

# Testing Via Internet - Live

Thomas Pettersen, Terje Kirkemyr Nilsen og Elin Synnøve Solberg

22. mai 2001

# Innhold

<b>1</b>	<b>Innledning</b>	<b>7</b>
1.1	Definisjon av datatekniske ord og uttrykk brukt i rapporten . . . . .	7
1.2	Problemområde og avgrensning . . . . .	8
1.2.1	Problemområde . . . . .	8
1.2.2	Avgrensning . . . . .	9
1.2.3	Oppgavebeskrivelse . . . . .	9
1.3	Målgruppe . . . . .	9
1.4	Formål . . . . .	10
1.5	Egen bakgrunn og kompetanse . . . . .	10
1.6	Arbeidsformer . . . . .	10
1.7	Organisering av rapporten . . . . .	11
1.8	Terminologi . . . . .	12
1.9	Praktisk . . . . .	12
<b>2</b>	<b>Kravspesifikasjon</b>	<b>13</b>
2.1	Brukerbeskrivelse . . . . .	13
2.1.1	Omgivelser . . . . .	13
2.1.2	Systemet har fire grupper av brukere. . . . .	13
2.1.3	Funksjon . . . . .	13
2.1.4	Operasjon . . . . .	14
2.1.5	Aspekter omkring livssyklus . . . . .	15
2.1.6	Ytelse . . . . .	15
2.1.7	Begrensninger . . . . .	15
2.1.8	Antagelser . . . . .	15
2.1.9	Hjelpesfunksjon . . . . .	15
2.2	Funksjonell kravspesifikasjon . . . . .	15
2.2.1	Funksjonell struktur og tverrelasjoner . . . . .	15
2.2.1.1	Superbruker: . . . . .	15
2.2.1.2	Lærer: . . . . .	16
2.2.1.3	Student: . . . . .	17
2.2.2	Data spesifikasjon . . . . .	17
2.2.2.1	Data rammeverk . . . . .	17
2.2.2.2	Data input . . . . .	17
2.2.2.3	Data output . . . . .	17
2.2.2.4	Tverrfunksjonelle datadefinisjoner . . . . .	17
2.2.3	Overordnede operasjonelle systemkrav . . . . .	17
2.2.3.1	Normal operasjon . . . . .	17
2.2.3.1.1	Modus og kontroll . . . . .	17
2.2.3.1.2	Ytelse . . . . .	17
2.2.3.1.3	Sikkerhet . . . . .	17
2.2.3.1.4	Oppstart og nedtagning . . . . .	20
2.2.3.1.5	Tilgjengelighet . . . . .	20
2.2.3.1.6	Innebygde tester . . . . .	20

2.2.3.2	Operasjon i feilsituasjoner . . . . .	20
2.2.3.2.1	Feilrapportering . . . . .	20
2.2.3.2.2	Gjenervervelse etter feil . . . . .	20
2.2.3.2.3	Sikkerhet . . . . .	20
2.2.3.2.4	Ytelse . . . . .	20
2.2.4	Funksjonelle krav . . . . .	20
2.2.4.1	Funksjonelle krav til lærernes spørsmålsmodul . . . . .	20
2.2.4.1.1	Input . . . . .	20
2.2.4.1.2	Prosessering . . . . .	21
2.2.4.1.3	Output . . . . .	21
2.2.4.1.4	Feilrapportering . . . . .	22
2.2.4.1.5	Gjenervervelse etter feil . . . . .	22
2.2.4.1.6	Sikkerhet . . . . .	22
2.2.4.2	Funksjonelle krav til lærernes statistikkmodul . . . . .	22
2.2.4.2.1	Input . . . . .	22
2.2.4.2.2	Prosessering . . . . .	22
2.2.4.2.3	Output . . . . .	22
2.2.4.2.4	Sikkerhet . . . . .	22
2.2.4.3	Funksjonelle krav til studentenes svarmodul . . . . .	22
2.2.4.3.1	Input . . . . .	22
2.2.4.3.2	Prosessering . . . . .	22
2.2.4.3.3	Output . . . . .	22
2.2.4.3.4	Feilrapportering . . . . .	22
2.2.4.3.5	Gjenervervelse etter feil . . . . .	23
2.2.4.4	Funksjonelle krav til studentenes statistikkmodul . . . . .	23
2.2.4.4.1	Input . . . . .	23
2.2.4.4.2	Prosessering . . . . .	23
2.2.4.4.3	Sikkerhet . . . . .	23
2.2.4.5	Funksjonelle krav til Superbrukerens verktøy . . . . .	23
2.2.4.5.1	Input . . . . .	23
2.2.4.5.2	Prosessering . . . . .	23
2.2.4.5.3	Output . . . . .	23
2.2.4.5.4	Feilrapportering . . . . .	23
2.2.4.5.5	Sikkerhet . . . . .	24
2.2.5	Spørsmålstyper . . . . .	24
2.2.5.1	Flervalgsspørsmål . . . . .	24
2.2.5.1.1	Spørsmålsbeskrivelse . . . . .	24
2.2.5.1.2	Input - lærere . . . . .	24
2.2.5.1.3	Input - studenter . . . . .	24
2.2.5.1.4	Statistikk . . . . .	24
2.2.5.2	Fylle inn tall eller ord . . . . .	24
2.2.5.2.1	Spørsmålsbeskrivelse . . . . .	24
2.2.5.2.2	Input - lærere . . . . .	25
2.2.5.2.3	Input studenter . . . . .	25
2.2.5.2.4	Statistikk . . . . .	25
2.2.5.3	Finne sekvens . . . . .	25
2.2.5.3.1	Spørsmålsbeskrivelse . . . . .	25
2.2.5.3.2	Input - lærere . . . . .	25
2.2.5.3.3	Input - studenter . . . . .	25
2.2.5.3.4	Statistikk . . . . .	26
2.2.5.4	Kombinere . . . . .	26
2.2.5.4.1	Spørsmålsbeskrivelse . . . . .	26
2.2.5.4.2	Input - lærere . . . . .	26
2.2.5.4.3	Input - studenter . . . . .	26
2.2.5.4.4	Statistikk . . . . .	26

2.2.5.5	Plassere ut noe på et bilde . . . . .	26
2.2.5.5.1	Spørsmålsbeskrivelse . . . . .	26
2.2.5.5.2	Input - lærere . . . . .	26
2.2.5.5.3	Input - studenter . . . . .	26
2.2.5.5.4	Statistikk . . . . .	27
2.2.5.6	Peke på et sted/areal i et bilde med progressive hint . . . . .	27
2.2.5.6.1	Spørsmålsbeskrivelse . . . . .	27
2.2.5.6.2	Input - lærere . . . . .	27
2.2.5.6.3	Input - studenter . . . . .	27
2.2.5.6.4	Statistikk . . . . .	27
2.2.5.7	Nye spørsmål . . . . .	27
2.2.5.7.1	Spørsmålsbeskrivelse . . . . .	27
2.3	Begrensninger . . . . .	27
2.3.1	Software design begrensninger . . . . .	27
2.3.1.1	Software standarder og språk . . . . .	27
2.3.1.2	Software grensesnitt . . . . .	27
2.3.1.3	Software pakker/verktøy . . . . .	27
2.3.1.4	Software kommunikasjonsstandarder og grensesnitt . . . . .	27
2.3.1.5	Database . . . . .	28
2.3.1.6	Operativsystem . . . . .	28
2.3.2	Hardware design begrensninger . . . . .	28
2.3.2.1	Hardware krav og omgivelser . . . . .	28
2.3.2.2	Hardware grensesnitt . . . . .	28
2.4	Aspekter omkring livssyklus . . . . .	28
2.4.1	Dokumentasjon . . . . .	28
2.4.2	Modul og integrasjonstesting . . . . .	28
2.4.3	Konfigurasjons- og versjonstyring . . . . .	28
2.4.4	Krav til support, service og vedlikehold . . . . .	28
2.4.5	Krav til utvidelser . . . . .	28
2.5	Aspekter omkring installasjon . . . . .	28
2.5.1	Overgang og omlegging . . . . .	28
2.5.2	Opplæring . . . . .	29
2.6	Utgivelser underveis . . . . .	29
2.7	Akseptansekrav . . . . .	29
<b>3</b>	<b>Analyse</b> . . . . .	<b>30</b>
3.1	Hva systemet skal inneholde . . . . .	30
3.2	Eksisterende løsninger . . . . .	30
3.3	Vurderinger . . . . .	30
<b>4</b>	<b>Design</b> . . . . .	<b>32</b>
4.1	Hvordan løsningen ble laget . . . . .	32
4.1.1	Layout . . . . .	32
4.1.2	Databasen . . . . .	32
4.1.3	Lærermodule . . . . .	32
4.1.3.1	Fag . . . . .	32
4.1.3.1.1	Nytt fag . . . . .	35
4.1.3.1.2	Endre fag . . . . .	35
4.1.3.1.3	Slette fag . . . . .	35
4.1.3.1.4	Statistikk . . . . .	35
4.1.3.2	Test . . . . .	35
4.1.3.2.1	Ny test . . . . .	38
4.1.3.2.2	Endre test . . . . .	39
4.1.3.2.3	Slette test . . . . .	39
4.1.3.2.4	Statistikk . . . . .	39

4.1.3.2.5	Tester generelt . . . . .	39
4.1.3.3	Spørsmål . . . . .	39
4.1.3.4	Flervalgsspørsmål . . . . .	42
4.1.4	Studentmodulen . . . . .	44
4.1.5	Superbrukermodulen . . . . .	47
4.1.5.1	Klasselister auto nye/slette . . . . .	47
4.1.5.2	Manuelt nye . . . . .	47
4.1.5.3	Endre klasse . . . . .	47
4.1.5.4	Slette klasse . . . . .	49
4.1.5.5	Lærere . . . . .	49
4.1.5.6	Ny lærer . . . . .	49
4.1.5.7	Endre lærer . . . . .	49
4.1.5.8	Slette lærer . . . . .	49
4.1.5.9	Overføre et fag fra en lærer til en annen . . . . .	52
4.1.5.10	Spørsmålstyper . . . . .	52
4.1.5.11	Karakterskala . . . . .	52
4.1.5.12	Personlig info . . . . .	52
4.1.5.13	Validering . . . . .	52
4.1.5.14	Godkjentlister . . . . .	53
4.1.5.15	Statistikk . . . . .	53
4.1.5.16	Påminnelser . . . . .	53
4.1.5.17	Fag og tester . . . . .	53
4.1.5.18	Logg ut . . . . .	53
<b>5</b>	<b>Implementering</b>	<b>54</b>
5.1	Valg av programmeringsspråk . . . . .	54
5.2	Valg av metode . . . . .	54
5.3	Bakgrunn for verktøyvalg . . . . .	60
5.4	Prinsipper vi har fulgt . . . . .	60
5.5	Beskrivelse av forløp . . . . .	61
5.6	Endelig løsning . . . . .	61
5.7	Legge til flere spørsmålstyper . . . . .	61
<b>6</b>	<b>Kvalitetssikring, testing og realisering</b>	<b>62</b>
6.1	Organisering av kvalitetssikring . . . . .	62
6.1.1	Dokumentasjon . . . . .	62
6.1.2	Standarder, rutiner, konvensjoner . . . . .	62
6.1.3	Konfigurasjonsstyring . . . . .	62
6.1.4	Problemrapporing og tiltak . . . . .	62
6.1.5	Kodekontroll . . . . .	63
6.1.6	Ressursbruk . . . . .	63
6.2	Kvalitetssikring av produktet og testing. . . . .	63
6.3	Kvalitetssikring av prosess . . . . .	63
6.4	Evaluering av kvalitetssikringsmal . . . . .	63
<b>7</b>	<b>Oppsummering</b>	<b>64</b>
7.1	Løsningen . . . . .	64
7.2	Erfaringshøsting . . . . .	64
7.2.1	Måloppnåelse . . . . .	64
7.2.2	Hva er gjort bra . . . . .	65
7.2.3	Utvidelsesområder for en totalløsning . . . . .	65
7.2.4	Deler av løsningen med forandringspotensiale . . . . .	66
7.2.5	Hva har vi lært gjennom prosjektperioden . . . . .	66
7.3	Emne for nytt hovedprosjekt . . . . .	66
7.4	Konklusjon . . . . .	67

<b>8</b>	<b>Verktøy</b>	<b>68</b>
8.1	Software systemet er avhengig av . . . . .	68
8.2	Verktøy vi har brukt under utviklingen av systemet . . . . .	69
<b>9</b>	<b>Litteraturliste</b>	<b>70</b>
9.1	Bøker og manualer . . . . .	70
9.2	Internettressurser . . . . .	70
<b>10</b>	<b>Vedlegg</b>	<b>72</b>
10.1	Vedlegg 1, Nettverksstruktur . . . . .	72
10.2	Vedlegg 2, Shellsript for backup . . . . .	73
10.3	Vedlegg 3, Første Gantt-skjema, Fremdriftsplan . . . . .	73
10.3.1	Liste over aktiviteter . . . . .	73
10.3.2	Milepæler . . . . .	74
10.3.3	Beslutningspunkter . . . . .	74
10.3.4	Tids- og ressursplaner . . . . .	74
10.4	Vedlegg 4, SQL-sript som oppretter databasen . . . . .	76
10.5	Vedlegg 5, Databaserapport . . . . .	79
10.6	Logg . . . . .	86
10.7	Statusrapporter . . . . .	88
10.7.1	Statusrapport I . . . . .	88
10.7.2	Statusrapport II . . . . .	90
10.7.3	Statusrapport III . . . . .	91
10.8	Møtereferater . . . . .	92
10.8.1	Møtereferat 12.01.2001 . . . . .	92
10.8.2	Møtereferat 19.01.2001 . . . . .	92
10.8.3	Møtereferat 31.01.2001 . . . . .	93
10.8.4	Møtereferat 09.02.2001 . . . . .	93
10.8.5	Møtereferat 23.02.2001 . . . . .	94
10.8.6	Møtereferat 23.03.2001 . . . . .	94
10.8.7	Møtereferat 03.04.2001 . . . . .	95
10.8.8	Møtereferat 20.04.2001 . . . . .	95
10.8.9	Møtereferat 27.04.2001 . . . . .	95
10.8.10	Møtereferat 04.05.2001 . . . . .	96
10.8.11	Møtereferat 16.05.2001 . . . . .	96

# Figurer

2.1	Hovedtrekk i systemet . . . . .	14
2.2	Dataflyt Diagram 0 . . . . .	16
2.3	Dataflyt-Diagram 1 . . . . .	18
2.4	Forenklet ER - diagram . . . . .	19
4.1	Sidens oppbygning, lærer. . . . .	33
4.2	Sidens oppbygning for student og superbruker. . . . .	33
4.3	Fullstendig ER-diagram . . . . .	34
4.4	Fag. . . . .	35
4.5	Nytt fag. . . . .	36
4.6	Test . . . . .	37
4.7	Ny test. . . . .	38
4.8	Tester generelt. . . . .	39
4.9	Spørsmålsdelen av lærermodul. . . . .	40
4.10	Sletting av spørsmål. . . . .	40
4.11	Nytt spørsmål generelt. . . . .	41
4.12	Dialog for å velge fil. . . . .	42
4.13	Flervalgsspørsmål, ett av to alternativer er riktig. . . . .	43
4.14	Flervalgsspørsmål, ett av mange alternativer er riktig. . . . .	43
4.15	Flervalgsspørsmål, ingen eller flere av mange alternativer er riktig. . . . .	43
4.16	Studenenes innloggingsside. . . . .	44
4.17	Registrering av student. . . . .	45
4.18	Studentens valg etter innlogging. . . . .	45
4.19	Studentens oversikt over tester. . . . .	46
4.20	Studenten starter en test. . . . .	46
4.21	Testen har startet. . . . .	47
4.22	Automatisk oppretting av klasselister. . . . .	48
4.23	Registrer ny klasse. . . . .	48
4.24	Endre klasse. . . . .	49
4.25	Ny lærer. . . . .	50
4.26	Endre lærer. . . . .	50
4.27	Slette lærer. . . . .	51
4.28	Overføre fag mellom lærere. . . . .	51
4.29	Forklarende tekst for spørsmålstyper. . . . .	52

# Kapittel 1

## Innledning

### 1.1 Definisjon av datatekniske ord og uttrykk brukt i rapporten

**ASP:** Active Server Pages. Programmeringsspråk som brukes til utvikling av dynamiske websider. Utviklet av Microsoft.

**Boolean:** Verdi som har verdien 0 / 1 eller false / true. Brukes for å besvare et spørsmål med usant / sant eller ja / nei.

**Cache:** Midlertidig lagring av data for raskere framhenting.

**CSS/Styles:** Cascading Style Sheets, mal for innstillinger av utseende på elementer i en nettside.

**Database:** System for lagring av datamengder, kan være både bilder, video, lyd, tekster og verdier. Dataene er tilgjengelig for systemet og dets brukere gjennom akksesering av databasen.

**DHTML:** Teknikk som benytter JavaScript for å operere på CSS.

**Frames:** Inndeling av nettleser-vinduet i flere mindre vinduer.

**Gantt-skjema:** Tidsplan med spesifisering av aktiviteter og milepæler.

**Grafisk brukergrensesnitt:** Utforming av presentasjon. Bevisste valg som er blitt gjort for å utforme og presentere sidene for brukerne.

**HTML:** Hyper Text Markup Language. Såkalt merkespråk som benyttes på internett. Et sett av koder som brukes for å lage design og funksjonalitet på nettsider.

**Integer:** Positivt eller negativt heltall.

**Internettforbindelse:** Forbindelse påkrevd for å koble seg til Internett.

**Internettjener/Webtjener/Webserver:** Maskin som benyttes for å tilby internettjenester.

**Java:** Et programmeringsspråk.

**JavaApplet:** Selvstendige Java programmer som kan kjøres via en nettleser.

**JavaScript:** Programkode som kjøres på klientsiden etter at en nettside er lastet ned.

**Javaservlets:** Programmeringsspråk for å lage dynamiske websider. Utviklet av Sun Microsystems.

**Kode:** Instruksjoner som Programkode skrevet i det angitte språket.

**Kompilering:** Å kompilere programkode vil si å oversette fra et programmeringsspråk til maskinkode som datamaskinen forstår. Til denne oppgaven blir det benyttet ferdige kompilatorprogrammer.



**Kravspesifikasjon:** Detaljert beskrivelse av hva systemet skal inneholde av funksjonalitet, hva systemet skal løse og på hvilken måte.

**Linux:** Unixbasert Operativsystem, alternativ til Microsoft Windows.

**Modul:** Delløsning. Vi deler totalløsningen inn i elementer/funksjoner etter hvilke oppgaver de skal dekke, navngir disse med hvilken tilhørighet de har, f.eks. Studentmodul, Lærermodule, osv...

**MySQL:** Et spesifikt databasesystem som støtter SQL.

**Nettleser:** Program som man bruker for å se på internettsider.

**Nettverk:** Flere datamaskiner knyttet sammen i et nett.

**PHP:** Programmeringsspråk for utvikling av dynamiske websider.

**Plattformuavhengighet:** Uavhengighet i forhold til hvilket operativsystem en benytter.

**Programmeringsspråk:** En spesifikk måte å kode på.

**SQL:** Standard Query Language, standardisert språk for å kommunisere med en database.

**Superbruker:** Den personen som får ansvaret for å oppdatere og administrere systemet etter det er blitt implementert på organisasjonens nett.

**Switch:** Koplingsboks for å koble sammen datamaskiner i ett nett.

**Testgenerator:** Den delen av løsningen der vi oppretter fag, tester og spørsmål. Også kalt Lærermodule.

**WYSIWYG HTML editor:** (What you see is what you get), html editorer som genererer html-kode ut fra et grafisk design.

**Zip:** Komprimeringsfunksjon for filer.

## 1.2 Problemområde og avgrensning

I dag må alle studenter som skal avlegge en obligatorisk øving eller test, levere den i papirformat ved personlig fremmøte ved undervisningsstedet. I tillegg må faglærer eller fagassistenter rette den innleverte oppgaven. Dette er en ressurskrevende oppgave og derfor kan testing av denne grunn virke hemmende på faglærernes ønsker eller behov for hyppigere testing eller evaluering. Mange studenter ønsker nok også å kunne måle sine kunnskapsnivåer ved å besvare statustester etter hvert som pensumet i aktuelle fag blir behandlet. Studentene kan måle sine ferdigheter eller progresjon innen et gitt fag mot hva faglærerne forventer av kunnskapsnivå på gjeldende tidspunkt. Studenten kan "regulere" sin innsats etter hva som er forventet av ham til gitte tidspunkt.

Spørsmålstypene er svært godt utformet i kravspesifikasjonen, og dette gir mulighet for å lage gode og varierte måter å stille spørsmål på. Det at løsningen er tilgjengelig for studentene via Internett vil gi mange muligheter. Studentene kan befinne seg på et spredt geografisk område, de kan gjennomføre tester når det måtte passe best innenfor grensene læreren har satt, og vi tror også at mange vil oppfatte det som mer interessant å ta testen via Internett, og dermed øke motivasjonen og interessen til faget.

### 1.2.1 Problemområde

Testing av studentene har tidligere vært vanskelig da det har vært for tidkrevende for faglærere å gjennomføre. Vår oppgave var å lage et verktøy som skulle være så enkelt at enhver lærer uansett datakunnskapsnivå, skal kunne lage sine tester beregnet på sitt fag eller fagområde til benyttelse av studentene. Løsningen skal ikke begrense seg til bruk bare innen et bestemt fagfelt, men være like funksjonelt og dekkende for alle HiG's avdelinger. Løsningen skal dekke de mest elementære spørsmålstypene i første omgang, men vi ser ikke bort fra at flere spørsmålstyper vil være aktuelle i fremtiden. Det å bruke moderne dataverktøy og kommunikasjonsmidler er selvfølgelig også kostnadseffektivt. Dette har ført til at man ønsker å bruke teknologien for å effektivisere arbeidet for faglærerne og testing av studenter. Løsningen skal muliggjøre testing og retting via Internett "live". Et av de beste mediene for elektronisk testing er Internett, og alle studenter har tilgang på Internett via HiG's studentdatalab'er og også de fleste har tilgang fra sine bosteder. HiG benytter også web'en aktivt i informasjonsøyemed ovenfor studentgruppene sine.

## 1.2.2 Avgrensning

Vår oppgave konsentrerte seg om testing av studenter via Internett. Studentene skal kunne svare på spørsmål relatert til faget for å se om de har de forkunnskapene som faget krever og for å se om de holder følge med den faglige progresjonen som forventes. Faglærer kan legge oppgaver ut på Internett som kreves bestått av studentene for at de skal kunne få gå opp til eksamen. Vi har laget et verktøy som gjør det enkelt for faglærere å lage spørsmål, sette disse sammen til oppgaver og legge disse ut på Internett. Vi laget også det systemet som studentene bruker for å svare på spørsmålene, og et verktøy som gjør at lærerne kan se hvordan resultatene er og presentere for studentene hvordan klassen ligger an faglig. Studentene kan se om de har fått godkjent sine oppgaver. Systemet skal rette og gi poeng til studenten på oppgaver hvor faglærerne har definert riktige svar på forhånd og oppgitt hvilke poengsummer som skal gis for riktige svar. Verktøyet skal ikke inneholde noen løsning på hvordan en god spørreundersøkelse bygges opp. Det er opp til den som lager en test å stille "rette" og pedagogiske riktige spørsmål.

Systemet er laget slik at det er mulig for programmerere å legge til spørsmålstyper senere. Vi skal definere noen ferdige spørsmålstyper og utvikle så mange av disse som vi klarer innenfor rammene av prosjektet. Det er viktig at systemet fungerer godt og at det er robust og har de viktigste funksjonene implementert før vi er ferdige med prosjektet. Vi skal lage en løsning i henhold til den kravspesifikasjonen som følger prosjektet. Systemet skal utvikles for kjøring på HiGs nettverk, og behøver ikke være en universell løsning.

## 1.2.3 Oppgavebeskrivelse

Det skal utvikles fem forskjellige moduler (refererer til: Figur 2 side 13 i Prosjektrapport fra 26.05.2000), dette er :

1. Endre innstillinger (superbruker).  
Endre konfigurasjon for systemet. Mulighet for å endre og opprette brukere.
2. Lage og redigere tester (lærer).  
Faglærerne ved HiG skal kunne legge ut tester og obligatoriske øvinger på Internett. Vi skal lage et verktøy som gjør det enkelt for faglærer å lage spørsmål, sette disse sammen til fullstendige øvinger og legge disse ut på Internett. Dette skal være et verktøy som skal være enkelt å bruke, slik at flest mulig lærere ved HiG tar det i bruk.
3. Statistikk fra testene (lærer).  
Lærerens verktøy for å se på hver enkelt student og studentenes samlede resultater fra testene, og presentere for studentene hvordan klassen ligger an faglig.
4. Spørremodul (student).  
Studentenes modul for å besvare allerede utlagte øvinger. Systemet skal rette og gi poeng eller status til studenten på oppgaver hvor faglærer har definert riktig svar på forhånd og oppgitt hvilke poengsummer som skal gis for riktig svar. Dette gjør at studentene faktisk ikke trenger å være fysisk til stede ved læringsstedet hele tiden.
5. Sjekk godkjent og status (student).  
Studentene kan se resultater fra sine tester.

Dessuten ønsket vi å utvikle et nytt brukergrensesnitt/design for systemet. Vi vil lage et brukergrensesnitt som ser litt mer ryddig ut, og gjenkjennelsesfaktoren for de enkelte skjermbildene skal være stor.

## 1.3 Målgruppe

Målgruppe for prosjektrapporten og oppgaven vil naturlig nok være:

- HiG som oppdragsgiver.
- Lærere ved avdelingene AT og AH.
- Ekstern sensor for hovedprosjektrapporten.

- En eventuell kommende hovedprosjektgruppe som skal jobbe videre med totalløsningen.
- Andre som har interesse for elektronisk læring og webutvikling generelt.

På grunn av prosjektrapportens brede målgruppe var et av våre mål at det meste skal være lettleselig og forståelig også for de som ikke er datakyndige. En del datafaglige ord og uttrykk er brukt i teksten, men det er på detaljbeskrivelser som er mest interessante for de som er interessert i utviklingen av systemet.

## 1.4 Formål

Oppdragsgiver ønsker en elektronisk løsning som er tidsbesparende for en som underviser mange. Oppgaven blir rettet og godkjentlister blir generert automatisk. . Faglærer kan måle kunnskapsnivået hos sine studenter ved hjelp av testene, det være seg at de er statustester eller obligatoriske oppgaver som må besvares for å få lov til å ta eksamen. Måling av ervervede kunnskaper kan være en god indikator på undervisningsform og kunnskapsformidling, og faglærer kan dermed planlegge sin undervisning på en bedre måte ved hjelp av systemet.

## 1.5 Egen bakgrunn og kompetanse

Prosjektdeltagerne er alle fra klasse 98HINDB fra datalinja på HiG. Alle har vi hatt grunnleggende opplæring innen programmering i C++, Delphi og Java. Vi har også hatt som obligatoriske fag Databaser I, Datamaskinens oppbygning og virkeområde, Systemutvikling I og II, Drift av flerbrukersystemer, Operativsystemer, Grafikk og bildebehandling, Programmering mot WWW, Algoritmiske metoder I og Datakommunikasjon. Alle i prosjektgruppa har hatt Klient og serversideprogrammering og Nettverksbasert multimedia som valgfag. Thomas Pettersen har i tillegg Kreativ problemløsning som valgfag. Terje Kirkemyr Nilsen og Elin Synnøve Solberg har hatt Algoritmiske metoder II og Elin har hatt Databasesystemer II som valgfag.

Thomas Pettersen og Terje Kirkemyr Nilsen har jobbet en del med Internett-løsninger tidligere, og kjenner derfor til en del av teknologiene som kan benyttes for å utvikle løsningen. Elin Synnøve Solberg har jobbet med organisasjonsarbeid tidligere, så vi håper vi kommer til å utfylle hverandre til det beste for løsningen av hovedprosjektoppgaven.

Vi valgte å jobbe med dette prosjektet fordi vi synes det er interessant å jobbe med avanserte Internett-baserte løsninger. Vi så at deler av oppgaven var meget utfordrende og at vi dermed kunne lære mye underveis i prosjektperioden. Ønsket om å utvikle noe som kunne tas i bruk ved HiG var også et avgjørende element i vårt valg av oppgave.

Vi måtte lære en del mer om internetteknologier, PHP, DHTML, JavaScript og SQL enn det vi hadde lært i de forskjellige fagene vi hadde tatt fra før. Vi erfarte at mye av kodingen kunne gjøres lettere ved å benytte seg av mer avanserte og gjennomtenkte SQL-spørringer. Videre var vi nødt til å bli vant til å jobbe sammen som en prosjektgruppe, og kommunisere med oppdragsgiver og veileder. Vi måtte finne metoder for versjonskontroll og backup-løsninger for å kunne jobbe flere på det samme prosjektet, dette lærte vi mye om underveis i prosjektet.

## 1.6 Arbeidsformer

Vi jobbet for det meste samlet på tildelt grupperom ved HiG, ved enkelte anledninger når arbeidsoppgaver var klart fordelt jobbet vi noe hver for oss med kodearbeid. Vi hadde morgenmøter med status på hvordan vi lå an med arbeidet på oppgaven og fordeling av eventuelt nye arbeidsoppgaver. Når enkeltdelegere i prosjektgruppa stod fast med noe i tildelte arbeidsoppgaver, diskuterte vi dette sammen til vi kom frem til en god løsning. Når vi kom over problemer som vi ikke klarte å løse selv tok vi kontakt med oppdragsgiver, veileder eller andre fagpersoner ved HiG for å få tips og råd for å komme videre med utviklingsarbeidet.

Statusmøter ble holdt en gang i uka, stort sett fredager kl. 1430, eller rett etter møte med veileder. Det ble levert tre statusrapporter til veileder i løpet av våren slik at han kunne følge med på fremdriften. Møte med veileder ble avholdt hver fredag kl. 12.30, etterfulgt av møte med oppdragsgiver når veiledning var ferdig. Frode Haug var som kjent både veileder og representant for oppdragsgiver for vår prosjektgruppe.

Vi delte prosjektet inn i fire faser: planleggingsfase med forprosjekt, analysefase, design og utviklingsfase og avslutningsfase. Under Design og utviklingsfasen fikk oppdragsgiver/veileder beskjed etter hvert som hver del av systemet ble lagt ut for testing. Korrigering ble gjort og deløsningen testet på nytt.

- Planleggingsfase med forprosjekt:  
Startet denne fasen med å innrede grupperommet. Satte opp arbeidsstasjoner og server, og konfigurerte alle maskinene så de fungerte mot Internett og vårt interne nettverk. Videre installerte vi all nødvendig software på alle arbeidsstasjoner. Serveren var ferdig installert fra før, men her var vi nødt til å sette opp en backuprutine (cron). Vi satte også opp tids- og ressursplaner. Dette var hovedpunktene i forprosjektet, andre gjøremål i denne fasen er dokumentert i forprosjektrapporten.
- Analysefase:  
Tok utgangspunkt i eksisterende kravspesifikasjon fra forrige TVIL-hovedprosjektgruppe og reviderte deler av denne. Vi måtte sette oss godt inn i denne for å finne ut av hva oppdragsgiver krevde av systemet og vår løsning av prosjektoppgaven. Videre ble det eksisterende databasedesignet vurdert og enkelte endringer på dette ble gjennomført.
- Designfase og utviklingsfase:  
Databasen ble opprettet og grovdesign av brukergrensesnitt ble skissert i denne fasen. Koding av de forskjellige modulene startet.
- Avslutningsfase:  
Jobbet med prosjektrapporten og forberedelser til framføringen av hovedprosjektet. En del finjusteringer og tilpassinger av modulene. Systemet ble installert på skolens webtjener slik at systemet ble tilgjengelig ved HiG.

Vi har valgt å bruke en inkrementell utviklingsmodell, da vi ville gjøre modulene helt ferdig før vi begynte på en ny. Vi valgte samtidig å dele utviklingsoppgavene mellom oss slik at deler av modulene kunne utvikles parallelt mellom gruppemedlemmene.

## 1.7 Organisering av rapporten

Rapporten er delt inn i ti kapitler. Den starter med en innholdsfortegnelse med sideanvisning som er veldig detaljert og viser absolutt alle delinndelinger under et kapittel, dette gjør rapporten lett å finne frem i. En figurliste med sideanvisning er lagt inn bak innholdsfortegnelsen. Alle sidene fra og med første kapitlet, er nummererte med nummerangivelse nederst på hver side. Vi har laget en ordliste over datatekniske ord og uttrykk som forekommer i rapporten for å gjøre leseprosessen for en ikke datakyndig leser lettere. Ellers har vi prøvd å skrive det meste i et mer lettfattelig og “norskformulert” språk, men enkelte plasser er det ikke til å unngå at formuleringene blir vel datatekniske. Dere som leser rapporten får bare bære over med oss på akkurat dette punktet. I første kapitlet er det også en innledning der vi beskriver prosjektoppgaven, prosjektmål, problemområde, avgrensninger og hovedprosjektgruppas arbeidsformer og bakgrunn. Det andre kapitlet inneholder kravspesifikasjonen. Dette er en utvidet/forbedret utgave av den første TVIL-hovedprosjektgruppas kravspesifikasjon. Som tredje kapittel kommer analysedelen, her beskriver vi hva systemet skal dekke og gjør noen vurderinger omkring dette. Design av systemet blir omtalt i kapittel fire. Her supplerer vi den forklarende teksten med mange skjermbildeeksempler. Implementasjonskapitlet omhandler både valg av programmeringsspråk, metoder og verktøy vi har benyttet. Kvalitetssikring, testing og realisering blir beskrevet i kapittel seks. En diskusjon om måloppnåelse blir gjennomført i neste kapittel før vi kommer med vår konklusjon. Her kommer vi også med noen betraktninger om hva vi har erfaringshøstet gjennom prosjektfasen. Til slutt kommer en opplisting av verktøy, litteraturliste og internettressurser vi har brukt. Kapittel ti inneholder bare vedlegg, disse er blant annet script, logg, Gantt-skjema og alle referater.

Rapporten er foruten å være vår hovedprosjektdokumentasjon, også ment å være utfyllende og forklarende beskrivelse av systemets benyttelsesområde for de som ønsker å benytte seg av avanserte valgfunksjoner eller den eller de som skal være administrator for systemet.

Vi håper at lesere av rapporten oppfatter den som gjennomgående entydig og forklarende. Vi har prøvd å dokumentere mest og best mulig med tanke på en eventuell videreutvikling av vår delløsning til å bli en totalløsning ved hjelp av en tredje TVIL-hovedprosjektgruppe.

Rapporten ble skrevet samtidig som det siste utviklingsarbeidet pågikk.

## 1.8 Terminologi

Vi har brukt betegnelsen *bruker* en del steder i rapporten, dette må ses i sammenheng med hvilken del av systemet som beskrives. Lærer beskrives som bruker når vi beskriver lærermodulen, student kan også beskrives som bruker når vi skriver om studentmodulen osv. Det kan også være andre enn lærere som lager tester på systemet og studenter som besvarer disse. for enkelhets skyld har vi kalt disse brukerne for lærere og studenter.

## 1.9 Praktisk

Vi har brukt Times med størrelse 8 som standard font gjennom hele dokumentet. Ved kodeeksempler og sql-script har vi brukt fonten Typewriter i blå utgave i størrelse 7. Marger: topp 25mm, bunn 15mm, venstre 25 og høyre 15mm på A4 ark.

Løsningen ligger tilgjengelig på internett for testing. Innlogging for superbruker finnes på <http://super.tvil.net>, innlogging for student finnes på <http://student.tvil.net> og innlogging for lærer finnes på <http://login.tvil.net> Du vil trenge brukernavn og passord for å logge på. Superbruker har brukernavn "superuser" og passord "superpwd". Det er opprettet en lærer med brukernavn "hov\_pros" og passord "2001". For å teste studentmodulen kan du logge på som "stu\_dent" med passord "tvileren".

# Kapittel 2

## Kravspesifikasjon

### 2.1 Brukerbeskrivelse

#### 2.1.1 Omgivelser

Systemet skal kjøres på en vanlig internettjener og være tilgjengelig for brukerne via Internett.

Internettjeneren skal kjøre Apache webserver, med støtte for MySQL databaser og PHP. Dette er tilgjengelig for Linux og Windows NT/2000. Vi optimaliserer løsningen for Linux operativsystem på serversiden. Brukerne skal kunne bruke sine nettlesere til å aksessere systemet. Vi forutsetter imidlertid at de har forholdsvis nye versjoner av nettlesere. Det kreves av systemet at brukerne har Microsoft Internet Explorer 5.0 eventuelt Netscape eller nyere. Systemet krever også at brukerne har tilknytning til Internett.

