Forord

Da vi våren 2004 først bestemte oss for å gå sammen og fordype oss i e-læring, hadde vi mange tanker om dette emnet. Vi hadde begge prøvd noen få e-læringssystemer før, men hadde alt i alt ingen god erfaring med dette. Vi følte at disse systemene var vanskelige i bruk og i det hele tatt ikke veldig godt egnet til situasjonen vi var i. Kanskje dette også er grunnen til at vi har brukt dem så lite?

Vi synes derfor det hadde vært artig å prøve å lage noe som var bedre. Noe som var laget på våre egne premisser. Dette var starten på hele den lange prosessen med å lage vårt eget e-læringssystem. Læringsrom i nett er et resultat av masse slit og mye god sjokolade, og nå sitter vi igjen med et program som forhåpentligvis kan billedliggjøre e-læring og dessuten motivere andre til å fokusere mer på emnet.

Vi ønsker å takke alle som har hjulpet oss med oppgaven. Spesielt vil vi takke veileder Arvid Staupe. Vi vil også takke venner og familie som har vært med å teste programmet. Dere har gitt oss mange nyttige tilbakemeldinger.

Arnstein Berg og Håvar Wehus

Trondheim, juni 2005

Sammendrag

Denne oppgaven tar for seg temaet e-læring, og spesielt samarbeidslæring over internett. Gjennom å utvikle store deler av et komplett e-læringssystem, har vi sett på teknologbruk og funksjoner i praksis. Den aller største delen av prosjektet har vært å implementere dette systemet, men i den forbindelse har vi også brukt mye tid på planlegging og refleksjon. Dette har gitt oss mange erfaringer med hvordan e-læring kan og bør legges opp.

Vi har spesielt fokusert på å ta i bruk teknologier som i liten grad er utnyttet i e-læring i dag. Samarbeidsfunksjoner har stått mye i fokus, og vi har sett på måter å gjøre både elev-lærer- og elev-elev-interaksjonen over internett bedre. Særlig gjelder dette bruk av synkron kommunikasjon. Vi har også sett en del på personlige verktøy og systemet vårt som en helhetlig IT-løsning for skoler. Det har samtidig vært et mål for oppgaven å øke bredden i e-læring gjennom å forsøke å tilpasse systemet til flere forskjellige undervisningstyper og aldersgrupper.

Systemet vårt ble som en del av oppgaven vist frem til et lite utvalg personer, for å få noen tilbakemeldinger på e-læringssystemet vårt. Formålet var å gjennomføre et case study, der vi ser hvordan de løsningene vi har diskutert oss frem til, ble mottatt av andre. Studien er utført i veldig liten skala, og kan i liten grad brukes til å generalisere holdninger til e-læring. Den kan derimot være et utgangspunkt for videre forskning, gjennom å se på de momentene vi prøvde ut og hvilke reaksjoner vi mottok.

Resultatet fra testpersonene var overordnet positivt, og vi fikk spesielt god tilbakemelding på at synkron kommunikasjon var ønskelig. Vi har ikke hatt anledning til å legge inn særlig avanserte synkrone funksjoner, men det lille vi hadde med vakte stor interesse. Synkrone kommunikasjonsmetoder ser ut til å være avgjørende for hvor lett det er å samarbeide med andre i systemet. Her er det viktig å føle at det er noen i den andre enden, og den følelsen gir ikke asynkrone metoder i så stor grad.

Studien viste videre at vi ikke hadde klart å tilpasse systemet vårt så godt til alle aldersgrupper som vi hadde håpet. Vi ser imidlertid at ved å bruke mer tid på å utvikle det, kan man forholdsvis enkelt forbedre brukervennligheten. Dersom man bare er villig til å ta i bruk det nyeste av teknologi i dag, er det få begrensninger i forhold til hva slags informasjon man kan overføre.

Innhold

Oppgavebeskrivelse

1 Innledning 6
1.1 Hvorfor er e-læring viktig? 6
1.2 Tilnærmingsmetode 6
1.3 Mål for programmet 7
1.4 Gjennomføringen av prosjektet 7
2 Problembeskrivelse 10
2.1 Bruksområder 10
2.2 Fagområder 12
2.3 Målgrupper 14

Teknisk del

3 Teknisk innledning 20
3.1 Drøfting av generelle begreper 20
3.2 Vurdering av brukergrensesnitt 23
3.3 Vurdering av utviklingsteknologi 24
3.4 Valg av arkitektur 28
4 Krav 30
4.1 Use caser 30
4.2 Aktører 54
4.3 Supplementære krav 54
5 Design og arkitektur 60
5.1 Arkitekturmessige mål og begrensninger 60
5.2 Sentrale scenarier 61
5.3 Programstruktur 69
5.4 Prosesser 85
5.5 Database 86
6 Brukermanual 90
6.1 Installasjon, oppstart og innstillinger 90
6.2 Skrivebord 94
6.3 Skoleweb 105
6.4 Samarbeidsrom 110
6.5 Administrasjonsverktøy 117

Diskusjon

7 Status og videre utvikling 127
7.1 Komponenter i programmet 127
7.2 Organisering av læremateriell 130
7.3 Mulighet for å jobbe frakoblet 130
7.4 Brukere og roller 130
7.5 Feilbehandling 130
7.6 Sikker kommunikasjon 130
7.7 Alternative klienter 131
7.8 Skalerbarhet 132
8 Evaluering 133
8.1 Evalueringsmetoder 133
8.2 Tilbakemeldinger 133
8.3 Vurdering av systemet 134
8.4 Vurdering av bruksområder 135
8.5 Vurdering av målgrupper 136
8.6 E-læring videre fremover 138
9 Konklusjon 139

Appendiks

Appendiks A: Demonstrasjon av Læringsrom i nett 141
Referanser 142
Kilder 142

Oppgave-beskrivelse

1 Innledning

I denne oppgaven vil vi se på hvilke muligheter som finnes innen e-læring, og i hvilke sammenhenger dette er aktuelt å bruke. Med dette ønsker vi å se på hvordan forholdene ligger til rette for å ta i bruk et e-læringssystem på forskjellige nivåer i skoleverket og andre utdanningsinstitusjoner, både i dag og i fremtiden.

1.1 Hvorfor er e-læring viktig?

Dagens samfunn blir stadig mer knyttet opp mot internett. Også i skolen har internett fått en viss appell, men utviklingen her ligger fortsatt en del hakk etter utviklingen man ser i mange andre deler av samfunnet. Samtidig øker etterspørselen av generell dataerfaring i samfunnet, og bruk av datamaskiner blir en stadig mer elementær del av kunnskapen. Det er derfor viktig å venne seg til å bruke datamaskiner relativt tidlig i oppveksten.

Videre gjør internett kommunikasjon over lange avstander mye enklere, og det kan i mange tilfeller nesten helt eliminere forskjeller i fysisk plassering. Dette er absolutt noe man kan ta med seg inn i skolen, der utveksling av informasjon og kommunikasjon generelt, står svært sentralt. Fjernundervisning har blitt en vanlig undervisningsform, og ved å kombinere dette med de mange mulighetene som ligger i internett, kan det åpnes for å legge opp mange nye fagområder som fjernundervisning. Fremtidens skoler kan bli mindre avhengig av lokaler for undervisning – man trenger kun en datamaskin med internettilkobling.

Innføring av et e-læringssystem i skolene vil også kunne gjøre en del vanlige arbeidsoppgaver ved skolen enklere. For eksempel kan man ved hjelp av et slikt system enklere distribuere informasjon, få tilgang til pensumlitteratur, gjøre og rette oppgaver, drive elevoppfølging, og utføre andre administrative oppgaver. Med dette kan e-læring virke positivt inn på mange områder innen skole og opplæring.

1.2 Tilnærmingsmetode

Det er vanskelig å drøfte seg frem til hvorvidt e-læring kan innføres i forskjellige sammenhenger, uten at man vet hva e-læring faktisk innebærer. Med andre ord er vi nødt til å finne ut hva som er mulig rent teknisk, før vi kan se på hva vi kan bruke det til. Vi velger derfor å gå grundig til verks, og se på hva vi konkret kan få til i et e-læringssystem. Planen går ut på å utvikle store deler av et e-læringssystem som vi har valgt å kalle Læringsrom i nett.

Vi er selvsagt klar over at det er flere e-læringssystemer på markedet allerede, men disse ligger etter vår mening forholdsvis langt tilbake for de teknologiske mulighetene som finnes i dag, særlig når det gjelder direkte samarbeid. Vi vil derfor prøve å se hva vi kan få til ved hjelp av utstrakt bruk av nye teknologier.

Vi vil helt klart ikke få tid til å utvikle et ferdig e-læringssystem, men vi håper å kunne implementere sentrale funksjoner på en slik måte at de lar seg demonstrere for brukere. Oppgaven blir således å se på som en slags case study [1], der vi prøver å utvide kunnskapen vi allerede har. Dette vil trolig ikke gi noen helt klare retningslinjer for hvordan et e-læringssystem må være, men vi håper det kan gi en god pekepinn på hva som er mulig, og hva man kan vente seg i fremtidige e-læringssystemer.

1.3 Mål for programmet

Vi vil med Læringsrom i nett lage et e-læringssystem, der man i utgangspunktet skal kunne gjøre alt man tradisjonelt har kunnet gjort på skolen. Man skal kunne jobbe med alle typer skolearbeid, både selvstudium, gruppearbeid og forelesning, med muligheter for å se og høre de man samarbeider med. Vi ønsker at systemet ikke skal kreve mye datakunnskap, slik at flere kan ta det i bruk enn de som i dag bruker e-læringssystemer.

I en utdanningssituasjon er kommunikasjon viktig, både kommunikasjon mellom elev og lærer og kommunikasjon elever imellom. Elever vil noen ganger trenge hjelp fra lærer, og ved øvingsoppgaver må en lærer rette og gi tilbakemelding på disse. I gruppearbeid, der samarbeid mellom elever spiller en stor rolle, vil det være nyttig å kunne ha mulighet til å dele felles dokumenter og å kunne kommunisere synkront, det vil si direkte kommunikasjon med tekst, lyd eller levende bilder. Eksisterende e-læringssystemer har liten eller ingen mulighet for synkron kommunikasjon. De systemene som har synkrone muligheter er ofte kompliserterte, og er rettet mot folk som har en god del datakunnskap fra før. For oss blir det derfor viktig å se på kombinasjonen av komponenter i et slikt system, for å få det så anvendelig som mulig.

Videre følger noen overordnede mål for systemet vårt:

- Det skal være et helhetlig system, som dekker flest mulige behov. Ett system der man kan gjøre alt.
- Systemet skal være enklere å bruke enn dagens systemer, slik at flere utdanningsgrupper kan ta det i bruk.
- Systemet skal tilby god tilgjengelighet og skalerbarhet. Det skal være mulige å bruke systemet på flere forskjellige måter, blant annet ved hjelp av nettleser og mobiltelefon.

Sist men ikke minst, ønsker vi at systemet vårt skal være en måte å vise frem og skape interesse for tanken om e-læring. Det kan for eksempel brukes til demonstrasjon ved videre arbeid mot et optimalt e-læringssystem. Det er viktig å få appell rundt om i samfunnet når man utvikler et nytt konsept, og da hjelper det gjerne å ha et eksempel å vise frem.

Se for øvrig Appendiks A for mer informasjon om nedlasting og demonstrasjon av programmet.

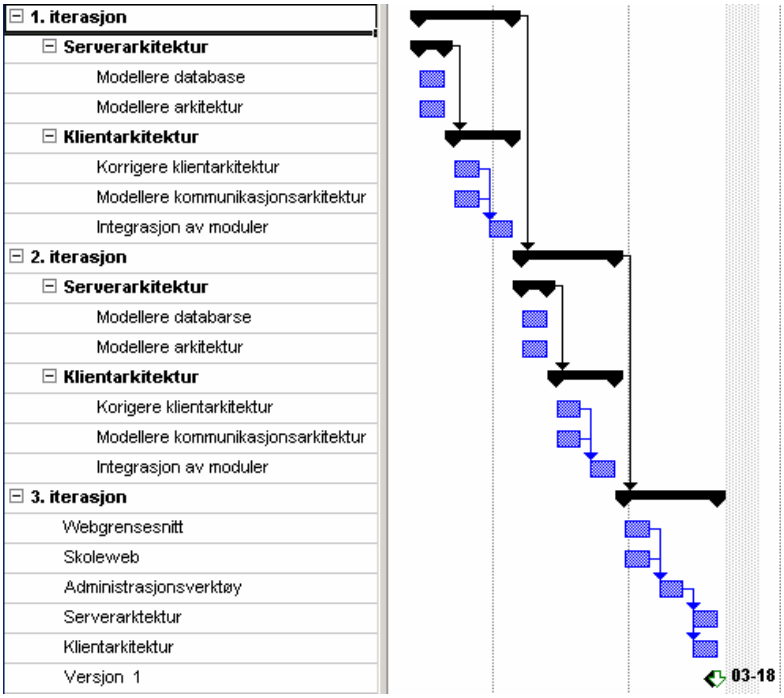
1.4 Gjennomføringen av prosjektet

1.4.1 Forprosjekt

Arbeidet med denne diplomoppgaven begynte allerede høsten 2004, da vi gjennomførte forprosjekt i fordypningsemnet TDT4705 IKT i læring. I dette forprosjektet la vi de fleste av planene for utviklingen av programmet. Vi lagde også en prototyp av brukergrensesnittet til et e-læringssystem. Denne prototypen ble i hovedsak laget ut fra våre egne oppfatninger om hvordan et e-læringssystem burde være. Vi satset på å lage en distribuert løsning basert på Microsoft .NET 2.0-rammeverket. Flere klienttyper ble vurdert, men vi endte opp med å implementere en klientapplikasjon og en webklient. Dette ga oss god innsikt både i teknologien vi hadde valgt å bruke, og i hvordan vi ville legge opp den videre utviklingen av systemet.

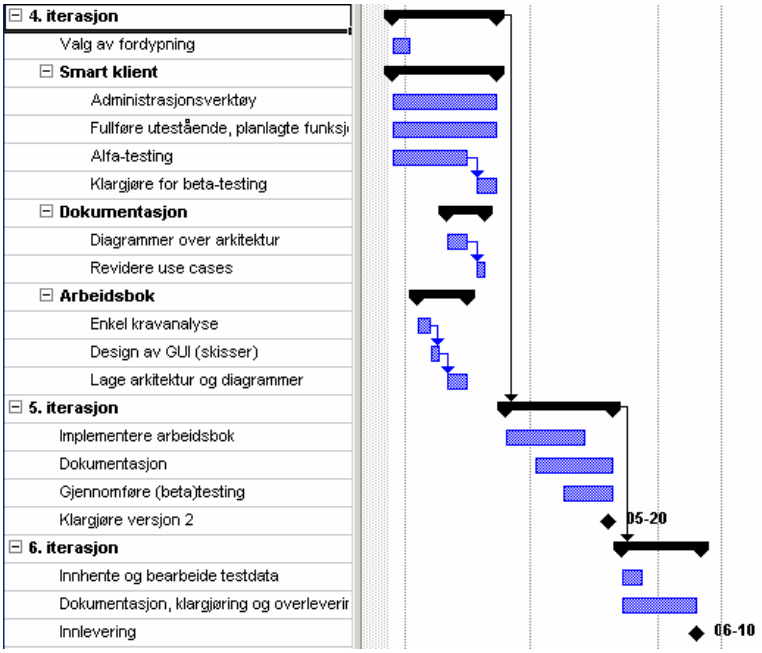
1.4.2 Hovedprosjekt

I diplomoppgaven våren 2005 har vi valgt å gå videre med klientapplikasjonen vi laget før jul. Selv om det bare fantes en betaversjon av .NET 2.0-rammeverket, så vi mange fordeler ved å bruke denne versjonen, og bestemte oss for å bruke den i videre implementasjon. Med erfaringer fra brukergrensesnittprototypen fant vi blant annet ut at vi måtte jobbe med den grunnleggende arkitekturen for å få klientprogrammet mer effektivt. Vi forkastet derfor prototypen, og begynte å utvikle en langt mer omfattende og solid arkitektur som skulle danne grunnlaget for all kommunikasjonen mellom klientprogram og server.



Figur 1-1 Iterasjon 1-3: Implementasjon av grunnleggende arkitektur og brukergrensesnitt

Frem til påske i 2005 holdt vi hovedsakelig på med implementasjon av den grunnleggende delen av arkitekturen og brukergrensesnittet (Figur 1-1). Tiden etter påske hadde vi på forhånd satt av til å gå nærmere inn på ett bestemt emne som vi ville implementere videre i systemet vårt. Fordi vi var usikre på hva som ville gi oss mest igjen, ventet vi til påske med å bestemme oss. Valget stod mellom arbeidsbok, alternative klienter og video/lyd-funksjon. Vi fant fort ut at arbeidsboken utgjorde en så viktig del av systemet at det var vanskelig å se noen helhet i systemet uten den. Vi valgte derfor å fokusere mye på arbeidsbok etter påske (Figur 1-2). I tillegg til å fortsette implementasjonen etter påske, demonstrerte vi også programmet for et lite utvalg personer, for å høre deres synspunkter.



Figur 1-2 Iterasjon 4-6: Videre utvikling av klient, personlig verktøy og gjennomføring av brukertesting

1.4.3 Arbeidsmetode

Vi har begge erfaring med Rational Unified Process (RUP) fra tidligere prosjekt. Vi har derfor hentet inspirasjon fra denne utviklingsprosessen og har satset på en iterativ og inkrementell utvikling. Utviklingsprosessen deles inn i flere iterasjoner der vi i slutten av hver iterasjon har hatt en kjørbar del av programmet. Hver iterasjon har vært inkrementell, det vil si at vi har lagt til mer og mer funksjonalitet etter hvert. Vi har også utviklet de vanskeligste delene av programmet først for å eliminere kritiske deler tidlig i prosessen. Dette har blant annet medført at vi prioriterte å få den grunnleggende kommunikasjonen mellom server og klient på plass tidlig i prosjektet.

Siden vi bare er to stykker, har ikke samkjøring av utviklingen vært noe problem. Vi har stort sett jobbet side om side, men har likevel delt inn i noen hovedansvarsområder:

Anvarsområde	Ansvarlig
Analyse og design	Håvar Wehus og Arnstein Berg
Server	Håvar Wehus
Agenter på klient	Arnstein Berg
Design av brukergrensesnitt	Håvar Wehus og Arnstein Berg

Figur 1-3 Ansvarsområder

Ansvarlig for utviklingen må ses på som den personen som hele tiden har hatt oversikt, og ledet arbeidet i forhold til dette området. For øvrig har vi begge jobbet mye på alle deler av systemet.

2 Problembeskrivelse

Det er mange spørsmål som stilles rundt det å innføre e-læringssystem og nettbasert undervisning i forskjellige former for opplæring. Gir nettbasert undervisning et bedre tilbud til studentene enn vanlig klasseromsundervisning? Vil det støtte alle former for undervisning? Vil det være enkelt nok til at alle kan klare å bruke det? Og ikke minst, er det til elevenes beste å sitte og jobbe ved hver sin datamaskin? Vi vil i dette kapitlet gå gjennom mange av aspektene, og på den måten prøve å danne en grunnforståelse for hvordan et e-læringssystem bør bygges opp, og hvilke funksjoner det må kunne tilby.

Mange av opplysningene i dette kapitlet er innhentet ved enkle empiriske metoder, og baserer seg i stor grad på egne erfaringer og refleksjoner. Selv om vi har underbygget disse så godt vi kan med opplysninger fra andre personer og medier rundt oss, er ikke disse i seg selv ment å være en forskningsstudie, men kun hypoteser for hvordan vi kan gå fram for å lage et godt e-læringssystem. Hypotesene vil så bli forsøkt verifisert gjennom implementeringen og utprøvingen av dette systemet.

2.1 Bruksområder

I dette kapitlet vil vi forsøke å se på mulige bruksområder for e-læringssystemer, og hvordan disse bruksområdene spiller inn på utformingen og bruken av et slikt system.

2.1.1 Forelesning

Hvis man tar utgangspunkt i grunnskole og videregående skole, skjer det meste av undervisningen som forelesninger. I de fleste andre opplæringssituasjoner har man også en eller annen form for forelesning. Det vil si at en person presenterer kunnskapen gjennom en monolog. I de fleste situasjoner foregår denne monologen synkront, det vil si at kunnskapen når brukeren akkurat i det øyeblikk foreleseren legger den fram. Et eksempel der man imidlertid ikke har det slik, er ved brevkurs.

En av de aller enkleste formene for nettbasert forelesning kan være en enkel webside der man presenterer kunnskapen som tekst og kanskje også med illustrasjoner. Dette vil omtrent tilsvare et brevkurs. Videre kan man godt utvide tilbudet med å legge ut opptak av lyd og bilde, på samme måte som man kan legge ved lyd- og videokassetter i et brevkurs. Det vi snakker om her er en asynkron monolog, det vil si at budskapet ikke når mottakeren i samme øyeblikk som det blir sagt eller skrevet.

Som nevnt tidligere foregår de fleste forelesningene synkront, normalt ved at læreren står foran klassen. Man kan oppnå synkron kommunikasjon på internett også, ved å sende tekst, lyd og video direkte. Teknologi for slik direktesending har vært forholdsvis utbredt på internett i flere år allerede.

Hva oppnår man så med å sende en forelesning direkte? Så lenge en forelesning bare består av en monolog, vil denne forelesningen være å se på omtrent som et vanlig fjernsynsprogram. Man kunne like gjerne tatt opp forelesningen og latt folk se den ved en annen anledning. Noen TV-programmer er best å se direkte, nemlig de programmene der seerne kan ringe inn. Her får man da en synkron dialog mellom seere og TV-verten.

Det kan altså se ut som det er først når man innfører en dialog i forelesningen at synkron kommunikasjon er nødvendig. I praksis vil dette si at elevene tar aktivt del i

forelesningen ved å stille spørsmål til foreleser, eller eventuelt at foreleser stiller spørsmål til tilhørerne. Man kan også si at direktesending gir en følelse av å være med på noe, men dette har ingen stor praktisk betydning for formidlingen av kunnskap. Dermed kan det virke som det bare er én ordentlig grunn til at man vil ha forelesningene direkte overført, nemlig at det går an å ha dialog mellom lærer og elever.

Uavhengig av om forelesningen skal foregå synkront eller asynkront, må den på en eller annen måte annonseres, slik at folk blir klar over den. En eller annen form for felles oppslagstavle er nok vanligst å bruke til dette. Er forelesningen asynkron vil det sannsynligvis være både mulig og hensiktsmessig å la brukerne selv velge når de vil se forelesningen. Er den derimot synkron, må man annonsere tidspunkt for når den starter, og alle brukerne som vil se forelesningen må være til stede da.

I en vanlig forelesning er det foreleseren som styrer ordet, og alle henvender seg til han eller hun når de vil si noe. Foreleser har normalt også kontroll over forskjellige hjelpemidler til undervisningen, slik som tavle, overhead, lydanlegg, videokanon og annet. I enkelte tilfeller kan man også tenke seg at tilhørerne kan bli gjort til forelesere, for eksempel dersom elevene skal presentere noe for klassen. Det bør da være mulig for den som er ansvarlig for forelesningen å delegere tilgangen til ordstyringa og tavla til en av tilhørerne.

2.1.2 Gruppearbeid via internett

Mens det er en person som er ansvarlig på en forelesning og all dialog går gjennom denne personen, vil det i gruppearbeider være naturlig at man har en dialog med flere likestilte parter. Siden gruppearbeid normalt bygger på en dialog, heller enn en monolog, er det naturlig å tro at synkron kommunikasjon vil være viktigere her enn i en forelesning. På et gruppemøte i den fysiske verden vil man normalt sitte sammen i et rom der alle kan se alle de andre, og ordet styres i fellesskap. Man kan gjerne ha hjelpemidler som bord, tavle eller lignende der alle har anledning til å tegne, skrive eller vise fram ting til de andre.

Dette ligner mye på en forelesning, men forskjellen er i hovedsak at man ikke har en ansvarlig person som styrer ordet. Alle kan delta aktivt ved å ta ordet og snakke til hvem de vil mens de andre hører på. Det er heller ingen som alene har tilgang til hjelpemidlene i rommet.

Som nevnt er synkron kommunikasjon godt egnet til å formidle en dialog, men det er også mulig å gjøre en dialog asynkron. Det er noe av dette vi ser i leserbrev i en avis, der man kan legge inn innlegg og svare på andres innlegg. Normalt tar denne kommunikasjonsformen forholdsvis lang tid, fordi man må vente en stund på svar fra den andre parten. Enda vanskeligere blir det hvis det er flere enn to parter, fordi man ikke vet hvem som leser hva på hvilket tidspunkt, og det er lett at to personer snakker forbi en tredjemann. Unntaket kan kanskje være i de tilfeller der man for eksempel arbeider relativt selvstendig med et prosjekt over tid, og bare ønsker innspill fra andre, uten at dette haster.

Grupper kan formes på flere måter, men to hovedprinsipper kan man trekke ut. Det ene er at gruppen blir dannet før man setter seg sammen for å jobbe, mens det andre er at gruppen blir dannet etter at man har satt seg ned. For eksempel kan en lærer bestemme på forhånd hvem som skal være på gruppe med hvem, eller han kan bare la folk sette seg sammen, og så danne samarbeidsgrupper ut fra hvordan elevene har plassert seg.

Grupper kan bestå i kortere eller lengre tid. I mange tilfeller vil man kunne komme og gå fra gruppen, og man vil at det man holder på med fortsatt skal være der når man kommer tilbake. Eventuelt kan en eller flere gruppemedlemmer ta med seg det man har jobbet med, slik at man kan jobbe videre på det senere.

2.1.3 Generelt egenarbeid

Ved egenarbeid vil det ofte ikke være så stort behov for å kommunisere med andre. Spørsmål kan imidlertid dukke opp og da kan man for eksempel få behov for å kontakte faglærer på en enkel måte. Igjen dukker dialogen opp, og man kan tenke seg at synkron kommunikasjon er å foretrekke. Man bør imidlertid også å ha asynkron kommunikasjon tilgjengelig, dersom man for eksempel jobber utenom vanlig arbeidstid.

For enkelte personer kan det trolig være nyttig å ha med seg arbeidet sitt rundt, slik at man kan jobbe med det på flere forskjellige steder. Sånn sett kan et nettbasert e-læringssystem komme godt med også til egenarbeid, siden man da kan ha tilgang til sitt arbeid fra en hvilken som helst datamaskin med internettilgang.

Man kan også tenke seg at elevene sitter og arbeider med datamaskiner i klasserommet eller på en datasal på skolen, der læreren er fysisk til stede. Hvis man sitter sammen med andre i et klasserom, vil e-læringssystemet hovedsakelig brukes til oppgaveløsning på et eget arbeidsområde, og kommunikasjonsdelen vil ikke være i bruk. Selv om man nå ikke har bruk for nettbasert kommunikasjon, kan systemet likevel gjøre nytte som en sentral lagringsplass, og dessuten til å distribuere oppgavene som skal løses. Hvis oppgavene ligger ferdig i systemet på forhånd, kan læreren enkelt instruere elevene til å gå inn i systemet og løse oppgavene.

2.1.4 Tester, øvinger og oppfølging

Med tester og øvinger mener vi oppgaver som skal leveres inn til læreren for å bli evaluert. Gjennomføring av tester og øvinger kan i stor grad fungere på samme måte som egenarbeid (jf. avsnittet over), men i stedet for å bare lagre arbeidet på sitt eget område, vil det nå være praktisk om systemet kan formidle innleveringen til læreren som skal evaluere. Tilbakemeldingen til eleven bør også kunne formidles enkelt av systemet.

Ved tester der man bare er interessert i å få ett riktig svar fra eleven (flervalgsoppgaver, gloseprøver, enkle matematikkstykker, osv), kan man også tenke seg at e-læringssystemet kan rette øvingene automatisk og gi tilbakemelding til eleven umiddelbart. Læreren kan da i etterkant få beskjed om hvordan eleven klarte oppgaven.

2.1.5 Administrasjon

Noe som er praktisk ved å samle alt i ett system, er muligheten til å administrere det sentralt. Man trenger bare ett register over alle personer, alle fag og alt pensum. Dette gjør det enkelt å legge til og oppdatere informasjon, og enkelt å ta sikkerhetskopier.

2.2 Fagområder

2.2.1 Teoretiske fag

Dersom man tar for seg noe av den mest grunnleggende lærdommen som blir

formidlet i skolesystemet, nemlig å lære å lese, skrive og regne finnes det allerede mange dataprogrammer for å hjelpe elevene med dette. Programmer for nettbasert undervisning finner man derimot ikke blant disse. Samtidig vet vi at lese- og skrivetrening i de aller fleste tilfeller skjer med både lærer og elever fysisk til stede på skolen, og da trenger de jo heller ikke å kommunisere via internett. Skal vi spekulere i om det er nødt til å være slik, er det naturlig å tro at elevene på dette stadiet ikke vil forstå hvordan de skal kunne operere et slikt system. Denne delen av opplæringen er på en måte mer praktisk enn teoretisk, og man kan ikke finne en god erstatning for det å ha en lærer fysisk til stede (jf. Kap. 2.2.2). Når man derimot har lært litt, kan man kanskje bruke datamaskinen som et verktøy for å øve inn kunnskapen enda bedre.

Et nettbasert undervisningssystem bør selvsagt ha muligheten til å tilby oppgaveløsing, og selv om dette er mulig å gjøre helt uten bruk av internett, vil det være en del fordeler knyttet til å ha systemet nettbasert (jf. Kap. 2.1.3). Videre kan man også tenke seg at det for eksempel går an å bruke tekstprating (chatting) over internett til å trene på å lese og skrive, men da bruker man ikke egentlig kommunikasjonen til noe nyttig i seg sjøl. Skrivetrening på data trener imidlertid bare tastaturskriving, mens det legges stor vekt på at barna skal lære å skrive for hånd i starten.

Når man har kommet litt lengre opp i skolesystemet, og er forbi den innledende fasen med å lære å lese og skrive, åpner det seg imidlertid mange måter å bruke datasystemet i undervisningen. Undervisningen kan bestå av nettbaserte forelesninger, der et skrevet pensum foreleses. Det er også vanlig å ha prosjekter og andre oppgaver som kan løses gruppevis. Man kan også jobbe med oppgaver alene, enten på skolen med læreren tilstede, eller hjemme. I det hele tatt kan man nok si at teoretiske fag er godt egnet i et e-læringssystem, fordi de i stor grad inneholder skrevet informasjon som er lett å formidle gjennom en datamaskin.

2.2.2 Praktiske fag

I praktiske fag er skrevet informasjon ikke lenger den dominerende informasjonskilden. Dermed er det grunn til å tro at disse fagene i utgangspunktet er mindre egnet å undervise på internett enn teoretiske fag. Dette kan man imidlertid kompensere for med ved å ta i bruk mer avansert teknologi. Ofte lærer man ved å se på hva en annen gjør, og så gjøre det samme selv. Hvis e-læringssystemet, ved hjelp av direktesendt video og lyd, kan gjøre det mulig for elevene å se og høre hva læreren gjør og sier, kan dette føre til at også praktiske fag er mulig å gjøre tilgjengelige over internett.

I mange praktiske fag er man avhengig av spesielt utstyr for å gjøre det man skal. I prinsippet kan fjernundervisning sikkert fungere i heimkunnskap, fordi de fleste har tilgang til et kjøkken der de måtte befinne seg. Verre er det dersom man for eksempel er avhengig av maskinelt utstyr som man ikke har tilgang til utenfor undervisningsstedet. Her er det altså egentlig ikke en mangel ved e-læringssystemet som setter grensene, men derimot en mangel ved det fysiske stedet der eleven sitter.

Et annet eksempel som vankelig lar seg kombinere med nettbasert undervisning er svømmeopplæring. En ting er at det mest sannsynlig er veldig upraktisk å dra med seg en datamaskin inn i en svømmehall og prøve å sette opp en internettforbindelse der. De fleste klarer heller ikke å lære seg å svømme bare ved å få vist svømmetak av en lærer, men er gjerne avhengig av fysisk hjelp. Her er det altså lærerens fysiske bistand som er nødvendig.

Det er også andre eksempler på opplæring der man er avhengig av å ha lærer til stede. Skal du for eksempel lære å kjøre bil, er det mest naturlig å ha læreren i bilen sammen med deg. Selv om noen av de aller mest innovative sjelene kanskje vil protestere, vil nok de fleste være enige om at det å ha en bærbar datamaskin med direkteforbindelse til kjørelæreren i setet ved siden av seg, ikke er en god løsning. Her er det rett og slett en risikofaktor ved å ikke ha læreren fysisk til stede. Eleven risikerer å skade både seg og andre, dersom læreren ikke har mulighet til å gripe inn fysisk.

I tillegg har man en del fagområder der den fysiske kontakten med personer rundt seg er av avgjørende betydning. Dette gjelder for eksempel de som er under opplæring for å bli lærere (ikke nettlærere, vel å merke) eller forskjellige typer foredragsholdere, skuespillere, osv. Her er det den fysiske kontakten med mennesker rundt seg man er ute etter, og et e-læringssystem vil trolig virke mot sin hensikt i disse tilfellene.

For praktiske fag er det altså ikke like enkelt å gå over til nettbasert undervisning. Det er iallfall vanskelig å gi et overordnet svar, siden praktiske fag er veldig forskjellige. Man må antakelig gå inn og vurdere hvert fag for seg.

2.3 Målgrupper

I dette kapitlet vil vi se på hvilke områder i skole og opplæring et nettbasert datasystem kan være nyttig. Vi tenker her på et system der det er lagt stor vekt på kommunikasjon og samarbeidslæring, men som også kan brukes som et enkelt verktøy der elevene kan løse oppgaver osv.

2.3.1 Barneskole

En typisk barneskole er plassert relativt nært elevenes egne hjem. Med få unntak bor elevene hjemme og reiser til og fra skolen hver dag. Undervisningen er i liten grad basert på selvstendig arbeid. Det meste av opplæringen foregår i nært samarbeid med lærer og medelever. Barneskolen utgjør også en svært viktig faktor i barnas utvikling gjennom lek og fysisk kontakt med andre barn. Her inngår også mange praktiske fag, som formingsfag, heimkunnskap og gymnastikk.

Det å være fysisk til stede er altså sentralt i barneskolen. Det er dette som er situasjonen i dag, og sannsynligvis er det også slik man ønsker å ha det. Undervisningen her er normalt ikke en enkel forelesning, men gjerne oppstykket med mye dialog mellom lærer og elever, samt mye oppgaveløsning med lærerens hjelp. Det å få all denne kommunikasjonen til å fungere godt i et distribuert datasystem hvor elevene ikke befinner seg fysisk på samme sted, vil kreve så avansert teknologibruk at det vil mest sannsynlig vil gå langt over det elevene har av kunnskap og forståelse i forhold til bruk av datasystemer. Dessuten ønsker man jo gjerne den fysiske kontakten som skolen gir mellom elevene.

Det er grunn til å tro at på barneskolenivå vil bruken av et datasystem stort sett begrense seg til enkel oppgaveløsning. Det er rett og slett ikke behov for å tilby elevene mulighet til å samarbeide fra forskjellige fysiske lokaliseringer. Dersom skolen likevel skulle velge å gå til anskaffelse av et slikt system på bakgrunn av muligheten til å bruke det til enkel oppgaveløsning, vil det ikke være noe i veien for at kommunikasjons- og samarbeidsfunksjonene finnes i systemet. Funksjonene bør imidlertid kunne ryddes til side, slik at de ikke forstyrrer vanlig bruk. Disse funksjonene kan dessuten være nyttige å ha i tilfeller der elever må være borte fra skolen i lengre tid. De kan også være nyttige for lærere, dersom det medfører at det går an å planlegge og lage oppgaver, rette prøver og drive med annen elevoppfølging

fra forskjellige fysiske lokaliseringer.

2.3.2 Ungdomsskole

Beliggenheten er ikke i fullt så stor grad som barneskolen knyttet til elevenes bosted, men de aller fleste elevene reiser til og fra skolen hver dag. Undervisningen er i litt større grad basert på selvstendig arbeid, der elevene kan bli gitt i oppgave å finne en del av informasjonen tilknyttet opplæringen selv. En del samarbeid med medelever kan være aktuelt, men det foregår stort sett på skolen. De praktiske fagene fra barneskolen er også aktuelle her.

Veldig mange av aspektene fra barneskolen gjelder også for ungdomsskolen. Arbeidet er imidlertid noe mer selvstendig her, og dessuten må man anta at elevene i denne aldersgruppen har større forutsetninger for å kunne nyttiggjøre seg av samarbeidsfunksjonene i et slikt system. Det vil altså trolig være gjennomførbart at elever på ungdomsskolenivå sitter fysisk atskilt og samarbeider over internett, men om det er en bedre løsning enn at elevene møter opp på skolen selv, kan diskuteres. Mange av elevene i denne alderen er ikke modne nok ennå til å ta ansvar for egen læring, og det vil trolig være vanskeligere å få dem til å gjøre ting dersom de jobber hjemmefra, enn om de faktisk sitter på skolen. Dessuten må elevene fortsatt møte på skolen for gjennomføringen av en del praktiske fag.

Samarbeidslæring over internett vil nok ikke bli den vanlige måten å drive undervisning på i ungdomsskolen, iallfall ikke med dagens teknologi, men det kan i mange tilfeller gi et nyttig supplement til andre undervisningsformer. For eksempel kan man tenke seg at dersom elever ønsker å samarbeide på eget initiativ utenom skoletiden, kan de gjøre dette i et nettbasert grupperom. I fag som fra før innebærer bruk av data, kan man også tenke seg at bruk av e-post, elektronisk overlevering av filer, sentral arkivering osv. er nyttige funksjoner som et e-læringssystem bør kunne tilby.

2.3.3 Videregående

På videregående nivå er en del av elevene, særlig utenfor byene, ikke lenger bosatt på deres opprinnelige hjemsted. De fleste bor imidlertid nær skolen, og møter normalt på skolen hver dag. Mens både barne- og ungdomsskoler er veldig likt oppbygd, er det en del mer variasjon i hvordan videregående skoler er organisert.

Det er stor variasjon i organiseringen mellom ulike linjer. Allmenne, økonomiske og administrative fag er nok den linja som skiller seg minst fra ungdomsskolen. Fagrettede videregående skoler har en langt større andel av praktiske oppgaver, og mye av undervisningen foregår utenom vanlige klasserom. Videre har man mulighet for å ta privatistfag. Dette er fag som ligger utenfor skolens vanlige undervisningstilbud, og de tilbys ofte av andre undervisningsinstitusjoner, gjerne helt eller delvis som fjernundervisning.

Når det gjelder muligheter for å ta i bruk samarbeidslæring over internett, så kan man enkelt si at videregående går ut fra barne- og ungdomsskolen i to forskjellige retninger. På den ene siden har vi allmennfag og andre linjer som i hovedsak har teoretiske fag. Dette er tilfeller hvor nettbasert undervisning kan være praktisk, siden elevene på dette nivået jobber mer selvstendig, og fordi de antakelig er mer modne for å ta i bruk denne teknologien. Videre underbygges dette av at elevene ikke er like stedsbundne til skolen, og kanskje vil ha muligheten til å jobbe fra sitt eget hjemsted i perioder.

På den annen side har man de fagrettede videregående skolene, der praktiske fag står mye mer sentralt. Her vil det være langt større forskjell mellom hver skole og hvert fag, og man må trolig vurdere hvert enkelt tilfelle individuelt når man skal se på om det er mulig og hensiktsmessig å gjøre undervisningen nettbasert. Se for øvrig kapittel 2.2.2.

2.3.4 Høyere utdanning

I overgangen fra å være elev til å bli student, følger også for alvor overgangen fra å påtrykkes lærdom til å faktisk måtte oppsøke den selv. Dette overfører mye av ansvaret som læreren har i lavere utdanning, til eleven. Dermed er ikke spørsmålet rundt det om læreren klarer å passe på at elevene gjør det de skal, så viktig lenger.

På mange universiteter og høyskoler er allerede internett mye brukt i undervisningssammenheng. Mye av faginformasjonen formidles på internett, og e-post er mange steder hyppig i bruk til å gi beskjeder både blant lærere og elever. Videre er det mange steder der elevene selv melder seg opp til eksamen i forskjellige fag via internett. Det må imidlertid også sies at mange studiesteder ikke tilbyr denne funksjonaliteten, særlig når det gjelder de mer praktiske fagområdene.

På høyere utdanningsnivå foregår altså det meste av læringen ved at elevene jobber selvstendig på eget initiativ. Riktig nok er dette begynt å endres litt tilbake fra det veldig frie, til mer obligatoriske innleveringer osv. Likevel må man kunne si at studentene i stor grad har kontroll over egen læring. Mange ser derfor dette som et reelt alternativ, som kan gjøre hverdagen deres lettere, og dermed vil være mer imøtekommende overfor innføringen av slike systemer.

Den grunnen som sannsynligvis er aller viktigst for å innføre et system for samarbeidslæring innen høyere utdanning, vil være at studentene ikke bor i nærheten av studiestedet. Disse studiene er normalt knyttet til byene der det er dyrt å bo. Mange av de som ikke bor så veldig langt fra studiestedet velger kanskje å bo hjemme og må da bruke en del av tiden sin på pendling, mens andre ser seg nødt til å bosette seg i byen bare fordi pendlingen blir for dyr eller tidkrevende. Dersom et slikt system kan medføre at man kan jobbe hjemmefra, vil mange av disse personene slippe å reise til studiestedet hver dag, og de kan spare tid og penger til pendlingen. Noen av de som har bosatt seg i byen ville kanskje også kunne latt være å gjøre det, dersom dette systemet hadde vært innført. Selv om mange studenter fortsatt vil foretrekke å bosette seg i nærmiljøet til studiestedet, vil nettbasert undervisning medføre at de som ikke ønsker dette, har et alternativ.

2.3.5 Annen opplæring

Man kan også se for seg samarbeidslæring over internett brukt i helt andre opplæringssituasjoner. Noe av det vi ikke har nevnt i foregående kapitler er for eksempel folkehøyskoler, voksenopplæring, etterutdanning og dessuten alle mulige former for kurs som tilbys ellers i samfunnet. Når det gjelder folkehøyskoler er veldig mange av disse i stor grad praktisk rettet, og det vil dermed ikke være så lett å tilpasse dem til nettbasert undervisning.

For voksenopplæring bør det helt klart være store muligheter. Voksenopplæring foregår i forholdsvis liten skala, og tilbudene er derfor ikke alltid tilgjengelige i nærmiljøet. Siden mange i denne fasen av livet har jobb og familie som de må ta seg av ved siden av, er de ofte ikke er så veldig mobile. Da er fjernundervisning løsningen for mange. Dette gjelder for så vidt også for etterutdanning. I dagens samfunn der

livslang læring står sentralt vil man være i en utdanningssituasjon resten av livet etter at man har fullført studier.

Når det gjelder andre mer generelle kurs, er mulighetene mange. Det kan for eksempel være salgskurs til ansatte i en bedrift, ADR-kurs for transport av farlig gods, tegne- og malekurs, pianotimer, osv. Her er det nesten uendelig mange muligheter. Alle kursene vil selvsagt ikke være like godt egnet til nettbasert undervisning, men alt som hittil har vært tilgjengelig på brevkurs og annen fjernundervisning, må man anta at det også er mulig å gjøre kurset nettbasert. For øvrige kurs må man sannsynligvis vurdere hvert kurs for seg, men det som står om fagområder (Kap. 2.2) gjelder i stor grad også her.

2.3.6 Elever som trenger ekstra tilpasninger

2.3.6.1 Funksjonshemmede

Det er i dag en selvfølge at man prøver å tilpasse undervisningen best mulig til funksjonshemmede. I et datasystem kan dette både være vanskeligere og lettere enn tilpasninger man må gjøre i den fysiske verden. Siden systemet tillater at man kan jobbe fra en datamaskin som er plassert hvor som helst, og få den samme tilgangen som om man var på skolen, kan dette trolig være til stor hjelp for de som har nedsatt mobilitet. Det vil muligens være nødvendig med tilpasning av betjeningsutstyret til datamaskinen, men det er neppe mye mer omfattende enn det man måtte ta tilpasset for eleven ved vanlig fysisk undervisning.

Syns og hørselshemming finnes det også mange hjelpemidler for i dag. For de som ikke er alt for svaksynte, er det et relativt lite problem å forstørre hele eller deler av skjermen. Hørselshemming vil ikke være noe stort problem ved generell datajobbing, men når man komme til den synkrone kommunikasjonen med lyd, vil dette selvsagt være noe man må ta hensyn til. Som nevnt er det mulig å bruke tekst under forelesning, og uten at vi har fått bekreftet det, må vi også anta at det vil fungere greit å formidle tegnspråk via en døvetolk med webkamera.

Når der gjelder helt blinde, har disse naturlig nok en del vansker med å bruke datamaskiner. Det finnes imidlertid programmer som kan lese fra en skjerm med syntetisk stemme. Dersom man kombinerer dette med tekstlig informasjon som overføres over nettet, kan man trolig oppnå en brukbar løsning. Den synkrone kommunikasjonen med lyd, vil dessuten være relativt uproblematisk for blinde å nyttiggjøre seg av. Ved å tilby pensum og faginformasjon elektronisk og kombinere dette med en skjermleser, vil man antakelig gjøre langt mer informasjon tilgjengelig for blinde, enn det som i dag er tilgjengelig for dem gjennom lydbøker og blindeskrift.

2.3.6.2 Elever med lærevansker

Å tilpasse et system til elever med lærevansker er nok ikke så likefrem som å tilpasse for konkrete funksjonshemminger. Dersom en elev egentlig vil lære, men bare trenger litt mer tid enn andre vil systemet vårt kunne tilby dette gjennom asynkrone forelesninger der man kan bruke så lang tid man vil på å se gjennom. Videre gjør muligheten for fjernarbeid at man kan gjøre ting hjemmefra på kveldstid, og dermed få mer tid til oppgaveløsning enn det som er avsatt i skoletiden.

For elever som er ukonsentrerte og umotiverte er det vanskelig å si hvilken effekt innføringen av et nettbasert undervisningssystem vil gi. Det kan selvsagt hende at en elev vil bli fascinert av systemet, synes det er morsomt å bruke, og dermed få mer lærelyst. Men det er minst like sannsynlig at en elev som blir satt til å jobbe med dette

systemet, uten direkte tilsyn, vil sluntre unna enda mer enn før. Det er vanskelig å utforme programmet slik at dette ikke skjer, fordi man i alle dataprogrammer er avhengig av at folk selv ønsker å bruke dem.

Teknisk del

3 Teknisk innledning

Frem til nå har vi i stor grad diskutert skolenes oppbygning og funksjon generelt, samt en del generelle tanker om hva e-læringssystemer bør tilby av funksjonalitet, og det er på tide å gå mer konkret inn på hvilke funksjoner vi vil ta med i systemet vårt. Vi prøvde innledningsvis i prosjektet å sette opp en oversikt over de funksjonene vi selv kunne identifisere fra vår egen skolegang, og se disse i forhold til de tidligere diskuterte temaene.

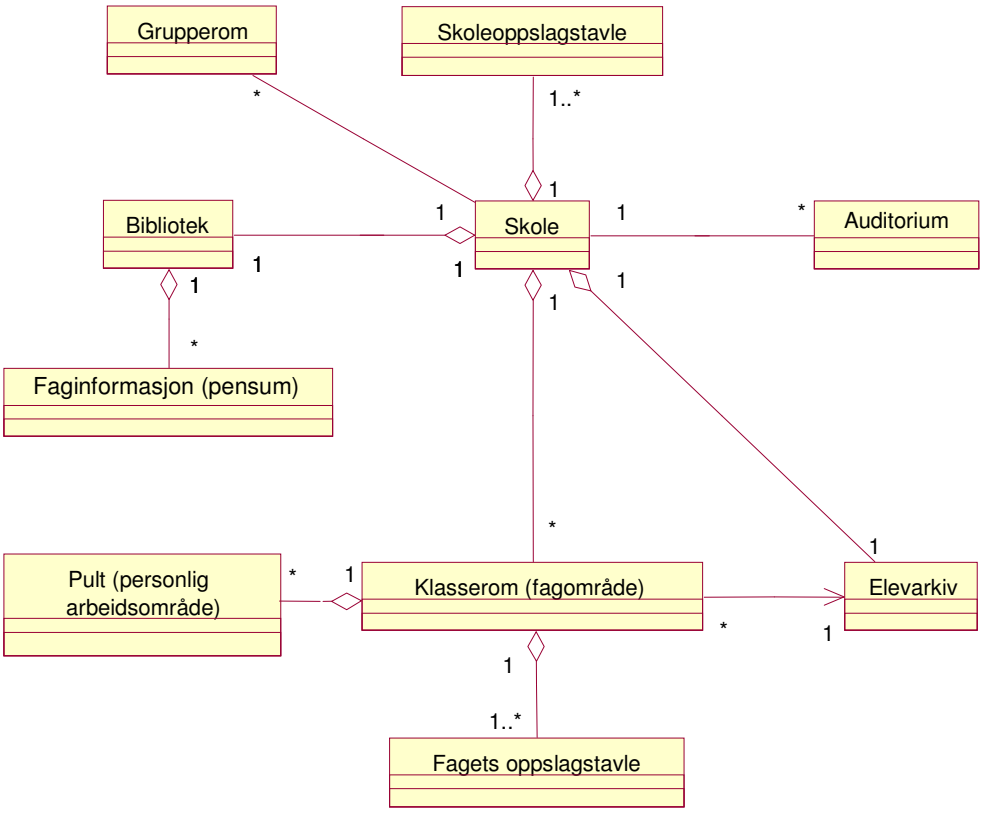
Vi kom da frem til at vi vil ha med personlige verktøy, samarbeidsfunksjoner og generell fagrelatert informasjon i systemet vårt. Det var viktig å prøve å avgrense disse funksjonsområdene litt i forhold til hverandre, slik at man får god oversikt over hvilken informasjon som tilhører hvilket område. Samtidig er samspillet mellom disse funksjonene viktig, og det må være mulig å kunne bruke funksjoner fra alle områdene samtidig. Det personlige området ditt, kalt skrivebord, er spesielt viktig å alltid ha med seg rundt om i systemet, fordi du gjerne skal notere og ha tilgang til dine arbeider i forbindelse med andre aktiviteter.

