

## HOVEDPROSJEKT:

# Læringssimulator

For å lære hvordan en avis lages

## FORFATTERE:

MARTIN HALVORSEN  
HANS CHRISTIAN SAGBAKKEN  
HÅVARD NARVESEN  
RUNE VANDLI

Dato: 18. mai 2005

## Sammendrag av Hovedprosjektet

Tittel:	LÆRINGSSIMULATOR		Nr. : 11
			Dato : 18.05.05
Deltaker(e):	Martin Halvorsen Håvard Narvesen Hans Christian Sagbakken Rune Vandli		
Veileder(e):	Professor Rune Hjelsvold		
Oppdragsgiver:	Innlandets Vitensenter		
Kontaktperson:	Stig Sægrov		
Stikkord (4 stk)			
Antall sider:66	Antall bilag:17	Tilgjengelighet (åpen/konfidensiell): Åpen	
<p><b>Kort beskrivelse av hovedprosjektet:</b></p> <p>Oppgaven har gått ut på å utvikle en prototyp av en læringssimulator for Innlandets Vitensenter. Vitensenteret skal bygge et senter hvor elever i grunnskolen skal utfordres i en aktiv læringsprosess. Senteret vil få navnet MedieEksperimentet og skal stå ferdig i 2006. Målet med læringssimulatoren er å vekke medieinteressen for elever i grunnskolen.</p> <p><b>Kort om læringssimulatoren:</b></p> <p>Elevene får utdelt hver sin håndholdte PC, en såkalt PDA. Med denne PDA-en og virtuelle redaksjoner skal elevene via en kreativ prosess sette sammen en nettavis. Elevene blir delt inn i grupper og fordelt på forskjellige redaksjoner, hvor de får presentert en handling. Deretter skal elevene lage en nettavis og publisere riktige artikler i forhold til den faktiske situasjonen.</p> <p>Elevene må samle informasjon om situasjonen og sende dette til redaksjonen. Denne informasjonen vil være utgangspunktet for hvilke artikler som skal publiseres i nettavisen. Oppgaven med å sette sammen nettavisen skal være en kreativ prosess og resultatet vil variere fra gruppe til gruppe.</p> <p>Læringssimulatoren tar i bruk mye ulik teknologi der blant annet trådløs kommunikasjon mellom PDA-er og server står sentralt.</p>			

## Forord

Læringssimulatoren er utviklet av 4 studenter ved Høgskolen i Gjøvik gjennom et hovedprosjekt våren 2005. Oppdragsgiver for prosjektet var Stig Sægrov, daglig leder av Innlandets Vitensenter AS.

Formålet med prosjektrapporten er å beskrive arbeidet prosjektgruppen har gjort gjennom prosjektperioden. Den vil også være til hjelp for andre som skal fortsette arbeidet med systemet. Rapporten inneholder beskrivelser av prosessen vi har gjennomført samt forklaringer, eksempler og annen nyttig dokumentasjon.

Vi vil gjerne takke veilederen vår, Professor Rune Hjelsvold for god oppfølging og veiledning gjennom prosjektperioden. En stor takk går også til Sivilingeniør Dag Solhaug, daglig leder ved ETC. I tillegg ønsker vi å takke Høyskolelektor Rune Lossius og Øyvind Johansen ved ETC for hjelp med .NET rammeverket.

Til slutt vil vi takke oppdragsgiver for lån av teknisk utstyr som bidro til at prosjektet ble ekstra praktisk og lærerikt.

Gjøvik, den 18. mai 2005

---

Rune Vandli

---

Hans Christian Sagbakken

---

Martin Halvorsen

---

Håvard Narvesen

# Innholdsfortegnelse

<b>1 INNLEDNING.....</b>	<b>7</b>
1.1 OPPGAVEDEFINISJON.....	7
1.2 MÅLGRUPPE .....	8
1.3 FORMÅL.....	8
1.4 BAKGRUNN OG KOMPETANSE .....	8
1.5 UTVIKLINGSMODELL .....	9
1.6 RAMMER OG ARBEIDSFORMER .....	9
1.7 RAPPORT ORGANISERING .....	10
<b>2 KRAVSPESIFISERING .....</b>	<b>11</b>
2.1 SYSTEMETS BRUKERE .....	11
2.2 GROV AVGRENSNING.....	12
2.3 OMGIVELSER .....	12
2.4 PROSJEKTGRUPPENS SYSTEM .....	13
2.4.1 Avvik fra Vitensenterets totale system .....	13
2.4.2 Prosjektgruppens system.....	13
2.5 BESKRIVELSE AV INNHOLDET ROLLESPILLET.....	14
2.5.1 Flyten i registreringer.....	14
2.5.2 Flyten ved valg av journalistrollen .....	15
2.5.3 Flyten ved valg av redaksjonsjournalistrollen .....	16
2.6 FUNKSJONELLE KRAV TIL HVER MODUL.....	17
2.6.1 Krav til Administratormodul og Bildemodul.....	17
2.6.2 Krav til Redaksjonsmodulen .....	18
2.6.3 Krav til PDA/Mediamodul.....	19
2.7 USE CASE DIAGRAMMER.....	20
2.7.1 Use Case diagram, redaksjonsmodulen .....	20
2.8 RISIKOANALYSE.....	22
2.8.1 Analyse av Bildemodulen.....	23
2.8.2 Analyse av PDAmodulen .....	23
2.8.3 Analyse av Instruktørmodulen (Administratormodulen) .....	23
2.8.4 Analyse av Redaksjonsmodulen.....	23
2.8.5 Oppsummering .....	24
2.9 VEDLIKEHOLD .....	25
2.10 DOKUMENTASJON.....	25
2.11 UTGIVELSER UNDERVEIS.....	25
2.12 KRAV TIL UTVIDELSER .....	25
<b>3 DESIGN .....</b>	<b>26</b>
3.1 LØSNING PÅ SERVER .....	27
3.1.1 Kort beskrivelse av modulene på server .....	28
3.2 LØSNING PÅ PDA.....	28
3.3 KOMMUNIKASJON.....	29
3.3.1 IP-adresser.....	29
3.3.2 Socket.....	29
3.3.3 Klientene (PDA).....	30
3.3.4 ASP.NET .....	30
3.4 GRAFISK BRUKERGRENSESNITT .....	31
3.4.1 Skjermbilder for administratormodulen.....	31
3.4.2 Skjermbildet for redaksjonsmodulen.....	33
3.4.3 Skjermbilder for PDAmodulen .....	34
3.5 KLASSEDIAGRAM .....	36
3.5.1 Klassediagram for PDAmodulen .....	36
3.5.2 Klassediagram for redaksjonsmodulen .....	37
3.5.3 Klassediagram for administratormodulen.....	38
3.5.4 Klassediagram for bildemodulen .....	39
3.5.5 Klassediagram for pakken Data.....	39
3.5.6 Klassediagram for pakken DB.....	40
3.6 PAKKEDIAGRAM FOR LØSNINGEN PÅ SERVER .....	41
3.7 VERKTØY I DESIGNPROSESSEN .....	41

<b>4 IMPLEMENTASJON</b>	<b>42</b>
4.1 UTVIKLINGSMILJØ	42
4.1.1 Valg av utviklingsmiljø	42
4.2 DATABASEPLATTFORM	43
4.3 KODESTANDARD	44
4.3.1 Eksempel på kode	45
4.4 KODING AV ULIKE KOMPONENTER I SYSTEMET	46
4.4.1 Koding av Administratormodulen	46
4.4.2 Koding av PDAmodulen	48
4.4.3 Koding av Bildemodulen	49
4.4.4 Koding av Nettmodulen	49
4.4.5 Koding av Redaksjonsmodulen	50
4.4.6 Logging av hendelser i systemet	51
4.5 UTVIKLING AV DATABASEN	52
4.6 GJENBRUK AV KODE I PROSJEKTET	53
4.7 VERSJONSHÅNDTERING	54
4.8 INSTALLASJON	54
<b>5 DISKUSJON AV RESULTATER</b>	<b>55</b>
5.1 FREMDRIFTSPLAN	55
5.2 ARBEID MED KRAVSPESIFISERING	56
5.3 REALISERING AV KRAV TIL SYSTEMET	56
5.3.1 Krav til administratormodul og bildemodul	56
5.3.2 Krav til redaksjonsmodulen	57
5.3.3 Krav til PDAmodulen	57
5.4 PROBLEMER UNDERVEIS	57
5.4.1 Problemer med oppsett av nettverk	57
5.4.2 Problemer med koding	58
5.5 VIDERE ARBEID MED SYSTEMET	59
5.5.1 PDAmodulen	59
5.5.2 Administrasjonsmodulen	59
5.5.3 Redaksjonsmodulen	60
5.6 DISKUSJON AV RETNINGSLINJER FRA OPPDRAGSGIVER	60
5.6.1 Utviklingsmiljø	60
5.6.2 Trådløst nettverk 802.11b	61
5.6.3 Klientene i systemet	62
5.7 KRITIKK AV OPPGAVEN	62
5.7.1 Fremdriftsplan	62
5.7.2 Dialog med oppdragsgiver	63
5.7.3 Versjonshåndtering	63
5.7.4 Koding	63
5.7.5 3D modellen	64
5.8 EVALUERING AV PROSJEKTET	64
5.8.1 Organisering	64
5.8.2 Arbeidsfordelingen	64
5.8.3 Subjektiv opplevelse av hovedprosjektet	65
<b>6 KONKLUSJON</b>	<b>66</b>
<b>BIBLIOGRAFI</b>	<b>67</b>
<b>VEDLEGG</b>	<b>68</b>

## Figurliste

Figur 2.1 Brukerne i læringssimulatoren	12
Figur 2.2 Flytdiagram for registreringen	14
Figur 2.3 Flytdiagram for journalistrollen	15
Figur 2.4 Flyten som journalist på desken	16
Figur 2.5 Brukere og valgmuligheter i redaksjonsmodulen	20
Figur 3.1 Illustrerer modulene på serveren og lagdelingen	27
Figur 3.2 Kommunikasjon mellom server og klient	29
Figur 3.3 Viser hvordan skillearket Meny er konstruert	31
Figur 3.4 Viser skjermbildet ved tildeling av redaksjoner	32
Figur 3.5 Den virtuelle redaksjonen	33
Figur 3.6 Registrering på PDA	34
Figur 3.7 Slik ser pressekortet ut ved endt registrering	34
Figur 3.8 Innsending av opplysninger	35
Figur 3.9 Klassediagram over PDA software	36
Figur 3.10 Klassediagram over redaksjonen	37
Figur 3.11 Klassediagram over administratormodul	38
Figur 3.12 Klassediagram over bildemodulen	39
Figur 3.13 Klassediagram over pakken Data	39
Figur 3.14 Klassediagram over database pakken kalt DB	40
Figur 3.15 Pakkene plassert i de lag de tilhører	41
Figur 4.1 Eksempel på kode	45
Figur 4.2 Håndtering av tilkobling fra PDA	47

## **1 Innledning**

Dagens teknologi blir stadig rimeligere og mer tilgjengelig, hvilket åpner for nye muligheter. I de siste årene har såkalte læringssimulatorer blitt populære. Felles for alle læringssimulatorer er at man som besøkende blir en aktiv deltaker i et rollespill. Man ønsker å simulere en virkelig setting for å la brukere prøve seg i ulike roller.