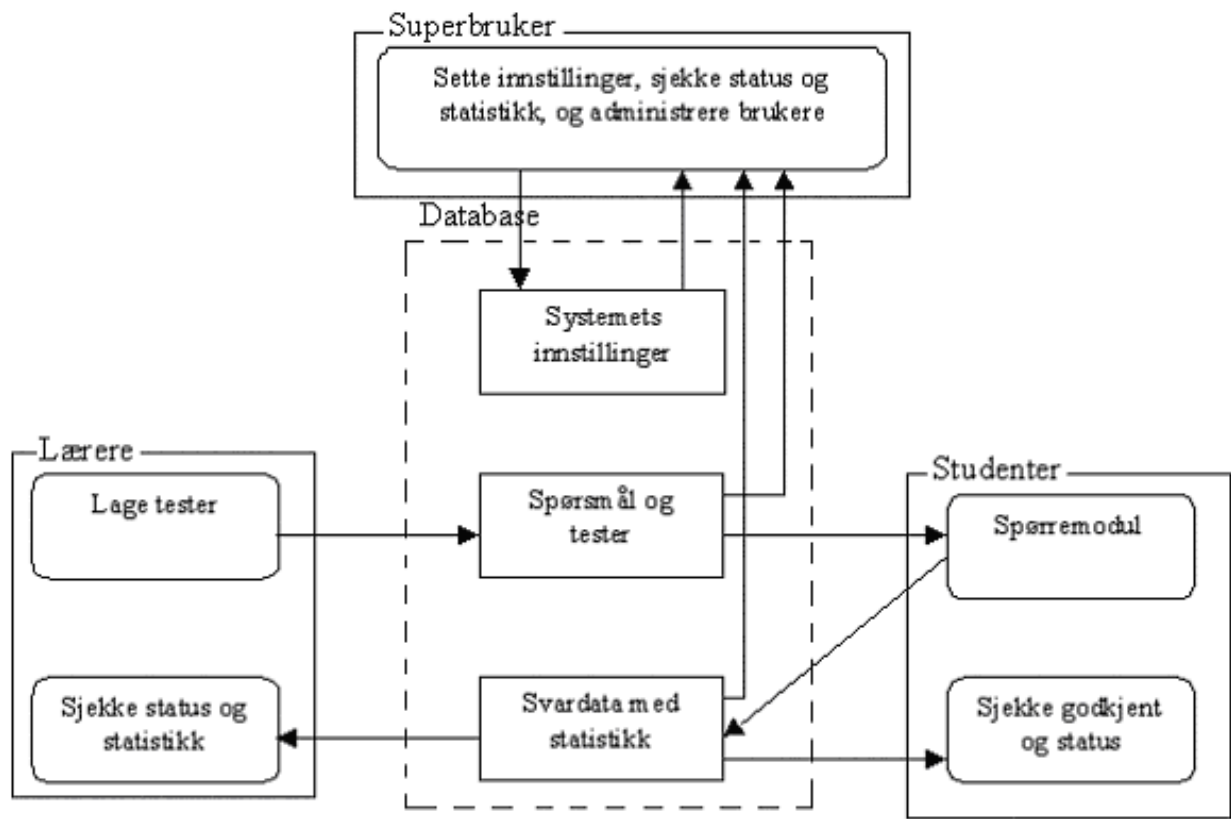
#### 2.1.2 Systemet har fire grupper av brukere.

- *Driftspersonell*: Den første gruppen er de som har ansvar for å installere og vedlikeholde systemet. Disse må kjenne til systemet og vi forventer at de lærer det de trenger gjennom rapporten fra prosjektet. Disse personene arbeider allerede med IT-drift i organisasjonen. De har allerede kjennskap til de omliggende systemene og drifter disse.
- *Superbruker*: Den andre gruppen er superbrukerne. De har spesiell tilgang til systemets konfigurering. Det kreves også av disse brukerne at de har en del kjennskap til systemet. Det forventes at en superbruker kan lære opp og bistå de andre brukerne.
- *Lærere*: Den tredje gruppen er de som lager tester på systemet. Dette er hovedsakelig lærere ved høyere utdanningsinstitusjoner, og de er mennesker med høyere utdanning. De har varierende kunnskap om systemet og IT generelt. Systemet skal lages slik at disse ikke skal behøve noen opplæring for å ta det i bruk. De må selv ta ansvar for at testene de lager er pedagogisk riktige eller at de måler studentenes kunnskapsnivået på en riktig måte. De vil ikke få hjelp av systemet med å bygge opp gode tester. Lærerne må imidlertid ha tilgang til Internett for å ta systemet i bruk.
- *Studenter*: Den siste gruppen er de som besvarer testene. Dette er studenter på forskjellige fagområder og nivåer. Vi forutsetter at de har kjennskap til bruk av Internett og generell kunnskap om bruk av nettleseren. Systemet skal være selvforklarende for disse. Det er de som lager testene som er ansvarlige for at denne brukergruppen er informert om testene og om hvordan testene skal besvares.

#### 2.1.3 Funksjon

Systemet har fem moduler. Den første modulen er å *endre innstillinger*. Denne modulen brukes av superbrukeren på systemet. Her kan han legge inn nye brukere, sette rettigheter og innstillinger, slette gamle data, og se forskjellig statistikk om systemet.

Den andre modulen er den som brukes av lærerne til å *lage og redigere tester*. Her legger lærerne inn fagenes navn og en del informasjon om testene. De kan velge blant forskjellige innstillinger angående tester, slik som hvem som kan redigere og kopiere tester. Videre registreres hvor mange prosent som må være riktig for at



Figur 2.1: Hovedtrekk i systemet

en test skal være bestått. De velger spørsmålstyper og parametre til disse, og legger inn spørsmålstekst og svaralternativer, eventuelt bilder, lyd, video, dokumenter og programmer.

Den tredje modulen brukes av lærerne for å få ut *statistikk fra testene*. De kan se hvilke studenter som har bestått tester der dette er vesentlig for undervisningen. De kan også slette statistikk.

Det finnes også en *spørremodul* for studentene, hvor spørsmålene er tilgjengelig med adgangskontroll. Studentene må selv registrere seg med brukernavn, passord, navn, klasse og e-postadresse første gang de skal benytte systemet. Her får de informasjon om testen og besvarer testen ved å flytte felter på skjermen, peker på bilder, utfører valg, fyller inn tekst eller utfører flervalgsalternativer. Ved utført test får de også respons på om testen ble godkjent eller ikke.

I den siste modulen kan studentene få *se statistikk* over klassens nivå på testen. Studenten kan dermed sammenligne sine resultater med klassenivået.

I figur 2.1 ser vi en enkel skisse av systemet. Brukerne av de forskjellige modulene er tegnet i figuren. Systemets fem moduler er rektangler med runde hjørner. Dataene i systemet ligger i databasen. Pilene viser flyt av data.

#### 2.1.4 Operasjon

Systemet bør være tilgjengelig så mye som mulig. Dette er avhengig av internettjeneren systemet kjøres på og henger sammen med oppetiden for organisasjonens andre internettjenester. Ved feil i systemet må dette rettes av de som er ansvarlig for vedlikehold av internettjenester for øvrig. Systemet skal selv takle feil i input og klare å gjenopprette seg selv etter en driftsstans i systemene. Sikkerhetskopiering av dataene i systemet tas gjennom de normale rutiner for sikkerhetskopiering av data i internettjeneren.

### 2.1.5 Aspekter omkring livssyklus

Dataene i systemet skal vedlikeholdes av superbrukeren. Det er han som avgjør når data skal slettes og hvilke brukere som har tilgang. Den tekniske driften og ansvaret for at systemet er tilgjengelig, har de som fra før driver internettjenester ved organisasjonen. Systemet skal være flyttbart, gjenbrukbart og plattformuavhengig. Alt prosjektgruppa utvikler skal dokumenteres, og samtlige filer skal kommenteres i kildekoden.

### 2.1.6 Ytelse

Ytelsen til systemet vil være begrenset av internettjeneren, nettleseren og internettforbindelsen. Likevel kreves det at svartiden fra databasen og frem til internettforbindelsen ikke skal overstige 5 sekunder.

### 2.1.7 Begrensninger

Systemet skal ikke føre til noen nye investeringer i programvare eller maskinvare.

Løsningen baserer seg på gratis programvare og bør kjøres på Linux Apache webserver. Både operativsystemet, webserveren og databasen er gratis. Systemet kan imidlertid også kjøres på Microsoft Windows NT/2000 med Apache og MySQL for Windows.

### 2.1.8 Antagelser

Vi forutsetter at alle brukerne har tilgang til Internett og en nettleser som støtter CSS, JavaScript, Java APPLETS , DHTML og HTML 4.0. Løsningen er optimalisert for Microsoft Internet Explorer 5.0 og Netscape Navigator 4.7. Organisasjonen må være tilknyttet Internett og ha en Internettjener som har støtte for PHP4 og MySQL-database.

### 2.1.9 Hjelpesfunksjon

Systemet skal ha en enkel hjelpesfunksjon som skal finnes i hele systemet. Denne skal gi forklaringer til alle valg og funksjoner samt ha beskrivelse for hvordan man går frem for å utføre de vanligste oppgavene. En mer utfyllende brukerveiledning skal lages til slutt, og denne veiledningen vil fokusere på lærernes oppgavegenerator.

## 2.2 Funksjonell kravspesifikasjon

### 2.2.1 Funksjonell struktur og tverrelasjoner

Figur 2.2 viser dataflyt mellom brukerne og systemet.

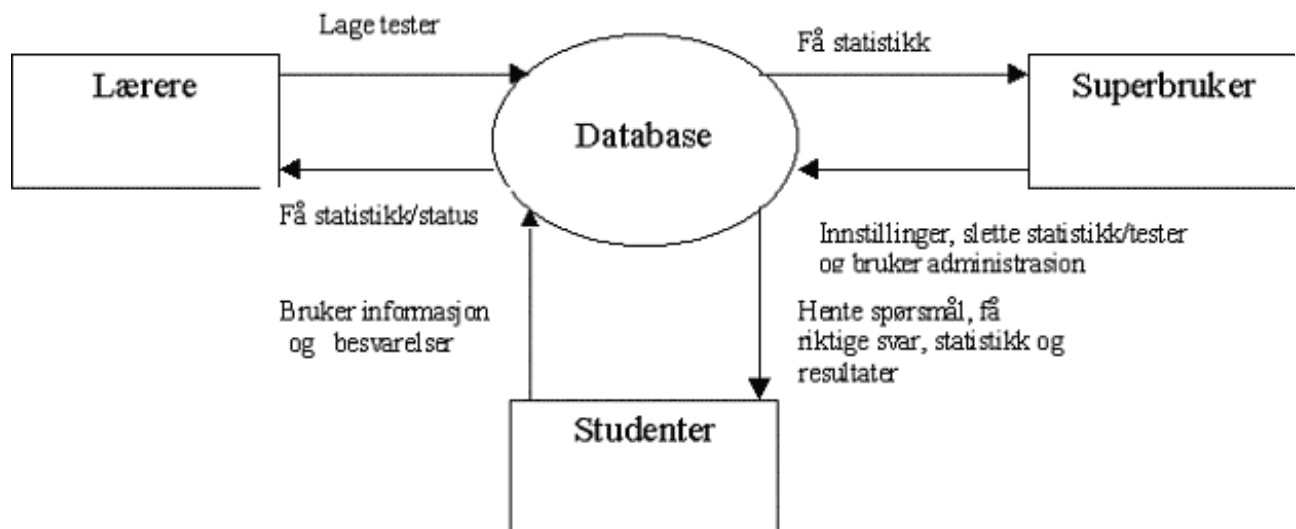
Lærerne lager tester. Her må de legge inn data om fagene og testene, de må sette innstillinger for de forskjellige testene, velge spørsmålstyper og legge inn spørsmål- og svardata. Dette gjøres i et eget passordbeskyttet område i løsningen. Tilbake kan de få statistikk over studentenes svar og sjekke hvilke studenter som har fått godkjent eller ikke fått godkjent testene. Superbrukeren kan sette forskjellige innstillinger som styrer hva lærerne kan gjøre. Superbrukeren kan slette gamle tester og statistikk. Det er superbrukeren som legger til og fjerner lærere. Superbrukeren legger også inn klasser og sletter dem. Superbrukeren kan også se en del statistikk for systemet. Studentene jobber også mot systemet via Internett. Studentene kan se spørsmålene og svare på disse. Studentene får tilbakebakemelding i form av oppgaveretting og kan be om statistikk om hvordan klassen ligger an.

I figur 2.3 ser vi en mer detaljer bilde av dataflyt i systemet. Vi har delt systemet i 17 funksjoner som er forklart under figuren.

#### 2.2.1.1 Superbruker:

1. Endre innstillinger. Her legger superbrukeren inn forskjellige innstillinger til systemet, de lagres i tabellen innstillinger i databasen.
2. Oppdatere klasselister. Her legger superbrukeren til eller sletter klasser i listen over klasser. Disse ligger i tabellen klasse i databasen.





Figur 2.2: Dataflyt Diagram 0

3. Legge til eller slette brukere. Her legger superbrukeren til nye lærere eller sletter gamle lærere i tabellen lærer i databasen.
4. Slette data. Her kan superbrukeren gå inn i databasen og slette gamle data. Han kan også sette datoer når ting skal slettes automatisk.
5. Overføre fag til annen lærer. Her kan superbrukeren gå inn og ta et fag som tilhører en lærer og gi det til en annen.
6. Se statistikk. Her kan superbrukeren se et utvalg av statistikk fra de forskjellige databasene og systemet.

#### 2.2.1.2 Lærer:

7. Lage, slette eller endre fag. Her kan læreren gå inn og lage nye fag, slette gamle fag eller endre innstillingene til de fagene som han eier. Her legges alle dataene om fag inn.
8. Gi andre adgang til sitt fag. Her bestemmer læreren hvem som får lesetilgang til sine fag og hvem som får tilgang til å endre fagenes data, samt åpne for kopiering av en test.
9. Lage, slette eller endre tester. Her kan læreren lage nye tester, slette gamle eller endre innstillinger til sine tester.
10. Kopiere eller koble til en test. Her kan læreren velge en av de andre lærernes tester og kopiere denne dersom eieren av testen har gitt rettigheter til det. Han kan også koble seg til en test og gi sine studenter adgang til å utføre testen, dersom læreren som eier testen har gitt tilgang til testen.
11. Lage, slette eller endre spørsmål. Her kan læreren lage nye, slette gamle eller endre de spørsmål som er hans egne eller som andre har gitt han tilgang til.
12. Kopiere spørsmål. Her kan læreren kopiere andres spørsmål og legge til de blant sine egne, dersom de har gitt han tilgang til det.
13. Lese av en Internett-adresse til testen. Her kan læreren se hvilken internettadresse testen har, han kan også se adressen til et fag med alle dets tester.
14. Se statistikk for et fag. Her kan læreren velge blant forskjellig statistikk om sitt fag og dets tester.

### 2.2.1.3 Student:

15. Lese test og legge inn svar. Her får studenten se testen og kan legge inn sine svar.
16. Få resultat. Når student har svart på en test, blir testen rettet av systemet, og studenten får se resultatet som er oppnådd på aktuelle test.
17. Se statistikk eller status. Studentene kan se klassens nivå på testen, og kan sammenligne sitt oppnådde resultat mot forventet verdi, grense for bestått test eller klassens gjennomsnittresultat.

## 2.2.2 Data spesifikkasjon

### 2.2.2.1 Data rammeverk

I figur 2.4 har vi lagt en forenklet oversikt over dataene i systemet.

Vi ser av figuren at en lærer kan ha ingen eller flere fag. Men et fag kun kan eies av en lærer. Et fag kan ha ingen eller flere tester. En test må tilhøre et fag, men den kan godt høre til flere fag, hvis andre lærere enn eieren av testen, kobler seg til testen. En test kan ha ingen eller flere spørsmål, men et spørsmål må høre til en test. Til et spørsmål hører et sett med spørsmålsdata. Disse dataene vil avhenge av type spørsmål som er definert. Til spørsmålsdataene kan det tilhøre ingen eller flere svardata, som er studentenes svar. En test kan ha ingen eller flere klasser og en klasse kan høre til ingen eller flere tester. En klasse kan ha ingen eller flere studenter, og en student kan høre til en eller ingen klasse. En student kan ha ingen eller flere testsvardata, disse må tilhøre en test og en student. En test kan ha ingen eller mange testsvardata. Her står om studentene har fått godkjent status eller ei, og antall forsøk som er benyttet av studenten. I tillegg kommer en tabell med innstillinger for systemet.

### 2.2.2.2 Data input

Superbrukeren og læreren legger inn sine data via nettleseren. De må identifisere seg med et brukernavn og passord for å få adgang til sin del av løsningen. Studentene autoriserer seg med brukernavn og passord for å utføre en test.

### 2.2.2.3 Data output

Superbrukeren får se statistikk for systemet i den samme delen som brukes til input. Lærerne får se statistikk for testene i sin del av systemet. Studentene ser statistikk og godkjentlister i en egen del av løsningen.

### 2.2.2.4 Tverrfunksjonelle datadefinisjoner

Alle moduler skal jobbe mot det samme sentrale settet med data. De må derfor bruke de samme datatypene. Detaljene omkring datatypene og modulenes input og output er beskrevet i punkt 2.2.4.

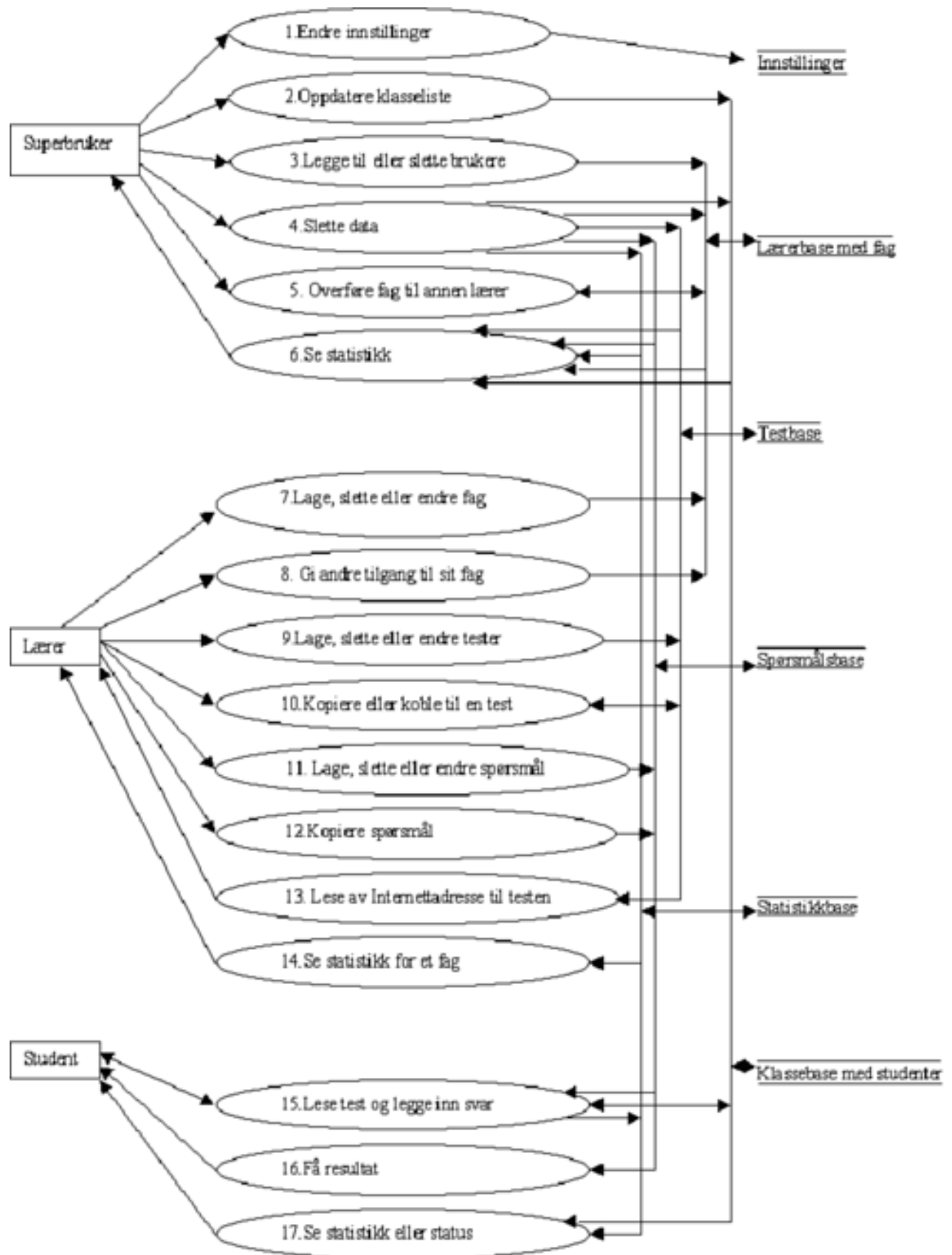
## 2.2.3 Overordnede operasjonelle systemkrav

### 2.2.3.1 Normal operasjon

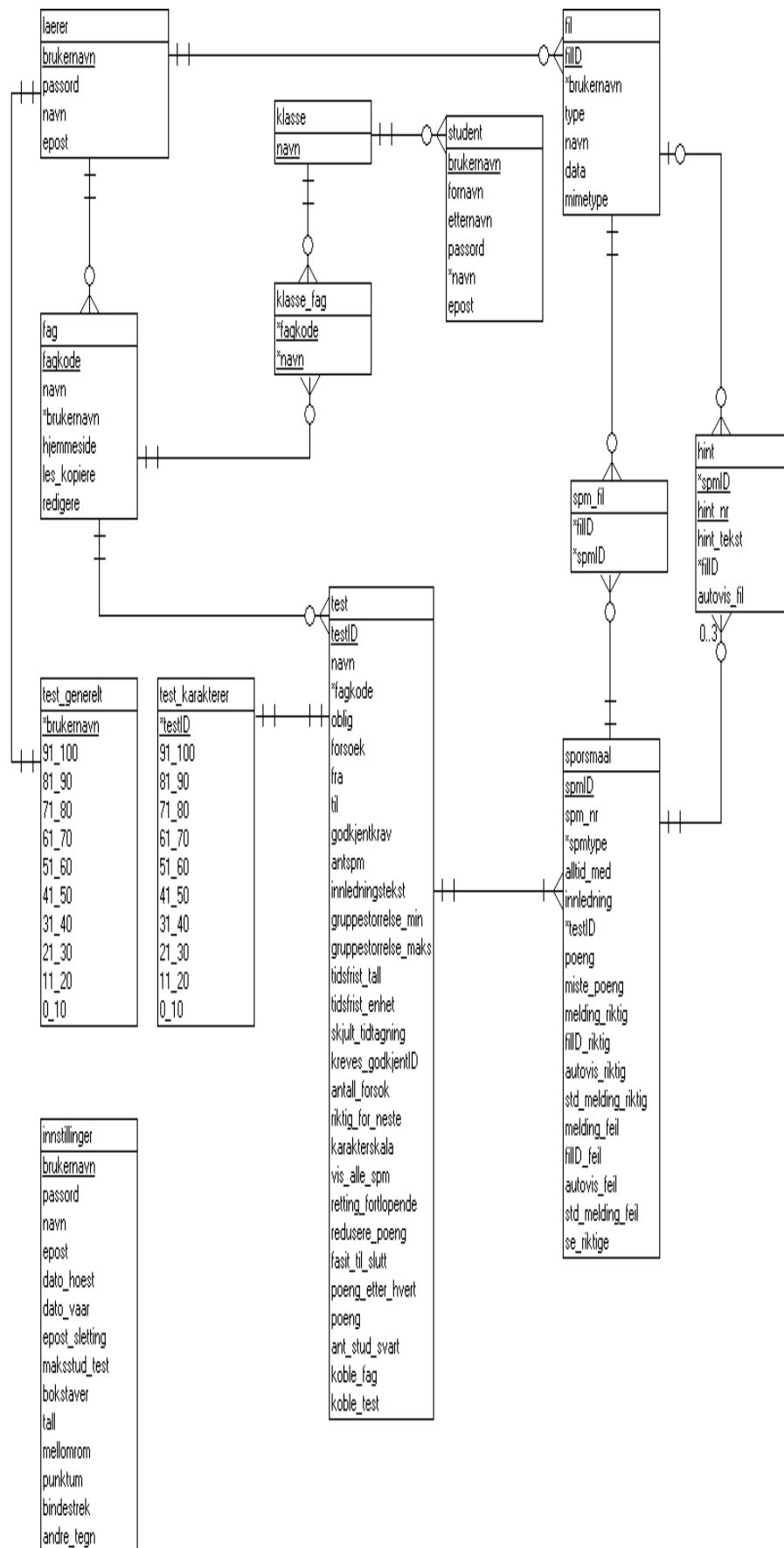
**2.2.3.1.1 Modus og kontroll** Når en bruker aksesserer løsningen, starter en sesjon. Hvis brukeren er inaktiv for lenge, blir sesjonen avbrutt. Systemet bruker kun maskinvare-ressurser når en sesjon er aktiv.

**2.2.3.1.2 Ytelse** Ytelsen til systemet vil være begrenset av internettjeneren, nettleseren og internettforbindelsen. Likevel kreves det at svartiden fra databasen og frem til internettforbindelsen ikke skal overstige 5 sekunder.

**2.2.3.1.3 Sikkerhet** Systemet skal takle alle typer feil i input. Superbrukermodulen og modulen for å lage tester skal ha adgangskontroll med brukernavn og passord. Studentene logger på med brukernavn og passord for å ta en test. Sikkerhet rundt installering og setting av brukerrettigheter, ivaretas av de som installerer systemet.



Figur 2.3: Dataflyt-Diagram 1



Figur 2.4: Forenklet ER - diagram

**2.2.3.1.4 Oppstart og nedtagning** Systemet er nytt og skal installeres på en internettjener. Systemet skal automatisk starte opp når internettjeneren startes og tas ned når internettjeneren tas ned. De som drifter internettjeneren kan avbryte systemet.

**2.2.3.1.5 Tilgjengelighet** Tilgjengelighet avgjøres av internettjeneren. Systemet skal ha samme oppetid som internettjeneren.

**2.2.3.1.6 Innebygde tester** Systemet kontrollerer all input for feil da alle data fra brukeren valideres. Systemet har avanserte valg som er standard innstilte og defaulte verdier, slik at de gir et godt resultat selv om brukeren ikke endrer disse valgene. Systemet kontrollerer også at forbindelsen til databasen er tilgjengelig og at lesing og skriving til databasen er feilfri. All sletting og overskriving av data skal bli varslet.

## 2.2.3.2 Operasjon i feilsituasjoner

**2.2.3.2.1 Feilrapportering** Feil i systemet, ved for eksempel feil i en spørring mot databasen, sendes til superbrukeren. Brukerne får melding dersom de prøver å gi feil input.

**2.2.3.2.2 Gjenervervelse etter feil** Etter å ha forsøkt å gi feil input skal brukerne bli bedt om ny input. Etter en datakrasj skal systemet starte opp samtidig med Internettjeneren. Det brukerne ikke hadde lagret når systemet krasjet, vil gå tapt.

**2.2.3.2.3 Sikkerhet** Feil skal ikke påvirke sikkerheten. Det skal ikke føre til tap av lagret data ved ukontrollert nedkjøring. Hvis en del av systemet er nede, er hele systemet ute av drift. Dersom systemet har vært nede, skal alle logge seg på på nytt.

**2.2.3.2.4 Ytelse** Småfeil, feil i input og brukerfeil skal ikke påvirke ytelsen.

## 2.2.4 Funksjonelle krav

### 2.2.4.1 Funksjonelle krav til lærernes spørsmålsmodul

**2.2.4.1.1 Input** Før læreren kan lage en test, må han opprette et fag. Han kan også redigere dataene til sine fag. Et fag må ha fagkode og fagnavn og knyttes opp mot de klassene som skal ta faget. Dersom læreren har en hjemmeside for seg selv, eller faget eller fagene testen er knyttet til, kan internettadressen linkes til testsiden. Læreren kan velge om en eller flere lærere skal ha tilgang til sine fag. Læreren skal velge om andre lærere skal kunne lese eller kopiere tester som tilhører dette faget. Læreren kan velge om andre lærere skal få editere tester i dette faget. Han kan velge om ingen, alle eller bare en gruppe skal få editere testene. Hvis det er gruppe som kan editere testene, må han velge hvilke lærere som er i denne gruppen. Under følger detaljert databeskrivelse:

Fagkode	Tekstfelt, 32 tegn, superbruker definerer tillatte tegn.
Fagnavn	Tekstfelt, 64 tegn.
Hjemmeside	Tekstfelt, 256 tegn.
Lese/kopiere	Boolean.
Editere av andre	3 valg.
De som kan editere	Lærere.
Klasser	Hvilke klasser som tar faget.

Når læreren har opprettet et fag, kan han opprette tester som blir tilhørende dette faget. En test har et navn og er enten en obligatorisk test eller en statustest. Alle statustester er frivillige. Læreren angir hvor mange forsøk studenten har dersom testen er en obligatorisk test, og hvor mange prosent som må være riktig for å få "bestått" status. Deretter angir læreren tidsrommet testen er tilgjengelig i. Han må også oppgi hvor mange spørsmål som totalt skal vises i testen, og dersom tilfeldig trekning av spørsmål er valgt, må spørsmålene som alltid skal være med i testen, avmerkes eller avhukes.

Det skal også finnes en undermeny med avanserte valg som er forhåndsinnstilt, slik at kun de som føler seg trygge med systemet behøver å gå inn for å endre på dem. De mest sannsynlige og brukte valg er her satt til

default. Dersom det er en obligatorisk test, kan læreren sette minimum og maksimum størrelse på gruppa som leverer svar sammen. Han kan også sette en tidsfrist som angir hvor lang tid studentene kan bruke på testen. Selv om læreren ikke har valgt å bruke dette, kan han likevel velge at systemet tar tiden på hver student, kun for statistikkens skyld. Han kan også velge at en annen test må være godkjent, før aktuelle test kan utføres. Han kan angi hvor mange forsøk studentene får per spørsmål. Han kan bestemme om studentene må svare riktig på et spørsmål for å få gå videre til neste spørsmål. Etter testen vil studenten få karakter. Læreren kan bestemme om den skal gis i halve karakterer eller i tideler. Noen av spørsmålene i testen kan være tilfeldig valgte, dvs. at kun noen av spørsmålene i testen er faste mens noen blir tilfeldig valgt for hver pålogging til testen. Læreren kan velge å vise alle spørsmålene; både de faste og hele utvalget av tilfeldig valgte spørsmål, dersom det ikke er en obligatorisk test. URL for en slik utvidet test skal da oppgis. Læreren kan også velge om spørsmålene skal rettes etter hver svarangivelse eller etter hver testutførelse. Oppnådd poengsum eller score kan vises etterhvert som testen blir utført, dette vil naturligvis ikke være tilfelle dersom testen er en obligatorisk test. I tillegg til dette kan læreren velge å kopiere andres tester eller koble seg til en annens test. En lærer skal kunne editere en test eller et spørsmål også mens testen er åpen for studentene. Dette kan føre til at en student svarer på et spørsmål, som ved svarangivelse er forandret. Studentene skal også kunne printe en fullversjon av spørsmålene tilhørende en test.

En lærer skal kunne slette en godkjentliste og alle oppsamlede data knyttet til en test. Han kan også øke antall svarforsøk for en student eller en gruppe med studenter og manuelt endre godkjentstatus for studenter.

Dataene for en test:

Navn	Tekstfelt, 256 tegn.
Status/oblig	Boolean.
Antall forsøk	Integer.
Dato fra	Dato/tid.
Dato til	Dato/tid.
Godkjent prosent	Integer, 0-100.
Antall spørsmål	Integer.
Innledningstest	Tekstfelt, ingen begrensning på tegn. Bilder, lyd, Video.
Størrelse, gruppe, min	Integer.
Størrelse, gruppe max	Integer.
Tidsfrist tall	Integer.
Tidsfrist type	Tekstfelt, ett tegn.
Tidtaking statistikk	Boolean.
Test forutsatt godkjent	Annen test.
Antall forsøk, spørsmål	Integer.
Må svare riktig, videre	Boolean.
Karakter, hel/halv/tidel	Boolean.
Vis alle	Boolean.
Fortløpende retting	Boolean.

Data til hvert spørsmål:

Spørsmålstype	Integer.
Spørsmålstekst	Tekstfelt, ingen grense på tegn.
Bilde	.gif eller .jpeg fil.
Lyd, video, program	Binærfil.
Være med, ja/nei	Boolean.
Poeng	Integer.

I tillegg kommer data som er spesifikke for denne spørsmålstypen. Disse blir beskrevet i kapittel 2.2.5.

**2.2.4.1.2 Prosessering** Lærernes valg lagres i den sentrale databasen. Det kontrolleres at valgene er gyldige.

**2.2.4.1.3 Output** Lærerne får tilbakemelding på det de gjør.

**2.2.4.1.4 Feilrapportering** Lærerne får beskjed dersom de prøver å gjøre noe som er feil. Det skal ikke få systemet til å gå ned.

**2.2.4.1.5 Gjenervervelse etter feil** Dersom systemet går ned må lærerne logge seg inn på nytt. Den siste enheten som læreren arbeidet på, et fag, en test eller et spørsmål, vil ikke være lagret.

**2.2.4.1.6 Sikkerhet** Lærerne må oppgi brukernavn og passord for å kunne se, lage eller editere tester.

## **2.2.4.2 Funksjonelle krav til lærernes statistikkmodul**

**2.2.4.2.1 Input** De velger hvilke fag eller test de vil se statistikk for og type statistikk. De kan bestemme hvilken statistikk som skal være tilgjengelig for studentene.

**2.2.4.2.2 Prosessering** Modulen henter data fra databasen og presenterer det for lærerne.

**2.2.4.2.3 Output** For fag kan lærerne få følgende statistikk:

- Godkjentliste. Viser alle studenter med godkjent-/ikke-godkjent-status i ett fag.
- Testoversikt. Viser hvor mange forsøk studenten har brukt per test. Bruk av hint.
- Gjennomsnittlig poengsum per test.

For tester kan lærerne få følgende statistikk:

- Hvor mange har svart riktig på et spørsmål.
- Antall svarforsøk per spørsmål.
- Vise andel rett/galt på hvert spørsmål.
- Gjennomsnittlig poengsum på et spørsmål.

Her er det fortsatt åpent for nye typer statistikk.

Statistikken avhenger av spørsmålstypen, se kapittel 2.2.5.

**2.2.4.2.4 Sikkerhet** Lærerne må oppgi brukernavn og passord for å se statistikk.

## **2.2.4.3 Funksjonelle krav til studentenes svarmodul**

**2.2.4.3.1 Input** For å finne riktig test kan studenten velge fra en liste over fag, og videre en test fra en liste over tester. Studenten autoriserer seg ved brukernavn og passord. Den første gangen studenten bruker løsningen må han velge et brukernavn og passord, samt registrere navn, klasse og e-postadresse. Dersom studenten glemmer passordet sitt, kan han få dette tilsendt på mail. Dataene for spørsmål er beskrevet i kapittel 2.2.5.

**2.2.4.3.2 Prosessering** Systemet retter svarene fortløpende og lagrer resultatene når studentene er ferdige med en test. Avhengig av hvilken type retting lærerne har valgt og hvilken testtype studenten skal gjennomføre, får studenten tilbakemelding i form av oppnådd score. Studentene kan få retting mellom hvert spørsmål, eller de kan få rettet sin besvarelse til slutt.

**2.2.4.3.3 Output** Det kan være at lærerne har lagt inn tips til hvert spørsmål. Da vil studentene få disse. Dette er nærmere beskrevet i kapittel 2.2.5. Til slutt vil studentene få poengsum på testen, hvis det er en statustest. Hvis det er en obligatorisk test vil studentene få vite om de har fått godkjent eller ikke, samt visning av oppnådd poengsum. Poengsummer og godkjent/ikke godkjent sendes til den sentrale databasen for systemet.

**2.2.4.3.4 Feilrapportering** Studenten får feilmelding dersom deler av systemet ikke er tilgjengelig.

**2.2.4.3.5 Gjenervervelse etter feil** Dersom systemet går ned, må studenten logge seg på og besvare på testen på nytt.

#### **2.2.4.4 Funksjonelle krav til studentenes statistikkmodul**

**2.2.4.4.1 Input** Studenten kan se statistikk over hva læreren har bestemt skal være tilgjengelig for dem, som for eksempel godkjentliste. Godkjentlistene vil bare inneholde godkjentstatus for studentene. Studenten har adgang til å se statistikk over godkjent-status for alle fag en tar tester i . Studenten må velge hvilket fag og hvilken test han vil se statistikk for. Studenten kan også få se egen poengsum eller oppnådd score i gjeldende test, få oppgitt fagdeltagernes gjennomsnittlige score og se minimum og maksimum oppnådd resultat på aktuelle test.

**2.2.4.4.2 Prosessering** Statistikkmodulen leser fra den sentrale databasen og tabellen test og presenterer statistikken i nettleseren. Studentene vil få statistikk. For en nærmere beskrivelse av statistikken: se lærernes statistikkmodul i 2.2.4.2.

Output vil være; oppnådd score i gjeldende test, fagdeltagernes oppnådde test-score, obligatoriske testers terskel for godkjentstatus og minimum og maksimum oppnådd resultat på aktuelle test.