Videre vil vi ta med noen begreper, vurderinger og valg vi har sett på i detalj, før vi fortsetter med krav og use cases som systemet vårt baserer seg på.

3.1 Drøfting av generelle begreper

3.1.1 Den fysiske skolen

Med begrepet ”den fysiske skolen” mener vi skolen slik den er uten bruk av elektroniske hjelpemidler. Dette er den skolen mange elever, iallfall i de lavere klassetrinn, er vant til i dag. Vi kan sette opp følgende diagram over hvordan mange skoler er oppbygd i dag.



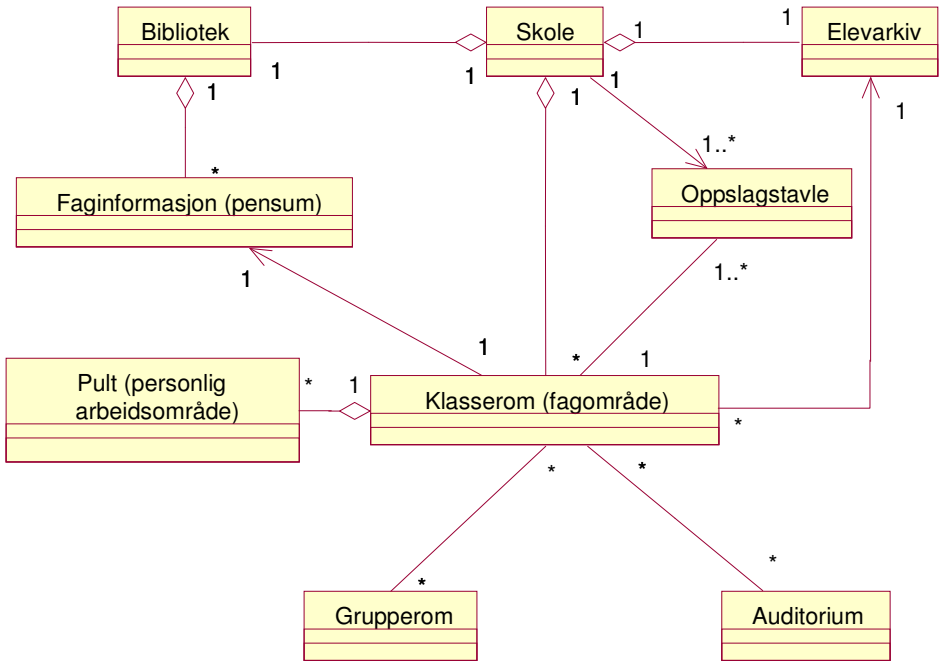
Figur 3-1 Den fysiske skolen

De fleste skoler har flere klasserom, som i mange tilfeller tilhører et bestemt fag eller en bestemt klasse. I tillegg har man gjerne andre rom som brukes i undervisningssammenheng, slik som auditorier, grupperom (kollokvierom), lesesaler og aulaer/vrimlearealer. Ofte finnes det også bibliotek og lærerrom/kontorer.

Når det gjelder informasjonsflyten i den fysiske skolen skjer denne på mange plan. Man har selvsagt det at læreren foreleser for elevene, men elevene kan også samarbeide seg i mellom. I tillegg har man en del informasjon som slås opp på oppslagstavler og lignende, på flere forskjellige steder på skolen. Informasjon kan man selvsagt også få direkte fra bøker, som finnes i et bibliotek, i klasserommet eller som man har med selv.

Ved de aller fleste undervisningssteder er også det å kunne teste at elevene faktisk tilegner seg kunnskapen de skal, en viktig funksjon. Det vil si at elevene får oppgaver som de skal leveres inn eller presenteres for lærerer, og ofte følger det også at læreren skal gi tilbakemelding på disse oppgavene.

3.1.2 Den virtuelle skolen



Figur 3-2 Den virtuelle skolen

Vi satser på å lage en modell som gjenspeiler mest mulig av strukturen i den fysiske skolen, men det er selvsagt også en del ting man kan gjøre enklere virtuelt enn man kan i den fysiske verden. For eksempel trenger vi ikke ta noe hensyn til ledige rom, så her er det bare å lage så mange og så store rom man trenger. Derfor legger vi opp til at man kan ha ubegrenset antall forelesningsrom (auditorier) og grupperom.

Oppslagstavler kan knyttes både direkte til skolen og til forskjellige klasser og fag, slik man i den fysiske skolen både har oppslagstavler ute i fellesarealer og inne i klasserommene. Kommunikasjonen som ofte kan foregå i fellesarealer, vil gjenspeiles i systemet gjennom praterom (gjerne med mulighet for lyd/bilde) som vil være tilgjengelig gjennom hele systemet.

3.1.3 Nettbasert undervisning/E-læringssystem

Det er gitt mange mer eller mindre gode definisjoner på begreper som nettbasert undervisning, e-læring og samarbeidslæring. For vår oppgave har vi valgt å se bort fra disse definisjonene, og satser heller på en mer åpen tolkning av begrepene. Når vi snakker om nettbasert undervisning, mener vi kort og godt undervisning der hovedvekten av informasjonen som utveksles går via internett. Og når vi snakker om et e-læringssystem, mener vi kort og godt et datasystem som kan benyttes i undervisningssammenheng. Rene administrasjonssystemer for skoler er imidlertid ikke inkludert i tolkningen.

Samarbeidslæring over internett tolker vi som at man bruker internett til å formidle undervisningen, og at læringsprosessen inkluderer mer enn bare elev-lærer-kommunikasjon. For eksempel kan dette dreie seg om gruppearbeid på internett der flere elever er involvert.

3.1.4 Samarbeidsrom

Samarbeidsrom, eller bare rom, er et sentralt begrep i systemet vårt. Det gjenspeiler en avgrenset, virtuell plassering i systemet. Tanken vår er at denne typen rom skal fungere akkurat som et rom gjør i den virkelige verden. Det er en møteplass for folk, og definerer samtidig en avgrensning i forhold til mennesker og informasjon rundt en selv.

Et rom kan brukes til å holde en forelesning, det kan brukes av en gruppe mennesker til å sitte og jobbe i, eller det kan bare være et oppholdsrom der folk kan kommunisere med hverandre. Det er disse tre romtypene vi har tatt utgangspunkt i for å definere hovedbruksområdene for rom i en skolesammenheng. I systemet vårt heter romtypene henholdsvis forelesningsrom, grupperom og praterom.

I den vanlige fysiske verden er rom noe en skole gjerne har veldig begrenset med, og de fleste rommene er normalt knyttet til en bestemt oppgave. I systemet vårt kan man derimot lage så mange rom man bare måtte ønske, og i stor grad bestemme hvilket formål de skal tjene. Et rom kan være så lite eller så stort man ønsker og man kan opprette nye rom når man får behov for det. Et rom kan være et klasserom, et auditorium, et kollokvierom, en lesesal, et bord i kantina, en plass i skolegården, og så videre. Når man vil møte andre mennesker lager man rett og slett et rom der man kan møtes.

3.2 Vurdering av brukergrensesnitt

Helt siden starten av forprosjektet høsten 2004, har vi jobbet med tanken om å lage et brukergrensesnitt som er så bra at det kan føre til at flere tar i bruk e-læringssystemet. Dette er selvsagt et veldig vagt formulert mål, som det vil være nesten umulig å finne en god måte å teste, men det kan iallfall ses på som en inspirasjonskilde for arbeidet vårt.

Mer konkret så ville vi satse på å gjøre brukergrensesnittet mest mulig intuitivt ved å bruke metaforer fra den fysiske skolen. Vi ville blant annet bruke uttrykk som rom, skrivebord, arbeidsbok, osv. I tillegg ville vi gjerne bruke ikoner og grafikk som elevene enkelt gjenkjenner. For eksempel kan man representere funksjonen for å gå til et rom, med et bilde av en dør. Ellers vil vi også prøve å benytte farger, og til en viss grad fonter, til å kategorisere funksjoner og gjøre systemet mer oversiktlig.

Hvis man går mer over mot det funksjonelle ved brukergrensesnittet, er det mange forskjellige måter å legge opp grensesnittet på. To viktige prinsipper er verktøyprinsippet og dialogprinsippet [2]. Verktøyprinsippet (Direct Manipulation) vil si at brukergrensesnittet representerer en samling med verktøy som brukeren kan benytte. Han kan for eksempel velge en bok og skrive inn i den, eller han kan velge en penn og en tavle, og begynne å tegne. Dialogprinsippet (Agents) vil si at brukeren ledes gjennom programmet i form av spørsmål av typen ”Hva vil du gjøre nå?”. Man kan kort si at verktøyprinsippet tilbyr mange veier direkte til funksjonene, mens dialogprinsippet leder deg gjennom systemet etter mer eller mindre forhåndsdefinerte veier gjennom systemet.

Dersom man vet nøyaktig hva man skal gjøre, er verktøyprinsippet raskt og effektivt, fordi brukeren ledes direkte til den funksjonen du vil benytte. Er man derimot usikker på hvilken funksjonalitet man har til rådighet i systemet, kan det være betryggende og forklarende å la dialogprinsippet lede brukeren rundt. Dialogprinsippet kan, iallfall i teorien, lages så godt at man ikke trenger noen opplæring for å bruke systemet, men

siden man gjerne må gå gjennom flere spørsmål for å få gjort forskjellige handlinger, vil det fort bli ineffektivt for brukere som likevel vet hvor de skal.

Siden systemet vårt dreier seg om en relativt stor samling funksjoner tilknyttet e-læring, vil vi i hovedsak satse på å bruke verktøyprinsippet. Dette kan kanskje virke litt rart i forhold til at vi også ønsker å lage brukergrensesnittet enkelt og intuitivt, men vi må også tenke på at folk skal bruke dette systemet til daglig, og da er rett og slett ikke dialogprinsippet effektivt nok. Vi håper at bruk av ikoner, farger og andre virkemidler kan kompensere for en del av den økte vanskelighetsgraden som verktøyprinsippet medfører. Dessuten kan det bli aktuelt å lage dialogbaserte grensesnitt for enkelte funksjoner i programmet, gjerne som et tillegg til å ha funksjonene tilgjengelige som verktøy.

3.3 Vurdering av utviklingsteknologi

Vi ville lage et e-læringssystem som er nettbasert. Dette er fordi man i et e-læringssystem skal formidle informasjon og kunnskap mellom personer som ofte befinner seg på forskjellige steder. I tillegg er det viktig at systemet er enklest mulig å få til å fungere på flest mulige typer datamaskiner. Dette vil føre til at systemet kan brukes av mange personer og under mange forhold. I startfasen av prosjektet tok vi utgangspunkt i teknologier som er vanlig å bruke i tilknytning til internett.

Teknologi	Opphav	Plattform	Objektorientert	Kodespråk	Database
ASP	Microsoft	Windows	Nei	VBScript	ADO, ODBC
.NET	Microsoft	Windows	Ja	VB, C#	ADO.NET, ODBC
JSP/Servlets	Sun Microsystems	Alle	Ja	Java	JDBC
PHP	Open source	Unix/Windows	Begrenset	PHP	ODBC

Tabell 3-1 Vurderte standarder for programmering av internettapplikasjoner

Siden prosjektgruppens medlemmer hadde erfaring med objektorientert analyse og design fra før, var det naturlig for oss å velge en objektorientert teknologi. Vi har mye erfaring med programmering i Java, men ingen av oss har prøvd JSP eller servlets før. Ett av gruppens medlemmer hadde en del erfaring fra .NET, etter å ha brukt denne teknologien i et saks- og arkivsystem for offentlig sektor.

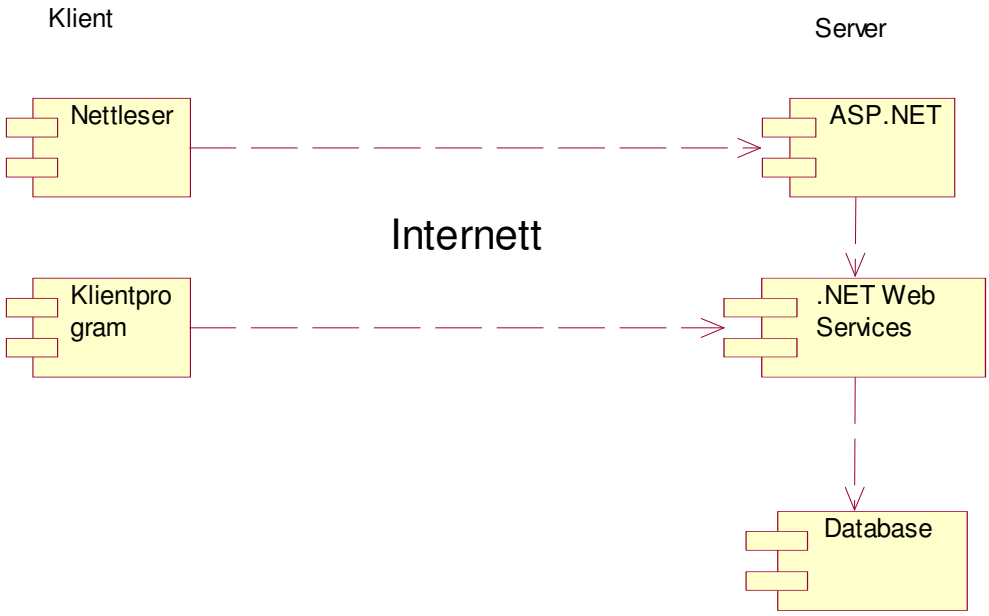
For å slippe at begge gruppemedlemmene måtte lære seg JSP og servlets, var det enklere for det ene medlemmet som ikke kunne .NET å lære seg det. I hovedsak innebar dette å lære seg C# som ligner mye på Java. Det at en av oss har erfaring med .NET fra før lettet også planleggingen, siden vi da visste hva som er mulig og hensiktsmessig å få til med teknologien, noe vi ikke hadde visst dersom ingen av oss kjente teknologien.

Det er samtidig en del andre fordeler med hele .NET-teknologien som vi kan benytte oss av i systemet vårt. Blant annet er .NET tett integrert både med webtjenester og Microsoft SQL Server (database), og dette kan man bruke til å lage en god kobling mellom funksjonalitet, grensesnitt og database. Dette er vesentlig siden vi har tenkt å lage et system som kan fungere på et vidt spekter av klienter. Det finnes også gode grafiske verktøy for utvikling av brukergrensesnitt i .NET, noe som er veldig nyttig når vi skal utvikle et så stort system innenfor en så avgrenset tidsperiode. Vel å merke støttes ikke .NET av så mange forskjellige klientmaskiner som for eksempel Java gjør, men det er likevel godt utbredt og i sterk vekst.

Tidlig i prosjektperioden ble vi klar over at versjon 2.0 av .NET hadde kommet ut som betaversjon. Vi så litt på denne versjonen og fant fort ut at en del svake punkter i versjon 1 var kraftig forbedret. Dette førte også til at vi endret planene om hva slags klient vi i hovedsak skulle lage. Vi hadde opprinnelig sett for oss en løsning der man i hovedsak bruker nettleseren, men at man bruker tilleggsmoduler for å oppnå avansert funksjonalitet. Vi hadde også sett for oss at det gikk an å lage et eget selvstendig kjørende klientprogram, men dette ville være så mye arbeid å implementere at vi ikke tenkte å gjøre det som en del av prosjektarbeidet. Med .NET 2.0 og dens teknologi for smarte klienter fant vi imidlertid ut at det å lage et klientprogram var det enkleste.

3.3.1 .NET 2.0

.NET er Microsoft sin forholdsvis nye teknologi for plattformuavhengighet, med stor vekt på kommunikasjon over internett. .NET kan brukes til å lage alle slags applikasjoner, men det er distribuerte applikasjoner med server og klienter som virkelig får utnyttet denne teknologien til sitt fulle.



Figur 3-3 Oversikt over hovedkomponentene i en distribuert .NET-applikasjon

3.3.1.1 Webtjenester

En klart nyttig side ved .NET er såkalte webtjenester (web services). Dette kan man se på som websider for programmer. En webtjeneste ligger på en webserver på internett og har en vanlig internettadresse som man kan velge å åpne. Men i stedet for å få opp en webside som er leselig for oss mennesker, får man opp en XML-side med informasjon for andre dataprogrammer. Man kan gi tjenesten parametere (i XML-format) og dermed få den til å utføre operasjoner på serveren, for så å få et svar tilbake i XML. Når man lager programmer i .NET blir oversettingen til og fra XML gjort automatisk, slik at det fortoner seg for utvikleren som om man bare kaller en enkel funksjon på serveren. Alle data som er serialiserbare, inkludert hele sett med data fra en database, kan overføres som XML mellom klient og server.

Tjenestene vil i de fleste slike applikasjoner, inkludert vår egen, stå for de funksjonelle delene av applikasjonen. Det er også disse tjenestene som står for all kommunikasjon med databasen. I det ferdige programmet vil tjenestene stå for all tilgangskontroll og det meste av kommunikasjonen mellom klientene (unntatt direkteoverført synkron kommunikasjon, såkalt peer-to-peer) vil gå gjennom dem. De blir derfor å se på som en ryggrad i applikasjonen.

Siden webtjenester bruker HTTP-protokollen, vil det nødvendigvis bli en del programmeringsmessige problemer med disse. Blant annet må alle funksjonskall gå fra klienten til serveren. Det er ikke mulig å få serveren til å kalle tilbake til klienten. Dette hadde helt klart vært praktisk når man for eksempel skal oppdatere visning av påloggede brukere, chat-vindu, osv. Grunnen er ganske enkelt at http fungerer slik at klienten kaller serveren og serveren svarer. Det er ikke mulig at serveren tar initiativ og kaller klienten.

Videre har ikke serveren oversikt over hvilke klienter som er tilkoblet og den husker ikke hvem som er hvem fra et kall til det neste. For hvert eneste kall til serveren er dermed klienten nødt til å identifisere seg spesifikt for å bli gjenkjent av systemet. Dersom man vil at serveren skal si fra til klienten om oppdateringer, er løsningen å gå utenom webtjenester og i stedet sette opp en varig forbindelse mellom de to, der begge kan sende data uavhengig av hvem som kaller først. Dette vil blant annet være nødvendig for å få til ekte synkron dataoverføring.

3.3.1.2 ASP.NET

Webtjenestene gir som nevnt ikke brukeren noe fornuftig informasjon i seg sjøl, og derfor trenger man et annet program på toppen av disse. Dette kan enten være et eget selvstendig program som kjører hos brukeren, eller man kan bruke ASP.NET. Det ASP.NET gjør, er å generere websider som er leselige for mennesker. ASP.NET er et program som kjører på webserveren, og som i stedet for å hente inn kommandoer fra mus og tastatur og presentere et grafisk resultat på en skjerm, henter kommandoer fra kallene til webserveren og presenterer HTML-kode som en nettleser kan vise for brukeren.

Fordelen med å bruke ASP.NET som klient for webtjenestene, er at den kjører på serveren. Dermed trenger man ikke å installere noe eget program på klientmaskinen, det holder å ha en vanlig nettleser. Det vil igjen si at man kan bruke nesten en hvilken som helst datamaskin som klientmaskin. Ulempen er imidlertid at nettlesere har veldig begrensede muligheter til å programmeres. Synkron overføring av lyd og video, er blant noe av det man ikke uten videre klarer å få til med en enkel nettleser.

3.3.1.3 Smarte klienter

Særlig i forbindelse med .NET 2.0 har begrepet smarte klienter (Smart Clients) [3] begynt å florere. I dette begrepet ligger en del ideer, som blant annet:

- En smart klient skal tilby brukeren like god funksjonalitet og brukeropplevelse som man har i et selvstendig program.
- Den skal nyttiggjøre lokale maskinressurser der dette er mulig, slik at man kan avlaste serveren og bruke datakraften som finnes i klientmaskinene.
- Den skal normalt være tilkoblet en server og effektivt utveksle data med omverdenen.
- Den skal selv kunne finne ut om man ikke har forbindelse til serveren, og tilby

brukeren best mulig funksjonalitet under gjeldende omstendigheter.

- Den skal være enkel og installere, og den skal holde seg oppdatert ved nye versjoner.

I .NET 2.0 er det mange nye funksjoner som gjør det enkelt å lage smarte klienter. Blant annet er punktet om å installere og oppdatere tatt hånd om i funksjonen "Click-once". Dette er en funksjon som gjør at man kan installere et program bare ved å klikke på en link på en hjemmeside. Programmet installeres da på brukerens maskin, og man får en vanlig snarvei i start-menyen dersom man ønsker det. Hver gang det startes vil programmet sjekke adressen det opprinnelig ble lastet ned fra, og finne ut om det er ny versjon tilgjengelig, og installere denne.

Gjennom å la klienten kommunisere med webtjenester, sikrer vi også en enkel kobling mot serverapplikasjonen. Vi kan bufre mye data på klientsiden, komprimere data før overføring, og på den måten utnytte mye av datakraften på klientmaskinen til å gjøre belastningen på overføringslinjen og serveren minst mulig.

Vi legger opp til at programmet skal tilbys på en internettadresse. Når man går inn på denne vil klientmaskinen identifiseres, og man vil automatisk kunne få tilbudt den beste klienten. Dersom man har mulighet til å kjøre . den smarte klienten vil man få opp denne, ellers vil man få opp websiden.

Siden det er mange funksjoner som bare vil være tilgjengelig i klientprogrammet, vil det være ønskelig at flest mulig av brukerne kan bruke dette. Dette programmet bør derfor være mest mulig plattformuavhengig. Vi har valgt å lage dette programmet i .NET, noe som gjør at det kan kjøres på alle systemer som støtter .NET-rammeverket. Foreløpig gjelder dette bare nyere Windows-plattformer, men det er mulig at det etter hvert vil bli tilgjengelig på andre systemer også.

Vi kunne lagd klientprogrammet i Java, da Java også støtter webtjenester, og dette ville gjort programmet mer plattformuavhengig. Vi valgte likevel å bruke .NET siden vi fra før hadde planlagt å lage serverdelen av systemet i .NET. Visual Studio har mange funksjoner for å generere brukergrensesnitt raskt og effektivt, noe som passet veldig bra siden vi hadde begrenset med tid til utvikling, og gjerne ville få flest mulig funksjoner til å teste. Det er imidlertid ikke noe problem å i ettertid lage et nytt klientprogram i Java, eller for den saks skyld en annen teknologi, og få dette til å spille sammen med serverdelen av systemet.

3.3.1.4 Database

Datalagring og gjenfinning er sentralt i et e-læringssystem, og dette innebærer at man må ha en stabil database i bakgrunnen. .NET-teknologien støtter de fleste databaser gjennom ADO.NET og ODBC-standarden, men for å utnytte alle funksjonene fullt ut trenger man Microsoft SQL Server. Vi har derfor valgt å bruke denne i vårt prosjekt.

3.3.1.5 Verktøy

Vi valgte å bruke Visual Studio .NET 2005, som støtter .NET 2.0, som utviklingsverktøy for hele systemet. Det er foreløpig bare en betaversjon som er tilgjengelig, men denne har fungert tilfredsstillende for vårt prosjekt. Dette programmet lar oss utvikle både programmer, webtjenester, ASP.NET-sider og database.

3.4 Valg av arkitektur

3.4.1 Serverarkitektur

I henhold til ideen om smarte klienter (kap 3.3.1.3) skal serveren gjøre minst mulig beregninger, og i stor grad brukes bare til å lagre og formidle informasjon. Dette er også hensiktsmessig med tanke på at vanlige servermaskiner gjerne inneholder forholdsvis lite regnekraft, men de er til gjengjeld veldig effektive til å flytte data. Nå vil sannsynligvis ikke et e-læringssystem inneholde så veldig mange krevende beregninger, men vi kan komme borti det for eksempel i forbindelse med komprimering av grafikk og lyd. Denne funksjonaliteten er det derfor hensiktsmessig å legge på klientmaskinen så sant det er mulig.

Vi legger opp til å ha enkle webtjenester som ikke gjør stort mer enn å formidle data mellom klient og database. Databasen er imidlertid optimalisert for å kombinere data til datasett, og det vil derfor være dumt å ikke benytte seg av kombinasjonsalgoritmene som ligger her når man for eksempel skal liste alle brukere som er med i en klasse, etc. Vi har tenkt å gjøre mest mulig prosessering i klienten, minst mulig generelt på serveren, og mest mulig i databasen igjen.

En ting som vi imidlertid er nødt til å gjøre generelt på serveren, er brukerautentisering og tilgangsstyring. Det holder ikke å gjøre dette på klientmaskinen, da det ikke er noe i veien for at andre personer kan lage egne klienter med funksjonalitet som strider med den som er tiltenkt systemet. Det er derfor helt nødvendig for å bevare integriteten i systemet, at serveren sjekker alle forespørsler den får, både for å sjekke om handlingen er tillatt og om innsendte data er konsistente.

3.4.2 Klientarkitektur

Vi har som nevnt valgt å fokusere på en smart klient som hovedklient for systemet vårt i dette prosjektet. Dette innebærer som nevnt ovenfor, at vi legger stor vekt på funksjonaliteten i klienten. Innledningsvis hadde vi tenkt å lage et system der størsteparten av dataprosesseringen foregår på serveren, og brukergrensesnittet kun består av en enkel webside eller lignende. Fordelen med denne løsningen er at man stiller veldig få krav til klientmaskinen, og dermed oppnår man veldig stor tilgjengelighet, siden man i praksis kan bruke programmet fra nesten hvilken som helst type datamaskin.

Ulempen ved å gjøre all prosessering på serveren er imidlertid at det stiller store krav til serveren. Med dagens PC-er, som gjerne har prosessor på flere GHz, 1 GB internminne, og flere hundre GB disk, er det ingen problemer å kjøre selv tunge beregninger på svært kort tid. Denne datakraften finnes jo allerede i brukernes maskiner, og den er dessuten langt rimeligere enn en server med like stor kapasitet som alle klientmaskinene samlet. Systemet blir også langt mer skalerbart når man flytter prosesseringen ut til brukerne, og man slipper i større grad å oppgradere serveren for å kompensere for økt bruk av systemet. Det er grunn til å tro at skalerbarhet er viktig for mange skoler, ikke minst i innledningsfasen for slike systemer, siden det er sannsynlig at bruken vil gradvis øke i det flere og flere begynner å bruke systemet til flere og flere oppgaver.

Vårt system vil trolig ikke inneholde så veldig mange prosessorkrevende operasjoner, siden det hovedsakelig brukes til å formidle informasjon som ikke trenger noen videre bearbeidelse. Unntaket er riktig nok sending av grafisk informasjon og lyd. Skal vi for eksempel formidle et bilde, vil vi satse på å komprimere dette hos brukeren som

legger det inn, sende det og lagre det komprimert på serveren. Når det så skal hentes frem igjen, overfører man bare den komprimerte versjonen til klientmaskinen, og lar denne dekomprimere det før visning. Ved å komprimere data før man sender dem fra brukeren, vil man også spare nettverket en del, noe som gjerne er en flaskehals i slike systemer, samt at man sparer lagringsplass på serveren.

Videre kan man spare både nettverk og server, dersom klienten ikke gjør forespørsler oftere enn nødvendig. Det er for eksempel lite vits i å lese inn informasjon fra serveren hver eneste gang man skal tegne opp en liste for brukeren. Vi vil derfor gå inn for å lage et forholdsvis avansert buffersystem i klienten som kan holde styr på hvilken informasjon du allerede har lest inn, og hva som må hentes inn fra serveren. På denne måten kan man spare mange kall mot serveren, og dermed også både nettverksressurser og serverressurser.

Som en følge av at vi kommuniserer med serveren ved hjelp av webtjenester, oppstår også det problemet at serveren ikke kan si fra til klienten at noe er oppdatert. Dermed er man avhengig av at klienten selv spør serveren etter oppdateringer. Dette er enda en god grunn til å bufre data, fordi da kan man lage seg et lokalt selvoppdaterende buffer, en slags primitiv databasereplikasjon. Da kan brukergrensesnittet alltid oppdateres mot det lokale bufferet, samtidig som man har egne prosesser (heretter agenter) som går i bakgrunnen og oppdaterer dette bufferet. Agentene vil da si fra til brukergrensesnittet hva som er oppdatert, slik at man slipper at opptegning til skjerm og lignende skjer mer enn nødvendig. Enda en fordel med denne løsingen er at agentene vil nå kontrollere all kommunikasjon mellom klienten og serveren, og siden dette skjer på bare ett nivå, vil det være langt enklere å optimalisere nettverks- og serverbelastningen.

4 Krav

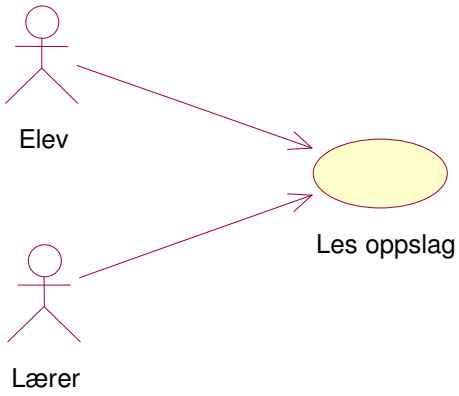
Dette kapittelet beskriver de kravene som er stilt til systemet.

I starten av utviklingen lagde vi story boards for de mest grunnleggende funksjonene vi mente et e-læringsprogram skulle ha. Dette ble gjort tidlig i prosessen fordi vi ville legge stor vekt på brukeropplevelsen, og vi ville derfor ha et grafisk tillegg til use casene. Story boardene ble kun brukt innledningsvis, og har ikke blitt oppdatert etter dette.

Sammen har use casene og kravene har dannet grunnlaget for det senere arbeidet i prosjektet. Underveis i prosjektet har vi oppdatert use casene og kravene slik at de til enhver tid stemmer overens med programmet.

4.1 Use caser

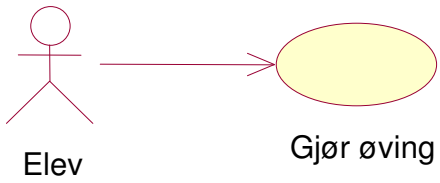
4.1.1 Les oppslag

<table>
<tr><td colspan="2">Use case id: U01 Opprettet: 2004.09.08 Sist endret: 2004.09.14</td></tr>
<tr><td colspan="2">Beskrivelse:
Brukeren kan lese meldinger om et fag på oppslagstavlen til et fag eller klasse</td></tr>
<tr><td> 
</td><td>Forhåndsbetingelse:
Programmet er startet og klar til bruk, og brukeren er logget på.

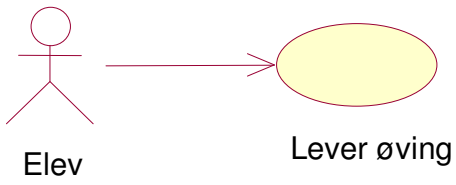
Hendelsesforløp:
• Brukeren velger fag eller klasse
• Fagets/klassens hovedside med oppslag vises

Sluttbetingelse:
Aktøren har lest det som står på oppslagstavlen.</td></tr>
<tr><td colspan="2">Utestående spørsmål:
•</td></tr>
</table>

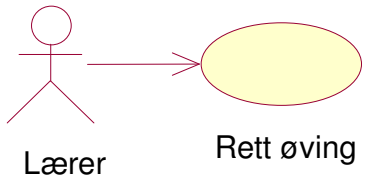
4.1.2 Gjør øving

Use case id: U02	Opprettet: 2004.09.08	Sist endret: 2004.04.05
Beskrivelse: En elev vil gjøre en øving.		
 	Forhåndsbetingelse: Programmet er startet og klar til bruk. En øving har blitt lagt ut. Hendelsesforløp: • Eleven trykker på en ny øving på internett eller på startsiden • Øvingsteksten vises i nettleseren i programmet • Eleven får opp et tomt øvingsdokument i editoren. • Eleven svarer på øvingsoppgaven. Sluttbetingelse: Eleven har gjort seg ferdig og velger å lagre eller levere øvingen.	
Utestående spørsmål: •		

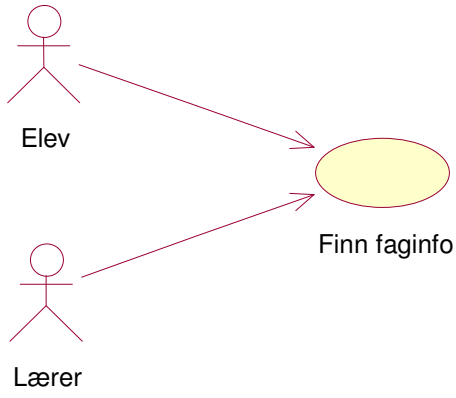
4.1.3 Lever øving

Use case id: U03	Opprettet: 2004.09.08	Sist endret: 2005.01.20
Beskrivelse: Eleven kan levere øvinger til faglærer.		
 	Forhåndsbetingelse: Programmet er startet og klar til bruk Hendelsesforløp: • Eleven går til sine egne oppgaver • Alle øvingsoppgavene vises • Eleven åpner øvingen som skal sendes • Eleven sender øvingen • Hvis brukeren ikke er logget på, vil han få beskjed om at øvingen blir lagret og at han må levere den på nytt når han er logget på. Sluttbetingelse: Oppgaven er sendt og er klar til retting.	
Utestående spørsmål: •		

4.1.4 Rett øving

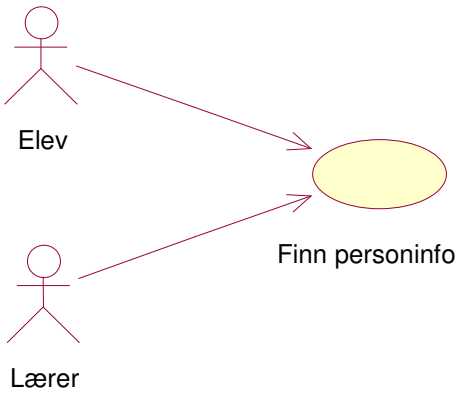
Use case id: U04	Opprettet: 2004.09.08	Sist endret: 2004.04.05
Beskrivelse: En lærer har fått inn svar på øving, og vil rette disse.		
	Forhåndsbetingelse: Programmet er startet og klar til bruk. Lærer har fått inn svar på en øving. Hendelsesforløp: • Læreren går inn i mappen der han har øvingen • Læreren får opp en liste over svarene som har blitt levert. • Han velger å åpne en av svarene. • Læreren kan skrive kommentarer eller rette øvingen. • Når læreren har rettet ferdig, kan han sende en rettet versjon tilbake Sluttbetingelse: Læreren har sendt den ferdig rettede øvingen tilbake til eleven.	
Utestående spørsmål: •		

4.1.5 Finn faginfo

Use case id: U05	Opprettet: 2004.09.08	Sist endret: 2004.09.14
Beskrivelse: En elev eller lærer kan finne informasjon om et fag på fagsiden.		
	Forhåndsbetingelse: Programmet er startet og klar til bruk, og brukeren er logget på. Hendelsesforløp: • Brukeren velger det faget han vil ha informasjon om • Informasjonen presenteres for brukeren og han kan navigere seg rundt. Sluttbetingelse: Brukeren har fått tak i ønsket informasjon.	
Utestående spørsmål: • Skal alle kunne se all informasjon om alle fag?		

4.1.6 Finn personinfo

Use case id: U06	Opprettet: 2004.09.08	Sist endret: 2004.09.14
Beskrivelse: Brukeren kan finne informasjon om andre brukere i systemet.		



Forhåndsbetingelse:
Programmet er startet og klar til bruk, og brukeren er logget på.

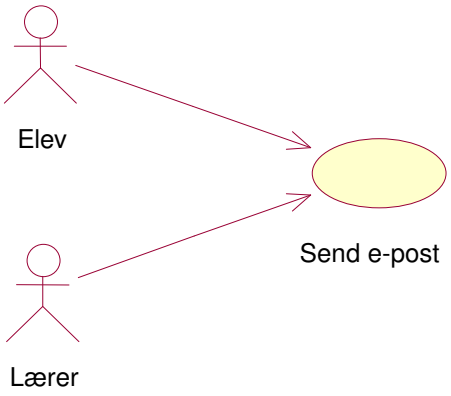
Hendelsesforløp:
- Brukeren går til en oversikt over brukere
- Brukeren velger hvilken bruker han/hun vil ha informasjon om fra lister, eller ved å søke.
- Informasjonen som eleven har tilgang til å se om den andre brukeren vises.

Sluttbetingelse:
Brukeren har fått tak i den informasjonen han/hun ønsket og har tilgang til.

Utestående spørsmål:
- Hvem skal ha tilgang til å se opplysninger om hvem?

4.1.7 Send e-post

Use case id: U07	Opprettet: 2004.09.08	Sist endret: 2004.04.05
Beskrivelse: En bruker ønsker å skrive og sende e-post.		



Forhåndsbetingelse:
Programmet er startet og klar til bruk.

Hendelsesforløp:
- Brukeren går til e-postleseren
- Brukeren åpner et ”ny e-post”-vindu.
- Brukeren skriver inn mottakeradresse og ønsket tekst.
- Han kan også velge å sende med vedlegg.
- Brukeren trykker på send-knappen.
- Hvis brukeren ikke er logget på får han melding om at e-posten vil bli lagret, og at han må sende den igjen når han logger på.

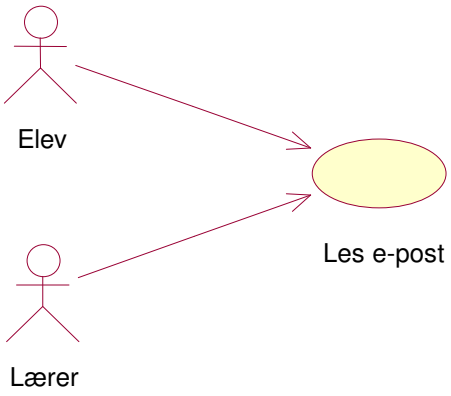
Sluttbetingelse:
E-posten ble sendt.

Utestående spørsmål:
-

4.1.8 Les e-post

Use case id: U08	Opprettet: 2004.09.08	Sist endret: 2004.04.05

Beskrivelse:
Brukeren har en egen e-postboks der innkommende e-post fra andre brukere av systemet og utenforstående blir lagt. Disse kan brukeren lese.



Forhåndsbetingelse:
Programmet er startet og klar til bruk.

Hendelsesforløp:
- Brukeren går til e-postleseren
- Hvis brukeren er pålogget får han opp en liste med alle e-postene i innboksen sin.
- Hvis brukeren er frakoblet vises en liste med e-postene som ligger lagret lokalt.
- Brukeren trykker på e-posten han vil lese.
- E-posten vises i et vindu på skrivebordet.
- Brukeren kan også velge å få opp e-posten i et eget vindu.

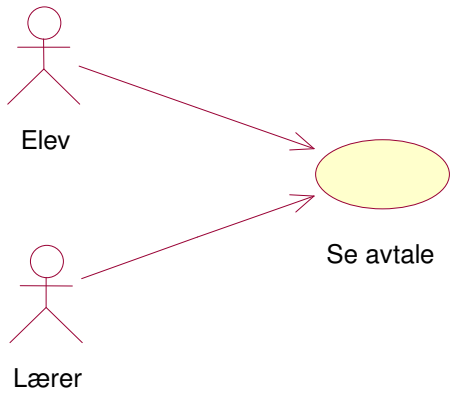
Sluttbetingelse:
Brukeren har fått lest e-posten.

Utestående spørsmål:
- Hvordan skal brukeren bli varslet om nye meldinger?

4.1.9 Se avtale

Use case id: U09	Opprettet: 2004.09.08	Sist endret: 2004.04.05

Beskrivelse:
Brukeren vil se informasjon om en avtale i kalenderen sin.



Forhåndsbetingelse:
Programmet er startet og klar til bruk.

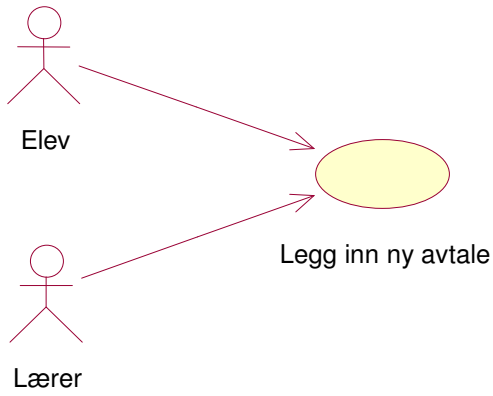
Hendelsesforløp:
- Brukeren åpner kalenderen.
- Brukeren kan velge om han vil se en dag, en uke eller en måned om gangen.
- Brukeren dobbelklikker på en avtale for å få opp mer utfyllende informasjon.

Sluttbetingelse:
Brukeren har fått sjekket den avtalen han skulle.

Utestående spørsmål:
-

4.1.10 Legg inn ny avtale

Use case id: U10	Opprettet: 2004.09.08	Sist endret: 2005.04.05
Beskrivelse: Brukeren vil legge inn en ny avtale i kalenderen sin.		



Forhåndsbetingelse:
Programmet er startet og klar til bruk.

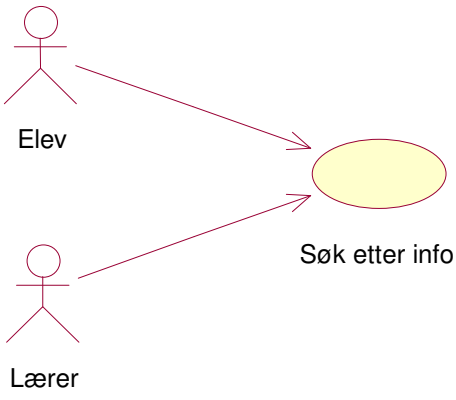
Hendelsesforløp:
- Brukeren åpner kalenderen.
- Brukeren trykker på ”ny avtale”-knappen eller dobbelklikker på en dag i kalenderen.
- Et nytt vindu vises, der brukeren kan skrive inn informasjon om den nye avtalen.
- Brukeren velger å lagre avtalen.

Sluttbetingelse:
Avtalen ble lagret.

Utestående spørsmål:
- Skal det være mulig å bli påminnet om avtalen?

4.1.11 Søk etter info

Use case id: U11	Opprettet: 2004.09.08	Sist endret: 2004.09.14
Beskrivelse: Brukeren skal finne ønsket informasjon. Det kan være informasjon om et fag eller pensumlitteratur.		



Forhåndsbetingelse:
Programmet er startet og klar til bruk, og brukeren er logget på.

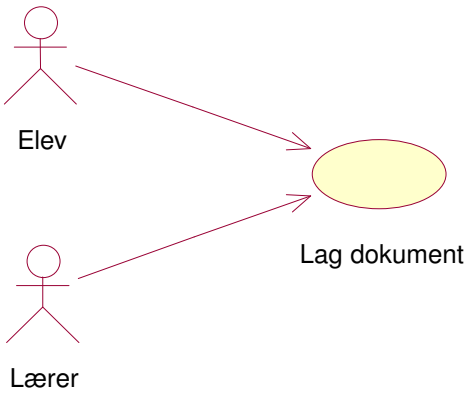
Hendelsesforløp:
- Brukeren går til søkeområdet
- Ønsket søkeord skrives inn
- En liste over treff kommer opp
- Brukeren velger ønsket oppslag i listen
- Det blir vist hvor brukeren kan finne det han/hun søker, eller det er en link til en side på internett

Sluttbetingelse:
Brukeren har fått tak i den litteraturen han/hun ønsket.

Utestående spørsmål:
- Skal alle ha tilgang til all pensumlitteratur for alle fag?

4.1.12 Lag dokument

Use case id: U12	Opprettet: 2005.04.05	Sist endret: 2004.04.05
Beskrivelse: En bruker vil lage et dokument.		



Forhåndsbetingelse:
Programmet er startet og klar til bruk.

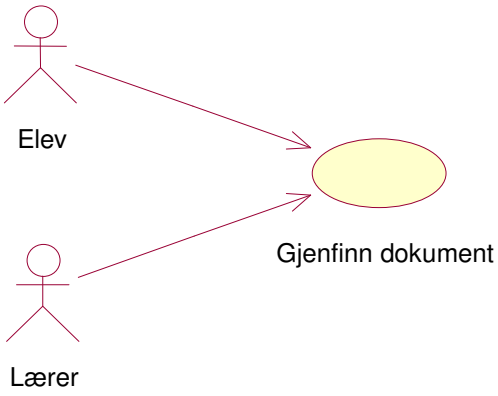
Hendelsesforløp:
- Brukeren åpner dokumenteditoren
- Velger å opprette et nytt dokument
- Et nytt dokument vises
- Brukeren skriver en tekst i dokumentet
- Arbeidet lagres etter kategori eller mappe
- Hvis brukeren ikke er pålogget får han melding om at det kun blir lagret lokalt

Sluttbetingelse:
Dokumentet ble opprettet.

Utestående spørsmål:
-

4.1.13 Gjenfinn dokument

Use case id: U13	Opprettet: 2005.04.05	Sist endret: 2004.04.05
Beskrivelse: En bruker vil finne et dokument, for eksempel et notat eller en øving, som brukere har lagret i sin egen arbeidsbok.		



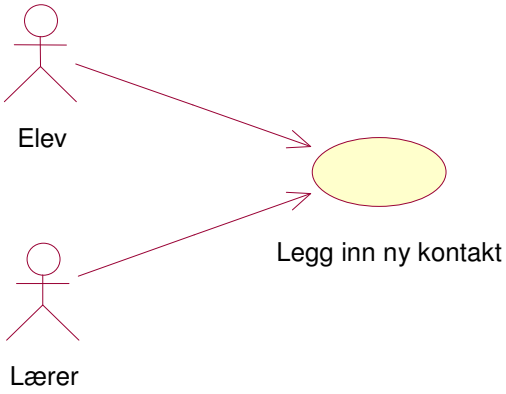
Forhåndsbetingelse:
Programmet er startet og klar til bruk.

Hendelsesforløp:
- Brukeren går til skrivebordet
- Åpner et søkevindu
- Brukeren skriver inn en søketekst og/eller velger en kategori han vil lete etter
- Resultatet på søket kommer opp

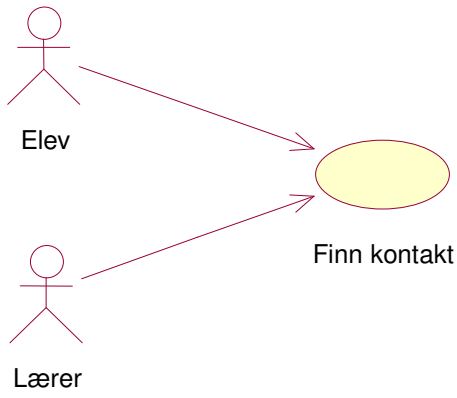
Sluttbetingelse:
Dokumentet ble funnet.