Formålet er at et besøk, enten det er ved et museum eller en lokalavis, skal være så lærerikt som mulig. De besøkende ved disse sentrene er med på å forme sitt eget besøk. Hovedpoenget med å lage slike senter bygger på det faktum at vi mennesker lærer best når vi fysisk utfører noe på egenhånd. Det finnes et kinesisk ordtak som sier:

”Det jeg hører, glemmer jeg. Det jeg ser, husker jeg. Det jeg gjør, forstår jeg.”

### **1.1 Oppgavedefinisjon**

Innlandets vitensenter skal bygge et senter hvor elever fra grunnskolen skal få øynene opp for medier. Senteret vil inneholde mye teknologi og skal utfordre brukernes kreative evner.

I læringssimulatoren vil elever delta i et interaktivt rollespill for å lage en moderne avis, nettavis, TV og radio. Dette vil foregå ved at hver enkelt elev får utdelt en håndholdt PC, det vil si en PDA (Personal Data Assistant). Ved hjelp av denne PDA-en skal elevene jobbe sammen i grupper for å sette sammen en avis. Gjennom rollespillet vil elevene fylle ulike roller og gjøre ulike oppgaver. Målet med en slik lab er å gi elever en ny og aktiviserende måte å lære om aviser. Temaer som vil berøres er presseetikk, kildekritikk og pressens betydning i samfunnet. I tillegg vil det fokuseres på å gjøre eksperimenter og små forsøk. Det endelige senteret som inneholder PDA-er, interaktive redaksjoner og mye annet teknisk utstyr, utgjør det som vil få navnet MedieEksperimentet.

I dette prosjektet har prosjektgruppen fokusert på å få på plass enkelte komponenter i systemet som utgjør læringssimulatoren. Målet for gruppen er å ende opp med en forenklet prototyp, der vi begrenser oss til et medium, nemlig å lage en nettavis.

## 1.2 Målgruppe

Målgruppen for dette prosjektet er oppdragsgiver og andre personer som har interesse av å vite mer om prosjektet og utviklingen. Brukergruppen er i hovedsak ungdomsskoleelever samt en eller flere administratorer som tar seg av det administrative i systemet. Prosjektgruppen har laget en prototyp av en læringssimulator og dersom vitensenteret er fornøyd med resultatet, vil de videreutvikle systemet.

Gruppen har gjennom hele prosjektperioden tatt hensyn til at sluttbruker av det endelige systemet hovedsakelig blir ungdomsskoleelever. Vi har derfor tilpasset brukervennligheten og det grafiske brukergrensesnittet til denne gruppen.

Målgruppen for rapporten er sensor, Innlandets Vitensenter og de som eventuelt skal videreutvikle dette systemet.

## 1.3 Formål

Dette prosjektet ble definert av Innlandets Vitensenter. Ved at oppdragsgiver ønsket å komme i gang tidlig med utviklingen av læringssimulatoren, åpnet det seg muligheter for prosjektgruppen. Visjonen til Innlandets Vitensenter er å lage en interessant læringssimulator hvor fortrinnsvis elever fra ungdomskolen skal synes det er gøy å delta.

Opgaven til prosjektgruppen går ut på å utvikle og sette opp et system som snart skal bli en del av læringssimulatoren. Her har gruppen fokusert på gjennomførbarhet og lagd et system som kan demonstreres.

Formålet med rapporten er å gi de som skal videreutvikle et system en innføring i hvordan systemet er laget.

## 1.4 Bakgrunn og kompetanse

Alle gruppemedlemmene går Master i teknologi, Mediteknikk ved Høgskolen i Gjøvik. De tre første årene følger vi bachelor i ingeniørfag, data. Fra høsten 2004 kunne skolen tilby en studieretning kalt "Programvareutvikling". Alle på gruppen valgte den nye studieretningen som er satt sammen av systemutvikling og programmering.

Gruppens programmeringserfaring stammer fra C++ og Java. Til sammen har hvert av gruppemedlemmene hatt tre kurs med C++ ("Grunnleggende programmering", "Objektorientert programmering" og "Algoritmiske metoder") og ett kurs i Java ("Programutvikling"). Systemutviklingsteori har vi jobbet med i to fag, "Systemutvikling 1" og "Objektorientert Systemutvikling".

Prosjektgruppen har dratt nytte av kunnskap vi har tilegnet oss på egenhånd. Dette gjelder spesielt innen 3D modellering. Figuren vi bruker flere steder i prosjektet er utviklet i 3D Studio Max. I tillegg har vi noe erfaring med utvikling av websider. Begge disse områdene har vi fått bruk for i arbeidet med oppgaven.

I dette prosjektet ble gruppen fra oppdragsgivers side oppfordret til å utvikle i .NET. Vi ble dermed nødt til å sette oss inn i en helt ny verden. Det har nødvendigvis tatt tid å bli kjent i det nye rammeverket.



## 1.5 Utviklingsmodell

Utviklingsmodellen for denne prosjektoppgaven har vært inkrementell systemutvikling. Ved å følge denne modellen kan man utvikle systemet modul for modul. Etter hvert som moduler utvikles, implementeres de inn i systemet.

Den inkrementelle utviklingen har innslag av både fossefallsmodellen og den evolusjonære utviklingsmodellen. Gruppen har hatt mulighet til å implementere de viktigste elementene fra kravspesifikasjonen først. Ved å bygge systemet stein for stein, har man god kontroll på hver enkelt modul. Man kan utvikle et inkrement ferdig for så å revidere og tilpasse inkrementet til det tilfredsstillende krav som er satt. En annen fordel er at man kan tidlig sette i gang å bygge systemet. I motsetning til fossefallsmodellen, gjør man ikke alt designet i en og samme prosess før man starter utviklingen.

Prosjektgruppen føler at den inkrementelle utviklingsmodellen har passet prosjektet godt. Dette til tross for at det i starten var litt usikkerhet rundt hvorvidt vi klarte å dele systemet opp i gode nok inkremitter.

## 1.6 Rammer og arbeidsformer

Hovedprosjektet for ingeniørfag, data, utgjør 20 av de totalt 30 studiepoengene gruppemedlemmene tar våren 2005. Arbeidet med oppgaven startet i januar 2005 og arbeidet har holdt på alle ukedager frem til innleveringsfristen.

I begynnelsen hadde ikke gruppen tilgang til noe grupperom. Heldigvis har alle gruppemedlemmene hver sin bærbare pc. På den måten greide gruppen seg ved å sitte i korridorene på skolen. Fra februar har datalab A115 fungert som grupperom. På dette rommet har systemet vært testet og satt opp gjennom prosjektperioden. Systemet har fysisk bestått av en server, et trådløst nettverk, et webkamera og to stk PDA-er. Dermed ble systemet testet underveis i utviklingen.

Siden gruppen består av fire medlemmer, har det vært naturlig for å jobbe to og to i mange sammenhenger. Både rapporten og programmet er laget ved at to og to har hatt ansvaret for moduler og oppgaver.

I noen tilfeller har prosjektgruppen utarbeidet oppgavelister. Oppgavelistene er dokumenter hvor vi ramser opp mindre og større arbeidsoppgaver som må utføres. Til hver enkelt oppgave på listen hører det til et felt med hvem som har ansvaret for oppgaven, et statusfelt og dato for fullføring. Oppgavelistene har bidratt til økt motivasjon, god oversikt og til å sikre at oppgaver ikke blir glemt, se vedlegg J.

Selve rapporten har gruppen bygget stegvis. Vi har delt hvert av hovedkapitlene opp i separate dokumenter og jobbet med dem underveis. Versjonshåndtering av dokumentene har gruppen praktisert. Første utkast av et dokument har fått versjonsnummer v1.0. Liten v står for Versjon. For hver store oppdatering, enten som følge av innspill eller andre årsaker, er det laget en kopi av filen og versjonsnummeret har økt. Det vil si at når det eksempelvis er gjort store endringer på kravspesifikasjonsdokument v1.3, lagres det oppdaterte dokumentet som kravspesifikasjonsdokument v1.4. På denne måten har prosjektgruppen hatt god kontroll på de ulike versjonene av dokumentene som utgjør hovedkapitlene i denne rapporten. Til koden er det benyttet et versjonshåndteringssystem, se kapittel 4.7.

## **1.7 Rapport organisering**

Inndelingen av innhold i rapporten baserer seg på malen for hovedprosjekt ved Høgskolen i Gjøvik.

Rapporten inneholder følgende kapitler:

### **1. Innledning**

I innledningen finnes oppgavedefinisjon, målgruppe, formål, bakgrunn og kompetanse, utviklingsmodell og rammer.

### **2. Kravspesifisering**

Dette er kontrakten mellom prosjektgruppa og oppdragsgiver. Kapitlet tar for seg kravene til systemet.

### **3. Design**

Kapitlet tar for seg designet av systemet. Inneholder skjermbilder, klassediagrammer og diverse diagrammer.

### **4. Implementasjon**

Kapitlet tar for seg implementeringen av funksjonelle krav til systemet.

### **5. Diskusjon og resultater**

Diskusjon rundt flere av aspektene i arbeidet med prosjektet.

### **6. Konklusjon**

Kort konklusjon

### **7. Vedlegg**

Supplement til hovedrapporten.

## 2 Kravspesifisering

I dette kapitlet beskrives krav til systemet sammen med brukerfunksjonalitet. Kravspesifiseringen er delt i to dokumenter av hensyn til dokumentets størrelse. Derfor er brorparten av diagrammer og beskrivelser plassert i vedlegg B. Diagrammene i vedlegget beskriver kravene fra dette kapitlet billedlig.

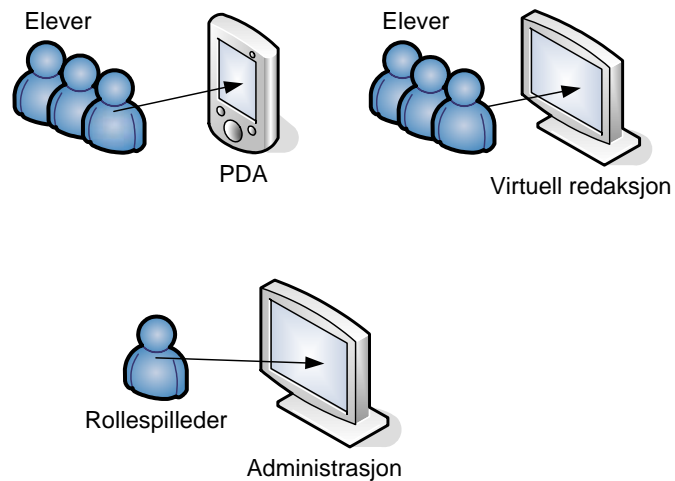
Da prosjektgruppen sammen med oppdragsiver skulle lage en beskrivelse av krav til prototypen av læringssimulatoren, fantes det allerede en spesifisering av den totale medielaben. Det ble tidlig klart at gruppen ikke ville greie å dekke alle krav til den fullstendige læringssimulatoren, på grunn av omfanget til et slik system. Til dette prosjektet er det derfor utarbeidet en egen spesifisering tilpasset utvikling av en prototyp.