**2.2.4.4.3 Sikkerhet** Studenten logger seg på systemet med brukernavn og passord for å avlese statistikk.

#### **2.2.4.5 Funksjonelle krav til Superbrukerens verktøy**

**2.2.4.5.1 Input** Superbrukeren kan sette en del innstillinger for systemet. Han kan legge til og fjerne lærere og klasser, og overføre fag til en ny lærer. Han kan også slette fag med tilhørende tester. Han kan bestemme antallet og hvilke tegn som er tillatt i fagkode. Det kan være store eller små bokstaver, tall, mellomrom og eventuelt andre tegn superbrukeren ønsker. Han kan nullstille godkjentlister og skrive eksisterende oppnådd godkjentlister fra tidligere semestre til flate filer. Han kan sette datoer for når han vil bli påminnet via e-post om at det er tid for å legge til eller slette klasser og godkjentlister. Dette gjøres to ganger hvert år; for høstsemesteret omlag 1.august og vårsemesteret omlag 1.januar. Programmet kommer med forslag basert på forrige års klasselister, hvordan klassene blir oppdatert. Hele klassekullet med studenter blir automatisk overflyttet, i tilfeller med avvik må superbruker manuelt editere klasselistene for å gjøre om disse. Dette gjelder både for å slette studenter fra eller legge nye studenter til klasselistene. Han kan sette maks antall studenter per test dersom flere kan besvare testen sammen. Han må legge inn sin e-post adresse for feilrapportering og påminnelse om tidspunkt for oppdatering av klasselister og godkjentlister. Superbruker sender mail til brukerne når en test er klar for testing. Superbruker setter URL for hver test. Hver spørsmålstype får sin forklarende innledningstekstfelt..... lagt inn av superbruker. Superbruker angir også karakterskalaen med oppgitte intervaller og angir hvilket karakterområde som omfatter godkjent statusen.

**2.2.4.5.2 Prosessering** Dersom superbrukeren har oppgitt sin e-post adresse, vil han få meldinger om feil som oppstår. Modulen bruker den sentrale databasen.

**2.2.4.5.3 Output** Superbrukeren kan få se en del statistikk for systemet. Dette er:

- Antall lærere.
- Antall studenter.
- Antall tester.
- Antall fag.
- Databasens forbruk av harddisk-plass.

Superbrukeren setter datoer for når han ønsker å få e-post fra systemet som en påminnelse om oppdatering av klasselister er nødvendig, dersom han har bedt om det. Han kan skrive ut gamle godkjentlister til tekstfiler.

**2.2.4.5.4 Feilrapportering** Superbrukeren får feilmeldinger dersom andre moduler er ute av funksjon.



**2.2.4.5.5 Sikkerhet** Superbrukeren må oppgi brukernavn og passord for å aksessere superbrukermodulen.

## 2.2.5 Spørsmålstyper

### 2.2.5.1 Flervalgs spørsmål

**2.2.5.1.1 Spørsmålsbeskrivelse** Til dette spørsmålet er det først en spørsmålstekst. I tilknytning til denne teksten kan det implementeres en fil inneholdende bilder, lyd, video, dokumenter eller programmer. Innen dette spørsmålet kan det velges mellom tre undertyper.

- Den første er at *en av to* er riktig. Hvis denne undertypen er valgt, kan det videre velges mellom ja/nei, sant/usant og riktig/galt, eller læreren kan definere sine to egne ord.
- Den andre undertypen er spørsmål der *én av m* er riktig. Alternativene kan ha tekst og det kan høre til bilder, lyd, video, dokumenter eller programmer.
- Den tredje undertypen er spørsmål der *n av m* er riktige. Alternativene kan ha tekst og det kan høre til bilder, lyd, video, dokumenter eller programmer. Ingen, flere eller alle kan være riktig. Dersom læreren ønsker det, kan studenten få se mellom hvert forsøk hvilke av spørsmålene som er riktig svart hittil. Studenten får oppgitt oppnådd score eller poeng i forhold til alternativene studenten har valgt riktig på. Svarrekkefølgen kan vektas og hintforespørsler kan gi fratrukk i oppnådd score. Det kan legges til inntil tre hint. Disse kan studenten be om, eller han vil få dem når han svarer feil.

Som tilbakemelding ved riktig eller galt svar kan lærerne velge mellom en standardmelding, eller selv legge inn tilbakemeldinger til hvert alternativ eller til hele spørsmålet. Lærerne legger inn data for spørsmålene og angir riktig svar. Dataene blir prosessert og dette blir output i studentenes svarmodul. Alternativene kommer i den rekkefølgen lærerne har oppgitt. I svarmodulen velger studentene ett eller flere alternativer. Disse blir lagret i systemet. Det gis fratrukk for gale svar.

#### 2.2.5.1.2 Input - lærere

Alternativer	Et eller flere. Bilder, lyd, video, dokument, program eller tekstfelt, 512 tegn.
Riktige svar	Ett eller flere.
Hint eller tips	Ett til hvert alternativ, eller tre til hele spørsmålet. Bilder lyd, video, dokument, program eller tekstfelt, 512 tegn.
Melding ved feil/riktig svar	Bilder, lyd, video, dokument, program eller tekstfelt, 256 tegn.
Få se riktige hittil	Boolean, ja/nei.

**2.2.5.1.3 Input - studenter** Svaralternativ - ett eller flere.

**2.2.5.1.4 Statistikk** Lærerne kan få ut hvor mange som valgt de forskjellige svaralternativene, og hvor mange som har svart helt riktig på et aktuelt spørsmål. De kan også få se gjennomsnittlig poengsum for dette spørsmålet. Lærerne kan få se hvor mange som har fått null poeng i spørsmålet. Lærerne kan hente statistikk over hvor mange hint studentene brukte i aktuelle spørsmål.

#### 2.2.5.2 Fyll inn tall eller ord

**2.2.5.2.1 Spørsmålsbeskrivelse** Læreren kan legge inn en forklarende tekst for hele spørsmålsstillingen. Den går ut på at studentene skal fylle ut et tall eller sette inn et ord i et felt. Det kan være flere felt med tekst foran. Til et tall kan det også oppgis en feilmargen i prosent eller en øvre og nedre toleransegrense. Til ord kan det være mulig å oppgi synonymer eller likeverdige ord og det skal være mulig for "enkle" skrivefeil; som for eksempel et enkelttegn kan mangle eller være ekstra, forskjell på små og store bokstaver, og to tegn kan være byttet om. Det kan oppgis inntil tre hint som studenten får, enten ved å svare feil eller ved aktivt å be om hint. Det kan også være enkelthint ved hvert tekstfelt. Hintene kan være bilder, lyd, video, dokumenter eller programmer. Læreren må bestemme hvordan hintene skal presenteres.....

#### 2.2.5.2.2 Input - lærere Per temafelt:

Tall	Flyttall, positivt eller negativt.
Feilmargin	Heltall, 0-100.
Øvre grense	Flyttall, positivt eller negativt.
Nedre grense	Flyttall, positivt eller negativt.
Løsning/svar	Tekstfelt, 128 tegn, ett eller flere løsninger/svar.
Godta skrivefeil	Boolean, ja/nei.
Skille mellom små/store bokstaver	Boolean, ja/nei.
Spørsmålstekst foran feltet	Tekstfelt, 1024 tegn.

For hele oppgaven:

Linjebryting	Boolean, Ja/nei.
Hint eller tips	Et til hvert alternativ, eller et til hele spørsmålet. Bilder, lyd, video, dokument, program eller tekstfelt, 256 tegn.
Melding ved riktig svar	Bilder, lyd, video, dokument, program eller tekstfelt, 512 tegn.
Få se riktige hittil	Boolean, ja/nei.

#### 2.2.5.2.3 Input studenter

Tall	Flyttall, ett eller flere.
Ord	Tekstfelt, 128 tegn, ett eller flere

**2.2.5.2.4 Statistikk** Lærerne kan få se gjennomsnittssvaret for hvert spørsmål og hvor mange som har svart helt riktig, samt gjennomsnittlig poengsumoppnåelse for dette spørsmålet. Læreren kan også få listet opp de ordene som var feil. Lærerne kan få se hvor mange som ikke har fått poeng i spørsmålet.

#### 2.2.5.3 Finne sekvens

**2.2.5.3.1 Spørsmålsbeskrivelse** Her skal studentene finne en rekkefølge blant flere ord, blant bilder eller ved å peke på ord, tekst eller areal i et bilde. Studenten må peke på feltene, bildene eller ordene i riktig rekkefølge. Lærerne må angi alternativer og riktig sekvens. Lærerne kan velge om studenten får vite hvilke valg som er på riktig plass eller hvilke som er i riktig rekkefølge.

Læreren kan legge inn tilbakemelding til riktige og gale svar. Disse kan være standard innebygde, et for hele spørsmålet eller et for hvert alternativ. Det kan være inntil tre hint per spørsmål. Studentene kan få hint, som er avhengig av hvor de peker eller bare en rekkefølge av hint. Lærerne legger inn bilder eller tekst og bestemmer rekkefølgen til sekvensene. Så gir de hint i forhold til område.....

#### 2.2.5.3.2 Input - lærere

Alternativer	To eller flere. Bilder, lyd, video, dokument, program eller tekstfelt, 512 tegn.
Riktig svar	Angi riktig rekkefølge.
Hint eller tips	Et til hvert alternativ, eller tre til hele spørsmålmulder, lyd, video, dokument, program eller tekstfelt, 512 tegn.
Melding ved riktig/feil svar	Bilder, lyd, video, dokument, program eller tekstfelt, 512 tegn.
Få se riktig plasserte hittil	Boolean, ja/nei.
Få se de i riktig rekkefølge hittil	Boolean, ja/nei.

#### 2.2.5.3.3 Input - studenter Peker ut eller velger rekkefølge.

**2.2.5.3.4 Statistikk** Lærerne kan få se hvor mange som har svart helt riktig på dette spørsmålet og gjennomsnittlig poengsum for spørsmålet. Lærerne kan få se hvor mange som har fått null poeng i spørsmålet. I tillegg kan lærer få opplysninger om hvor mange som benyttet hint-funksjonen og hvor mange hint som ble benyttet til de forskjellige spørsmålene.

#### 2.2.5.4 Kombinere

**2.2.5.4.1 Spørsmålsbeskrivelse** Her skal studenten koble sammen alternativer med svar, disse kan være bilder, lyd, video, dokument, program eller tekst. Dette gjør han ved å dra og slippe med musen. Lærerne legger inn bilder og tekst og setter sammen riktige kombinasjoner. Det kan være inntil tre hint på hele spørsmålet eller et hint per alternativ. Det kan være flere valg enn riktige par på begge sidene. Læreren kan legge inn tilbakemelding til riktige og gale svar. Disse kan være standard innebygde, en for hele testen eller en for hvert par. Lærerne må velge om studentene får se riktige svarpar ved flere forsøk. Lærerne kan også velge om en objekt kan brukes flere ganger. Studenten skal be om hint eller få det automatisk hvis svaret ikke er riktig. Ved en situasjon der alle tillatte hint er brukte, kan lærer velge at riktig svaralternativ skal vises.

##### 2.2.5.4.2 Input - lærere

Alternativer	To eller flere. Bilder, lyd, video, dokument, program eller tekstfelt, 512 tegn.
Riktig svar	Angi riktige par.
Hint eller tips	Et til hvert par, eller tre til hele spørsmålet. Bilder, lyd, video, dokument, program eller tekstfelt, 512 tegn.
Melding ved riktige/feil svar	Bilder, lyd, video, dokument, program eller tekstfelt, 512 tegn.
Få se riktige par hittil	Boolean, Ja/nei.
Bruke flere ganger	Boolean, Ja/nei.

##### 2.2.5.4.3 Input - studenter Velge par.

**2.2.5.4.4 Statistikk** Lærerne kan få se hvor mange som har svart helt riktig på dette spørsmålet og gjennomsnittlig poengsum for spørsmålet. Lærerne kan få se hvor mange som har fått null poeng i spørsmålet. Lærerne kan få se hvor mange som har fått til de ulike parene.

#### 2.2.5.5 Plassere ut noe på et bilde

**2.2.5.5.1 Spørsmålsbeskrivelse** Her må en lærer legge ut et bilde som danner en bakgrunn. På dette bildet skal studenten plassere ut alternativene som læren har lagt inn. Disse kan være bilder, lyder, video, dokumenter, programmer eller tekstfelter. Lærerne må angi arealer ved hjelp av sirkler, rektangler eller mange kant. Lærerne kan legge inn inntil tre hint per alternativ. Studenten skal be om hint eller få det automatisk hvis svaret ikke er riktig. Læreren kan legge inn tilbakemelding til riktige og gale svar. Disse kan være standard innebygd, et for hele testen eller et for hvert alternativ. Lærerne må velge om studentene får se riktige svar ved flere forsøk. Lærerne kan velge om alternativene skal plasseres ut flere ganger.

##### 2.2.5.5.2 Input - lærere

Alternativer	Et eller flere. Bilder, lyd, video, dokument, program eller tekstfelt, 512 tegn.
Riktig svar	Angi riktige areal for plassering.
Hint eller tips	Et til hvert alternativ, eller tre til hele spørsmålet. Bilder, lyd, video, dokument, program eller tekstfelt, 512 tegn.
Melding ved riktige/feil svar	Bilder, lyd, video, dokument, program eller tekstfelt, 512 tegn.
Få se riktig plasserte hittil	Boolean, Ja/nei.
Bruke flere ganger	Boolean, Ja/nei.

##### 2.2.5.5.3 Input - studenter Plassere ut alternativer.

**2.2.5.5.4 Statistikk** Lærerne kan få se hvor mange som har svart helt riktig på dette spørsmålet og gjennomsnittlig poengsum for spørsmålet. Lærerne kan få se hvor mange som har fått null poeng i spørsmålet. Lærerne kan få se hvor mange som har fått til de ulike plasseringene av arealene.

### 2.2.5.6 Peke på et sted/areal i et bilde med progressive hint

**2.2.5.6.1 Spørsmålsbeskrivelse** Her må en lærer legge ut et bilde som danner en bakgrunn på skjerm-bildet. Hvert spørsmål går ut på å peke på det som læreren ber om. Læreren må angi arealer ved hjelp av sirkler, rektangler eller mangekanter. Læreren bør legge inn inntil tre hint for hele spørsmålet. Disse kan inneholde bilder, lyder, video, dokumenter, programmer eller tekst. Studenten får automatisk hint hvis svaret ikke er riktig. Dersom studenten ikke klarer å løse oppgaveteksten ved hjelp av oppgitte hint, kan det gis direkte indikasjon på hva svaret skal være ved hjelp av pil eller skraver/fargelagt område. Læreren kan legge inn tilbakemelding til riktige og gale svar. Disse kan være standard innebygd for hvert svarresultat (riktig eller galt svar) eller et for hele spørsmålet.

#### 2.2.5.6.2 Input - lærere

Bilde	Bilde.
Riktig svar	Angi riktige areal for plassering.
Hint eller tips	Inntil tre stykker til hvert spørsmål. Bilder, lyd, video, dokument, program eller tekstfelt, 512 tegn.
Melding ved riktige/feil svar	Bilder, lyd, video, dokument, program eller tekstfelt, 512 tegn.

#### 2.2.5.6.3 Input - studenter

Peke på bildet.

**2.2.5.6.4 Statistikk** Læreren kan få se hvor mange som har svart helt riktig på dette spørsmålet og gjennomsnittlig poengsum for spørsmålet. Læreren kan få se hvor mange som har fått null poeng i spørsmålet. Læreren kan få se hvor mange som har fått til de ulike plasseringene på arealene.

### 2.2.5.7 Nye spørsmål

**2.2.5.7.1 Spørsmålsbeskrivelse** Systemet holdes åpent for å legge til nye spørsmålstyper i senere versjoner.

## 2.3 Begrensninger

### 2.3.1 Software design begrensninger

#### 2.3.1.1 Software standarder og språk

Systemet skal utvikles i PHP4, Java, JavaScript og HTML. Det skal benyttes MySQL database.

#### 2.3.1.2 Software grensesnitt

Nettleser, HTML, logisk brukergrensesnitt. Systemet skal kommunisere etter internettstandard og bruke TCP/IP-forbindelse.

#### 2.3.1.3 Software pakker/verktøy

Det er ønskelig at systemet skal kunne kjøres på de internettjenere og nettlesere som er markedsledende dvs. Internettjenere som støtter PHP og nettlesere som støtter Java, CSS og JavaScript.

#### 2.3.1.4 Software kommunikasjonsstandarder og grensesnitt

Kommunikasjon mot brukerne skal foregå via Internett. Brukerne skal kunne bruke vanlig internettforbindelse og nettlesere som støtter Java og JavaScript.

#### **2.3.1.5 Database**

Det skal kun være en sentral database, ikke en på hver maskin. Databasen skal være en SQL-database, og vi har valgt å bruke MySQL-server fordi denne er gratis å bruke for utdannings- og forskningsinstitusjoner.

#### **2.3.1.6 Operativsystem**

Den sentrale databasen skal kjøres på et system som støtter PHP og MySQL. Brukernes operativsystem er uten begrensninger så lenge de tilfredsstillt kravet om oppdatert nettleser. (IE 5.0 NN 4.7)

### **2.3.2 Hardware design begrensninger**

#### **2.3.2.1 Hardware krav og omgivelser**

Kravene til hardware og omgivelser begrenses av internettjeneren.

#### **2.3.2.2 Hardware grensesnitt**

Dette begrenses av brukernes internettforbindelse og operativsystem.

## **2.4 Aspekter omkring livssyklus**

### **2.4.1 Dokumentasjon**

All kode skal dokumenteres. Det utarbeides full oversikt over alle filer og kataloger, hva de inneholder og deres funksjoner.

### **2.4.2 Modul og integrasjonstesting**

Vi tester hver modul etter at den er ferdig og retter feil i hver enkelt før vi setter systemet sammen. Etter at prosjektet er ferdig kan det fortsatt komme nye versjoner.

### **2.4.3 Konfigurasjons- og versjonstyring**

Den endelige versjonen kalles 1.0. Dersom vi lager nye versjoner kalles med små endringer kalles disse 1.1, 1.2 osv. Dersom vi lager helt nye systemer som baserer på det gamle, kaller vi det 2.0, 3.0 osv. Ikke-lanserte versjoner er nummerert lavere enn 1.0.

### **2.4.4 Krav til support, service og vedlikehold**

Superbrukeren har ansvar for support. Superbruker og de som driver andre IT-systemer i organisasjonen tar seg av drift og vedlikehold. Systemet skal ikke behøve noe kontinuerlig vedlikehold, vedlikehold skal kun bestå av feilretting, dersom feil oppstår.

### **2.4.5 Krav til utvidelser**

Systemet skal være enkelt å utvide med nye moduler. Det skal enkelt kunne legges til nye spørsmålstyper.

## **2.5 Aspekter omkring installasjon**

### **2.5.1 Overgang og omlegging**

Systemet er helt nytt og skal installeres på en internettjener.

### **2.5.2 Opplæring**

Vi skal ikke gi opplæring. Superbruker gjør seg kjent med systemet gjennom prosjektrapporten og er ansvarlig for å hjelpe andre i gang. Systemet skal være selvforklarende og inneholde en hjelpefunksjon for hvert skjermbilde.

## **2.6 Utgivelser underveis**

Vi skal levere tre statusrapporter. Disse skal leveres 15.02.01, 30.03.01 og 10.05.01. Disse statusrapportene skal inneholde gruppas vurderinger vedrørende vårt ståsted i forhold til tidsskjemaet og vår fremdriftsplan, hva vi har gjort ferdig og hva vi holder på med av arbeidsoppgaver, litt om samarbeid og motivasjon innen prosjektgruppa og hvilke problemer og muligheter vi ser vedrørende hovedprosjektarbeidet fremover i tid.

## **2.7 Akseptansekrav**

Systemet skal tilfredsstillе de kravene som er stilt i denne kravspesifikasjonen. I tillegg kan oppdragsgiver komme med tilføyninger og rettinger frem til 01.03.01. Etter at den endelige versjonen er utgitt 07.05.01 regner vi systemet for akseptert av oppdragsgiver. Feil som oppdages etter 14.05.01 og som kan rettes innen 08.06.01, vil gjøres av prosjektgruppa.

# Kapittel 3

## Analyse

### 3.1 Hva systemet skal inneholde

Krav til funksjonalitet til systemet ble definert i prosjektrapport gjennom kravspesifikasjonen. Den ble ytterligere finspesifisert etter hvert som utviklingen av systemet skred frem og gjennom dialog med oppdragsgiver og veileder. Systemet skal inneholde de forskjellige modulene skissert i kravspesifikasjonen.

Systemet skal også ha et layoutmessig utforming som tilfredsstillende oppdragsgivers ønsker for et slikt system. Det må ha en egen superbrukermodul og denne må være passordbeskyttet. Superbruker setter en del innstillinger for systemet og oppretter nye klasselister basert på forrige års lister.

Han kan også opprette helt nye klasser. Han oppretter lærere som brukere, slik at disse kan logge seg på og begynne å bruke systemet. Superbruker kan også slette fag og tester og skrive godkjentlister til flate filer for langtidslagring. Lærerne må kunne logge seg på og opprette fag/kurs, tester og spørsmål under disse. Spørsmålene kan være av forskjellige typer, slik at læreren har mulighet til å teste studentene på flere forskjellige måter. Læreren kan sette forskjellige innstillinger på testene og spørsmålene. Eksempler på dette kan være åpnings- og lukkedato for en test, vekt på hvert spørsmål i en test og godkjentgrense for testen. En del enkel statistikk knyttet opp mot systemet og antall brukere skal også med i løsningen.

Studentene må logge seg inn med brukernavn og passord for å få adgang til å besvare en test uansett om denne er en obligatorisk test eller en statustest. Systemet må inneholde en spørsmålspresentatør hvor studentene får presentert spørsmålene og kan svare på disse.

### 3.2 Eksisterende løsninger

I og med at prosjektgruppa fra i fjor hadde gjort en såpass god research på eksisterende løsninger, brukte vi begrenset tid på dette. Vi lette litt men fant ingen eksisterende løsninger som tilfredsstilte oppdragsgivers krav. De fleste ferdige løsningene inneholder for få spørsmålstyper og for dårlig eller helt manglende mulighet til å føre godkjentlister. De fleste eksisterende løsninger er kun åpne for flervalgsspørsmål (multiple choice).

### 3.3 Vurderinger

Vi hadde en ganske klar formening om hva vi skulle lage før hele utviklingsfasen startet, fordi vi hadde kravspesifikasjonen fra forrige års prosjektgruppe. Underveis i analyse og design-fasen oppdaget vi hvor komplekst og stort vårt hovedprosjektoppgaveområde var, og at den eksisterende kravspesifikasjonen hadde noen mangler i forhold til kravene til oppdragsgiver.

Det ble besluttet å benytte et annet programmeringsspråk.

Underveis skjønte vi at vi ikke hadde mulighet for å komme helt i mål med utvikling av løsningen, og oppdragsgiver prioriterte de forskjellige delmodulene. Alle administrasjonsløsninger både når det gjelder opprettelse av fag, tester og spørsmål skulle gjennomføres. Likeledes superbrukermodulen som skal definere alle innstillinger og brukere til systemet. Systemet er utvidbart og kan utvides med flere funksjonaliteter, for eksempel vil en utvidelse av statistikkfunksjonene og eventuelt flere spørsmålstyper kunne legges til senere. Dette gjør det mulig

for en eventuell kommende hovedprosjektgruppe ved HiG å arbeide videre for å gjøre løsningen komplett. Dette er nok også nødvendig for å få løsningen opp på et nivå som gjør at den skal kunne taes i bruk for fullt ved HiG.

Et overordnet mål med vår løsning av systemet var at det grafiske brukergrensesnittet skulle være entydig og forklarende gjennom hele systemet. Videre skulle det være "Windows-like" og mest mulig gjenkjennbart gjennom de forskjellige komponentene og de forskjellige sidene.

Kvitteringsvindu for vellykkede forandringer når det gjelder fag, tester og spørsmål fra lærers side, gikk vi bort fra. I stedet for kvitteringsvinduer, valgte vi feilmeldingsbeskjeder ved feilsituasjoner. Ved en mislykket editering eller forandring, får brukeren en standard feilmelding tilbake. Vi benytter også kvitteringsvinduer før sletting slik at læreren ikke skal kunne slette noe ved et uhell.



# Kapittel 4

## Design

### 4.1 Hvordan løsningen ble laget

#### 4.1.1 Layout

Sidene utforming er bygget opp med en rekke filer som blir inkludert. Vi inkluderer en fil for forbindelse til databasen og en annen fil for stylesheet. I studentmodulen og i superbrukermodulen er sidene bygget opp som en tabell. I tillegg til forbindelse til databasen inkluderes det her en fil som inneholder starten på en tabell, samt en funksjon som tar imot parametre på hvilke filer som skal brukes som meny og tittel på siden. Selve cellen i midten fungerer som sideinnhold, og den avsluttes med en fil som inkluderes nederst. Her avsluttes alle tabeller.

På denne måten bygger vi opp en side uten å benytte frames, men ivaretar den samme funksjonaliteten. Når det blir såpass mange sider er det nødvendig å samle gjentakende operasjoner på ett sted. Dette gjør seg spesielt gjeldende dersom man må endre på noe, eks brukernavn og passord mot databasen.

#### 4.1.2 Databasen

Vi startet med å designe databasen da den er selve hjørnesteinen i vår løsning. Vi jobbet videre med den eksisterende databasen, men modifiserte en del på denne fordi vi fant noen mangler på den. Databasestrukturen ble skikkelig gjennomgått før vi begynte å kode. Den største forskjellen fra den eksisterende databasestrukturen til vår database, var at vi valgte å legge filenes binære data inn i databasen. Dette gjorde vi fordi vi hadde gode erfaringer med det fra andre prosjekter vi hadde jobbet med. I tillegg kunne vi oppnå å spare tid og unødvendig ekstra koding i utviklingsprosessen.

Vi brukte verktøyet "Modelator 4.0" for å designe databasen fordi vi med det verktøyet kunne modellere den grafisk og få generert et sql-script automatisk ut fra dette. Dette gjorde vi fordi vi da både fikk en detaljert modell, en detaljert rapport og sql-scriptet som genererer databasen i én arbeidsoperasjon.

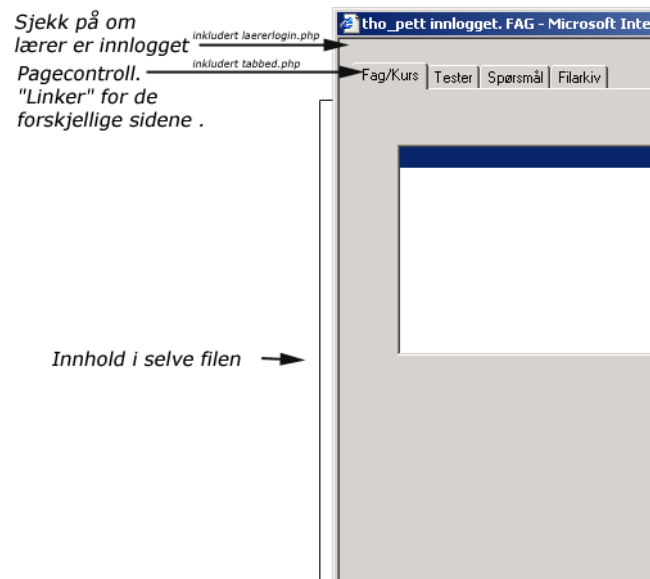
Vi tok utgangspunkt i kravspesifikasjonen samt databasescriptet fra forrige års TVIL-prosjektgruppe da vi startet på design av databasen. Først ble sentrale tabeller som "laerer", "fag", "test", "sporsmaal" og "innstillinger" opprettet. Vi jobbet videre med mindre sentrale tabeller helt til databasen var komplett.

Vi var nødt til å sette det åpent for å legge til nye spørsmålstyper til løsningen, dette løste vi i databasen ved å legge alle "felles" data for et spørsmål i en egen tabell, samt spørsmålstypen. Spørsmålsdata som er avhengig av spørsmålstypen la vi i egne tabeller som kan legges til senere. Se Figur 4.3 for å studere endelig design på databasen.

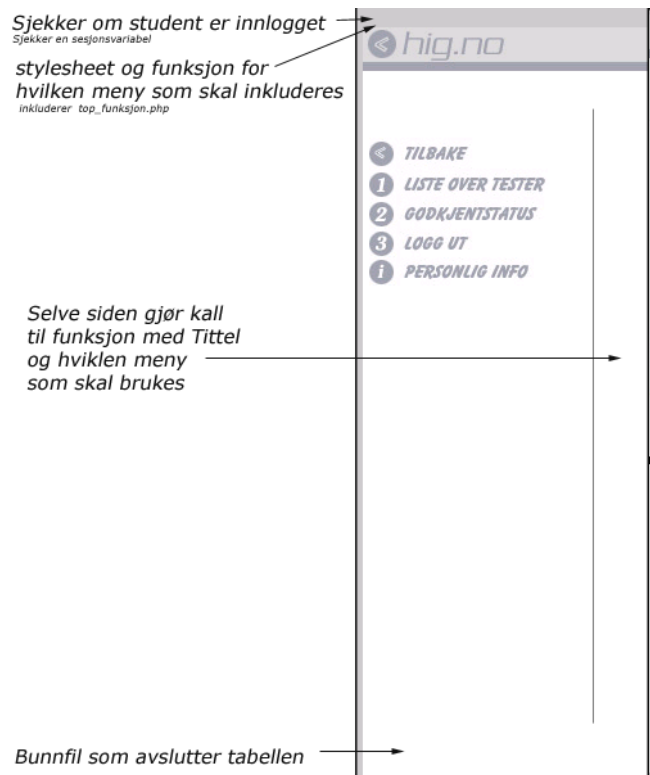
#### 4.1.3 Lærermodulen

##### 4.1.3.1 Fag

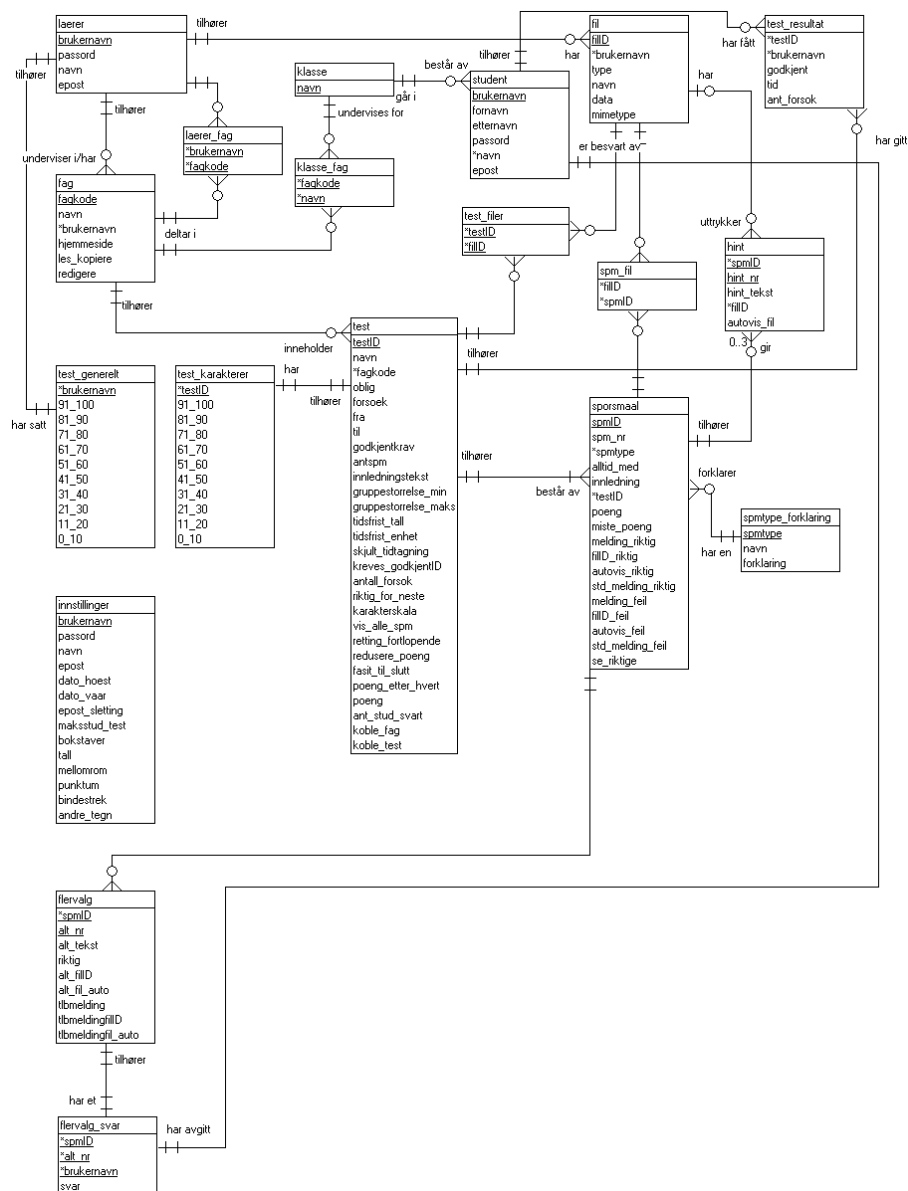
Her får læreren en oversikt over alle fag han har opprettet. Nederst på siden vises det litt informasjon om valgt fag. Læreren kan se hvor mange obligatorisk oppgaver og hvor mange statustester som finnes i faget. Det er også link til fagets hjemmeside og en link til en side som viser oversikt over alle tester og oblig'er under faget. Dette kan læreren dra nytte av dersom han ønsker å gi studentene en direkte link fra egen hjemmeside til fagets tester. Læreren får også se hvilken epostadresse han er registrert med. Læreren kan opprette nye fag, endre på



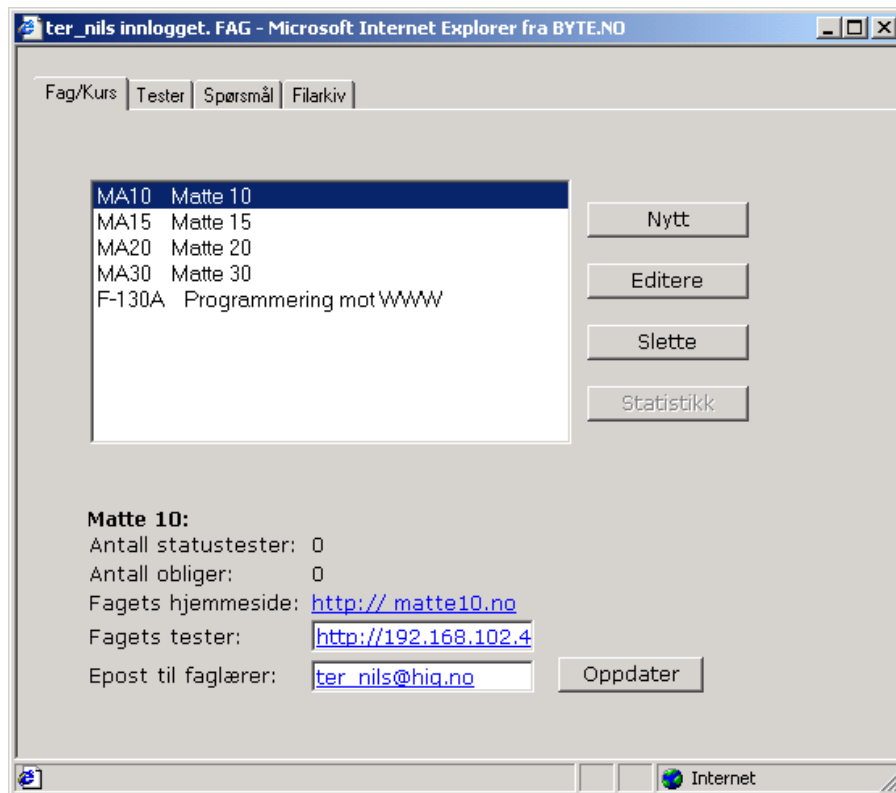
Figur 4.1: Sidens oppbygning, lærer.



Figur 4.2: Sidens oppbygning for student og superbruker.



Figur 4.3: Fullstendig ER-diagram



Figur 4.4: Fag.

eksisterende fag og slette ett fag. Dersom han ønsker å slette eller endre på ett fag, må lærer først merke et fag i listen, for deretter å trykke endre eller slette. Se Figur 4.4.

**4.1.3.1.1 Nytt fag** Dersom læreren velger å opprette ett nytt fag vil det komme frem et nytt vindu på topp av det andre. Læreren må skrive inn fagkode og navn på faget. Han kan også skrive inn adressen til fagets hjemmeside. Det må bestemmes hvilke klasser som skal ha tilgang til faget, og det gjøres ved å markere de respektive klassene i listen til høyre, for så og trykke på pilen som bringer klassene over til venstre liste. Samme prinsipp gjelder når læreren skal bestemme hvem som skal ha mulighet til å endre på faget. Læreren kan velge om han skal være den eneste som kan editere, om alle kan editere, eller velge ut noen fra listen til høyre. Læreren bestemmer også om fagets tester kan leses og kopieres av andre lærere. Når bruker trykker OK registres faget. Se Figur 4.5.

**4.1.3.1.2 Endre fag** Læreren møter samme skjermbilde som på nytt fag, se Figur 4.5. Han kan her endre faget sitt som han måtte ønske men han kan imidlertid ikke endre på fagkoden.

**4.1.3.1.3 Slette fag** Læreren får en dialogboks om han vil slette eller ikke. I denne dialogboksen er knappen "Nei" standard valgt, slik at ingen fag blir slettet ved uhell. Velger læreren å slette faget oppdateres listen og faget og alle tester og spørsmål er slettet.