Utestående spørsmål:
-

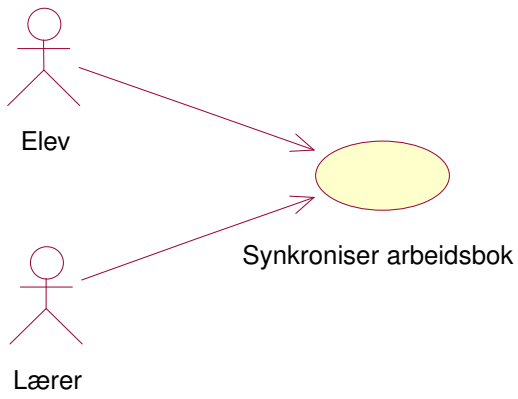
4.1.14 Legg inn ny kontakt

Use case id: U14	Opprettet: 2005.04.05	Sist endret: 2004.04.05
Beskrivelse: En bruker vil legge inn en ny kontakt i adresseboken sin.		
	Forhåndsbetingelse: Programmet er startet og klar til bruk. Hendelsesforløp: • Brukeren åpner adresseboken. • Brukeren trykker på ”ny kontakt”-knappen. • Brukeren får opp et vindu der han kan skrive inn informasjon om kontakten • Brukeren velger å lagre den nye kontakten. Sluttbetingelse: Kontakten ble lagret.	
Utestående spørsmål: •		

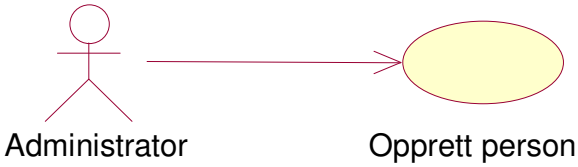
4.1.15 Finn kontakt

Use case id: U15	Opprettet: 2005.04.05	Sist endret: 2004.04.05
Beskrivelse: Brukeren ønsker å finne en av kontaktene i arbeidsboken.		
	Forhåndsbetingelse: Programmet er startet og klar til bruk. Hendelsesforløp: • Brukeren åpner adresseboken. • Brukeren kan bla gjennom listen eller søke i et eget søkefelt for å finne kontakten. • Brukeren finner kontakten han leter etter. Sluttbetingelse: Brukeren fant kontakten han var ute etter.	
Utestående spørsmål: •		

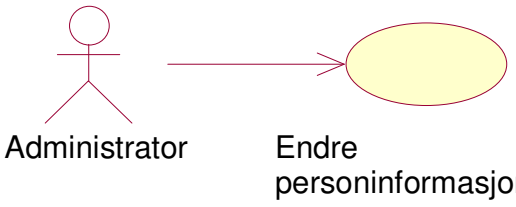
4.1.16 Synkroniser arbeidsbok

Use case id: U16	Opprettet: 2005.04.05	Sist endret: 2004.04.05
Beskrivelse: Arbeidsboken er lagret på en server, men kan også lagres på en lokal maskin. For å være sikker på at det som er lagret på begge steder skal være oppdatert, kan man synkronisere dataene.		
	Forhåndsbetingelse: Programmet er startet og klar til bruk. Brukeren er logget inn. Hendelsesforløp: • Brukeren trykker på synkroniseringsknappen. • Brukeren får opp et vindu med synkroniseringsalternativer. • Brukeren kan velge om han vil laste ned hele arbeidsboken til den lokale maskinen, laste opp til server, synkronisere automatisk eller ikke lagre noe lokalt • Arbeidsboken synkroniseres. Sluttbetingelse: Arbeidsboken er synkronisert.	
Utestående spørsmål: •		

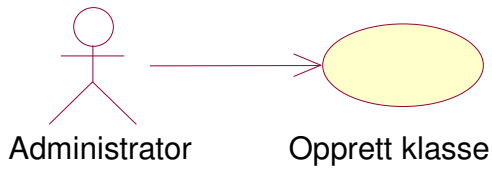
4.1.17 Opprett person

Use case id: U17	Opprettet: 2005.04.05	Sist endret: 2004.04.05
Beskrivelse: En administrator kan opprette en ny person i systemet.		
	Forhåndsbetingelse: Administratoren er logget inn i systemet. Hendelsesforløp: • Administratoren åpner administrasjonsverktøyet • Administratoren velger å opprette en ny person • Administratoren kan legge inn personinformasjon • Personen lagres Sluttbetingelse: Personen har blitt lagret	
Utestående spørsmål: •		

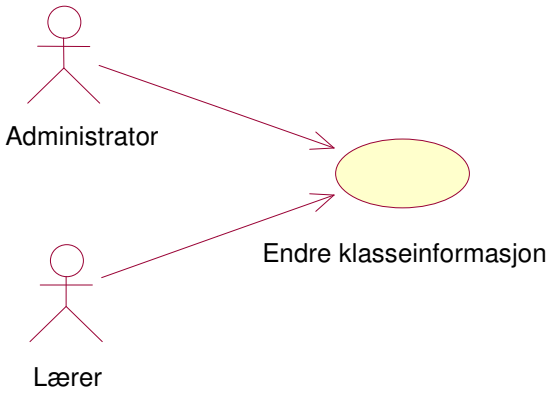
4.1.18 Endre personinformasjon

Use case id: U18	Opprettet: 2005.04.05	Sist endret: 2004.04.05
Beskrivelse: En administrator vil endre informasjonen som har blitt lagret om en person.		
 	Forhåndsbetingelse: Administratoren er logget inn i systemet. Hendelsesforløp: • Administratoren åpner administrasjonsverktøyet • Administratoren velger en person fra en liste • Personinformasjon vises • Administratoren endrer informasjonen om personen • Personen lagres Sluttbetingelse: Personen har blitt lagret	
Utestående spørsmål: •		

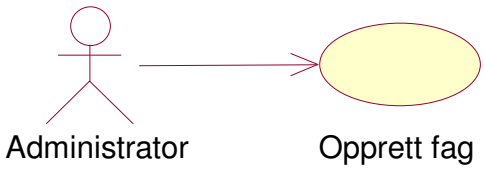
4.1.19 Opprett klasse

Use case id: U19	Opprettet: 2005.04.05	Sist endret: 2004.04.05
Beskrivelse: En administrator kan opprette en ny klasse.		
 	Forhåndsbetingelse: Administratoren er logget inn i systemet. Hendelsesforløp: • Administratoren åpner administrasjonsverktøyet • Administratoren velger å opprette en ny klasse • Administratoren kan legge inn klasseinformasjon • Klassen lagres Sluttbetingelse: Klassen har blitt lagret	
Utestående spørsmål: •		

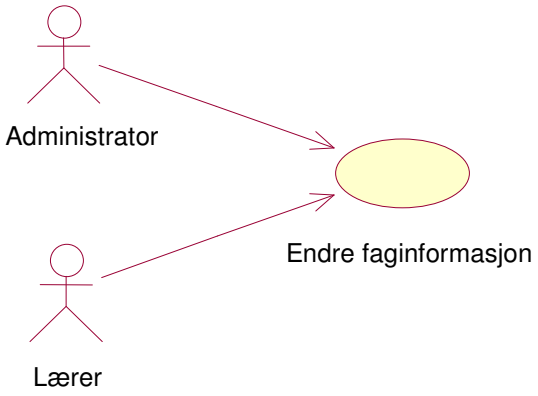
4.1.20 Endre klasseinformasjon

Use case id: U20	Opprettet: 2005.04.05	Sist endret: 2004.04.05
Beskrivelse: En administrator eller faglærer vil endre informasjonen som har blitt lagret om en klasse. Det skal blant annet kunne endres hvilke personer som er med i klassen og hvilke fag klassen har.		
	Forhåndsbetingelse: Programmet er startet og klar til bruk. Brukeren er logget inn og har tilgang til å endre klasseinformasjon. Hendelsesforløp: • Brukeren åpner administrasjonsverktøyet • Brukeren velger en klasse fra en liste • Brukeren endrer informasjonen om klassen • Brukeren lagrer klassen Sluttbetingelse: Klassen har blitt lagret	
Utestående spørsmål: •		

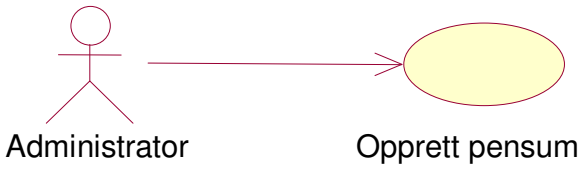
4.1.21 Opprett fag

Use case id: U21	Opprettet: 2005.04.05	Sist endret: 2004.04.05
Beskrivelse: En administrator kan opprette et nytt fag.		
	Forhåndsbetingelse: Administratoren er logget inn i systemet. Hendelsesforløp: • Administratoren åpner administrasjonsverktøyet • Administratoren velger å opprette et nytt fag. • Administratoren kan legge inn faginformasjon • Administratoren lagrer faget Sluttbetingelse: Faget har blitt lagret	
Utestående spørsmål: •		

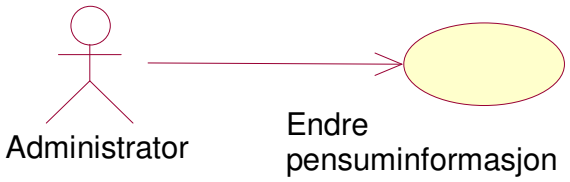
4.1.22 Endre faginformasjon

Use case id: U22	Opprettet: 2005.04.05	Sist endret: 2004.04.05
Beskrivelse: En administrator eller faglærer kan legge til og forandre informasjonen som står på fagets hjemmeside og fagets pensum og forelesninger. Det er også mulig å endre hvilke personer som er med i faget, hva som skal være pensum og hvilke øvinger faget skal ha.		
	Forhåndsbetingelse: Brukeren er logget inn og har tilgang til å endre faginformasjon. Hendelsesforløp: • Brukeren åpner administrasjonsverktøyet • Brukeren velger et fag fra en liste • Brukeren endrer informasjonen om faget • Brukeren lagrer faget Sluttbetingelse: Faget har blitt lagret	
Utestående spørsmål: •		

4.1.23 Opprett pensum

Use case id: U23	Opprettet: 2005.04.05	Sist endret: 2004.04.05
Beskrivelse: En administrator kan opprette en ny pensummodul.		
	Forhåndsbetingelse: Administratoren er logget inn i systemet. Hendelsesforløp: • Administratoren åpner administrasjonsverktøyet • Administratoren velger å opprette en ny pensummodul • Administratoren kan legge inn pensuminformasjon • Administratoren lagrer pensummodulen Sluttbetingelse: Pensummodulen har blitt lagret	
Utestående spørsmål: •		

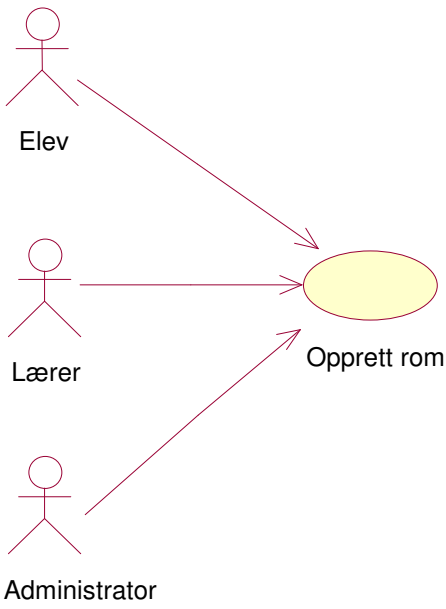
4.1.24 Endre pensuminformasjon

Use case id: U24	Opprettet: 2005.04.05	Sist endret: 2004.04.05
Beskrivelse: En administrator vil endre informasjonen om en pensummodul.		
 	Forhåndsbetingelse: Administratoren er logget inn i systemet. Hendelsesforløp: • Administratoren åpner administrasjonsverktøyet • Administratoren velger en pensummodul i en liste • Administratoren endrer pensummodulen • Pensummodulen lagres Sluttbetingelse: Pensummodulen har blitt lagret	
Utestående spørsmål: •		

4.1.25 Opprett rom

Use case id: U25	Opprettet: 2004.09.08	Sist endret: 2005.04.05

Beskrivelse:
En bruker kan sette opp virtuelle rom.



Forhåndsbetingelse:
Brukeren er logget inn og har tilgang til å opprette nye rom

I rom
Hendelsesforløp:
- Brukeren velger å opprette et nytt rom
- Brukeren kan legge inn rominformasjon
- Rommet opprettes og åpnes

Sluttbetingelse:
Eleven/læreren har kommet inn i det nye rommet.

I administrasjonsverktøyet
Hendelsesforløp:
- Brukeren åpner administrasjonsverktøyet
- Brukeren velger å opprette et nytt rom
- Brukeren kan legge inn rominformasjon
- Brukeren lagrer rommet

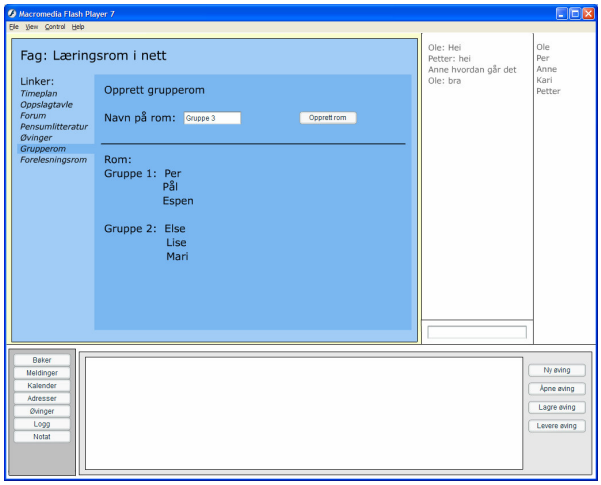
Sluttbetingelse:
Rommet har blitt lagret

Utestående spørsmål:
- Skal alle kunne lage sine egne rom?
- Hvem kontrollerer fellestavlen?

4.1.25.1 Use case story board for opprettelse av et grupperom

- Velger å sette opp et nytt rom

Brukeren velger å sette rommodusen til grupperom og skriver inn et navn på rommet. Det skal også være mulig å velge om rommet skal være låst til bestemte brukere, og om rommet skal være permanent eller midlertidig. Rommet kan også knyttes opp mot et fag.

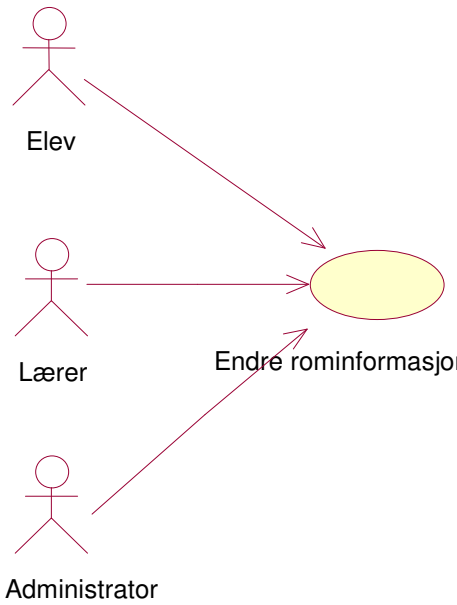


- Rommet opprettes og brukeren kommer inn i rommet. Brukeren kan endre innstillinger på rommet.

4.1.26 Endre rominformasjon

Use case id: U26	Opprettet: 2004.09.08	Sist endret: 2005.04.05

Beskrivelse:
En bruker vil endre informasjonen som har blitt lagret om et rom. Det kan være informasjon om hvem som har tilgang til rommet, om rommet skal være permanent eller ikke, eller om rommet skal være tilknyttet et fag eller ikke.



Forhåndsbetingelse:
Brukeren er logget inn og har tilgang til å opprette nye rom

I rom
Hendelsesforløp:
- Brukeren velger å endre rominformasjon
- Brukeren endrer informasjonen om rommet
- Brukeren lagrer rommet

I administrasjonsverktøy
Hendelsesforløp:
- Brukeren åpner administrasjonsverktøyet
- Brukeren velger et rom fra en liste
- Brukeren endrer informasjonen om rommet
- Brukeren lagrer rommet

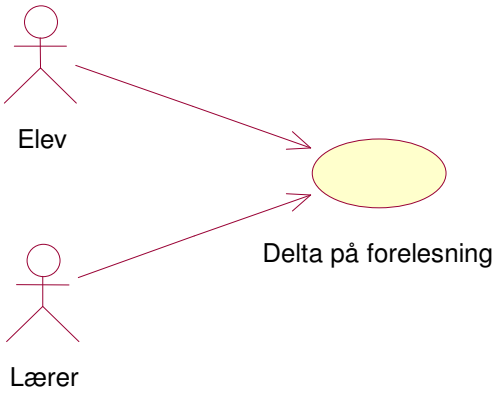
Sluttbetingelse:
Rommet har blitt lagret

Utestående spørsmål:
-

4.1.27 Delta på forelesning

Use case id: U27	Opprettet: 2004.09.08	Sist endret: 2004.09.28

Beskrivelse:
En elev (eller lærer) følger med på det som blir forelest i plenum over nettet.



Forhåndsbetingelse:
Brukeren har tilgang til gjeldende forelesning, og er logget på.

Hendelsesforløp:
- Brukeren velger å følge en forelesning
- Brukeren kommer til et virtuelt rom, der forelesningen skal finne sted
- Foreleseren starter forelesningen.
- Brukeren kan be om ordet underveis. Når han får ordet kan han skrive/snakke inn til systemet, slik at læreren og de andre tilhørerne ser/hører det.

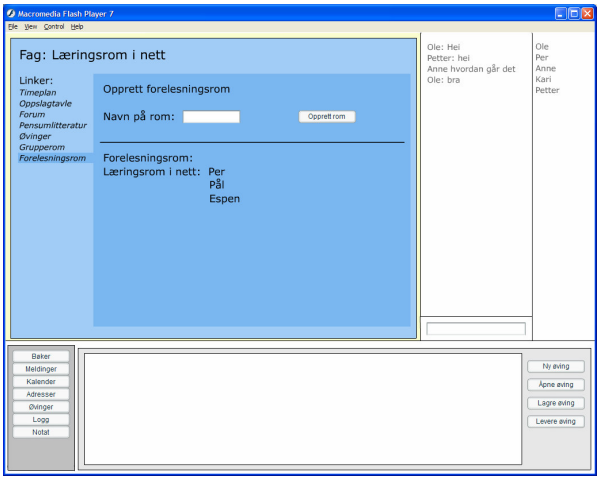
Sluttbetingelse:
Forelesningen er ferdig.

Utestående spørsmål:
- Hvordan skal man annonsere tid og sted for forelesning?
- Hva når man kommer for sent?
- Kan man se opptak av forelesningen, og i tilfelle hvilke muligheter har man da?
- Skal det være mulig å gå på en forelesning som ikke er knyttet opp mot et fag?

4.1.27.1 Use case story board

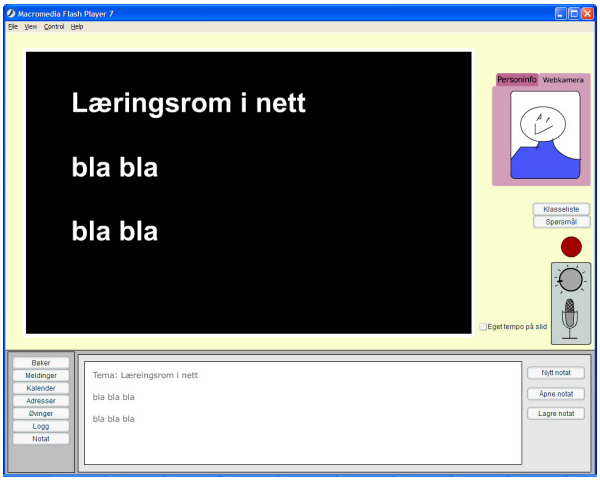
- Eleven velger å følge en forelesning

Eleven er inne på fagsiden og følger linken til forelesningsrom. Der ser han alle forelesningene som foregår i det gjeldende faget og hvem som allerede deltar på forelesningen. Ved å trykke på romnavnet kommer han til forelesningsrommet.



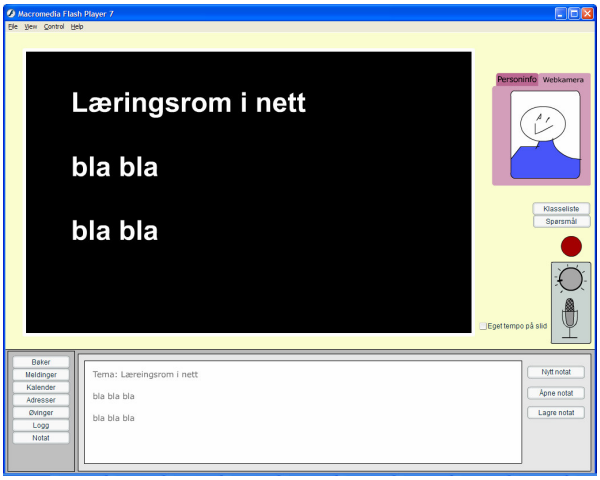
- Eleven kommer til et virtuelt rom, der forelesningen skal finne sted

  Informasjon om foreleseren vil vises i et panel til høyre. Hvis webkamera er tilgjengelig, vil publikum kunne se video av foreleseren. Hvis ikke vil det være mulig å se et bilde av foreleseren.



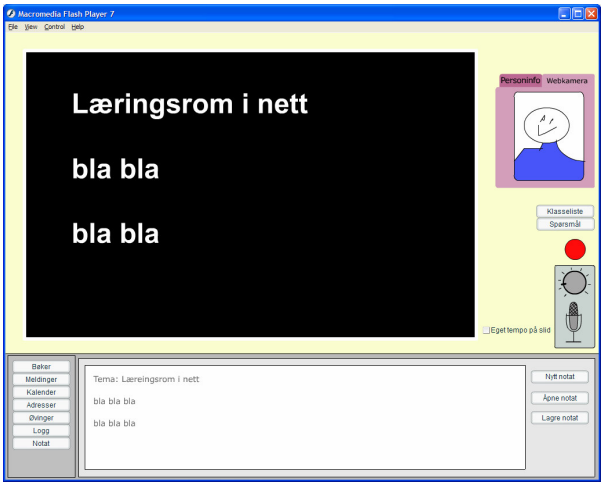
- Foreleseren starter forelesningen.

  Publikum kan velge å følge foreleseren sitt tempo på lysarkene eller de kan selv velge tempo, og bla selv.



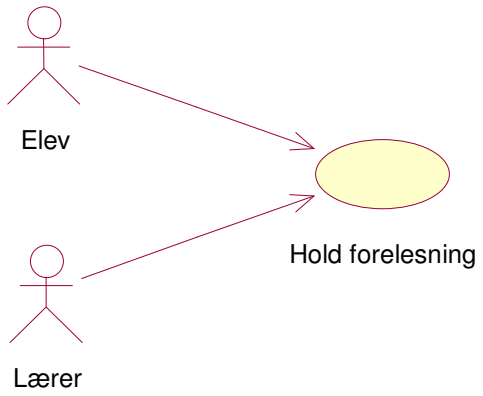
- Elever kan be om ordet underveis. Når eleven får ordet kan han skrive/snakke inn til systemet, slik at læreren og de andre elevene ser/hører det.

  Hvis noen har spørsmål til foreleseren kan de trykke på en spørsmålsknapp og vente til de kan få ordet. Læreren får beskjed og kan velge å gi ordet til disse, eller læreren kan se spørsmålet hvis eleven skriver inn et spørsmål. Det bør være en måte å lagre spørsmål og svar slik at andre som ser opptak av forelesningen også får se stilte spørsmål.



4.1.28 Hold forelesning

Use case id: U28	Opprettet: 2004.09.08	Sist endret: 2004.01.20
Beskrivelse: Læreren (evt. elever) kan holde forelesninger i plenum over nettet.		



Forhåndsbetingelse:
Brukeren har tilgang til å holde forelesninger i det aktuelle faget, og er logget på.

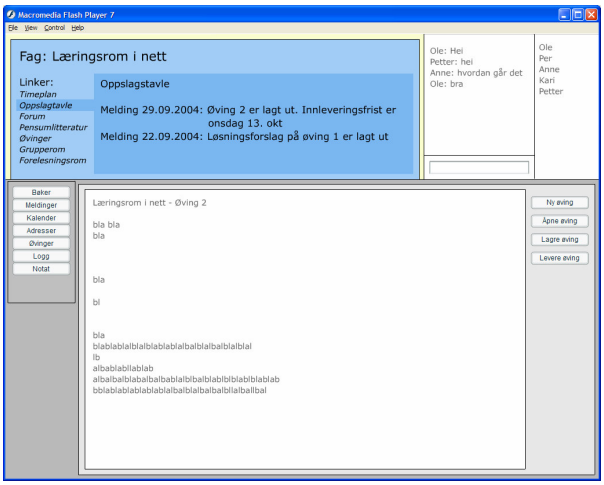
Hendelsesforløp:
- En bruker velger å holde forelesning
- Foreleseren åpner rommet der han har blitt foreleser, og andre kan komme og høre på
- Foreleseren starter forelesningen
- Foreleseren styrer hva som blir vist på den felles tavlen

Sluttbetingelse:
Brukeren har fått holdt forelesningen.

Utestående spørsmål:
- Hvem bestemmer hvem som kan holde forelesning?
- Skal det være mulig å lagre forelesningen?
- Hvordan skal spørsmål bli lagret sammen med forelesningen?
- Hvordan skal man starte en forelesning som ikke hører til et fag? For eksempel gjesteforelesere, seminarer, bedriftspresentasjoner.

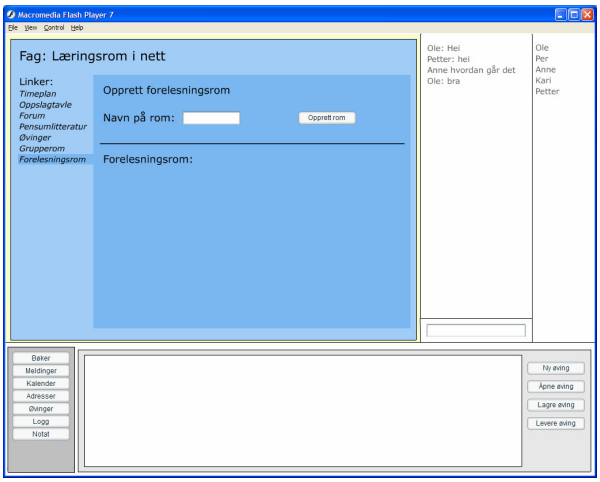
4.1.28.1 Use case story board

- Aktøren har startet opp programmet. Fagsiden er øverst i bildet.



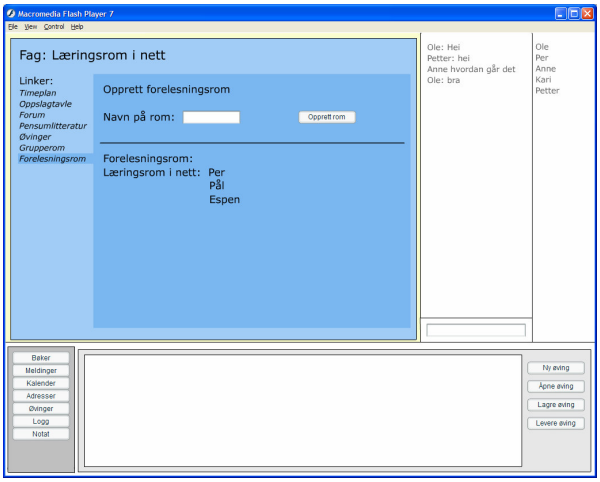
- En bruker velger å holde forelesning

  Foreleseren skriver inn navn og oppretter et forelesningsrom. Navnet kan være fagets navn eller dagens tema.



- Brukeren åpner rommet der han har blitt foreleser, og andre kan komme og høre på.

  Disse rommene er synlige fra forelesningslinken på fagsiden, der det står navnet på rommet. Det står også hvor mange som er tilstede i rommet.

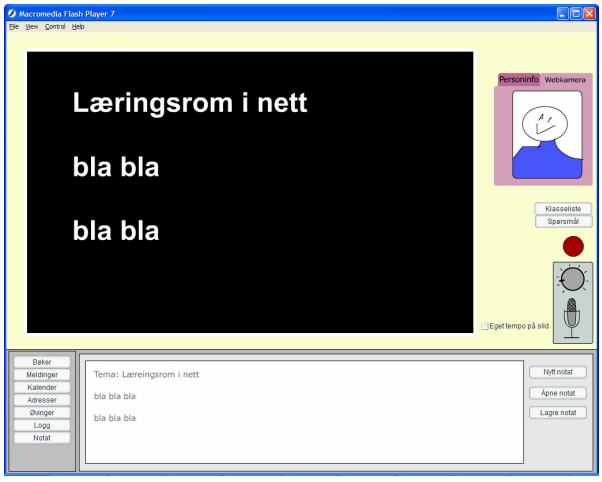


- Foreleser starter forelesningen

  Foreleseren kan se hvem som er tilstede i rommet og starter når det passer.

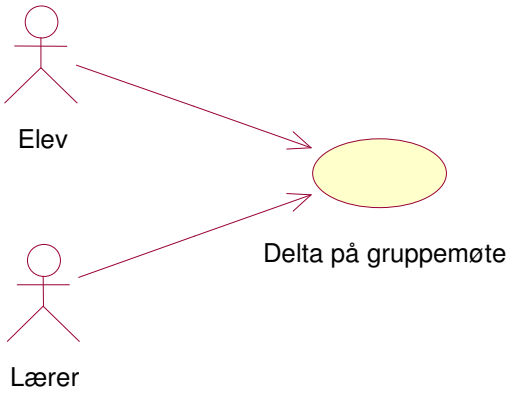
- Foreleser styrer hva som blir vist på fellestavlen.

  Hvis noen har spørsmål får foreleseren vite dette, og kan la disse få ordet, eller spørsmålsstilleren skriver inn et spørsmål.



Hvis forelesningen skal tas opp på video er det mest hensiktsmessig å la spørsmålsstilleren få prate i stedet for å skrive.

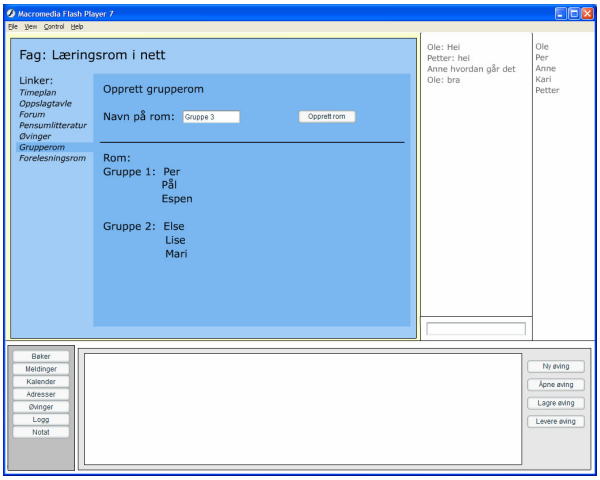
4.1.29 Delta på gruppemøte

Use case id: U29	Opprettet: 2004.09.08	Sist endret: 2004.09.28
Beskrivelse: Eleven eller læreren deltar på et gruppemøte i et virtuelt grupperom.		
	Forhåndsbetingelse: Eleven/læreren har tilgang til systemet og gjeldende rom, og er logget på. Hendelsesforløp: • Brukeren velger et rom • Rommet der møtet skal finne sted åpnes. • Man kan kommunisere via lyd, video eller tekst, eller kombinasjoner av dem, avhengig av rommodusen. Sluttbetingelse: Gruppemøtet er ferdig	
Utestående spørsmål: • Hvordan skal man annonsere tid og sted for møtet? • Hva når man kommer for sent? • Hvem skal kontrollere gruppebordet? • Kan man lagre/se referat fra møtet?		

4.1.29.1 *Use case story board for et møte i et grupperom*

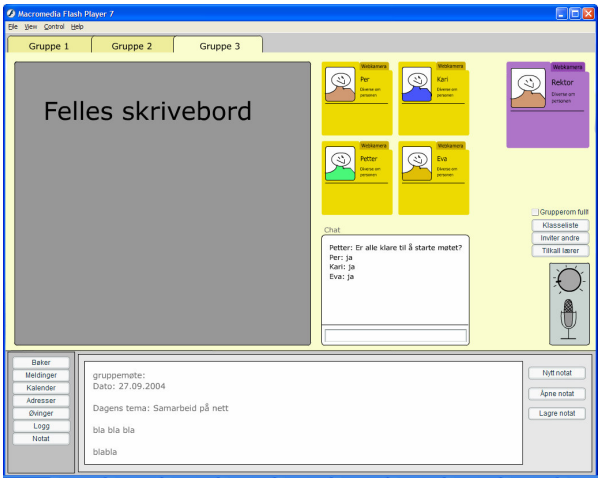
- Brukeren velger et grupperom

Rommene som er opprettet ligger i en liste på fagsiden. Brukeren vil kun se de åpne rommene og de private som han har tilgang til. Ved å trykke på romnavnet kommer han inn i rommet.



- Det virtuelle rommet, der møtet skal finne sted åpnes.

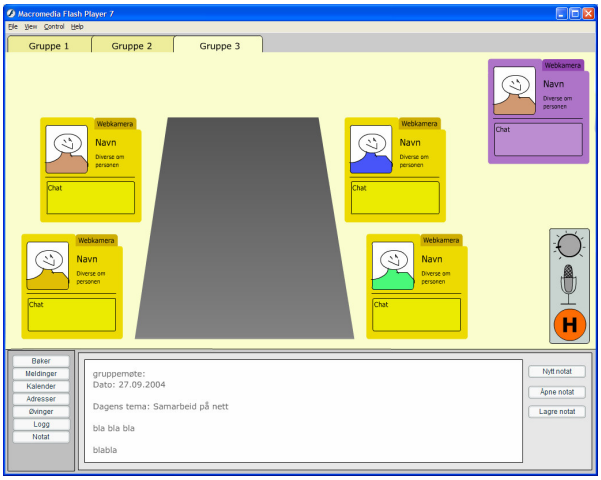
Der kan han/hun se de andre som deltar på møtet. På bordet kan man tegne, skrive, og vise forskjellige data. Alle på møtet ser hele tida det samme. Hvis gruppen trenger hjelp kan de kalle på en lærer. Læreren kan da gå inn på grupperommet og observere og hjelpe gruppen. Læreren kan også komme inn på grupperommet selv om gruppen ikke har bedt om det.



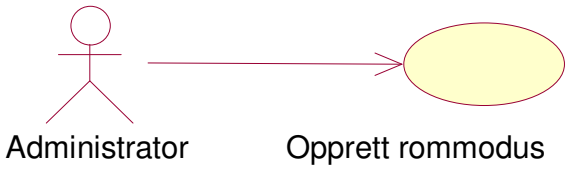
- Man kan kommunisere via lyd, video eller tekst, eller kombinasjoner av dem, avhengig av rommodusen.

Hvis alle har webkamera og mikrofon kan man bruke dette. Det går også an å bruke enten webkamera eller mikrofon. En rød lampe lyser for å vise at din mikrofon er slått på.

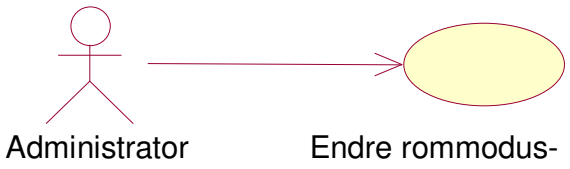
En alternativ grupperomlayout der gruppens medlemmer sitter rundt bordet slik man gjør i den virkelige verdenen.



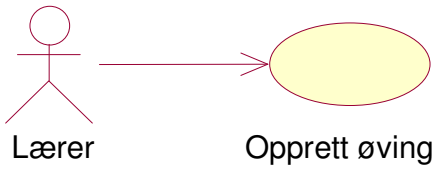
4.1.30 Opprett rommodus

Use case id: U30	Opprettet: 2005.04.05	Sist endret: 2004.04.05
Beskrivelse: En administrator kan opprette en ny rommodus i systemet.		
 	Forhåndsbetingelse: Administratoren er logget inn i systemet. Hendelsesforløp: • Administratoren åpner administrasjonsverktøyet • Administratoren velger å opprette en ny rommodus • Administratoren kan legge inn rommodusinformasjon • Administratoren lagrer rommodus Sluttbetingelse: Rommodus har blitt lagret	
Utestående spørsmål: •		

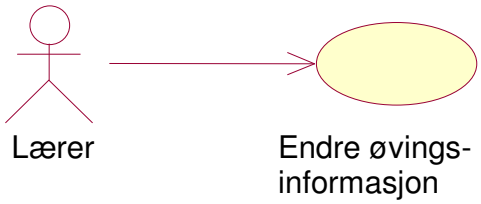
4.1.31 Endre rommodusinformasjon

Use case id: U31	Opprettet: 2005.04.05	Sist endret: 2004.04.05
Beskrivelse: En administrator vil endre informasjonen som har blitt lagret om en rommodus.		
 	Forhåndsbetingelse: Administratoren er logget inn i systemet. Hendelsesforløp: • Administratoren åpner administrasjonsverktøyet • Administratoren velger en rommodus fra en liste • Administratoren endrer informasjonen om rommodus • Administratoren lagrer rommodusen Sluttbetingelse: Rommodusen har blitt lagret	
Utestående spørsmål: •		

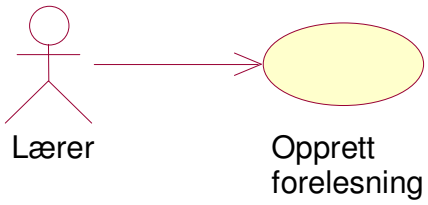
4.1.32 Opprett øving

Use case id: U32	Opprettet: 2005.04.05	Sist endret: 2004.04.05
Beskrivelse: En lærer i et fag kan opprette nye øvinger i et fag		
 	Forhåndsbetingelse: Læreren er logget inn og har tilgang til å endre faginformasjon. Hendelsesforløp: • Læreren åpner administrasjonsverktøyet • Læreren velger et fag fra en liste • Faginformasjonen kommer opp • Læreren velger å legge til øving i faget • Læreren skriver inn øvingsinformasjon • Øvingsinformasjon lagres Sluttbetingelse: Øvingsinformasjonen har blitt lagret	
Utestående spørsmål: •		

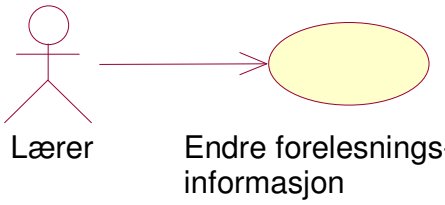
4.1.33 Endre øvingsinformasjon

Use case id: U33	Opprettet: 2005.04.05	Sist endret: 2004.04.05
Beskrivelse: En lærer i et fag kan endre øvingsinformasjonen til en øving		
 	Forhåndsbetingelse: Læreren er logget inn og har tilgang til å endre faginformasjon. Hendelsesforløp: • Læreren åpner administrasjonsverktøyet • Læreren velger et fag fra en liste • Faginformasjonen kommer opp • Læreren velger en øving fra en liste • Læreren endrer øvingsinformasjon • Øvingsinformasjon lagres Sluttbetingelse: Øvingsinformasjonen har blitt lagret	
Utestående spørsmål: •		

4.1.34 Opprett forelesning

Use case id: U34	Opprettet: 2005.04.05	Sist endret: 2004.04.05
Beskrivelse: En lærer i et fag kan opprette nye forelesninger i et fag		
	Forhåndsbetingelse: Læreren er logget inn og har tilgang til å endre faginformasjon. Hendelsesforløp: • Læreren åpner administrasjonsverktøyet • Læreren velger et fag fra en liste • Faginformasjonen kommer opp • Læreren velger å opprette en ny forelesning i faget • Læreren skriver inn forelesningsinformasjon • Forelesningsinformasjon lagres Sluttbetingelse: Forelesningsinformasjonen har blitt lagret	
Utestående spørsmål: •		

4.1.35 Endre forelesningsinformasjon

Use case id: U35	Opprettet: 2005.04.05	Sist endret: 2004.04.05
Beskrivelse: En lærer i et fag kan endre forelesningsinformasjonen for en forelesning		
	Forhåndsbetingelse: Læreren er logget inn og har tilgang til å endre faginformasjon. Hendelsesforløp: • Læreren åpner administrasjonsverktøyet • Læreren velger et fag fra en liste • Faginformasjonen kommer opp • Læreren velger en forelesning fra en liste • Læreren endrer forelesningsinformasjon • Forelesningsinformasjon lagres Sluttbetingelse: Forelesningsinformasjonen har blitt lagret	
Utestående spørsmål: •		

4.2 Aktører

4.2.1 Elev

Aktør	Beskrivelse
 Elev	En person på en utdanningsinstitusjon som er der for å lære. En elev vil kunne arbeide for seg selv eller sammen med andre i grupperom, og kan delta på forelesninger. Eleven har ingen tilgang til administrative funksjoner i systemet.

4.2.2 Lærer

Aktør	Beskrivelse
 Lærer	Person ved utdanningsinstitusjonen som har ansvar for å undervise og veilede elevene. En lærer har flere rettigheter i e-læringssystemet enn en elev. Han kan endre på informasjon om fag og klasser han har ansvar for.

4.2.3 Administrator

Aktør	Beskrivelse
 Administrator	En bruker som har alle administrative rettigheter i systemet.

4.3 Supplementære krav

4.3.1 Prioriteter

Følgende prioriteter er brukt for å beskrive viktigheten til krav:

Pri.	Beskrivelse
A	Høy – Dette er krav som må oppfylles for at systemet skal oppnå de overordnede målene som er satt.
B	Middels – Dette er krav som bør oppfylles, men som ikke er absolutt nødvendige for å nå de overordnede mål.
C	Lav – Dette er krav som det er ønskelig at man oppfyller, men som ikke har noen stor betydning for de overordnede målene.

Tabell 4-1 Kravprioriteter

4.3.2 Plattform og arkitektur

ID	Krav	Pri.	Opphav
K1.01	Alle data skal lagres sentralt på en server.	A	(10.09.04)
K1.02	All kommunikasjon skal gå via programvaren på serveren. Unntak er direkte filoverføring, samt direkte overføring av lyd, bilde og pratetekst, som kan sendes direkte mellom to klienter.	A	(10.09.04)
K1.03	Alle deler av systemet må for brukeren framtre som ett system, slik at alle endringer gjort på et sted, umiddelbart trer i kraft gjennom hele systemet.	A	(10.09.04)
K1.04	Systemet må tilrettelegges for standardløsninger for sikkerhetskopiering. Det må finnes automatiserte rutiner for tilbakeføring av sikkerhetskopier ved strømbrudd, maskinhavari, etc.	A	(10.09.04)
K1.05	Det må være mulig for en bruker å ta personlig kopi av sine lagrede arbeider i systemet, for eksempel for å jobbe uten nett, eller for egen arkivering for eksempel ved endt studium.	A	(10.09.04)
K1.06	Det bør finnes en funksjon for å oppdatere egne arbeider i systemet fra en personlig kopi som brukeren har gjort endringer i.	B	(10.09.04)
K1.07	Det bør finnes funksjoner for å renske opp lagrede data som tilhører personer som har sluttet ved undervisningsstedet eller av annen grunn ikke lenger har tilgang til systemet. Funksjonen bør også innebære overføring av nødvendig informasjon til bortsettingsarkiv.	B	(10.09.04)
K1.08	Systemet bør kunne nås fra alle vanlige nettlesere på en hvilken som helst klientmaskin, inkludert mobiltelefon, med så mange funksjoner operative som mulig. Det bør opplyses for brukeren om hvilke funksjoner som ikke er tilgjengelige med den klientkonfigurasjonen vedkommende kjører.	B	(10.09.04)

Tabell 4-2 Krav til plattform og arkitektur

4.3.3 Brukere, roller og tilgangskontroll

ID	Krav	Pri.	Opphav
K2.01	All tilgang til informasjon og funksjoner styres av, og bare av programvaren på serveren.	A	(10.09.04)
K2.02	Det må være mulig for en bruker å ha flere roller. Det skal kunne defineres en standardrolle som brukeren automatisk får når han/hun logger inn.	A	(10.09.04)
K2.03	En rolle bør kunne defineres globalt for hele skolen, men også kun innenfor et fag, eller forelesning/grupperom.	B	(10.09.04)
K2.04	Tilgangen bør kunne gjøres tidsbegrenset, dvs. med utløpsdato.	B	(10.09.04)
K2.05	Det bør være mulig å (i et tidsrom) gi en person en annen persons rolle, for eksempel for vikarierende lærere.	B	(10.09.04)
K2.06	Det skal ikke vises noe informasjon som en bruker ikke har tilgang til å se. Ved forespørsler (f. eks. ved søk) skal det ikke gis tilslag på informasjon som brukeren ikke har lov å se. All informasjon som en brukeren ikke har tilgang til å se, skal for brukeren fremtre som om den ikke eksisterte.	A	(10.09.04)
K2.07	Det må være mulig å søke etter adresser og annen info blant brukerne.	A	(10.09.04)
K2.08	Hver bruker må kunne begrense den informasjonen som kan gis om vedkommende til andre brukere.	A	(10.09.04)

Tabell 4-3 Krav til brukere, roller og tilgangskontroll

4.3.4 Brukergrensesnitt generelt

ID	Krav	Pri.	Opphav
K3.01	Alle deler av brukergrensesnittet bør være basert på felles prinsipper, slik at de gir et helhetlig inntrykk. Navngiving av funksjoner og lignende bør være konsistent i hele systemet.	B	(10.09.04)
K3.02	Det bør være mulig å endre visningsspråk i hele brukergrensesnittet etter hver enkelt brukers ønske.	B	(10.09.04)
K3.03	Formater for tall og datoer bør følge gjeldende visningsspråk.	B	(10.09.04)
K3.04	Det må finnes hjelpefunksjoner som dekker alle funksjoner tilgjengelige for brukerne.	A	(10.09.04)
K3.05	Det må gis opplysninger om alle feilsituasjoner til brukeren. I situasjoner der det er aktuelt bør man referere til ufyllende informasjon om feilen.	A	(10.09.04)
K3.06	Det må ikke være mulig å foreta valg og registrere informasjon i systemet som strider med de underliggende regler for informasjonsstrukturen.	A	(10.09.04)
K3.07	Brukergrensesnittet bør ut fra brukerens tilknytning og rolle i den aktuelle delen av systemet, tilby rask tilgang til de funksjoner det forventes av brukeren kan ha bruk for.	C	(10.09.04)
K3.08	Det bør være muligheter for å tilpasse systemet til forskjellig alder og erfaringsnivå på brukerne.	B	(10.09.04)

Tabell 4-4 Generelle krav til brukergrensesnittet

4.3.5 Skole og fag

ID	Krav	Pri.	Opphav
K4.01	Både generelt for hele skolen og for hvert enkelt fag skal det finnes oppslagstavler, der generelt publisert informasjon kan slås opp.	A	(10.09.04)
K4.02	Tavlene bør være mulige å dele inn etter tema, formalitet, målgruppe, osv.	B	(10.09.04)
K4.03	I utgangspunktet skal bare lærer/administrasjon ha adgang til å henge opp oppslag, men det bør finnes tavler der alle kan henge opp informasjon.	B	(10.09.04)
K4.04	Man bør kunne foreta litteratursøk og finne bøker via systemet.	B	(10.09.04)

Tabell 4-5 Skole- og fagrelaterte krav

4.3.6 Arbeidsbok

ID	Krav	Pri.	Opphav
K5.01	Hver bruker har en personlig ”skolesekk” som følger brukeren uansett hvor vedkommende er i systemet.	A	(10.09.04)
K5.02	De personlige verktøyene kan innebære kalender/avtalebok, referanser til bøker/informasjon tilknyttet en spesiell bruker, notatbøker, samt område for å lagre arbeider.	C	(10.09.04)
K5.03	Systemet skal tilby funksjonalitet som muliggjør løsing av aktuelle oppgavetyper i systemet, og levering av dem til faglærer.	A	(10.09.04)
K5.04	Det må være mulig for faglærer å kunne rette oppgavene og gi tilbakemelding til eleven. Tilbakemelding kan innebære kopi av oppgaven med påførte kommentarer/rettelser.	A	(10.09.04)
K5.05	Det bør være mulig å etter eget ønske knytte egen kalender/avtalebok til publisert informasjon, slik at publisert info (med dato/klokkeslett) innenfor visse kriterier automatisk kommer opp som avtaler.	B	(10.09.04)
K5.06	Man bør kunne velge å få påminnelser om nært forestående avtaler.	B	(10.09.04)
K5.07	Avtalebok bør i ettertid kunne fungere som en logg, ved at man kan knytte notater og annen info til avtalene.	C	(10.09.04)

Tabell 4-6 Krav til arbeidsbok

4.3.7 Forelesningsrom

ID	Krav	Pri.	Opphav
K6.01	Det skal være mulig å delta på forelesninger via systemet.	A	(10.09.04)
K6.02	En bruker skal, dersom vedkommende har tilgang til det, kunne sette opp en ny virtuell forelesningssal, og invitere andre brukere til å delta.	A	(10.09.04)
K6.03	Foreleser er ordstyrer i forelesningssalen, men andre bør også kunne få ordet. Deltakere på en forelesning bør kunne signalisere til ordstyrer at de vil ha ordet.	B	(10.09.04)
K6.04	Foreleser styrer en tavle som alle på forelesningen kan se. Denne oppdateres kontinuerlig hos alle brukerne.	A	(10.09.04)
K6.05	Forelesningstavla bør kunne vise et hvilket som helst innhold som ellers er mulig å vise på en vanlig datamaskin.	B	(10.09.04)
K6.06	Det bør være mulig å direkte overført lyd og video av den som snakker.	B	(10.09.04)

Tabell 4-7 Krav til forelesningsrom

4.3.8 Grupperom

ID	Krav	Pri.	Opphav
K7.01	Det skal være mulig å holde gruppemøter via systemet, der to eller flere brukere kan sitte og jobbe rundt et felles virtuelt bord.	A	(10.09.04)
K7.02	En bruker skal, dersom vedkommende har tilgang til det, kunne sette opp et nytt virtuelt grupperom, og invitere andre brukere til å delta.	A	(10.09.04)
K7.03	Man bør kunne velge om man vil ha ordstyrer på et gruppemøte eller ikke.	B	(10.09.04)
K7.04	”Bordet” som deltakerne sitter rundt er felles og kan ses av alle. Denne oppdateres kontinuerlig hos alle brukerne.	A	(10.09.04)
K7.05	”Bordet” bør kunne vise et hvilket som helst innhold som ellers er mulig å vise på en vanlig datamaskin.	B	(10.09.04)
K7.06	Det bør være mulig å direkte overført lyd og video av den som snakker.	B	(10.09.04)
K7.07	Deltakere bør kunne midlertidig legge fram egne arbeider på gruppebordet for å vise det til andre.	B	(10.09.04)
K7.08	Den som setter opp grupperommet bør ha mulighet til å angi hvem som har lov til å møte i det aktuelle rommet.	B	(10.09.04)
K7.09	Det bør være mulig å delta på møtet kun som tilskuer. Man bør dog ha mulighet til å kommunisere med gruppa.	B	(10.09.04)

Tabell 4-8 Krav til grupperom

4.3.9 Kommunikasjon

ID	Krav	Pri.	Opphav
K8.01	Man skal kunne sende og motta e-post både fra brukere i systemet og utenforstående.	A	(10.09.04)
K8.02	Man skal kunne kommunisere med andre brukere i systemet i pratekanaler.	A	(11.09.04)

Tabell 4-9 Krav til kommunikasjon

4.3.10 Søking

ID	Krav	Pri.	Opphav
K9.01	Alle felt med begrenset lengde (dvs. ikke binær- og fritekstfelt) bør være søkbare i de sammenhenger der dette er aktuelt.	B	(10.09.04)
K9.02	Store og små bokstaver bør kunne behandles som ekvivalente ved søk.	B	(10.09.04)
K9.03	For datoer og tallverdier bør man kunne søke på et intervall av verdier.	B	(10.09.04)
K9.04	Det bør være mulig å bruke trunkering, samt maskering av enkelttegn ved søk.	B	(10.09.04)
K9.05	Tidkrevende søk bør kunne avbrytes. Det bør også opplyses til brukeren på forhånd dersom et søk antas å kunne være tidkrevende.	B	(10.09.04)
K9.06	Det bør opplyses om antall tilslag ved søk.	B	(10.09.04)
K9.07	Brukeren bør selv kunne velge hva søkeresultatene skal sorteres etter.	B	(10.09.04)
K9.08	Brukeren skal kunne søke etter pensumlitteratur i biblioteket.	B	(10.09.04)
K9.09	Brukeren skal kunne søke etter info på fagsida.	B	(10.09.04)
K9.10	Brukeren skal kunne søke i egen arbeidsbok	B	(10.09.04)

Tabell 4-10 Krav til søking

5 Design og arkitektur

Dette kapittelet viser programmets underliggende design. Dette er en teknisk beskrivelse av programmet. Sammen med brukermanualen viser vi hvordan programmet fungerer.

