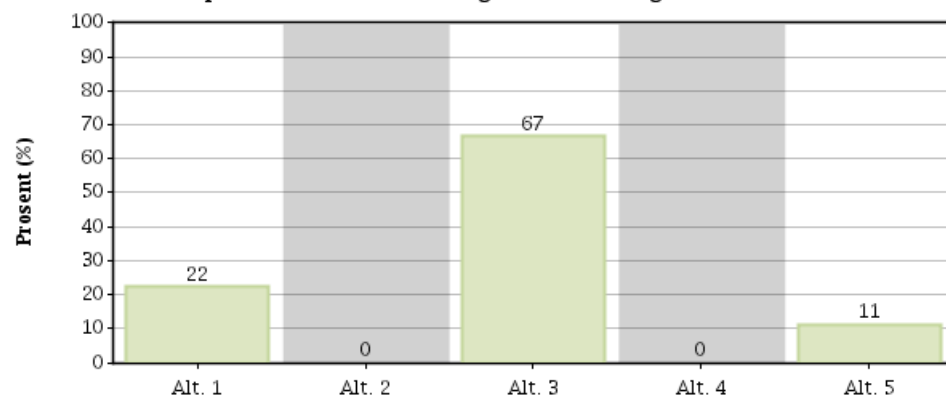
Alternativ nr: 1 Ingen	0
Alternativ nr: 2 Epost	0
Alternativ nr: 3 Nettsider	0
Alternativ nr: 4 Nettprat eller nettmøter	1
Alternativ nr: 5 Blogger	5
Alternativ nr: 6 Video over internett	0
Alternativ nr: 7 Lyddoptak over internett	2
Alternativ nr: 8 Diskusjonsforum på nett	0
Alternativ nr: 9 Andre	0
Alternativ nr: 10 Ikke sikker	1

Spørsmål: Hvilken nettbasert tjeneste synes du er minst viktig i ditt arbeid?

□ Alt. 1: Ingen 2: Epost 3: Nettsider 4: Nettprat eller nettmøter 5: Blogger 6: Video over internett 7: Lyddoptak over internett 8: Diskusjonsforum på nett 9: Andre 10: Ikke sikker

Spørsmål nr: 9**Din rolle i etter- og videreutdanning av lærere.** Antall svar: 9

Alternativ nr: 1 Foreleser i regi av PLU	2
Alternativ nr: 2 Foreleser	0
Alternativ nr: 3 Koordinator/veileder for lærerstudenter	6
Alternativ nr: 4 Lærerstudent	0
Alternativ nr: 5 Annet	1
- Daglig leder av DELTA (www.ntnu.no/delta)	

Spørsmål: Din rolle i etter- og videreutdanning av lærere.

□ Alt. 1: Foreleser i regi av PLU 2: Foreleser 3: Koordinator/veileder for lærerstudenter 4: Lærerstudent 5: Annet - Daglig leder av DELTA (www.ntnu.no/delta)

10.2 Webutviklingsmetode eksempel 1

Jfr. kap. 4.4 Utviklingsteori så presenteres her de initielle stegene i (Kong et al, 2005)s webutviklingsmetode, anvendt på plu-veven.

Analyse-steget

Oppgave: *"analystene og klienten utforsker forretningsprosessen for webapplikasjonen som blir bygd. Potensielle applikasjonsbrukere og deres mål blir identifisert. Oppgaver for å oppnå målene blir oppdaget."*

Plu-veven på kort sikt

Potensielle brukere:

Praksisveilederne ved partnerskolene

PLU-ansatte

Lærerstudenter

Forelesere/lærere

Deres mål:

Praksisveilederne: Heve sin veiledningskompetanse i FoU-kunnskap overfor lærerstudentene. Få etter- og videreutdanning.

PLU-ansatte: Lære praksisveilederne opp i FoU og etter- og videreutdanning

Lærerstudenter: Få kunnskaper om/lære FoU-arbeid.

Forelesere/lærere: Dele stoff rundt FoU og etter- og videreutdanning.

Oppgaver for å oppnå målene:

Praksisveilederne ved partnerskolene:

Gjennomføre kurspakker/moduler på nettstedet

Delta i forum

Surfe etter informasjon på nettstedet

PLU-ansatte, forelesere og lærere:

Design av kurspakker og fagmessig innhold

Legge ut fagmessig innhold på nettstedet

Være aktive i forum

Lese/evaluere oppgavebesvarelser

Lærerstudenter:

Gjennomføre kurspakker/moduler for å lære FoU.