I kapitlet er det satt fokus på å avklare hva gruppen skal utvikle av den totale medielaben til Innlandets Vitensenter.

### 2.1 Systemets brukere

I selve rollespillet har man to hovedbrukere. Disse er elev og rollespiller. Brukergruppene har ulikt ansvar og det er eleven som har mest interaksjon med systemet. Rollespiller har kun administrasjonen å forholde seg til. Eleven skal bruke både PDA og redaksjon. I samråd med oppdragsiver har gruppa i denne oppgaven valgt å se bort ifra vedlikehold og personen som blir ansvarlig for dette.

Navn	Beskrivelse	Rolle
Elev	Hovedbruker av systemet.	Skal registrere seg vha PDA, ta bilde, velge rolle i redaksjon, innhente informasjon, sette sammen nettavis osv.
Rollespiller	Administrator. Ordensvakt gjennom rollespillet.	Skal knytte deltakere opp mot redaksjon og sette i gang/stanse rollespillet. Veilede elever gjennom rollespillet.



**Figur 2.1 Brukerne i læringssimulatoren**

## 2.2 Grov avgrensning

Mot slutten av prosjektperioden, ca 10. mai, skal prosjektgruppens system være ferdig. Nedenfor følger de oppgaver som vårt system skal dekke:

- Registrering av PDA med tilhørende "eier" gjennom gjeldende rollespill. Dette innebærer registrering av fornavn, etternavn og klasse. I tillegg skal eleven ta et bilde av seg selv. Når dette er gjort, blir PDA med tilhørende elev registrert som aktiv.
- Rollespiller skal kunne knytte en registrert elev med PDA til en redaksjon i administrasjonen.
- En redaksjon skal da kjøres med valgte deltakere.
- Et rollespill startes og oppgaver tildeles elever.
- Arbeid mot nettavis, der deltakerne i redaksjonen deles i "utøvende journalister" og "desk arbeidere". Dermed skal den ene delen av elevene hente informasjon, mens de andre bearbeider informasjon og setter sammen en nettavis.

## 2.3 Omgivelser

Fra oppdragsgivers side er det bestemt at løsningen skal utvikles vha .NET rammeverket. Til applikasjon på PDA skal det benyttes C# .NET Compact Framework. Til programvare som skal utvikles til PC, skal det benyttes C# .NET fra standardrammeverket i Microsoft Visual Studio .NET.

Systemet vil omfatte SQL, ASP (Active Server Pages), C# .NET, C#.NET Compact Framework, WLAN nettverk og oppsett av server.

Operativsystem er MS Windows Mobile på PDA og MS server 2003 Enterprise Edition på server.

## **2.4 Prosjektgruppens system**

### **2.4.1 Avvik fra Vitensenterets totale system**

Prosjektgruppen ligger foran Vitensenteret når det gjelder utviklingen av medielaben. Av den grunn har ikke gruppen tilgang på diverse utstyr som dokkingstasjon, lyddusj, skriver og lydopptaks enhet. Gruppen har sammen med oppdragsgiver nedprioritert enkelte moduler. De viktigste avvikene følger under.

#### Moduler og funksjonalitet som ikke vil prioriteres:

- Vedlikeholdsmodul
- Filsynkroniseringsmodul
- Internettmodul
- Oppsummeringsmodul
- Mediemodul (Lyddusj og film nedprioriteres i første omgang)
- Eksperimenter med opptak av lyd/film
- Oppsummering med projektor
- "Fax"

At gruppa ikke kommer til å prioritere noen av modulene innebærer at disse ikke er nødvendig for å kjøre rollespillet, men de er en nødvendig del av et helhetlig system.

Alle punktene ovenfor vil bli vurdert implementert i systemet dersom gruppen får tid innenfor de rammene som er avsatt for dette prosjektet. Gruppen vil uansett forsøke å ta hensyn til alle moduler under utviklingen, siden alt skal på plass i det endelige systemet til Innlandets Vitensenter. Prosjektgruppen ser ikke bort fra at det må revurderinger til for hva som skal implementeres underveis i utviklingen.

### **2.4.2 Prosjektgruppens system**

Prosjektgruppens prototyp vil ta for seg noen av modulene som er spesifisert fra oppdragsgivers side. Gruppen vil forsøke å lage en begrenset løsning som kan stå på egne bein under utviklingen. På denne måten får gruppen testet de fleste sidene ved systemet. Det viktigste for prosjektgruppen blir å demonstrere gjennomførbarhet.

Prosjektgruppen ønsker å dekke deler av følgende moduler:

- PDAModulen
- Instruktørmodul
- Mediemodul(Dvs. bildemodul)
- Redaksjonsmodul
- Nettavismodul
- Rollespillmodul

For at gruppen skal kunne lage en demo av rollespillet, må vi berøre mange av modulene. Ved at gruppen jobber mot flere av modulene, blir systemet testet i en helhet.

## 2.5 Beskrivelse av innholdet rollespillet

I samråd med oppdragsgiver, er det laget figurer som viser gangen i rollespillet. Prosjektgruppen har som mål å lage en prototyp som baserer seg på tankegangen fra disse figurene.

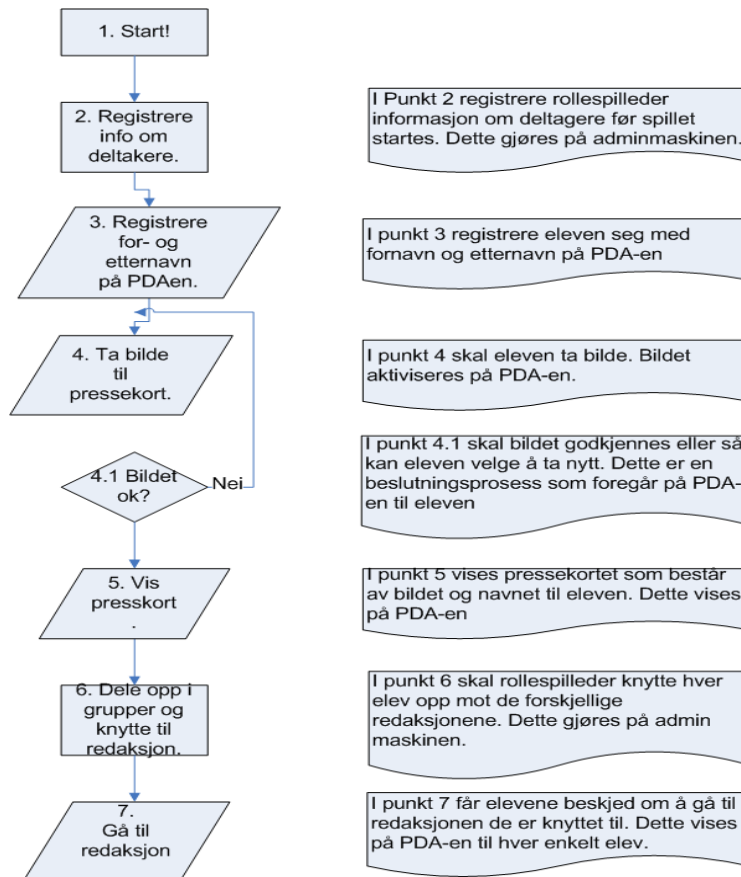
### 2.5.1 Flyten i registreringen

Alle elevene som skal være med i rollespillet, ankommer lokalene til læringssimulatoren. Elevene samles slik at rollespilleleder kan gi informasjon om rollespillet og hvordan dette fungerer. Alle elevene vil få utdelt en PDA hver og få en kort innføring i hvordan denne skal brukes under rollespillet.

Rollespilleleder registrerer informasjon om gruppen som skal bruke læringssimulatoren. Typisk gruppeinformasjon er klasse og skole.

Når elevene har fått forståelse for hvordan PDA-en skal brukes, kan de registrere navnet sitt på registreringssiden. Etter vellykket registrering skal elevene få muligheten til å kunne ta et bilde av seg selv via et webkamera. Måten dette blir gjort på er at eleven setter seg i en stol foran webkamera og eleven trykker selv på en knapp på PDA-en. Når dette er gjort, vil elevene få en melding på skjermen om at de må møte i redaksjonen som de tilhører.

Registreringen skal være det første som skjer i rollespillet. Her skal systemet lagre viktig informasjon om de ulike brukerne som skal brukes gjennom hele rollespillet. På figur 2.2 er det beskrevet hvordan registreringen skal foregå.



Figur 2.2 Flytdiagram for registreringen

### 2.5.2 Flyten ved valg av journalistrollen

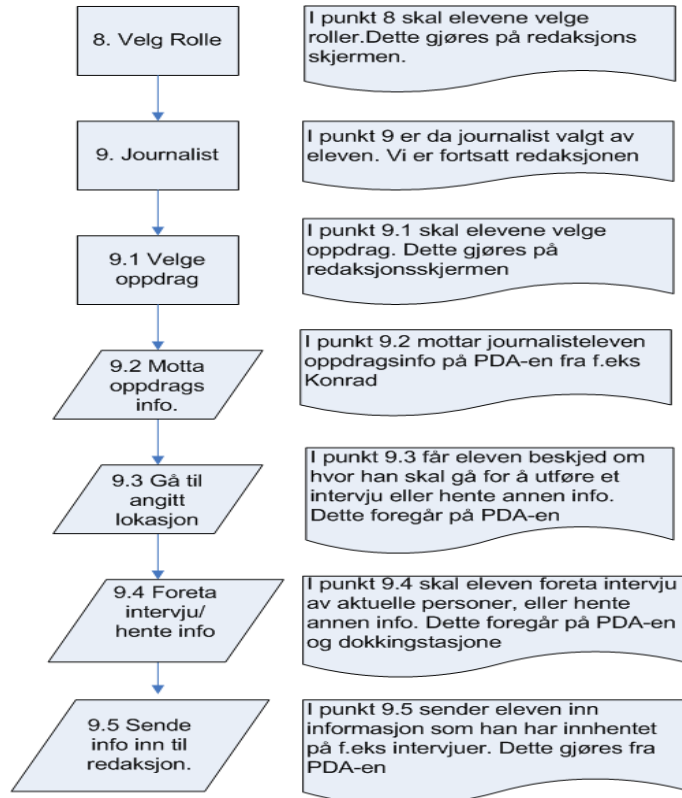
Når en elev er registrert og senere tilknyttet en redaksjon, skal eleven foreta et valg. Dette valget består i å velge rolle i den virtuelle redaksjonen. Det skal være to roller, den ene rollen er utøvende journalist og den andre er å jobbe som redaksjonsjournalist.