Vi viser i 5.2 scenarier som har vært sentrale når vi har designet arkitekturen. Arkitekturen i programmet vises ved hjelp av ulike diagrammer. Programstrukturkapittelet viser systemet inndelt i pakker og klasser. I prosesskapittelet vises hvilke tråder og prosesser programmet består av. Datamodellen vises i databasekapittelet. De ulike kapitlene blir representert som UML-diagrammer.

5.1 Arkitekturmessige mål og begrensninger

5.1.1 Hovedmålene for arkitekturen:

- Klienten skal gjøre mest mulig arbeid selv og vil ha minst mulig kommunikasjon mot serveren
- Hvis man gjør en forandring et sted i systemet, vil dette oppdateres videre til resten av systemet.
- Programmet skal være skalerbart med mulighet for å utvide programmet med andre typer klienter (webklient, mobilklient)
- Programmet skal kunne takle et stort antall brukere

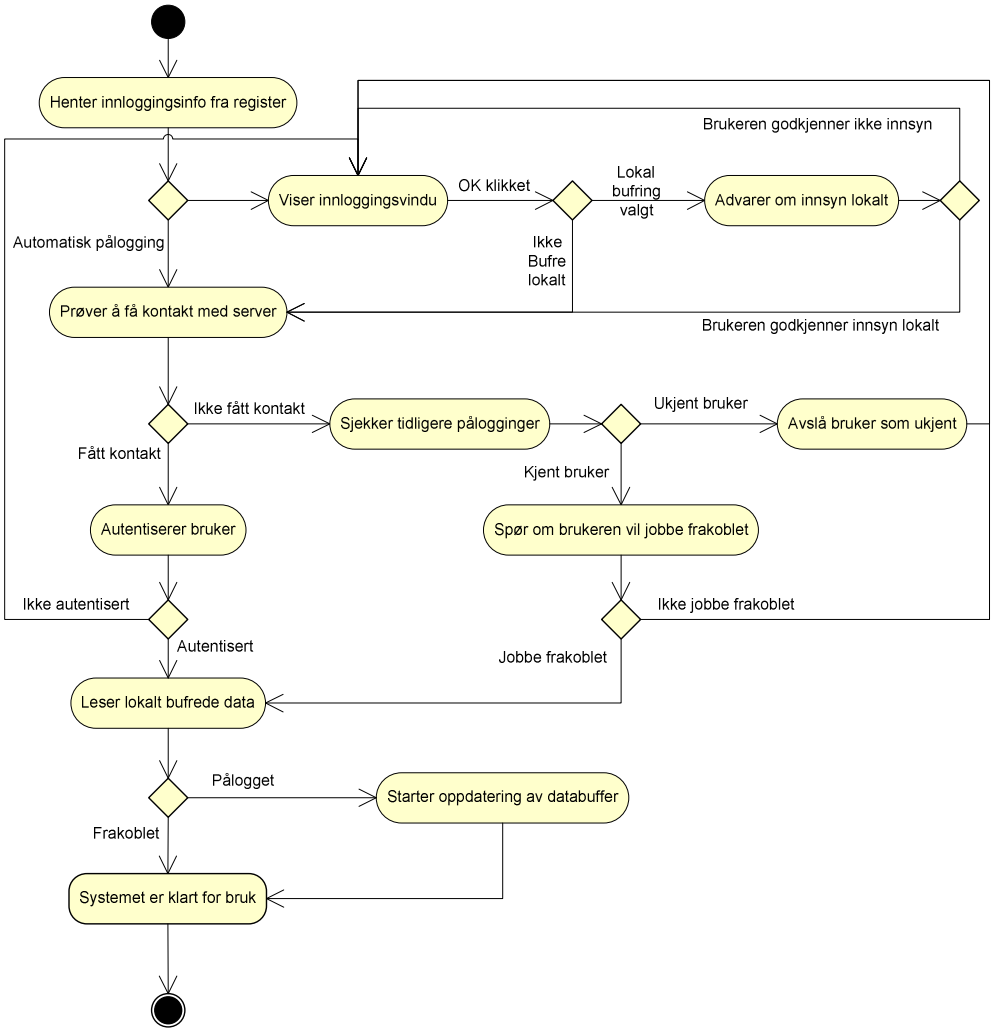
5.1.2 Begrensninger:

- Programmet er plattformavhengig. Både klient og server må kjøres på Windows. Det er mulig å lage klienter for andre plattformer, men serveren kan kun kjøres i Windows-plattformen.

5.2 Sentrale scenarier

5.2.1 Tilkoblingsscenarier

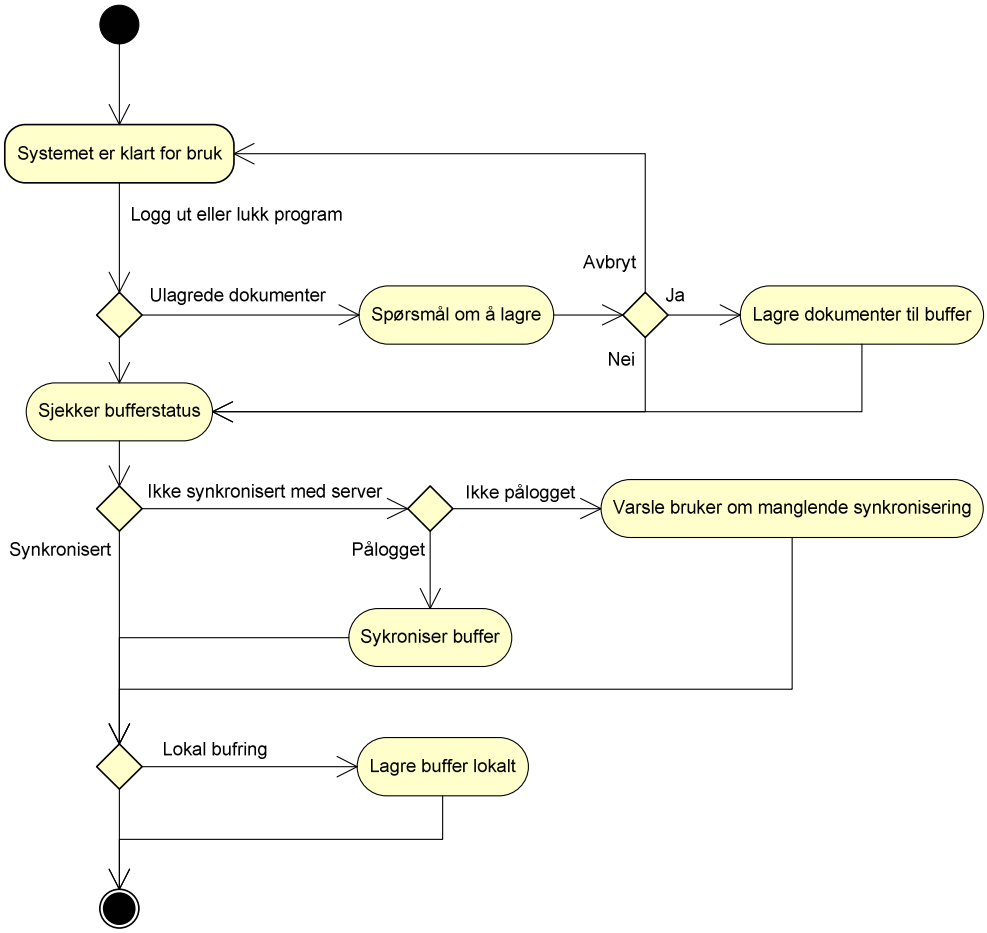
5.2.1.1 Start program



Figur 5-1 Aktivitetsdiagram for Start program

Dette scenariet skjer hver gang man starter programmet. I oppstarten må programmet vite om man vil bufre data lokalt eller ikke, og om maskinen er tilkoblet internett. Hvis man ikke er tilkoblet internett, kan man velge å jobbe frakoblet, men da har programmet begrenset funksjonalitet. I frakoblet modus må brukeren ha valg at programmet skal bufre data lokalt. Klienten vil da hente og lagra alle data lokalt på maskinen. Hvis en bruker har vært logget på systemet med samme datamaskin tidligere kan programmet, huske tidligere innloggingsvalg. Dette gjelder brukerens passord, om man vil logge inn automatisk eller om man vil bufre data lokalt.

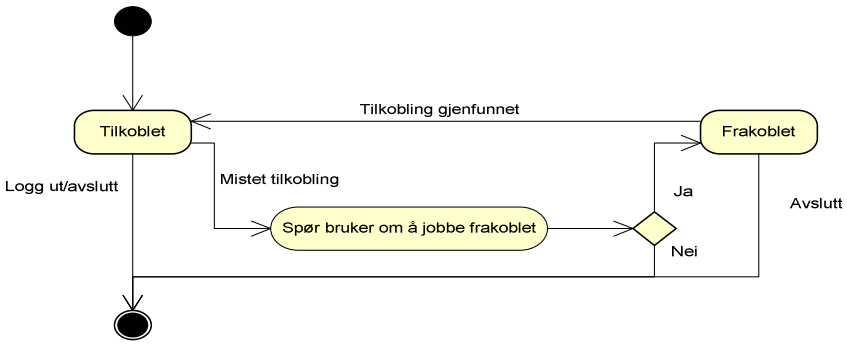
5.2.1.2 Avslutt program



Figur 5-2 Aktivitetsdiagram for Avslutt program

Ved avslutting av programmet får man spørsmål om man vil lagre ulagrede dokumenter. Klienten vil synkronisere seg med serveren hvis den har nettilgang. Tilslutt vil klienten lagre bufferne lokalt, hvis brukeren har tillatt det.

5.2.1.3 Endring i tilkobling

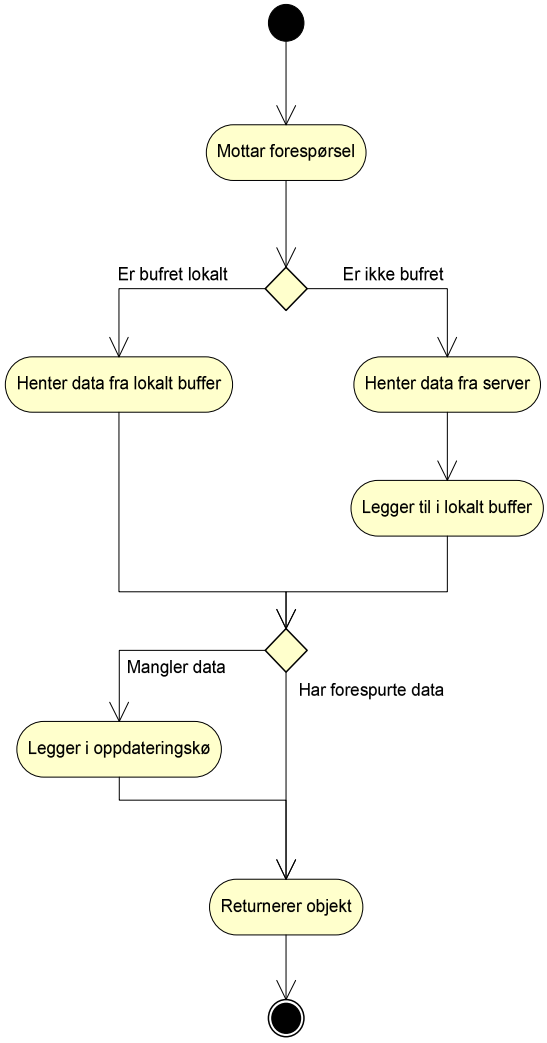


Figur 5-3 Tilstandsdiagram for Endring i tilkobling

Ved endring i tilkobling blir agentene informert om tilkoblingsstatus for å kunne vite om programmet skal bufre data og holde seg oppdatert med databasen.

5.2.2 Databufferscenarier

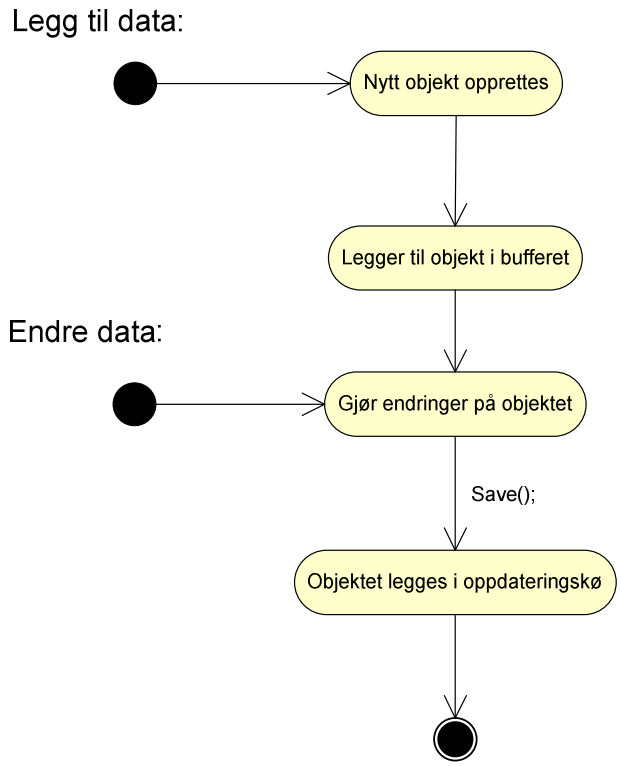
5.2.2.1 Hent data



Figur 5-4 Aktivitetsdiagram for Hent data

Dette scenariet viser hva som skjer når agentene får en forespørsel, for eksempel fra brukergrensesnittet. Data blir bare hentet fra server hvis det ikke ligger noe lagret lokalt.

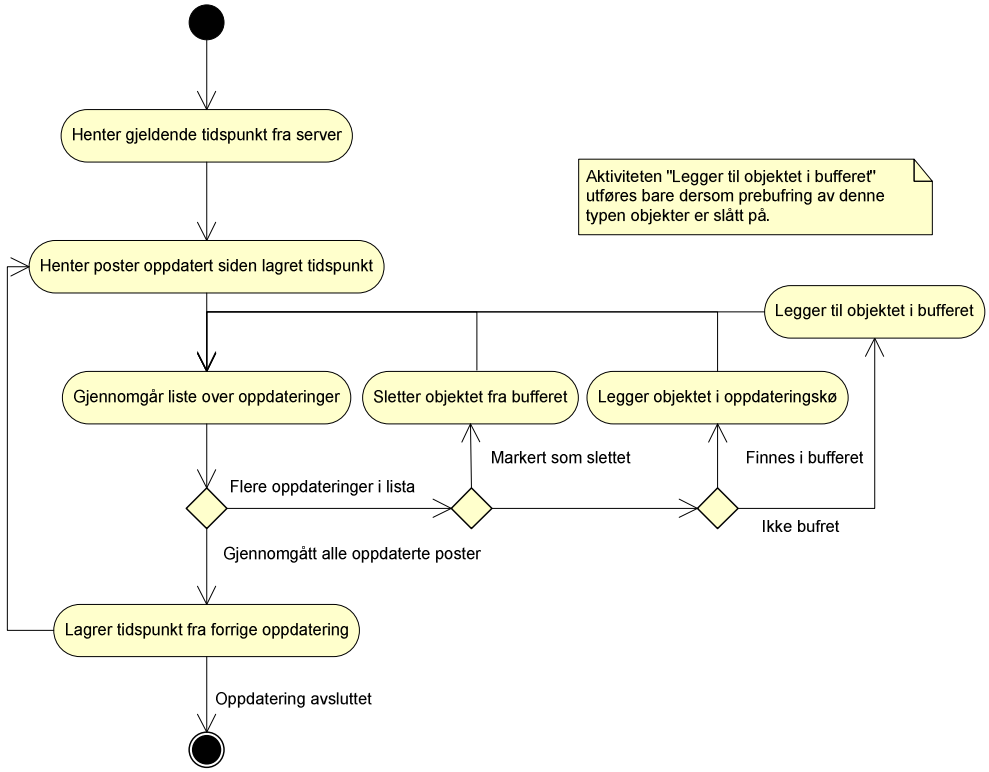
5.2.2.2 Legg til/endre data



Figur 5-5 Aktivitetsdiagram for Legg til/endre data

Her vises hva som skjer når en bruker lagrer noe som skal bufres eller endrer noen som allerede finnes i bufferet. Hvis klienten er tilkoblet internett vil objektet oppdateres i databasen når det blir behandlet fra oppdateringskøen.

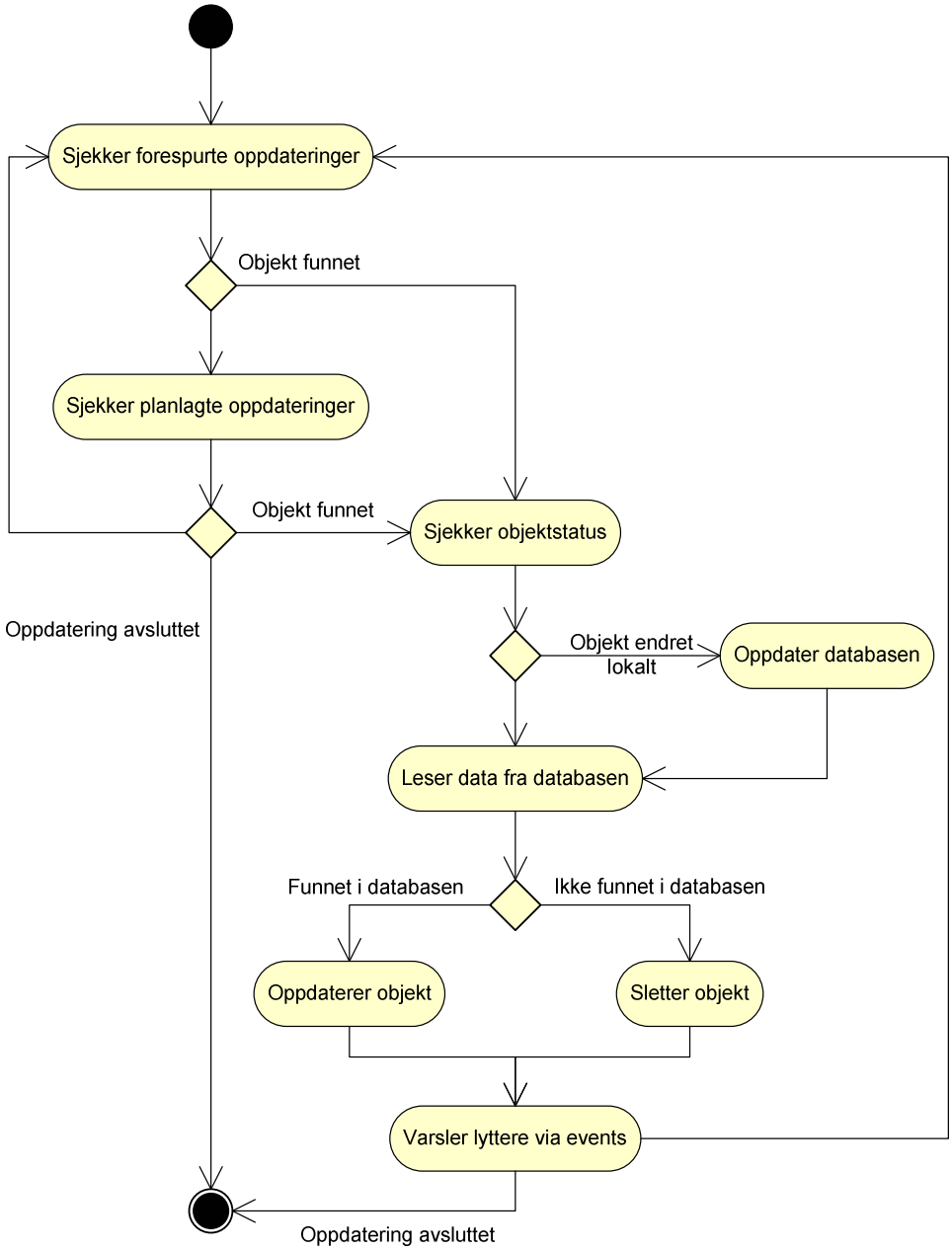
5.2.2.3 Se etter oppdateringer



Figur 5-6 Aktivitetsdiagram for Se etter oppdateringer

I tilkoblet tilstand vil agentene stadig oppdateres mot databasen. Agentene bestemmer selv hvor ofte bufferet blir oppdatert. Bufferobjekter vil bli opprettet, endret eller slettet slik at innholdet i bufferet er samsvar med det som finnes i databasen.

5.2.2.4 Oppdater buffer



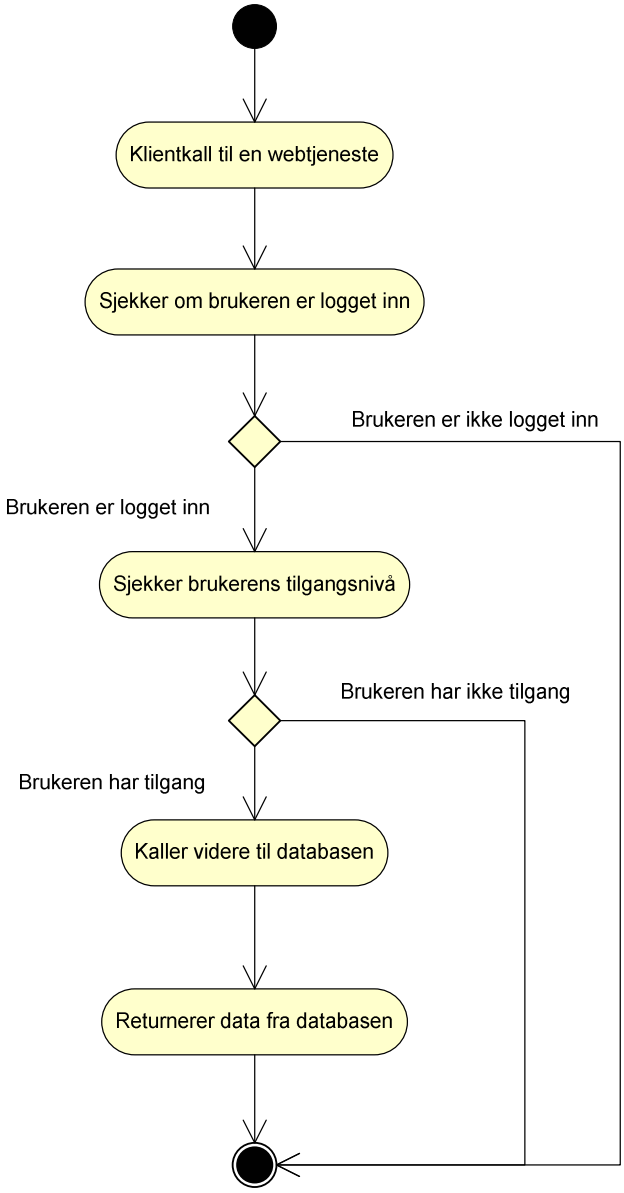
Figur 5-7 Aktivitetsdiagram for Oppdater buffer

I køen for forespurte oppdateringer ligger det en bruker har bedt systemet om. Planlagte oppdateringer er de oppdateringene som hele tiden skjer i bakgrunnen. De forespurte vil alltid bli utført før de planlagte endringene. Først når køen for forespurte oppdateringer er tom vil programmet utføre de planlagte oppdateringene. Et bufferobjekt kan ha status som lagt til, endret eller slettet. Etter at et objekt er oppdatert vil agenten få tilbake fra databasen hva som har blitt endret. På denne måten

sikres det at bufferne er oppdatert med det som finnes i databasen og at det blir registrert dersom oppdateringen ikke har vært vellykket. Systemet blir tilsutt varslet om oppdateringen.

5.2.3 Serverscenarier

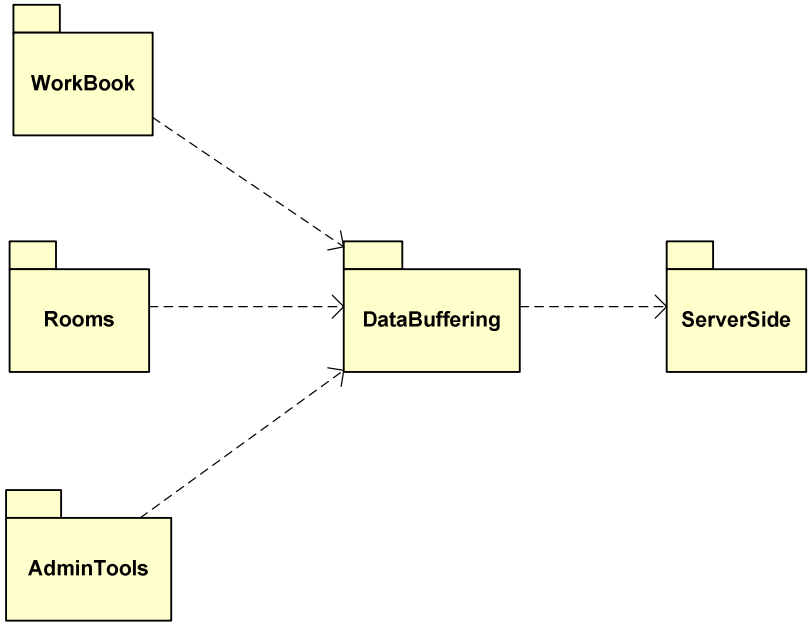
5.2.3.1 Kall til webtjenester



Figur 5-8 Aktivitetsdiagram for Kall til webtjenester

5.3 Programstruktur

5.3.1 Klient

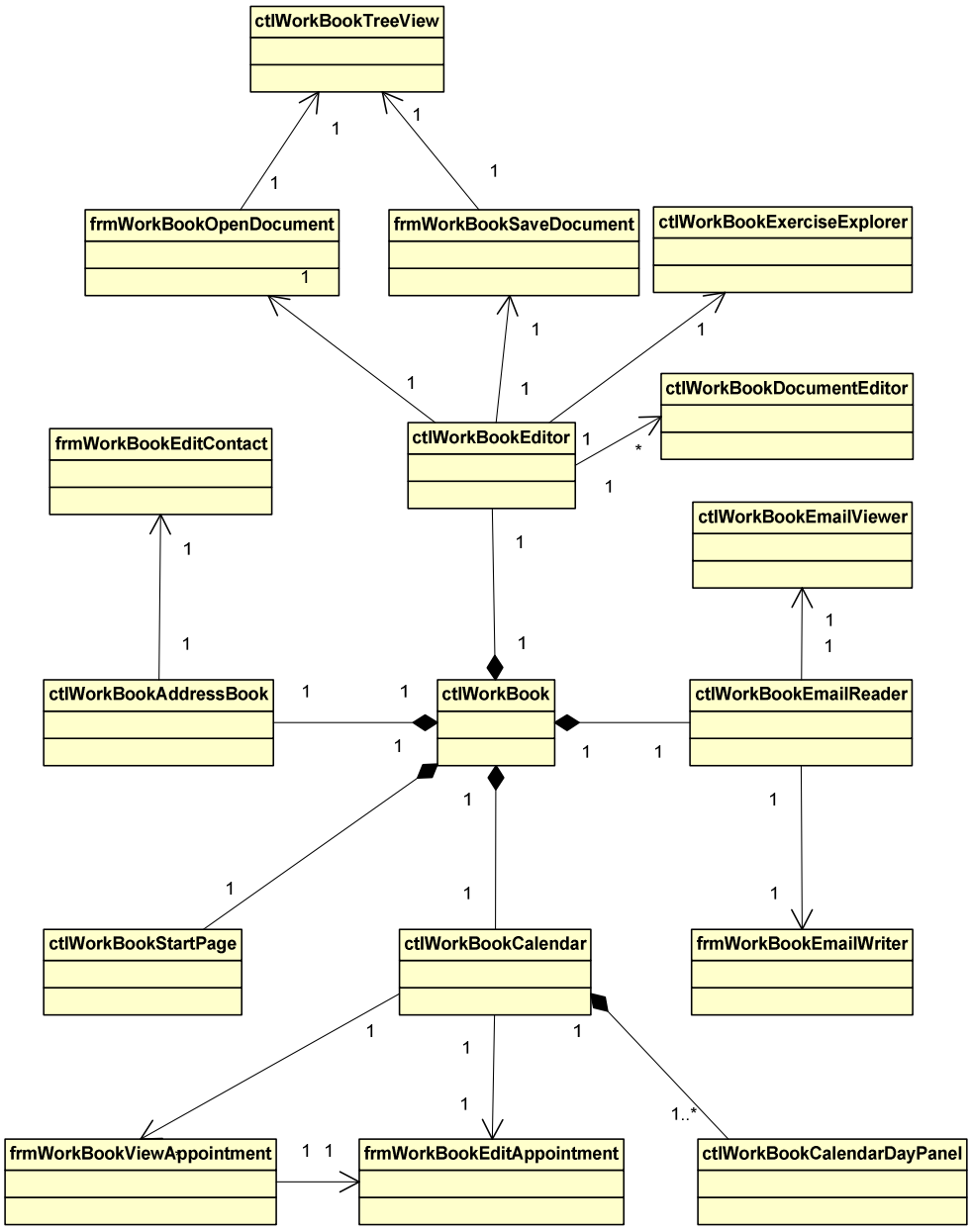


Figur 5-9 Oversikt over pakker på klienten

Pakkene WorkBook, Rooms, AdminTools har klasser for brukergrensesnittet. Databuffering-pakken er en samling med kontrollerklasser. Disse klassene finner ut hvilke data som trengs å bli hentet fra serveren og hva som allerede finnes i bufferet. ServerSide har klasser som sørger for kommunikasjon med serveren.

I brukergrensesnittpakkene er det to forskjellige typer klasser. Det ene er selvstendige vindusklasser, som har forstavelsen ”frm”. Det andre er brukerkontrollklasser, med forstavelsen ”ctl”, som er komponenter i en vindusklasse. I DataBuffering er det noen agentklasser. Disse har forstavelsen ”agt”. ServerSide har kommunikasjonsklasser som har forstaversen ”con”.

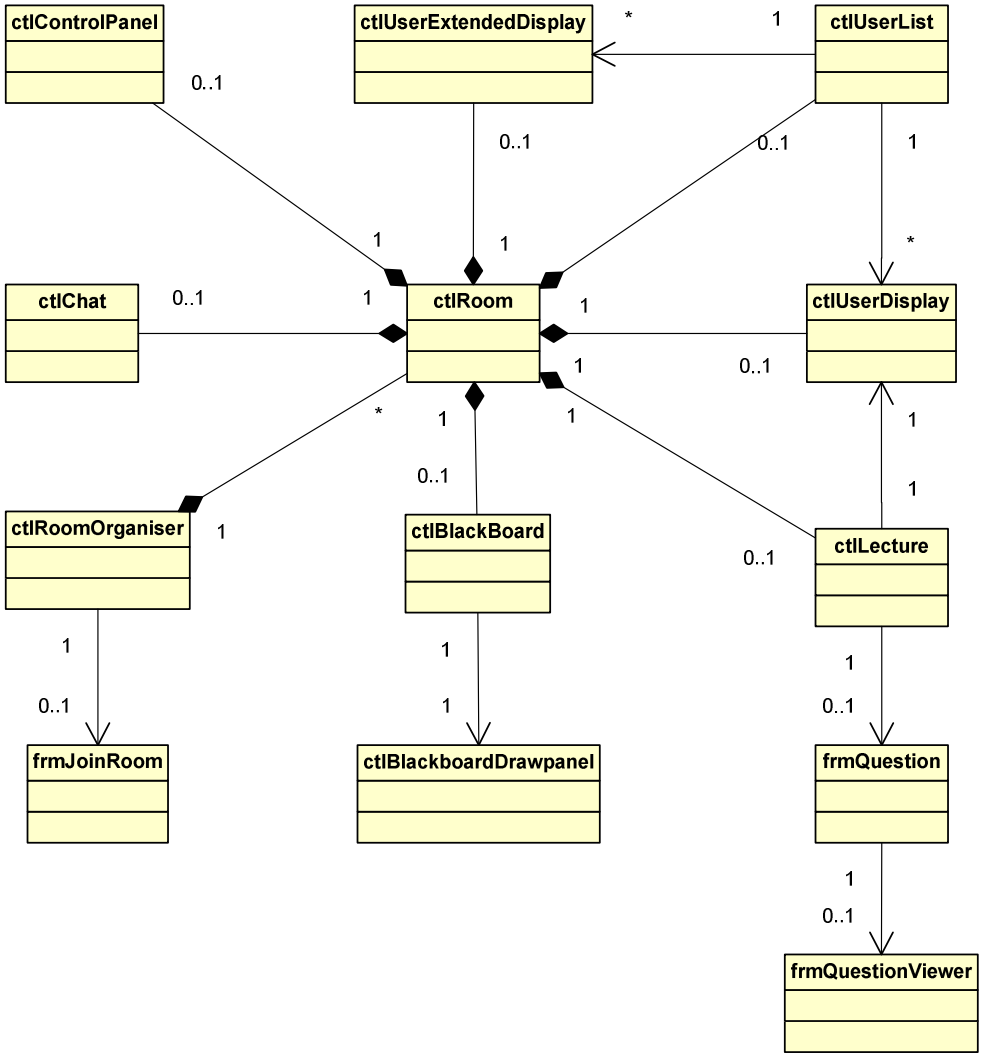
5.3.1.1 WorkBook



Figur 5-10 Klassediagram for WorkBook

Klassenavn	Beskrivelse
ctlWorkBook	Klassen er hovedrammen for arbeidsboken, og har en meny for å navigere til komponentene i arbeidsboken.
ctlWorkBookStartPage	Dette er den startsiden som er det første som vises i arbeidsboken.
ctlWorkBookEditor	Arbeidsområde som kan innholde dokumenteditorer og øvingsoversikt.
ctlWorkBookDocumentEditor	En dokumenteditor der man kan skrive, lagre og gjenfinne arbeider.
ctlWorkBookExerciseExplorer	Denne klassen har en oversikt over øvinger til alle fagene til en lærer.
ctlWorkBookSaveDocument	Klasse for å lagre et dokument
ctlWorkBookOpenDocument	Klasse for å gjenfinne et dokument
ctlWorkBookTreeView	En oversikt over en brukers arbeidsbøker, vist i en trestruktur
ctlWorkBookEmailReader	Denne e-postleseren viser alle e-poster som har blitt sendt og mottatt.
ctlWorkBookEmailViewer	Dette er en klasse som viser innholdet i en e-post
frmWorkBookEmailWriter	I dette vinduet kan man skrive og sende e-post.
ctlWorkBookCalendar	I kalenderen kan man holde oversikt over avtaler.
frmWorkBookViewAppointment	Et vindu for å se detaljer om en avtale.
frmWorkBookEditAppointment	I dette vinduet kan man endre informasjonen som er lagret om en avtale.
ctlWorkBookCalendarDayPanel	Klassen er et panel som viser avtaler på en bestemt dag
ctlWorkBookAddressBook	Adresseboken har en liste med kontakter
frmWorkBookEditContact	I dette vinduet kan man editere informasjon om en kontakt

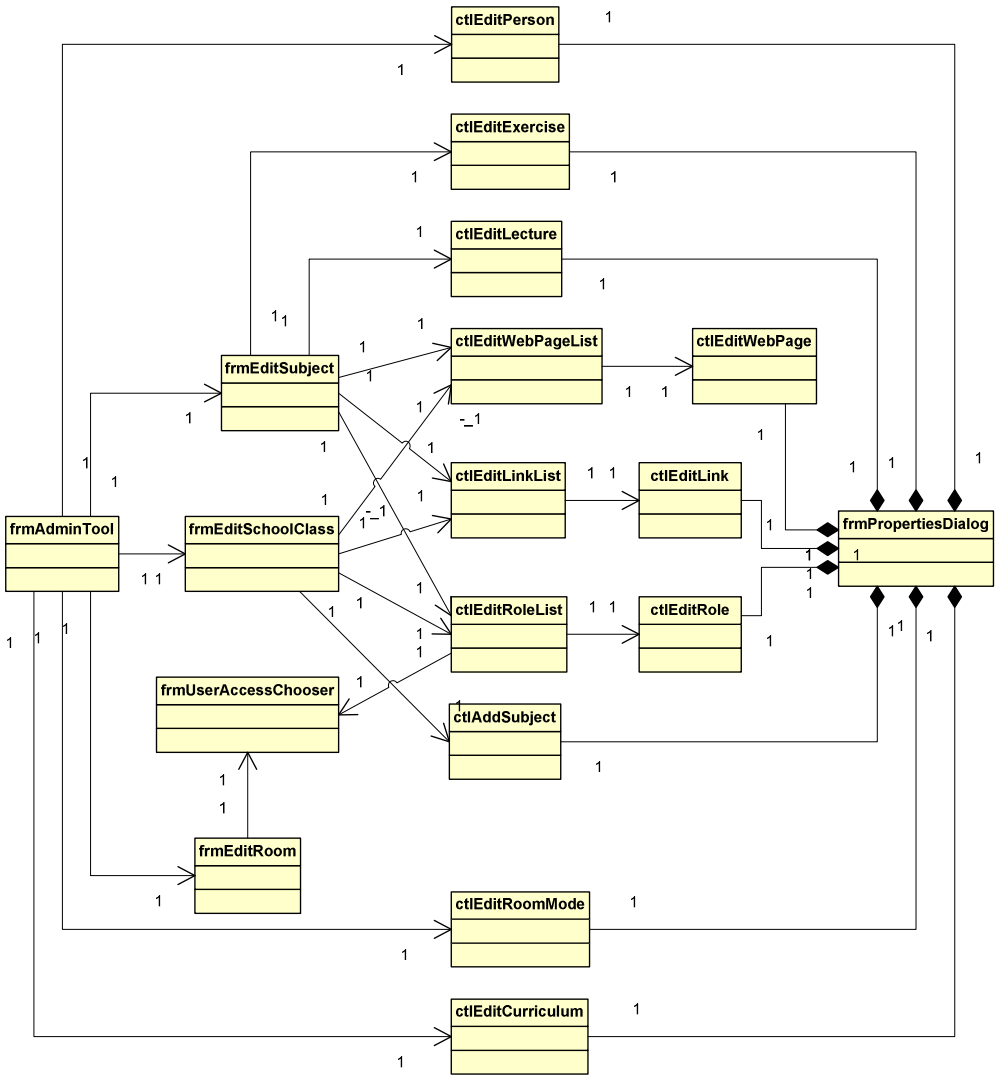
5.3.1.2 Rooms



Figur 5-11 Klassediagram for Rooms

Klassenavn	Beskrivelse
ctlRoomOrganiser	Dette er hovedrammen for rom. Klassen kan innholde flere rom og websider.
ctlRoom	Denne klassen har all informasjon om selve rommet, og kan inneholde flere forskjellige typer romkomponenter.
ctlChat	En romkomponent som lar brukeren kommunisere synkront med tekst med de andre brukerne i rommet.
ctlUserList	En romkomponent som viser en liste over hvem som er i rommet.
ctlUserDisplay	Viser bilde av en bruker. Denne komponenten brukes av ctlUserList og ctlLecture.
ctlUserExtendedDisplay	Viser bilde og litt informasjon om en bruker. Komponenten brukes av ctlUserList.
ctlControlPanel	Denne romkomponenten lar brukeren sette innstillinger for lyd og video.
ctlLecture	En romkomponent som viser informasjon om en forelesning. Klassen bruker frmQuestion.
frmQuestion	Her vise alle spørsmål som har blitt stilt i en forelesning. Det er også å sende inn nye spørsmål.
frmQuestionViewer	Dette vinduet viser detaljer om et spørsmål.
ctlBlackBoard	Dette er en romkomponent som styrer fellestavlen. Klassen bruker ctlBlackBoardDrawPanel.
ctlBlackBoardDrawPanel	Dette er en fellestavle som brukerne i rommet kan tegne på.
frmJoinRoom	I dette vinduet får brukeren opp en liste med de rommene som finnes i systemet.

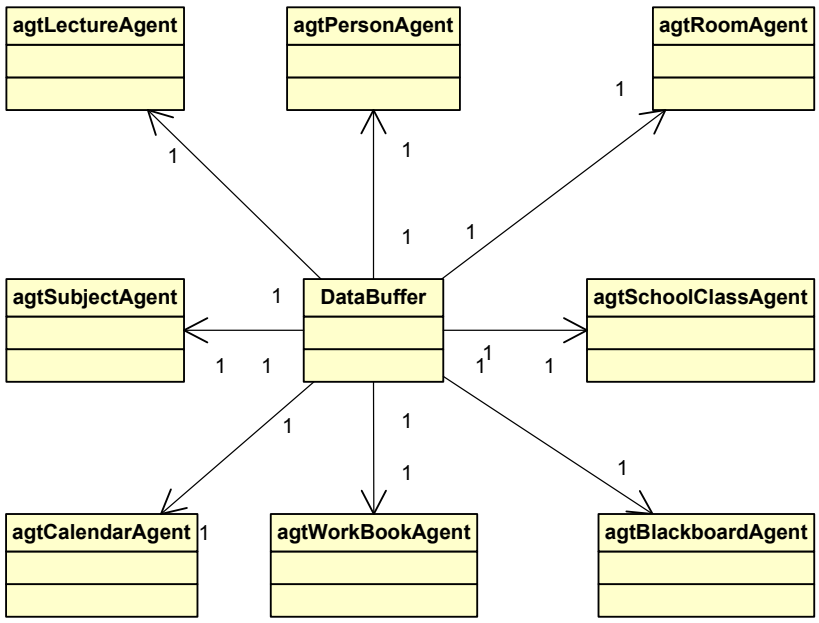
5.3.1.3 AdminTools



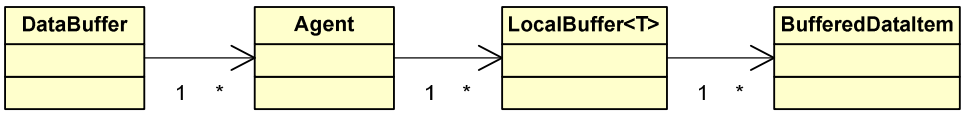
Figur 5-12 Klassediagram for AdminTools

Klassenavn	Beskrivelse
frmAdminTool	Dette er hovedvinduet til administrasjonsverktøyene.
frmEditRoom	Denne klassen brukes når man vil endre detaljer om et rom.
frmEditSchoolClass	Her kan man endre detaljer om en klasse.
frmEditSubject	Her kan man endre detaljer om et fag.
frmUserAccessChooser	I dette vinduet kan man velge brukere og velge hvilen rolle disse skal ha.
ctlEditRoleList	Dette er en liste med brukere og deres roller.
ctlEditRole	I denne komponenten velger man en rolle til en person.
ctlEditLinkList	Dette er en liste med linker som hører til en klasse eller et fag.
ctlEditLink	Her kan man legge til eller endre på en link.
ctlEditWebPageList	Dette re en liste med websider som hører til en klasse eller et fag.
ctlEditWebPage	Her kan man legge til websider eller endre på websider man allerede har lagt til.
ctlEditExrcise	I denne klassen kan man legge til eller endre en øving.
ctlEditLecture	Her kan man endre eller legge til en forelesning.
ctlEditPerson	Her kan man endre eller legge til en person.
ctlEditRoomMode	Dette er en klasse hvor man kan legge til eller endre en rommodus.
ctlEditCurriculum	I denne klassen kan man legge til eller endre en pensummodul.
frmPropertiesDialog	Dette er en vindusramme som noen av ctl-klassene bruker, slik at de får et likt utseende.

5.3.1.4 DataBuffering



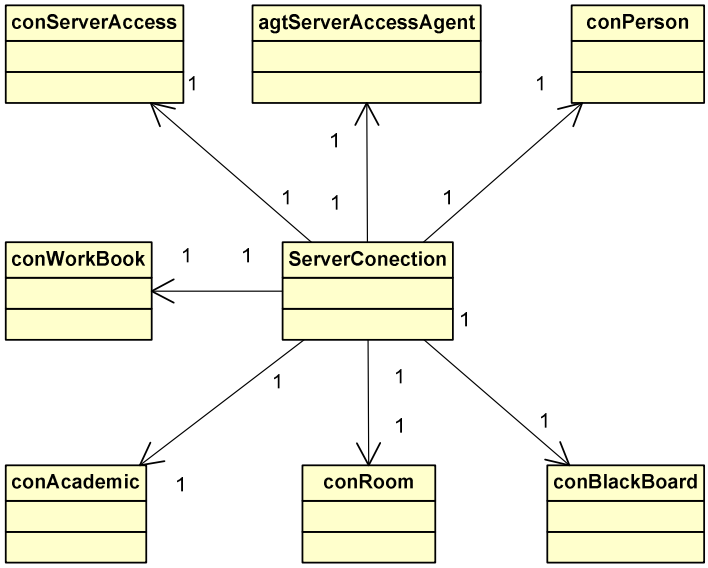
Figur 5-13 Klassediagram



Figur 5-14 Generell oppbygning av agenter og deres bufre

Klasse	Beskrivelse
Databuffer	DataBuffer er en statisk klasse som i hovedsak knytter sammen dataagentene med resten av programmet. Klassen inneholder referanser til hver av agentene, og i tillegg en Initialize-metode som starter agentene. Den inneholder også noen funksjoner for samkjøring av agentene. Klassen er en singletonklasse. Det vil si at det bare opprettes en instans av klassen.
Agentklassene	Dette er en samling klasser som indirekte formidler informasjon mellom programmet og databasen på serveren. Hver agent står for formidling av en eller flere typer informasjonsobjekter. Agenten sørger alltid for å ta vare på de dataene som tidligere har blitt hentet fra databasen, slik at man neste gang ikke trenger å hente dem fra serveren. I tillegg sjekker den serveren for oppdateringer, slik at objekter man har liggende lokalt blir oppdatert når det skjer endringer på serveren. Agentene samordner også i stor grad kallene mot serveren, slik at oppdateringer osv skjer i en ordnet rekkefølge. Hver av agentene er singletonklasser.
LocalBuffer	Dette er en generisk klasse som definerer et buffer av en viss type objekter. Klassen inneholder en liste over alle de bufrede objektene, samt oppdateringskøer, oppdateringsintervall og innstillinger for prebufring.
BufferDataItem	Dette er en abstrakt klasse som definerer rammene rundt dataobjektene som ligger bufret i agentene. Alle dataobjekter i programmet er knyttet til denne klassen gjennom arv. Klassen inneholder også metoder for å synkronisere dette objektet med databasen.

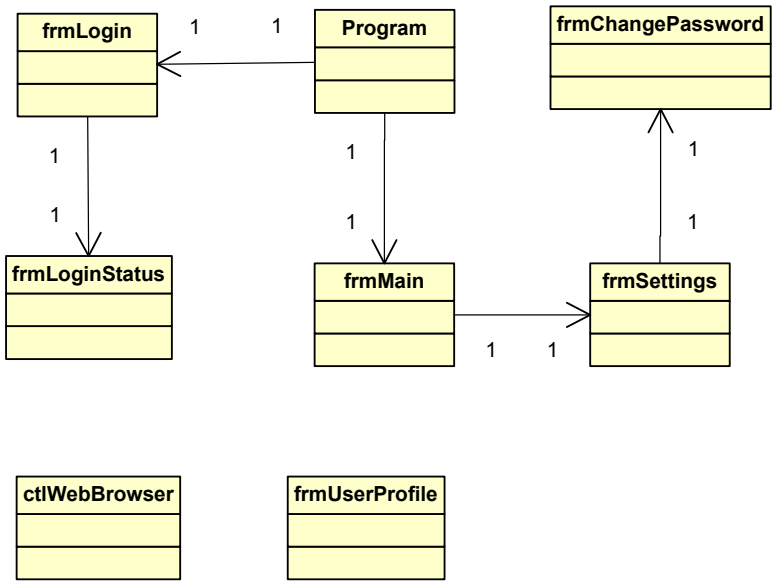
5.3.1.5 *ServerSide*



Figur 5-15 Klassediagram for ServerSide

Klassenavn	Beskrivelse
ServerConnection	Dette er en statisk klasse som er med og formidler den direkte kommunikasjonen mot serveren. Den inneholder referanse til alle de lokale webtjenesterepresentantene. Den formidler også kall til innlogging/utlogging og pinging til ServerAccessAgent. Klassen inneholder en Initialize-metode som setter opp tilkobling til serveren. Denne klassen er en singletonklasse.
agtServerAccessAgent	Denne klassen inneholder metoder for å logge inn og ut på serveren, og den inneholder også en Pinging-egenskap som kan slås av og på. Når Pinging slås på, vil agenten kalle en metode på serveren med jevne mellomrom og på den måten fortelle serveren av denne klienten fortsatt er tilkoblet. Denne klassen er en singletonklasse.
conAcademic	Dette er en webtjenesterepresentant som kommuniserer med webtjenesten AcademicWS.asmx. Denne tar seg av alle kall mot serveren som har med faglig informasjon å gjøre. Dette er en singletonklasse.
conBlackBoard	Dette er en webtjenesterepresentant som kommuniserer med webtjenesten BlackBoardWS.asmx. Denne tar seg av alle kall mot serveren som har med fellestavlen å gjøre. Dette er en singletonklasse.
conPerson	Dette er en webtjenesterepresentant som kommuniserer med webtjenesten PersonInfoWS.asmx. Denne tar seg av alle kall mot serveren som har med personinformasjon å gjøre. Dette er en singletonklasse.
conRoom	Dette er en webtjenesterepresentant som kommuniserer med webtjenesten RoomWS.asmx. Denne tar seg av alle kall mot serveren som har med rominformasjon å gjøre. Dette er en singletonklasse.
conServerAccess	Dette er en webtjenesterepresentant som kommuniserer med webtjenesten ServerAccessWS.asmx. Denne tar seg av alle kall mot serveren som har med nettilgang og innlogging/utlogging å gjøre. Dette er en singletonklasse.
conWorkBook	Dette er en webtjenesterepresentant som kommuniserer med webtjenesten WorkBookWS.asmx. Denne tar seg av alle kall mot serveren som har med. Dette er en singletonklasse.