**4.1.3.1.4 Statistikk** Ikke implementert.

#### 4.1.3.2 Test

Læreren har en liste over sine fag i den øverste nedtrekksmenyen. Her kan han velge å se alle fag han har mulighet til å endre på ved å krysse av for dette til høyre. Da vil listen også omfatte alle fag der læreren er gitt tillatelse til å gjøre endringer. Når læreren har valgt et fag vil testene komme til syne i boksene under. Statustester og

Registrere nytt fag - Microsoft Internet Explorer

**Fagkode:** F-130A

**Fagnavn:** Programmering mot WWW

**Hjemmeside:** http:// higweb.hig.no/at/data/progwww/

**Klasser:** 98hindb

**Alle klasser:** 01hinda, 01hindb, 01hindu, 02hindb, 97hindb, 99hinda, 99hindb

**Kan editeres av:**

Bare meg selv

Alle

Gruppe

**Brukere:** fro\_haug

**Alle brukere:** sti\_hage, ter\_nils, testbruker, tho\_pett

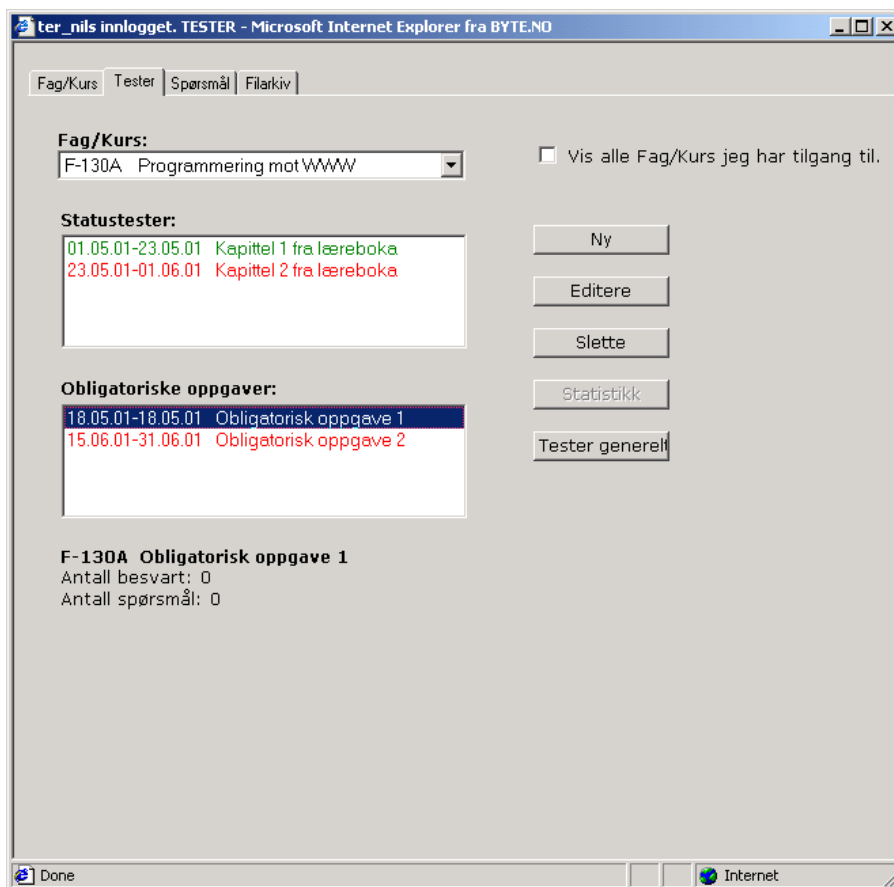
**Kan leses og kopieres av:**

Bare meg selv

Alle

Ok Avbryt

Figur 4.5: Nytt fag.



Figur 4.6: Test

**Ny test i faget:** F-130A Programmering mot WWW

**Testens navn:** Kapittel 1 fra læreboka

**Testtype:**

Obligatorisk oppgave med 1 forsøk

Statustest

**Innledningstekst:**

Statustest som tar for seg kapittel 1 i læreboka.  
Lykke til.

**Godkjentgrense:** 80 prosent

**Tilgjengelig fra:** 01.05.2001 (dd.mm.åååå)

**Tilgjengelig til:** 23.05.2001 (dd.mm.åååå)

**Avanserte innstillinger:**

OK Avbryt

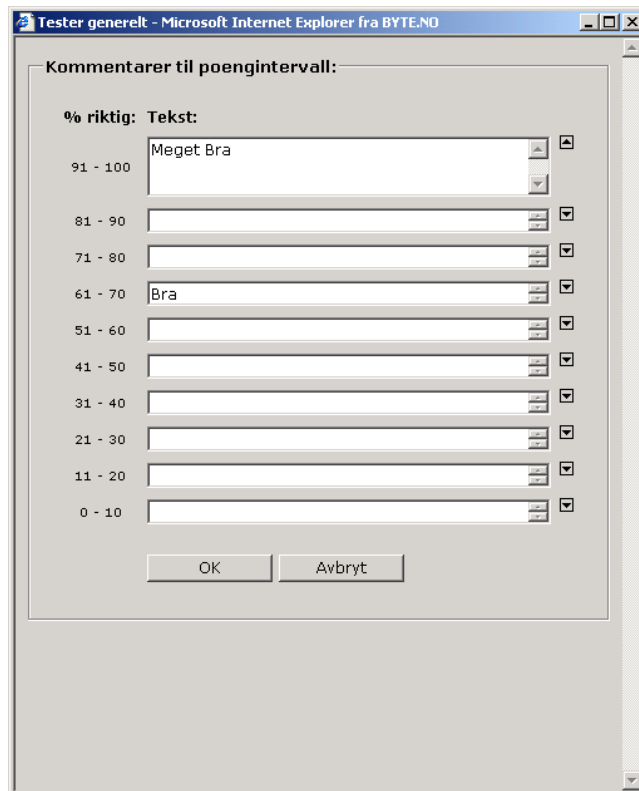
Figur 4.7: Ny test.

obligatoriske oppgaver listes i hver sin boks, og sorteres på dato de åpnes på og dato de stenges. Åpne tester har grønn tekst og stengte har rød. Dersom læreren markerer en test vil informasjon om testen komme til syne nederst i skjermbildet. Her vises antall studenter som har besvart, og hvor mange spørsmål testen inneholder. Se Figur 4.6.

**4.1.3.2.1 Ny test** Dersom læreren velger å opprette en ny test vil det komme frem et nytt vindu på topp av det andre. Her må læreren registrere testens navn, om det er en statustest eller obligatorisk test, samt antall forsøk hvis obligatorisk test. Læreren skriver inn en innledningstekst til studentene og bestemmer hvor mange prosent som må være besvart riktig for at testen skal godkjennes. Læreren må også bestemme tidsrommet for når testen skal være tilgjengelig for studentene.

Det er en rekke flere innstillinger under "avanserte innstillinger" som kommer til syne dersom man klikker på pilen som peker nedover ved siden av overskriften. Dersom testen er en statustest kan læreren bestemme om studenten må svare riktig på hver spørsmål for å komme til neste. Han kan også sette antall svarforsøk per spørsmål. Læreren kan gi studenten mulighet til å få alle spørsmålene i testen dersom tilfeldig trekning av spørsmål er valgt under opprettelse av spørsmål. Som statustest kan læreren også velge å gi studenten fasit etter hvert spørsmål og om studenten skal få fasiten til slutt. Dersom læreren lager en obligatorisk oppgave kan læreren angi om det er tillatt at flere leverer sammen, og størrelsen på gruppa.

Felles for både obligatorisk oppgave og statustest er at læreren velger om det skal gis poengsum etter hvert spørsmål, om man skal redusere vekt med 1 for hvert hint eller galt spørsmål og om man skal ta tiden på studenten. Tidtaking kan foregå på to forskjellige måter, enten kun for statistikken skyld, eller at det faktisk er satt en viss tidsramme studenten kan bruke på testen. Læreren kan nekte en student å ta testen dersom en



Figur 4.8: Tester generelt.

tidligere test ikke er godkjent. Han kan også velge og gi studenten en karakter på avlagt test, og bestemmer om den skal angis i hele eller tidels karakterer. Brukeren trykker OK for å lagre testen. Se Figur 4.7

**4.1.3.2.2 Endre test** For å endre eller slette en test må læren først markere testen. Har læreren valgt å endre en test vises samme skjermbilde som ved registrering av ny test. Læreren kan endre alle parametre.

**4.1.3.2.3 Slette test** Har læreren valgt å slette en test kommer det opp en dialogboks som spør brukeren om bekreftelse før testen slettes. I denne dialogboksen er knappen "Nei" standard merket slik at man unngår ufrivillig sletting av tester. Ved bekreftelse slettes testen og listen over tester oppdateres.

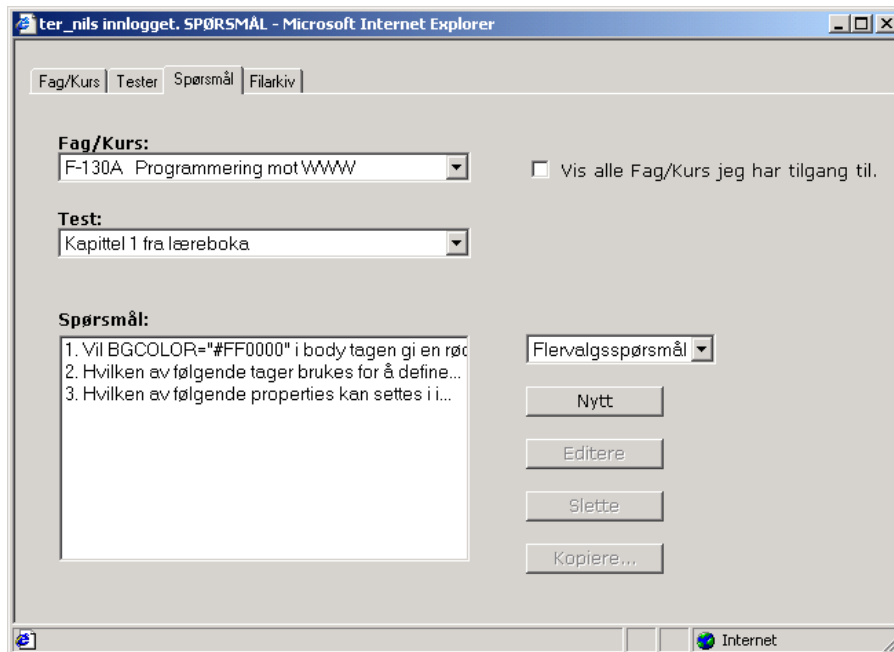
**4.1.3.2.4 Statistikk** En "lagre til disk" standard dialogboks kommer frem. lærer kan velge å lagre godkjentlister til fil eller åpne tekstfiler direkte

**4.1.3.2.5 Tester generelt** Trykkes "Tester generelt" åpnes det et nytt vindu over det andre. Her kan læreren sette standard kommentarer til poengintervall. Dette er altså tilbakemeldingen studentene får avhengig av oppnådd resultat. Kommentarene er delt inn i prosent med intervaller på ti. Disse kommentarene blir gjeldene for samtlige tester læreren har. Læreren kan velge å la enkelte intervaller være tomme. Alle underliggende poengintervaller vil få samme verdi som den sist gitte. Læreren kan for eksempel legge inn "Meget bra" som kommentar på 91%-100% riktig, og la alle intervaller ned til 61%-70% være tomme. Alle intervaller mellom disse vil gi en tilbakemelding til studenten som sier "Meget bra". Pilene til høyre for tekstboksene øker antall linjer i tekstboksen og letter skrivningen dersom det skal skrives mye. Se Figur 4.8.

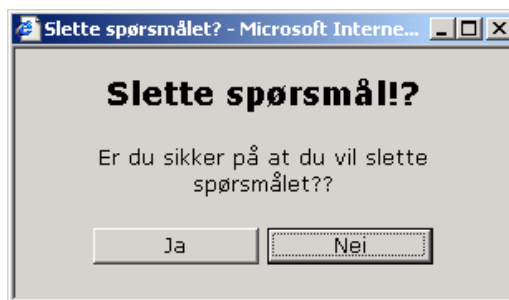
#### 4.1.3.3 Spørsmål

Skjermbildet læreren møter når han går inn på "Spørsmål" ser ut som på Figur 4.9. Dette er den delen av lærermodulen som gir læreren mulighet til å vedlikeholde og opprette nye spørsmål i tester han har tilgang til.





Figur 4.9: Spørsmålsdelen av lærermodul.



Figur 4.10: Sletting av spørsmål.

Dette er tester i egne Fag/Kurs eller tester i Fag/Kurs som en annen lærer eventuelt har gitt han tilgang til. For å arbeide med tester han som han ikke selv er eier på må han krysse av for "Vis alle Fag/Kurs jeg har tilgang til".

Først må læreren velge Fag/Kurs, da får han muligheten til å velge tester som tilhører valgt Fag/Kurs. Når test er valgt kan han begynne å legge til nye spørsmål, editere spørsmål og slette eksisterende spørsmål i testen. Spørsmål som er opprettet under valgt test listes opp. Knappene "Editere" og "Slette" er inaktive så lenge et spørsmål ikke er valgt i listen med spørsmål. Velger læreren et spørsmål i listen og trykker "Slette" får han opp en dialogboks hvor slettingen må bekreftes før den utføres, se Figur 4.10. I denne dialogboksen er knappen "Nei" standard valgt, slik at spørsmålet ikke slettes hvis læreren trykker "enter" ved et uhell.

Ved å velge spørsmålstype og trykke på "Nytt" får læreren opp et nytt vindu med mulighet for å opprette ett nytt spørsmål. Samme vindu åpnes hvis han trykker "Editere", men da er alle felter ferdig utfylt. Nye vindu som åpnes sjekker brukerens skjermopløsning slik at de ikke kommer utenfor det brukeren kan se.

På Figur 4.11 kan vi se et generelt grensesnitt for å opprette/editere et spørsmål. Feltet i midten er avhengig av valgt spørsmålstype. Alle spørsmål har en spørsmålstekst hvor spørsmålet formuleres. Denne teksten kan ha en tilhørende fil. Dette kan være bilde, audio, video, dokument eller et program. Læreren kan her enten laste opp en fil fra sin lokale harddisk eller bruke en allerede opplastet fil, se Figur 4.12. Velger han å laste opp fra egen harddisk må han trykke "Browse" og finne filen i den standard dialogboksen for å åpne fil i Windows. Filen blir da lastet opp og lagt til i filarkivet. Velger han å bruke en fil fra filarkivet velges denne fra en liste over alle hans tidligere opplastede filer. Automatisk visning av filen velges dersom læreren vil at filen skal vises

**Nytt flervalgsspørsmål - Microsoft Internet Explorer**

**Spørsmålstekst:**

Spørsmålstypeavhengig felt.

**Hint:**

1.

2.

3.

**Tilbakemelding:**

**Riktig:**  
Standard: Melding:

**Galt:**  
Standard: Melding:

**Avanserte innstillinger:**

**Være med uansett:**

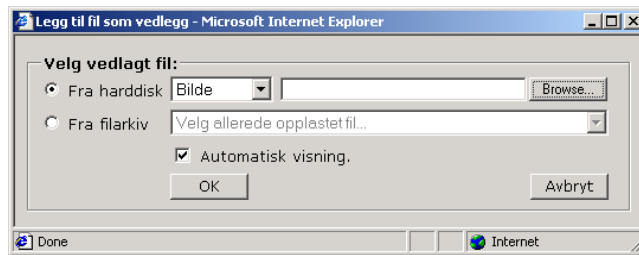
Ja

Nei, er med tilfeldig

**Poengberegning:**

Vekt:    
 Reduser med 1 ved hint/feil svar

Figur 4.11: Nytt spørsmål generelt.



Figur 4.12: Dialog for å velge fil.

automatisk for studentene når de får presentert spørsmålet under en test.

Videre legges spørsmålstypeavhengige data inn, dette beskrives senere i dette kapitlet.

For hvert spørsmål kan læreren legge inn opp til tre hint. Disse kan også ha tilhørende filer, som tilegnes på samme måte som for spørsmålsteksten. Disse hintene får studenten hvis han ønsker det under testen.

Læreren kan bestemme tilbakemeldingen studentene skal få ved riktig og galt svar på spørsmålet. Dette kan være enten en standard tilbakemelding, en egendefinert tilbakemelding eller en kombinasjon av begge deler. Egendefinert tilbakemelding kan også ha en tilhørende fil, som velges på samme måte som for spørsmålsteksten og hintene. Læreren kan også fjerne all tilbakemelding ved riktig/galt svar.

Avanserte innstillinger er standard ikke synlig. Brukeren kan åpne og lukke disse ved å trykke på pilen ved siden av overskriften. Dette løste vi også med DHTML. Under avanserte innstillinger kan læreren bestemme om spørsmålet alltid skal være med i testen eller om det er med tilfeldig. Spørsmålsrepresentatøren tar hensyn til dette og spørsmål trekkes ut ifra hvor mange spørsmål som skal være med i testen. Vekting på hvert spørsmål kan spesifiseres, og standardverdi her er vekt per spørsmål som er lagt inn for testen som spørsmålet tilhører. Reduksjon med ett poeng ved hint eller feil svar kan også spesifiseres for hvert spørsmål, standardverdi her er også samme verdi som for testen spørsmålet tilhører.

#### 4.1.3.4 Flervalgsspørsmål

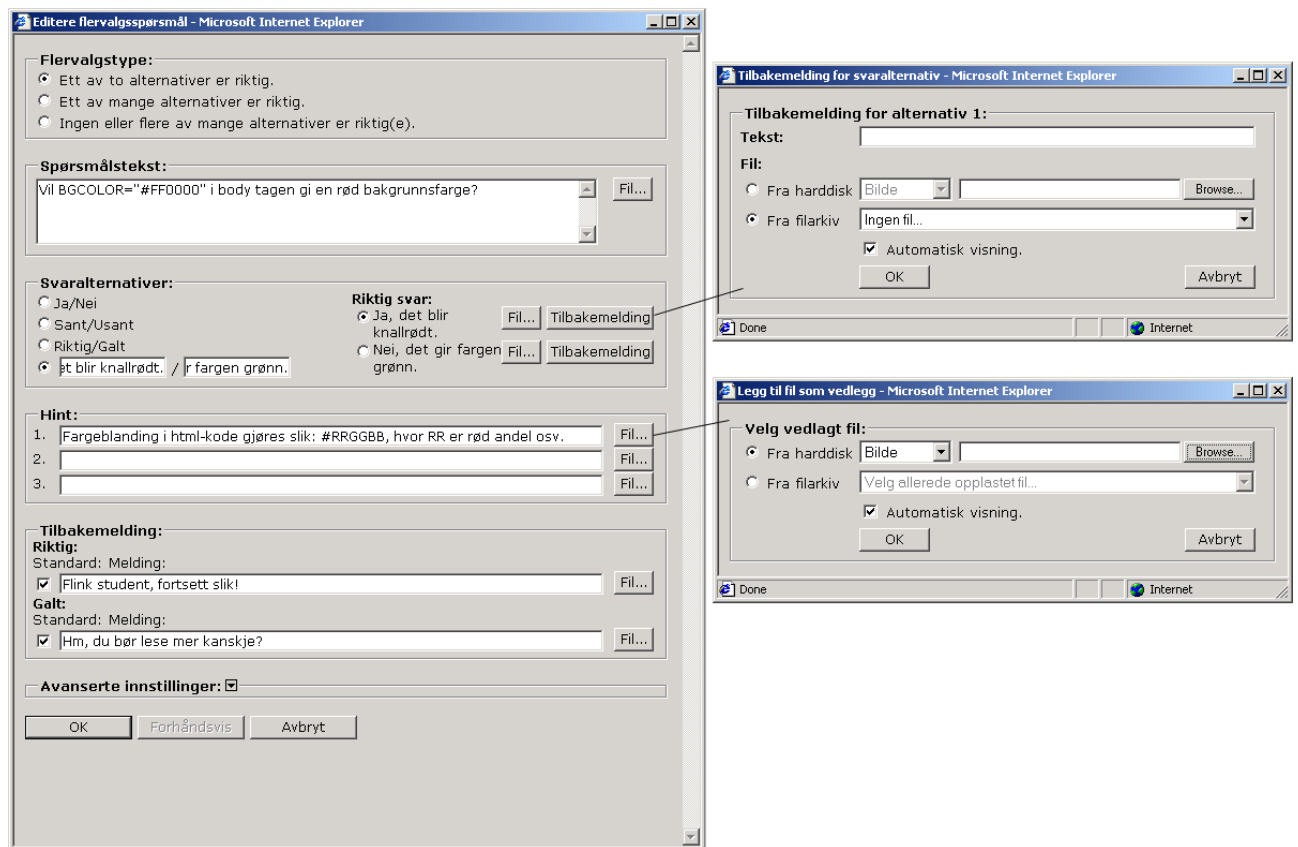
I flervalgsspørsmål kan man velge mellom tre undertyper, det første er at ett av to alternativer er riktig. Neste mulighet er at ett av mange alternativer er riktig. Siste undertype er at ingen eller flere av mange alternativer er riktig. Dette løste vi ved å benytte DHTML og "layers" som dynamisk endrer den spørsmålstypeavhengige delen av spørsmålssiden uten å laste den på nytt. På Figur 4.13 kan man se hvordan registrering av første undertype, hvor ett av mange alternativer er riktig, foregår.

Det som er spesielt for denne undertypen er at læreren bare kan velge å bruke "Ja/Nei", "Sant/Usant", "Riktig/Galt" eller to egendefinerte ord eller setninger. Videre må han velge hvilket av de to alternativene som er riktig. Dette gjør han ved å velge det riktige alternativet i en liste med to dynamisk oppdaterte tekster ved å trykke på tilhørende "radiobutton". For hvert alternativ kan en fil velges, samt at det kan velges en tilbakemelding til brukeren for dette alternativet. Disse kan for eksempel brukes til å forklare hvorfor et alternativ er riktig eller ikke.

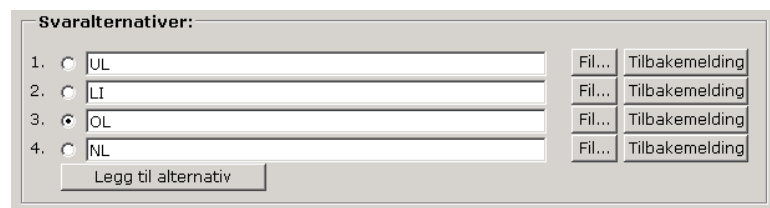
For undertypen hvor ett av mange alternativer er riktig, må læreren taste inn sine alternativer. På figur 4.14 kan vi se bare den spørsmålstypeavhengige delen av registreringen. Teksten for hvert alternativ må skrives inn og riktig svaralternativ må velges. Her kan kun ett alternativ markeres som riktig. Standard er fire svaralternativer, men flere kan legges til etter behov. Dette skjer også dynamisk uten å laste siden på nytt. Bare de alternativene som det registreres data på tas med i testen. Tilhørende filer og tilbakemeldinger kan registreres hvis læreren ønsker det. Filer velges også her, på samme måte som forklart tidligere .

Tilbakemeldingen som kan registreres for hvert alternativ kan inneholde en tekst og en fil. Dette registreres ved hjelp av et eget vindu. Dette fungerer på samme måte som for valg av fil beskrevet tidligere, men har i tillegg et tekstfelt for tilbakemeldingsteksten.

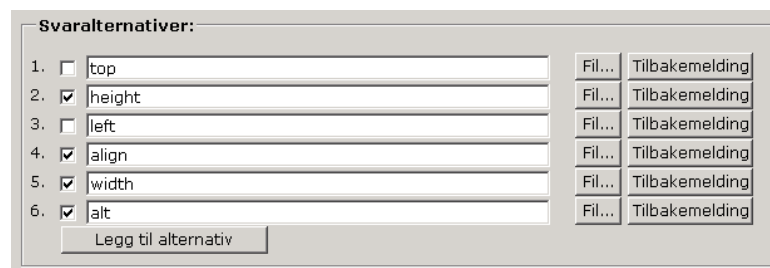
Undertypen hvor ingen eller flere av mange alternativer er riktig fungerer veldig likt som for tilfellet når ett av mange alternativer er riktig. Forskjellen er at læreren kan markere flere alternativer som riktige. Se utdrag av skjermbilde for denne undertypen på Figur 4.15. Her kan vi se at det er lagt til to nye alternativer i forhold til hva som er standard ved å trykke to ganger på knappen "Legg til alternativ".



Figur 4.13: Flervalgsspørsmål, ett av to alternativer er riktig.



Figur 4.14: Flervalgsspørsmål, ett av mange alternativer er riktig.



Figur 4.15: Flervalgsspørsmål, ingen eller flere av mange alternativer er riktig.



Figur 4.16: Studenenes innloggingsside.

#### 4.1.4 Studentmodulen

Ved første gangs bruk av systemet må studenten registrere seg. Dette gjør han ved å trykke på linken "Registrer deg!". Vi har bevisst valgt et enkelt brukergrensesnitt i studentens modul, fordi denne skal brukes av studenter som er på forskjellige nivåer rent datateknisk. Systemet skal kunne brukes på alle avdelinger ved HiG.

Her registrerer studenten all nødvendig informasjon om seg selv. Fornavn, etternavn, klasse, brukernavn, passord og epostadresse. Systemet foreslår en epostadresse når studenten taster inn ønsket brukernavn. Den foreslåtte adressen blir [brukernavn]@hig.no. Nå er studenten klar til å logge inn for første gang.

Når studenten har logget inn står han overfor tre enkle hovedvalg. Han kan se liste over tester, sjekke godkjentstatus på allerede besvarte tester og han kan logge ut. Studenten kan i tillegg endre personlig informasjon hvis han måtte ha behov for det. Vi har her også prøvd å fått til et enkelt brukergrensesnitt for ikke å "skremme" studenter som ikke er så vant til internettløsninger.

Velger studenten å få se en liste over tester, får han opp en oversikt over alle tester som er tilgjengelig for ham. Øverst i listen kommer alle obligatoriske oppgaver som er åpne, så kommer alle statustester som er åpne. Videre kommer lukkede obligatoriske oppgaver og statustester. De forskjellige tabellene med obligatoriske oppgaver og statustester er alle sortert på åpnings dato først og deretter på lukke dato. Dette for å få listet de mest aktuelle testene øverst. For å ta en test trykker studenten på testens navn, som er en link for å starte aktuell test. Studenten har fortsatt muligheten til å se på godkjentstatus på tester, logge ut og endre personlig informasjon. Han kan også gå tilbake til forrige skjermbilde, hovedvalg etter innlogging.

Før selve testen starter, får studenten opp ett skjermbilde hvor testens navn og innledningstekst står øverst. Videre kommer mer informasjon om testen. Dette er antall forsøk studenten har på testen, godkjentkrav, gruppestørrelse og eventuell tidsfrist. Informasjonen studenten får her er litt forskjellig etter om det er en statustest eller en obligatorisk test. Studenten må taste inn brukernavn for de andre som eventuelt skal være med på testen, og trykker på "Start testen" når han er klar.

Hvis inntastede brukernavn for gruppelemmer ikke finnes i studentdatabasen, gis det melding om hvilke det gjelder og en link til registreringssiden for studenter. Studenten kan bla frem og tilbake i spørsmålene når testen har startet. Han kan be om hint og avgi svar. Spørsmålene kommer i tilfeldig rekkefølge. Det foretas en trekning blant de spørsmålene som alltid skal være med først. Deretter trekkes det blant de spørsmålene som er med tilfeldig. Til slutt stokkes alle de uttrukne spørsmålene nok en gang. Når testen er ferdig, når det er avgitt svar på alle spørsmål får studenten opp ett skjermbilde med status og resultat for testen. Antall riktige og gale svar oppsummeres og studenten får en karakter hvis det skal gis karakter på testen.

hig.no info

**Registrere Student.**  
 Du må registrer deg før du kan ta ibruk TVIL. Velg et brukernavn som er lett å huske, gjerne det samme som du har fått tildelt av HIG til bruk på datalabbene.

Glemmer du passordet, er det bare å oppgi brukernavnet, og vi sender deg passordet på den oppgitte mailadressen. For ordens skyld er det også lurt å oppgi en mailadresse som du sjekker regelmessig.

Fornavn:   
 Etternavn:   
 Du går i klasse:   
 Ønsket brukernavn:   
 Ønsket passord:   
 Gjenta passord:   
 epostadresse:

OK Avbryt

Figur 4.17: Registrering av student.

hig.no info

**1 VIS LISTE OVER TESTER**

**2 SJEKK GODKJENTSTATUS**

**3 LOGG UT AV TVIL**

**INNLOGGET:**  
 Ola Normann  
 Klasse: 98hindb  
 mail: ola\_norm@hig.no

**1 ENDRE INFO**

Velkommen til TVIL-student.  
 Skal du ta en test? Trykk "Trykk vis liste over tester", nr 1.  
 Sjekke om en test du har tatt er godkjent? Trykk "sjekk godkjentstatus" nr 2.  
 For sikkerhets skyld bør du logge ut når du er ferdig.

Figur 4.18: Studentens valg etter innlogging.

hig.no info

Ingen tilgjengelige obligatoriske oppgaver.

**Tilgjengelige statustester:**

Testnavn:	Utvidet:	Fag:	Fra - Til:
Kapittel 1 fra læreboka		F-130A Programmering mot WWW	01.05.2001 - 23.05.2001

**Lukkede obligatoriske oppgaver:**

Testnavn:	Utvidet:	Fag:	Fra - Til:
Obligatorisk oppgave 1		F-130A Programmering mot WWW	18.05.2001 - 18.05.2001
Obligatorisk oppgave 2		F-130A Programmering mot WWW	15.06.2001 - 31.06.2001

**Lukkede statustester:**

Testnavn:	Utvidet:	Fag:	Fra - Til:
Kapittel 2 fra læreboka		F-130A Programmering mot WWW	23.05.2001 - 01.06.2001

< TILBAKE  
 1 LISTE OVER TESTER  
 2 GODKJENTSTATUS  
 3 LOGG UT  
 i PERSONLIG INFO

Figur 4.19: Studentens oversikt over tester.

hig.no info

Kapittel 1 fra læreboka

Statusstest som tar for seg kapittel 1 i læreboka.

Lykke til.

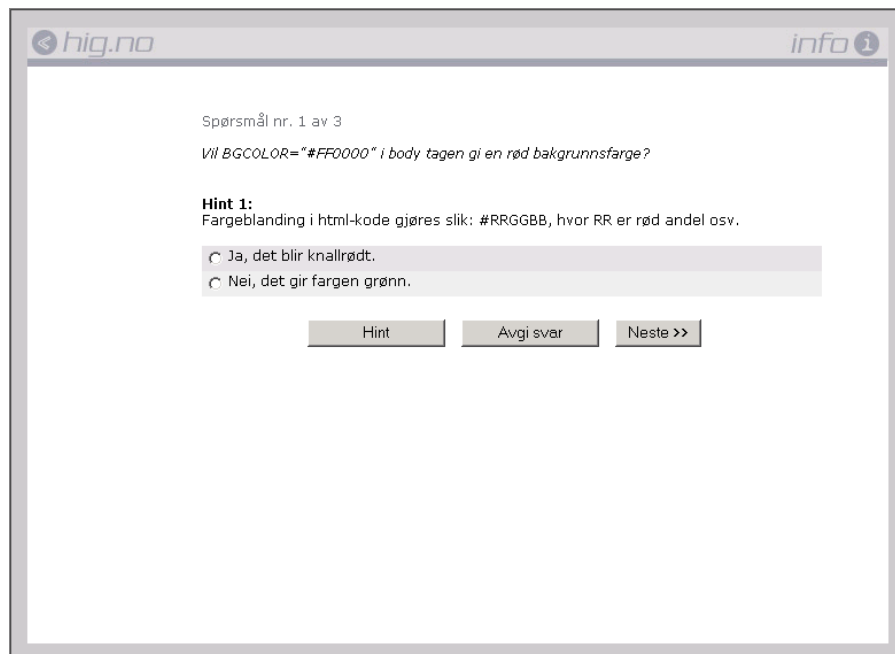
Antall forsøk: 1  
 Godkjenning: 80%  
 Minimum gruppestørrelse: 1  
 Maksimum gruppestørrelse: 4  
 Tidsfrist: 2 timer

Deltager nr. 1: Deltager nr. 2: Deltager nr. 3: Deltager nr. 4:  
 ola\_norm

Start testen

< TILBAKE  
 1 LISTE OVER TESTER  
 2 GODKJENTSTATUS  
 3 LOGG UT  
 i PERSONLIG INFO

Figur 4.20: Studenten starter en test.



Figur 4.21: Testen har startet.

### 4.1.5 Superbrukermodulen

Superbrukeren har ubegrenset tilgang til systemet etter innlogging. Superbrukeren må logge på systemet med brukernavn og passord. Har superbruker glemt passordet sitt kan han få det tilsendt på mail ved å klikke på "Glemt passord". Han må da fylle inn brukernavnet sitt for å få tilsendt passordet på registrert mailadresse. Etter innlogging møter superbrukeren en meny der han kan gjøre sine valg.

Han kan administrere klasselister, lærere, angi forklaring til de forskjellige spørsmålstypene, administrere godkjentlister, se enkel statistikk over systemet, administrere fag og tester samt endre informasjon om seg selv.

#### 4.1.5.1 Klasselister auto nye/slette

Her får superbruker anledning til å opprette og slette klasser automatisk eller manuelt. Han kan også endre på klassenavnene manuelt. En automatisk opprettelse av klasser foregår slik at systemet beregner utfra hvilket år vi er i hvilke klasser som skal opprettes. Har vi 30 klasser der klassenavnet begynner på 01 og fortsetter med xxxx i databasen vil systemet foreslå å opprette 30 klasser med navn 02xxxx der xxxx er navnet på linjen. Dette forutsetter at superbruker oppretter disse klassene før året er omme. Superbruker krysser av for de klassene han godkjenner skal opprettes. Systemet sørger for innlegging i databasen.

Automatisk sletting foregår på samme måte, men her hentes tre år gamle klasser fra databasen og foreslås slettet. Systemet vil i år 2001 foreslå å slette klasser med 98 som prefiks. 98hindb, 98hinda med flere vil komme opp foreslått slettes. Superbruker krysser av for de klassene han godkjenner at skal slettes. Systemet fjerner klassene fra databasen når han trykker slett.

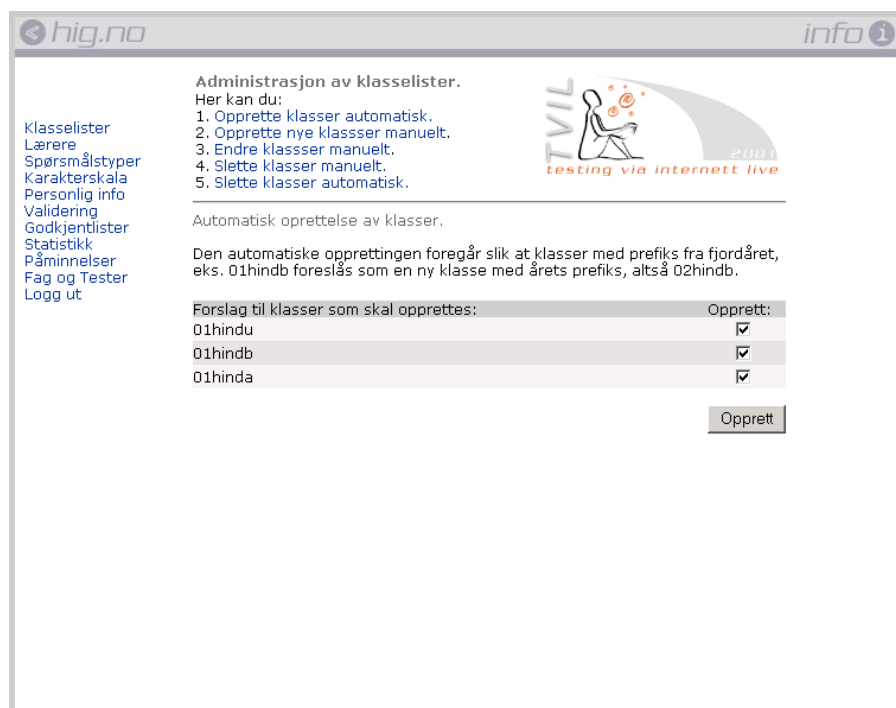
#### 4.1.5.2 Manuelt nye

Superbrukeren får se alle allerede registrerte klasser i tabellen til høyre. Han skriver inn klassenavnet og trykker "legg til". Klassen lagres i databasen og vinduet oppdateres. Se figur 4.23.