Den sentrale delen av redaksjonen vil være en datamaskin som skal gi alle elevene som tilhører redaksjonen, informasjon som inngår i rollespillet. Informasjonen vil bli gitt i form av tekst og en hjelper med navn Konrad. Når informasjon om rollespillets story er gitt, vil hver elev få muligheten til å velge hvilken arbeidsoppgave de vil ha. For elevene er det to muligheter. Man kan velge å jobbe som journalist ute i felten eller å jobbe på desken der man skal sette sammen en nettavvis.

De elevene som er tildelt journalistrollen, vil hele tiden bevege seg i et rom med oppsatte informasjonsplakater og stedsnavn. Rommet skal simulere at de er ute i en virkelig verden. I denne verdenen skal de bevege seg rundt og besøke forskjellige plasser. PDA-en vil være deres hovedkommunikasjonsmiddel i rollespillet. Elevene vil få velge mellom flere påstander på PDA-en sin.

Informasjonen som journalisteleven har tilegnet seg vha intervju, vil danne grunnlaget for hvilke opplysninger som vedkommende velger. Personene som intervjues er folk som er direkte involvert i storyen, f.eks "Kapteinen på Skibladner". De valgte påstandene sendes så inn til redaksjonen slik at redaksjonselevne får dem opp på skjermen sin og kan finne frem til artikler som er gode.

På figur 2.3 vises flyten i systemet når eleven har valgt rollen som utøvende journalist.



**Figur 2.3 Flytdiagram for journalistrollen**

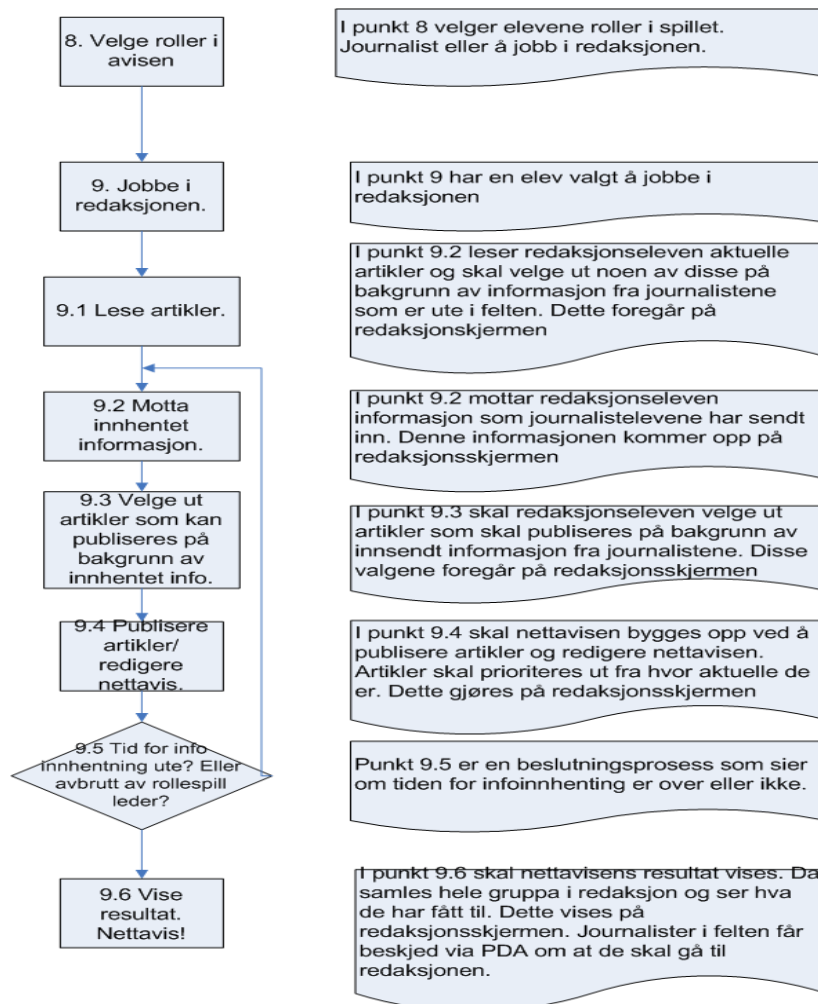
### 2.5.3 Flyten ved valg av redaksjonsjournalistrollen

Elevene som sitter i redaksjonen med nettavisen, vil få informasjonen gitt via skjermen i den virtuelle redaksjonen. I starten vil elevene i redaksjonen for nettavisen få opp en del artikler som omhandler rollespillets story. Disse artiklene er ferdiglagde på forhånd.

Noen av artiklene skal utgjøre den ferdige nettavisen. Hvilke artikler som er riktig å publisere, vil man få hjelp til fra journalistelevene i felten.

På forhånd finnes en rekke påstander som kan være både riktige og gale. Et eksempel på en riktig påstand kan være at "Mjøsormen bare er oppspinn", et eksempel på en gal påstand kan være at "Mjøsormen er død". Disse påstandene skal journalist elevene velge mellom og sende inn til de som jobber i redaksjonen. Ut fra påstandene som kommer inn til redaksjonen, skal redaksjonselevene til slutt velge artikler til nettavisen.

I figur 2.4 er det beskrevet hvordan arbeidsflyten skal gå når eleven(e) jobber som journalist på desken i den virtuelle redaksjonen.



**Figur 2.4 Flyten som journalist på desken**



## 2.6 Funksjonelle krav til hver modul

### 2.6.1 Krav til Administratormodul og Bildemodul

Krav 1	På administratormodulen skal bilder tas fysisk ved hjelp av bildemodulen. Bildet blir vist til den aktuelle PDA som tok bildet. Bildet skal enten godkjennes eller ikke godkjennes. Disse valgene tas på PDA enheten. Selve bildet knipses fra PDA til hver elev (se avsnitt 2.7.3).
Krav 2	Rollespillederen skal kunne se en oversikt over alle som har registrert seg og logget på systemet. Her kan Rollespilleder se hvem som har klart og logge seg på og kan eventuelt hjelpe de som ikke klarte det. Informasjon som vises er fornavn, etternavn samt en PDA id.
Krav 3	Når alle elevene har registrert seg med fornavn og etternavn på PDA, skal rollespill leder ved hjelp av adminmaskinen knytte påloggede elever til de forskjellige redaksjoner. Det skal også være mulig å knytte opp PDA-ene på forhånd, slik at rollespillet er fullstendig satt opp før start dersom rollespilledere ønsker dette.
Krav 4	Rollespillet styres fra adminmaskinen og rollespilleder kan starte og stoppe rollespillet fra denne maskinen. Når alle registrerte elever har logget seg på og blitt knyttet til en redaksjon, vil rollespill leder kunne starte rollespillet fra admin maskinen. Rollespill leder vil også senere kunne stoppe rollespillet hvis noe skulle gå galt.
Krav 5	Rollespilleder skal kunne se hvor langt en gruppe i redaksjonen har kommet i rollespill prosessen. Rollespillet deles opp i faser, og rollespilleder vil ut fra dette få en oversikt over hvor langt hver gruppe har kommet i rollespillet

### 2.6.2 Krav til Redaksjonsmodulen

Krav 1	Alle påloggede elever som er knyttet til en redaksjon, vil få en innledning om problemsituasjonen og storyen, og skal ut fra dette kunne sette seg inn i problemsituasjonen. Dette vil foregå på hovedskjermen i redaksjonen.
Krav 2	Ut ifra presentasjonen av storyen skal elevene kunne velge sin rolle i redaksjonsarbeidet. Dette gjøres på hovedskjermen i redaksjonen. Det finnes to roller. Disse er journalistrollen og redaktørrollen. Elevene i redaktørrollen skal stå i redaksjonen og motta info fra elevene i journalistrollen som innhenter opplysninger. Informasjonen sendes til redaksjonen. De som jobber i redaksjonen skal ut fra disse opplysningene prioritere artikler som til slutt danner en nettavis. Minimum en og maksimum to elever kan være redaktør.
Krav 3	Elevene skal kunne velge oppdrag ut ifra hvilken rolle den enkelte har valgt. Dette gjøres på hovedskjermen i redaksjonen. Oppdragene tildeles på den enkeltes PDA gjennom hele rollespillet.
Krav 4	Redaksjonen skal vise artikler slik at redaktørene skal kunne velge mellom ulike artikler for så å sette sammen en nettavis. Artiklene er ferdig laget og ligger i databasen.
Krav 5	Redaksjonen skal vise tilgjengelige saksopplysninger som er sendt inn av journalister som har vært ute på oppdrag. Ut fra disse saksopplysningene skal redaktøren velge riktige artikler og publisere dem.
Krav 6	Elevene i redaktørrollen skal kunne velge artikler som er riktige i forhold til den faktiske situasjonen og legge disse til i nettavisen.
Krav 7	Ut ifra opplysninger som blir sendt inn av journalister som er ute på oppdrag, skal redaktøren kunne bytte prioritering på artiklene som er tilgjengelige i redaksjonen.
Krav 8	Redaktøren skal ha mulighet til å fjerne artikler fra nettavisen som han mener er uriktige i forhold til den faktiske situasjonen.

### 2.6.3 Krav til PDA/Mediamodul

Krav 1	For at eleven skal kunne delta i rollespillet, må de registrere seg med fornavn og etternavn. Dette gjøres på PDA-enhetene. Etter at feltene fornavn og etternavn er skrevet inn, skal eleven trykke på "registrer" knappen.
Krav 2	Det er valgfritt for elevene å ta bilde. Dersom eleven ikke ønsker å ta bilde, vil et standard bilde bli brukt. Bildet skal brukes til å lage pressekort. Bildet skal fysisk tas fra server hvor det er tilkoblet et webkamera. Eleven velger å ta bilde fra PDA enheten ved å trykke på knappen "ta bilde". Bildet blir overført til PDA der eleven får valget om å godkjenne eller ikke godkjenne bildet. Dersom eleven ikke godkjenner bildet vil han da få valget om å ta nytt bilde. Velger eleven å godkjenne bildet blir dette lagret i databasen.
Krav 2.1	Sammen med oppdragsgiver har gruppa valgt å ta bilde til pressekort allerede i registreringen. Dvs at alle som skal registrere seg, kan ta bilde før rollespillet setter i gang. Med kun et kamera og 30 elever, vil det nødvendigvis bli kø. Dermed er det mulig at dette heller bør foregå ved hver redaksjon for å unngå treghet i registreringsprosessen. Fordelen med å gjøre det tidlig, er at pressekortet er klart i det elevene ankommer redaksjonene.
Krav 3	Eleven skal kunne få opp profilen sin med navn og bilde. Dette forutsetter at registrering har blitt gjort og godkjent. Hvis dette er godkjent vil eleven kunne få presentert profilen sin, og se hvordan pressekortet sitt ser ut.