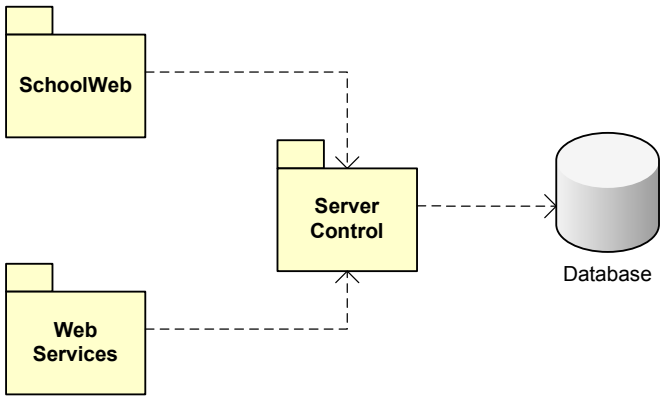
5.3.1.6 Andre klasser



Figur 5-16 Klassediagram

Klassenavn	Beskrivelse
Program	En statisk singletonklasse som kjører programmet
frmLogin	Dette er vinduet der brukeren skriver inn brukernavn og passord.
frmLoginStatus	Et vindu som viser hva som skjer i innloggingsprosessen.
frmMain	Dette er hovedvinduet for hele programmet. Vinduet inneholder rom- og arbeidsbokkomponenten.
frmSettings	I dette vinduet kan en bruker endre en del innstillinger.
frmChangePassword	Vindu for å endre passord.
ctlWebBrowser	Denne komponenten er en nettleser
frmUserProfile	Dette er et vindu som viser informasjon om en bruker.

5.3.2 Server

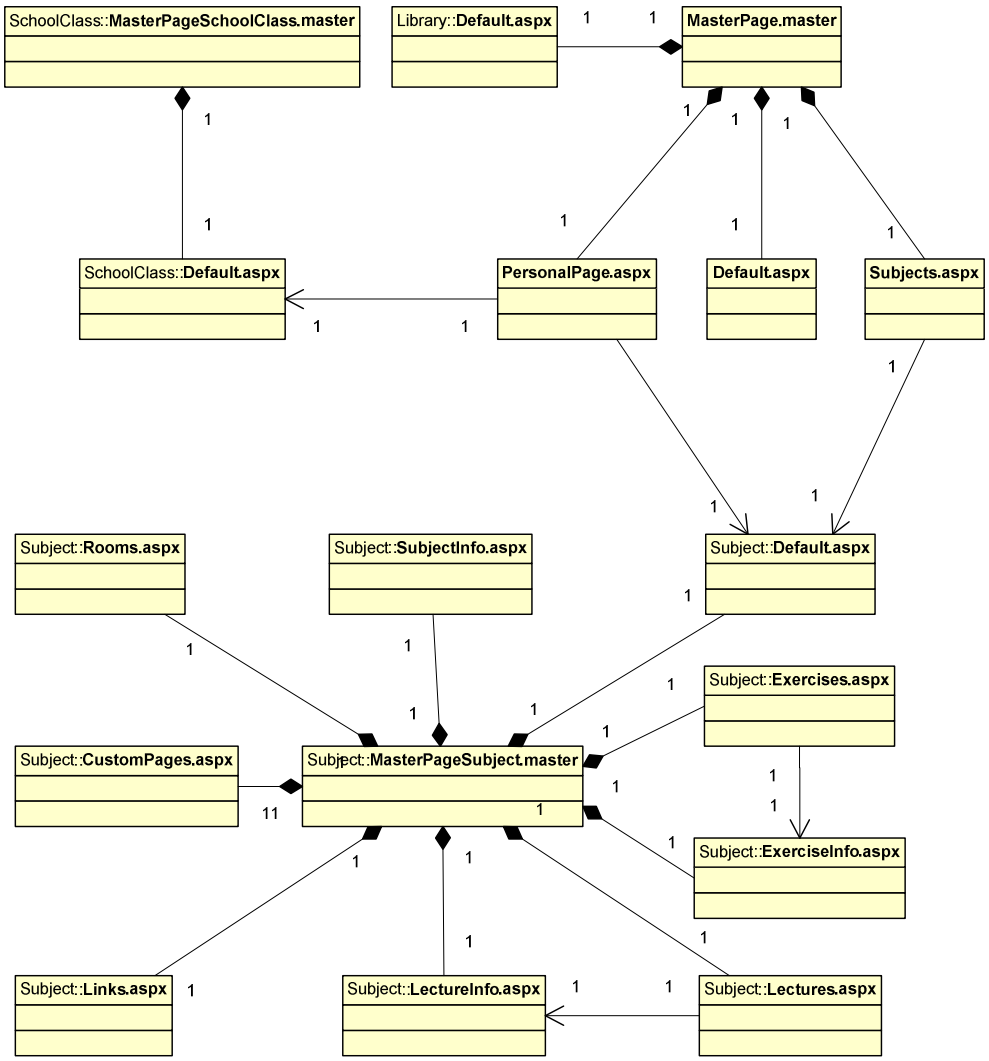


Figur 5-17 Oversikt over pakker på server

SchoolWeb er skolens websider med informasjon om blant annet klasser og fag som

skolen har. Web Services er webtjenestene som finnes på serveren. Det er disse som blir kalt fra klientprogrammer når de vil kommunisere med databasen. Server Control har kontrollklasser som blant annet kontrollerer brukerrettigheter og sørger for kommunikasjon med databasen.

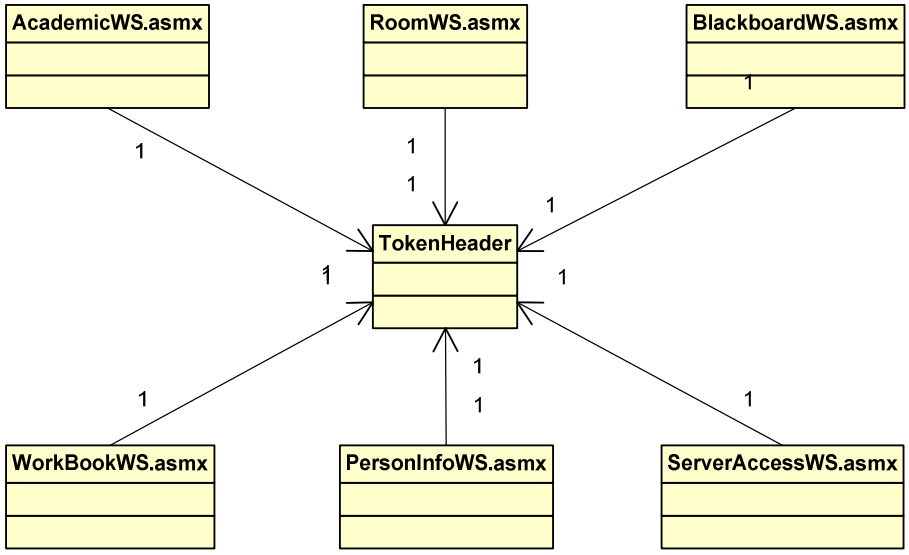
5.3.2.1 SchoolWeb



Figur 5-18 Klassediagram for SchoolWeb

Klassenavn	Beskrivelse
MasterPage.master	Dette er en mal for skolens webside.
Default.aspx	Dette er startsiden til skolens hjemmeside.
PersonalPage.aspx	Denne siden er en personlig side for hver bruker av systemet. Det er linker til en brukers fag og klasser på denne siden.
Subjects.aspx	På denne siden er det en liste over skolens fag.
Library::Default.aspx	Forsiden til skolens bibliotek.
SchoolClass::MasterPageSchoolClass.master	Dette er en mal for klassens webside.
SchoolClass::Default.aspx	Forsiden til en klasses webside. Her vises en liste med meldinger og avtaler for klassen.
Subject::MasterPageSubject.master	Dette er en mal for fagets webside.
Subject::Default.aspx	Forsiden til et fags webside. Her er en liste med meldinger og avtaler for faget.
Subject::SubjectInfo.aspx	En side med informasjon om et fag.
Subject::Rooms.aspx	Denne siden har en liste med rommene som hører til faget.
Subject::CustomPages.aspx	Siden viser en egendefinert webside som en lærer har lagt til faget.
Subject::Links.aspx	Her er det en liste med linker som er relevante til et fag.
Subject::Lectures.aspx	Denne websiden har en liste over forelesningene til faget.
Subject::LectureInfo.aspx	Her vises detaljer om en forelesning.
Subject::Exercises.aspx	Denne siden har en liste med øvinger i faget.
Subject::ExerciseInfo.aspx	Her vises detaljer om en øving.

5.3.2.2 Web Services

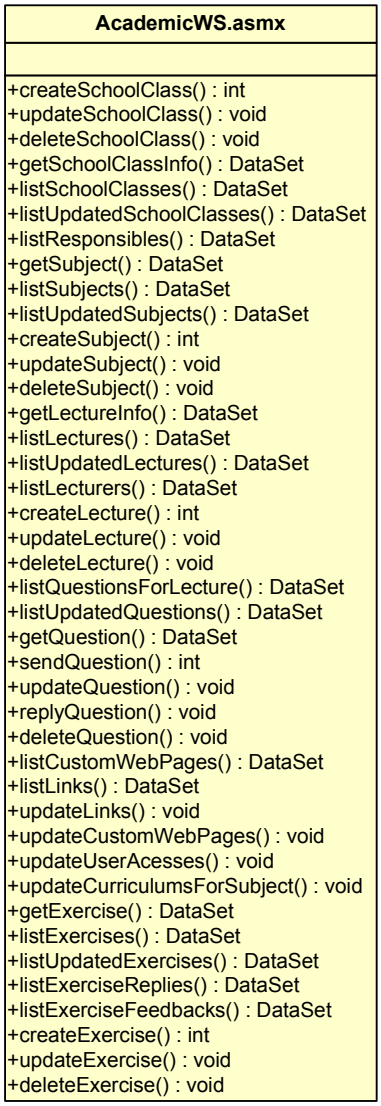


Figur 5-19 Klassediagram for Web Services

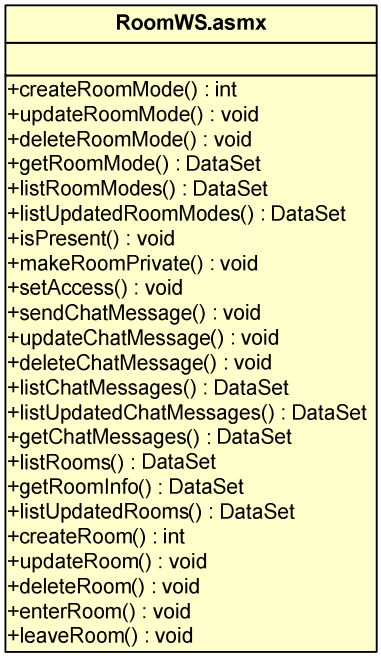
Alle webtjenesteklassene bruker TokenHeader-klassen. Denne klassen blir sendt som meldingshode i hvert kall til en metode i en av webtjenestene. I dette meldingshodet

ligger en unik signatur for hver bruker av systemet. Dette blir sendt med slik at serveren kan identifisere alle som kaller webtjenestene. På denne måten vil ingen utenforstående få tilgang til serveren.

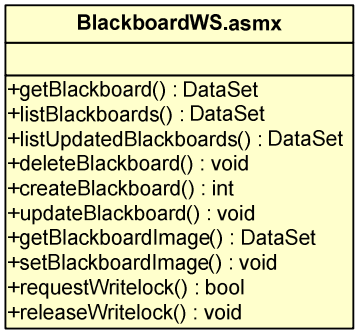
Figur 5-20 til figur 5-25 viser hvilke metoder som finnes i webtjenestene.



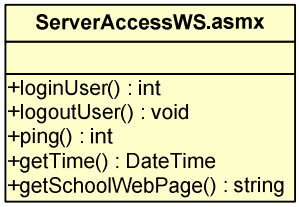
Figur 5-20 Metodeoversikt for Academic.asmx



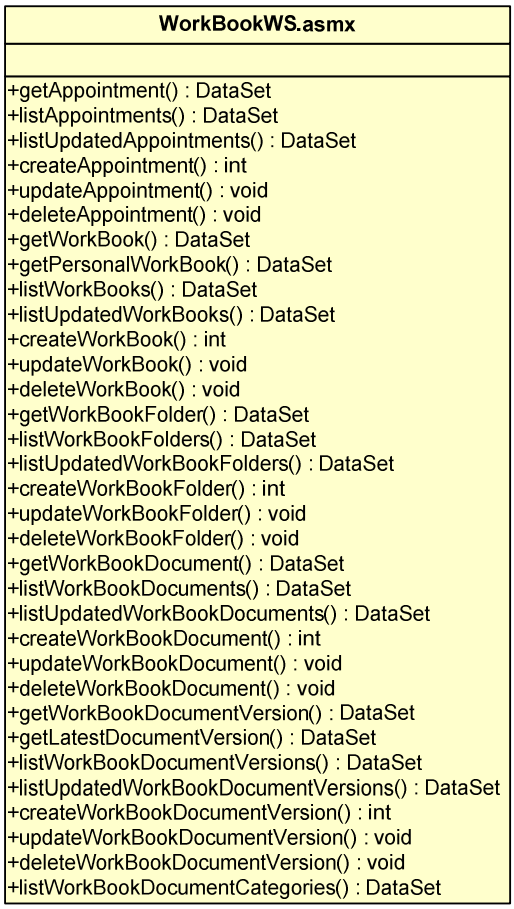
Figur 5-21 Metodeoversikt for RoomWS.asmx



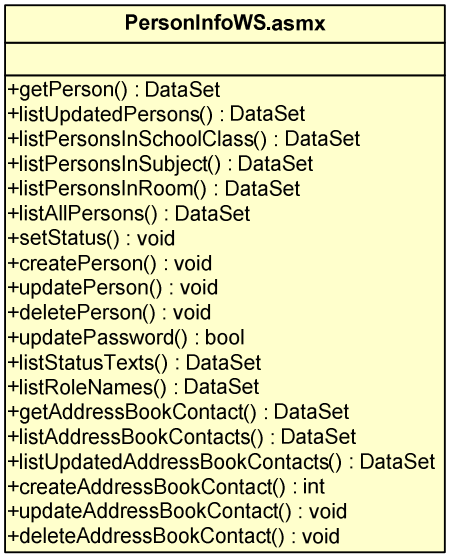
Figur 5-22 Metodeoversikt for Blackboard.asmx



Figur 5-23 Metodeoversikt for ServerAccessWS.asmx



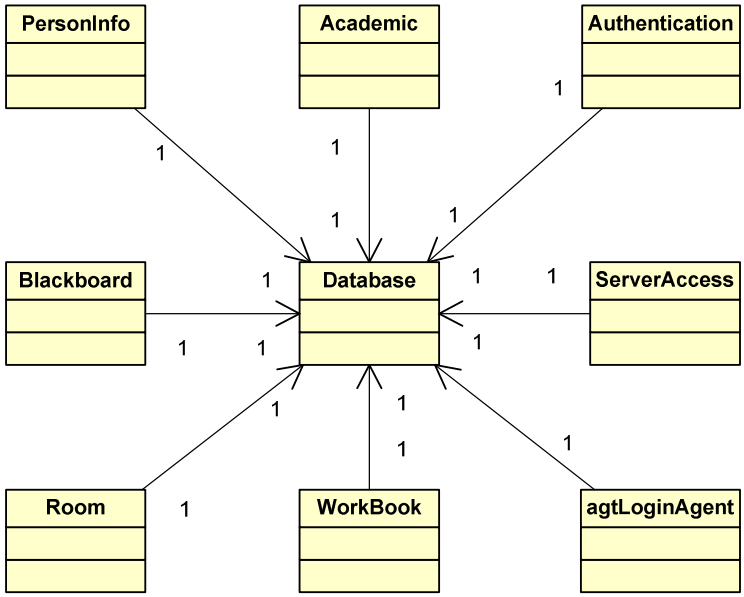
Figur 5-24 Metodeoversikt for WorkBookWS.asmx



Figur 5-25 Metodeoversikt for PersonInfo.asmx

Figur 5-26 Metodeoversikt for WorkBook.asmx

5.3.2.3 Server Control



Figur 5-27 Klassediagram for Server Control

Klassenavn	Beskrivelse
Academic	Dette er en kontrollklasse for AcademicWS.asmx. Klassen skal blant annet kontrollere en brukeres tilgang til ulike fag og skoleklasser.
agtLoginAgent	Denne klassen sjekker kontinuerlig at brukere som ikke er koblet til systemet har blitt logget ut riktig. Hvis den finner noen brukere som ikke har blitt logget av riktig, vil den selv logge av disse brukerne.
Authentication	I denne klassen autentiseres brukerne slik at bare brukere av systemet får tilgang til webtjenestene.
Blackboard	Denne kontollklassen for BlackboardWS.asmx skal kontrollere at kun brukere som er i et rom får tilgang til tavlen.
Database	Dette er det eneste klassen som har direkte kontakt med databasen. Alle klasser som vil ha kontakt med databasen må bruke denne klassen.
PersonInfo	Dette er en kontrollklasse for PersonInfoWS.asmx.
Room	Dette er en kontrollklasse for RoomWS.asmx. Den kontrollerer hvem som har tilgang til et rom.
ServerAccess	Denne kontrollklassen for ServerAccessWS.asmx kontrollerer netttilgangen til en bruker, og inn- og utlogging.
WorkBook	Dette er en kontrollklasse for WorkBookWS.asmx. Den kontrollerer hvem som har tilgang til arbeidsbøkene i systemet.

5.4 Prosesser

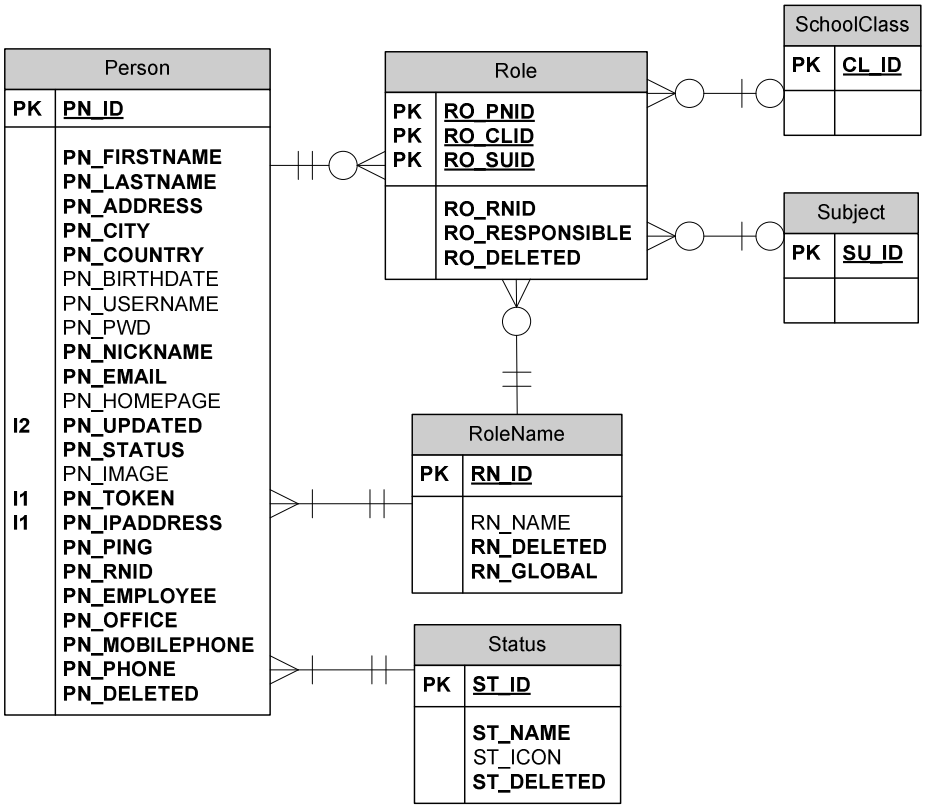
I programmet er det en del tråder som jobber med å oppdatere bufferne. Vedlikeholdstråder, som ligger i agentene, jobber med å holde bufferne oppdatert og i samsvar med databasen. Når det har skjedd en forandring i databasen, blir dette lagt i en oppdateringskø på klienten. Agentene kan ha flere forskjellige vedlikeholdstråder. Hver agent har også en oppdateringstråd. Disse trådene behandler dataene som er i oppdateringskøen. Forespørsler fra klienten blir lagt i en annen oppdateringskø med høyere prioritet. På denne måten vil klienten behandle dens brukers forespørsler først, og deretter endringer fra databasen. Disse trådene finnes i agentene:

Agent	Trådnavn
agtBlackboardAgent	maintainBlackboardListThread processUpdateRequestsThread
agtCalendaAgent	maintainAppointmentListThread processUpdateRequestsThread
agtLectureAgent	maintainLectureListThread maintainQuestionListThread processUpdateRequestsThread
agtPersonAgent	maintainPersonListThread maintainAddressBookContactListThread maintainUsersSubjectsThread maintainUsersSchoolClassesThread processUpdateRequestsThread
agtRoomAgent	maintainRoomListThread maintainRoomModeListThread maintainChatMessageListThread maintainUsersInRoomThread processUpdateRequestsThread
agtSchoolClassAgent	maintainSchoolClassListThread maintainPersonsInSchoolClassesThread processUpdateRequestsThread
agtSubjectAgent	maintainSubjectListThread maintainExerciseListThread maintainCurriculumListThread maintainPersonsInSubjectsThread processUpdateRequestsThread
agtWorkBookAgent	maintainWorkBookListThread maintainWorkBookFolderListThread maintainWorkBookDocumentListThread maintainWorkBookDocumentVersionListThread processUpdateRequestsThread

5.5 Database

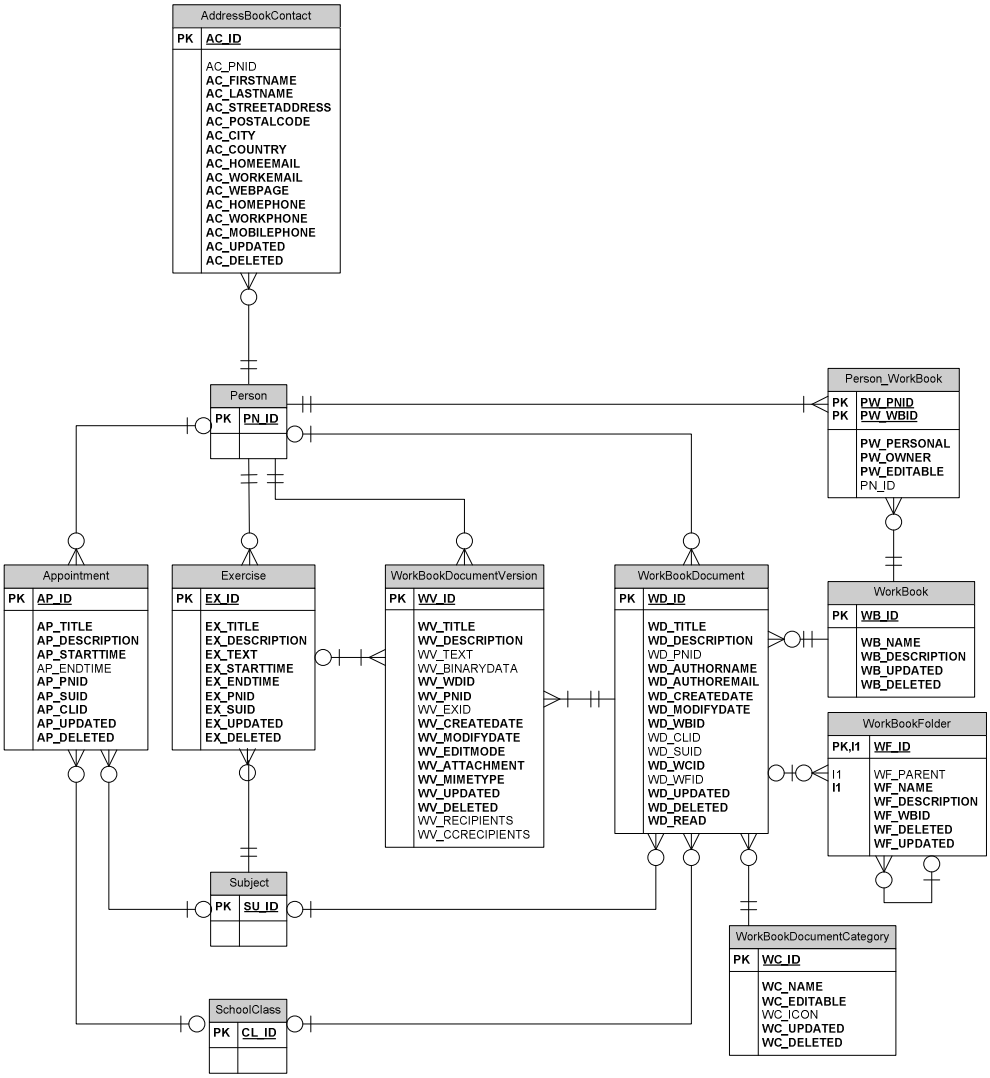
Databasen er en relasjonsdatabase som er laget i Microsoft SQL Server. All kommunikasjon med databasen skjer med prosedyrer. Dette er ferdig programmerte spørringer som blir kompilert inn i databasen. Prosedyrene er optimalisert til databasen og er derfor raskere enn å bruke spørringer direkte fra programmet. Siden spørringene skjer i prosedyrene, trenger man ikke kompilere programmet på nytt hvis man vil gjøre en forandring i en spørring. Nedenfor vises databasen inndelt i grupper for å lettere få oversikt.

5.5.1 Person og roller



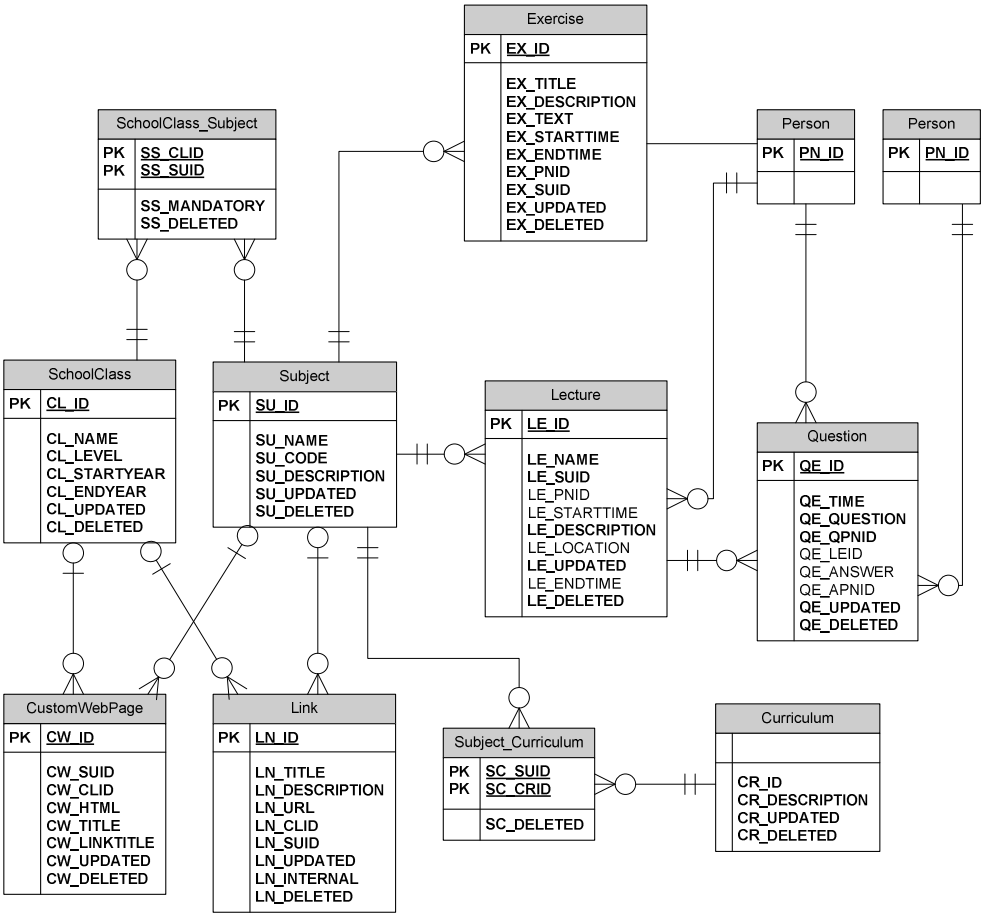
Figur 5-28 Datamodell for person og roller

5.5.2 Skrivebord



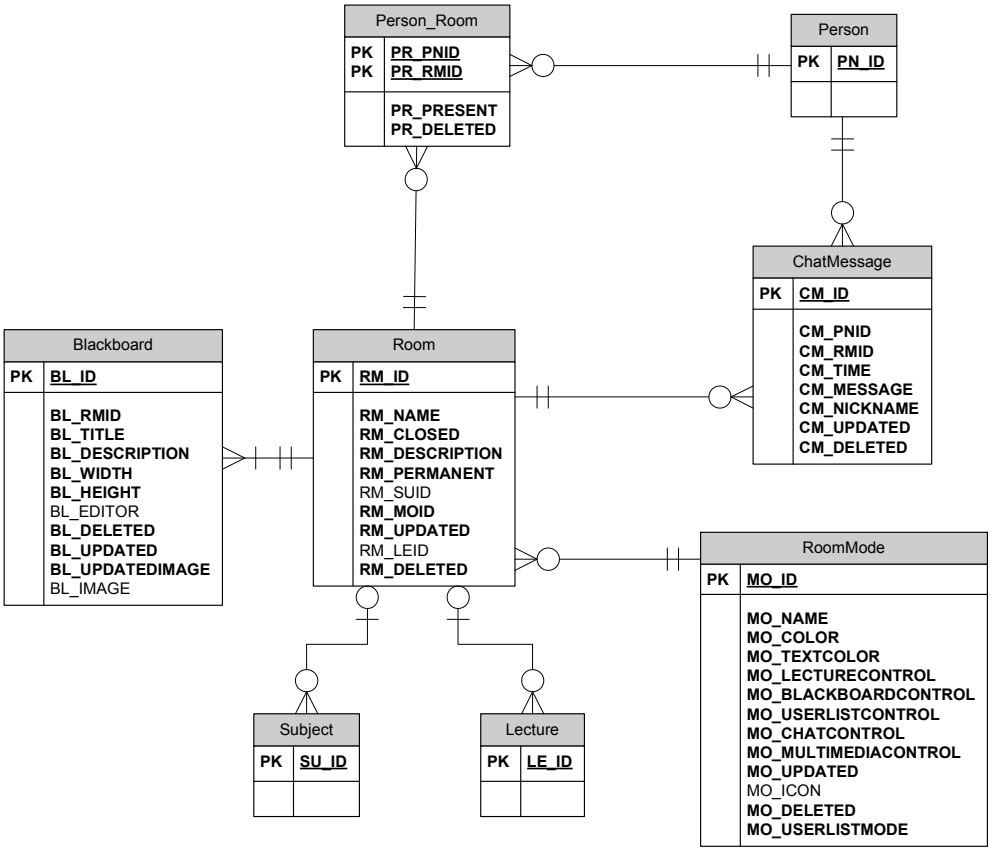
Figur 5-29 Datamodell for skrivebord

5.5.3 Fag og klasser



Figur 5-30 Datamodell for fag og klasser

5.5.4 Rom



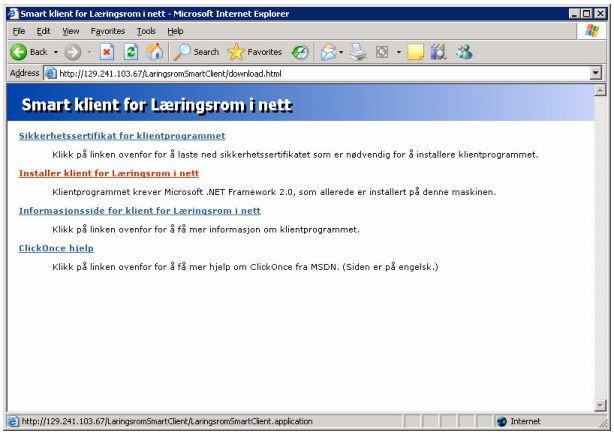
Figur 5-31 Datamodell for rom

6 Brukermanual

6.1 Installasjon, oppstart og innstillinger

6.1.1 Nedlasting og installasjon

For å kunne bruke smart klient for Læringsrom i nett, må man laste ned denne og installere den på den maskinen man sitter ved. Dette gjøres ved å gå til programmets nedlastingsside på internett (Figur 6-1) (Se Appendiks A). Microsoft sitt .NET-rammeverk versjon 2.0 er påkrevd for å kunne kjøre klientprogrammet. Nettsiden vil automatisk sørge for at dette rammeverket blir installert dersom det ikke finnes på datamaskinen din fra før. På nedlastingssiden finnes også linker både til generell informasjon om klientprogrammet, og til informasjon om .NET og ClickOnce (installasjonssystemet) generelt.

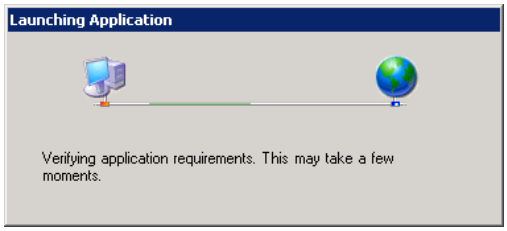


Figur 6-1 Nedlastingsside for klientprogrammet.

Denne utgaven av Læringsrom i nett er ikke blitt digitalt signert av en godkjent aktør, men i stedet er et testsertifikat utgitt av utviklerne blitt brukt. Dette medfører at du må følge linken øverst på siden, laste ned dette sertifikatet og installere det manuelt, før du får mulighet til å installere programmet. I fremtiden vil det være naturlig å skaffe et ordentlig godkjent sertifikat for programmet, og brukeren kan da installere programmet uten å måtte tenke på sertifikatet.

6.1.2 Oppstart og innlogging

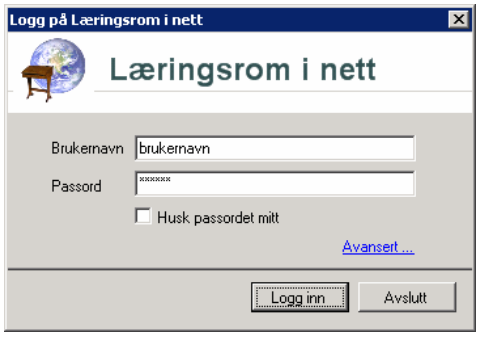
Når programmet er installert kan det startes både fra startmenyen i Windows og fra internettsiden det ble installert fra. .NET-rammeverket vil undersøke om programmet er installert, og samtidig undersøke om det finnes nyere versjoner på internett. Dersom versjonen man har er utdatert, lastes det automatisk ned nyeste versjon slik at programmet alltid er oppdatert. Dette er viktig for at alle klientene skal kunne fortsette å kommunisere med webserveren dersom denne blir oppdatert.



Figur 6-2 Rammeverket henter programinformasjon fra internett.

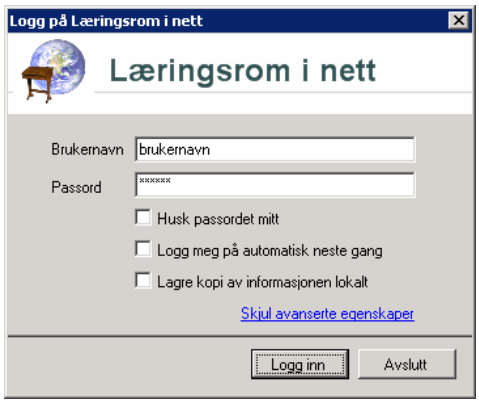
Når programmet starter kommer man først til innloggingsvinduet (Figur 6-3) (dersom man ikke tidligere har valgt automatisk pålogging). I utgangspunktet vises dette

vinduet i en enkel versjon der man kan skrive inn brukernavn og passord. I tillegg kan man krysse av for at programmet skal huske passordet, slik at dette automatisk fylles inn neste gang man skriver inn brukernavnet.



Figur 6-3 Vanlig innloggingsvindu

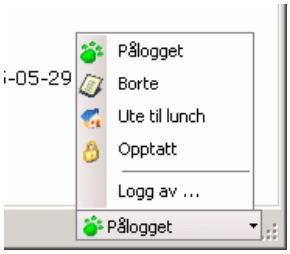
Dersom man klikker på ”Avansert …” nede til høyre i det vanlige innloggingsvinduet, vil man få opp flere innloggingsvalg (Figur 6-4). For det første kan man krysse av for ”Logg meg på automatisk neste gang”, noe som fører til at klienten bruker gjeldende brukernavn og passord til å logge på systemet neste gang, uten å vise innloggingsvinduet. Dette forutsetter at man har skrevet riktig brukernavn og passord, ellers vil man få opp innloggingsvinduet igjen.



Figur 6-4 Avansert innloggingsvindu

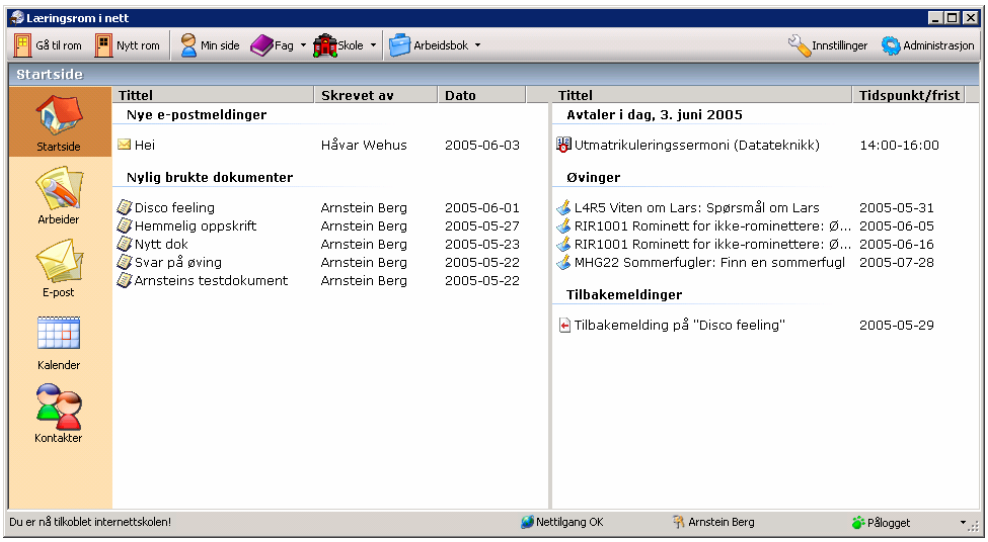
Det andre valget brukeren får i det avanserte innloggingsvinduet, er å lagre informasjon lokalt. All informasjon som systemet henter inn fra webserveren, legges normalt i maskinens minne mens programmet kjører, slik at ikke informasjonen må hentes via internett hver eneste gang man skal vise den. Med dette valget avkrysset, vil dette lagres på maskinens harddisk når programmet avsluttes, slik at man ikke trenger å vente på at all informasjonen skal lastes ned fra internett neste gang programmet startes. Dette er imidlertid ikke ferdig implementert i gjeldende utgave av programmet, men vil etter planen fungere i fremtidige utgaver.

Når du har logget deg inn i systemet (eller du har fått tilgang til å jobbe frakoblet), vil hovedvinduet komme opp (Figur 6-6). Her har man tilgang til samarbeidsrom, personlig skrivebord og skoleweb (se etterfølgende kapitler).



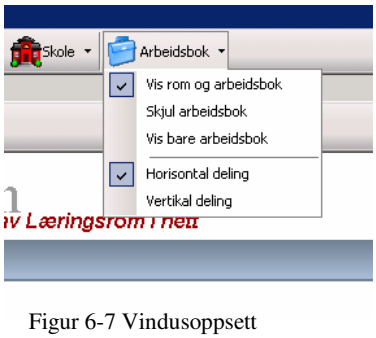
Figur 6-5 Statusmeny

På statuslinjen nederst i hovedvinduet, vil man se om man har kontakt med webserveren og hvilken bruker som er logget inn. Det er mulig å velge egen status fra menyen nederst til høyre (Figur 6-5), slik at andre brukere som skal ha tak i en vet om man er til stede eller borte fra datamaskinen.



Figur 6-6 Hovedvinduet etter innlogging

Når programmet startes vil arbeidsboken, med brukerens egen personlige startside vises over hele hovedvinduet. Arbeidsboken utgjør den personlige delen av systemet. Idet brukeren velger å gå til et rom eller en side i skolewebben, vil et fellesområde åpnes øverst i vinduet. Samarbeidsrom og skoleweb vises i dette området. Vinduet består nå både av en personlig del (nederst) og en fellesdel (øverst), og disse er delt med en skillelinje. Denne linjen kan brukeren forskyve opp og ned, og ved hjelp av "Vindusoppsett"-menyen på verktøylinja øverst i vinduet, kan man også velge bare en av delene alene. Menyen lar deg også velge å ha skillelinjen vertikalt over skjermen, slik at fellesdelen vises til venstre og den personlige delen til høyre.



Figur 6-7 Vindusoppsett

6.1.3 Innstillinger

Klientprogrammet er ment å kunne startes og så fungere, uten at brukeren trenger å gjøre tekniske innstillinger. Det er med hensikt laget slik at man ikke har noen valg rundt nettverkstilkoblingen, eller bruken av andre ressurser i maskinen.

De eneste innstillingene som i skrivende stund er implementert i klientprogrammet er personinnstillinger (Figur 6-8). Disse kan nås fra "Innstillinger"-knappen øverst til høyre i hovedvinduet. Her kan du endre dine egne personalia, inkludert navn, telefon og adresser. I denne utgaven av programmet kan alle brukere endre sine egne personalia, med unntak av fødselsdato, men i senere versjoner kan det være naturlig å innskrenke denne muligheten, slik at endringer av personalia må gå via skolens administrasjon.

Videre kan brukeren endre sitt skjermnavn og personlige bilde. Skjermnavn er det som vises ved siden av brukeren i samarbeidsrom og enkelte andre steder. Dette er ment å være et uhøytidelig kallenavn, typisk personens fornavn. I alle sammenhenger der dette brukes, vil det også være mulig å få se personens fulle navn, for eksempel ved å bevege musen over skjermnavnet. Det personlige bildet er ment å representere brukeren i blant annet grupperom.

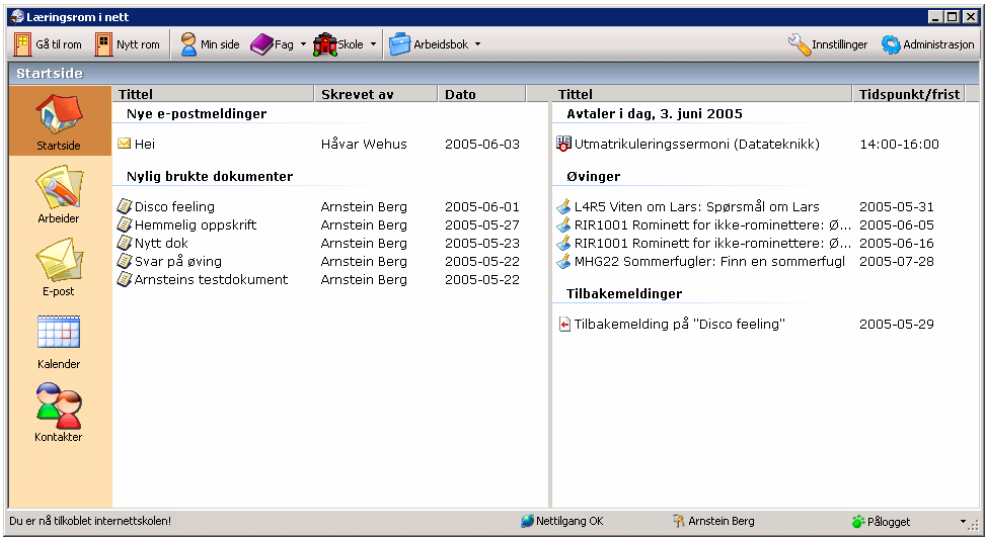
Figur 6-8 Personinnstillinger

I personlige innstillinger kan man også endre passord. Dette gjøres ved å klikke på knappen ”Endre passord …” nederst i innstillingsvinduet. Man får da opp et vindu der man først må skrive inn det gjeldende passordet, og så det nye passordet. Det nye passordet må skrives to ganger, for å være sikker på at brukeren ikke har skrevet feil.

6.2 Skrivebord

Skrivebordet er en fellesnevner for alle de personlige delene av systemet. Det vil si dine egne dokumenter, øvinger, e-postmeldinger, osv. Skrivebordet er det første du får opp når du logger deg på, og den er tilgjengelig overalt i systemet.

6.2.1 Startsiden



Figur 6-9 Personlig arbeidsbok med brukerens startside

Startsiden (Figur 6-9) er ment å være et sted der man lett kan finne det man er ute etter. Startsiden består av to spalter. Til venstre har man egne dokumenter og inngående e-post, mens den høyre spalten viser avtaler, øvinger og tilbakemeldinger. De forskjellige kategoriene vises bare dersom det er noe innhold i dem.

Inngående e-post vises på startsiden så lenge den er ulest. Det vil si at når du åpner den i e-postleseren (Kap. 6.2.3) fjernes den fra startsiden. Det er dermed lett å se hvilke meldinger du har fått siden sist du leste e-post. Ved å dobbeltklikke på e-postmeldingen vil man få åpnet denne i e-postleseren.

Startsiden viser også de ti dokumentene du sist har gjort endringer på. Dette for å raskt finne tilbake til det du holdt på med sist du var inne i systemet. Dokumentene åpnes i dokumenteditoren (Kap. 6.2.2) ved å dobbeltklikke på dem. I denne versjonen av programmet kommer ikke dokumenter i denne listen før du ender noe og lagrer dem, men i senere versjoner kan det være aktuelt å gjøre slik at listen oppdateres ut fra hvilke dokumenter du åpner, og således blir en ”Nylig åpnede dokumenter”-liste, slik man er vant til fra for eksempel Windows.

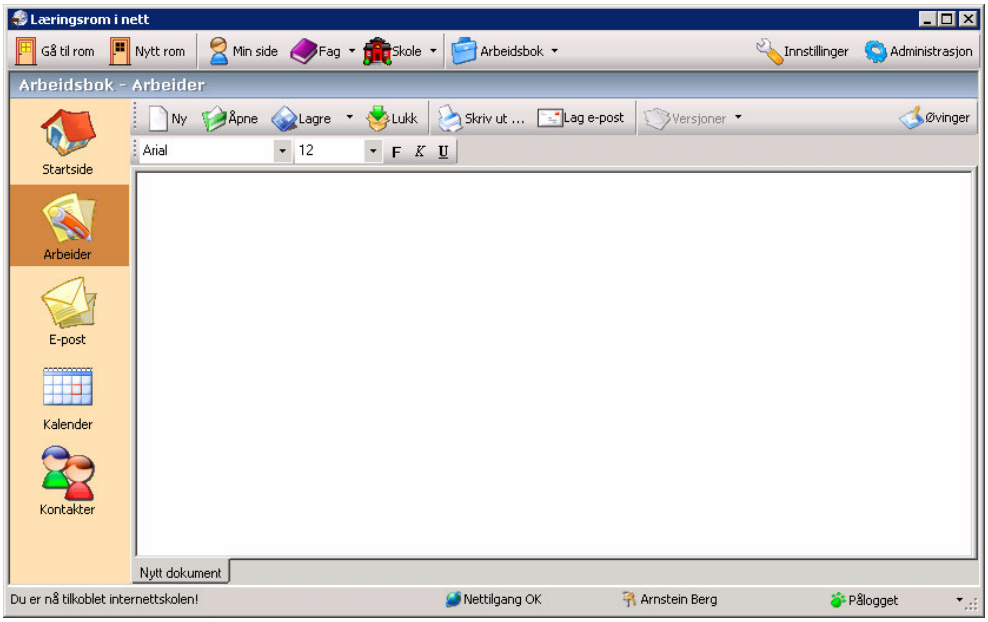
Øverst i den høyre spalten kommer dagens avtaler opp, dersom det er noen denne dagen. Her vil både dine personlige avtaler komme, samt avtaler som er lagt inn for fag og klasser som du er med i. Forelesninger i fagene dine kommer også opp her.

Videre i den høyre spalten vises alle dine øvinger som du ikke har svart på ennå. Det vil si offentliggjorte øvinger i dine fag, som du ikke har sendt inn noe svardokument på ennå. Dersom man er lærer eller assistent i et fag vil også disse øvingene komme opp i listen. For informasjon om hva som skjer når man klikker på øvinger, se kapittel 6.2.2.6.

6.2.2 Arbeidsbok

Arbeidsboken er den mest sentrale delen av skrivebordet. Det er her man skriver dokumenter og løser øvinger, samt retter øvinger som lærer.

6.2.2.1 Skrive dokumenter



Figur 6-10 Nytt dokument i arbeidsboken

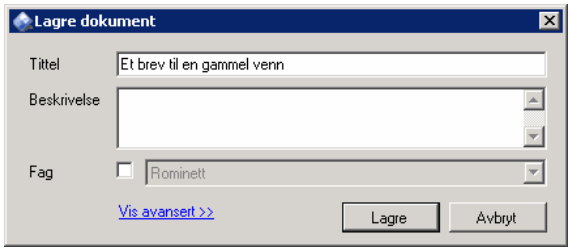
Når man går inn i arbeidsboken uten å ha klikket på et dokument på startsida eller lignende, vil man først bare få opp et stort tomt felt. Man kan nå velge å lage et nytt dokument eller åpne et eksisterende ved å benytte seg av henholdsvis ”Ny”- eller ”Åpne”-knappen på toppen av det grå feltet. Hvis man velger å opprette et nytt dokument åpnes en side med et tomt dokument (Figur 6-10). I denne versjonen av programmet er tekstbehandleren veldig lite utbygget, og er kun ment å tilby den aller mest nødvendige funksjonaliteten som kreves for å kunne bruke funksjonene for dokumenter, e-post og øvinger. En verktøylinje med funksjoner for å endre skrifttype og størrelse, er også inkludert, men denne er kun ment som et eksempel på hva som kan komme av tekstbehandlingsfunksjoner. Formateringen av teksten vil ikke bli lagret.