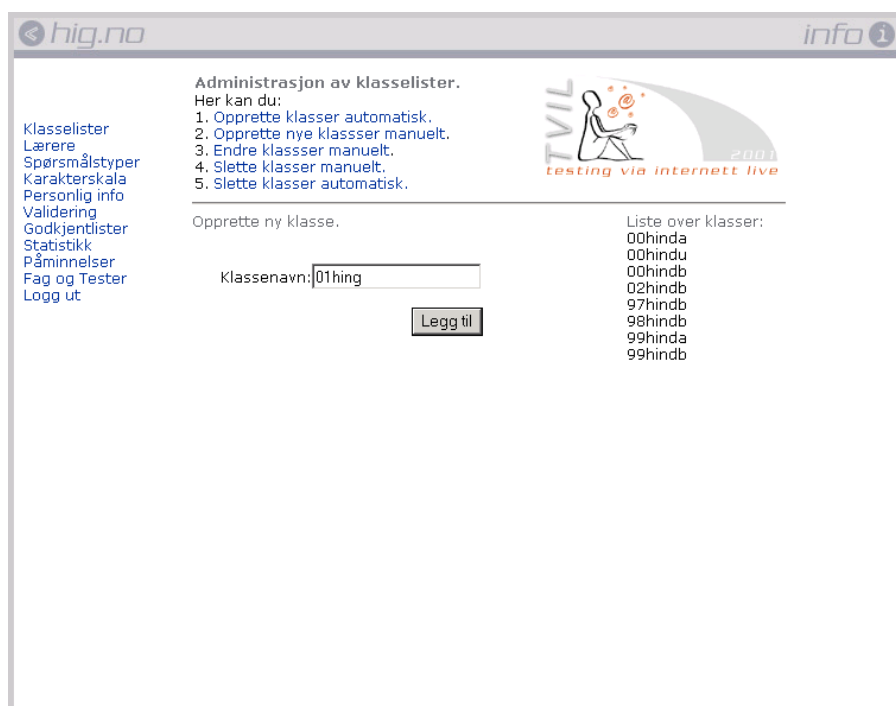
#### 4.1.5.3 Endre klasse

Her får superbrukeren listet alle klasser i tabellen til høyre. Han må klikke på klassen han vil endre på i denne tabellen. Klassenavnet kommer da opp i tekstboksen og kan endres. superbruker trykker "endre" og faget legges på ny inni databasen. Vinduet oppdateres. Se Figur 4.24.

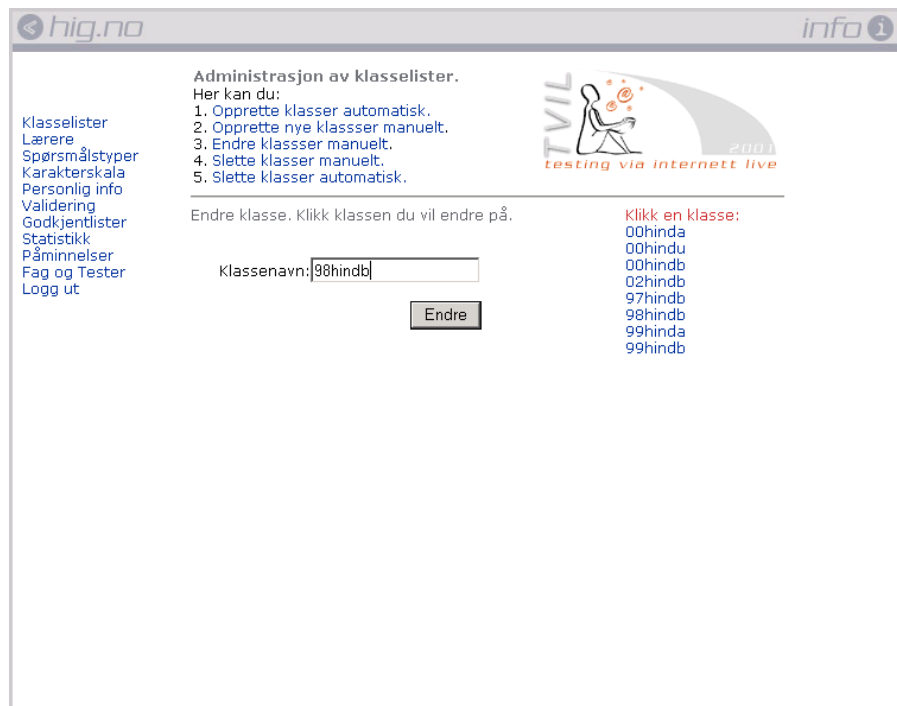




Figur 4.22: Automatisk oppretting av klasselister.



Figur 4.23: Registrer ny klasse.



Figur 4.24: Endre klasse.

#### 4.1.5.4 Slette klasse

Her får superbrukeren også listet alle klasser i tabellen til høyre. Han må klikke på klassen han vil slette i denne tabellen. Klassenavnet kommer da opp i tekstboksen. Superbruker må deretter trykke "slett" og klassen fjernes fra databasen. Vinduet oppdateres.

#### 4.1.5.5 Lærere

Her kan superbruker legge til, slette og endre en lærers data. Han har også mulighet for å overføre en lærers fag til en annen.

#### 4.1.5.6 Ny lærer

Superbruker ser en liste til høyre i skjermbildet med allerede registrerte lærere. Han må skrive inn brukernavn, navn, passord og den nye lærerens epostadresse i de respektive tekstboksene. Etter å ha trykket "Legg til" lagres læreren i databasen. Vinduet oppdateres. Se Figur 4.25.

#### 4.1.5.7 Endre lærer

Superbruker ser også her listen til høyre i skjermbildet. Han må trykke på læreren som skal endres. Når dette er gjort vil lærerens data komme frem i tekstboksene. Nå kan superbruker endre på lærerens data, bortsett fra brukernavnet. Når han trykker "Endre" lagres endringene i databasen. Se Figur 4.26.

#### 4.1.5.8 Slette lærer

Lærerne listes også her i tabellen til høyre. Superbruker må klikke læreren som ønskes slettet. Lærerens data kommer til syne i tekstboksene. Dersom superbruker trykker "Slett" vil læreren slettes fra databasen. Vinduet oppdateres. Se Figur 4.27.

hig.no info

**Administrasjon av lærere.**  
Her kan du:

1. Opprette ny lærer.
2. Slette en lærer.
3. Endre på en lærer.
4. Overføre en lærers fag til en annen

**TVIL**  
testing via internett live

Klasselister  
Lærere  
Spørsmålstyper  
Karakterskala  
Personlig info  
Validering  
Godkjentlister  
Statistikk  
Påminnelser  
Fag og Tester  
Logg ut

Ny lærer. Fyll inn skjema.

Brukernavn:

Navn:

Passord:

Epost:

Legg til

Liste over Lærere:  
Thomas Pettersen  
Terje Kirkemyr Nilsen  
Stina Hagen  
Frode Haug  
Fornavn Etternavn

Figur 4.25: Ny lærer.

hig.no info

**Administrasjon av lærere.**  
Her kan du:

1. Opprette ny lærer.
2. Slette en lærer.
3. Endre på en lærer.
4. Overføre en lærers fag til en annen

**TVIL**  
testing via internett live

Klasselister  
Lærere  
Spørsmålstyper  
Karakterskala  
Personlig info  
Validering  
Godkjentlister  
Statistikk  
Påminnelser  
Fag og Tester  
Logg ut

Endre lærer. Klikk lærerens navn i listen til høyre.

Brukernavn: tho\_pett

Navn:

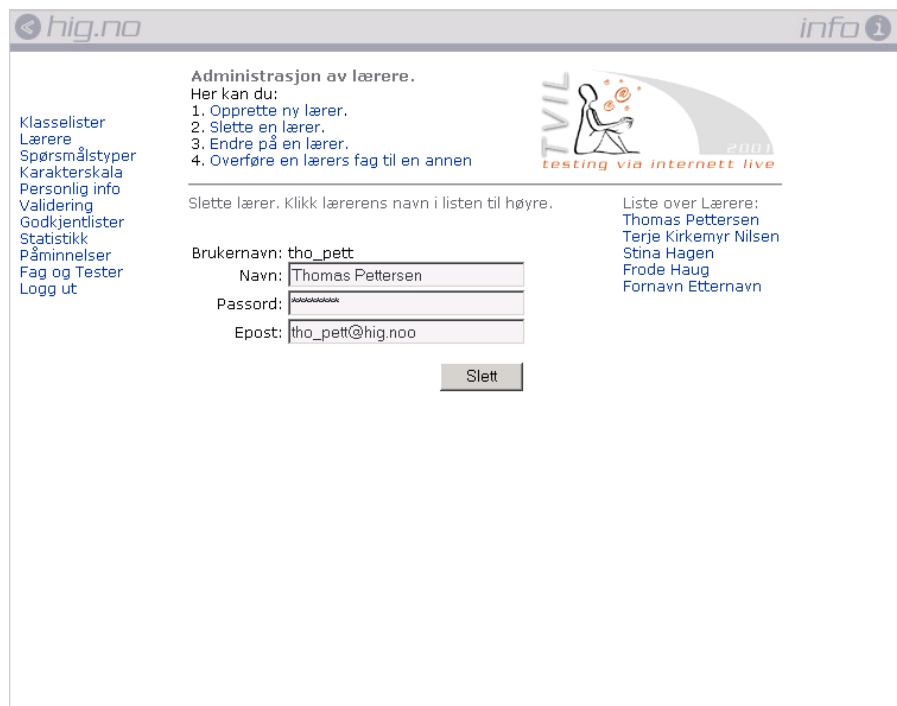
Passord:

Epost:

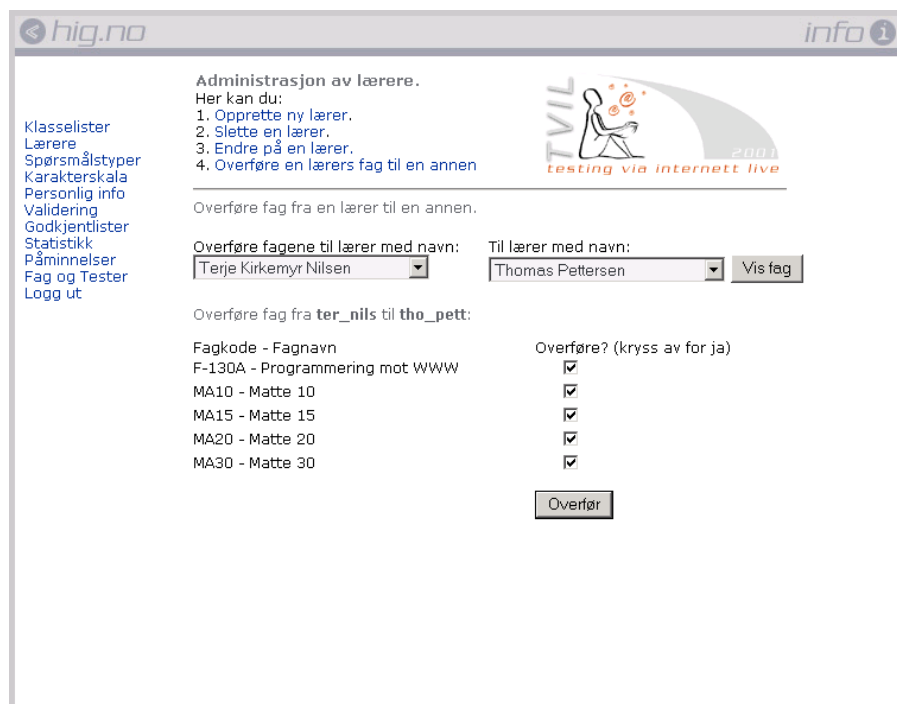
Endre

Liste over Lærere:  
Thomas Pettersen  
Terje Kirkemyr Nilsen  
Stina Hagen  
Frode Haug  
Fornavn Etternavn

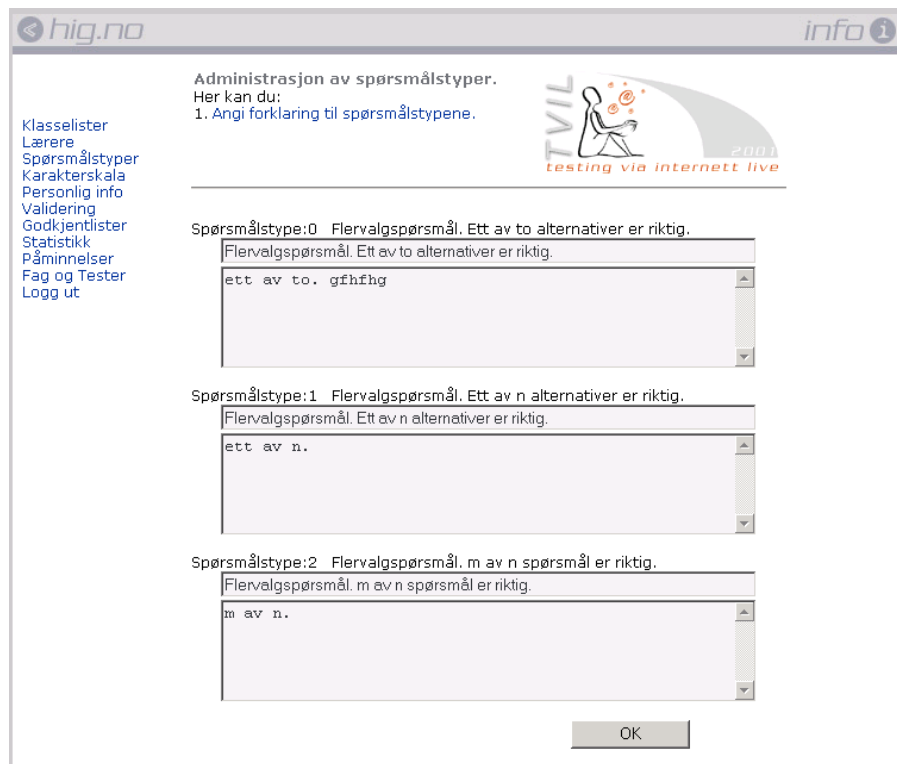
Figur 4.26: Endre lærer.



Figur 4.27: Slette lærer.



Figur 4.28: Overføre fag mellom lærere.



Figur 4.29: Forklarende tekst for spørsmålstyper.

#### 4.1.5.9 Overføre et fag fra en lærer til en annen

Her vises to nedtrekksmenyer. Superbruker velger hvilken lærer det skal overføres fra og hvilken lærer det skal overføres til. Dernest trykkes "vis fag". Alle fagene til læreren han valgte å overføre fra listes opp. Her kan superbruker markere de fagene det er aktuelt å overføre. Etter å ha trykket "Overfør" gjøres nødvendige endringer i databasen og fagene har skiftet eier. Se Figur 4.28.

#### 4.1.5.10 Spørsmålstyper

De forskjellige spørsmålstypene har en forklarende tekst som studenten kan spørre etter dersom han er usikker på hva de går ut på. Superbruker gir denne teksten. Denne siden viser alle spørsmålstyper med hva spørsmålstypen kalles og den forklarende teksten. Han kan gjøre endringer i både navnet på spørsmålstypen og den forklarende teksten. Når endringene er gjort trykker han OK og endringene lagres i databasen. Vinduet oppdateres. Se Figur 4.29.

#### 4.1.5.11 Karakterskala

Ikke implementert.

#### 4.1.5.12 Personlig info

Superbruker kan endre sine data. Han kan endre brukernavn, navn, passord og epostadresse. Siden viser tekstfelder med superbrukers data. Han kan endre disse som han selv vil. Når han trykker OK vil han bli sendt til innloggingssiden for å måtte logge på på nytt.

#### 4.1.5.13 Validering

Ikke implementert.

#### **4.1.5.14 Godkjentlister**

Siden inneholder en liste over samtlige tester som finnes i databasen, sortert på fagkode og deretter på type. Superbruker kan lagre godkjentlistene for testene til fil, men han kan også tømme listene. Velger læreren å lagre listen til fil, kommer det opp en dialogboks som spør brukeren om hvor han vil lagre filen. Filen inneholder informasjon om brukernavn, fornavn, etternavn, bestått/ikke bestått/ikke tatt testen. Listen er sortert classesvis.

#### **4.1.5.15 Statistikk**

Ikke implementert.

#### **4.1.5.16 Påminnelser**

Ikke implementert.

#### **4.1.5.17 Fag og tester**

Siden viser samtlige fag og tester i databasen. Superbruker har mulighet til å slette fag med alle dets tester. Dette gjøres ved at han merker av de fag han vil slette. Valgte fag og testene som tilhører faget slettes når superbruker trykker "Slett".

Superbruker har også mulighet til å slette en gitt test under et fag. Han må da klikke på fagnavnet i tabellen for å få en oversikt over testene som ligger under dette faget. Her kan han markere de testene han vil slette. Superbruker må trykke på "slett" for å få fjernet testene. All tilhørende informasjon som for eksempel spørsmålene til testen, slettes også.

#### **4.1.5.18 Logg ut**

Det er tryggest av superbruker bruker "logg ut" knappen når han forlater løsningen. Alternativt kan han lukke nettleseren når han er ferdig. Skulle han verken logge ut eller lukke nettleseren kan uvedkommende få adgang til systemet. Går det mer enn en time vil allikevel løsningen har sperret for adgang, det har gått for lang tid uten aktivitet.

## Kapittel 5

# Implementering

### 5.1 Valg av programmeringsspråk

PHP ble valgt da vi synes dette er det beste språket for utvikling av dynamiske nettløsninger der grafisk brukergrensesnitt spiller en rolle. PHP4 er dessuten laget for å kjøre på Linux servere, noe vi finner særdeles raskt og stabilt. Disse serverne begynner å bli mer og mer vanlig i utdanningsinstitusjoner fordi de er stabile og krever minimalt med vedlikehold. I operativsystemet Linux ligger det en webserver som kalles Apache. Apache er den mest utbredte webserveren på verdensbasis. Den er også gratis i anskaffelse og bruk. Alternative programmeringsspråk ville være å benytte JavaServlets eller ASP. ASP ble fort skrinlagt da vi satte høye krav til oppetid av systemet. Flere tilbakemeldinger fra erfarne webutviklere sier at ASP ikke er så stabilt og raskt som PHP eller JavaServlets. Dette har vi også selv erfart ved mindre løsninger der vi har benyttet ASP. JavaServlets ville ha vært et alternativ hadde det ikke vært stilt så høye krav til det grafiske brukergrensesnittet. JavaServlets er lite fleksibelt med tanke på dette i forhold til PHP og ASP. To av medlemmene i gruppa kjente best til PHP fra før og var kjent med resurser og hjelpemidler laget for dette språket. Vi ville derfor spare mye tid på opplæring som vi hadde trengt ved valg av ASP eller JavaServlets.

Vi valgte å benytte en rask og stabil database. Så lenge løsningen skulle kjøres på en Linux maskin falt valget på MySQL. MySQL er en rask, stabil og godt implementert database til bruk i nettløsninger. MySQL er utviklet for å kjøre på Linux. Databasen er gratis i anskaffelse og bruk.

### 5.2 Valg av metode

Vi beskriver de spesielle metodene som ble benyttet for utvikling av systemet her. Alle detaljer er ikke beskrevet, men alle filer er vedlagt på cd-rom dersom det er ønske om å studere disse.

Vi benyttet Modelator for å designe databasen, dette programmet har en innebygd funksjon som eksporterer databasemodellen til et sqlscript. Vi tok utgangspunkt i det eksporterte scriptet og korrigert litt på dette slik at vi kunne benytte fordelene som MySQL har i forhold til vanlig sql. Det viktigste her er *autoincrement* som det ikke var støtte for å modellere rett inn i Modelator.

```
# Tabellstruktur for tabell 'laerer'
# Lærere registreres i denne tabellen:
CREATE TABLE laerer (
brukeravn varchar(64) DEFAULT '' NOT NULL, # Lærerens brukeravn
passord varchar(16) DEFAULT '' NOT NULL, # Lærerens passord
navn varchar(64) DEFAULT '' NOT NULL, # Lærerens virkelige navn
epost varchar(255), # Lærerens epostadresse
PRIMARY KEY (brukeravn)
);
```

Vi gikk gjennom hver tabell som ble generert fra eksport funksjonen i Modelator og korrigererte disse slik at vi fikk ønsket databasestruktur. Her ser vi tabellen laerer som lagrer informasjon om hver lærer som er registrert på systemet. Viktige felter her er brukeravn og passord. Brukeravnet er unikt for hver lærer, og benyttes for å logge inn på lærermodulen.

```

# Tabellstruktur for table 'fag'
CREATE TABLE fag (
  fagkode varchar(16) DEFAULT '' NOT NULL,      # fagkode, primærnøkkel for fag
  navn varchar(64),                             # Fagets navn
  brukernavn varchar(64) DEFAULT '' NOT NULL,   # Fremmednøkkel mot 'laerer', fagets eier
  hjemmeside varchar(255),                     # URL til fagets hjemmeside
  les_kopiere tinyint(4),                      # skal faget kunne leses/kopieres av andre lærere [0 | 1]=[false | true]
  redigere tinyint(4),                         # Hvem andre kan redigere faget? [0 | 1 | 2]=[bare meg selv | alle | gruppe]
  PRIMARY KEY (fagkode),                      # Definerer primærnøkkel
  KEY FK2_laerer_fag (brukernavn)             # Indeks på fremmednøkkel for å optimalisere spørringer
);

```

Videre kommer tabellen for fagdata. Her registreres fagkode som må være unik, samt fagets navn og hjemmeside. Her settes det også parametre på om andre lærere kan lese eller kopiere faget. Feltet redigere gir informasjon om hvem av de andre lærerne enn eieren som kan editere tester og spørsmål under aktuelt fag, dette kan være "bare meg selv", "alle" eller "gruppe". Med feltet brukernavn lager vi en relasjon mot tabellen lærer, og kan da bestemme hvem som er eier av faget.

Hele sql scriptet ligger ved som vedlegg nr. 4. Alle tabellene og relasjonene mellom disse kan vi se i Figur 4.3. Databasen består av 19 entitetstyper og 137 attributter, dette gir et inntrykk av størrelsen og kompleksiteten på databasen.

Mange av attributtene i databasen ble brukt for å angi innstillinger. Dette er heltallsverdier som har spesielle betydninger. Listen under forklarer dette mer spesifikt. Vi gjorde det på denne måten fordi disse verdiene var lette å sjekke mot når vi skulle programmere løsningen. Dette gjør også at nye verdier kan legges til for å få flere valgmuligheter for en innstilling. Blant annet ble attributtet "spmtype" i tabellen "sporsmaal" brukt til å angi hvilken spørsmålstype det aktuelle spørsmålet var. Tallet 0 eller at feltet er tomt betyr hvor annet ikke er spesielt beskrevet at funksjonen ikke er i bruk.

#### Forklaringer på spesielle innstillinger i tabellen test:

**tidsfrist\_enhet** Enheten som gjelder for tidsfristen på spørsmålet, 0=sekunder, 1=minutter, 2=timer

**kreves\_godkjent** Annen test som kreves godkjent før denne testen kan tas. Er feltet tomt kreves ingen annen test godkjent før aktuell test.

**riktig\_for\_neste** Må riktig svar avgis på et spørsmål før man kan ta neste spørsmål i testen, 0 betyr nei, 1 betyr ja

**karacterskala** Karacterskala for tilbakemelding på resultat, 0 betyr hel/halv karakterer, 1 betyr tidels karacterskala.

**vis\_alle\_spm** Alle spørsmål skal være med i testen, uansett om det noen av spørsmålene er satt til å være med tilfeldig. 0 betyr nei, 1 betyr ja

**retting\_fortlopende** Skal studenten få retting etter hvert spørsmål, 0 betyr nei, 1 betyr ja.

**redusere\_poeng** Skal studentens poeng for spørsmålet reduseres med en ved feil svar eller hint? 0 betyr nei, 1 betyr ja

**fasit\_til\_slutt** Skal studenten få vite fasit på alle oppgaver i testen til slutt? 0 betyr nei, 1 betyr ja.

**poeng\_etter\_hvert** Skal studenten få vite poengsum etter hvert spørsmål? 0 betyr nei, 1 betyr ja.

Alle spesielle innstillinger og verdier for disse i databasen er spesifisert i vedlegg 5, databaserapport. Hele databasescriptet ligger i en egen fil "Endelig\_database.sql" som kan kjøres for å få opprettet database ved installasjon av systemet.

Da databasen var ferdig utviklet startet vi på jobben med brukergrensesnitt for lærermodulen. I arbeidet med dette brukte vi Dreamweaver 4.0. Dette er en kraftig WYSIWYG HTML editor som gjorde vårt arbeid med utviklingen av brukergrensesnitt en del raskere. Videre kodet vi PHP-script som tar seg av kommunikasjon med databasen. Vi var også nødt til å benytte en god del DHTML for å tilfredsstille oppdragsgivers krav til funksjonalitet. Dette var nytt for oss alle så vi brukte en del tid på å sette oss inn i dette.



Innloggingsfunksjonen for lærermodulen ble kodet først, denne sjekker brukernavn og passord opp mot databasen. Denne filen tar seg også av cashekontroll, dette gjelder både på klient og serversiden. Ingen av sidene i lærermodulen skal caches, da de kreeres dynamisk ved aksessering. Scriptet benytter Apaches innebygde autentiseringskontroll, og sjekker det inntastede brukernavnet og passordet mot databasen. Ett kritisk punkt for denne innloggingsfunksjonen er sjekk på om inntastet passord og passord i databasen er like, hvis brukeren taster inn et brukernavn som ikke finnes i databasen blir tomt passord godkjent hvis vi ikke i tillegg sjekker om brukernavnet ble funnet i databasen. Se følgende kodeeksempel for å se hvordan vi løste dette. Denne filen er inkludert i toppen av alle sider som skal passordbeskyttes i denne modulen.

### [lærerlogin.php]

```
<?
// Loginsjekk for lærermodulen, inneholder i tillegg cashecontrol funksjonelitet:
// cachefiz gjøres i http-header:
header ("Expires: Mon, 26 Jul 1997 05:00:00 GMT"); // Dato i fortiden
header ("Last-Modified: " . gmdate("D, d M Y H:i:s") . " GMT"); // Alltid modifisert
header ("Cache-Control: no-cache, must-revalidate"); // HTTP/1.1
header ("Pragma: no-cache"); // HTTP/1.0
if (!isset($PHP_AUTH_USER))
{
Header("WWW-Authenticate: Basic realm=\"Lærerlogin\"");
Header("HTTP/1.0 401 Unauthorized");
echo "Dette er et passordbeskyttet område, korrekt passord må testes inn.\n";
exit;
}
else
{
// Kobler til databasen og sjekker passordet:
include "../system/database.php";
$sql = "SELECT passord FROM laerer WHERE brukernavn='".$PHP_AUTH_USER'";
$resultat = mysql_query($sql)
or die("SQL-Feil: $sql");

$rad = mysql_fetch_array($resultat);

// Nå sjekke med mysql_num_rows, ellers blir ukjent brukernavn og tomt passord godkjent:
if (strcmp($rad["passord"], $PHP_AUTH_PW) != 0 // mysql_num_rows($resultat) != 1)
{
Header("WWW-Authenticate: Basic realm=\"Lærerlogin\"");
Header("HTTP/1.0 401 Unauthorized");
echo "Dette er et passordbeskyttet område, korrekt brukernavn og passord må testes inn.\n";
exit;
}

mysql_free_result($resultat);
}
?>

<META HTTP-EQUIV="Expires" CONTENT="0"> <!-- metatag som skal forhindre klienside caching -->
```

I lærermodulen hadde vi ganske ofte behov for å åpne nye vinduer, f.eks. ved registrering og endring av fag/kurs, tester og spørsmål. Dette gjorde vi med et javascript. Det som er spesielt i dette javascriptet er at størrelse og plassering av de nye vinduene er avhengig av oppløsningen som brukeren kjører på. Dette måtte vi gjøre fordi vi ellers kunne risikere at deler av de nye vinduene kunne bli plassert utenfor synlig område av skjermen til brukeren. I kodeeksemplet under kan du se javascriptfunksjonen som løser dette.

```
function nytt_vindu(url, tittel)
{
l = (screen.availwidth/2) - (500/2); // Plasserer vinduet midt på skjermen i horisontal retning
if (screen.availheight < 640) // Hvis tilgjengelig oppløsning i horisontal retning er mindre enn 640
{
h = screen.availheight-55; // Vinduet høyde settes til tilgjengelig høyde trukket fra 55 piksler
t = 0; // Vinduet plasseres helt øvers på skjermen
}
else // Hvis tilgjengelig høyde er større eller lik 640
{
t = (screen.availheight/2) - (630/2); // Vinduet plasseres midt på skjermen i vertikal retning
h = 640; // Vinduet høyde settes til 640
}
// Åpner vinduet
vindu = window.open(url, tittel, 'scrollbars, resizable=yes, status=no, width=500,height=' + h + ',left=' + l + ',top=' + t);
vindu.focus(); // Setter fokus på vinduet.
}
}
```

Når nytt fag skal opprettes eller et fag skal editeres har vi lagd en noe spesiell løsning hvor klasser som skal ha tilgang til fagets tester kan velges fra en liste over alle klasser og flyttes over i en annen liste. Dette er beskrevet nærmere i kapitlet om design. Her har vi også brukt javascript. På dette tidspunkt var ikke avgjørelsen om at løsningen i første omgang skulle fungere i Internet Explorer 5.0 tatt, så akkurat denne funksjonen måtte vi jobbe litt ekstra med for å få den til å fungere i Netscape Navigator 4.7 også. Vi klarte å skrive en egen funksjon som gjorde dette, men den fungerte i kun i Internet Explorer 5.0. Etter en del leting på Internett fant vi et sett med funksjoner som gjorde akkurat det vi trengte, og den fungerte i Netscape Navigator 4.7. Her er begge listene til enhver tid sortert alfabetisk. Valg av hvem som kan editere faget er også implementert på samme måte. Se følgende kodeeksempel for å se hvordan javascriptfunksjonene som styrer denne spesielle funksjonaliteten er skrevet. Funksjonene fant vi på <http://javascript.internet.com/forms/menu-swapper.html>

```
// Funksjon som flytter elementer mellom to listbozer:
function move(fboz,tboz)
{
    for(var i=0; i<fboz.options.length; i++)
    {
        if(fboz.options[i].selected && fboz.options[i].value != "")
        {
            var no = new Option()
            no.value = fboz.options[i].value
            no.text = fboz.options[i].text
            tboz.options[tboz.options.length] = no
            fboz.options[i].value = ""
            fboz.options[i].text = ""
        }
    }
    BumpUp(fboz)
    SortD(tboz)
    unselect(fboz)
}

// Funksjon som fjerner tomme elementer i en listboz (Netscape Navigator krever denne)
function BumpUp(boz)
{
    for(var i=0; i<boz.options.length; i++)
    {
        if(boz.options[i].value == "")
        {
            for(var j=i; j<boz.options.length-1; j++)
            {
                boz.options[j].value = boz.options[j+1].value
                boz.options[j].text = boz.options[j+1].text
            }
            var ln = i
            break
        }
    }
    if(ln < boz.options.length)
    {
        boz.options.length -= 1
        BumpUp(boz)
    }
}

// Funksjon som sorterer innholdet i en listboz i alfabetisk rekkefølge:
function SortD(boz)
{
    var temp_opts = new Array()
    var temp = new Object()
    for(var i=0; i<boz.options.length; i++)
    {
        temp_opts[i] = boz.options[i]
    }
    for(var x=0; x<temp_opts.length-1; x++)
    {
        for(var y=(x+1); y<temp_opts.length; y++)
        {
            if(temp_opts[x].text > temp_opts[y].text)
            {
                temp = temp_opts[x].text
                temp_opts[x].text = temp_opts[y].text
                temp_opts[y].text = temp
                temp = temp_opts[x].value
                temp_opts[x].value = temp_opts[y].value
                temp_opts[y].value = temp
            }
        }
    }
}
```

```

    }
    for(var i=0; i<boz.options.length; i++)
    {
        boz.options[i].value = temp_opts[i].value
        boz.options[i].text = temp_opts[i].text
    }
}

```

Både under registrering og editering av tester og spørsmål har vi avanserte innstillinger som har standardverdier som ikke er nødvendig å endre for å ta systemet i bruk. Disse avanserte innstillingene kan brukes av mer fortrolige brukere. Disse er i utgangspunktet skjult for brukeren, han må trykke på en pil for å åpne dem. Dette gjøres også med javascript og manipulering av stylesheetverdier, såkalt DHTML. Annen funksjonalitet som vi benyttet samme teknikk på er flervalgsspørsmål når skjermbildet endres avhengig av hvilken undertype som er valgt og når svaralternativer legges til dynamisk. På denne måten unngår vi å laste siden på nytt ved en del anledninger, og dette er etter oppdragsgivers ønske. Her kan du se javascriptfunksjonen som åpner og lukker de avanserte innstillingene.

```

function toggle_advanced(pic, element)
{
    z=document.getElementById("avanserte").style;
    if(z.overflow == 'hidden')
    {
        pic.src="up.gif";
        z.overflow= 'visible';
    }
    else
    {
        pic.src="down.gif";
        z.overflow= 'hidden';
    }
}

```

Vi har implementert "Nytt spørsmål" helt etter oppdragsgivers ønsker. Her åpnes nye vinduer når filer skal tilegnes spørsmålstekst, svaralternativer, hint og tilbakemeldinger. Dette løses ved at tilhørende "hidden" form elementer på hovedsiden for nytt spørsmål settes ut fra hvilke filer som velges og om de skal auto-vises. Dette gjorde vi også dynamisk ved hjelp av javascript. På denne måten klarte vi å sende html-format én gang for å registrere alle data om et spørsmål. Dette var viktig da brukeren også skulle kunne trykke "Avbryt" hvis han ikke ville endre på/opprette nytt spørsmål.

Da vi var ferdige med å utvikle lærermodulen med første spørsmålstype startet vi med utviklingen av studentmodulen. Dette var noe oppdragsgiver ønsket ettersom han var interessert i å få ferdig et system som var klart til å kjøre med bare en spørsmålstype i første omgang. Flere spørsmålstyper kan enkelt utvikles og legges til ved et senere tidspunkt.

I presentasjonsdelen i studentmodulen ble vi ikke helt ferdige med å implementere all funksjonalitet. Vi ble ferdige med hovedtrekkene, men en del av de avanserte innstillingene i lærermodulen rakk vi ikke å implementere her. Vi startet med å lage innloggingsside for studentene, for deretter å lage oversikten over tester. Denne oversikten viser de åpne obligatoriske testene og statustestene i to tabeller øverst. Nederst kommer en liste over tester som er lukket. Studenten får en oversikt over testens innstillinger før testen startes. Dette er slikt som antall forsøk på testen, godkjentkrav og andre testrelaterte data som leses direkte ut fra databasen. Før testen starter blir testens spørsmål trukket ut slik at riktig antall spørsmål i testen blir riktig i forhold til de innstillinger som er gjort i lærermodulen. Spørsmål blir først trukket fra de spørsmålene som alltid skal være med, videre blir det trukket fra de spørsmålene som er med tilfeldig til det er korrekt antall spørsmål i testen. ID for alle spørsmål som skal være med i testen blir lagret i en array som blir registrert som en sesjonsvariabel. Dette gjorde vi fordi studentene skulle kunne bla frem og tilbake i spørsmålene etter at testen hadde startet. Se kodeeksempelet under for å se hvordan trekking av spørsmål i testen ble kodet.

```

første del av [ta_test.php]
<?
/*
ta_test.php:
Parametre inn:
tID som er aktuell test's testID i databasen.
Initierer sesjonsvariable, viser testinformasjon for studenter og sjekker om testen virkelig er åpen.
Alle testens spørsmål trekkes vilkårlig ut i fra antall spørsmål som alltid skal være med og spørsmål
som skal være med tilfeldig og totalt antall spørsmål i testen.