## 2.7 Use Case diagrammer

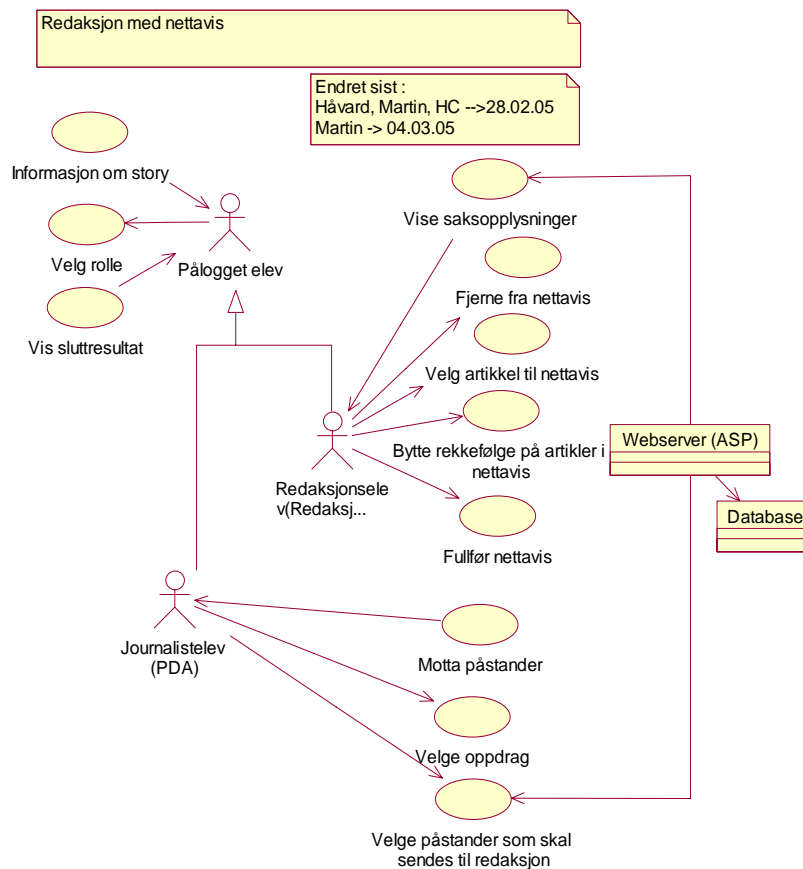
Kravene til systemet er skissert blant annet ved hjelp av Use Case-diagrammer. Disse diagrammene viser hvem som er brukere og hva de skal gjøre i systemet. I dette kapitlet ligger det et eksempel på et slik diagram med tilhørende beskrivelser. Resten av Use Case-diagrammene for systemet finnes i vedlegg B.

### 2.7.1 Use Case diagram, redaksjonsmodulen

Diagrammet nedenfor, se figur 2.5, forklarer hva som skjer under rollespillet når redaksjonsmodulen kjøres. I denne modulen vil eleven ha en del funksjoner til hjelp for å forme en nettavis.

Brukerne vil deles i to grupper. Enten skal en elev være journalistelev som samler inn informasjon, eller så skal eleven jobbe direkte i den virtuelle redaksjonen. I redaksjonen får eleven(e) presentert artikler. Med meldinger som strømmer inn fra reportere, vil man ha grunnlag for å velge å publisere enkelte artikler. Elevene skal samarbeid for å lage innholdet i nettavisen.

Vi har valgt å gi påloggede elever mulighet til å velge mellom redaktørrollen eller journalistrollen. En elev som velger redaktørrollen vil styre nettavisen. En redaktørelev vil holde på med nettavisen mens journalisten velger et oppdrag som foregår ute i felten. Den påloggede eleven velger altså hvilken rolle den ønsker. I tillegg vil alle elevene få informasjon om story og en oppsummering.



**Figur 2.5 Brukere og valgmuligheter i redaksjonsmodulen**

### Høynivå Use Case beskrivelser, Nettavismodul

Navn	Velge rolle
Aktør	Pålogget elev
Forutsetning	At to eller flere elever er knyttet til en redaksjon
Resultat	At elevene er delt inn i stillinger som journalister og redaktører.
Mål	Gi elevene mulighet til å velge innfallsvinkler i det journalistiske arbeid.

Navn	Informasjon om story
Aktør	Utøvende journalistelev og redaksjonselev
Forutsetning	At to eller flere elever er knyttet til en redaksjon
Resultat	At man har mottatt nok informasjon til å vite hva hele "jobben" går ut på.
Mål	Sikre at elevene vet hva de skal gjøre og forstår problemstillingen.

Navn	Velg artikkel til nettavis
Aktør	Redaksjonselev
Forutsetning	At det finnes artikler å velge mellom
Resultat	At artikkelen vises i nettavisen
Mål	Å oppdatere nettavisen med artikkel

Navn	Bytte rekkefølge på artikler i nettavis
Aktør	Redaksjonselev
Forutsetning	At nettavisen inneholder mer enn en artikkel
Resultat	At endring av rekkefølgen ble utført
Mål	At elevene kan få avgjøre rekkefølgen og viktigheten av artiklene selv

Navn	Fjerne fra nettavis
Aktør	Redaksjonselev
Forutsetning	At det er minst en artikkel i nettavisen
Resultat	At artikkelen ble vellykket fjernet.
Mål	At redaksjonselevne skal kunne oppdatere nettavisen

Navn	Vise saksopplysninger
Aktør	Redaksjonselev
Forutsetning	At rollespillet har startet og at roller er tildelt
Resultat	Redaksjonseleven får informasjon som gir han/henne grunnlag for å vurdere innhold i de ferdiglagde artiklene.
Mål	Å hjelpe redaksjonseleven til å velge riktige artikler som skal utgjøre nettavisen.

Navn	Velge oppdrag
Aktør	Utøvende journalistelev
Forutsetning	At en eller flere elever har valgt å jobbe som utøvende journalister.
Resultat	At alle har fått tildelt en jobb og rollespillet kan starte for alvor.
Mål	Gi elevene mulighet til å bestemme arbeidsfordelingen.

Navn	Motta påstander
Aktør	Utøvende journalistelev
Forutsetning	At roller er tildelt og at rollespillet er startet
Resultat	Eleven får muligheten til å gjøre seg opp meninger om hva som er gale og riktige påstander.
Mål	La eleven vurdere informasjon for senere å foreta valg. Det kan være flere ulike måter å motta påstander på. Elevene skal samle informasjon ved å få oversikt over saken.

Navn	Vis sluttresultat
Aktør	Pålogget elev
Forutsetning	At det finnes en avis å presentere.
Resultat	At elevene får presentert resultatet av det de har laget
Mål	Gi en avslutning på rollespillet der elevene får se hva de har jobbet med.

Navn	Velge påstander som skal sendes til redaksjon
Aktør	Utøvende journalistelev
Forutsetning	At opplysninger og påstander har blitt presentert
Resultat	Påstander er sendt til redaksjonen og brukes som grunnlag for å publisere artikler
Mål	La elevene bidra til å velge riktige artikler ved at han/hun har innhentet korrekte opplysninger på en journalistisk korrekt måte.

## 2.8 Risikoanalyse

I perioden etter arbeidet med kravspesifiseringen, ønsket gruppen å kartlegge hvor skoen trykket mest. Gruppen har tatt utgangspunkt i en analyse hentet fra en bok om UML av Craig Larman [1].

I analysen som følger kartlegges de brukstilfeller, også kalt Use Case, som er listet opp for systemet i kravspesifikasjonen. Analysen tar utgangspunkt i en subjektiv gradering hvor det er tre faktorer som danner grunnlaget for utfallet. De tre faktorene er arkitekturisk betydning, risiko og kritikalitet.

Tabellen nedenfor viser hvordan den vektete rangering brukes i analysen. Høyere tall i kolonnen for vekt, betyr større innflytelse.

Faktor	Vekt	Område
Kritikalitet	1	0-3
Arkitekturisk betydning	2	0-3
Risiko	3	0-3

Brukstilfellene er listet opp modul for modul. Hver modul har sin tabell med tilhørende brukstilfeller. Til sist samler vi alle brukstilfellene inn i en tabell hvor de plasseres innenfor en av tre kategorier. Her plasseres de etter høy, middels eller lav prioritet. Denne tabellen er plassert til slutt i dette dokumentet.

Utrekningen av vektet sum gjøres på følgende måte. For eksempel vil et brukstilfelle med kritikalitet lik 1, arkitekturisk betydning lik 1 og risiko lik 1, få følgende regnestykke med utgangspunkt i tabellen ovenfor:

$$\text{Kritikalitet}(1 \cdot 1) + \text{Arkitekturisk betydning}(1 \cdot 2) + \text{Risiko}(1 \cdot 3) = 6$$

**2.8.1 Analyse av Bildemodulen**

Brukstilfelle	Kritikalitet	Arkitekturisk betydning	Risiko	Vektet Sum
Ta bilde (Fysisk)	2	3	2	14
Lagre bilde	3	1	1	7
Totalt				21

**2.8.2 Analyse av PDAModulen**

Brukstilfelle	Kritikalitet	Arkitekturisk betydning	Risiko	Vektet Sum
Deaktivere knapper	2	0	0	2
Velge informasjon som skal sendes til redaksjon	2	2	2	12
Motta info fra intervjuobjekter	1	2	2	11
Vis bilde	2	1	1	7
Godkjenn / Underkjenn bilde	1	0	0	1
Registrere navn	3	2	2	13
Vis profil	1	2	1	8
Velge oppdrag	2	2	2	12
Ta bilde	2	2	2	12
Totalt				78

**2.8.3 Analyse av Instruktørmodulen (Administratormodulen)**

Brukstilfelle	Kritikalitet	Arkitekturisk betydning	Risiko	Vektet Sum
Oversikt over påloggede elever	1	1	1	6
Registrere info om gruppe	3	2	2	13
Vis oversikt over aktive PDA-er	1	2	2	11
Knytte pålogget elev til en redaksjon	3	3	2	15
Start rollespill	3	3	1	12
Stans rollespill	0	3	1	9
Oversikt redaksjon	0	3	1	9
Totalt				73

**2.8.4 Analyse av Redaksjonsmodulen**

Brukstilfelle	Kritikalitet	Arkitekturisk betydning	Risiko	Vektet Sum
Informasjon om story	1	1	0	3
Velg rolle	3	2	1	10
Vis sluttresultat	1	2	1	8
Vise saksopplysninger	3	2	2	13
Fjerne fra nettavis	2	2	1	9
Bytte rekkefølge på artikler i nettavis	1	2	1	8
Fullfør nettavis	0	1	1	5
Velge artikkel til nettavis	3	2	1	10
Totalt				66

### 2.8.5 Oppsummering

I tabellen nedenfor er alle brukstilfellene samlet. Her blir de rangert etter hvor høy vektet sum de har fra tabellene ovenfor. Denne listen gir en oversikt over hvilke systemkrav som vil kreve mest ressurser. Prosjektgruppen har benyttet denne listen som et verktøy gjennom utviklingen.