I fremtidige versjoner skal man ha tilgang til en ordentlig tekstbehandler med alle de vanligste funksjonene man kjenner fra andre tekstbehandlere. I tillegg ønsker vi å gjøre det mulig å lage og redigere andre typer dokumenter i systemet vårt, for eksempel regneark og figurer/diagrammer. I utgangspunktet går det an å legge inn alle mulige filtyper i databasen, men hvordan disse vil bli vist i vårt program, og om de eventuelt vil kreve bruk av andre programmer for visning og redigering, er ennå et åpent spørsmål.

6.2.2.2 Lagre dokumenter

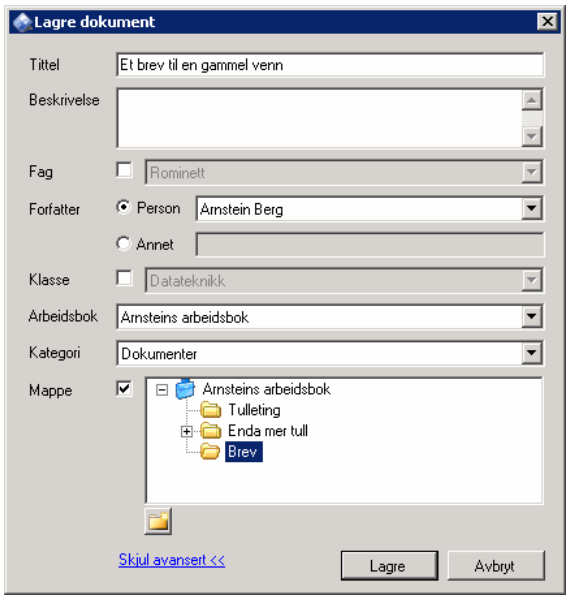
For å lagre dokumentet du skriver på, klikker du ”Lagre”-knappen på verktøylinjen ovenfor dokumentet. Dersom dette er et nytt dokument, vil du få opp en dialogboks der du kan skrive inn tittel og beskrivelse til dokumentet (Figur 6-11). Dette er søkbar informasjon som du kan bruke til å finne igjen dokumentet siden. Du kan også velge et

fag som dette dokumentet er tilknyttet. I mange tilfeller vil også fag på forhånd være valgt ut fra sammenhengen der dokumentet ble opprettet. Merk avkrysningsboksen og velg et fag fra nedtrekksmenyen.



Figur 6-11 Lagre dokument, enkel visning

Ved å klikke på linken "Vis avansert >>" nederst til venstre i dialogboksen, får du opp enda flere valg (Figur 6-12). Dette er informasjon som man kan velge om man vil spesifisere, og som kan hjelpe til i organiseringen av arbeidsboken din. Normalt er innlogget bruker satt som forfatter, men man kan også spesifisere en annen person i systemet, eller man kan skrive inn en tekst som identifiserer forfatter(e) dersom det er flere personer eller personer utenfor systemet. I tillegg kan man velge å angi klasse. Dette er også informasjon som kan hjelpe til når man skal gjenfinne dokumentet siden.



Figur 6-12 Lagre dokument, avansert visning

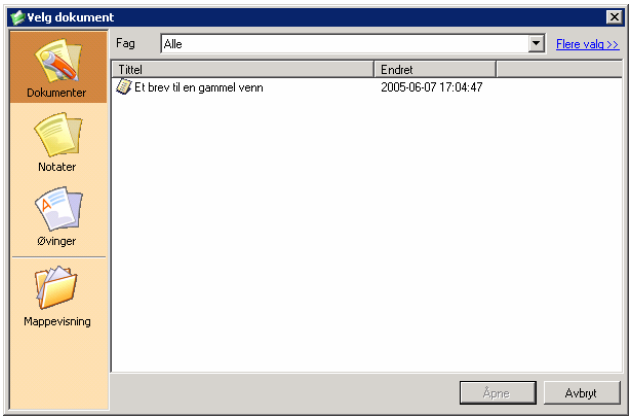
Videre er man nødt til å spesifisere arbeidsbok og kategori. Dette blir automatisk satt til din personlige arbeidsbok og en kategori basert på sammenhengen der dokumentet ble opprettet, hvis du ikke velger noe annet. Alle brukere har en personlig arbeidsbok, men i tillegg går det an å ha flere arbeidsbøker. Det er for eksempel mulig at en gruppe som jobber sammen, har en felles arbeidsbok. Man kan da lett gi de andre gruppemedlemmene tilgang til dokumentet ved å lagre det i den felles arbeidsboken.

Du kan også velge å spesifisere en mappe å lagre dokumentet i. Dette er den måten mange erfarne databrukere er vant til å organisere dokumentene sine på, og den fungerer på samme måten her. Mange brukere med mindre dataerfaring kan imidlertid synes det er vanskelig å finne igjen dokumenter når de ligger i mapper, og derfor har vi valgt å gjøre det valgfritt om man vil benytte seg av mapper. Uansett om mappe er spesifisert eller ikke, kan man finne igjen dokumentet bare ved å søke på fag, klasse, kategori og en eventuell søketekst. Med den lille knappen like nedenfor mappetreet kan man for øvrig lage en ny mappe under den man har valgt.

Dersom du velger å lagre et allerede eksisterende dokument, vil ikke dialogboksen komme opp, og dokumentet lagres med den informasjonen som ble satt sist. Dersom du ønsker å endre noe av informasjonen kan du klikke på nedoverpilen like til høyre for "Lagre"-knappen, og velge "Lagre som …" fra menyen som kommer opp.

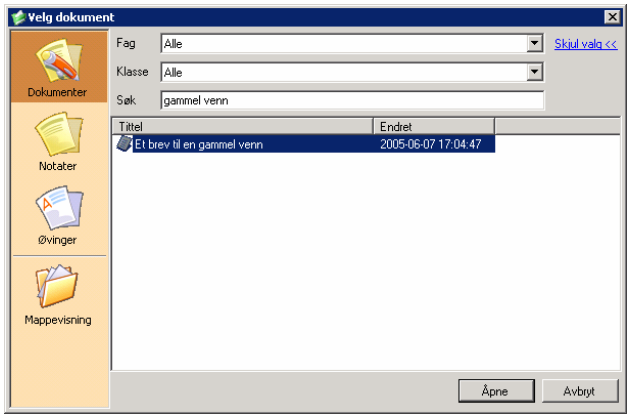
6.2.2.3 *Gjenfinning av dokumenter*

Dersom man vil åpne et nylig brukt dokument, finner man en snarvei til dette på startsiden. Skulle man ønske å åpne et dokument som man ikke har brukt på en stund må man gå inn i arbeidsboken og klikke på ”Åpne”-knappen der. Man vi da få opp dialogboksen for å åpne dokumenter (Figur 6-13).



Figur 6-13 Åpne dokument, enkel visning

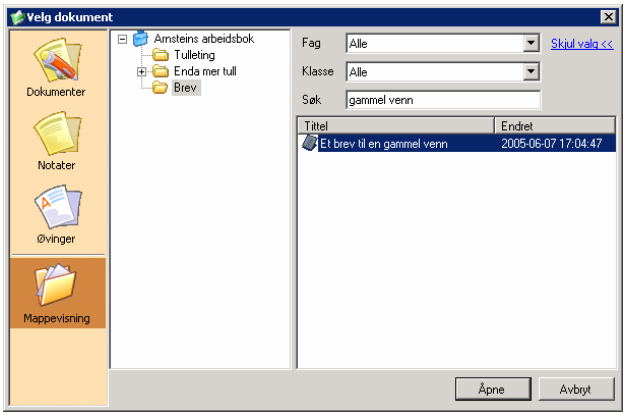
I denne dialogboksen finner man først de forskjellige kategoriene i margen til venstre. I dokumentlisten vises alle dokumentene i alle dine arbeidsbøker i den valgte kategorien. Øverst i vinduet kan man dessuten velge et fag, og da vil man bare få opp dokumenter i den valgte kategorien, tilknyttet det valgte faget.



Figur 6-14 Åpne dokument, flere valg

Hvis man klikker på linken ”Flere valg >>” øverst til høyre i vinduet, får man også mulighet til å velge klasse, samt å skrive inn en søketekst (Figur 6-14). Dokumentlisten oppdateres kontinuerlig til å vise dokumenter i valgt fag og klasse som oppfyller søkekriteriet.

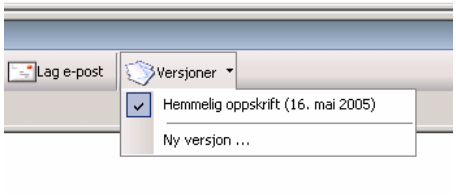
Dersom man foretrekker å finne fram til dokumentet ved å bla seg gjennom mapper på ”gamlemåten”, er dette mulig ved å velge ”Mappevisning” i venstremargen (Figur 6-15). Da får man opp et mappetre, der man kan velge hvilken arbeidsbok og/eller mappe man vil lete i. Dokumentlisten vil nå ikke skille på kategorier, men den vil fortsatt begrenses til å bare vise dokumenter innen valgt fag og klasse som inneholder spesifisert søkestreng i tittel eller beskrivelse.



Figur 6-15 Åpne dokument, mappevisning

6.2.2.4 *Dokumentversjoner*

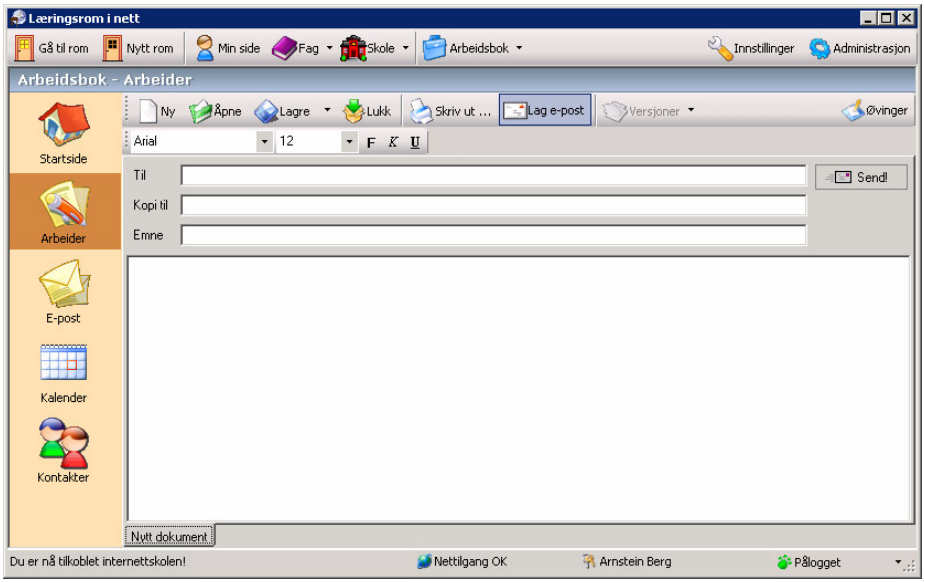
Som en følge av at det skal være mulig å ha flere utgaver av et dokument, har vi lagt inn støtte for å legge til flere versjoner i et dokument. Man kan se de forskjellige versjonene til et dokument ved å klikke på "Versjoner" på verktøylinjen over dokumentet (Figur 6-16). Her kan man også legge til ny versjon. Denne menyen er bare tilgjengelig etter at dokumentet har blitt lagret.



Figur 6-16 Dokumentversjonsmeny

Når man lager en ny versjon vil denne bli en kopi av den forrige versjonen. Tittelen på en versjon blir satt til dokumentets tittel, men den blir ikke endret siden, så gamle versjoner heter det samme som dokumentet het da de ble opprettet. Man kan alltid bare endre på nyeste (gjeldende) versjon, men gamle versjoner kan vises ved å velge dem i versjonsmenyen. Når man sender dokumentet som e-post (se kapittel 6.2.2.5) eller leverer det som svar på en øving(se kapittel 6.2.2.6), vil versjonen man sender bli låst, og man er da nødt til å opprette en ny versjon for å kunne redigere videre på dokumentet. Dette er for å sikre at man alltid selv har kopi av det man har sendt til andre. Når man får tilbakemelding fra læreren på en øving man har levert inn, vil denne tilbakemeldingen også komme som en versjon av dokumentet.

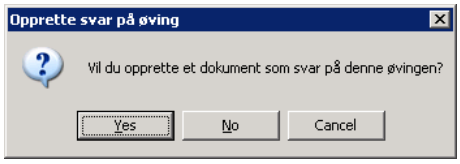
6.2.2.5 *Send som e-post*



Figur 6-17 Dokument i e-postmodus

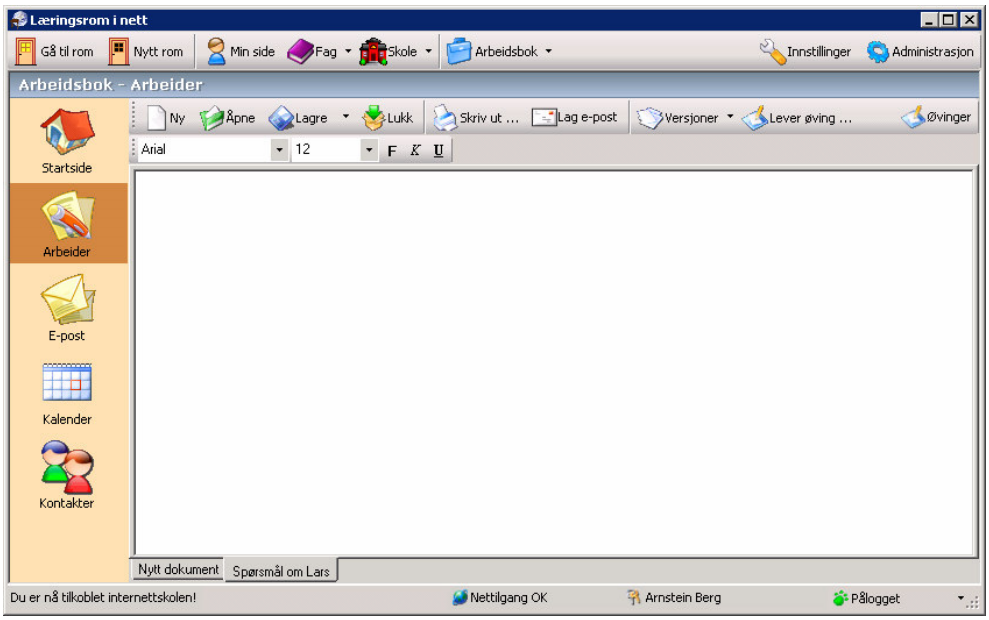
Dersom du vil sende et dokument man har skrevet som e-post, klikker du på "E-post"-knappen ovenfor dokumentet. Da vil du få opp felter for å skrive inn mottakeradresse(r) og emne (Figur 6-17). Normalt vil dokumentets tittel settes som emne, men du kan endre dette selv. Klikk på "Send"-knappen for å sende dokumentet som e-post. Den dokumentversjonen du skrev på vil nå bli merket som sendt, og er ikke lenger redigerbar. Du kan opprette en ny versjon av dokumentet hvis du vil fortsette å skrive på dokumentet (se kapittel 6.2.2.4). På denne måten vil du alltid ha en kopi av det du har sendt til andre.

6.2.2.6 *Levere øvinger*



Figur 6-18 Opprette svar på øving

Systemet vårt kan brukes til å svare på øvinger og til å gi tilbakemelding på dem. På startsiden får man opp sine egne øvinger i den høyre spalten (Figur 6-9). Ved å dobbeltklikke på en øving her, vil man få opp et spørsmål om å opprette et dokument som svar på øvingen (Figur 6-18). Hvis man svarer ja på dette spørsmålet, får man opp et nytt dokument i arbeidsboken med tittelen satt til øvingens tittel. Samtidig vises øvingsteksten i skoleweb-delen av hovedvinduet. Dersom man svarer nei, vil bare øvingsteksten komme opp, uten at noen dokument opprettes.



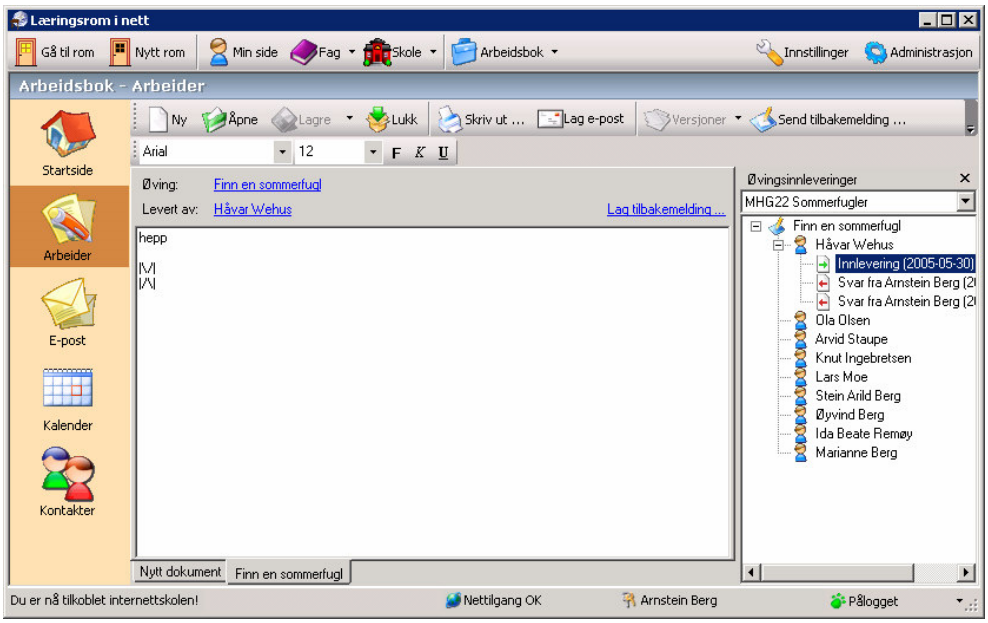
Figur 6-19 Svardokument opprettet og klart for å fylle inn besvarelse

Svardokumentet man får opp er som et helt vanlig dokument, bortsett fra at det har kategorien automatisk satt til øving. I tillegg inneholder det en referanse til øvingen det er svar på, slik at systemet vet hvor det skal leveres når man er ferdig med å svare. Man kan for øvrig lagre dokumentet og redigere det akkurat slik man vil, helt til man ønsker å levere det. Det eneste man må passe på er at dokumentet må ha en selv som forfatter. I senere utgaver av programmet vil dette bli endret slik at innleveringen er knyttet til hvem som faktisk leverer inn, og ikke til forfatter på dokumentet (som jo kan endres i ettertid).

Når man er fornøyd med svaret sitt, klikker man på ”Lever øving …” på verktøylinjen over dokumentet. Her må man bekrefte at man faktisk ønsker å levere dette dokumentet, i gjeldende utgave, som svar på øvingen. Hvis man går videre vil gjeldende versjon av dokumentet bli merket som levert og vil bli låst for videre redigering. Man kan imidlertid lage en ny versjon av dokumentet og redigere videre på denne, men dette vil ikke påvirke den innleverte versjonen.

6.2.2.7 *Rette øvinger*

Alle lærere og assistenter i et fag har anledning til å rette øvinger i dette faget. Øvinger man skal rette vil også komme opp på startsiden, eller man kan få dem opp ved å klikke på ”Øvinger”-knappen øverst til høyre i dokumenteditoren. Enten man bruker denne knappen eller dobbeltklikker på en øving på startsiden, vil man få opp panelet ”Øvingsinnleveringer” på høyre side i dokumenteditoren (Figur 6-20). I dette panelet kan man velge et fag man er lærere eller assistent for, og man får da opp alle øvingene i dette faget.



Figur 6-20 Dokumenteditoren med innleverte øvinger til høyre

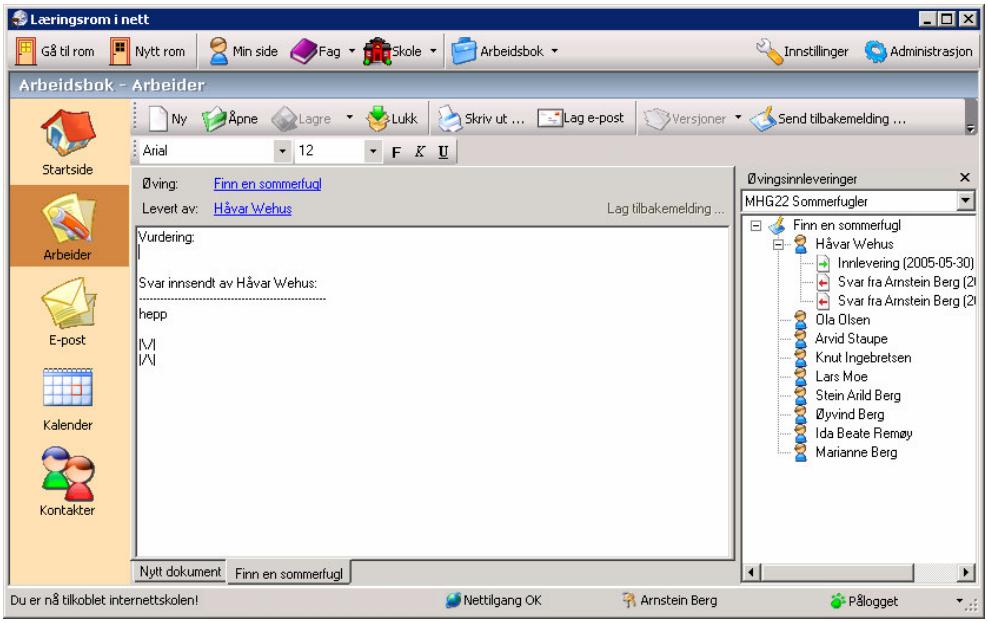
Hver øving kan utvides og under denne finner man alle elevene i faget. Videre kan man gå inn og se hva hver enkelt elev har levert inn i denne øvingen. I tillegg til å se det eleven har levert inn kan man også se det andre lærere og assistenter har gitt som tilbakemelding. På denne måten er det lett for en lærer å se hvilke elever som har svart på øvingen og hvilke som har fått tilbakemelding. Og dersom eleven må levere flere ganger, vil hele historien med innleveringer og tilbakemeldinger komme opp. Svar merkes med grønn pil mot høyre, mens tilbakemeldinger merkes med rød pil i motsatt retning.

Dersom man dobbeltklikker på en av innleveringene åpnes denne som et dokument, og læreren kan se hva eleven har svart på øvingen. Ovenfor svaret kan man også se hvem som har levert inn og hvilken øving dette er svar på. Ved å klikke på linken ”Lag tilbakemelding …” får man opp en utgave av teksten med plass til å skrive inn vurdering (Figur 6-21). Læreren kan så skrive inn en vurdering, samt sette inn kommentarer i teksten som eleven har skrevet. Selve formateringen av dette svaret vil trolig kunne være mer konfigurerbart i fremtidige utgaver.

For å sende tilbakemeldingen tilbake til eleven, klikker man på knappen ”Send tilbakemelding …” på verktøylinjen over dokumentet. Også her må man bekrefte at man vil levere tilbake teksten slik den står. Teksten man skriver på vil da bli lagt til som en ny versjon i eleven sitt svardokument, og den vil i tillegg komme opp i innleveringslisten til høyre. I denne versjonen av programmet må man lage

tilbakemeldingen og sende den i én operasjon, men i fremtidige versjoner regner vi med å legge til mulighet for at læreren kan lagre tilbakemeldingen for å jobbe med den over flere arbeidsøkter, før den til slutt sendes til eleven. Når eleven mottar tilbakemeldingen, vises den på elevens startside blant nye tilbakemeldinger. Eleven kan i ettertid også se tilbakemeldingen som en versjon under det dokumentet han eller hun leverte inn som svar på øvingen.

Som med andre dokumenter er det bare ren tekst som støttes i denne versjonen av programmet, men i fremtidige utgaver vil det være mulig å også støtte innleveringer i andre formater.

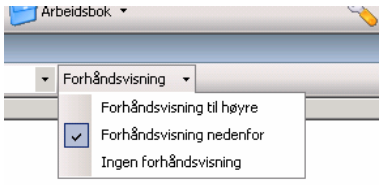


Figur 6-21 Tilbakemelding på innlevert øving

6.2.3 E-post

Det blir mer og mer vanlig at brukere i en institusjon har sin egne personlige e-postkonto tilknyttet denne institusjonen, noe som også legges opp til i dette systemet. Alle brukere kan lese sin e-post ved å velge ”E-post” fra arbeidsbokmenyen. Man får da opp en e-postleser lignende de vanligste e-postleserne som finnes på markedet (Figur 6-23).

Her vises en liste med alle e-postmeldinger du har mottatt. Uleste meldinger er merket med fet tekst. Ved å klikke på en e-post åpnes den i vinduet for forhåndsvisning. Med forhåndsvisningsmenyen (Figur 6-22) kan man velge å vise dette vinduet til høyre eller nedenfor listen med e-post. Man kan også velge å fjerne det.

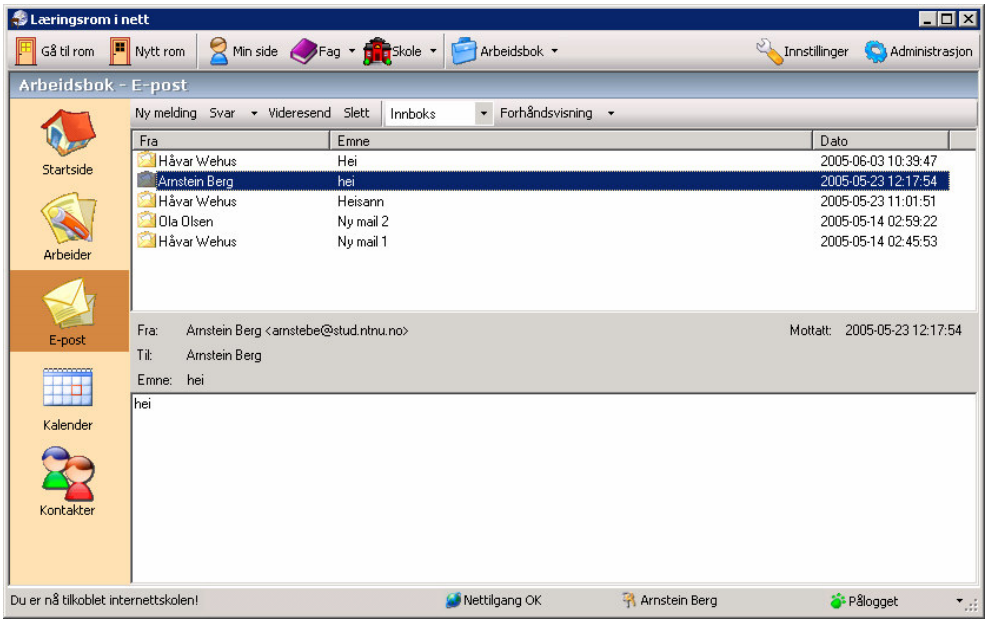


Figur 6-22 Forhåndsvisningsmeny

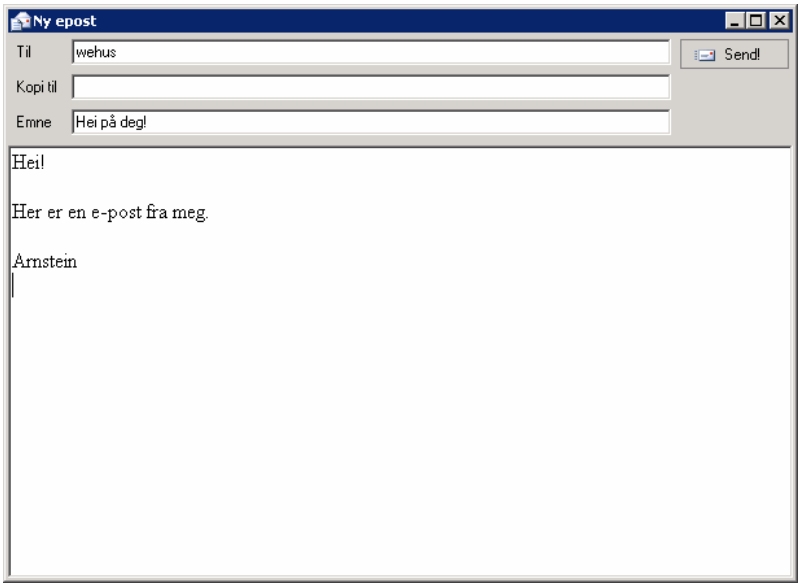
E-postleseren inneholder også funksjoner for å skrive ny e-post, svare, videresende og slette. Alle disse funksjonene er tilgjengelig fra verktøylinjen i e-postleseren. Hvis man velger å lage ny, svare eller videresende, får man opp vinduet for å skrive e-post (Figur 6-24). Dette er helt tilsvarende å velge å sende et dokument som e-post (se kapittel 6.2.2.5),

men her trenger man ikke opprette noe dokument først – man skriver bare teksten rett inn i vinduet. Når du sender e-posten blir den lagret i utboksen din. Du kan vise utboksen din ved å velge "Utboks" fra nedtrekksmenyen på verktøylinjen i e-postleseren.

E-postfunksjonaliteten i denne versjonen av systemet er ikke fullstendig implementert, og det går derfor ikke an å sende e-post ut til andre e-postadresser, og man kan ikke motta e-post fra andre utenforstående. Det er imidlertid mulig å sende e-post internt til andre brukere av systemet ved å skrive inn brukernavnet deres i feltet for mottakeradresse.



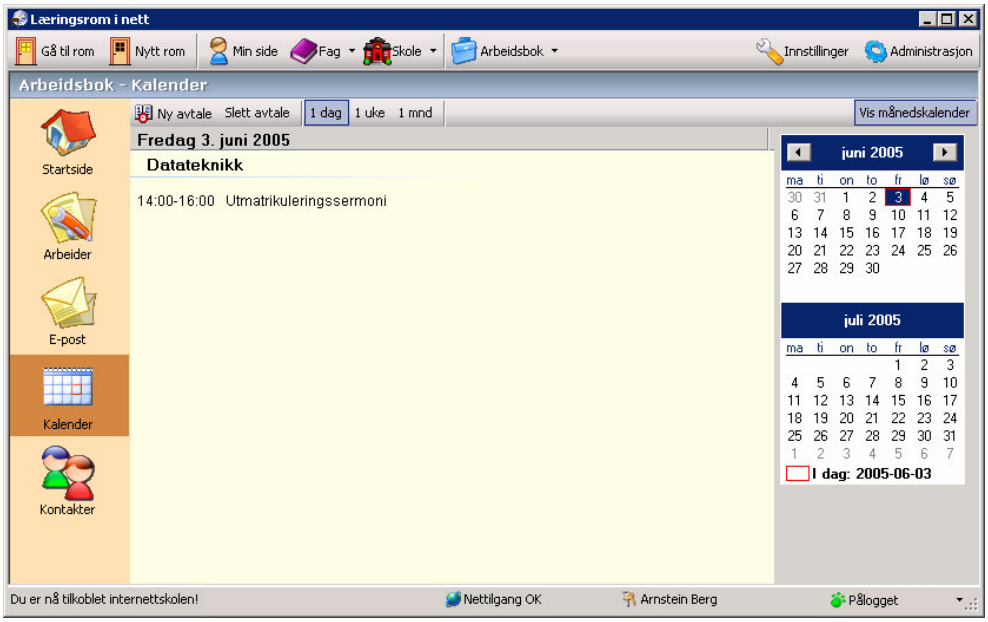
Figur 6-23 Din innboks i e-postleseren



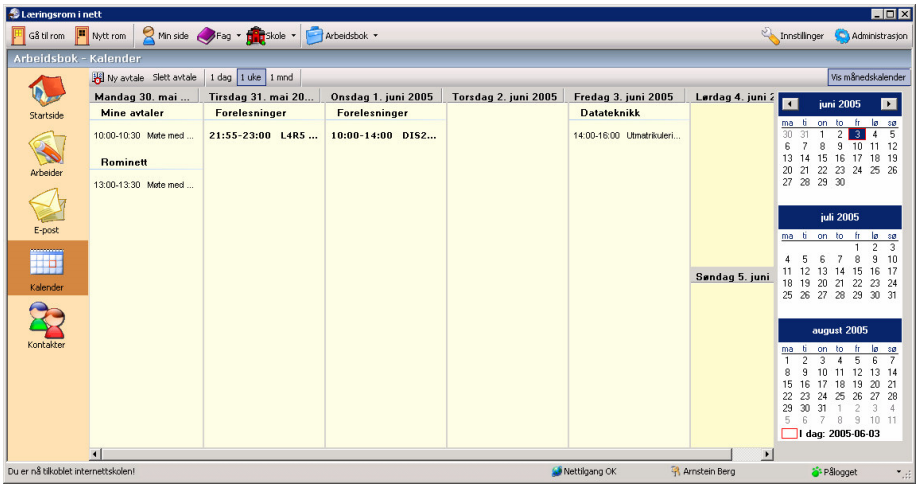
Figur 6-24 Vindu for å skrive ny e-post

6.2.4 Kalender

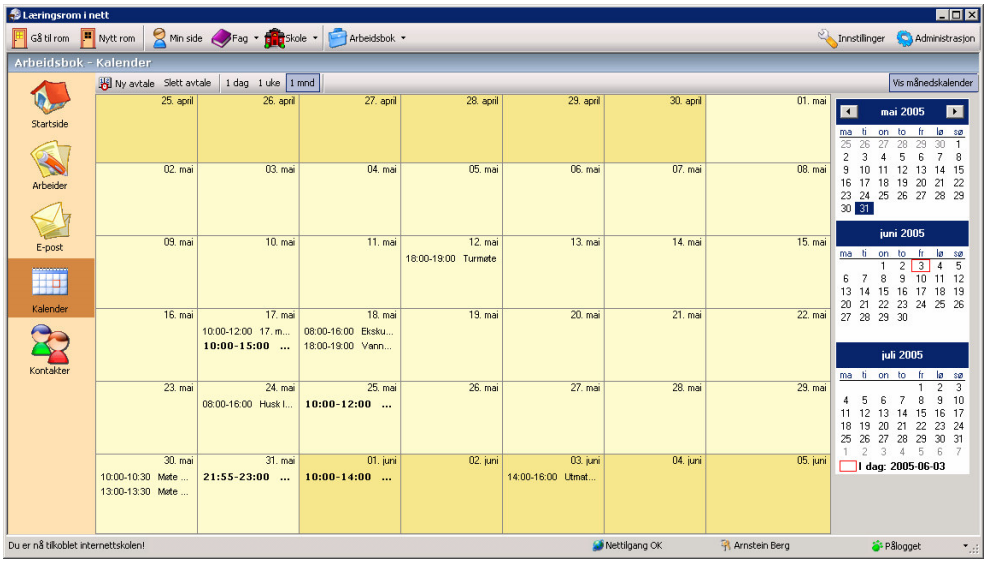
Kalenderen er tilgjengelig under ”Kalender”-knappen i skrivebordmenyen. Her finner du oversikt over alle dine avtaler og huskelapper. I tillegg til dine personlige avtaler vises også avtaler som er registrert for fag og klasser du er med i, samt forelesninger i fagene dine. Du kan velge om du vil se kalenderen for en dag (Figur 6-25), en uke (Figur 6-26) eller en måned om gangen (Figur 6-27). Man kan få opp mer informasjon om en avtale ved å dobbeltklikke på den. Da kan man også velge å redigere avtalen, hvis det er ens egen avtale eller den tilhører et fag man er lærer for.



Figur 6-25 Kalender, vis dag

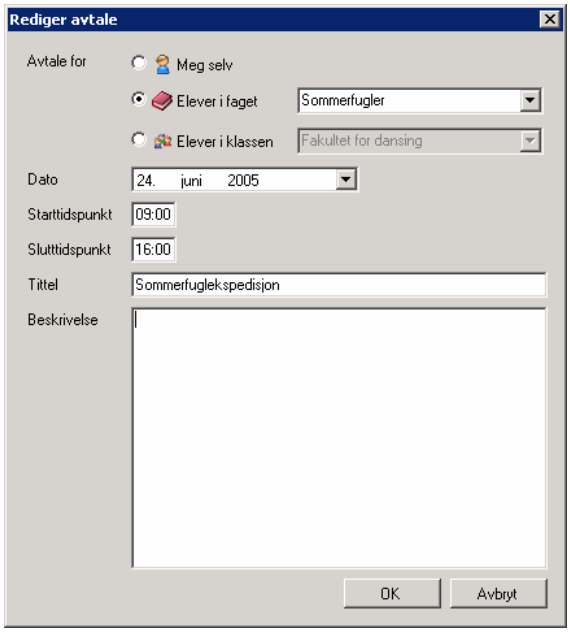


Figur 6-26 Kalender, vis uke



Figur 6-27 Kalender, vis måned

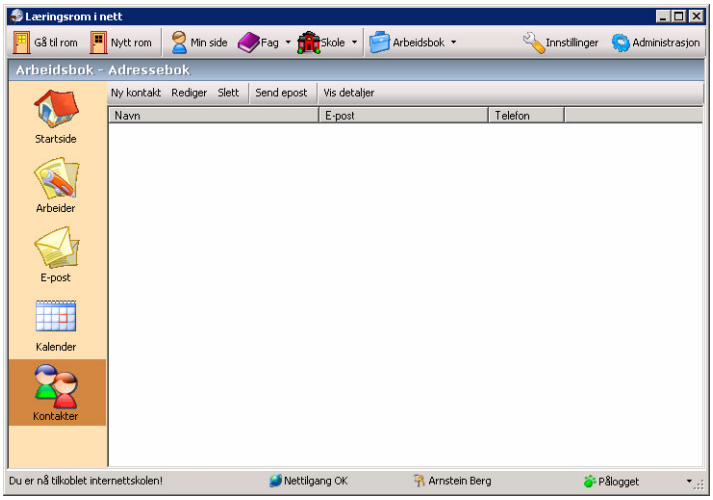
For å registrere en ny avtale klikker du på knappen "Ny avtale" på verktøylinjen i kalenderen, og du får da opp vinduet for å legge inn en ny avtale (Figur 6-28). Her kan du først angi hvem avtalen gjelder. Dersom du er lærer for et fag eller en klasse, kan du registrere avtaler for dette faget eller denne klassen. Videre kan du legge inn dato og tidspunkt, samt tittel og beskrivelse for avtalen.



Figur 6-28 Registrer ny avtale

6.2.5 Kontakter

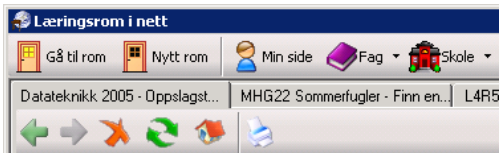
Dette er en del av skrivebordet som ikke er implementert i denne versjonen av programmet. Planen er at her kan du få oversikt over folk du kjenner, både de som er brukere i dette systemet, og andre utenforstående. Man kan lagre personlige detaljer, som adresse og telefonnummer. For andre brukere i dette systemet vil du også kunne se om de er pålogget, og i så fall klikke på dem for å åpne et praterom (Kap. 6.4.2) for å kommunisere direkte med dem.



Figur 6-29 Foreløpig utseende på kontaktoversikten

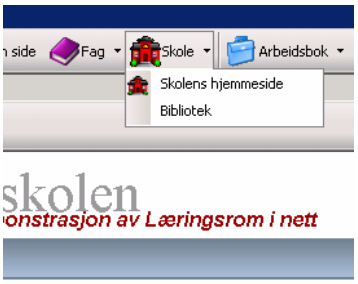
6.3 Skoleweb

Man kan få opp skolens websider i en nettleser hovedvinduet. Denne fungerer som er standard nettleser, der man kan navigere mellom linker og skrive ut en webside. I verktøylinjen er det knapper for å få tilgang til de mest viktige skolewebsidene (Figur 6-30). Websidene kan også nås med en hvilken som helst annen nettleser.



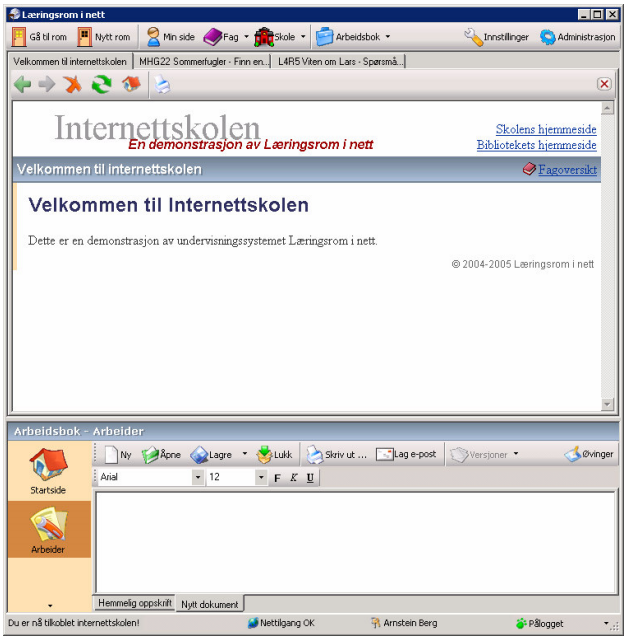
Figur 6-30 Snarveier til skolens websider

6.3.1 Skolens side



Figur 6-31 En måte å komme til skolens hjemmeside er gjennom skolemenyen i hovedvinduet

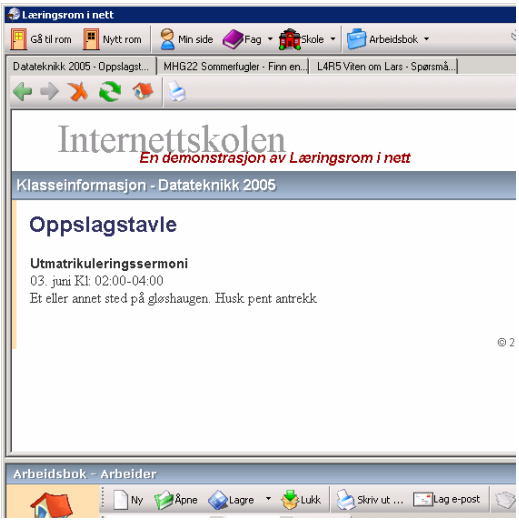
I denne begrensede utgaven av programmet er det ikke så mye informasjon på forsiden til skolens webside(Figur 6-32). Når programmet blir tatt i bruk av en virkelig skole vil det kunne være nyheter og generell informasjon om skolen på siden. Det kan også være en oppslagstavle og et diskusjonsforum for brukerne av systemet på siden.



Figur 6-32 Forsiden til skolens webside

6.3.2 Klassens side

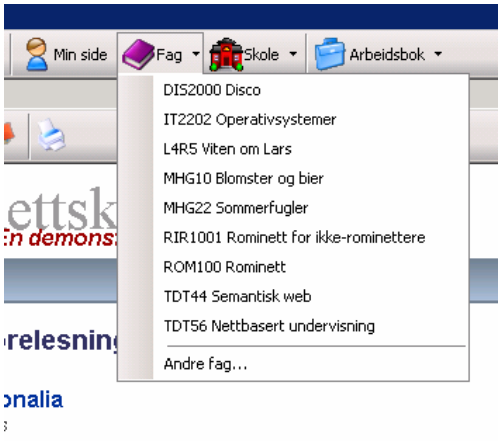
6.3.2.1 Oppslagstavle



Figur 6-33 Klassens oppslagstavle

Hver klasse har sin egen hjemmeside. I denne versjonen er det bare en oppslagstavle for klassen på siden. I kommende versjoner vil det på denne siden kunne være generell informasjon om klassen og diskusjonsforum hvis dette er ønskelig. Her kan det også være en oversikt over hvilke fag klassen har, og hvilke av disse fagene som er obligatoriske.

6.3.3 Fagets side



Man kan åpne et fags hjemmeside fra fagmenyen i hovedvinduet (Figur 6-34). På fagets hjemmeside er det en meny på venstre side(Figur 6-35). Her er det linker til oppslagstavle, informasjon om faget, forelesninger, pensum, øvinger og fagets rom. I tillegg kan det være egendefinerte linker og websider hvis en foreleser i et fag har lagt til dette.

Figur 6-34 Meny med alle fagene en bruker har

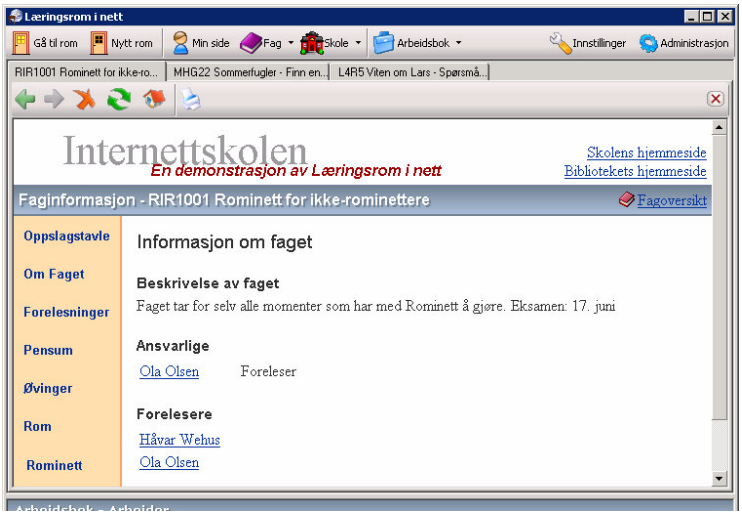
6.3.3.1 Oppslagstavle

På forsiden til et fag er det en oppslagstavle der viktige beskjeder kan henges opp (Figur 6-35). Det er lærere og assistenter i et fag som kan skrive beskjeder på oppslagstavlen. På fagsiden skal det også være et diskusjonsforum der elever i faget har tilgang.

Figur 6-35 Fagets oppslagstavle

6.3.3.2 *Faginformasjon*

Faget har også en side der det er generell informasjon om faget (Figur 6-36). Her er det en kort beskrivelse av faget. Også ansvarlige personer og forelesere er listet opp. Hvis man trykker på navnet vil man få opp et vindu for å sende e-post til vedkommende (Figur 6-24).



Figur 6-36 Informasjon om faget

6.3.3.3 *Forelesninger*

Figur 6-37 Liste med forelesninger

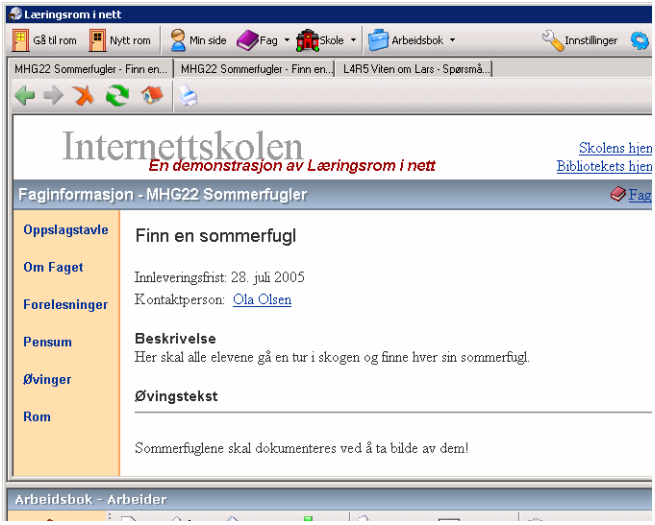
Her er det en liste med de forelesningene som er i faget (Figur 6-37). Hvis man trykker på forelesningsnavnet kan man se detaljer om forelesningen. I tillegg til å se hva det skal foreleses i, tid og sted for forelesningen og hvem som skal forelese, kan man her se spørsmål som har blitt stilt til forelesningen. Man skal også kunne finne forelesningsnotater og eventuelt video fra en forelesning.

6.3.3.4 *Pensum*

Pensum er ikke implementert i denne versjonen.

6.3.3.5 *Øvinger*

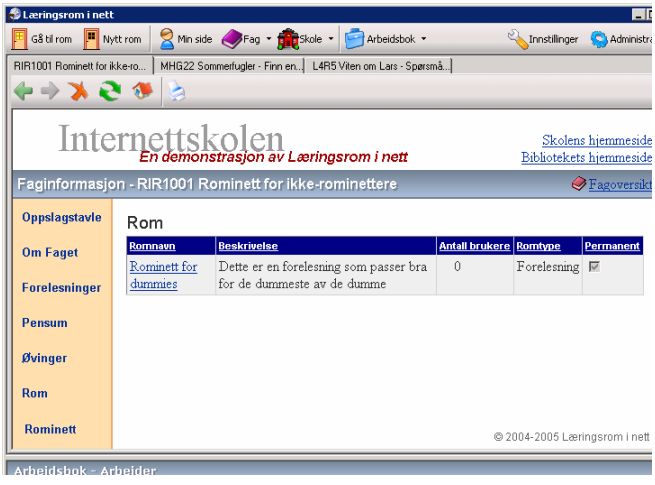
Hvis man trykker på øvinger i menyen får man opp en liste med fagets øvinger (Figur 6-38). Når man velger en av disse får man opp detaljer om denne øvinger. Her kan man se innleveringsfrist, kontaktperson og selve øvingsteksten. Denne siden kommer opp automatisk når en bruker velger å svare på en øving.



Figur 6-38 Øvinger

6.3.3.6 *Rom*

Her kan man se alle rommene som er i et fag (Figur 6-39). Hvis man klikker på rommet fra klientprogrammet vil man automatisk komme inn i rommet. Hvis man derimot besøker siden fra en annen nettleser, vil man få beskjed om å laste ned klientprogrammet for å kunne gå inn i rommet. Man vil i fremtidige versjoner av programmet kunne få spørsmål om å åpne programmet hvis dette er allerede er installert på maskinen.