```

```

Eventuelt flere deltagere enn pålogget student kan testes inn og postes til test.php som sjekker om disse
brukernavnene er gyldige.
*/
// sjekker om bruker har logget på
$username="";
session_start(); // Starter eller finner gjeldende session
session_register("testID"); // Testens id i databasen
session_register("spm_i_test"); // Alle spørsmål som skal inngå i testen
session_register("oblig"); // 0=statustest, 1=oblig
session_register("spmnr"); // Spørsmålnummer som studenten jobber på
session_register("antall_hint"); // Hvor mange hint har stud. fått på hvert spørsmål
session_register("antall_forsok"); // Antall forsøk som er brukt på hvert spørsmål
$spmnr=1;

$spm_i_test = null; // Nullstiller sessionsvariabel
$antall_hint = null;
$testID=$tID;
if (!$username)
header("Location: index.php");
// sender bruker til loginsiden dersom ikke logget på

include "../system/database.php"; // HER HENTER VI DIR-VARIABLEN
include "../topp_funksjon.php";
//sender hvilken fil som skal være meny og tittel på hele siden.
menyvalg("../meny/student.php", "TYIL");

// Klargjør spørsmål som skal være med i testen:
// Finner hvor mange spørsmål testen skal bestå av:
$sql = "SELECT antspm FROM test WHERE testID=$testID";
$res = mysql_query($sql)
or die("Feil: $sql<br>".mysql_error());
$rad = mysql_fetch_array($res);
$antspm=$rad["antspm"];

// Henter alle spørsmål som hører til testen:
$sql = "SELECT spmID, alltid_med, spm_nr FROM sporsmaal WHERE testID=$tID ORDER BY alltid_med DESC";
$res = mysql_query($sql)
or die("Feil: $sql<br>".mysql_error());

$spm_alltid_med = array ();
$spm_trekkes = array ();

// Hvis antspm i databasen er mindre enn antall som alltid skal være med
// overstyrer antall spørsmål som alltid skal være med
if($antspm<sizeof($spm_alltid_med)) $antspm=sizeof($spm_alltid_med);

while( $rad = mysql_fetch_array($res) )
{
    if($rad["alltid_med"]) // Legger spørsmål som alltid skal være med i en egen array:
        array_push($spm_alltid_med, $rad["spmID"]);
    else // Og legger spørsmål som skal trekkes i en egen array:
        array_push($spm_trekkes, $rad["spmID"]);
}
// Oppretter tabell med alle spørsmål som skal være med i testen:
srand ((double)microtime()+1000000); // Initierer random generator
shuffle ($spm_alltid_med); // Stokker arrayene:
shuffle ($spm_trekkes);

$spm_i_test[0]=0;
for($i=1; $i<=$antspm; $i++)
{
    if( $spmID = array_pop($spm_alltid_med) );
    else if( $spmID = array_pop($spm_trekkes) );
    $spm_i_test[$i] = $spmID;
}
array_shift($spm_i_test); // Shifter første element av arrayen
shuffle ($spm_i_test); // Stokker spørsmålsrekkefølgen
array_unshift($spm_i_test,0); // Shifter inn dummy element på starten av array
unset($spm_i_test[0]); // Nullstiller første element

```

Når alle sesjonsvariable er initiert er testen klar til å starte. Studenten trykker på "Start test" og testen er i gang. Når testen er startet kan studenten bla frem og tilbake i spørsmålene. Spørsmålstekst, hint, alternativ og tilbakemeldinger kan alle ha tilhørende filer. Filene presenteres med ikoner som symboliserer filens type ved siden av riktig element. Vi brukte en og samme fil for å hente ut aktuelle filer fra databasen, og et enkelt javascript som automatisk åpner filene hvis de er satt med auto-visning. For å vise en fil setter vi opp en link til denne filen med filens ID som parameter. Ved å følge denne linken får studenten riktig fil tilbake. Bilder åpnes i et eget vindu, mens audio, video, dokumenter og programmer lastes ned og åpnes i det program som

er tilordnet denne filtypen i Windows. Linken peker til <hent\_fil.php?filID=ID> i dokumentet, hvilken fil som blir returnert fra databasen er avhengig av hvilken filID som blir medsendt. Se i kodeeksempelet under hvordan filen hentes ut fra databasen og returneres til brukeren.

```
<?
if($filID) {
    include "database.php"; // Hvis filid er medsendt
    $query = "select navn, data, mimetype from fil where filID = '$filID'"; // Oppretter tilkobling til databasen
    $result = mysql_query($query); // Kjører spørring som henter ut fildata fra databasen
    $rad = mysql_fetch_array($result);
    $binfil = $rad["data"]; // Henter filens binære data
    // Sender headerinformasjon med riktig mimetype og filnavn:
    Header("Content-type: " . $rad["mimetype"]);
    Header("Content-Disposition: filename=\"\" . $rad["navn"] . "\"");
    print $binfil; // Sender filens binære data tilbake
}
?>
```

Når svar er avgitt på alle spørsmål i testen, rettes disse og studenten får en tilbakemelding som er avhengig av lærerens innstillinger for testen. Resultatet lagres i databasen for senere generering av godkjentlister.

Superbrukermodulen ble utviklet parallelt med presentasjonsdelen av studentmodulen. Dette gjorde vi også fordi oppdragsgiver ville ha et system som gikk an å begynne å bruke når vi var ferdige med prosjektoppgaven. I denne modulen var vi ikke så avhengig av å utvikle en meget brukervennlig løsning. Dette fordi denne modulen kun skal brukes av et lite antall brukere som er godt kjent med systemet. Vi lagde et brukergrensesnitt som dekket de viktigste funksjonene slik at systemet kunne taes i bruk.

### 5.3 Bakgrunn for verktøyvalg

HiG har en Linux server med Apache som internettjener som vi kunne benytte til løsningen vår. I prosjektperioden benyttet vi en egen lokal webserver som er identisk med serveren skolen benytter. Dette gjorde oss litt mer fristilt, da vi ikke var avhengig av HiG's internettjener's opetid. Vi ville ha så høy hastighet som mulig mot serveren vi jobbet mot, og dette ble optimalt da serveren vår ble koblet på samme switch som arbeidsstasjonene våre. Databasen ble designet i Modelator. I dette programmet kan man designe databaser ved å tegne ER-diagrammer som igjen kan konverteres til sql-script. Gode muligheter for å gi utfyllende kommentarer til tabeller og felter finnes også. Selve kodingen har foregått i Emacs og Edit+. Dette er gode multi-språk editorer der du får god oversikt over koden, både med fargevalg, nummerering av linjer og oversikt over filer i samme prosjekt. Macromedia DreamWeaver 4.0 ble brukt sammen med Adobe Photoshop 6.0 og ImageReady 5.5 for layout og design. DreamWeaver 4.0 er en god editor som har støtte for WYSIWYG prinsippet der både HTML, CSS og PHP er implementert i samme fil. Selve rapporten er skrevet i LyX. LyX er utmerket til rapportskrivning, med gode muligheter for styring av oppbygging og layout.

Vi har benyttet Microsoft Project som prosjektstyringsverktøy. Dette har i hovedsak begrenset seg til Gantt-skjema og fordeling av arbeidsoppgaver samt oppfølging av disse. Vi har jobbet i både Windows og Linux miljø under utviklingen. Vi valgte å jobbe på begge plattformer fordi vi ville benytte de beste verktøyene vi hadde tilgjengelig fra de to operativsystemene.

### 5.4 Prinsipper vi har fulgt

Arbeidsoppgavene innen hver modul ble fordelt mellom gruppe medlemmene.

Koding av disse ble utført med den funksjonaliteten som skulle dekkes i henhold til tidligere definerte krav. Etter hvert som de forskjellige delmodulene ble ferdige la vi de ut på webtjeneren på et eget releaseområde. Etter en del testing på funksjonalitet, ble disse vurdert i samråd med veileder og oppdragsgiver med henblikk på både layout og funksjonalitet. Ved ønske om forandring fra veileder og oppdragsgivers side, ble kode forandret og oppdatert versjon ble lagt ut igjen for ny evaluering. Vi tilstrebet en løsning med "gode" layoutmessige løsninger. Sidene skulle fortone seg enkle og forklarende i sin form.

Konsistentitet i forhold til benevnelse /navngivelse ble tilstrebet, men noen avvik var ikke til å unngå. Alle navn på tabeller og felter i databasen er entydige; de samme benevnelsene blir brukt gjennom hele systemet. Der det ikke var til å unngå å bruke samme navn har vi valgt navn som allikevel er godt beskrivende for variabelens innhold.

Løsningen er "ren" og forklarende i sin utforming. Vi har bevisst valgt å ikke benytte oss av for mye fargebruk. Fargene vi har brukt skal være forklarende i forhold til sin funksjon. Dersom en test som for eksempel er lukket på grunn av tidssperre, vil testnavnet bli skrevet ut med rød tekst. En åpen test vil fremkomme i opplistingen med grønn tekstfarge. Ved utvelgelse av elementer fra for eksempel en nedtrekksmeny, vil valgte linje vises med blå bakgrunn.

## 5.5 Beskrivelse av forløp

Vi startet med å designe databasen. Deretter opprettet vi den på vår lokale webtjener. Neste skritt var å lage et grafisk brukergrensesnitt oppdragsgiver var fornøyd med for lærermodulen. Det var meget viktig at designet ble mest mulig likt et Windows program selv om dette er en løsning som skal kjøres i nettleseren. Internet Explorer 5.0 støtter mange flere funksjoner enn andre nettlesere. Oppdragsgiver bestemte da at lærernes modul kun er støttet i Internet Explorer 5.0 eller høyere. Studentmodulen og Superbrukermodulen er allikevel også utviklet for brukere av Netscape Navigator 4.7 eller høyere, samt Opera 5.0 eller høyere. Vi satte dernest i gang med å dele lærermodulen inn i mindre deler og fordele disse mellom oss. Det ble også bestemt at vi kun skulle utvikle spørsmålstype A i første omgang, og heller implementere flere spørsmålstyper til slutt dersom tiden strekker til. Etter at lærermodulen var ferdig startet vi med studentmodulen. Også her jobbet vi etter samme prinsipp. Det første vi gjorde var å lage et grafisk brukergrensesnitt. Her var vi mer frie til design, og vi valgte å lage det så enkelt og rent som mulig. Vi ville ha det så enkelt i bruk som mulig, slik at også uerfarne internettbrukere føler seg trygge med å bruke løsningen. Etter at designet var på plass laget vi en registreringsfunksjon. Vi synes også at det var på sin plass at studenten kan få tilsendt passordet sitt på epost, dersom han har glemt det. Denne modulen ble utviklet. Deretter laget vi innloggingsfunksjonen, listen over tester studenten kan ta, siden der studenten kan endre personlig informasjon og til slutt logg ut funksjonen. Listen som viser godkjentstatus i de forskjellige fagene studenten går i ble laget helt til slutt sammen med de andre statistikkmodulene. Dette fordi vi viste at det kunne være mulig å utvikle disse med mye gjenbruk av kode. Presentasjonsmodulen som fremlegger testene for studenten ble utviklet parallelt med superbrukermodulen og statistikkmodulene. Superbrukermodulens delfunksjoner ble prioritert av oppdragsgiver og vi utviklet de i den rekkefølgen. Det ble dessverre ikke tid til å utvikle alle punktene her, men det som ble utviklet er allikevel tilstrekkelig til at løsningen kan brukes.

## 5.6 Endelig løsning

Vår endelige løsning av hovedprosjektoppgaven ligger på vedlagte CD-rom. I mappen "Tvil" ligger selve løsningen strukturert slik at det er klart til kopiering over til en webserver. I mappen "Docs" ligger denne rapporten, samt andre dokumenter vi har brukt under prosjektet. I mappen "Script" ligger scriptet som oppretter databasen. Systemet er enkelt å integrere på andre systemer. Har man en webserver som støtter PHP og MySQL er det svært enkelt. Man kjører scriptet for opprettelse av databasen, og kopierer over alle filene fra mappen "Tvil" slik de ligger på CD'en over til webserveren.

Man må endre internettadressen og banen til løsningen i system/database.php, topp\_funksjon.php samt /laerer/topp\_funksjon\_laerer.php.

Superbrukerens passord blir satt i scriptet under opprettelse av databasen. Superbruker kan nå logge inn på systemet via nettleseren, og bør endre passordet med det samme for å unngå misbruk av systemet. Han kan nå begynne å registrere lærere som skal ha tilgang til systemet.

## 5.7 Legge til flere spørsmålstyper

Systemet er åpent for å legge til flere spørsmålstyper. Dette ble et viktig punkt i vår løsning av hovedprosjektoppgaven, da vi tidlig i prosjektfasen innså at vi ikke kunne rekke å utvikle alle spørsmålstypene. Implementering av flere spørsmålstyper vil ikke påvirke den eksisterende løsningen. Vi har i vår kravspesifikasjon definert seks spørsmålstyper, disse er også definert i database med spørsmålstypenummer 0 til og med 5. Databasen kan utvides til å dekke flere spørsmålstyper, disse må da gis et spørsmålsnummer økende fra seks.

## Kapittel 6

# Kvalitetssikring, testing og realisering

### 6.1 Organisering av kvalitetssikring

Prosjektgruppa har ikke utarbeidet en formell kvalitetsplan for prosjektet, men fulgte punktene som er listet opp i dette kapitlet. Kvalitetsgjennomganger ved viktige milepæler ble til dels benyttet. Gruppa hadde ingen interne grupperegler/kjøreregler for prosjektperioden.

#### 6.1.1 Dokumentasjon

Alle filer ble dokumentert underveis i form av vanlig kommentarer til koden, samt en standard header med utdypende beskrivelse av hva fila inneholder/gjør eller beskrivende filnavn. Vi har tilstrebet å dokumentere hver fil fullstendig med hvilke parametre den tar i mot og hva den sender videre etter at fila er ferdig utviklet. Alle filer, kataloger, klasser, funksjoner og dokumenter til kildekode ble også versjonifisert.

#### 6.1.2 Standarder, rutiner, konvensjoner

Vi har utarbeidet navnekonvensjon for filer og variable. Eksempelvis har vi brukt samme navn på variable i PHP-koden som i databasen. Vi lagde en felles mal for PHP-filer som inneholder header, parameterliste og forklarende tekst.

Vi har tatt daglig backup av prosjektområdet på serveren. Backup ble lagt på zip-diskett, hvor det ble lagret under en katalog som ble merket med aktuelle dagens dato for å ha kontroll på versjonene. Dette ble gjort ved hjelp av cron som er en automatiseringsfunksjon på Linux plattformen. Vi satte opp et system som tok backup én gang i timen av alle kode-filene, samt fullstendig backup av alt som er produsert én gang i døgnet. Alle nye backup-jobber startet med å lage en katalog med aktuell dato og tidspunkt. Se shellscript som utfører disse backup-jobbene i vedlegg 2.

#### 6.1.3 Konfigurasjonsstyring

Alt som produseres ble lagt på vår egen server. Her ble det opprettet et eget “releaseområde”, som var åpent for alle tilknyttet HiG’s nettverk og ble brukt til testing. Vi har i tillegg operert med et “utviklingsområde” som vi jobbet direkte mot, dette er kun åpent for medlemmene av prosjektgruppa. All dokumentasjon, referater og andre dokumenter som blir produsert under prosjektets gang, ble lagt på et eget område på samme server.

På vårt grupperom hadde vi ordnet oss med en arbeidslogg som hvert gruppelem førte underveis. I denne førte man opp tidsforbruk på hovedprosjektarbeid samt endringer og versjons oppdateringer.

#### 6.1.4 Problemrapporing og tiltak

Eventuelle problemer skulle rapporteres på morgenmøter og statusmøter i gruppa, og problemstillinger skulle bli tatt opp med veileder dersom vi ikke klarte å løse problemet internt i prosjektgruppa. Det var da ønskelig fra prosjektgruppas side at veileder skulle gi råd til prosjektgruppa om videre arbeid innen prosjektet. Prosjektgruppa har hatt ukentlige møter med veileder hver fredag kl.12.30, dersom ikke annet var avtalt. Prosjektleder avtalte med veileder om disse møtene. Korte referat fra disse møtene er vedlegg til denne rapporten. Prosjektgruppa

avholdt evalueringsmøter rett etter møtene med veileder. Videre fremdrift og fordeling av arbeidsoppgaver ble diskutert på disse møtene.

### **6.1.5 Kodekontroll**

Vi delte arbeidsoppgavene mellom oss, og vi prøvde å tilstrebe at alle gruppe-medlemmene måtte godkjenne all kode for å unngå kvalitetsforskjeller og misforståelser underveis i prosjektarbeidet.

### **6.1.6 Ressursbruk**

Vi hadde fått tildelt grupperom i B-bygget, i 3. etasje rom nummer B348. Der avtalte vi å møtes hver morgen kl 09.00, avvik fra dette skulle meldes til de andre i prosjektgruppa. Vi jobbet i fellesskap til forelesningene begynte. Arbeidsmengde og tidsrom for hovedprosjektet ble delt inn i to perioder; én for tiden før påske og én for etter påske. Dette ble gjort på grunn av at vi også hadde obligatoriske fag og valgfag med undervisning og eksamen før påske. Etter påsken hadde vi mer tid tilgjengelig til bruk i hovedprosjektsammenheng. Vi estimerte tidsforbruk per uke før påske til 18 timer, etter påske 40 timer. Arbeidsbelastningen etter påske ble mye større enn hva vi estimerte. Det er totalt gjennomført 1300 timer på prosjektet, fordelt på de tre medlemmene i gruppa.

## **6.2 Kvalitetssikring av produktet og testing.**

Etter hvert som deler av systemet ble ferdigutviklet, ble dette testet av hovedprosjektdeltagerne. Dette ble gjort for å oppdage feil og mangler med tanke på funksjonalitet. Oppdragsgiver konsentrerte seg om brukervennlighet og grafisk brukergrensesnitt. Veileder etterlyste også innspill fra den tidligere referansegruppa på layouten til løsningen, men ingen ønsker ble rapportert tilbake. Alle valg som avvek fra den opprinnelige kravspesifikasjonen vår, ble diskutert i gruppa og i plenum med veileder.

## **6.3 Kvalitetssikring av prosess**

Kommunikasjon med oppdragsgiver og veileder ble opprettholdt løpende. Alle avtalte møter ble avholdt og rapporter ble innlevert innenfor de tidsrammene vi hadde definert i planleggingsfasen. En mer uformell og fortløpende evaluering av arbeidsmåter, innstillinger og arbeidsoppgaveoppnåelse ble ivarettatt av prosjektleder. Møtereferat og statusrapporter er lagret som møtereferat dd.mm.yyyy og statusrapporter dd.mm.yyyy. Møtereferatene er lagt ved som vedlegg til hovedprosjektrapporten.

Vi har i prosjektperioden levert tre statusrapporter; 15.02., 30.03. og 10.05. Rapportene ble levert veileder, de to første i papirformat og den siste i form av vedlegg til en mail. Rapportene ga en generell vurdering av hovedprosjektarbeidet; slik som eventuelle avvik i kvalitet og fremdrift i forhold til aktuelle milepæler, årsakssammenhenger og hvilke tiltak prosjektgruppa ville initiere i forhold til dette. Statusrapportene var handlingsorienterte og beskrev i korthet hvordan prosjektarbeidet generelt gikk på gitte tidspunkt. Disse ble gode indikatorer på hvordan vi tidsmessig lå an i forhold til skissert tidsplan. Gjennom disse statusrapportene kunne også avvik og tiltak i forhold til kvalitet og fremdrift skisseres dersom vi ønsket det. Metoder for å møte spesielle utfordringer i fortsettelsen kunne også bli skissert her. Gantt-skjemaet ble brukt aktivt for å evaluere fremdrift og status i gruppa.

## **6.4 Evaluering av kvalitetssikringsmal**

Punktene under 6.1, 6.2 og 6.3 gav oss en ganske god ramme å jobbe innenfor, og dette har for vår gruppe fungert tilfredsstillende nok. Vi skulle kanskje hatt utviklet en plan for hvordan vi skulle håndtere uforutsette hendelser; en eventuell "kriseplan". Vi stolte kanskje blindt på at ingen graverende og prosjekttruende situasjoner ville eller kunne oppstå.

Vi hadde utarbeidet en plan for hvilken rekkefølge vi skulle utvikle og implementere modulene, og denne ble fulgt tilnærmet. Korte referat av arbeidsoppgaver under utvikling ble i form av mail rapportert til veileder under prosjektets siste tre uker.



# Kapittel 7

## Oppsummering

### 7.1 Løsningen

Vi er kommet frem til et system som ved full utvidelse kan tilfredsstillere oppdragsgivers ønsker og behov for en nettbasert testgenerator. I funksjonalitet og grafisk brukergrensesnitt er løsningen blitt optimal etter kravspesifikasjonen og oppdragsgivers ønsker. Vi har lagt vekt på at brukeren ikke behøver å være datakyndig for å skjønne fremgangsmåten ved å aksessere systemet. Systemet vi har utviklet er etter vår mening både brukervennlig og fiffig i sin utforming.

Utfordringen til lærerne som brukergruppe, blir å stille spørsmålene riktig i forhold til spørsmålstypene vi har kategorisert. En nettbasert løsning for læring som innehar slike muligheter som denne ville være revolusjonerende for mange utdanningsinstitusjoner. Vi mener vi har laget et produkt som, ved full utvidelse, ville være meget interessant for disse.

Vi synes vi har benyttet en teknologi som var helt korrekt for løsningens krav, imidlertid ble det mye jobbing for å oppnå de detaljspesifikke kravene som kom frem muntlig med oppdragsgiver. Vi føler at det grafiske brukergrensesnittet er blitt bra, men mener at det hadde vært bedre og ikke minst enklere å benytte en mer tradisjonell oppbygging og design for nettløsninger. Selve kodingen gikk meget bra og raskt. Vi har allikevel satt oss inn i mange nye metoder både i PHP og MySQL. Disse har begge meget god dokumentasjon, og det finnes et kjempenettverk av mennesker ute i verden som er villige til å dele kode, besvare spørsmål og diskutere innenfor denne type programmering.

Det har vært greit å jobbe med PHP og MySQL, men vi fant ut at det var komplisert å lage en løsning som ser bra ut i både Netscape Navigator 4.7 og Internet Explorer 5.0. Dette til dels for at de takler HTML-kode noe annerledes, men mest fordi de takler DHTML forskjellig. Det er på sin plass at det blir utviklet en standard som kan støttes av samtlige nettleserne, slik at man slipper å utelukke, eller lage forskjellige versjoner for hver av disse. Netscape har dessuten valgt å bruke andre navn på de samme funksjonene som finnes i Internet Explorer 5.0. De eneste som taper på at det finnes to standarder er brukerne. Lærermodulen er utviklet med støtte i Internet Explorer 5.0 eller høyere. Studentmodulen og superbrukeren kan i tillegg brukes med Netscape Navigator 4.7 eller Opera.

### 7.2 Erfaringshøsting

#### 7.2.1 Måloppnåelse

Vi rakk ikke å utvikle løsningen i sin helhet slik den er beskrevet i kravspesifikasjonen. Den største kritiske faktoren var tiden. Vi brukte for lang tid før vi kom frem til et grafisk brukergrensesnitt som tilfredstilte oppdragsgivers krav til layout. Kravspesifikasjonen fra den forrige hovedprosjektgruppa ble omgjort en del, enkelte deler ble skrevet helt om, noe ble mer spissformulert og vi prøvde å synliggjøre hva systemet skulle dekke på en mer utfyllende måte. Resultatet ble at løsningen innehar og skal inneholde en del funksjonalitet som det viste seg var tidkrevende å utvikle. Vår tidsbruk på de forskjellige modulene ble langt større enn hva vi på forhånd i planleggingsfasen kunne forutse. Det ble nedlagt mye tid på justeringer av skjermbilder og de forskjellige sidenes oppbygging. Det var også store krav om layout og visualitet slik at løsningen ble mest

mulig lik standard Windows applikasjoner. Vi er svært fornøyde med det grafiske brukergrensesnittet, tatt i betraktning at dette skal kjøres i en nettleser.

En løsning som støttet alle nettleser gikk vi bort fra og konsentrerte arbeidet med å utvikle en systemløsning med støtte i Microsoft Internet Explorer 5.0. Dette forenklet vårt utviklingsarbeid i en viss grad med tanke på kravene til grafisk brukergrensesnitt som ble gitt i forbindelse med utviklingen av lærermodulen.

Alle spørsmålstyper skissert i kravspesifikasjonen ble det ikke tid til å utvikle. I samråd med veileder og oppdragsgiver kom vi til enighet og gjensidig forståelse om hvilke spørsmålstyper vi skulle gi høyest prioritet i utviklingen. Enkelte deler av de forskjellige modulene ble også luket ut som mindre viktige, og utelatt. Dette gjelder lærerens mulighet til å kopiere an annen lærers test, eller å koble testen sin til en annen. Vi måtte også utelukke utviklingen av flere spørsmålstyper enn type A. Oppdragsgiver og veileder har ikke brukt tid på å teste løsningens funksjonalitet. Vi har selv stått for denne testingen, og det er avdekket en del mindre feil.

### 7.2.2 Hva er gjort bra

Vi er fornøyd med design og layout. Vi har klart å lage lærermodulen så lik Windows-applikasjoner som mulig. På den måten er brukervennligheten ivaretatt ovenfor de som kjenner Windows-miljøet. Vi er også fornøyde med implementasjon av CSS, og måten de forskjellige sidene i de forskjellige modulene er bygget opp. Dette gjør det særdeles enkelt å utvide løsningen senere.

### 7.2.3 Utvidelsesområder for en totalløsning

I kravspesifikasjonen har vi definert seks spørsmålstyper. Det ble et for omfattende arbeid for TVIL-prosjektgruppa å utvikle alle spørsmålstypene. Vi har utviklet den første spørsmålstypene helt etter kravspesifikasjonen. Denne er betegnet som spørsmålstype A. For at systemet skal kunne inneha den funksjonalitet oppdragsgiver ønsket av systemet, er det påkrevd at systemet videreutvikles for også å omfatte de siste fem spørsmålstypene. Da vil en totalløsning av systemet etter vår mening, være et godt dekkende system for testing av studenter. Systemet kan også utvides med enda flere spørsmålstyper dersom det skulle vise seg å være behov for det.

Det er mulig å utvikle flere statistikk funksjoner enn hva vi har lagt opp til i vår løsning. Vår løsning inneholder bare statistikk over antall tester i et valgt fag og antall studenter som har besvart en bestemt test eller ei. Videre bør superbruker se en del statistikk over ressursbruk av systemet. Dette vil kunne være antall fag med tester, antall tester under systemet og antall brukere blant lærere. Systemets brukere og da i særdeleshet lærere og superbruker, vil være ideelle innspillpartnere angående nye ønskede statistikktyper.

Ved bruk av systemet vil brukergruppa avdekke eventuelle behov for ytterligere flere statistiske funksjoner. Vi har laget en løsning som bare støtter Microsoft Internet Explorer 5.0 som nettleser for lærermodulen og vi forutsetter dermed at Microsoft Internet Explorer 5.0 opprettholder sin popularitet som nettleser blant lærere som brukergruppe. Lærerne bør kunne ha mulighet til å endre sitt passord ved bruk av systemet. Når superbruker registrerer lærere som skal ha tilgang til systemet legger han også inn brukernavn og passord for disse. Vi mener systemet bør tillate læreren å endre dette selv, gjerne ved første gangs innlogging, eller når som helst senere. Superbrukerens mulighet til å bestemme hva slags input lærerne kan bruke må også lages. Det er beskrevet i kravspesifikasjonen at superbruker kan bestemme ulovlige tegn i Fagkode.

Det må også lages støtte for at superbrukeren setter institusjonens domene. Dette skal brukes til default-verdier i mailadresser. Vi har satt denne statisk i løsningen som @hig.no. Det bør også lages en påminnelses-funksjon som varsler superbrukeren om når det er på tide å slette/opprette ny klasser og lagre godkjentlister til flate filer.

Løsningen kan gjøres om til andre språk. Dette kan for eksempel løses ved å lage en todimensjonal array indeksert på språkets navn. Arrayen kan bygges opp slik: `ordliste[engelsk][lykke til] = "Good luck";` og for tysk: `ordliste[tysk][lykke til]="Viel gluck";` osv. Når ordet skal hentes ut, sender vi med språkets navn og ordet på norsk. En print av `ordliste[engelsk][lykke til]` vi gi resultatet "Good luck". Alternativt kan man lage en egen tabell i databasen som omhandler språkvalg. Da kan superbruker legge inn flere språk uten hjelp av koding.

Det bør også lages en hjelpefunksjon som kan fremkalles ved de forskjellige valg-alternativene. Vi ser for oss at denne kunne vært laget som en todelt brukermanual for brukergruppene av systemet; lærerne og studentene. Brukermanualen kunne vært laget som et eget dokument og den burde beskrive mer utfyllende funksjonaliteten til systemet utover de forklaringene som kommer frem ved bruk av systemet. En god del utfyllende ordforklaringer til funksjonalitet er gitt i denne hovedprosjektrapporten, som vi forutsetter at avanserte brukere av

systemet vil sette seg inn i. En utvidelse av hovedprosjektet kunne være å lage en hjelpefunksjon som kunne tilkalles aktivt i systemet.

Systemet kan også benyttes til testing av andre brukergrupper. Eventuelle andre organisasjoner kan ha et behov for å teste sine medlemmer. Flere organisasjoner og bedrifter tilbyr opplæringsprogram og evalueringsfunksjoner til sine ansatte eller medlemmer. Disse organisasjonene og bedriftene kan også ha et behov for å teste kunnskapsnivået eller måloppnåelse til en bestemt brukergruppe. Vårt system kan også brukes til dette formålet. Funksjonaliteten til vårt systemet skulle være dekkende for et slik organisasjon eller bedrift's behov. Komponentene til systemet har et stort gjenbrukspotensiale, så svært mye av det vi har utviklet kan gjenbrukes slik det er eller ved å forandre litt på ordlyd eller benevnelser.

#### **7.2.4 Deler av løsningen med forandringspotensiale**

Lærermodule burde kanskje vært laget i et objektorientert språk, som for eksempel C++ eller Java. Hadde man valgt å lage denne modulen som en frittstående Windows-applikasjon i C++ ville man måtte oppdatere alle klientene ved revisjon av programmet. Dessuten ville læreren være avhengig av at programmet var installert på arbeidstasjonen han ville benytte for å bruke systemet. Men man kunne derimot ha laget denne modulen som en Java applet som lærerene aksesserer via nettleseren. Læreren trenger da bare tilgang til Internett for å bruke systemet. Kravene om layout og skjermbildenes oppbygging og funksjonalitet hadde i så tilfelle vært mye enklere å oppnå. Superbrukermodulen kunne også ha vært laget som en selvstendig Windows-applikasjon. Det er ikke like store krav til tilgjengelighet for denne modulen, og den er heller ikke like mye i bruk som resten av løsningen. Disse innstillingene trengs stort sett bare å endres et par ganger i året. Sikkerheten ville blitt betydelig større dersom den hadde vært en selvstendig applikasjon installert på superbrukers arbeidstasjon. Den nettbaserte løsningen vi har utviklet er riktignok passordbeskyttet, men passordet krypteres ikke. Man unngår fristelsen til innbrudd på systemet hvis døren ikke engang finnes...

#### **7.2.5 Hva har vi lært gjennom prosjektperioden**

Vi har lært en del om nettbasert læring og testing ved hjelp av Internett. For at verktøyet skal kunne bli brukt i den grad som oppdragsgiver ønsker, var det viktig at systemet hadde en helhetlig og gjennomgående enkel oppbygging. Vi har erfart at brukervennligheten har stor betydning, og brukerterskelen må legges lavt. Etter hvert som brukerne føler seg fortrolige med løsningen, kan de begynne å bruke de mer avanserte innstillingene. Det skal være lett å begynne å bruke systemet, men også tilfredstille de mer avanserte brukerne.

Vi har erfart at det tar lang tid å utvikle komplekse systemer, og da spesielt når man tar høyde for å utvikle en totalløsning. Vi har lært en god del PHP4, HTML, JavaScript og DHTML gjennom prosjektperioden. Vi har også lært mer om avansert spørring i SQL. Vi har blitt mer erfarne med tanke på drift og oppsett av Linux server og ikke minst Apache webserver. Kravspesifisering og analyse av eksisterende dokumentasjon er vi blitt mye flinkere til å gjøre. Vi er også blitt flinkere til å beskrive metoder og valg i et mer "norsk"-formulert og lettfattelig språk enn hva som er vanlig blant utviklingsrapporter innen data. Vi har høstet en del erfaringer vedrørende det å jobbe i en prosjektgruppe. Vi er en sammensatt gruppe med ulik erfaringsbakgrunn og ulik terskel på kunnskapsnivå. Ved prosjektstart fikk vi tildelt et prosjektrom som vi har brukt som arbeidsbase. Motivasjonen innad i gruppa har vært optimal for å oppnå best mulig resultat. For i det hele tatt å komme i mål med vår deløsning av oppgaven, valgte vi tidlig å fordele oppgavene innad i gruppa slik at vi fikk nyttegjort våre ressurser på best mulig måte. Vi har hatt en prosjektfase uten konfliktsituasjoner. Vi har ikke tatt høyde for å utvikle en kriseplan i tilfelle store forandringer innen hovedprosjektets rammer og miljø. Vi har stolt på at vi ville kunne takle eventuelle krisesituasjoner innad i gruppa eller i samråd med veileder. Likevel er nok vår største erfaring å jobbe i prosjekt med utvikling av en så stor løsning. Vi har blitt flinke til å kommunisere, motivere, arbeide mot delmål og milepæler og fordele arbeidsoppgaver. Vi har dessuten blitt flinkere til å estimere tidsbruk.

### **7.3 Emne for nytt hovedprosjekt**

Arbeidet som gjenstår med å få løsningen ferdig blir av mer funksjonsmessig art, og det er en forutsetning at man tilegner seg kjennskap til PHP, MySQL, HTML og JavaScript. Det vil være minimalt med arbeid når det gjelder design og layout, og man kan begynne å kode tidlig i prosjektfasen. Kravspesifikasjonen er ferdig utviklet og databasen er ferdig modellert. Det er få elementer som skal utvikles for å nå de funksjonsmessige

kravene, og oppgaven vil i hovedsak omhandle utvikling av de siste fem spørsmålstypene. Vi ønsker en eventuell ny prosjektgruppe lykke til.

## 7.4 Konklusjon

Vi har kommet frem til en løsning som kan tas i bruk slik den foreligger. Den er også et meget godt grunnlag for videreutvikling mot en totalløsning slik den er beskrevet i kravspesifikasjonen. Det ble dessverre ikke tid til å gjøre løsningen ferdig, men oppdragsgiver har gitt meget gode tilbakemeldinger på det som er utført. Det ble nedlagt mye arbeid i spesifikke detaljer, spesielt når det gjelder design og layout. Vi har konsentrert oss om å få høy kvalitet på det som er gjennomført, fremfor en mindre god komplett løsning. Vi føler at dette har vært en korrekt måte å jobbe på. Det har vært mye moro og utrolig lærerikt. Temaet er meget interessant og vi vil nok se at trenden med nettbaserte læringsvektøy vil øke i tiden fremover.

# Kapittel 8

## Verktøy

Vi har benyttet to hovedkategorier verktøy;

I            Vi har benyttet verktøy som kun er nødvendig for utviklingen av systemet

II           Vi har verktøy som systemet er avhengig av for å kjøre.

Se listing av disse i de to følgende tabeller:

### 8.1 Software systemet er avhengig av

Verktøy	Beskrivelse
Apache/el.l.	Systemet må kjøres på en webtjener, helst av typen Apache. Systemet kan installeres på andre typer webtjenere også, men systemet er utviklet og testet på en Apache webtjener, så det er anbefalt å bruke en slik.
PHP 4	PHP kreves for at systemet skal fungere. Hovedtyngden i systemet er laget ved hjelp av PHP script, og det kreves PHP versjon 4 eller høyere.
MySQL	Systemet er avhengig av en MySQL databaseserver som må være tilgjengelig fra webtjeneren.

## 8.2 Verktøy vi har brukt under utviklingen av systemet

Verktøy	Beskrivelse
Dreamweaver 4	WYSIWYG webeditor vi brukte for å lage layout. Dette verktøyet brukte vi for å spare tid på HTML-koding.
Edit +/Emacs	Teksteditor som har støtte for fargekoding for alle programmeringsspråk/scriptspråk vi skulle bruke.
Photoshop/Image Ready	Bildebehandlingsprogram vi brukte i forbindelse med all bildebehandling.
MS-Project	Gantt-diagram ble laget ved hjelp av dette verktøyet.
LyX	Dette er en avansert open source dokument prosessor. LyX bruker L <sup>A</sup> T <sub>E</sub> X, som er en open source typesettingsmotor, i bakgrunnen. Kjører best på Unix plattform. Hele prosjektrapporten er skrevet i LyX.
Java SDK 1.3.0	Javapakken brukte vi for å kompilere javaprogrammer/appleter som vi må bruke på enkelte av de mest avanserte spørsmålstypene. Vi må ta i bruk appleter der vanlig HTML og PHP ikke kan løse problemstillingen.
Samba	Samba er en daemon som vi kjører på vår Linux Redhat 6.2 tjener for å få tilgang til denne også fra våre Windows 2000/95 klienter.
Cron	Automatiseringstjeneste som ble kjørt på vår Linux tjener. Denne tjenesten tar seg av backup. Feil blir varslet vha. mail.
Modelator	Verktøy for modellering av databaser. Utviklet av forfatteren av læreboka i Databaser I.
InternetExplorer	Nettleser fra Microsoft (MS).
Netscape Navigator	Nettleser fra Netscape.
Opera	Norskutviklet Nettleser.