Rangering	Brukstilfelle (Use Case)	Kommentar
Høy ( 10 → )	Velge artikkel til nettavis Vise saksopplysninger Velg rolle Start rollespill Knytte pålogget elev til redaksjon Vise oversikt over aktive PDA-er Registrere info om gruppe Ta bilde (Virtuelt, Pda) Velge oppdrag Registrere navn Motta info fra intervjuobjekter Velge påstander som skal sendes til redaksjon Ta bilde (Fysisk, bildemodul)	Fundamentale elementer for systemet. Stor betydning for arkitekturen.
Middels (6 – 9)	Bytte rekkefølge på nettavis Fjerne fra nettavis Vis sluttresultat Stans rollespill Oversikt over påloggede elever Vis profil Vis bilde Oversikt redaksjon	Ikke-kritiske elementer, men som samtidig har stor betydning. Middels betydning for arkitektur.
Lav (0 – 5)	Informasjon om story Godkjenn / Underkjenn bilde Deaktivere knapper Fullfør nettavis	Minimale effekter på arkitektur. Mindre betydningsfulle.



## **2.9 Vedlikehold**

Det vil ikke bli gjort noen form for vedlikehold av dette systemet av prosjektgruppen. Dette skyldes at hovedprosjektet avsluttes medio mai 2005.

## **2.10 Dokumentasjon**

All dokumentasjon av koden lages som egne nettsider for hver pakke i systemet. Denne dokumentasjonen genereres automatisk av Visual Studio ved korrekt kommentering av koden.

## **2.11 Utgivelser underveis**

Det skal finnes kun en fullstendig versjon av systemet. Prosjektgruppen ønsker likevel å gjøre det mulig å kjøre demoer på systemet underveis. Demoer av systemet underveis passer godt for utviklingsmodellen prosjektgruppen benytter. Det er planlagt å ha en demo klar til oppdragsgiver ved utgangen av påskeferien.

## **2.12 Krav til utvidelser**

Systemet skal kunne videreutvikles ved en senere anledning. Ved utvidelser skal det tas utgangspunkt i "Teknisk spesifisering av Innlandets Vitensenter sin medielab", se vedlegg P. Gruppen satser på at god dokumentasjon skal gjøre det forholdsvis enkelt å sette seg inn i hvordan systemet virker.

### **3 Design**

Form let med designkapittelet er   gi en detaljert oversikt over hvordan kravene som er fastsatt i kravspesifiseringen skal realiseres.

Designet av totalsystemet har blitt til bit for bit. I utviklingsfasen av hver modul, er det utarbeidet klassediagrammer, systemdiagrammer, og flytdiagrammer. Forslag til strukturering av klasser og pakker er laget for modul i utviklingen.

I dette kapittelet er systemet beskrevet fra flere ulike vinkler. Systemet skal grovt sett best  av to delsystemer. Det ene delsystemet er modulen som skal dekke systemkravene for klienten (PDA), og det andre delsystemet er modulene som er plassert p  serveren. Det f rste som beskrives er l sningen p  server.

### 3.1 Løsning på server

Serveren og løsningen som skal ligge her, vil bli selve hjørnesteinen i systemet. Her skal de aller fleste modulene ligge og rollespillet skal styres på alle nivå.

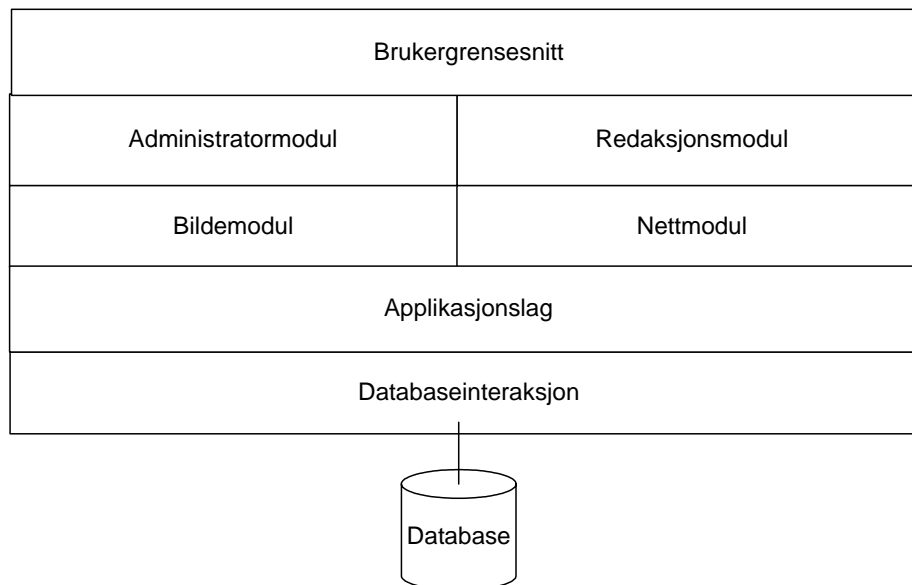
Systemet på server baserer seg på 3-lags arkitektur. I dette tilfellet skilles det på brukergrensesnitt, applikasjonslag og databaselag. Etter ønske fra oppdragsgiver skal systemet utvikles etter modell-view-controller prinsippet.

Ved å lagdele systemet øker man effektiviteten ved utvikling og på ytelse. For eksempel ønsker flere av modulene i systemet å snakke med databasen. I den forbindelse brukes Databaselaget til å snakke med databasen. Klasser med funksjonalitet som flere moduler ønsker å benytte seg av, legges i pakker for felles bruk i det laget de tilhører. De ulike lagene tilbyr tjenester til lag over seg og bruker tjenester fra lag under seg. Man har altså et hierarki hvor de ulike lagene snakker sammen.

Nedenfor følger en figur som beskriver systemet på server. På figur 3.1 er de ulike modulene plassert. Hver modul har sin unike oppgave å løse. Administratormodulen og redaksjonsmodulen er plassert øverst siden de kommer i direkte kontakt med brukere. Bildemodulen og Nettmodulen vil indirekte brukes av rollespilleder og elever og kan derfor plasseres nærmere applikasjonslaget. Med andre ord er administratormodulen og redaksjonsmodulen nært knyttet til brukergrensesnittet i motsetning til bildemodulen og nettmodulen.

Laget under brukergrensesnittet kalles applikasjonslaget. Her er det skilt ut funksjonalitet som ikke har så mye med brukergrensesnittet å gjøre. I applikasjonslaget skal det plasseres pakker og klasser for databehandling.

Det nederste laget skal inneholde funksjonalitet modulene bruker for interaksjon med datakilden. En fullstendig modul vil være sammensatt av elementer fra flere av lagene.



**Figur 3.1** Illustrerer modulene på serveren og lagdelingen

### **3.1.1 Kort beskrivelse av modulene på server**

#### **Administratormodul**

Administratorapplikasjonen skal brukes av rollespilleren og benyttes til å registrere en klasse/gruppe som kommer for å delta i rollespillet. Videre skal rollespilleren knytte hver elev opp mot en redaksjon slik at de kan delta på rollespillet. Rollespillet startes fra administratorapplikasjonen. Rollespilleren vil via denne modulen blant annet kunne se påloggede elever og aktive PDA-er.

#### **Redaksjonsmodul**

Redaksjonsapplikasjonen skal benyttes i arbeidet med å lage selve nettavisen. Etter at storyen er presentert skal gruppene i hver redaksjon velge roller. En elev i den virtuelle redaksjonen vil være redaksjonselev og resten vil være journalister. Redaksjonseleven og journalistelevne skal samarbeide via denne modulen for å publisere en nettavis.

Eleven som jobber i redaksjonen vil motta påstander fra journalistelevne som er ute og samler informasjon om de aktuelle sakene. Ut ifra disse påstandene skal redaksjonseleven velge riktige artikler.

#### **Nettmodul**

Nettmodulen inneholder nettsider som brukes i rollespillet. I systemet skal nettmodulen brukes når man henter nettsider på det lokale nettverket i Læringssimulator. Når en annen modul gjør en forespørsel om en nettside på serveren, vil nettmodulen bearbeide og presentere nettsiden.

Nettmodulen brukes for eksempel av Administratormodulen ved ønske om å se en liste over elever som har registrert seg.

#### **Bildemodul**

Denne modulen håndterer forespørsler fra PDA om å ta bilde. I dette systemet brukes bildemodulen kun til å ta bilde til pressekort. Det skal brukes et webkamera til å ta bilder og dette kameraet skal fysisk være tilkoblet serveren. Bildemodulen skal bestå av løsning for å ta bilder tilpasset læringssimulatoren. I det endelige systemet til Innlandets Vitensenter skal også bildemodulen brukes til eksperimenter og innspilling av video med kommentarer.

### **3.2 Løsning på PDA**

Den sentrale delen av systemet sett med elevenes øyne, er PDA-en. Hver deltaker er knyttet til en unik PDA og denne brukes fra start til slutt i rollespillet. Klassediagram for løsning på PDA er plassert lenger bak i designkapittelet, se 3.5.1.

På PDA skal det kun være en modul. Denne modulen skal ha alt ansvar for å løse kravene til PDA-ene i Læringssimulatoren.

PDAmodule skal benyttes til registrering av fornavn og etternavn til elevene. Ut ifra disse opplysningene vil det bli dannet et pressekort sammen med et bilde. Videre vil denne modulen benyttes til å motta oppdragsinfo, foreta intervjuer og velge informasjon/påstander som skal sendes til redaksjonen.

De tre viktigste elementene i PDAmodule vil være:

- Brukergrensesnittet
- Tilkobling/forbindelsen med server
- Nettleser

### 3.3 Kommunikasjon

Kommunikasjon mellom klientene (PDA-ene) og server er et vesentlig moment i systemet.

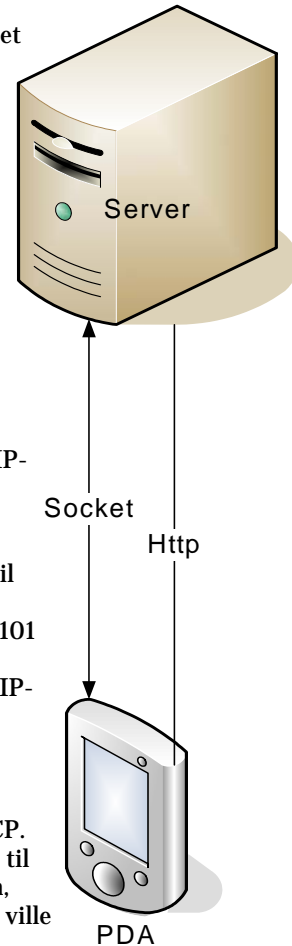
På figur 3.2 vises hvordan kommunikasjonen skal løses. Mesteparten av informasjonen PDA skal vise, skal hentes vha http og nettsider. I tillegg benyttes socket slik at server har mulighet for å bruke push. Det vil si at server skal kunne bestemme hva PDA-en skal foreta seg. Dette gjelder når server vil starte et intervju eller dirigere PDA-ene til en bestemt nettside.