Delta i forum

Surfe etter informasjon på nettstedet

Vi kan samle oppgavene i 6 scenarier:

Gjennomføre kurspakker/moduler på nettstedet

Legge ut fagmessig innhold på nettstedet

Delta i forum på nettstedet

Surfe etter informasjon på nettstedet

Design av kurspakker

Lese/evaluere oppgavebesvarelser

Flyten i oppgavene

Gjennomføre kurspakker/moduler på nettstedet

Registrere seg

Logge inn

Laste ned og besvare oppgaver

Levere inn besvarelser på nett

Legge ut fagmessig innhold på nettstedet

Registrere seg

Logge inn

Velge tema eller riktig kursmodul stoffet skal legges inn under

Laste inn fil (dokument, presentasjon, lydfil), videolink eller skrive artikkel

Delta i forum på nettstedet

Logge inn (valgfritt)

Navigere til (riktig) forum

Poste i forumet

Surfe etter informasjon på nettstedet

Logge inn (valgfritt)

Velge tema

Velge ressurser (filer eller tekst) under tema

Velge tematisert kurspakke og last ned

Velge utematiserte ressurser. Laste ned eller lese/se direkte

Velge kurspakke og last ned

Design av kurspakker

Logge inn

Velge tema

Navngi og beskriv kurs

Laste opp kursfiler

Skrive inn kurstekst og oppgaver på nett

Navngi og beskriv kurs

Laste opp kursfiler

Skrive inn kurstekst og oppgaver på nett

Lese/evaluere oppgavebesvarelser

Logge inn

Lese oppgavebesvarelser/resultater av tester

Laste ned oppgavebesvarelser

Lese/laste ned tilbakemeldinger

10.3 **James Robertsons evalueringsskjema for CMS**

- **Forretningsmål og strategier**

Man må vite hvorfor man trenger et CMS/nettbasert verktøy. CMS skal oppfylle disse målene. Målene må reflektere strategier på lang sikt og direksjoner for organisasjonen.

- **Identifisere krav**

Involvér alle parter i innsamlingen av kravspesifikasjon (IT-ansvarlige, brukere, organisasjonsenheter).

Her kommer kravspesifikasjonen som PLU utarbeidet med studentgruppa inn. Ble alle parter involvert i innsamlingen? Ble alle krav identifisert?

- **Strukturere kravene**

Grupper kravene i kategorier, f.eks som følger:

- **Innholdsskaping**

Funksjonalitet påkrevd av innholdsskaperne i systemet

- **Integrert forfattermiljø**

Systemet må støtte et sømløst og kraftig miljø for innholdsskaping.

- **Separering av innhold og presentasjon**

For å gjøre det mulig å publisere til forskjellige formater. Må være stilbasert.

- **Flerbruker-skriving**

Må kunne håndtere at flere brukere samtidig skriver innhold.

Låsemekanismer.

- **Gjenbruk**

En enkelt side (eller paragraf) må kunne brukes i forskjellige kontekster, og leveres til forskjellige brukergrupper.

- **Metadata**

Metadata er viktig for store innholdslagre. Nøkkelord og topic maps.

- **Kraftfull lenking**

Krysslenker mellom sider må være stabile mot omstrukturering.

- **Ikke-teknisk skriving**

Forfattere av tekster må ikke behøve å beherske HTML eller lignende teknisk kunnskap.

- **Lettbrukelighet og effektivitet**

Lett å lage og behandle innhold.

- **Innholdsbehandling**

Nøkkelkrav kan være:

- **Versjonskontroll og arkivering**

Streng versjonskontroll er nødvendig for opprettholdelse av lovmessige ansvar, backup og katastrofehandtering.

- **Arbeidsflyt**

Kraftfull arbeidsflyt-modell for desentralisert innholdsskaping.

Motstandsdyktig mot organisasjonsforandringer.

- **Sikkerhet**

For å beskytte innholdets integritet.

- **Integrasjon med eksterne systemer**

CMS kan være bare ett av flere systemer for å presentere informasjon på websidene. Må lett kunne integreres med eksisterende systemer, gjennom åpne eller industrielle standarder.