Figur 6-39 Liste med rom

6.3.3.7 *Egendefinerte sider*

Hvis en lærer ønsker å lage egne html-sider for et fag kommer disse opp i menyen til venstre på fagsiden.

6.3.3.8 *Linker*

Det kan være både interne og eksterne linker knyttet til et fag. Interne linker vil bli listet som punkter i menyen. Dette kan være linker til andre hjemmesider som faglærer

vil bruke i tillegg til, eller i stedet for de genererte. Hvis det legges inn linker til eksterne sider på nettet, vil disse bli vist i en liste over linker på en ny side.

6.3.4 Andre sider

6.3.4.1 Fagoversikt

Fagoversikten har alle fagene på skolen (Figur 6-40). Hvis en bruker ønsker å se informasjon om et fag han ikke har, kan han finne det her. Man kan komme til siden fra hovedvinduet i klientprogrammet eller øverst til venstre på hjemmesiden. I denne versjonen av programmet får alle som besøker en fagside tilgang til samme informasjon. I kommende versjoner vil man bare få et begrenset innsyn på fagsiden hvis man ikke selv har faget.

Figur 6-40 Liste med fag

6.3.4.2 Bibliotekets side

Hvis skolen har et bibliotek, kan det knyttes til skolens hjemmeside. Biblioteket er tilgjengelig fra klientens hovedvindu eller fra øverst til venstre på hjemmesiden.

6.4 Samarbeidsrom

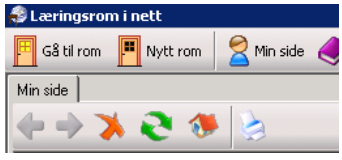
Samarbeidsrommene er en sentral del av systemet, og den mest essensielle delen av samarbeidslæringen. Det er her man kan møte andre personer, gjøre arbeidsoppgaver i samarbeid med andre, og ellers dele løst og fast av informasjon.

6.4.1 Oppsett av rom

Rom er som sagt en essensiell funksjon når man skal kommunisere med andre mennesker i systemet vårt. Disse funksjonene er derfor gjort lett tilgjengelige øverst til venstre i hovedvinduet (Figur 6-41). Her kan man velge å gå til et eksisterende rom, eller man kan opprette et nytt.

6.4.1.1 Lag et rom

Alle brukere av systemet har tilgang til å lage rom. Dette henger sammen med tanken om at rom er en nærmest ubegrenset ressurs, og helt essensielt i kommunikasjonen i systemet. Det vil imidlertid være en viss forskjell på hvilke typer rom som er tilgjengelige og hvilke innstillinger de har for hver bruker.

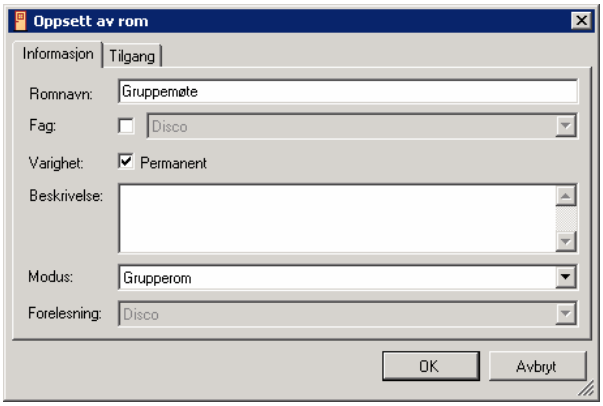


Figur 6-41 Snarvei til rom

For å lage et nytt rom klikker man på knappen ”Nytt rom” øverst i hovedvinduet, og man får da opp rominnstillingsvinduet (Figur 6-42). Her går det an å legge inn navn og

beskrivelse for rommet. I tillegg kan man angi romtypen (se kapittel 6.4.2) og om rommet er permanent eller ikke. Rom som ikke er permanente vil automatisk bli slettet av systemet når den siste brukeren forlater rommet. Se for øvrig kapittel 7.1.1.

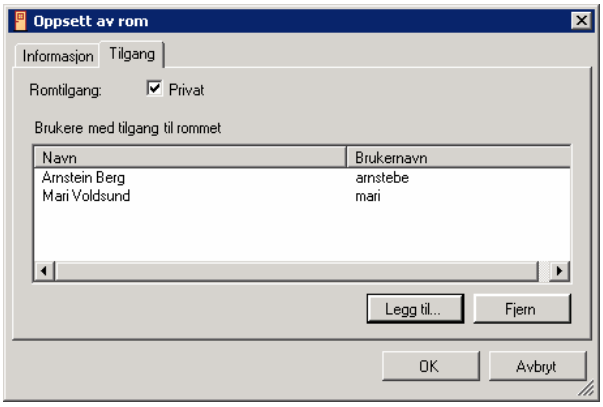
Det er også mulig å angi fag og forelesning som rommet tilhører. Å knytte et rom til et fag innebærer ikke noe annet enn at rommet vil bli listet under dette faget. Å knytte rommet til en forelesning betyr imidlertid at man også angir hvem som styrer rommet. For rom med forelesning, vil det kun være foreleser som har tilgang til å skrive på tavlen og styre hvem som kan snakke. For øvrige rom har alle i rommet tilgang til tavlen, så sant ingen andre holder på å skrive på den.



Figur 6-42 Redigere rominnstillinger

6.4.1.2 Gi tilgang til rom

I utgangspunktet er rommene man oppretter tilgjengelig for alle brukere i systemet, men man kan velge å gjøre rommet privat, slik at bare enkelte brukere får tilgang. Dette gjøres ved å gå til skillearket ”Tilgang” i rominnstillingsvinduet (Figur 6-43). Her angir man først at rommet er privat, og deretter kan man legge til brukere ved å klikke på knappen ”Legg til …” nedenfor brukerlisten.



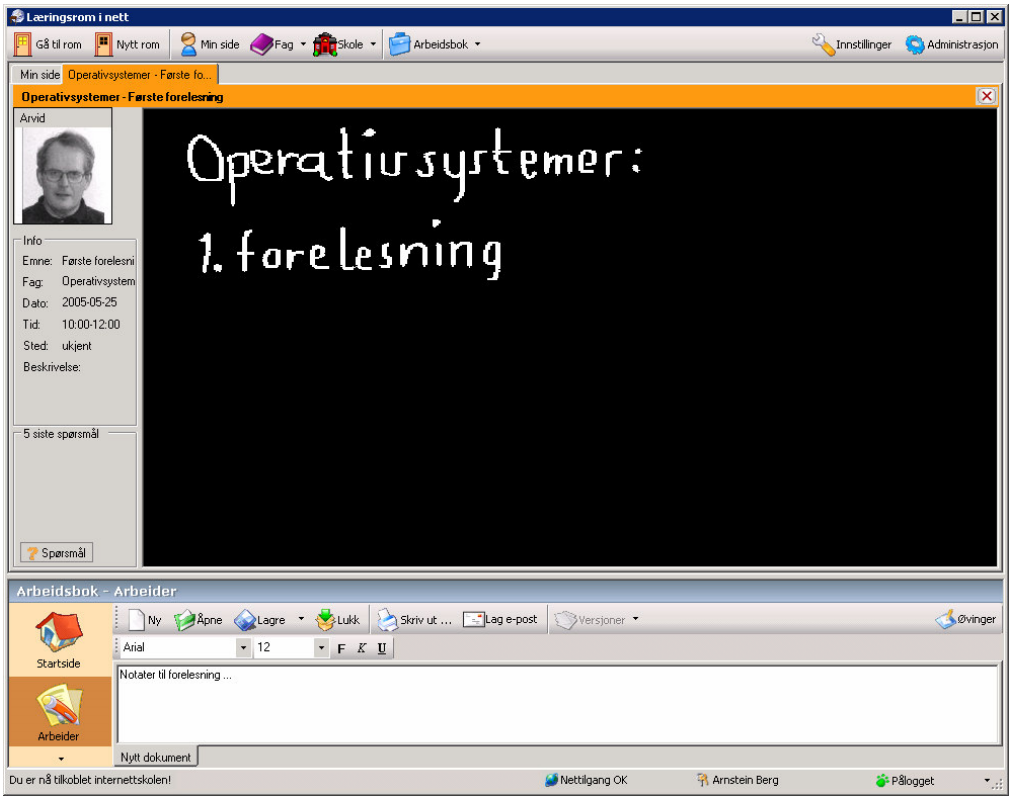
Figur 6-43 Endre romtilgang

6.4.2 Romtyper

Det finnes som tidligere nevnt forskjellige typer rom som man kan lage. Romtypene defineres av rommodi, som angir hvilke komponenter som skal være med i rommet. Brukerne kan lage egne rommodi i tillegg til de som allerede finnes. Tabell 6-1 viser rommodi som allerede er definert i systemet:

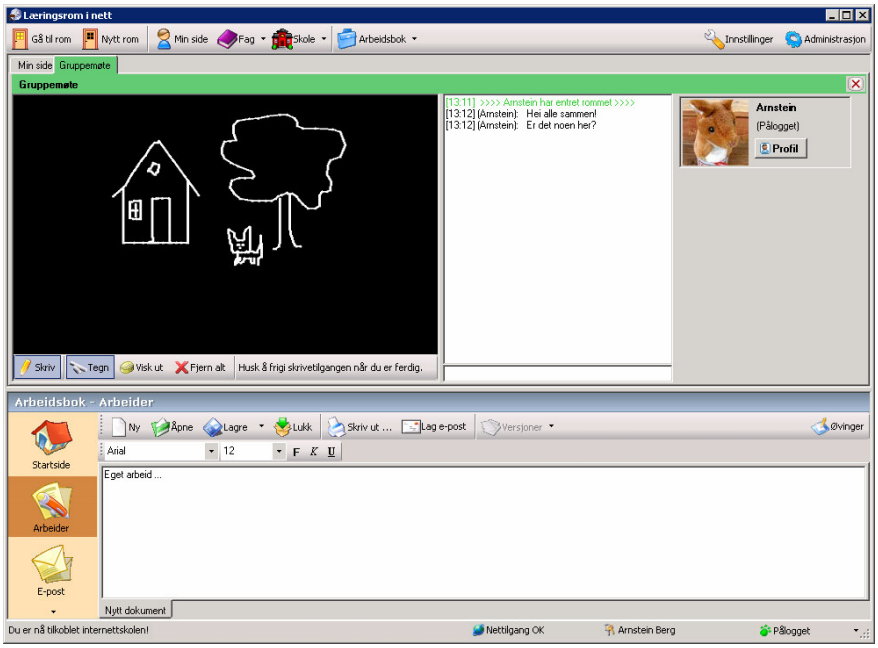
	Forelesningsrom	Grupperom	Praterom
Forelesningspanel	✓		
Brukerliste		✓	✓
Fellestavle	✓	✓	
Pratevindu		✓	✓
Multimediepanel[1]			

Tabell 6-1 Komponenter i standard rommodi

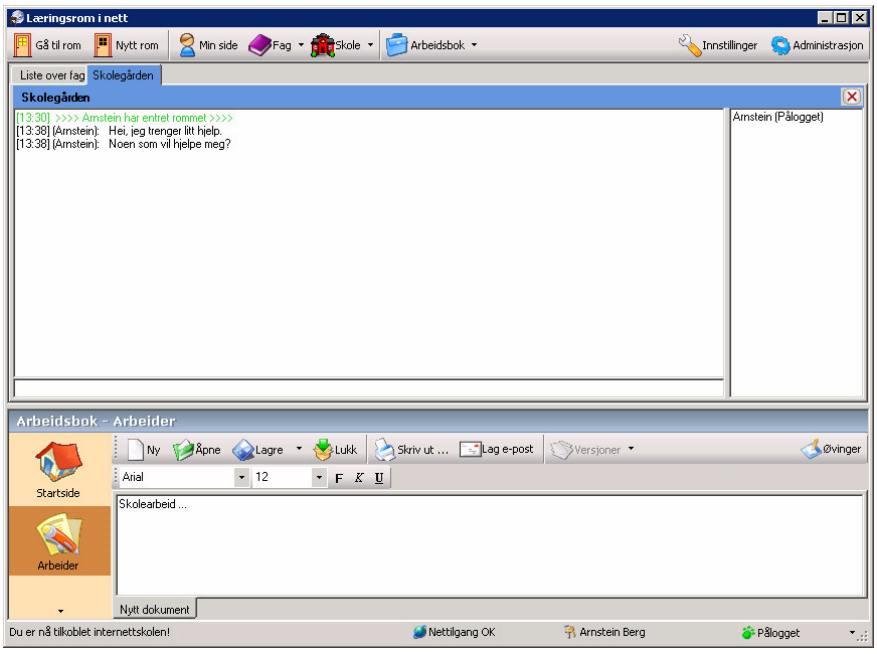


Figur 6-44 Forelesningsrom

[1] Multimediepanel er ikke implementert i denne versjonen av klientprogrammet og derfor utelatt fra rommodiene.



Figur 6-45 Grupperom



Figur 6-46 Praterom

6.4.3 Komponentene

Alle de forskjellige rommene i systemet er bygd opp etter en felles struktur, og forskjellen på dem, består i hvilke komponenter de består av. Hvilke komponenter et rom inneholder bestemmes av rommodusen til det aktuelle rommet. Foreløpig finnes

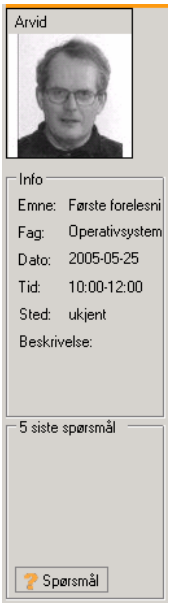
komponentene forelesningspanel, brukerliste, fellestavle, pratevindu og multimediepanel. Disse komponentene er nærmere forklart nedenfor. Senere kan det bli lagt til andre komponenter hvis ny funksjonalitet ønskes.

6.4.3.1 Forelesningspanel

(Figur 6-47) Dette er et felt i vinduet som viser informasjon om den pågående forelesningen. Dette er altså en komponent som er spesielt tiltenkt forelesningsrom. Her vises først og fremst et bilde av foreleser. I vanlige forelesninger har man ikke noen brukerliste, og derfor er det nødvendig å ha bildet her når man senere skal ha mulighet til å bruke webkamera og eventuelt direkteoverføring av lyd og bilde.

I tillegg vises generell informasjon som tittel på forelesning, fagnavn og tidspunktet for forelesningen (siden det går an å gå inn i rommet også utenom forelesningstiden).

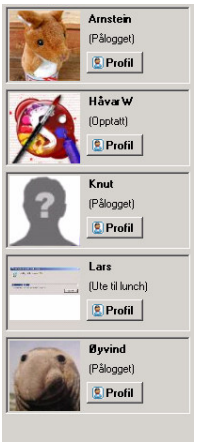
Forelesningspanelet inneholder også støtte for å sende inn spørsmål til forelesningen. I det nederste feltet i panelet vises de fem siste spørsmålene som har blitt stilt. Det er også en knapp merket ”Spørsmål”, for å åpne spørsmålsvinduet. Her kan man også stille egne spørsmål, og man kan klikke på et allerede stilt spørsmål for å se svaret på dette. Spørsmål kan stilles både før, under og etter forelesningen, og man kan også se spørsmålene på forelesningens side på skolewebben.



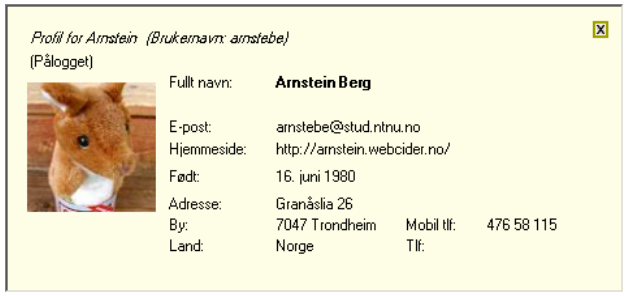
Figur 6-47 Forelesningspanel

6.4.3.2 Brukerliste

(Figur 7-48) Brukerlisten viser hvilke brukere som er i rommet. Det finnes tre måter å vise den på. Enten som en liste med ”kort” med navn og bilde (avbildet), som en liste med bare bilder, eller som en enkel liste med alle navnene under hverandre. Hver bruker kan veksle mellom disse tre visningene ved å høyreklikke på listen, og velge riktig visning fra menyen som kommer opp. I alle de tre visningene kan man klikke på brukeren sitt navn eller bilde og få opp et bilde med informasjon om denne personen (Figur 6-49).



Figur 6-48 Brukerliste



Figur 6-49 Vindu med brukerinformasjon

I ikke-private rom vil listen til enhver tid vise brukerne som er i rommet akkurat nå. I private rom vil man få opp alle brukerne som har tilgang til rommet, men de som ikke er til stede i rommet blir skyggelagt.

6.4.3.3 Fellestavle

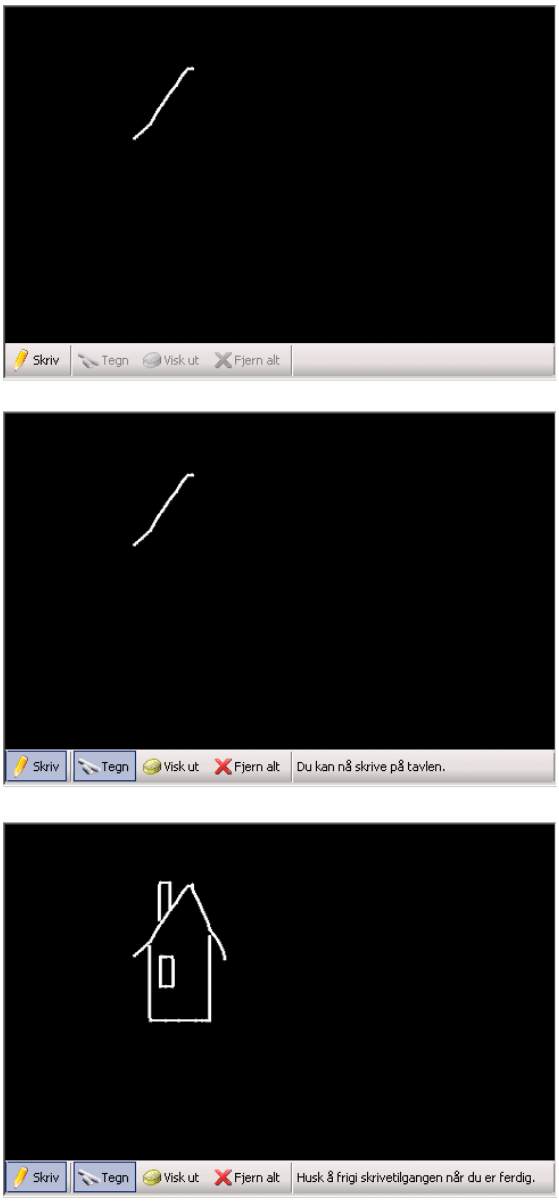
(Figur 6-50) Fellestavlen er en avansert komponent som lar brukere i rommet overføre direktesendt grafisk informasjon. I denne versjonen av programmet kan man bare tegne enkle strektegninger på tavlen, men i fremtiden kan man tenke seg å også støtte overføring av tekst og formler, så vel som video og skjermbilder fra andre programmer.

6.4.3.3.1 Tilgangskontroll

I utgangspunktet, der den viser hva andre brukere tegner på den. Dersom du er elev i en forelesning er fellestavlen i en mottakende modus der du ikke har mulighet til å tegne på tavlen. Det er her bare foreleser som kan skrive på tavlen. En elev vil bare se selve tegneområdet av tavlen, ikke verktøylinjen.

Er du derimot i et grupperom vil alle ha tilgang til å skrive på tavlen, men bare en om gangen. Når man har mulighet til å få skrive på tavlen, får man opp en verktøylinje under tavlen, der man blant annet har knappen ”Skriv”, som brukes for å be om skrivetilgang. Når man klikker på denne knappen, vil en forespørsel sendes til serveren, og dersom ingen andre skriver på tavlen da, vil man få skrivetilgang og beskjed om dette. I motsatt fall får man beskjed om at tavlen er i bruk.

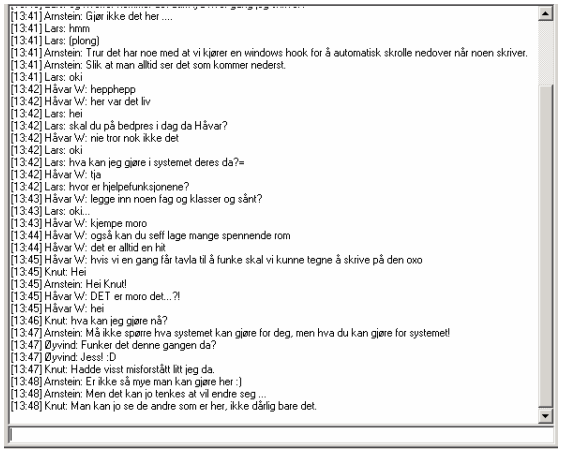
Når man har fått skrivetilgang, vil tavlen settes i en sendende modus, og det man tegner på den sendes til serveren og videre ut til de andre brukerne i rommet. Man har verktøy for å tegne, viske ut og tømme tavlen helt. Når man er ferdig med å tegne, klikker man på ”Skriv”-knappen en gang til, for å frigi skrivetilgangen og gi andre brukere tilgang til å skrive. Tavlen går da tilbake til mottakende modus.



Figur 6-50 Fellestavle. Øverst: Tavlen viser hva andre personer tegner på den. I midten: Du har fått skrivetilgang og andre kan se hva du tegner. Nederst: Når du har tegnet ferdig på tavlen, må du frigi skrivetilgangen for at andre skal kunne få skrive.

6.4.3.4 Pratevindu

(Figur 6-51) Pratevinduet er en enkel komponent for direktesending av tekst mellom brukerne i rommet. Mange personer kjenner dette fra andre programmer som ”chat”. Komponenten består av en liste med tekstmeldinger som alle brukerne i rommet kan se, og et tekstfelt nederst der man selv kan legge inn tekst i listen, slik at andre personer ser den. Alle tekstmeldingene merkes med tidspunkt og hvem som har skrevet dem.



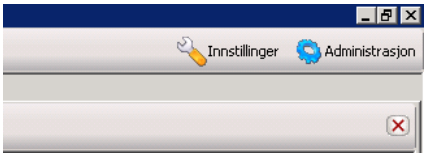
Figur 6-51 Pratevindu

6.4.3.5 Multimediepanel

(Ikke avbildet.) Dette er et panel som kun er laget for å illustrere funksjonene knyttet til direktesending av video og lyd. Siden denne funksjonen ikke finnes i gjeldende utgave av programmet, har ikke dette panelet noen funksjon ennå. Planen er at dette panelet skal inneholde volumknapp og eventuelt andre funksjoner for å velge lydinnstillinger, innstillinger for kamera osv. Det er imidlertid usikkert om dette panelet noen gang vil bli laget på denne måten, da lyd- og videofunksjoner trolig vil være en veldig sentral del av hvert rom. Man vil kanskje heller legge disse funksjonene inn i den underliggende strukturen i rommet, i stedet for å ha det i en komponent.

6.5 Administrasjonsverktøy

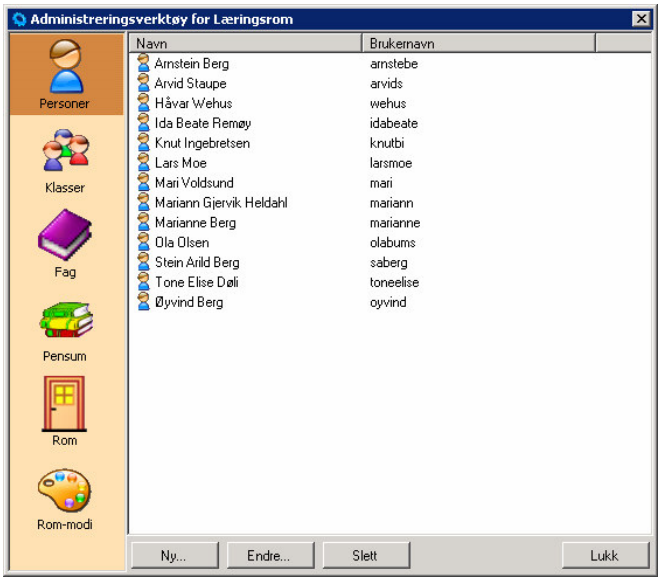
Administrasjonsverktøyet nås ved å klikke på "Administrasjon"-knappen øverst til høyre i hovedvinduet (Figur 6-52). I denne utgaven av programmet kan alle få opp dette verktøyet, men i den ferdige utgaven vil det være naturlig at bare personer som har administrative funksjoner vil få se denne knappen.



Figur 6-52 Snarvei til administrasjonsverktøy

6.5.1 Administrere personer

Når man åpner administrasjonsverktøyet og velger personer i menyen til venstre i vinduet, får man opp en liste over alle personene som er registrert i systemet (Figur 6-53). Disse personene kan både være aktive brukere (med mulighet til å logge seg på systemet), deaktiverte brukere (tidligere aktive) og personer som bare har en passiv rolle i systemet. Passive brukere kan være personer som ikke har tilgang til systemet, men som står oppført fordi de for eksempel er kontaktpersoner.



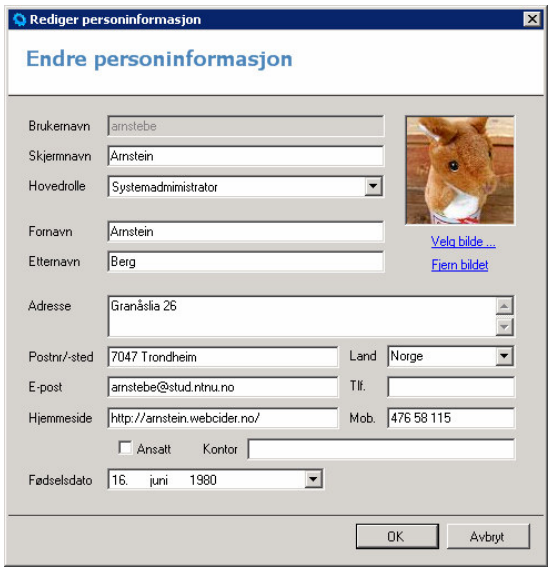
Figur 6-53 Oversikt over personer i systemet

Ved å benytte knappene nederst i vinduet kan man legge inn en ny person, redigere en merket person i lista, eller slette en eller flere merkede personer i lista.

6.5.1.1 Endre persondetaljer

Når man velger å legge inn ny person eller endre en eksisterende person i systemet, vil man få opp personinformasjonsvinduet (Figur 6-54). Her kan man registrere personalia, skjermnavn og personlig bilde (jf. Kap. 6.1.3).

Når man oppretter en ny bruker kan man også skrive inn brukernavn,



Figur 6-54 Vindu for å endre persondetaljer

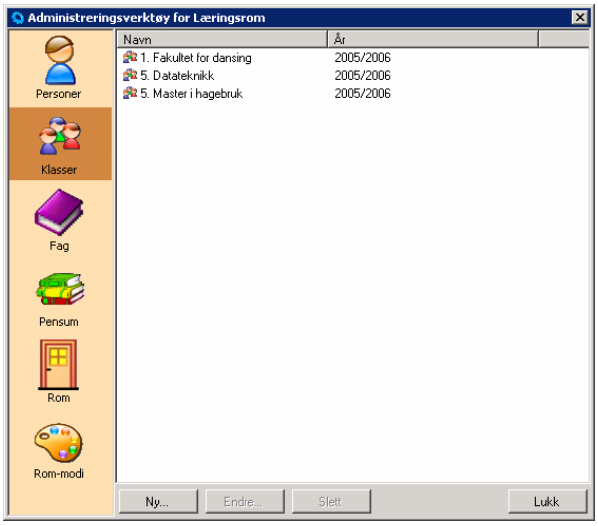
men dette kan ikke endres siden. Når en ny bruker registreres vil den tildeles et tilfeldig passord av systemet. Av sikkerhetsgrunner vil man aldri kunne få vist passordet i klientprogrammet, men man må i stedet bruke spesialprogramvare for å hente dette ut av databasen. Den aktuelle brukeren kan deretter endre passord til det han/hun måtte ønske.

Man kan også angi en person sin standardrolle i systemet. Standardrollen er den rollen en person normalt har, for eksempel lærer eller elev. Denne rollen gir ikke personen noen reell tilgang i systemet (med unntak av systemadministrator), men den vil være retningsgivende for hvordan personene vises i lister, på skolewebben, osv. Reell tilgang gis på klasse-, fag- og romnivå (se neste delkapittel), og brukeren kan her gis en annen rolle enn det som er brukerens standardrolle.

På nåværende tidspunkt finnes det ingen måte å opprette eller endre en bruker til noe annet enn å være en aktiv bruker. Dette vil bli implementert i en fullstendig utgave av programmet.

6.5.2 Administrere klasser

Dersom man går inn i administrasjonsvinduet og velger klasser fra menyen til venstre, får man opp en liste over klasser som er registrert i systemet (Figur 6-55). Klasser er i dette systemet et forholdsvis løst definert begrep, og er en persongruppe, som man kan knytte fag og andre egenskaper til. Personer som er med i en klasse vil bli listet under denne i sammenhenger der det er aktuelt, automatisk få tilgang til fag som denne klassen har.

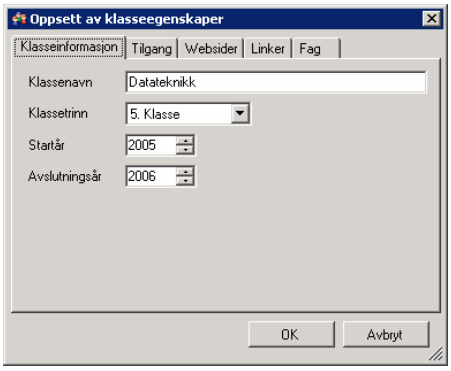


Figur 6-55 Liste over klasser som er registrert i systemet

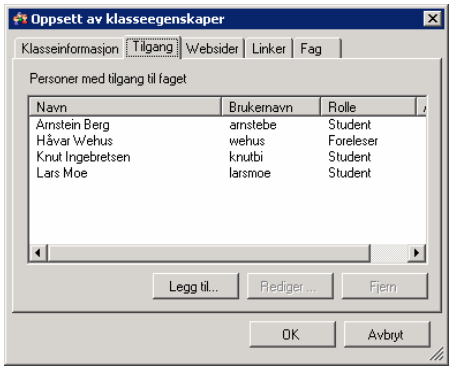
Ved å benytte knappene nederst i vinduet kan man legge inn en ny klasse, redigere en merket klasse i lista, eller slette en eller flere merkede klasser i lista.

6.5.2.1 Endre klasseinformasjon

Når man velger å opprette eller endre en klasse vil man få opp klasseinformasjonsvinduet (Figur 6-56). Her kan man angi navnet på klassen, klassetrinn og året klassen starter og slutter.



Figur 6-56 Redigere klasseinformasjon

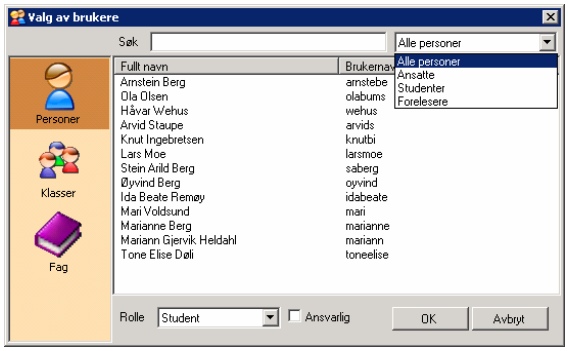


Figur 6-57 Personers rolle i klassen

6.5.2.2 *Personer i klasse*

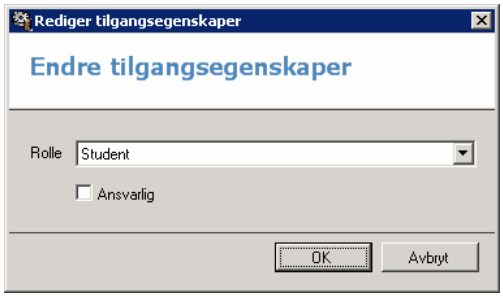
Ved å velge skillearket ”Tilgang” i vinduet for å redigere klasseinformasjon, vil man få opp en liste over hvilke personer som er med i klassen og hvilken rolle de har i denne sammenhengen (Figur 6-57). Ved å bruke knappene nedenfor lista kan man legge til nye personer i klassen, endre rollen til en person eller fjerne personer fra klassen.

For å legge til personer i klassen, klikker man ”Legg til …” og får da opp et vindu der man kan velge hvilke personer man vil gi tilgang (Figur 6-58). Her kan man velge en eller flere personer fra de personene som er registret i systemet. Det er mulig å søke seg fram til riktig person ved å skrive inn hele eller deler av navnet i søkefeltet på toppen, og/eller velge personens standardrolle i menyen øverst til høyre. Videre går det an å velge hele klasser og fag, ved å klikke på ikonene i menyen til venstre i vinduet. Dette vil føre til at alle personene i en klasse eller et fag, blir lagt til i denne klassen. Nederst i vinduet kan man dessuten velge hvilken rolle de nye personene skal ha i klassen.



Figur 6-58 Velg brukere du vil gi tilgang til klassen

Det er mulig å endre en person sin rolle i klassen etterpå, ved å velge denne personen i tilgangsvinduet (Figur 6-57) og klikke ”Rediger …”. Da vil vinduet for å endre rolle komme opp (Figur 6-59).



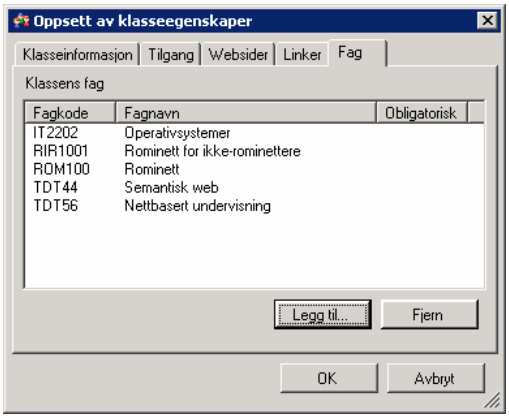
Figur 6-59 Rediger en persons tilgang

I gjeldende utgave av systemet begrenser ikke rollene tilgangen på annen måte enn at det bare er lærere og assistenter som kan rette øvinger i et fag. Når en mer fullstendig tilgangsstyring er på plass i systemet, er planen at kun lærere skal ha tilgang til å endre informasjonen tilknyttet klassen, og at kun ansvarlig lærer skal kunne endre andre personers tilgang i klassen.

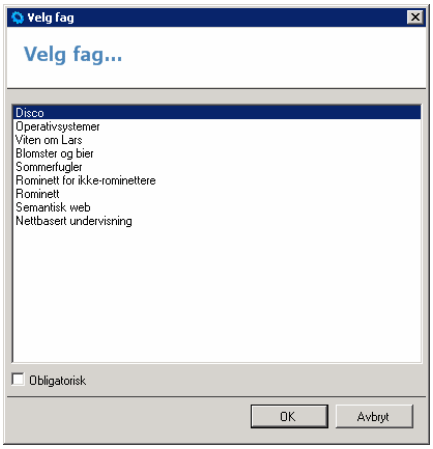
6.5.2.3 *Fag i klasse*

For å se hvilke fag som er tilknyttet en klasse kan man velge skillearket ”Fag” i klasseinformasjonsvinduet (Figur 6-60). Her kan man legge til og fjerne fag fra klassen. Dette dreier seg om koblinger til fag som allerede er registrert i systemet (se kap 0).

Ved å klikke på knappen ”Legg til …” under faglisten, får man opp et vindu der man kan velge fag man vil legge til (Figur 6-61). Her går det også an å velge om faget skal være obligatorisk for klassen. Denne funksjonen er i skrivende stund ikke implementert, men vil i framtiden fungere slik at alle elever/studenter som legges til i en klasse, vil automatisk også bli innmeldt med samme rolle i de obligatoriske fagene. For øvrige fag vil ikke personer automatisk få tilgang til et fag, selv om de er med i en klasse som har faget.



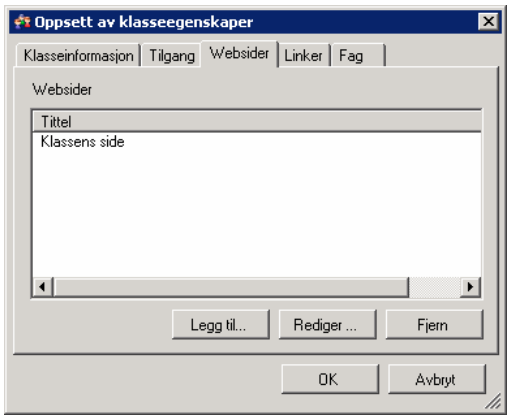
Figur 6-60 Oversikt over fag i klassen



Figur 6-61 Legg til fag i klassen

6.5.2.4 *Websider i klasse*

Ved å velge skillearket ”Websider” i klasseinformasjonsvinduet får man opp en liste over egendefinerte internettsider tilhørende klassen (Figur 6-62). Dette er sider som vil komme opp på skolewebben (Kap 6.3) i tillegg til den automatisk genererte delen.



Figur 6-62 Liste med websider i klassen

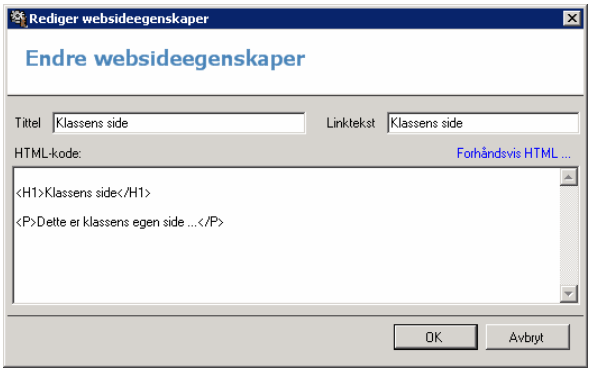
Når man velger å legge til eller endre en side som allerede finnes i lista, får man opp et vindu der man kan legge inn koden for siden (Figur 6-63). I tillegg er det mulig å spesifisere tittelen på siden, og hva som skal stå på linken i menyen.

Foreløpig kan man bare legge inn HTML-kode, men senere vil det trolig være aktuelt å legge inn en HTML-editor, som gjør at man kan redigere sidene i et miljø som ligner en vanlig tekstbehandler. Siden HTML-koden som legges inn blir plassert inne i sideoppsettet som allerede finnes på skolewebben, vil det ikke være snakk om å skrive inn all koden som skal til for å lage et HTML-dokument, men bare den delen som utgjør den synlige delen av siden. Det vil si det som står mellom

”<body …>” og ”</body>” i et vanlig HTML-dokument. Ved å klikke linken ”Forhåndsvis HTML …” vil man få opp et vindu der man kan se hvordan den koden man har skrevet inn vil se ut i nettleseren.

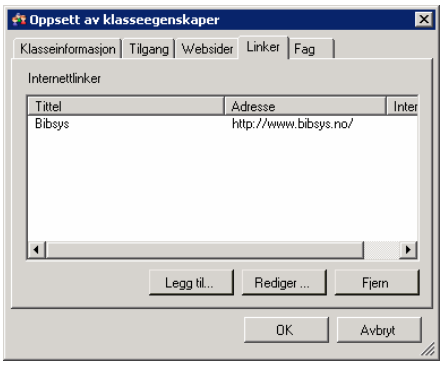
6.5.2.5 *Linker i klasse*

I forbindelse med skolewebben (Kap. 6.3) kan man også legge til egendefinerte linker til andre steder på internett. For å få opp en liste over linker som er tilknyttet klassen, klikker man på skillearket ”Linker” i klasseinformasjonsvinduet (Figur 6-64).

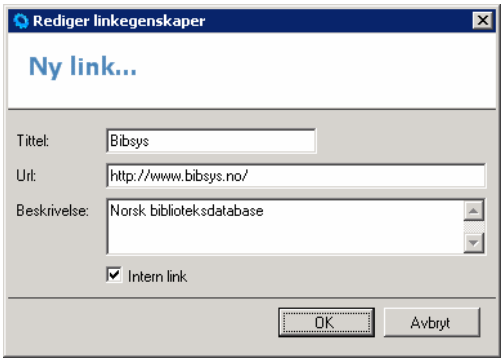


Figur 6-63 Rediger webside

Når man velger å legge til eller redigere en link, vil man få opp et vindu der man kan angi tittel (teksten på linken), url (internettadresse) og beskrivelse (Figur 6-65). Man kan også angi om linken er intern eller ekstern.



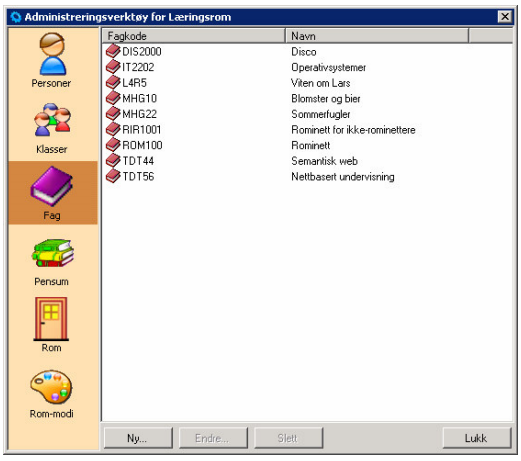
Figur 6-64 Oversikt over linker i klassen



Figur 6-65 Rediger link

6.5.3 *Administrere fag*

Ved å klikke på fagikonet i menyen til venstre i administrasjonsvinduet får man se en liste over fag som er registrert i systemet (Figur 6-66). Man kan legge til, endre eller slette fag ved å bruke knappene nederst i vinduet.

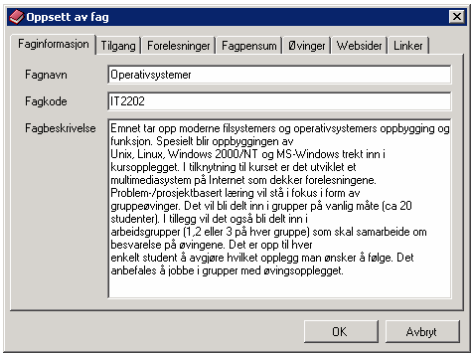


Figur 6-66 Liste over fag som er registret i systemet

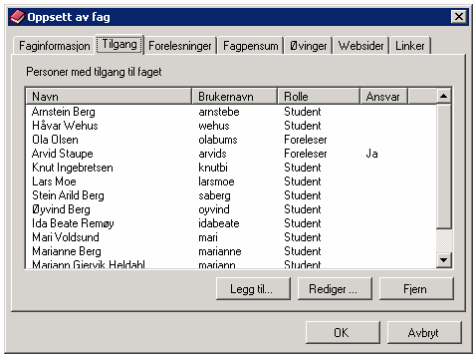
6.5.3.1 *Endre faginformasjon*

Når man velger å legge til et nytt fag eller endre et eksisterende fag, vil man få opp faginformasjonsvinduet (Figur 6-67). Her kan man skrive inn fagnavn, kode og fagbeskrivelse. I senere utgaver av programmet vil

det trolig bli mulig å legge inn flere detaljer om et fag.



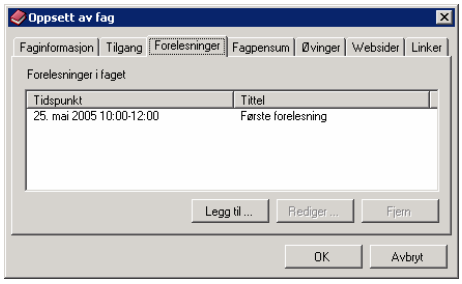
Figur 6-67 Oversikt over linker i klassen



Figur 6-68 Rediger link

6.5.3.2 *Personer i fag*

På samme måte som for fag kan man gi personer forskjellige roller i et fag. Ved å velge skillearket "Tilgang" i faginformasjonsvinduet får man opp en liste over hvilke personer som er med i faget og hvilken rolle de har. Se kapittel 6.5.2.2 for informasjon om hvordan man legger til og fjerner personer i tilgangslisten.



Figur 6-69 Liste over forelesninger i faget

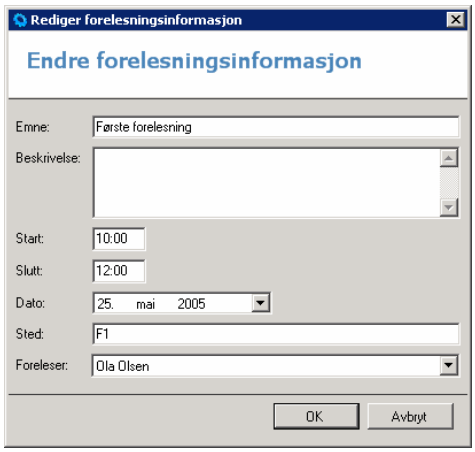
Det er verdt å merke seg at man må angi tilgang til klasser og fag individuelt. En elev får ikke automatisk tilgang til et fag, selv om denne eleven er med i en klasse som har faget. Dette er fordi det skal være mulig å ta fag uavhengig av klasser, og for valgfag er det ikke hensiktsmessig at alle elevene i en klasse får tilgang til alle fagene i klassen. I fremtidige utgaver av programmet vil imidlertid elever som meldes opp i en klasse automatisk bli oppmeldt i de obligatoriske fagene i klassen.

6.5.3.3 *Forelesninger i fag*

Under skillearket "Forelesninger" i faginformasjonsvinduet får man en liste over forelesninger som er lagt inn i faget (Figur 6-70). Det er mulig å legge til, redigere og slette forelesninger ved å benytte knappene nedenfor lista.

Når man velger å legge til eller redigere en forlesning vil man få opp et vindu der man kan angi emne, beskrivelse, tid, sted og foreleser (Figur 6-70).

I gjeldende versjon av programmet er det bare mulig å angi enkeltstående forelesninger, men i fremtiden vil det være naturlig å legge inn mulighet for å angi faste ukentlige og evt. daglige



Figur 6-70 Endre forelesningsinformasjon

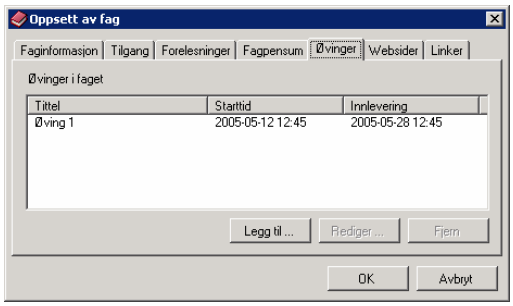
forelesninger.

6.5.3.4 Pensum i fag

Denne funksjonen er i skrivende stund ikke implementert i systemet. Her vil det bli mulighet til å knytte pensummoduler til faget (Kap 6.5.4).

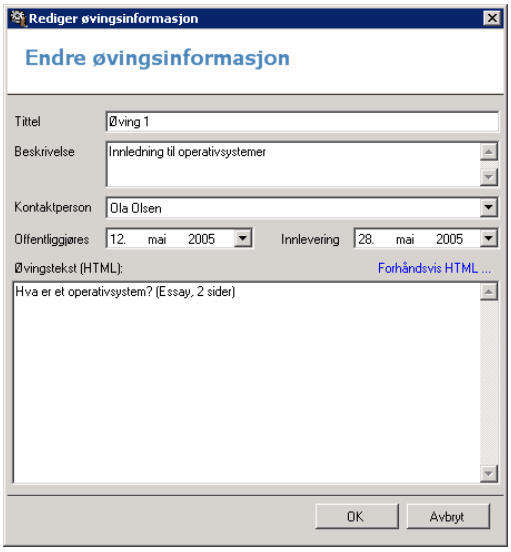
6.5.3.5 Øvinger i fag

Under skillearket "Øvinger" i faginformasjonsvinduet finner man en liste over øvingene som er registrert i dette faget (Figur 6-71). Med begrepet øvinger menes oppgaver som elevene skal eller kan gjøre, og som blir levert inn til læreren for å få tilbakemelding. For å legge til, redigere og slette en øving kan man bruke knappene nedenfor lista.



Figur 6-71 Liste over øvinger i et fag

Når man velger å opprette eller endre en øving, vil man få opp et vindu med øvingsinformasjon (Figur 6-72). Her kan man angi tittel, beskrivelse og kontaktperson for øvingen. Kontaktpersonen er den personen som vil vises som ansvarlig for øvingen, men alle lærere og assistenter i et fag kan rette og gi tilbakemelding på en innlevert øving.



Figur 6-72 Rediger øvingsinformasjon

Det kan også angis dato for offentliggjøring og innlevering av øvingen. For elever vil øvingen ikke bli synlig før på dato for offentliggjøring, og innleveringsdatoen vil bli stående som frist.

I tillegg kan man skrive inn øvingsteksten som HTML-kode. Denne funksjonen er tilsvarende funksjonen for å legge inn kode for webside i fag og klasser (se kapittel 6.5.2.4), og de samme muligheter og begrensninger gjelder her. En HTML-editor skal også legges til her i fremtiden, slik at man ikke trenger å kunne HTML for å kunne formatere en øvingstekst.

Øvingsdelen av de administrative verktøyene er bare for å definere øvinger. Innlevering, retting og tilbakemelding av øvinger skjer i forbindelse med arbeidsboken (Kap. 6.2.2).

6.5.3.6 Websider i fag

Denne funksjonen er helt tilsvarende funksjonen som finnes i klasse. Se kap 6.5.2.4.