#### 3.3.1 IP-adresser

En IP-adresse består av fire 3-sifrede tall. Formatet til en IP-adresse er 32-bits numerisk adresse, skrevet som fire tall separert med punktum. Tallene kan være fra 0 til 255. [2]

Alle enheter på nettverket i Læringssimulator skal ha det til felles at de tre første tallene som utgjør IP-adressen, er identiske. PDA-ene skal tildeles IP-adresser fra 192.168.0.101 og oppover. For eksempel vil PDA nr 15 i det fremtidige systemet få adresse lik 192.168.0.115. De tre siste sifrene i IP-adressen skal brukes som ID til hver enkelt PDA i læringssimulatoren.

Det ble tidlig besluttet å benytte faste IP-adresser for klientene og server i systemet. Alternativet ville vært DHCP. DHCP ble vurdert, men fordi behovet for å ha IP-adressen til hver enkelt PDA som en identifikator i læringssimulatoren, medførte at valget falt på å bruke faste IP-adresser. DHCP ville medført automatisk tildeling av IP-adresser.



**Figur 3.2 Kommunikasjon mellom server og klient**

#### 3.3.2 Socket

I systemet er det brukt socket til styringskommunikasjon mellom klient og server. En socket består av en nodeadresse, adresse til en enhet på nettverket, og et portnummer. Disse to faktorene identifiserer tjenesten på nettverket. For eksempel indikerer port nummer 80 på en Internett-node at det dreier seg om en Web-tjener. [2]

Hovedårsaken til å benytte socket ligger i muligheten til å sende kommandoer fra server til klientene, også kalt "Push". Socket brukes for eksempel hver gang server skal kommandere en PDA til en gitt nettside. Denne muligheten har man ikke via http-protokollen.

Http skal brukes til de formål der man skal hente informasjon og vise en side på PDA. I tillegg skal det nyttes http i administratormodulen til å presentere data fra datakilden.

### **3.3.3 Klientene (PDA)**

Et annet vesentlig punkt i løsningen, var krav fra oppdragsgiver om å gjøre klientene så tynne som mulig. Dette er løst ved å implementere en nettleser i programvaren på PDA-ene. Registreringen av bruker løses ved at nettleseren dirigeres til en nettside som ligger på serveren. Nettsidene som brukes i systemet er utviklet vha ASP.NET-teknologi.

Oppdragsgiver opererer med navnet "lubne klienter". Dette innebærer at klientene kan ha noe logikk/funksjonalitet, men hovedsakelig er klientene passive. Eksempel er funksjonalitet for å kunne spille av lydfiler og video lokal fra PDA-en

### **3.3.4 ASP.NET**

ASP står for Active Server Pages. ASP-filene i ASP.NET, får etternavnet .aspx. En ASPX-fil er en spesifikasjon for nettsider hvor web-serveren genererer HTML-siden i det brukeren åpner dokumentet. Når web-serveren ser filer med denne endelsen, vet den at filen skal prosesseres før den sendes. [2]

Bruk av ASP er en stor fordel når PDA-ene skal hente informasjon. I og med at PDA-ene ikke er i nærheten av en PC når det kommer til prosessorkraft og minne, vinner man mye på at serveren foretar brorparten av arbeidet.

### 3.4 Grafisk brukergrensesnitt

I dette delkapittelet presenteres skjermbildene i systemet og illustrasjoner skal vise hvordan de ulike modulene skal presenteres for brukeren. Det er lagt vekt på å lage skjermbildene så intuitive som mulig.

På bakgrunn av informasjonen fra oppdragsgiver, var fargevalget enkelt. Mye av handlingen i den endelige læringssimulatoren, slik den foreligger i dag, baserer seg på vann og Mjøsa. I alle systemmoduler som har interaksjon med brukere, var det derfor naturlig å velge en lys blåfarge.

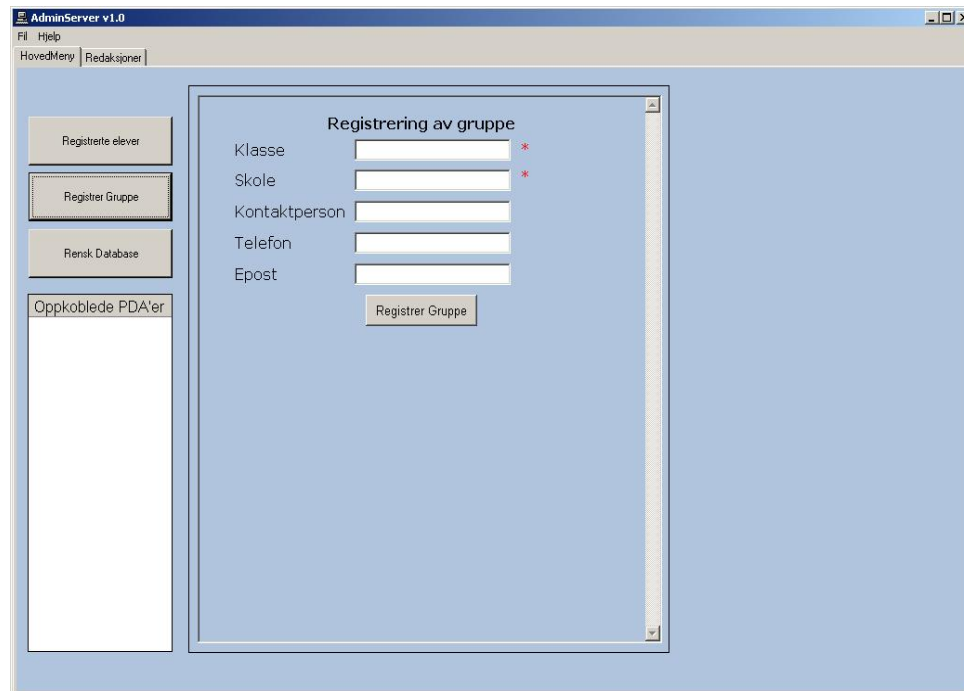
#### 3.4.1 Skjermbilder for administratormodulen

Administratormodulen er der hvor rollespilleren styrer rollespillet. Brukergrensesnittet består av to skilleark hvor all funksjonaliteten er plassert.

Skillearket med navnet "Meny" består av tre knapper med ulik funksjonalitet. Knappene inneholder funksjonalitet for å registrere gruppe, vise registrerte elever og renske variabel data fra datakilden.

Videre finnes en liste, plassert nede til venstre, hvor det vises PDA-er som er aktive på nettverket. De PDA-ene som vises i denne liste har vellykket opprettet en socketforbindelse med server.

Midt på skillearket finnes nettleseren. Her vises informasjon avhengig av hvilken knapp man trykker på.

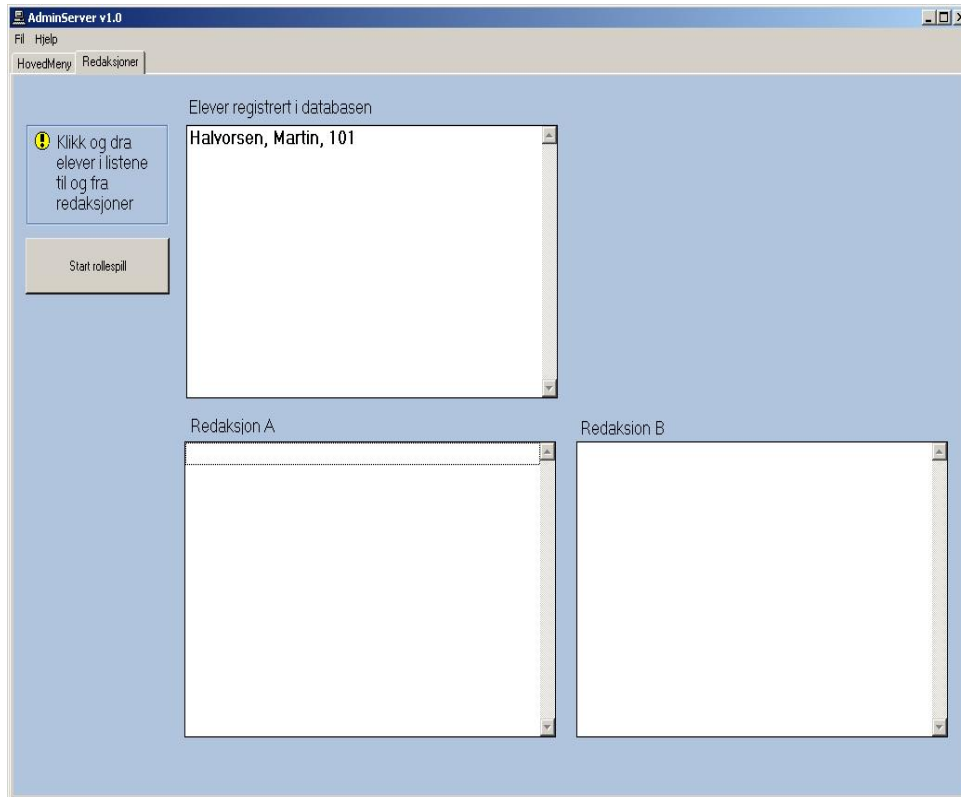


**Figur 3.3 Viser hvordan skillearket Meny er konstruert**

Skillearket med navnet "Redaksjoner" gir Rollespilleder mulighet til å fordele registrerte elever til de ulike redaksjonene.

På skillearket er det plassert tre lister. Den øverste listen oppdateres regelmessig og viser de elevene som vellykket har gjennomført registreringen. Nedenfor listen med elever som har registrert seg, finnes to andre lister. Disse listene representerer en redaksjon hver seg, hhv redaksjon A og B. Mellom de tre listene kan man klikke og dra elever til man er fornøyd med fordelingen.

Avslutningsvis starter man rollespillet ved å trykke på knappen "Start rollespill". Da vil de elevene som befinner seg i listen til redaksjon A og B knyttes til riktig gruppe og spillet kan begynne.