# Kapittel 9

## Litteraturliste

### 9.1 Bøker og manualer

“Software Engineering” (Fifth Edition) \_  
Addison-Wesley  
Ian Sommerville  
ISBN 0-201-42765-6

“Prosjektarbeid, utviklings og endringskompetanse”  
Universitetsforlaget  
Harald Westhagen  
Per A. Johannesen  
Ole Faafeng  
Erik Røine  
Eline Wesmann  
ISBN 82-00-22459-7

“Netts@mfunn”  
Tano Aschehoug  
Kristin Braa  
Per Hetland  
Gunnar Liestøl  
ISBN 82-518-3938-6

“Principles of Network and System Administration”  
Wiley  
Mark Burgess  
ISBN 0-471-82303-1

“Drift av flerbrukersystemer” kompendium  
HiG  
Jon Thingvold

### 9.2 Internettressurser

MySQL: [http://www.mysql.com/Manual\\_chapter/manual\\_toc.html](http://www.mysql.com/Manual_chapter/manual_toc.html)

Klient og serverside-programmering: [http://higweb.hig.no/at/data/client\\_server/index.phtml](http://higweb.hig.no/at/data/client_server/index.phtml)

LyX: <http://www.lyx.org>

PHP: <http://www.php.net>  
<http://www.phpbuilder.com/>

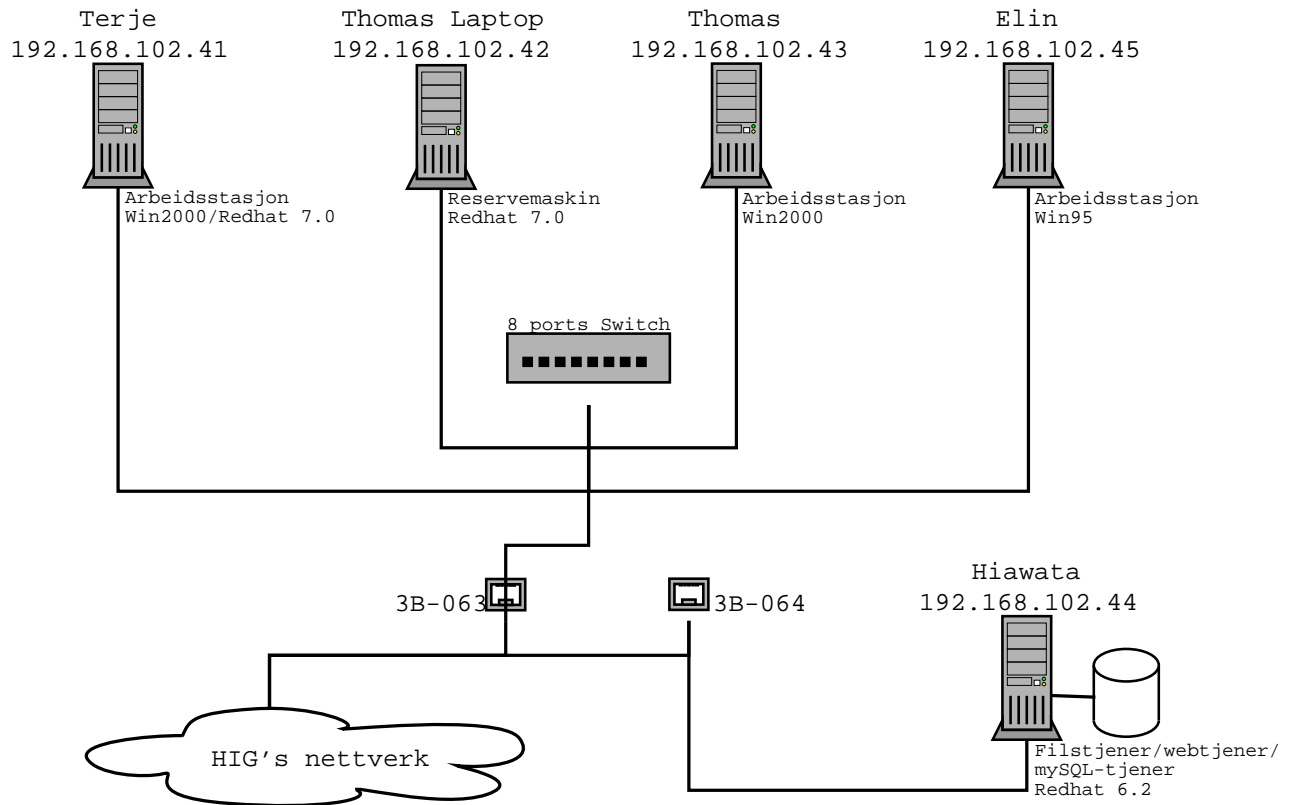
Generelt: <http://www.w3schools.com>  
<http://javascript.internet.com/forms/menu-swapper.html>  
<http://www.nextel.no/om/leksikon/>



# Kapittel 10

## Vedlegg

### 10.1 Vedlegg 1, Nettverksstruktur



Slik har vi valgt å sette opp våre PC'er i nett. På denne måten får alle tilgang til vår felles filserver, samt Internett.

## 10.2 Vedlegg 2, Shellscript for backup

Vi satte opp backup-jobber som kjøres daglig ved hjelp av cron. Dette gjorde vi ved å lage to shellscript som tar seg av selve backupen, samt at vi måtte konfigurere cron til å starte disse scriptene til riktige tider.

Dette scriptet kjører daglig og tar backup av hele prosjektområdet med dokumenter, kode og alt som er produsert til zip-diskett.

Filnavn: /etc/cron.tvil/daglig

---

```
#!/bin/bash
dato='/bin/date'
cp /home/tvil/ "/mnt/zipdrive/daglig/$dato" -rf
sync
```

---

Dette scriptet kjøres en gang i timen og tar backup av all kode som er produsert:

Filnavn: /etc/cron.tvil/time

---

```
#!/bin/bash
dato='/bin/date'
cp /home/tvil/public_html/ "/mnt/zipdrive/time/$dato" -rf
sync
```

---

Oppføringer i /etc/crontab

---

```
05 * * * * root /etc/cron.tvil/time
15 0 * * * root /etc/cron.tvil/daglig
```

---

## 10.3 Vedlegg 3, Første Gantt-skjema, Fremdriftsplan

I og med at vi jobber ut ifra kravspesifikasjonen som ble laget i 2000, ble en av våre viktigste aktiviteter å sette oss grundig inn i denne.

Vi har valgt å jobbe i en inkrementell systemutviklingsmodell; vi gjorde oss ferdig med en funksjon eller “modul” før vi begynte å utvikle en ny. Likevel ble modulene på et vis utviklet parallelt, siden vi praktiserte arbeidsfordeling med funksjoner eller moduler mellom oss. Implementering og testing av modulene skjedde skrittvis, dette er i tråd med inkrementell systemutvikling. Skissering av rekkefølgen av de forskjellige modulene fremkommer i Gantt-skjemaet (vedlegg 3).

### 10.3.1 Liste over aktiviteter

- Forprosjekt
- Etablere utviklingsmiljø
- Analyse og revisjon av eksisterende kravspesifikasjon
- Utvikle mal for og grovdesign grafisk brukergrensesnitt
- Utvikle og opprette database
- Fordele arbeidsoppgaver innen modulutvikling
- Koding og testing av moduler
- Rapportskrivning
- Testing på endelig versjon
- Korrekturlesing, trykking av hovedprosjektrapport

- Øving fremføring
- Fremføring hovedprosjekt
- Siste feilrettinger av systemet

### **10.3.2 Milepæler**

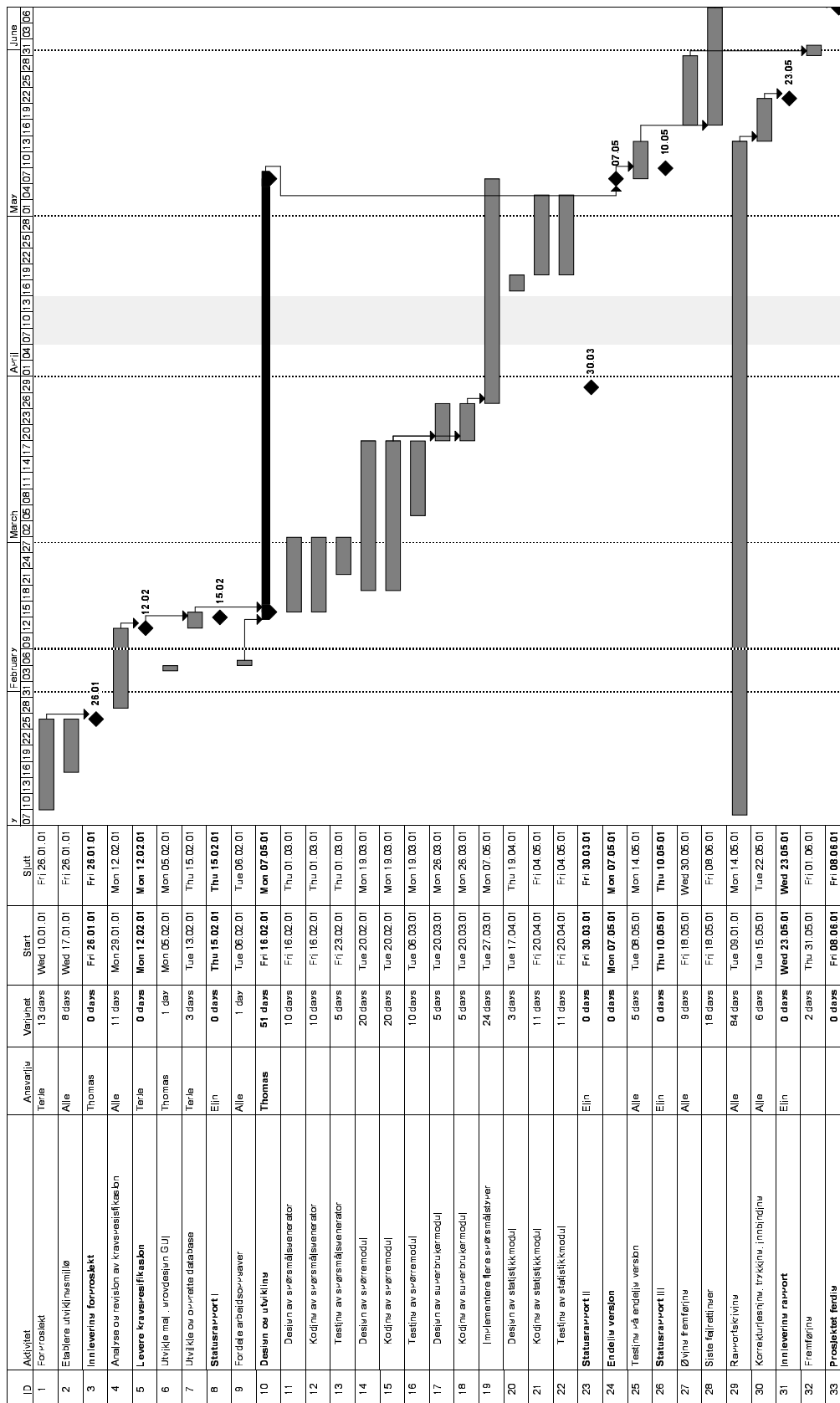
- Innlevering forprosjekt
- Levere utbedret kravspesifikasjon
- Statusrapport 1
- Statusrapport 2
- Statusrapport 3
- Innlevering hovedprosjektrapport
- Endelig versjon av systemet
- Hovedprosjektet ferdig

### **10.3.3 Beslutningspunkter**

Prosjektet ble igangsatt etter at forprosjektrapporten med framdriftsplan ble levert til veileder 26.01.2001. Dersom det dukket opp beslutningspunkter underveis i prosjektet, skulle disse avgjøres primært av prosjektgruppa i gruppemøter, sekundært i møte med oppdragsgiver. Dette avhang selvfølgelig av beslutningspunktets art. Dersom det oppsto store avvik eller betydelige forsinkelser med deler av prosjektet, skulle dette tas opp og avgjøres i samråd med veileder hva som skulle gjøres for å komme best mulig ut av denne situasjonen. De enkelte milepælene i vår framdriftsplan ble også våre beslutningspunkter.

### **10.3.4 Tids- og ressursplaner**

Vi har jobbet med tids- og ressursplaner og fått samlet dette i et Gantt-skjema. Se figur under.



Vi feilestimerte tiden en del under forprosjektet når vi satte opp fremdriftsplan. Utviklingen av lærermodulen spesielt brukte vi lenger tid på enn vi på forhånd estimerte. Se forøvrig på logg i vedlegg 6, hvor det fremgår

hvordan fremdriften i realiteten ble. Dette var pga flere faktorer som for oss var mer eller mindre ukjente da vi satte opp fremdriftsplanen:

- Systemutvikling II prosjekt som var tidkrevende.
- En del mer jobb med forberedelse av eksamen i Nettverksbasert multimedia enn beregnet. Dette var en praktisk gruppeeksamen som krevde at alle gruppemedlemmene var med på øvinger og forberedelser til eksamen.
- Et omfattende prosjekt i Klient og Serversideprogrammering på ett vekttall som telte 50% på sluttkarakteren i det faget.
- Oppgaven var mer omfattende og kompleks enn det vi på forhånd hadde forutsett.
- Vi måtte lære flere nye teknikker enn det vi på forhånd hadde forutsett. Dette gjelder spesielt JavaScripting og DHTML som måtte brukes for å få til ønsket funksjonalitet på lærermodulen.
- Vi hadde liten eller ingen erfaring med tidsestimering fra liknende prosjekter og generelt fra før. Programmeringsprosjekter er vanskelige å tidsestimere.

Dette gjorde at vi fikk liten tid til å jobbe på studentmodulen, men vi gjorde vårt beste for å få denne mest mulig brukbar slik at oppdragsgiver kunne ta i bruk systemet. Prosjektrapportarbeidet jobbet vi med underveis i hele prosjektperioden, med spesielt mye jobbing med denne på slutten. Da vi så at prosjektrapporten kom til å bli tilfredsstillende fortsatte vi på finpussen som oppdragsgiver ville at vi skulle gjøre.

## 10.4 Vedlegg 4, SQL-sript som oppretter databasen

```
# Databasescript for oppretting av tabeller for systemet TVIL v2.0
# Alle fag registreres i denne tabellen

# Tabellstruktur for tabell 'innstillinger'
# Innstillinger for systemet
CREATE TABLE innstillinger (
  brukernavn varchar(16) DEFAULT '' NOT NULL, # Superbrukerens brukernavn
  passord varchar(16) DEFAULT '' NOT NULL, # Superbrukerens passord
  navn varchar(64) DEFAULT '' NOT NULL, # Superbrukerens navn
  epost varchar(128) DEFAULT '' NOT NULL, # Superbrukerens epostadresse
  dato_hoest date, # Dato på høsten hvor superbruker skal få visse påminnelser
  dato_vaar date, # Dato på våren hvor superbruker skal få visse påminnelser
  epost_sletting date, # Dato for påminnelse på epost om sletting av klasser
  maksstud_test tinyint(4), # Maksimalt antall studenter som kan delta på en test
  bokstaver tinyint(4), # Lovlig med bokstaver i fagkoder. [0 | 1]
  tall tinyint(4), # Lovlig med tall i fagkoder. [0 | 1]
  mellomrom tinyint(4), # Lovlig med "space" i fagkoder. [0 | 1]
  punktum tinyint(4), # Lovlig med . punktum i fagkoder. [0 | 1]
  bindestrek tinyint(4), # Lovlig med bindestrek i fagkoder. [0 | 1]
  andre_tegn text, # Andre lovlige tegn i fagkode.
  PRIMARY KEY (brukernavn) # Definerer primærnøkkel.
);

# Tabellstruktur for tabell 'laerer'
# Lærere registreres i denne tabellen:
CREATE TABLE laerer (
  brukernavn varchar(64) DEFAULT '' NOT NULL, # Lærerens brukernavn
  passord varchar(16) DEFAULT '' NOT NULL, # Lærerens passord
  navn varchar(64) DEFAULT '' NOT NULL, # Lærerens virkelige navn
  epost varchar(255), # Lærerens epostadresse
  PRIMARY KEY (brukernavn)
);

# Tabellstruktur for tabell 'laerer_fag'
# Tabell for å gi andre lærere rettigheter til å editere andres fag
CREATE TABLE laerer_fag (
  brukernavn varchar(64) DEFAULT '' NOT NULL, # Brukernavn for lærer som skal få rettigheter
  fagkode varchar(16) DEFAULT '' NOT NULL, # Fagkoden læreren skal ha rettigheter til å endre
  PRIMARY KEY (brukernavn,fagkode), # Definerer primærnøkkel
  KEY FK3_laerer_laerer_fag (brukernavn),
  KEY FK4_fag_laerer_fag (fagkode)
);

# Tabellstruktur for table 'fag'
```

```

CREATE TABLE fag (
  fagkode varchar(16) DEFAULT '' NOT NULL,      # fagkode, primærnøkkel for fag
  navn varchar(64),                             # Fagets navn
  brukernavn varchar(64) DEFAULT '' NOT NULL,   # Fremmednøkkel mot 'laerer', fagets eier
  hjemmeside varchar(255),                     # URL til fagets hjemmeside
  les_kopiere tinyint(4),                      # Skal faget kunne leses/kopieres av andre lærere [0 | 1]=[false | true]
  redigere tinyint(4),                         # Hvem andre kan redigere faget? [0 | 1 | 2]=[bare meg selv | alle | gruppe]
  PRIMARY KEY (fagkode),                       # Definerer primærnøkkel
  KEY FK2_laerer_fag (brukernavn)             # Indeks på fremmednøkkel for å optimalisere spørringer
);

# Tabellstruktur for tabell 'test'
# Data som registreres for hver test:
CREATE TABLE test (
  testID int(10) unsigned DEFAULT '0' NOT NULL auto_increment, # ID for testen, autoincrementeres
  navn varchar(128),                                           # Testens navn
  fagkode varchar(16) DEFAULT '' NOT NULL,                    # Fagkode for fag/kurs som testen tilhører
  oblig tinyint(4),                                           # Er testen en oblig, [0 | 1]
  forsok tinyint(4),                                          # Antall forsøk som kan brukes på testen
  fra date,                                                  # Dato testen er åpen fra
  til date,                                                  # Dato testen lukkes
  godkjentkrav tinyint(4),                                    # Godkjentkrav for testen i prosent riktig
  antspm tinyint(4) DEFAULT '0',                              # Antall spørsmål som skal være med i testen
  innledningstekst text,                                     # Testens innledningstekst
  gruppestørrelse_min tinyint(4),                           # Minimum gruppestørrelse på testen
  gruppestørrelse_maks tinyint(4),                          # Maksimum gruppestørrelse på testen
  tidsfrist_tall int(11),                                    # Tidsfrist tall for testen
  tidsfrist_enhet tinyint(4),                                # Enhet for tidsfrist, [0 | 1 | 2] [sekunder | minutter | timer]
  skjult_tidtagning tinyint(4),                              # Skal skjult tidtagning for statistikkens skyld benyttes, [0 | 1]
  kreves_godkjentID int(11),                                # ID for annen test som kreves godkjent før denne testen kan tas
  antall_forsok tinyint(4),                                  # Antall forsøk pr spørsmål i testen
  riktig_for_neste tinyint(4),                              # Må studenten svare riktig for å få neste spørsmål i testen? [0 | 1]
  karakterskala tinyint(4),                                 # Karakterskala som skal benyttes, [0 | 1] [hel/halv | tidels]
  vis_alle_spm tinyint(4),                                  # Skal alle spørsmål vises hvis testen ikke er oblig, [0 | 1]
  retting_fortlopende tinyint(4),                          # Fortløpende retting, studenten får retting etter hvert spørsmål, [0 | 1]
  redusere_poeng tinyint(4),                                # Standardverdi for poengreduksjon ved feil svar/hint, [0 | 1]
  fasit_til_slutt tinyint(4),                              # Skal fasit vises når testen er over? [0 | 1]
  poeng_etter_hvert tinyint(4),                            # Skal studenten få poengsum etter hvert spørsmål? [0 | 1]
  poeng int(11),                                           # Standardverdi for Poeng/vektning for hvert spørsmål i testen [0-10]
  ant_stud_svalt int(11) DEFAULT '0',                      # Antall studenter som har besvart testen
  koble_fag varchar(16),                                    # Koble til et annet fag
  koble_test int(10),                                      # Koble til en annen test
  PRIMARY KEY (testID),
  KEY FK5_fag_test (fagkode)
);

# Tabellstruktur for tabell 'test_filer'
# Filer som tilhører en test
CREATE TABLE test_filer (
  testID int(11) DEFAULT '0' NOT NULL,                # ID for testen filen tilhører
  filID bigint(20) DEFAULT '0' NOT NULL,              # ID for filen
  PRIMARY KEY (testID,filID),
  KEY FK9_fil_test_filer (filID),
  KEY FK10_test_test_filer (testID)
);

# Tabellstruktur for tabell 'sporsmaal'
# Generelle data for et spørsmål:
CREATE TABLE sporsmaal (
  spmID bigint(20) unsigned DEFAULT '0' NOT NULL auto_increment, # Primærnøkkel, id for spørsmålet
  spm_nr smallint(5) unsigned DEFAULT '0' NOT NULL,             # Spørsmålsnummer
  spmtype tinyint(4) DEFAULT '0' NOT NULL,                     # Spørsmålstypen, [0 | 1 | 2 | 3 | 4 | 5 | 6 | 7], hvor 0-2 er flervalgstypene
  alltid_med tinyint(4) DEFAULT '0' NOT NULL,                  # Sier om spørsmålet alltid skal være med, [0 | 1]
  innledning text,                                           # Spørsmålsteksten
  testID int(11) DEFAULT '0' NOT NULL,                       # Fremmednøkkel mot test, sier hvilken test spørsmålet tilhører
  poeng int(11),                                             # Antall poeng/vekt på spørsmålet
  miste_poeng int(11),                                       # Miste poeng ved feil svar eller hint, [0 | 1]
  melding_riktig text,                                       # Tekstlig melding ved feil svar på spørsmålet
  filID_riktig bigint(20),                                   # Fil som skal vises ved riktig svar på spørsmålet
  autovis_riktig tinyint(4),                                 # Autovisning av fil ved riktig svar på spørsmålet. [0 | 1]
  std_melding_riktig tinyint(4),                            # Skal standardmelding vises ved riktig svar på spørsmålet. [0 | 1]
  melding_feil text,                                        # Tekstlig melding ved galt svar på spørsmålet
  filID_feil bigint(20),                                    # Fil som skal vises ved galt svar på spørsmålet.
  autovis_feil tinyint(4),                                  # Autovisning av fil ved galt svar på spørsmålet. [0 | 1]
  std_melding_feil tinyint(4),                              # Skal standardmelding vises ved galt svar på spørsmålet. [0 | 1]
  se_riktige tinyint(4),                                    # Skal riktige svar vises for spørsmålet, [0 | 1]
  PRIMARY KEY (spmID),
  KEY FK12_test_sporsmaal (testID)
);

# Tabellstruktur for tabell 'hint'
# Tabell med eventuelle hint for et spørsmål

```

```

CREATE TABLE hint (
    spmID bigint(20) DEFAULT '0' NOT NULL,          # Fremmednøkkel for sporsmaal hintet tilhører
    hint_nr tinyint(1) DEFAULT '0' NOT NULL,       # Hint nummer
    hint_tekst text,                                # Hintets tekst
    filID bigint(20),                               # Fil kan legges til for å utdype hintet
    autovis_fil tinyint(4),                         # Autovisning av fil for hint. [0 | 1]
    PRIMARY KEY (spmID, hint_nr)
);

# Tabellstruktur for tabell 'student'
# Data for student:
CREATE TABLE student (
    brukernavn varchar(16) DEFAULT '' NOT NULL,    # Studentens brukernavn
    fornavn varchar(64) DEFAULT '' NOT NULL,      # Studentens fornavn
    etternavn varchar(128) DEFAULT '' NOT NULL,   # Studentens etternavn
    passord varchar(16),                           # Studentens passord for pålogging
    navn varchar(64) DEFAULT '' NOT NULL,        # Klassen studenten tilhører
    epost varchar(64),                             # Studentens epostadresse
    PRIMARY KEY (brukernavn),
    KEY FK8_klasse_student (navn)
);

# Tabellstruktur for tabell 'fil'
CREATE TABLE fil (
    filID bigint(20) unsigned DEFAULT '0' NOT NULL auto_increment, # Primærnøkkel for fil, autoincrementeres
    brukernavn varchar(64) DEFAULT '' NOT NULL,    # Brukernavn for filens eier, (laerer)
    type tinyint(4) DEFAULT '0' NOT NULL,         # Filtypen, [0,1,2,3,4] = [bilde, audio, video, dokument, program]
    navn varchar(128) DEFAULT '' NOT NULL,        # Filens navn
    data longblob NOT NULL,                       # Filens binære data
    mimetype varchar(100) DEFAULT '' NOT NULL,    # Filens mimetype
    PRIMARY KEY (filID),                          # Definerer primærnøkkel
    KEY FK1_laerer_fil (brukernavn)
);

# Tabellstruktur for tabell 'klasse'
# Tabell med alle klasser som kan bruke systemet:
CREATE TABLE klasse (
    navn varchar(64) DEFAULT '' NOT NULL,         # Klassens navn
    PRIMARY KEY (navn)
);

# Tabellstruktur for tabell 'klasse_fag'
# Tabell som kobler klasser til fag:
CREATE TABLE klasse_fag (
    fagkode varchar(16) DEFAULT '' NOT NULL,     # Fakode
    navn varchar(64) DEFAULT '' NOT NULL,        # Klassenavn som kan ta faget
    PRIMARY KEY (fagkode, navn),                 # Definerer primærnøkkel
    KEY FK6_klasse_klasse_fag (navn),
    KEY FK7_fag_klasse_fag (fagkode)
);

# Tabellstruktur for tabell 'spm_fil'
# Tilegner filer til et spørsmål
CREATE TABLE spm_fil (
    filID bigint(20) DEFAULT '0' NOT NULL,       # Fremmednøkkel mot fil
    spmID bigint(20) DEFAULT '0' NOT NULL,       # Fremmednøkkel mot sporsmaal
    autovis tinyint(4) DEFAULT '0' NOT NULL,     # Gir om filen skal autovises eller ikke. [0 | 1]
    KEY FK11_fil_spm_fil (filID),
    KEY FK13_sporsmaal_spm_fil (spmID)
);

# Tabellstruktur for tabell 'test_generelt'
# Generelle kommentarer til forskjellige resultatintervaller
CREATE TABLE test_generelt (
    brukernavn varchar(64) DEFAULT '' NOT NULL,  # Brukernavn for lærer som har kommentert
    91_100 text,                                 # Kommentar ved 91-100% riktig
    81_90 text,                                  # Kommentar ved 81-90% riktig
    71_80 text,                                  # Kommentar ved 71-80% riktig
    61_70 text,                                  # Kommentar ved 61-70% riktig
    51_60 text,                                  # Kommentar ved 51-60% riktig
    41_50 text,                                  # Kommentar ved 41-50% riktig
    31_40 text,                                  # Kommentar ved 31-40% riktig
    21_30 text,                                  # Kommentar ved 21-30% riktig
    11_20 text,                                  # Kommentar ved 11-20% riktig
    0_10 text,                                    # Kommentar ved 0-10% riktig
    PRIMARY KEY (brukernavn)
);

# Tabellstruktur for tabell 'test_karakterer'
# Spesielle kommentarer til forskjellige resultatintervaller for en spesiell test:
CREATE TABLE test_karakterer (
    testID int(11) DEFAULT '0' NOT NULL,         # ID for testen kommentarene gjelder

```

```

91_100 text,                # Kommentar ved 91-100% riktig
81_90 text,                 # Kommentar ved 81-90% riktig
71_80 text,                 # Kommentar ved 71-80% riktig
61_70 text,                 # Kommentar ved 61-70% riktig
51_60 text,                 # Kommentar ved 51-60% riktig
41_50 text,                 # Kommentar ved 41-50% riktig
31_40 text,                 # Kommentar ved 31-40% riktig
21_30 text,                 # Kommentar ved 21-30% riktig
11_20 text,                 # Kommentar ved 11-20% riktig
0_10 text,                  # Kommentar ved 0-10% riktig
PRIMARY KEY (testID)
);

# Tabellstruktur for tabell 'spmttype_forklaring'
# Forklaring for de forskjellige spørsmålstypene.
CREATE TABLE spmttype_forklaring (
    spmttype tinyint(10) DEFAULT '0' NOT NULL,    # ID for spørsmålstype
    forklaring text,                               # Forklarende tekst for spørsmålstypen
    navn text NOT NULL
);

# Tabellstruktur for tabell 'test_resultat'
# Testresultater legges inn i denne tabellen:
CREATE TABLE test_resultat (
    testID int(11) DEFAULT '0' NOT NULL,          # ID for resultat
    brukernavn varchar(16) DEFAULT '' NOT NULL,    # Studenten som har fått aktuelt resultat sitt brukernavn
    godkjent tinyint(4) DEFAULT '0' NOT NULL,      # Er testen godkjent eller ikke, [0 | 1]
    tid float(10,2) DEFAULT '0.00' NOT NULL,       # Tidsforbruk på testen
    ant_forsok tinyint(4) DEFAULT '1' NOT NULL,    # Antall forsøk studenten har brukt på testen
    KEY FK15_student_test_resulta (brukernavn),
    KEY FK16_test_test_resultat (testID)
);

# Tabellstruktur for tabell 'flervalg'
# Her lagres spørsmålstype avhengige data for spørsmålstypen flervalg:
CREATE TABLE flervalg (
    spmID bigint(20) DEFAULT '0' NOT NULL,        # Fremmednøkkel mot sporsmaal
    alt_nr tinyint(4) DEFAULT '0' NOT NULL,       # Alternativ nummer
    alt_tekst text,                               # Alternativets tekst
    riktig tinyint(4) DEFAULT '0' NOT NULL,      # Bool som sier om dette alternativet er riktig eller ikke. [0 | 1]
    alt_filID bigint(20),                          # Fremmednøkkel mot tabellen fil, tilegner fil til alternativet
    alt_fil_auto tinyint(4),                       # Gir om filen skal autovises eller ikke. [0 | 1]
    tlbmelding text,                              # Tilbakemeldingstekst for alternativet
    tlbmeldingfilID bigint(20),                   # Fil som tilhører tilbakemeldingen for alternativet.
    tlbmeldingfil_auto tinyint(4),               # Gir om tilbakemeldingsfilen skal autovises eller ikke. [0 | 1]
    PRIMARY KEY (spmID,alt_nr)                   # Definerer primærnøkkel
);

# Tabellstruktur for tabell 'flervalg_svar'
# Svardata for flervalgsspørsmål:
CREATE TABLE flervalg_svar (
    spmID bigint(20) DEFAULT '0' NOT NULL,        # ID for spørsmål svaret tilhører
    alt_nr tinyint(4) NOT NULL,                   # Alternativ nr svaret tilhører
    brukernavn varchar(16) NOT NULL,             # Brukernavn for student som har avgitt svar
    svar tinyint(4),                              # Svar som er avgitt på alternativet
    PRIMARY KEY (spmID, alt_nr, brukernavn)
);

# Oppretter standardinnstillinger for systemet, det som er viktig her er superbrukerens brukernavn og passord,
# passord bør endres straks etter at systemet er installert:
INSERT INTO innstillinger
(brukernavn, passord, navn, epost,
dato_hoest, dato_vaar, epost_sletting, maksstud_test,
bokstaver, tall, mellomrom, punktum,
bindestrek, andre_tegn)
VALUES
('superuser', 'superpwd', 'Fornavn Etternavn', 'super@hig.no',
'0000-04-15', '0000-11-15', '0000-05-30', 10,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1);

```

## 10.5 Vedlegg 5, Databaserapport

Denne databaserapporten finnes i HTML-versjon på <http://www.tvil.net/databaserapport.html>. HTML-versjonen er mer oversiktlig og lettelleselig enn den rene tekstversjonen som er lagt med som vedlegg her.

```

Innholdsrapport:
Modell: z:\database\er-modell.mr4
Datatyper: C:\Program Files\MetodeData\Modelator\DType\mySQL.ddf

```



Opprettet 2/5/2001, rapportert 1:27:14 PM 5/20/2001

#### Entitetstypeliste

```
laerer  laerer
        Tabell med lærerens data

        _brukernavn_  Id      Ndv   VARCHAR(64)  VARCHAR(64)
        lærerens brukernavn for pålogging

        passord       Ndv   VARCHAR(16)   VARCHAR(16)
        passord for pålogging

        navn          Ndv   VARCHAR(64)   VARCHAR(64)
        virkelig navn

        epost         VARCHAR(255)  VARCHAR(255)
        epostadressen

fil      fil
        Tabell med lærerens filer. Dette kan være bilder, lyd, video,
        dokumenter og programmer

        _filID_ Id      Ndv   BIGINT  BIGINT
        *brukernavn  Ndv   VARCHAR(64)  VARCHAR(64)
        Fremmednøkkel mot tabellen "laerer"

        type         Ndv   TINYINT  TINYINT
        filtype, [0,1,2,3, 4] = [bilde, lyd, video, dokument, program]

        navn         Ndv   VARCHAR(128)  VARCHAR(128)
        filnavnet

        data         Ndv   LONGBLOB     LONGBLOB
        filens binære data

        mimetype     Ndv   VARCHAR(100)  VARCHAR(100)
        Filens mime-type

fag      fag
        _fagkode_   Id      Ndv   VARCHAR(16)  VARCHAR(16)
        fagkode

        navn         VARCHAR(64)   VARCHAR(64)
        fagets navn

        *brukernavn  Ndv   VARCHAR(64)   VARCHAR(64)
        fremmednøkkel mot lærer, fagets eier

        hjemmeside  VARCHAR(255)  VARCHAR(255)
        url til fagets hjemmeside

        les_kopiere  TINYINT  TINYINT
        skal faget kunne leses/kopieres av andre lærere [0 | 1]=[false | true]

        redigere     TINYINT  TINYINT
        Hvem andre kan redigere faget? [0 | 1 | 2]=[bare meg selv | alle | gruppe]

laerer_fag  laerer_fag
        Kobler lærer mot fag dersom de har rettigheter til å redigere

tester

        *brukernavn_  Id      Ndv   VARCHAR(64)  VARCHAR(64)
        fremmednøkkel mot laerertabell

        *fagkode_    Id      Ndv   VARCHAR(16)  VARCHAR(16)
        fremmednøkkel mot fag

test       test
        Testenes data

        _testID_    Id      Ndv   INT          INT
        Primærnøkkel for testen

        navn         VARCHAR(128)  VARCHAR(128)
        fagets navn

        *fagkode     Ndv   VARCHAR(16)   VARCHAR(16)
        fagkoden

        oblig       TINYINT  TINYINT
```

```

er testen obligatorisk? [0 | 1]

forsoek          TINYINT TINYINT
antall forsøk studenten har på denne testen

fra              DATE    DATE
startdato for testen

til              DATE    DATE
sluttdato for testen

godkjentkrav     TINYINT TINYINT
Prosentandel riktig for godkjent [0-100]

antspm           TINYINT TINYINT
antall spørsmål i aktuell testen.

innledningstekst TEXT    TEXT
innledningstekst

gruppestorrelse_min TINYINT TINYINT
minimum gruppestørrelse på testen

gruppestorrelse_maks TINYINT TINYINT
maksimum gruppestørrelse

tidsfrist_tall    INT     INT
antall tidsenheter studenten har til rådighet for å besvare testen

tidsfrist_enhet   TINYINT TINYINT
enhet for tidsfrist, [0 | 1 | 2]=[sekuner | minutter | timer]

skjult_tidtagning TINYINT TINYINT
er tidtagningen skjult (kun for statistikk), [0 | 1]=[false | true]

kreves_godkjentID INT     INT
fremmednøkkel mot test som må være godkjent før aktuell test kan tas.

antall_forsok    TINYINT TINYINT
antall forsøk per spørsmål

riktig_for_neste TINYINT TINYINT
Må studenten svare riktig for å komme videre til neste spørsmål? [0 | 1]=[nei | ja]

karakterskala    TINYINT TINYINT
hel/halv eller tidelskarakterskala. [0 | 1] = [hel/halv | tidel]

vis_alle_spm     TINYINT TINYINT
Skal alle spørsmål vises hvis ikke obligatorisk test?

retting_fortlopende TINYINT TINYINT
Læreren kan velge om studenten skal få retting etter hvert spørsmål. [0 | 1]=[nei | ja]

redusere_poeng   TINYINT TINYINT
Poengredusering ved feil svar

fasit_til_slutt  TINYINT TINYINT
Skal fasit for testen vises til slutt? [0 | 1]

poeng_etter_hvert TINYINT TINYINT
Skal poengberegninger vises etterhvert? [0 | 1]

poeng            INT     INT
Standard vektning for hvert spørsmål i testen.

ant_stud_svalt   INT     INT
Antall studenter som har tatt testen.

koble_fag        VARCHAR(16) VARCHAR(16)
Fagkode som dette faget skal kobles til.

koble_test       INT     INT
ID for test som denne testen skal kobles til.

test_karakterer test_karakterer
Kommentar til studentene ut fra opnådd resultat på en test.
Denne overstyrer test_generelt hvis den finnes.

*_testID_        Id     Ndv    INT    INT
fremmednøkkel mot test

91_100           TEXT    TEXT

```

Kommentar ved 91-100% riktig

81_90	TEXT	TEXT
Kommentar ved 81-90% riktig		
71_80	TEXT	TEXT
Kommentar ved 71-80% riktig		
61_70	TEXT	TEXT
Kommentar ved 61-70% riktig		
51_60	TEXT	TEXT
Kommentar ved 51-60% riktig		
41_50	TEXT	TEXT
Kommentar ved 41-50% riktig		
31_40	TEXT	TEXT
Kommentar ved 31-40% riktig		
21_30	TEXT	TEXT
Kommentar ved 21-30% riktig		
11_20	TEXT	TEXT
Kommentar ved 11-20% riktig		
0_10	TEXT	TEXT
Kommentar ved 0-10% riktig		

test\_generelt test\_generelt  
 Generelle kommentarer studentene skal få etter utført test  
 avhengig av resultatet.

*brukeravn_	Id	Ndv	VARCHAR(64)	VARCHAR(64)
91_100		TEXT	TEXT	
Kommentar ved 91-100% riktig på en test				
81_90		TEXT	TEXT	
Kommentar ved 81-90% riktig på en test				
71_80		TEXT	TEXT	
Kommentar ved 71-80% riktig på en test				
61_70		TEXT	TEXT	
Kommentar ved 61-70% riktig på en test				
51_60		TEXT	TEXT	
Kommentar ved 51-60% riktig på en test				
41_50		TEXT	TEXT	
Kommentar ved 41-50% riktig på en test				
31_40		TEXT	TEXT	
Kommentar ved 31-40% riktig på en test				
21_30		TEXT	TEXT	
Kommentar ved 21-30% riktig på en test				
11_20		TEXT	TEXT	
Kommentar ved 11-20% riktig på en test				
0_10		TEXT	TEXT	
Kommentar ved 0-10% riktig på en test				

klasse klasse  
 Tabell som inneholder alle klasser som er tilgjengelige, superuser har kontroll over denne.