- **Rapportering**

Systemet må rapportere situasjoner som oppstår.

- **Publisering**

Denne delen av CMS-syklusen sørger for at innholdet som er lagret blir generert som ferdige sider.

Nøkkelkrav her kan være:

- **Stilark**

Visning kontrolleres ved hjelp av stilark. Gir fleksibilitet og ekspanderbarhet.

- **Sidemaler**

Sidelayout er spesifisert via sidemaler.

- **Utvidbarhet**

Må støtte en prosess for kontinuerlig forbedring av brukergrensesnitt-design

- **Støtte for flere forskjellige formater**

Må kunne publisere til f.eks. HTML (Web), trykksaker, PDF, håndholdt (WAP), m.m. Bør kunne støtte mulighet for framtidige, kommende formater.

- **Personalisering**

Forskjellig informasjon blir presentert ut fra forskjellige brukerprofiler eller metadata i kildeinnholdet.

- **Bruksstatistikk**

CMS'et må tillate omfattende innsamling av bruksstatistikk, inkludert:

- Mest populære sider
- Daglig bruk
- Søkemotorbruk

Dette muliggjør sporing av de vellykkede og mindre vellykkede sidene av portalen. Brukbarhetsproblemer kan identifiseres.

- **Presentasjon**

De publiserte sidene må oppfylle en viss standard. Nøkkelkrav her kan være:

- **Brukbarhet**

Lettbrukelighet, læringskurve og effektivitet er nøkkelord. Kan testes ved prototyper.

- **Aksesserbarhet**

Det presenterte materialet må oppfylle standarder som W3Cs Web Accessibility Initiative (WAI).

- **Støtte på tvers av nettlesere**

Sidene må kunne vises i alle store nettlesere eller i de nettleserne som brukergruppen bruker. Eksempler kan være Internet Explorer, Firefox, Safari og Opera.

- **Begrenset klient-side funksjonalitet**

Det kan være hensiktsmessig å begrense hvilke klient-side teknologier (Java, JavaScript, Flash etc.) som trengs for å se innhold på webportalen.

- **Hastighet**

Sidestørrelser må være minimale slik at brukernes lastetider er akseptable. Hva er typiske oppkoblingsmetoder for brukerne? (T1, LAN, kabel, ISDN, etc.)

- **Gyldig HTML**

Alle sider må oppfylle nåværende HTML-spesifikasjon. Dette gir kompatibilitet på tvers av nettlesere.

- **Effektiv navigasjon**

Brukere må gis konsistente, omfattende og brukbare navigasjonshjelpemidler.

- **Metadata**

Alle sider må gi tilstrekkelig metadata for å tillate effektiv indeksering og søking. F.eks. Tilfredsstill standarden Dublin Core.

■ **Kontrakter og forretning**

Nøkkelord kan være:

● **Trening**

Leverandøren må liste opp treningsmaterialet som eksisterer for CMS'et, og treningstjenestene disse kan gi.

● **Dokumentasjon**

CMS'et må suppleres med adekvat dokumentasjon for brukere, administratorer og utviklere.

● **Garanti**

Garantiperioden som gis straks programvaren blir tatt i bruk.

● **Vedlikeholdsavtaler**

Leverandøren må skissere sine foretrukne støtteforetak, inkludert Service Level Agreements og oppgraderingsprosesser.

● **Påkrevde ressurser**

Maskinvare-, programvare- og operativsystemressursene påkrevd av CMS'et.

● **Ferdigheter påkrevd**

Hvilke ferdigheter og kunnskaper er påkrevd innen **din** organisasjon for å tilpasse og vedlikeholde CMS'et?

● **Kostnad**

Faste kostnader og kostnader per bruker. Det siste er viktigst for større organisasjoner.

● **Skalerbarhet**

Lastnivåene CMS'et støtter, pluss tilleggsressurser (maskin- og programvare) påkrevd for økt bruk.

● **IT-begrensninger**

Spesifiser eksisterende maskin- eller programvare som systemet må grense mot, eller kjøre på, f.eks. operativsystemer, databaser eller webservere.

● **Referansesider**

Leverandøren må gi et antall eksempler på nettsteder med vellykket implementering av systemet. Disse må være sammenlignbare med din egen organisasjon.