6.5.3.7 *Linker i fag*

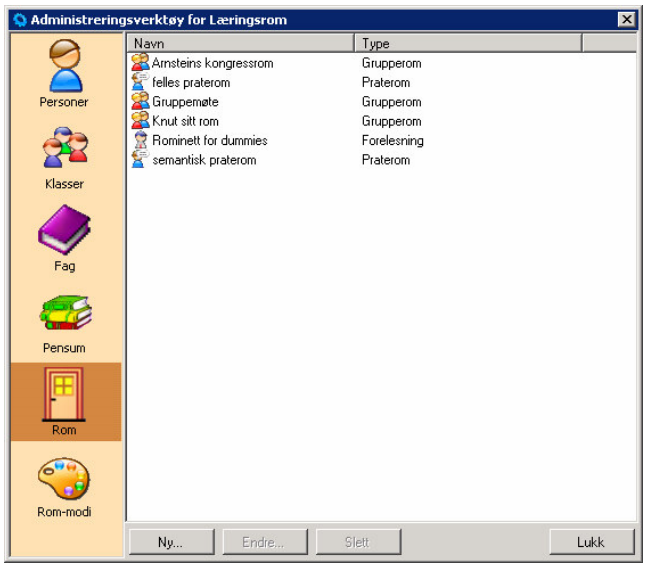
Denne funksjonen er helt tilsvarende funksjonen som finnes i klasse. Se kap 6.5.2.5.

6.5.4 Administrere pensummoduler

Denne funksjonen er i skrivende stund ikke implementert i systemet. Her vil det bli mulighet for å lage og administrere moduler med lærestoff som kan knyttes til fag og forelesninger.

6.5.5 Administrere samarbeidsrom

Ved å klikke på romikonet i menyen til venstre i administrasjonsvinduet får man se en liste over rommene som er lagt inn i systemet og hvilken type rom de er (Figur 6-73). Man kan legge til, endre eller slette rom ved å bruke knappene nederst i vinduet.



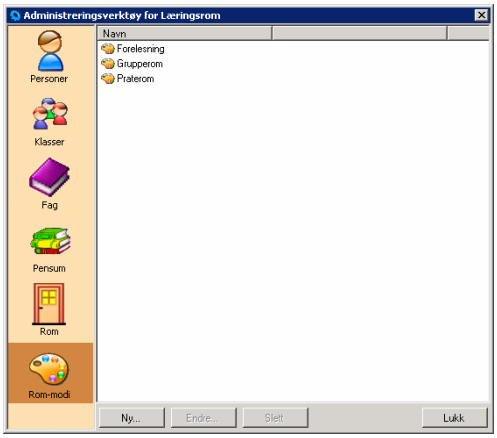
Figur 6-73 Liste over rom som er registrert i systemet

Funksjonen for å endre et rom her er helt lik den man finner ved å klikke ”Nytt rom” i hovedvinduet, men i tillegg er det her mulig å endre innstillinger i eksisterende rom. I senere versjoner vil det være naturlig å begrense noen av rominnstillingene for vanlige personer, og bare la personer med administrativ tilgang få full tilgang. Det vil med andre ord bli forskjell på hva man kan velge i rominnstillingsvinduet når man åpner det her i administrasjonsverktøy i forhold til om man åpner det direkte fra hovedvinduet.

6.5.6 Administrere rommodi

Ved å klikke på rommodi-ikonet i menyen til venstre i administrasjonsvinduet får man se en liste over rommodi som er registrert i systemet (Figur 6-74). Man kan legge til, endre eller slette rommodi ved å bruke knappene nederst i vinduet.

Se kapittel (Kap. 6.4.2) for mer informasjon om hva rommodi er, og hva de brukes til.

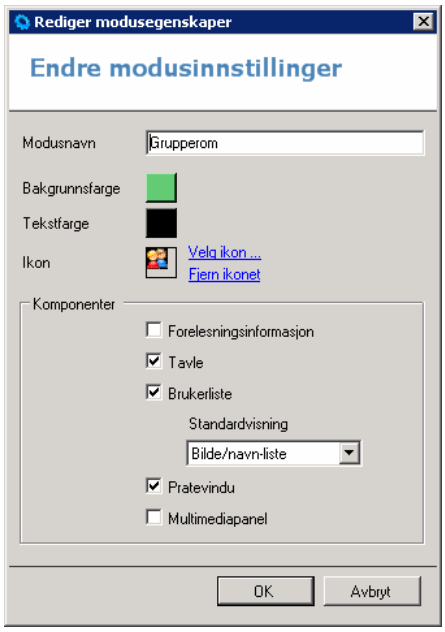


Figur 6-74 Liste over rommodi i systemet

6.5.6.1 *Redigere rommodusdetaljer*

Når man velger å opprette eller redigere en rommodus, får man opp et vindu med

innstillinger for rommodus (Figur 6-75). Her kan man angi modusnavn og fargene som kjennetegner denne modusen. I denne versjonen blir disse fargene brukt på skillearkene til de forskjellige rommene med denne modusen. I senere versjoner kan det også være aktuelt å bruke fargene enda mer tydelig for å identifisere romtypen. Det går også an å spesifisere ikon for modusen. Dette er ikke implementert eller i programmet ennå, men er ment å brukes i lister og på skillearkene til rommene.



Figur 6-75 Rommodusinnstillinger

Hvert rom er bygd opp av komponenter og i en rommodus angir man hvilke av disse komponentene som skal vises. Komponentene man kan velge å inkludere eller ekskludere i denne typen rom er

- Forelesningspanel
- Fellestavle
- Brukerliste
- Pratevindu
- Multimediepanel

Se kapittel 6.4.3 for mer informasjon om romkomponentene.

Diskusjon

7 Status og videre utvikling

7.1 Komponenter i programmet

7.1.1 Samarbeidsrom

Man må i gens versjon velge modus for hvert enkelt rom. Tanken med modus var at vi enkelt skulle kunne skille mellom ulike typer rom. Under utvikling av programmet fant vi ut at dette har gitt flere begrensninger enn det har vært til nytte. Noen av personene som har prøvd programmet, skjønte ikke helt hva rommodus var. I mange sammenhenger ønsker man å forandre på romegenskaper mens man er i rommet. Dette er det liten mulighet til i dagens system. Ved å kunne ha et standard oppsett for hvordan et rom skal se ut ved oppstart, vil man fortsatt kunne ha forskjellige utseende på rommene, men hver enkelt bruker vil kunne velge hvilke komponenter de vil se mens de er i rommet.

Rommene kan være tungvinne å opprette. Det er mange innstillingsmuligheter som man i mange tilfeller ikke har nytte av. Dette gjelder spesielt når elever skal opprette egne rom. For det første er det liten vits i at en elev kan rom som er tilgjengelig for alle. I tillegg bør det være muligheter for en lærer å velge hvilke komponenter som skal være mulig å velge mellom for en bruker som er i rommet. På denne måten kan læreren forenkle funksjonaliteten til et rom, og dermed gjøre det lettere å bruke for mindre erfarne brukere.

Hvis en bruker vil kommunisere med en annen bruker, skal han kunne ha muligheten til å velge brukeren fra en liste, for så å kunne velge å opprette et nytt rom med den han vil prate med. Hvis det er ønskelig at andre skal få tilgang til rommet, kan disse inviteres i ettertid. På denne måten blir det enklere å ha kontakt med andre brukere av systemet, og man slipper mange unødvendige innstillinger hvis man bare vil prate med noen.

Når en bruker blir invitert til et rom, vil det bli sendt en melding til denne brukeren. Han kan da velge å gå til dette rommet. Hvordan meldingen blir vist, vil avhenge av statusen til brukeren, slik at han ikke blir forstyrret hvis han ikke vil. Det bør kunne sendes med en kommentar til invitasjonen. Dette kan også brukes hvis en gruppe har spørsmål til en lærer, og vil at han skal kommet til rommet de sitter i.

Læreren bør også kunne opptre som en observatør uten å forstyrre gruppen, men han vil kunne ha overordnet tilgang til fellestavlen og mikrofonen. Dette gjelder spesielt hvis det bare er en som har kontroll over disse om gangen. På denne måten kan læreren hjelpe hvis det er behov for det, og han kan også observere og evaluere elevene i gruppen uten å blande seg.

Hvis man vil lage et forelesningsrom i dagens løsning må man første opprette en ny forelesning, og så kan man opprette et nytt rom med denne forelesningen. En bedre løsning her vil være å la brukeren som oppretter forelesningen velge om han vil opprette et rom sammen med forelesningen.

Det skal være mulig å lagre en forelesning, slik at andre som ikke har vært tilstede vil kunne få mulighet til å se et opptak i ettertid. Man bør kunne ha mulighet til å følge med på en forelesning med lyd eller kun tekst i tillegg til eventuelle lysark. For brukere med treg internettforbindelse vil det være mest hensiktsmessig å kunne lese det foreleser sier i stedet for å ha lyd, siden lyd krever mye større linjekapasitet.

I prototypen vår er det en begrenset funksjonalitet for spørsmålsstilling i en forelesning. I det ferdige programmet vil det i spørsmålsvinduet være mulig å velge tema for spørsmålet, og tidspunktet spørsmålet ble stilt vil bli lagret. Slik vil det være

mulig å sortere etter emner eller tidspunktet det ble stilt.

Det kan i mange tilfeller være interessant å kunne gå inn på et rom i ettertid og se hva som har blitt sagt og vist i rommet tidligere. I gjeldende versjon av programmet er det kun tekstmeldinger i pratevinduet som blir husket for ettertiden. Det finnes imidlertid ingen funksjon for å gå tilbake og se gamle tekstmeldinger, men dette vil helt klart komme i en senere versjon. Videre vil det være naturlig å kunne lagre ting som har blitt vist på tavlen, og gjerne også opptak av video- og lydoverføring. En funksjon som også kan bli lagt til er muligheten til å laste ned "opptak" av aktivitetene i rommet via skolewebben, som video, lyd, bildeserie eller tekstlogg.

7.1.2 Fellestavle

Fellestavlen kan nå kun tegnes på av en person om gangen, og det er bare mulig å tegne med en hvit penn på en svart tavle. Det skal bli mulig med utvidede tegnemuligheter. En av mulighetene her er å gi fellestavlen lik funksjonalitet som et enkelt tegneprogram, der man i tillegg til å tegne streker i ulike farger kan tegne ferdig definerte figurer.

En viktig utbedring er å legge til funksjonalitet for å vise andre programmer på fellestavlen. Da vil man lett kunne holde forelesninger med lysark, vise rene tekstdokumenter eller video. En utfordring her vil være å få overføringen til klientene rask nok. Spesielt gjelder dette hvis man skal vise video.

En annen funksjon som vil være praktisk, vil være å kunne vise et bilde av sin personlige arbeidsbok i tavlen. I et grupperom vil det være nyttig å kunne vise sitt eget arbeid til resten av gruppen. I grupperommet vil det også være praktisk at flere brukere kan tegne og skrive samtidig på samme tavle. Dette vil være enklere å bruke siden man da slipper å vente på å få tilgang til å skrive på tavlen. Det er imidlertid snakk om en teknisk utfordring å få oppdatert kontinuerlig begge veier, når flere tegner samtidig. I forelesningen er det ønskelig at kun en har kontroll over tavlen. Det kan imidlertid være muligheter for å la andre få tilgang hvis for eksempel noen av tilhørerne skal få skrive på tavlen. Det er også aktuelt å legge til logging av innholdet på tavlen.

7.1.3 Multimedia

Dagens løsning har ikke støtte for synkron overføring av lyd eller video. En slik utvidelse vil være viktig for å kunne utnytte et samarbeidsprogram fullt ut. Bruk av lyd vil være sentral i en forelesning der en foreleser taler til en forsamling og det kan være aktuelt i sammenheng med videovisning. I et grupperom vil det være viktig med lyd for å kunne diskutere sammen.

For møter der det deltar forholdsvis få personer, vil det være mest hensiktsmessig med en peer-to-peer-løsning. Det vil si at kommunikasjonen går direkte fra en klientmaskin til en annen. Dette vil være en raskere overføring slik at det blir liten forsinkelse på lyden, og det vil bli lettere å føre samtaler og diskusjoner. Peer-to-peer-overføring krever mye av en klientmaskin, siden man må sende all utgående kommunikasjon til alle de andre brukerne.

I større grupper er det hensiktsmessig at det er mer kontroll med hvem som snakker til enhver tid. Dette er forenelig med teknologien, siden man da vil benytte en serverbasert løsning der serveren tar imot lyden fra den som snakker, og sender den videre ut til alle som hører på. I en slik løsning vil det kreves at de som vil si noe må be om, og deretter få ordet. Ved mange brukere krever dette mye av serveren, men servere er tradisjonelt sett raskere og har bedre båndbredde enn vanlige klientmaskiner. Det vil også være en fordel i forhold til at en klientmaskin som sender lyd ikke vil bli unødvendig belastet hvis det er mange brukere som skal høre på. Det

bør være mulig å automatisk eller manuelt stille på lydkvaliteten. Hvis det settes automatisk, vil kvaliteten kunne bli stilt ned når det er mange brukere tilstede for å få mindre forsinkelse.

Ved synkron videooverføring vil man komme bort i mange av de samme problemene som ved lydoverføring. I et grupperom vil det være aktuelt med webkamera for hver av brukerne mens i en forelesning vil det være et en-til-mange-forhold der foreleseren har et webkamera rettet mot seg. I et en-til-mange-forhold vil det være aktuelt med kringkasting eller video på forespørsel. Slike videooverføringer kan løses med Windows Media Services som finnes i Windows Server 2003. Kringkasting kan brukes i visning av en forelesning som skal sendes på et bestemt tidspunkt. Forelesninger som det har blitt tatt opptak av kan en elev velge å se på et senere tidspunkt. Opptaket kan bli lagt ut på et fags hjemmeside, slik at det blir enkelt å finne. Hvis det er ønskelig å gjennomføre et fag med minst mulig ressurser, kan det være ønskelig å kun tilby opptak av forelesninger i et fag. Slike fag vil stille større krav til en elevs selvdisiplin, og vil egne seg best på høyere utdanning

7.1.4 Skrivebord

Skrivebordkomponenten har tidligere blitt kalt arbeidsbok. Vi endret navnet fra arbeidsbok til skrivebord fordi dette passet bedre med hva som finnes i denne komponenten.

Dokumentene kan i dagens versjon være sortert i mapper og/eller etter kategori. Det er imidlertid få muligheter til å navigere i sine egne dokumenter. Derfor skal det i kommende versjoner være en utforsker der dette skal være mulig. I en slik utforsker vil man kunne organisere og slette sine egne dokumenter.

Delte arbeidsbøker fungerer i nåværende versjon. Alt ligger til rette for dette i datamodellen, og det fungerer for en bruker å ha flere arbeidsbøker. Administrasjon av disse er imidlertid ikke implementer, slik at det er ikke mulig å legge til eller fjerne andre arbeidsbøker. Når man har felles arbeidsbøker, kreves det en viss tilgangskontroll på dokumentene. Slik vi ser for oss vil det gå an å låse hvert dokument til en bruker. Det er kun denne brukeren som kan skrive på dokumentet. Brukeren kan senere låse opp dokumentet for at andre skal kunne skrive på dokumentet.

I dagens program er det ingen enkel måte å få en oversikt hvilke andre brukere som finnes i systemet. Det vil være praktisk med en slags adressebok der man kan finne venner og andre klassekamerater. Man kan da lettere opprette rom med eller finne informasjon om andre brukere.

7.1.5 Skoleweb

Det vil være opp til hver enkelt skole å bestemme hva som skal være på skolens webside. Skolen vi mest sannsynlig bruke en egen hjemmeside til forsiden. En oppslagstavle med nyheter er nok et minimum av det som vil være på siden. Ved bruk av masterpages kan hele skolewebben få et likt utseende som denne skoleforsiden.

7.1.6 Administrasjon

Administrasjonsverktøyet vil bli oppdatert i henhold endringene i resten av systemet. Noe som vil bli lagt til er diverse automatiseringsfunksjoner som det å automatisk melde elever opp i obligatoriske fag, automatisk gruppeinndeling, automatisk overføring av elever til høyere klassetrinn. Inndeling i grupper vil være godt egnet i forbindelse med prosjektarbeider. Gruppene kan få opprettet egne grupperom, der bare gruppemedlemmene har tilgang, og de kan ha en felles e-postadresse for hele gruppen. Ved en slik felles e-postadresse, kan alle i gruppen se den samme e-posten, og dermed

også se alle svar som andre på gruppen skriver.

7.2 Organisering av læremateriell

Vi har lagt opp til alle fag kan inneholde pensummoduler. Disse modulene skal til sammen utgjøre et fags læreplan. En pensummodul kan inngå i flere fag og man kan konstruere nye fag med forskjellige modulkombinasjoner.

7.3 Mulighet for å jobbe frakoblet

En viktig funksjon vil være å kunne arbeide frakoblet. Arkitekturen i programmet er tilrettelagt for å kunne støtte dette. I klienten er det ennå ikke implementert. Det er stort sett innloggings- og utloggingsprosess det vil bli forandringer. Inn- og utloggingsprosessen vil foregå slik som det har blitt forklart i Kapittel 5.2.1. Det å at man arbeide frakoblet innebærer at man ikke er avhengig av internett for å kunne bruke programmet. Man kan velge å laste med dataene som ligger på serveren og ta med seg arbeidet sitt til et sted uten internettforbindelse.
Det er også en viktig del av den smarte klienten å kunne oppdage om internettforbindelsen blir brutt, slik at brukeren kan jobbe videre i frakoblet modus hvis han ønsker det. Når klienten får internettforbindelse igjen vil den automatisk koble seg til serveren igjen. Dette for at brukerne av systemet skal kunne jobbe så uforstyrret som mulig.
Det vil også være muligheter for å synkronisere sine lokale filer med databasen, slik at man kan være sikker på at det som ligger lokalt er det samme som på serveren.

7.4 Brukere og roller

Rollene er opprettet for at brukere med ulike roller skal få ulik tilgang til deler av systemet. I dagens versjon er det nesten ingen forskjell på tilgangsnivåene til rollene. Dette har vi gjort bare for at alle skal få tilgang til å se alle delene av systemet. I et ferdig program vil det som et minimum være roller for elev, lærer, assistent og administrator. Elevrollen har mist tilgang. Elevene har ingen administrative rettigheter. De kan bare endre noen innstillinger om seg selv. Assistenter og lærere har litt flere rettigheter. De kan endre informasjon om fag og klasser de har. De har også mulighet til å gå inn i private rom som tilhører fag de er lærere i. Administratorer er ikke vanlige brukere av undervisningssystemet. De kan ikke delta i klasser fag eller rom, og har ikke sitt personlige skrivebord. Derimot har de alle administrative rettigheter. De er også de eneste som har mulighet til å legge til og endre på personer i systemet.

7.5 Feilbehandling

Det finnes i dag ingen overordnede rutiner for å ta hånd om feil som skjer i systemet. De feilene som oppstår blir skrivet til en feilmeldingslogg. Denne loggen har vi brukt under utviklingen av programmet for å oppdage og eliminere så mange feil som mulig. De som bruker systemet vil derfor sjelden få meldinger hvis det skjer en feil, og må enkelte ganger starte programmet på nytt.

Feil som skjer på serveren kan være at man ikke har tilstrekkelige rettigheter for å utføre en operasjon eller at data av en eller annen grunn ikke har blitt lagret. Det er spesielt slike feil en bruker bør få vite om. De feilene som oppstår på klienten vi klienten selv ta seg av. Brukeren vil ikke merke noe til disse feilene.

7.6 Sikker kommunikasjon

I denne versjonen av programmet har vi i liten grad tatt hensyn til sikker kommunikasjon. Dette er selvsagt noe man må tenke på i det fullstendige systemet, da man skal sende en del personlig informasjon over nettet. Som et minimum må man ha kryptert forbindelse ved innloggingen, slik at man ikke sender passordet i klartekst. For at ikke personer som ikke er brukere av systemet skal få tilgang til webtjenestene, må man identifisere seg for hvert kall fra klienten til webtjenesten. For å slippe å måtte

sende over brukernavn og passord for hvert kall til en webtjeneste, mottar hver bruker et unikt token ved innlogging. Dette tokenet blir brukt som en nøkkel hver gang man kaller tjenesten. Siden dette tokenet bare gjelder for den maskinen man selv sitter på, kan ingen andre nyttiggjøre seg dette, og det kan derfor sendes ukryptert. Dette fører til at man ikke trenger kryptert forbindelse etter innloggingen, noe som sparer en del prosessering og båndbredde. Noe av informasjonen som sendes mellom klient og server bør kanskje også krypteres, men da går det an å gjøre dette spesifikt for denne informasjonen.

7.7 Alternative klienter

7.7.1 Webklient

Klientapplikasjonen vil alltid være tilgjengelig fra internett med ”click-once”-teknologien (kap 3.3.1.3). Denne applikasjonen vil være det beste valget for alle som har mulighet til å bruke den. Hvis man av en eller annen grunn ikke skulle få installert programmet, enten på grunn begrensede rettigheter på offentlige maskiner, at man har operativsystem som ikke støtter .NET-rammeverket, eller at man av en eller annen grunn ikke ønsker å installere klientapplikasjonen, kan man bruke en webklient. Klienten vil være en noe begrenset versjon i forhold til klientapplikasjonen. Den vil blant annet ikke ha synkron kommunikasjon. Derfor vil det heller ikke være noe poeng å implementere samarbeidsrom. I webklienten vil man kunne ha mulighet for å ha det man har på skrivebordet sitt. Dette er i hovedsak arbeidsboka, e-post og kalender. I tillegg skal man også kunne få fram skolewebben.

Webklienten vil kunne identifisere hvilken konfigurasjon man har på maskinen sin, og automatisk vise den best mulige konfigurasjonen. Hvis brukeren har muligheter for å legge inn .NET-rammeverket eller allerede har det installert, vil webklienten foreslå å installere den smarte klientapplikasjonen. Hvis denne ikke blir installert vil websiden vises på vanlig måte.

7.7.2 Mobile klientapplikasjoner

En mobil klientapplikasjon vil kunne ha mye av den samme funksjonaliteten som den smarte klientapplikasjonen. Dette fordi den kan benytte seg av de samme webtjenestene. Med hastigheten på 3G-nettet vil man ha tilsvarende hastighet som en middels rask bredbåndslinje og det vil være mulig å delta på grupperom og følge en forelesning med lyd og video. Med tanke på dagens opplastingshastigheter på 3G vil det lønne seg med serverbaserte løsninger for synkron overføring av lyd og bilde, slik at telefonen ikke trenger å sende til flere. Brukergrensesnittet til en mobil klientapplikasjon vil bli en del forskjellig fra den smarte klientapplikasjonen. I hovedsak på grunn av at skjermstørrelsen er mindre enn en PC-skjerm og ofte også forskjellig fra mobiltelefon til mobiltelefon. Siden man ikke har mus eller tastatur vil det være viktig at det er mulig å navigere rundt med færrest mulig tastetrykk.

7.7.3 Plattformuavhengige klienter

For at programmet skal være tilgjengelig for brukere som ikke har Windows, kan det lages en klientapplikasjon som vil fungere uavhengig av operativsystem. En slik plattformuavhengig klient kan implementeres i for eksempel Java. Denne applikasjonen vil kunne bruke de felles webtjenestene og på den måten bli en erstatning for de som ikke kan bruke .NET-applikasjonen.

7.8 Skalerbarhet

7.8.1 Antall brukere

Det at mesteparten av arbeidet skjer på klienten, gjør at det blir mindre jobb på serveren for hver klient. Dette fører igjen til at man kan legge til flere klienter uten at dette vil føre til mye større belastning på serveren. Man vil derfor slippe å oppgradere serveren så ofte.

7.8.2 Komponenter i programmet

Ved innføring av et slikt system kan det være ønskelig å ikke ta i bruk alle delene av programmet med en gang. Det kan være en grei overgang til et e-læringssystem hvis man i starten for eksempel bare tar i bruk skrivebordet eller rommene. Når man etter hvert føler seg mer trygg på e-læringskonseptet kan man ta i bruk flere deler av systemet. Siden programmet vårt er bygget opp av mange forskjellige komponenter vil dette ikke være noe problem.

7.8.3 Deling av server

Skolen trenger ikke ha sin egen server hvis de ikke ønsker det. Det å drifte egne servere krever en del ekstra resurser. Man må ha ekstra fagkyndig personell og man må fysisk sikre maskinvaren. Hvis skoler ikke selv vil ha ansvar for en slik server kan de benytte seg av servere som distributøren av programmet drifter. Det vil da være mulig at flere skoler som ikke ha behov for en egen server, deler samme den samme fysiske serveren.

8 Evaluering

Hele grunnlaget for å lage vårt eget e-læringssystem i denne oppgaven var å innhente erfaringer og evaluere arbeidet vi har gjort.

8.1 Evalueringsmetoder

Vi har ikke hatt tid til å kjøre noen omfattende tester av e-læringssystemet vårt, men vi har selvsagt prøvd ut og evaluert funksjoner parallelt med utviklingen. I tillegg har vi valgt å demonstrere systemet for en del andre personer, og på den måten fått noen kommentarer og tilbakemeldinger.

Alle personene vi testet systemet på, var forholdsvis vant til å bruke en datamaskin. Bakgrunnen deres for øvrig, var noe forskjellig. Vi skulle gjerne demonstrert programmet for et videre spekter av brukere, men på grunn av tiden vi hadde til rådighet, måtte vi gå til folk vi kjenner i nærmiljøet.

Følgende personer fikk demonstrasjon av systemet vårt:

- En lærer med 27 års erfaring fra grunnskole, som har jobbet flere år som IT-ansvarlig i skolen. Han er godt vant til å bruke datamaskiner, og således ikke en ”gjennomsnittlig” lærer, men vi fikk veldig mange gode tilbakemeldinger. Ingen erfaringer fra andre e-læringssystemer. Demonstrasjonen ble gjort via telefon.
- Tre andre datastudenter. To fra vår egen klasse (med erfaring fra Blackboard og It’s learning) og en fra ingeniørlinje på høyskole (med erfaring fra Blackboard).
- Fem realfagsstudenter. Alle med erfaring fra It’s learning, to av dem hadde også brukt dette da de gikk på videregående.

Ikke alle personene fikk systemet like grundig demonstrert, og de så også systemet vårt på litt forskjellige faser i utviklingen. Likevel tror vi at alle fikk et godt innblikk i hva slags system dette var, og hvilke muligheter som lå i det. Alle personene fikk god tid til å prøve ut systemet selv, men samtidig ledet vi dem også inn mot en del funksjoner vi ville at de skulle prøve.

Grundigst av alle var demonstrasjonene, var den vi gjorde overfor læreren. Riktig nok måtte vi gjøre dette via telefon, og fikk således ikke observert brukerinteraksjonen med egne øyne, men totalt pratet vi sammen i nesten fem timer, så det er liten tvil om at vi fikk mye ut av dette.

8.2 Tilbakemeldinger

Målet vårt med å demonstrere systemet var å få tilbakemeldinger som kunne gi oss en pekepinn på hva som var bra og hva som var dårlig med systemet vårt. Vi er samtidig klar over at utvalget av personer vi latt prøve systemet, ikke på noen måte er representativt for elever og lærere generelt. Derfor har vi ikke tenkt å generalisere noen av de tilbakemeldingene vi fikk, men bare gå gjennom dem og prøve å diskutere om dette kan være verdt å se nærmere på.

Alle brukerne vi demonstrerte programmet for, ga uttrykk for at de var positivt overrasket. Vi fikk mange tilbakemeldinger på brukergrensesnittet, og alle utenom datastudentene sa at de store ikonene var bra. Av læreren vi spurte fikk vi også mange konkrete forslag til endring av knapper og tekster enkelte steder.

Personene trengte litt hjelp til å finne ut hva de kunne gjøre med programmet. Vi hadde kanskje ikke forberedt dem godt nok på hva programmet var ment å brukes til. Hvis vi hadde gitt dem mer kontrete oppgaver, ville muligens disse reaksjonene vært

annerledes. Dette var imidlertid en innledende fase i brukertestingen, og slike ting må man gå i større detalj på ved videre testing.

Flere av de som hadde erfaring fra andre e-læringssystemer, sa at programmet vårt var bedre enn det de hadde prøvd før. Blant annet synes flere at vårt program lå langt over It's learning, særlig på brukervennlighet, men også på funksjonalitet. Læreren, som ikke hadde erfaring med andre e-læringssystemer, mente at programmet vårt sikkert kom til å kreve en del opplæring, men sa samtidig at han ikke visste om noe program for lignende typer oppgaver (e-post og kontor/tekstbehandler ble nevnt), som ville være lettere å ta i bruk.

Fellestavlen så ut til å fascinere mange, og de fleste var nysgjerrige på hva som egentlig lå av tekniske muligheter i den. De var absolutt interessert i å kunne legge ut bilder der, og noen spurte også om det var mulig å vise andre programmer på den. Noe som mange savnet var imidlertid hvordan man får kontakt med vennene sine. "Hvordan kan jeg se hvem som er her?" var et gjennomgående spørsmål. Slik systemet er nå, må man gå inn på et rom før man treffer folk, og dette er ikke så lett å finne ut av for en som er ny og ikke vet hvor han eller hun skal gå.

Totalt sett fikk vi mest positiv respons på programmet. En av realfagsstudentene oppsummerte det veldig greit: "Her har jeg alt jeg trenger. Huskelapper, øvinger jeg må levere, alle dokumentene mine og faginformasjon."

8.3 Vurdering av systemet

Alt i alt er vi veldig godt fornøyd med det systemet vi har laget. Det er et system som kan vises frem, og som kan gi også utenforstående et innblikk i hva e-læring kan være. Vi føler absolutt at vi har klart å belyse temaet på en god måte. Vi klarte selvsagt ikke å lage et ferdig e-læringssystem på den tiden vi hadde til rådighet, men vi klarte å få på plass mange viktige funksjoner, slik at systemet ble så fullstendig at det kunne testes på andre personer.

.NET har vært en positiv erfaring, og dette rammeverket har spart oss for mye arbeid. Vi trengte blant annet ikke tenke noe på nettverk og overføringsprotokoller, og har dermed kunnet rettet fokus mot funksjonalitet og pedagogiske detaljer. Hvis man ser bort fra forstudiet og prototypen høsten 2004, samt en del tid som har blitt brukt på planlegging og dokumentasjon, har utviklingsarbeidet fram til der man er i dag, tatt litt over ett årsverk. Så vidt vi kjenner til fra andre prosjekter, er dette ganske lite med tanke på størrelsen og funksjonaliteten i systemet. Vi kan således si at .NET (og Visual Studio) er et effektivt verktøy for utviklere.

Vi mener også vi har truffet bra med arkitekturen i programmet. Dette gjelder særlig agentløsningen på klienten (Kap. 3.4.2). Dette var noe vi arbeidet lenge med, men da vi mot slutten av prosjektet skulle lage arbeidsbok, fikk vi virkelig se hva det var godt for. Etter litt planlegging av hvordan arbeidsboken skulle implementeres, hadde vi funksjoner for å lagre og hente ut dokumenter fra serveren i gang i løpet av noen få timer. Dette viser at arkitekturen fungerer bra, og ikke minst at det er veldig lett å utvide programmet med flere funksjoner. Vi kan også legge til at denne agentarkitekturen enda ikke har vært gjennomgått siden før påske, og vi tror det vil være mulig å optimalisere den en god del mer. Det ser også veldig greit ut å utvide enkelte av agentene til å bruke direktekommunikasjon (ikke webtjenester) der hvor vi er avhengig av rask kontinuerlig oppdatering.

Brukergrensesnittet ser også ut til å bli bra, men her gjenstår det flere ting i forhold til de målene vi egentlig hadde sett for oss. Blant annet tenkte vi innledningsvis at vi ville ha mulighet for forskjellig vanskelighetsgrad på brukergrensesnittet, og gjerne også mulighet for å velge bort funksjoner. Dette er vi fortsatt usikre på hvordan vi kan løse, men sannsynligvis vil det være aktuelt å innføre et valgfritt dialogprinsipp [2] i deler

av systemet. Videre kan vi også gjøre mye med grensesnittet vi har, gjennom å ha enda flere og tydeligere ikoner, farger, osv. Dette er i henhold til hva mange av personene vi viste programmet til, ønsket seg.

Under utviklingen skulle vi ønske at vi kunne testet samarbeidsfunksjonene litt bedre enn det vi fikk gjort. Særlig hadde det vært ønskelig å kunne testet med mange brukere innlogget samtidig. Dette hadde gitt oss en pekepinn på hvordan systemet fungerer ytelsesmessig med mange brukere. Vi vet imidlertid at alle hovedkomponentene vi benytter oss av på serversiden (SQL Server, Internet Information Services og ASP.NET) er beregnet på store serverløsninger med flere titalls tusen brukere per døgn. Det er derfor grunn til å tro at med riktig maskinvare og konfigurasjon for øvrig, vil systemet vårt være i stand til å takle selv de aller største læringsinstitusjonene.

Av funksjonalitet som fortsatt er på planleggingsstadiet, viste utprøvingen at kontakter er det som er mest prekært å få på plass. Det var mange som spurte seg hvordan de kunne komme i kontakt med andre folk, og det er ikke så enkelt å lete rundt i alle rommene for å finne folk du vil kommunisere med. Det hadde vært langt enklere å ha en liste over alle brukere, samt mulighet for å legge dine venner i en egen liste, slik at man raskt og enkelt kan se om en annen person er pålogget, og innlede en samtale.

8.4 Vurdering av bruksområder

Vi vil her se på hvordan e-læringssystemet vårt kan imøtekomme noen av de bruksområdene som vi har beskrevet i problembeskrivelsen (Kap. 2).

8.4.1 Forelesning

Som nevnt i problembeskrivelsen, er forelesning via internett i sin aller enkleste form, allerede utbredt i bruk. Det vil si å bare ha en tekst, gjerne med bilder, på en webside. Dette er mulig å få til i vårt system, ved å koble denne siden inn under faginformasjon. Hvis man ønsker å legge ut video- eller lydopptak av en forelesning, er dette også mulig å legge ut her. Slik systemet er nå, må man riktig nok må man manuelt legge ut opptaket på internett og så linke filen inn i faginformasjonen. Det er imidlertid ikke vanskelig å gjøre det mulig å legge slike opptak rett inn i systemet, slik at de kommer opp på fagsiden automatisk.

Hvis man derimot ønsker synkron kommunikasjon, er mulighetene foreløpig begrenset til overføring av tekst og enkle strektegninger. Med en forholdsvis liten oppjustering av fellestavlen, vil man også få tilgang til å vise stillbilder. Vi har også forberedt systemet til en viss grad, for direkteoverføring av lyd og video. Det vil da være snakk om å vise levende bilder både av personer, og av diverse fagmessig informasjon på fellestavlen. Det finnes mange løsninger for slik overføring på markedet i dag, og vi ser for oss at man baserer seg på eksisterende komponenter som man legger inn i programmet vårt. Vi vet imidlertid ikke hvilke komponenter som er aktuelle å bruke og hva som er den beste løsningen her.

8.4.2 Gruppearbeid via internett

For gruppearbeid gjelder de samme momentene som for synkron forelesning (jf. forrige avsnitt). Deling av tilgang til fellestavlen er også støttet, slik at alle kan tegne på den etter tur. Siden asynkront gruppearbeid er en såpass uaktuell løsning for de fleste grupper (jf Kap. 2.1.2), har vi valgt å ikke prioritere den i det hele tatt i systemet vårt. Det er imidlertid ikke noe problem å legge inn et forum (også kjent som Message Board eller Bulletin Board) i forbindelse med skolewebben, der folk kan skrive innlegg og svare på dem.

8.4.3 Generelt egenarbeid

Egenarbeid er noe vi er helt sikre på at vil være mulig å få til på en tilfredsstillende måte i systemet vårt. Her er det ingen teknisk avanserte kommunikasjonsformer inne i bildet, og det å sende dokumenter frem og tilbake mellom server og klient, er godt utprøvd i andre systemer. Hvis man ønsker å ta kontakt med noen mens man jobber, har man gode muligheter til dette både ved hjelp av e-post, eller man kan også bruke synkrone kommunikasjonsformer, på samme måte som ved gruppearbeid.

8.4.4 Tester, øvinger, oppfølging

Øvinger og oppfølging er støttet av systemet vårt gjennom at man kan koble egenarbeidsfunksjonene mot øvinger som registreres i systemet. Da er de enkle både å levere inn og gi tilbakemelding på, og man slipper i stor grad at øvinger og tilbakemeldinger kommer bort på veien mellom elev og lærer.

Vi har også diskutert at det kan være aktuelt å ha forskjellige typer øvinger i systemet. Foreløpig er det bare støtte for å levere inn et vanlig tekstdokument, men det bør ikke være noe i veien for å ha spesielle ”øvingsdokumenter”, som kan være skjemaer som eleven skal fylle ut. Dette kan for eksempel benyttes til gloseprøver, flervalgsoppgaver og andre oppgaver med flere spørsmål. Dette er helt klart teknisk mulig å få til. Man kan også tenke seg at det går an å registrere en elektronisk fasit i systemet, slik at systemet kan rette øvingene og gi tilbakemelding både til elev og lærer automatisk.

8.4.5 Administrasjon

Systemet vårt tilbyr mulighet for å administrere både brukere, klasser og fag, i tillegg til rom og andre mer tekniske deler av systemet. Så langt er dette forholdsvis enkelt laget, men det er mulig å bygge ut dette med alt det man normalt har registrert i skolenes arkiv i dag. Mange skoler har i dag egne datasystemer for administrasjonen, og dette bør det ikke være noe i veien for å kunne inkludere i e-læringssystemet. Det er selvsagt en del sikkerhetstiltak i forhold til innsyn og lignende, som må vurderes, men dette er helt klart teknisk mulig å få til.

8.5 Vurdering av målgrupper

Som nevnt tidligere har vi ikke nådd vårt opprinnelige mål når det gjelder brukervennlighet, og kanskje er det ikke mulig å gjøre dette e-læringssystemet så enkelt at det kan brukes av alle på alle utdanningsnivåer. Det er imidlertid flere ting man kan gjøre blant annet med brukergrensesnittet. Veivisere og hjelpesider kan legges til, og valgmuligheter kan økes eller innskrenkes. Vi vil her gå nærmere inn på de forskjellige målgruppene vi diskuterte innledningsvis i problembeskrivelsen (Kap. 2), og se i hvilken grad programmet vårt passer for disse gruppene.

8.5.1 Grunnskole

Det er liten tvil om at brukervennlighet er veldig viktig for barn. Læreren vi snakket med, var veldig usikker på om det ville være mulig å innføre programmet vårt i barneskolen, slik skolen er lagt opp nå. Ungdomsskolen var han litt mer åpen for, men her mente han også at det kom til å være vanskelig å innføre det på alle. Det vil nok i første omgang være noe som de mest oppegående elevene kan få leke seg med, la han til. Ellers mente han at dersom det kunne få lov å komme gradvis, og at det samtidig ble tilbudt god opplæring for de ansatte, ville nok en del av funksjonene kunne bli tatt i bruk. Spesielt var han åpen for at man kunne drive egenarbeid og oppgaveløsning (jf. kap 8.4).

Skolens begrensede ressurser til innkjøp av maskinvare er også et stort problem. Og slik ordningen for grunnskolene er i dag, så er de aller fleste midlene som deles ut til

IT behovsprøvd, og de deles ut til elever, ikke til hele skoler. Det betyr at bare de som fra før av har problemer med å følge den vanlige undervisningen, vil få midler til å ha egen PC på skolen. Læreren vi snakket med mente at dette var uforenelig fordi disse elevene ikke er i stand til å benytte seg av mulighetene i systemet.

Dette tegner selvsagt ikke så godt for innføring av programmet vårt i grunnskolen. Likevel er det mange av de største hindrene som muligens vil forsvinne over tid. For eksempel blir datamaskiner stadig en større del av hverdagen for folk flest. Og når man etter hvert får lærere i skolen som har vokst opp med datamaskiner selv, blir nok terskelen for å ta det i bruk på skolen atskillig lavere. Datautstyr blir generelt billigere og mer tilgjengelig også, så i fremtiden blir det sannsynligvis ikke så vanskelig å tilby elevene tilgang til egen PC på skolen.

Visse ting er det likevel vanskelig å endre på i grunnskolen. Blant annet er grunnskolen knyttet til elevenes nærmiljø, og alle møter opp på skolen hver dag. Alle er enige om at dette er slik man ønsker å ha det også i fremtiden. Derfor er behovet for fjernundervisning i liten grad til stede her. Man kan selvsagt fortsatt bruke systemet vårt til egenarbeid, oppgaveløsning og administrasjon, men da har man samtidig tatt bort noen av de viktigste faktorene ved programmet vårt. Dersom skolene fortsatt skal måtte betale for all funksjonaliteten, vil de kanskje ikke synes det er lønnsomt. Løsningen er da å tilby enkle utgaver av systemet billigere.

Noe som også kan skape problem når man vil innføre et slikt system i grunnskolen, er at lærerne her har relativt mye undervisning (mer enn 20 uketimer), og da blir det fort liten tid til å sette seg inn i noe nytt. Man kan selvsagt lokke med at det går an å spare tid på sikt ved å bruke systemet vårt, men dersom lærerne ikke får satt av tid utover det de normalt har til å forberede seg, vil dette sannsynligvis bli en svært stor tidsmessig investering for mange. Det må trolig en endring i lærernes arbeidsvilkår til for å bedre dette.

8.5.2 Videregående

Mange av de problemene som dukker opp ved innføring av e-læring i grunnskolen, er langt mindre for videregående opplæring. En full stilling på videregående ofte ned mot halvparten så mange uketimer som på grunnskolen, og dermed har man mer tid til å se seg rundt etter nye løsninger for undervisningen, enn det en grunnskolelærer har. Ofte har videregående skoler flere midler til å bruke på IT, og i mange tilfeller er de allerede pålagt å ha datamaskiner tilgjengelige for en stor del av elevene.

Her er også behovet for fjernundervisning noe mer til stede, siden mange ikke lenger bor i nærheten av skolen. Samtidig er elevene langt mer modne og vil trolig lettere kunne klare å ta i bruk et slikt system. Således blir det mer lønnsomt for videregående skoler å innføre et e-læringssystem enn det er for en grunnskole.

8.5.3 Høyere utdanning

I høyere utdanning elimineres veldig mange av problemene fra grunnskolen. Mange steder er allerede IT i utstrakt bruk, og det er normalt tilstrekkelig ressurser til å kunne tilby de fleste studentene tilgang til egen PC på skolen. I tillegg har mange av studentene her langt mer erfaring med datasystemer generelt, og vil sannsynligvis ha mye lettere for å finne fram i brukergrensesnittet i e-læringssystemet vårt.

Et av de aller viktigste argumentene for å ta e-læring i bruk i høyere utdanning, er at her er det virkelig aktuelt å bruke fjernundervisning (jf. Kap. 2.3.4). I høyere utdanning vil man dermed kunne få bruk for alle funksjonene i programmet vårt, og det vil da være langt mer lønnsomt å anskaffe.

Selv om det kan være vanskelig å påvise, er vår egen erfaring at folk i høyere utdanning er langt mer motiverte for å ta et slikt system i bruk. Sannsynligvis henger

dette sammen med at man nå i større grad har fått kontroll over sitt eget liv, enn det man hadde på lavere nivå i utdanningen. Dette fører kanskje til at man setter mer pris på å bli tilbudt løsninger som kan være til hjelp i hverdagen. Da vi viste systemet vårt til medstudenter, fikk vi iallfall utelukkende inntrykk av at de synes dette var et flott tiltak som de håpet at en gang ville bli tatt i bruk.

8.5.4 Annen opplæring

Når det gjelder annen opplæring har vi forholdsvis lite grunnlag for å trekke slutninger i forhold til om programmet egner seg til disse tilfellene. Vi kan imidlertid tenke oss at forskjellige former for etterutdanning vil ha behov for fjernundervisning (jf. Kap. 2.3.5), og da er det naturlig at vårt program kan brukes der på samme måte som for videregående og høyere utdanning. Dette gjelder også for voksenopplæring, men her har muligens ikke studentene så gode datakunnskaper generelt, så systemet vil antakelig ikke være brukervennlig nok for alle.

For generelle kurs rundt om i markedet er det også vanskelig å si hvordan vårt system ville blitt mottatt. Vi kan imidlertid anta at fjernundervisningsfunksjonene også kan komme til nytte her. Mange kurs og seminarer krever at folk reiser langt for å delta. Mange kurs pågår kanskje også så sjelden og i liten skala, at det ikke er hensiktsmessig å ha egne lokaler til dem. Det kan da være en løsning å ha "lokaler" på nett, noe som sannsynligvis er langt billigere.

8.6 E-læring videre fremover

Som vi har vært innom tidligere i diskusjonen, er det mange av problemene ved innføring av e-læring som sannsynligvis vil endre seg over tid. For eksempel vil datakunnskapen til folk flest øke etter hvert som flere vokser opp med datamaskiner rundt seg. Når både elevenes foreldre og lærere er vant til å bruke data fra deres egen oppvekst, vil det trolig være naturlig at eleven også lærer å bruke dette, både på skolen og hjemme. Da vil ikke lenger e-læringssystemet virke så nytt og skremmende, men det vil være et naturlig verktøy omtrent som bok og blyant er i dag.

Kanskje det i fremtiden også vil komme mer standardiserte løsninger, slik at man mer eller mindre kan ha sin personlige livslange konto i et generelt e-læringssystem. Så kan man bare flytte rundt internt i systemet når man bytter skole, eller skal ta kurs og annet senere i livet. Brukernavn og passord er det samme, men du bestemmer da hvilke fag og klasser du er med i på tvers av fysiske skoler.

Det finnes mange nettskoler i dagens samfunn, men disse er i stor grad basert på asynkron kommunikasjon, omtrent som et litt avansert brevkurs. Vi tror det vil ha mye å si at man kan få inn synkron kommunikasjon, og gjerne mellom flere parter. I dagens nettskoler opptrer ofte læreren og de andre elevene bare som en liste med navn, og dette kan fort gjøre det hele blir upersonlig og man føler at man sitter alene og studerer. Det at man kan oppfatte de andre som mer tilstedeværende, gjennom å se og høre dem, kan gi en følelse av at man er mer til stede. For mange tror vi dette vil være avgjørende for at de skal se e-læring som et reelt alternativ til å være fysisk til stede.

Vi tror helt klart at e-læring vil bli vanlig i de aller fleste typene utdanning. Ikke nødvendigvis enerådende, men kanskje som et tillegg til den måten undervisningen foregår på i dag. For enkelte typer utdanning vil det være en del problemer med å innføre e-læring, men av alle problemene vi har sett på i denne oppgaven, er det få som egentlig er relatert til tekniske løsninger. For de fleste formål er det fullt mulig å formidle kunnskap på en tilfredsstillende måte over internett.

9 Konklusjon

Gjennom temaene vi har belyst i denne oppgaven har vi funnet mange gode grunner til å bruke e-læring i større grad enn det som er tilfelle i dag. Vi har også sett en del problemer knyttet til innføring av e-læring, men i veldig mange tilfeller er dette problemer som sannsynligvis vil bli løst over tid. Problemene er i liten grad knyttet til muligheter i teknologien som eksisterer.

Gjennom å bruke Microsoft .NET som hovedteknologi i utviklingen, har vi sett at dette er en effektiv utviklingsteknologi, som er godt egnet til typiske e-læringssystemer. Gjennom bruk av webtjenester og smarte klienter oppnår man god funksjonalitet og kommunikasjon, og programmet blir raskt og enkelt både å sette opp og bruke.

Fra personer som prøvde ut systemet vårt fikk vi mange tilbakemeldinger som antydet at de var positive til å ta i bruk et slikt system i fremtiden. Det var særlig de synkrone kommunikasjonsfunksjonene våre som fenget, selv om disse utgjorde en veldig liten del av systemet. Det var i hovedsak disse funksjonene som bidro til å gi følelsen av at det er noen i den andre enden, noe som er helt sentralt i samarbeidslæring. Det er grunn til å tro at synkron kommunikasjon vil være en veldig viktig faktor i den totale brukeropplevelsen, og i mange tilfeller vil det være avgjørende for om e-læring kan tas i bruk.

Appendiks

Appendiks A: Demonstrasjon av Læringsrom i nett

E-læringssystemet vårt, Læringsrom i nett, er tilgjengelig fra internett og er mulig å prøve for alle som har brukernavn og passord til systemet. Mer informasjon om nedlasting og kjøring finnes på adressen:

http://www.webcider.no/rominett/

For utprøving i forbindelse med evaluering av denne oppgaven, har vi opprettet følgende brukere som man kan bruke ved innlogging:

Brukernavn	Passord
Test1	t1
Test2	t2
Test3	t3
Test4	t4
Test5	t5
Test6	t6
Test7	t7
Test8	t8
Test9	t9
Test10	t10

Tabell 0-1 Brukernavn og passord til testing

Det henstilles om at de som vil teste ut systemet avtaler seg imellom hvilke brukernavn de vil bruke, da det vil oppstå feil hvis flere prøver å logge på med samme brukernavn. For opprettelse av flere brukere, samt generelle spørsmål til hvordan man kommer inn i systemet, ta kontakt med oss. Kontaktinformasjon står websiden nevnt ovenfor.

Referanser

[1] Judith Bell: Doing Your Research Project, third edition (Open University Press/McGraw Hill 1999, ISBN (pb) 0-335-20388-4 (hb) 0-335-20389-2), side 10.
[2] Preece, Rogers, Sharp: Interaction Design (Wiley 2002, ISBN 0-471-49278-7), kap 2.3.
[3] Microsoft Developer Network: Smart Client Definition (http://msdn.microsoft.com/smartclient/understanding/definition/)

Kilder

I forbindelse med utviklingsarbeidet har vi spesielt brukt disse kildene:

- A Step Up from the ASP.NET, .NET Starter Kits (http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnvs05/html/SKinVS2005.asp)
- Data Access in ASP.NET 2.0 (http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnvs05/html/dataaccess.asp)
- Master Pages in ASP.NET 2.0 (http://msdn.microsoft.com/library/en-us/dnvs05/html/masterpages.asp?frame=true)
- Digging into SOAP Headers with the .NET Framework (http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnservice/html/service06182002.asp)
- Web Services Security (http://www.rassoc.com/gregr/weblog/stories/2002/06/09/webServicesSecurity.html)
- Personalization with ASP.NET 2.0 (http://msdn.microsoft.com/library/en-us/dnvs05/html/Person_fin.asp?frame=true)
- Working with .NET Threads (http://www.code-magazine.com/article.aspx?quickid=0309071)