_navn_	Id	Ndv	VARCHAR(64)	VARCHAR(64)
Klassens navn, primærnøkkel				

klasse\_fag klasse\_fag  
 \*\_fagkode\_ Id Ndv VARCHAR(16) VARCHAR(16)  
 \*\_navn\_ Id Ndv VARCHAR(64) VARCHAR(64)

test\_resultat test\_resultat  
 Tabell med alle forsøk på løsning av tester.

*testID	Ndv	INT	INT
Fremmednøkkel mot test			
*brukeravn	Ndv	VARCHAR(16)	VARCHAR(16)
Fremmednøkkel mot student			

```

godkjent          Ndv    TINYINT TINYINT
Sier om aktuell student har fått godkjent denne testen.

tid              Ndv    FLOAT   FLOAT
Forteller hvor lang tid aktuell student brukte på dette forsøket av testen

innstillinger    innstillinger
Tabell med systeminnstillinger

brukernavn      Id     Ndv    VARCHAR(16)  VARCHAR(16)
Superbrukerens brukernavn for pålogging på superbrukermodul

passord         Ndv    VARCHAR(16)  VARCHAR(16)
Superbrukerens passord for pålogging på superbrukermodul

navn            Ndv    VARCHAR(64)   VARCHAR(64)
Superbrukerens virkelige navn

epost           Ndv    VARCHAR(128)  VARCHAR(128)
Superbrukerens epostadresse

dato_hoest      DATE   DATE
Dato på høsten hvor superbruker skal få visse påminnelser

dato_vaar       DATE   DATE
Dato på våren hvor superbruker skal få visse påminnelser

epost_sletting DATE   DATE
Dato for påminnelse på epost om sletting av klasser

maksstud_test  TINYINT TINYINT
Maks antall studenter som kan ta en test samlet

bokstaver      TINYINT TINYINT
Lovlig med bokstaver i fagkode

tall            TINYINT TINYINT
Lovlig med tall i fagkode [0 | 1]

mellomrom      TINYINT TINYINT
Lovlig med mellomrom i fagkode

punktum        TINYINT TINYINT
Lovlig med punktum i fagkode

bindestrek     TINYINT TINYINT
Lovlig med bindestrek i fagkode

andre_tegn     TEXT    TEXT
Andre lovlige tegn i fagkoder

sporsmaal      sporsmaal
Tabell med felles data for spørsmål

spmID_Id       Ndv    BIGINT  BIGINT
primærnøkkel for spørsmål

spm_nr         Ndv    SMALLINT SMALLINT
Spørsmål nummer i testen

*spmtype       Ndv    TINYINT TINYINT
spørsmålstypen, [0 | 1 | 2 | 3 | 4 | 5 | 6 | 7]=[ 1 av 2 | 1 av mange | n av m | fyll inn tall/ord | finn sekvens | kombinere
| plassere på bilde | peke på areal i bilde ]

alltid_med     Ndv    TINYINT TINYINT
Skal dette spørsmålet være med uansett hvis test inneholder tilfeldig spm? [0 | 1]

innledning     TEXT    TEXT
Innledningstekst for spørsmålet.

*testID        Ndv    INT     INT
Fremmednøkkel mot test

poeng          INT     INT
Maks antall poeng for dette spørsmålet

miste_poeng    INT     INT
Antall poeng som tapes hvis feil svar

melding_riktig TEXT    TEXT
Melding ved riktig svar

```

```

filID_riktig          BIGINT  BIGINT
Fil ved riktig svar

autovis_riktig       TINYINT TINYINT
Autovisning av fil ved riktig svar

std_melding_riktig   TINYINT TINYINT
Skal standard tilbakemelding gis ved riktig svar? [0 | 1]

melding_feil         TEXT    TEXT
Melding ved feil svar

filID_feil           BIGINT  BIGINT
Fil ved feil svar

autovis_feil         TINYINT TINYINT
Autovisning av fil ved feil svar

std_melding_feil     TINYINT TINYINT
Skal standard tilbakemelding gis ved feil svar? [0 | 1]

se_riktige           TINYINT TINYINT
Vise riktig svar på spørsmålet? [0 | 1]

test_filer          test_filer
*_testID_          Id      Ndv      INT      INT
fremmednøkkel mot test

*_filID_           Id      Ndv      BIGINT  BIGINT
fremmednøkkel mot fil

spm_fil             spm_fil
*_filID            Ndv      BIGINT  BIGINT
*_spmID            Ndv      BIGINT  BIGINT

student             student
Data om hver enkelt student som benytter seg av systemet

_brukernavn_       Id      Ndv      VARCHAR(16)  VARCHAR(16)
brukernavn for senere pålogging

fornavn            Ndv      VARCHAR(64)  VARCHAR(64)
fornavnet

etternavn          Ndv      VARCHAR(128)  VARCHAR(128)
etternavnet

passord            VARCHAR(16)  VARCHAR(16)
passord for senere pålogging

*navn              Ndv      VARCHAR(64)  VARCHAR(64)
fremmednøkkel mot tabellen klasse

epost              VARCHAR(64)  VARCHAR(64)
Studentens epostadresse

hint                hint
*_spmID_           Id      Ndv      BIGINT  BIGINT
Fremmednøkkel mot spørsmålstabell

_hint_nr_          Id      Ndv      TINYINT TINYINT
Hint nr, lovlige verdier: [1..3]

hint_tekst         TEXT    TEXT
Tekstlig hint som skal hjelpe studenten med aktuelt spørsmål.

*_filID            BIGINT  BIGINT
En fil som skal hjelpe studenten med aktuelt spørsmål.

autovis_fil        TINYINT TINYINT
Skal filen som hører til hintet autovises? [0 | 1]

flervalg           flervalg
*_spmID_           Id      Ndv      BIGINT  BIGINT
Fremmednøkkel mot sporsmals tabell

_alt_nr_           Id      Ndv      TINYINT TINYINT
Aktuelt spørsmåls alternativ nummer

alt_tekst          TEXT    TEXT
Alternativets tekst

riktig             Ndv      TINYINT TINYINT

```

```

Bool som sier om dette alternativet er riktig eller ikke. [0 | 1]

alt_filID          BIGINT BIGINT
Fremmednøkkel til alternativets tilhørende fil

alt_fil_auto      TINYINT TINYINT
Sier om fila automatisk skal vises hvis mulig. [0 | 1]

tlbmelding        TEXT TEXT
Tilbakemelding til studenten ved feil svar på dette alternativet

tlbmeldingfilID   BIGINT BIGINT
Fil som tilbakemelding ved feil svar på dette alternativet

tlbmeldingfil_auto TINYINT TINYINT
Skal tilbakemeldingsfila autovises? [0 | 1]

spmtyp_forklaring spmtyp_forklaring
Navn og forklaring for de forskjellige spørsmålstypene legges
her. Denne teksten skal forklare brukeren hvordan de
forskjellige spørsmålstypene skal løses.

  _spmtyp_   Id   Ndv   TINYINT TINYINT
Primærnøkkel

  navn       TEXT TEXT
Spørsmålstypens navn.

  forklaring TEXT TEXT
Forklaring for spørsmålstypen.

flervalg_svar flervalg_svar
Tabell med svardata for hvert flervalgsspørsmål for en student.

*_spmID_   Id   Ndv   BIGINT BIGINT
Fremmednøkkel mot flervalg/sporsmaal tabellen

*_alt_nr_   Id   Ndv   TINYINT TINYINT
Fremmednøkkel mot flervalg tabellen

*_brukernavn_ Id   Ndv   VARCHAR(16) VARCHAR(16)
brukernavnet for student som har avgitt svar.

  svar       TINYINT TINYINT
Studentens svar på dette alternativet [0 | 1]

```

#### Relasjonstypeliste

```

laerer_fil
  laerer tilhører 0..m fil
  fil har 1..1 laerer
  _brukernavn_ *brukernavn

laerer_fag
  laerer tilhører 0..m fag
  fag underviser i/har 1..1 laerer
  _brukernavn_ *brukernavn

laerer_laerer_fag
  laerer 0..m laerer_fag
  laerer_fag 1..1 laerer
  _brukernavn_ *brukernavn_

fag_laerer_fag
  fag 0..m laerer_fag
  laerer_fag 1..1 fag
  _fagkode_ *fagkode_

fag_test
  fag tilhører 0..m test
  test inneholder 1..1 fag
  _fagkode_ *fagkode

klasse_klasse_fag
  klasse undervises for 0..m klasse_fag
  klasse_fag 1..1 klasse
  _navn_ *navn_

fag_klasse_fag
  fag deltar i 0..m klasse_fag
  klasse_fag 1..1 fag
  _fagkode_ *fagkode_

klasse_student
  klasse går i 0..m student
  student består av 1..1 klasse
  _navn_ *navn

fil_test_filer

```

```

        fil          0..m    test_filer
        test_filer   1..1    fil
        _filID_     *_filID_
test_test_filer
        test         0..m    test_filer
        test_filer   1..1    test
        _testID_     *_testID_
fil_spm_fil
        fil          0..m    spm_fil
        spm_fil      1..1    fil
        _filID_     *_filID_
test_sporsmaal
        test         tilhører 1..m    sporsmaal
        sporsmaal    består av 1..1    test
        _testID_     *_testID_
sporsmaal_spm_fil
        sporsmaal    0..m    spm_fil
        spm_fil      1..1    sporsmaal
        _spmID_     *_spmID_
test_test_karakterer
        test         tilhører 1..1    test_karakterer
        test_karakterer har 1..1    test
        _testID_     *_testID_
student_test_resultat
        student      tilhører 0..m    test_resultat
        test_resultat har fått 1..1    student
        _brukernavn_ *_brukernavn_
test_test_resultat
        test         tilhører 0..m    test_resultat
        test_resultat har gitt 1..1    test
        _testID_     *_testID_
laerer_test_generelt
        laerer       tilhører 1..1    test_generelt
        test_generelt har satt 1..1    laerer
        _brukernavn_ *_brukernavn_
sporsmaal_hint
        sporsmaal    tilhører 0..3    hint
        hint         gir      1..1    sporsmaal
        _spmID_     *_spmID_
fil_hint
        fil          har      0..m    hint
        hint         uttrykker 0..1    fil
        _filID_     *_filID_
sporsmaal_flervalg
        sporsmaal    0..m    flervalg
        flervalg     1..1    sporsmaal
        _spmID_     *_spmID_
spmtypetype_forklaring_sporsmaal
        spmtypetype_forklaring har en 0..m    sporsmaal
        sporsmaal    forklarer 1..1    spmtypetype_forklaring
        _spmtypetype_ *_spmtypetype_
flervalg_flervalg_svar
        flervalg     tilhører 1..1    flervalg_svar
        flervalg_svar har et 1..1    flervalg
        *_spmID_     *_spmID_
        _alt_nr_     *_alt_nr_
student_flervalg_svar
        student      er besvart av 1..1    flervalg_svar
        flervalg_svar har avgitt 1..1    student
        _brukernavn_ *_brukernavn_

```

## 10.6 Logg

Dato: Ant. h: Aktivitet: Ting i parentes er prosjekter/eksamener i andre fag

Før jul:10		Forprosjektarbeid, research
12.01	0,5	Møte med Frode
17.01	3	Forprosjektarbeid
18.01	6	Forprosjekt
19.01	3,5	Møte med Frode + forprosjekt
20.01	3	Serverkonfigurasjon/installasjon, webserver, mysqlserver
21.01	5	Forprosjekt, serverkonfigurasjon webserver, mysqlserver, backup
22.01	2	Fremdriftsplan
23.01	3	Backuprutiner, fremdriftsplan

24.01	4	Skriving av forprosjektrapport, ganttskjema, plakat
25.01	3	Forprosjektrapport, trykking av plakat
26.01	4	Siste finpuss forprosjektrapport + møte med Frode
29.01	5	Gjennomgang kravspek
30.01	3	Gjennomgang kravspek
31.01	4	Møte med frode, gjennomgang kravspek
01.02	6	Justering kravspesifikasjon
02.02	5	Kravspesifikasjon, webside (www.tvil.net), plakat
05.02	6	Mer retting på kravspesifikasjon
06.02	2	Kravspesifikasjon
07.02	3	Kravspekgjennomgang, databasedesign
08.02	5	Databasedesign, download Linux, installasjon på arbeidsstasjon for rapportskrivning, kravspesifikasjonsarbeid
09.02	3	Møte med Frode, kravspesifikasjon, databasedesign
11.02	2	Kravspesifikasjonsarbeid
12.02	3	Databasedesign, GUI grovdesign
13.02	3	Ferdigstilling kravspesifikasjon, printing
14.02	2	Planlegging, fordeling av arbeidsoppgaver.
15.02	4	Databasedesign, design/oppretting av hovedside for systemet
16.02	2	Databasedesign, mer jobbing med hovedside for systemet
19.02	3	Databasedesign, hovedside for systemet
20.02	3	Databasedesign, opprettet databasen
21.02	2	Hoveddesign for lærermodulen (tabs), innloggingsfunksjon
22.02	3	Hoveddesign for lærermodulen (tabs), innloggingsfunksjon
23.02	3	Møte med Frode, design lærermodulen, innloggingsfunksjon og testing av denne
26.02	2	Design lærermodulen
27.02	3	Design lærermodulen
28.02	3	grovdesign av fag/kurs (lærer)
01.03	5	grovdesign av fag/kurs
02.03	3	Møte med Frode, grovdesign av fag/kurs
05.03	2	grovdesign av tester
06.03	2	grovdesign av tester
07.03	3	grovdesign av tester
08.03	5	Studering av flervalgsspørsmål, grovdesign
09.03	2	Møte med Frode, Studering av flervalgsspørsmål, grovdesign
12.03	0	(System Utvikling II Prosjekt)
13.03	0	(System Utvikling II Prosjekt)
14.03	0	(System Utvikling II Prosjekt)
15.03	2	(System Utvikling II Prosjekt)
16.03	0	(Innlevering av SU prosjekt)
19.03	3	Javascripting for nytt/edit fag
20.03	2	Javascripting for nytt/edit fag
21.03	0	(Klient server prosjekt)
22.03	0	(Klient server prosjekt)
23.03	0	Møte med Frode (Klient server prosjekt)
26.03	0	(Klient server prosjekt, eksamenslesing)
27.03	0	(Klient server prosjekt, eksamenslesing)
28.03	0	(Klient server prosjekt, eksamenslesing)
29.03	1	(Klient server prosjekt, eksamenslesing)
30.03	0,5	(Innlevering klientserver)
02.04	0	<b>EKSAMEN I KLIENT OG SERVERSIDEPROGRAMMERING</b>
03.04	2	Møte med Frode, Fag/kurs på lærermodulen
04.04	0	(Eksamenslesing)
05.04	0	(Eksamenslesing)
06.04	0	(EKSAMEN I SYSTEMUTVIKLING II)



07-17.04	>- P - Å - S - K - E - - - F - E - R - I - E -<
17.04	6 Fag/Kurs på lærermodulen
18.04	(EKSAMEN I NETTVERKSBASERT MULTIMEDIA)
19.04	8 Nytt editere fag/kurs, slette fag/kurs
20.04	7 Møte med Frode, Fag/kurs, tester
23.04	7 Tester, spørsmål på lærermodulen
24.04	7,5 Ny/editere test, spørsmål på lærermodulen
25.04	6,5 Ny/editere test, slette test, flervalgsspørsmål, slette spørsmål
26.04	7 Retting på fag/kurs og tester etter gjennomgang med Frode, flervalgsspørsmål, tester avanserte innstillinger
27.04	8 Møte med Frode, Flervalgsspørsmål, tilordne fil til alternativer, spørsmålstekst og tilbakemeldinger.
28.04	6,5 Flervalgsspørsmål, tilordne tilbakemelding med fil til alternativer.
30.04	12 Design nytt flervalgsspørsmål/tester avanserte innstillinger/rapportskriving
01.05	7,5 Design nytt flervalgsspørsmål/tester avanserte innstillinger/rapportskriving
02.05	7 Design nytt flervalgsspørsmål/tester avanserte innstillinger/rapportskriving
03.05	9 Design nytt flervalgsspørsmål/tester avanserte innstillinger/rapportskriving
04.05	11 Møte med Frode, Funksjonalitet nytt flervalgsspørsmål/tester avanserte innstillinger/rapportskriving
05.05	10 Funksjonalitet nytt flervalgsspørsmål/studentmodul, innlogging, design/rapportskriving
07.05	7,5 Nytt flervalgsspørsmål/studentmodul/rapportskriving
08.05	9 Nytt flervalgsspørsmål/studentmodul/rapportskriving
09.05	8 Nytt flervalgsspørsmål/superbrukermodul/rapportskriving
10.05	9,5 Installasjon av arbeidsstasjon etter disskrasj, editering av flervalgsspørsmål.
11.05	8 Editering av flervalgsspørsmål, studentmodul
12.05	6,5 Editering av flervalgsspørsmål, studentmodul
13.05	2 Klargjøring/opplasting loke.hig.no
14.05	14 Siste finpuss editering av flervalgsspørsmål, påbegynt studentenes spørsmålspresentatør.
15.05	12 Studentenes spørsmålspresentatør./Superbrukermodul/Rapportskriving, hent_fil.php som henter filer fra basen
16.05	9 Møte med Frode, Studentenes spørsmålspresentatør., filvisning/Superbrukermodul/Rapportskriving
17.05	4 Studentenes spørsmålspresentatør., filvisning/Superbrukermodul/Rapportskriving
18.05	14 Jobbet med prosjektrapporten.
19.05	17 Jobbet med prosjektrapporten.
20.05	17 Jobbet med prosjektrapporten.
21.05	14 Gjennomgang prosjektrapport på tlf med Frode, Siste finpuss og korrekturlesing på prosjektrapporten. Superbrukermodul.
22.05	10 Trykking og innbinding av prosjektrapporten. Superbrukermodul, finpuss på spørsmålspresentatør.
Sum	431 timer

## 10.7 Statusrapporter

### 10.7.1 Statusrapport I

#### STATUSRAPPORT I

#### Status

#### Planlegging

#### Organisering av arbeidet

Så langt i prosjektfasen har vi jobbet samlet i gruppa med forprosjektrapporten, evaluering av eksisterende kravspesifikasjon og utarbeidelse av forbedret kravspesifikasjon. Vi følger stort sett den tidsramma som vi har

lagt i forprosjektrapporten. Fordeling av arbeidsoppgaver innen modulene har vi ikke gjennomført ennå, dette vil bli gjort i løpet av den nærmeste tiden.

## **Løsning og koding**

Grovdesignet av GUI er utarbeidet; etter testing fungerer dette like bra i Netscape Navigator 4.7 som i Internet Explorer 5.0. Databasen er stort sett gjenbruk av eksisterende database, med en del tilføyelser/endringer i forhold til påloggingsrutiner.

## **Rapportskriving**

Forprosjektrapporten er ferdig og godkjent av veileder. Kravspesifikasjonen ble levert veileder 14.02.2001. Den øvrige rapportskrivningen blir utført etter hvert som prosjektperioden skrider frem.

## **Avsluttede oppgaver**

Forprosjektet ble levert 26.01.2001 og godkjent av veileder 09.02.2001. Analyse og revisjon av kravspesifikasjon er utført. Vår kravspesifikasjon ble levert veileder 14.02.2001. Databasen er revidert og utvidet med påloggingsdata for studenter.

## **Problemer og muligheter**

På dette stadiet i utviklingen av en løsning, kan det være vanskelig å bestemme seg for hva vi skal starte med å utvikle. Elementene innen en modul og de forskjellige modulene er knyttet nært sammen, så mye av utviklingsarbeidet henger nært sammen. Vi har imidlertid definert utviklingsrekkefølgen av de forskjellige modulene i vår fremdriftsplan, og intensjonen vår er å holde denne i størst mulig grad. Det er ikke utenkelig at vi kan komme til å utvikle og implementere deler av en modul, for så å bestemme at vi går videre med neste utviklingsfase og tilhørende modul. Viktigheten med å utvikle de essensielle delene av modulene, vil stå i fokus for vårt design.

## **Evaluering av status**

Vi har satt oss godt inn i prosjektoppgavens problemområder og gjort en del forandringer i forhold til den eksisterende kravspesifikasjonen utarbeidet av den første TVIL-prosjektgruppa. Vi følger tidsrammene til løsning av oppgaver i henhold til Gantt- skjemaet i forprosjektrapporten. Slik vi ser det, befinner vi oss der vi burde være i prosjektfasen.

## **Oppgaver vi jobber med**

Vi utvikler et brukervennlig GUI som er både forklarende i sitt design, samtidig som den er enkel og ryddig i sin utforming. I dette utviklingsarbeidet blir det lagt vekt på en enhetlig utseende av web-siden, samtidig som at det til en hver tid skal være klart for brukeren "hvor" han befinner seg.

## **Tidsfrister**

Tidsfristene vi har definert i forprosjektet er stort sett overholdt .

## **Motivasjon innen gruppa**

Gruppa har arbeidet strukturert hittil i prosjektperioden. Tidsforbruket til prosjektarbeidet har vært moderat og i henhold til det vi i utgangspunktet hadde planlagt.

## **Veilederkontakt**

Våre møter med veileder har vært avklarende og hjulpet oss videre med prosjektarbeidet.

## 10.7.2 Statusrapport II

### STATUSRAPPORT II

#### Status

#### Planlegging

Vi ligger langt bak vår tidsplan skissert i Gantt-skjemaet i forprosjektrapporten. Vi ser at vi kanskje ikke blir ferdige med alle spørsmålstypene, men jobber med å komme best ut av de oppgavene vi fullfører. Løsningen vil lett kunne bli utvidet med flere spørsmålstyper på et senere tidspunkt. Vi anser det som viktigst på dette punktet i prosjektfasen, å gjennomføre de modulene vi klarer å fullføre, med best mulig løsning.

#### Organisering av arbeidet

Vi har fordelt arbeidsoppgaver slik at Elin har jobbet med kravspesifikasjon og Thomas og Terje har jobbet med design og utvikling av "Fag" under spørsmålsgenerator.

#### Løsning og koding

Vi har gått bort fra å lage en løsning som fungerer i Netscape Navigator 4.7 fordi dette medfører for mye ekstra arbeid i form av både koding og testing.

#### Rapportskriving

Vi dokumenterer hver enkelt del av systemet etter at den er ferdigstilt.

#### Avsluttede oppgaver

Databasen er ferdig utviklet, bortsett fra tabeller som tilhører de forskjellige spørsmålstypene. Disse vil vi legge til når vi kommer til design av de forskjellige spørsmålstypene.

#### Problemer og muligheter

Oppgaven er stor og kompleks, så utviklingen blir vanskelig. Vi jobber mot en forholdsvis stor database med mange tabeller, relasjoner og attributter. Kravspesifikasjonen fra i fjor er på mange punkt litt vanskelig å tolke og enkelte krav er kompliserte å implementere.

#### Evaluerings av status

Gantt-skjema fra forprosjektrapporten sliter vi med å følge pga større arbeidsmengde ifm. obligatorisk oppgave i SU II, prosjektoppgave i Klient og Serversideprogrammering og eksamensforberedelser. Dessuten vil utviklingen av spørsmålsgeneratoren ta lengre tid enn vi forutså og den eksisterende databasen måtte modifiseres en del.

#### Oppgaver vi jobber med

Jobber for tiden med spørsmålsgeneratoren, mer spesifisert registrering av Fag/Kurs som vi nesten er ferdig med og registrering av tester. Ellers jobber vi med forbedringer av kravspesifikasjonen spesielt ifm. de forskjellige spørsmålstypene.

#### Tidsfrister

Vi skulle vært ferdig med spørsmålsgenerator, spørremodul og superbrukermodulen så vi ligger et stykke bak skjema.

## **Motivasjon innen gruppa**

Gruppa har arbeidet strukturert hittil i prosjektperioden. Tidsforbruket til prosjektarbeidet har vært moderat men litt mindre enn vi i utgangspunktet hadde planlagt. Det har blitt litt mindre tid til prosjektarbeid pga andre obligatoriske oppgaver og eksamensforberedelser. I Gantt-skjema i forprosjektrapporten har vi lagt opp til fri i påsken men kommer til å jobbe noe allikevel.

## **Veilederkontakt**

Våre møter med veileder har vært avklarende og hjulpet oss videre med prosjektarbeidet.

### **10.7.3 Statusrapport III**

#### **STATUSRAPPORT III**

## **Planlegging**

Planlegger ikke så mye på dette tidspunktet i prosjektfasen.

## **Organisering av arbeidet**

Vi har fordelt arbeidsoppgaver slik at Elin har gått tilbake til å jobbe med hovedprosjektrapporten og Thomas og Terje har jobbet med design og utvikling av deløsningene.

## **Løsning og koding**

De siste deløsningene er lagt ut på release-området til TVIL. Vi prøver å optimalisere våre deløsninger med best mulig løsning både når det gjelder funksjonalitet. Kommenterer koden parallelt med utviklingen.

## **Rapportskriving**

Rapporten er under skriving, mangler en del på skjermbeskrivelser for de forskjellige deløsningene. Kodeeksempler vil bli tatt med etter hvert.

## **Avsluttede oppgaver**

Ingen oppgaver er helt ferdige siden forrige statusrapport.

## **Problemer og muligheter**

Alle deløsninger tar lengre tid å utvikle enn hva vi kunne forutse tidligere i planleggingsfasen. Tidsfaktoren er vårt mest kritiske punkt.

## **Oppgaver vi jobber med**

Jobber for tiden med spørsmålgeneratoren, mer spesifisert spørsmålstype A og studentpresentasjonsmodulen.

## **Tidsfrister**

Vi skulle vært ferdig med betaversjon av systemet 07.05.2001. Vi ligger langt etter vår tidsplan.

## Motivasjon innen gruppa

Motivasjonen innad i prosjektgruppa er det ingenting å utsette på. Etter hvert som delløsningene nå begynner å bli ferdige og vi ser proporsjonene på en totalløsning, er det litt synd at vi ikke får anledning til å fullføre hele prosjektoppgaven. Gruppa er samlet tilbake til prosjektrummet vårt; alle jobber sammen helt til ettermiddagen og noe arbeid blir tatt med hjem. Vi har den siste tiden jobbet ganske strukturert. Tidsforbruket til prosjektarbeidet har økt betraktelig siden siste statusrapport, dette er også i tråd med hva vi i utgangspunktet hadde planlagt.

## Veilederkontakt

Vi har alle deltatt på de planlagte oppfølgingsmøtene på fredagene. I tillegg har Terje og Thomas hatt uformelle møter med veileder etter hvert som delløsningene er blitt lagt ut på release-området. Kontakten mellom veileder og prosjektgruppe har naturligvis tatt seg endel opp i tiden etter påsken.

## 10.8 Møtereferater

### 10.8.1 Møtereferat 12.01.2001

Møte mellom prosjektgruppa og veileder

12.01.2001

**Deltagere: Terje, Thomas, Elin og Frode**

- PA: Frode
- PL: En av oss
- RG (Referansegruppe): Evt. senere. Frode sender mail om interesserte
- Forprosjektet leveres Frode.
- Gjenbruk av databasen.
- Skal det være baseuavhengighet? Forslag om å bruke MySQL.
- Gjøre en og en modul ferdig om gangen.
- Påloggingsrutiner: må bestemmes.
- Prosjektavtaleutkast utlevert. Standardkontrakt: 5 eksemplarer

---

Internt møte etter møtet med Frode

1. Thomas lager en grovskisse for brukergrensesnittet.
2. Ekstra heading med hva modulen "gjør" + filnavn.
3. Alle referat og annen dokumentasjon dateres med ddmmyyyy.

### 10.8.2 Møtereferat 19.01.2001

Møte mellom prosjektgruppa og veileder

19.01.2001

**Deltagere: Terje, Thomas, Elin og Frode**

- Dato for kravspek; opp til oss
- Eksisterende kravspek: Frode går gjennom den til mandag

- Møte med Frode onsdag kl.14.15 og antageligvis ut dagen. Til dette møtet må vi se nøye på kravspek'en. => Fredagsmøtet går ut.
- Test: Beskjed til bruker om test ferdig/lagt ut?
- Hva med nye årskull? Oppgradering av klasser?
- Kontrakter: 2 forskjellige. Prosjektavtale ordner vi i gruppa.
- Databasen: ikke alle felter er komplette.

### 10.8.3 Møtereferat 31.01.2001

Møte mellom prosjektgruppa og veileder

31.01.2001

**Deltagere: Terje, Thomas, Elin og Frode**

- Forprosjekt; bra. I teksten skal tall fra 1 til 10 skrives med bokstaver. Si litt mer om testmodulen. Brukergrensesnittet. Før på ansvarshavende + involverte deltagere i de forskjellige aktivitetene på Gantt-skjemaet.
- Lever gammel + ny versjon av forprosjektet til Frode
- Pkt.: 3.2.2.2: Lærer også nettbasert. Superbruker bruker også nettleser!
- Indeks.html =>AH/AT
- Se notat fra Frode
- Pkt.:1.3.1.6:Ytelse. Brukervennlighet.
- Holde superbrukermodul "litt" åpen for evt. Delphi.
- Lage sjekkboks for de mest vanlige/brukte tegn.
- Henvisning/advarsel ved sletting av fag, bør løsningen inneholde.

### 10.8.4 Møtereferat 09.02.2001

Møte mellom prosjektgruppa og veileder

09.02.2001

**Deltagere: Terje, Thomas, Elin og Frode**

- Levert nytt forprosjekt
- Databasen: Tabell -spørsmål må utvides noe
- Test generelt s.42 i eks. kravspek, utvidelse av s.44
- Liste over oppnådd score skal snus.
- Endring av pålogging: ikke under klasse.
- GUI; vi har foreløpig bare tabeller i GUI. Siden bygges opp ved hjelp av kombinasjoner av disse.

### 10.8.5 Møtereferat 23.02.2001

Møte mellom prosjektgruppa og veileder  
23.02.2001

**Deltagere: Terje, Thomas, Elin og Frode**

- Tilbakemelding kravspek: Litt mer om problemområde, testing og bruk av Internett. Bakgrunn til prosjektgruppa + fag vi har hatt.
- Kombinere <=> sette noe inn i et bilde; lærer må selv definere hva som skal settes inn.
- Opp til læreren å definere progressive hint.
- Etter 3 hint; kommer rett svar ( eksempelvis vha. pil) frem.
- Superbruker: Slette student + flytte studenter til annen klasse.
- Mulighet for at noen spørsmål blir tilfeldig valgt.
- URL: direkte valg fra fagets hjemmeside må være mulig
- Revider statistikk for student; mulig å se opplisting av sine fag + status på oblig'er.
- Pkt.:2.2.4.3.6; går ut.
- Spørsmålstypene A-F; beskrive disse nærmere
- Neste møte: fredag 02.03.2001 kl.1300.

### 10.8.6 Møtereferat 23.03.2001

Møte mellom prosjektgruppa og veileder  
23.03.2001

**Deltagere: Terje, Thomas, Elin og Frode**

- Gi beskjed om når Frode bør gå på release-området.
- Rapport: se Frode's skriv
- Netscape Navigator-løsning går vi bort fra
- "Innlogget som fro\_haug" =>bort
- Edit-boks for innskriving av epost-adresse, Epost til lærer: <Editerbar m/defaultverdier>; blå m/link
- Sesjon lukkes ved shutdown
- Logg ut lærer på "førstesida" som presenterer systemets sider
- Når en kommer til "Fag/kurs", automatisk markering av den første i opplistingen. Opplistingen blir sortert på fag.
- Antall statustester og obligatoriske tester skal komme på en linje
- Fagnavnene skal angis slik: Faget " ....."
- Justering av tabeller til fast posisjon. Tilstrebe lik avslutning også i høyre marg.
- Fjerne all "fleipetekst" som vi har lagt ut.
- Legge til i heading "Registrer nytt fag" (NYTT mangler)
- Kan leses og kopieres av: Alle ( som default)
- Litt mer blank plass foran "Kan editeres av: " og "kan leses/kopieres av: "

- Kvitteringsbilde etter å ha laget nytt/editert? Beslutning: ja, skal stå i 2-3 sekunder
- Mulighet til å endre fagnummer? Beslutning: bestemmes senere
- Undersøke endel cache-feil til neste gang
- Ved sletting skal default-verdi være "Nei", kun ved aktiv enter vil sletting skje.

### 10.8.7 Møtereferat 03.04.2001

Møte mellom prosjektgruppa og veileder  
03.04.2001

**Deltagere: Terje, Thomas, Elin og Frode**

- Komme i havn med løsning?
- Spørsmålstypene?
- Frode hadde skissert i hvilken rekkefølge ting utover bør utvikles.
- Enighet om god funksjonalitet på det som blir ferdig
- "Utlogging"; tas også med
- Større felt for å skrive inn fagnavn i; rammen utvides

### 10.8.8 Møtereferat 20.04.2001

Møte mellom prosjektgruppa og veileder  
20.04.2001

**Deltagere: Terje, Thomas, Elin og Frode**

- Fag; -utlogging. "ENDRE" gjøres om til "OPPDATER". Fjerne punktum etter fagnr. Link til fagets tester må være reell. Bør ha med små/mindre kvitteringsvinduer (som ved sletting)
- "Nytt": "Brukere" og "Alle Brukere"; bokser litt opp til høyre
- "Editere"; Cashing når egentlig er oppdatert
- Vi har nå 4 uker igjen av prosjektperioden. Dette vil si ca. 1 uke per del som gjenstår. Noe kan utvikles parallelt.
- Frode's skisse fra forrige møte, ble benyttet igjen. Et minimumskrav var at vi klarer å komme i havn med de 5 punktene med størst/høyest prioritering.

### 10.8.9 Møtereferat 27.04.2001

Møte mellom prosjektgruppa og veileder  
27.04.2001

**Deltagere: Terje og Frode**

- Frode etterspurte gruppas arbeidsinnsats og motivasjon siste tiden, vi kom frem til at tilbakemelding til Frode på arbeid som var gjort var for dårlig.
- Ble enige om korte statusrapporter per mail om arbeidsoppgaver som pågår for å bedre kontakten mellom gruppa og veileder
- Så på en del funksjonalitet på "ting" Terje har jobbet på; ble enige om hva som skulle rettes opp/ændres.
- Gjennomgikk hvordan nytt/edit flervalgsspørsmål skulle bygges opp.



### 10.8.10 Møtereftrat 04.05.2001

Møte mellom prosjektgruppa og veileder

04.05.2001

**Deltagere: Terje, Thomas, Elin og Frode**

- Kwitteringsvindu/-boks går ut av løsningen. Bruker bare feilmeldingsvindu som popper opp.
- Registrere nytt fag; Klasser + Alle klasser i overskrift; lages i fete typer
- Etter svaralternativer: Fjern "ordvalg", legg til "tilbakemelding" bak "fil"
- Vekting: Sett en defaultverdi. Skrollbar med vektverdier går fra 1-10
- Skal stå "Velg vedlagt fil"
- Tilbakemelding for svaralternativer
- Se videre på dersom man bare har ett fag

### 10.8.11 Møtereftrat 16.05.2001

Møte mellom prosjektgruppa og veileder

16.05.2001

**Deltagere: Terje, Thomas, Elin og Frode**

- **Opplysninger:** Tirsdag + onsdag i neste uke er Frode her. Rapporten skal leveres onsdag kl.12.00. Dersom vi ønsker innspill fra Frode angående denne, må rapporten mailes til ham, og da helst fredag eller senest mandag.
- **Konklusjon:** settes til slutt i rapporten
- **Studentvisningsmodul:** Ta bort auto-visning på svaralternativene, men ikke på "Tilbakemelding" + "spørsmålsbeskrivelse". Her skal auto-visning av filer være mulig.
- **Fag; "Fagets tester":** link. Skal være en oversikt over alle tester i et spesielt fag. Mer space mellom fagkode og fagnavntekst i opplistingen er ønskelig.
- Når man er inne og ønsker **opplisting av tester i et fag**, angis disse med angivelse om de er oblig'er eller statutester, fagnummer og testnavn, fagets navn trenger ikke å være med i denne opplistingen, da vi ønsker opplisting av tester i et bestemt fag.
- **"Editere tester";** Forutsette at en annen test er bestått; trenger ikke å displaye seg selv. Sortere testene etter om de er statutester eller oblig'er. Angis med S: "Testnavn" eller O: "Testnavn".
- **Oversiktside** med alle testenes URL bør lages.
- **"Spørsmål":** Feltbredden bør reguleres til å bli lik. Ta bort avhukingsboks "Vis alle Fag/Kurs".
- **"Vis tester":** Grånyanser byttes ut med grønn, tilstrebe mer kontrast i opplistingen.
- **"Sjekk godkjentstatus":** bør utvikles ferdig både for studenten og læreren
- **Superbrukermodul:** Bør ferdigutvikle: slette fag/tester, spørsmålstypers forklaringstekst og lovlig karakterkala med angivelse av beste og dårligste karakter (angis ved bruk av float)
- **Studentens spørsmålsmodul:** Angi spørsmålsnummer; f. eks ; spørsmål nr.2 av 4, når spørsmålet presenteres for studenten.