**Figur 3.4** Viser skjermbildet ved tildeling av redaksjoner



### 3.4.2 Skjermbildet for redaksjonsmodulen

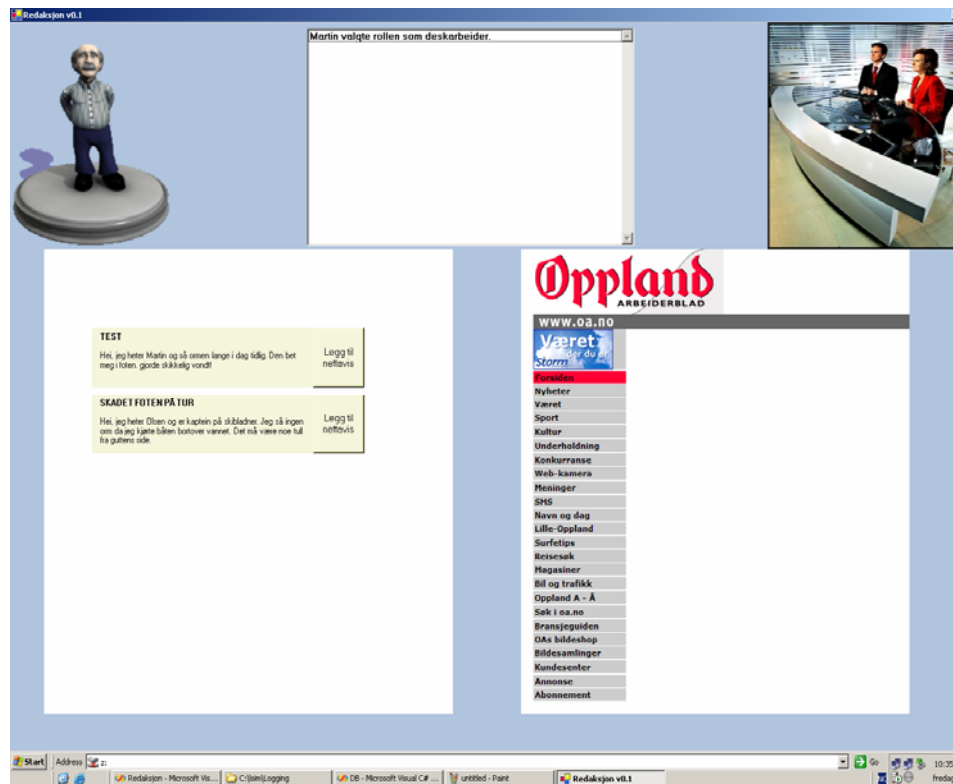
Skjermbildet viser hvordan den virtuelle redaksjonen ser ut. Her vil roller og oppdrag fordeles, og resultatet av arbeidet elevene har gjort vises. Brukeren i den virtuelle redaksjonen arbeider med valg av artikler på bakgrunn av opplysningene som journalistene sender inn til redaksjonen.

Konrad, hjelperen i læringssimulatoren, er plassert øverst til venstre. Herfra har Konrad god oversikt og følger nøye med på hva elevene foretar seg. Konrad kommer med tips og hint gjennom arbeidet med å få på plass nettavisen.

Til høyre for Konrad, finnes et informasjonsfelt. Her vil meldinger fra journalistene tilknyttet redaksjonen dukke opp.

Under Konrad, nederst til venstre på skjermbildet, finnes et område med artikler. Artiklene i dette panelet representerer mulige artikler for elevene å publisere. Til hver artikkel i dette området, er det tilknyttet en knapp. Trykker man på denne knappen, legges artikkelen til i nettavisen nederst til høyre.

Området nederst til høyre gir de som jobber i redaksjonen en forhåndsvisning av nettavisen.



Figur 3.5 Den virtuelle redaksjonen

### 3.4.3 Skjermbilder for PDAModulen

Hver PDA er utstyrt med en penn. Pennen brukes til å fysisk trykke på den trykkløssomme skjermen på PDA-en. Denne pennen fungerer stort sett slik en mus gjør på en vanlig PC. Nedenfor følger noen utvalgte skjermbilder fra PDA-en.

I denne situasjonen skal brukeren fylle inn fornavn, etternavn og velge gruppetilhørighet fra rullgardinlisten. Når brukeren trykker med pennen i tekstfeltet, vil det dukke opp et virtuelt tastatur. Dette tastaturet brukes til å skrive opplysninger.

Denne informasjonen blir så lagret i systemet ved å trykke på knappen "Registrer deg!".



**Figur 3.6 Registrering på PDA**



**Figur 3.7 Slik ser pressekortet ut ved endt registrering**

Neste sted i registreringen er å ta et bilde. Ved endt registrering vil knappen "Ta bilde" dukke opp.

Ved å trykke "Ta bilde", blir et bilde tatt 2 sekunder etter at knappen er trykket.

Brukeren kan ta et nytt bilde til han blir fornøyd. Da brukeren føler seg komfortabel med pressekortet, trykker han eller hun på knappen "→" avslutter registreringen.

Skjermbildet til høyre viser hvordan PDA-en ser ut når en journalistelev skal velge påstander.

Eleven kan huke av en eller flere av påstandene som er knyttet til oppdraget. Til sist trykker brukeren på knappen "Send inn" og de som jobber i redaksjonen mottar meldingene fra journalisten.



**Figur 3.8 Innsending av opplysninger**

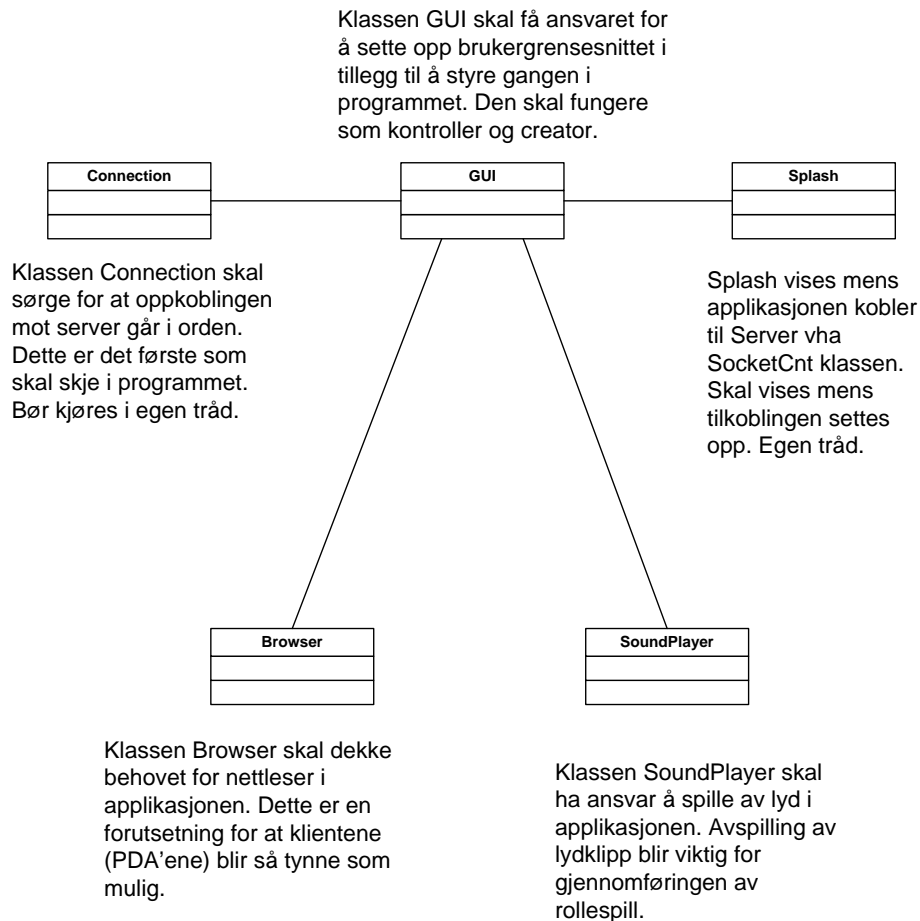
### 3.5 Klassediagram

Gruppen har laget klassediagram over modulene i læringssimulatoren. Disse diagrammene vil nok avvike noe fra den virkelige klassestrukturen som blir implementert. Gruppen regner med å måtte endre og legge til flere klasser når tiden er inne for implementasjon.

Design klassediagrammer og sekvensdiagrammer finnes i vedlegg C. Her er klasser og pakker presentert med attributter og metoder og ligner mer på implementasjon

#### 3.5.1 Klassediagram for PDAModulen

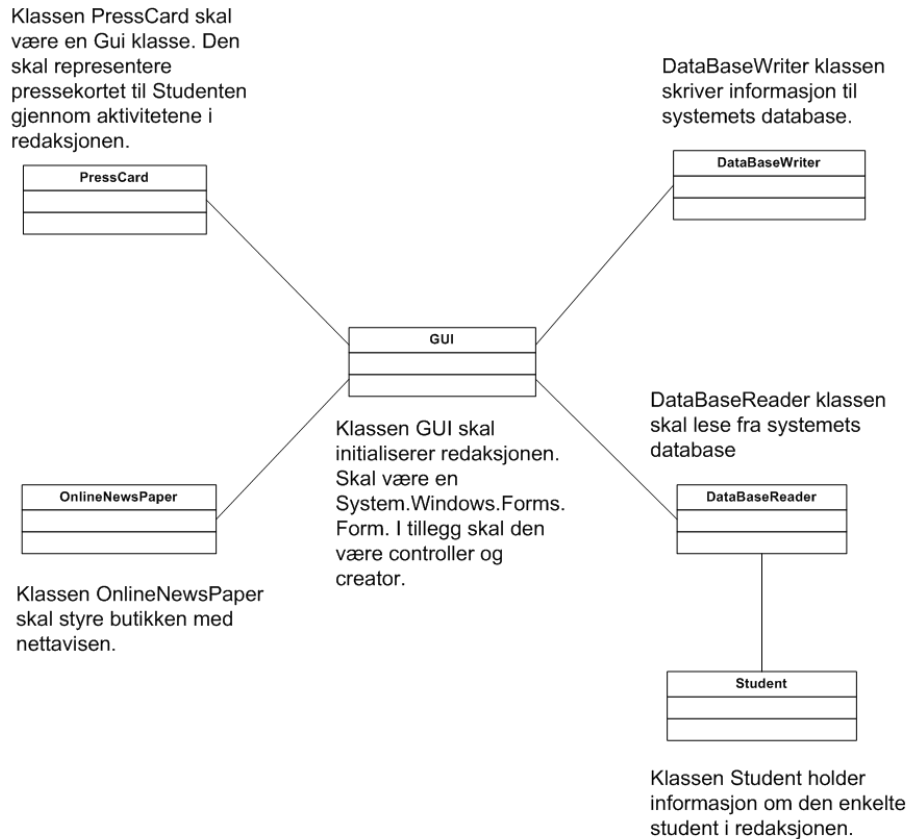
Under følger et klassediagram over PDAModulen i læringssimulatoren. På figur 3.9 ønsker vi kartlegge de viktigste klassene.



**Figur 3.9 Klassediagram over PDA software**

### 3.5.2 Klassediagram for redaksjonsmodulen

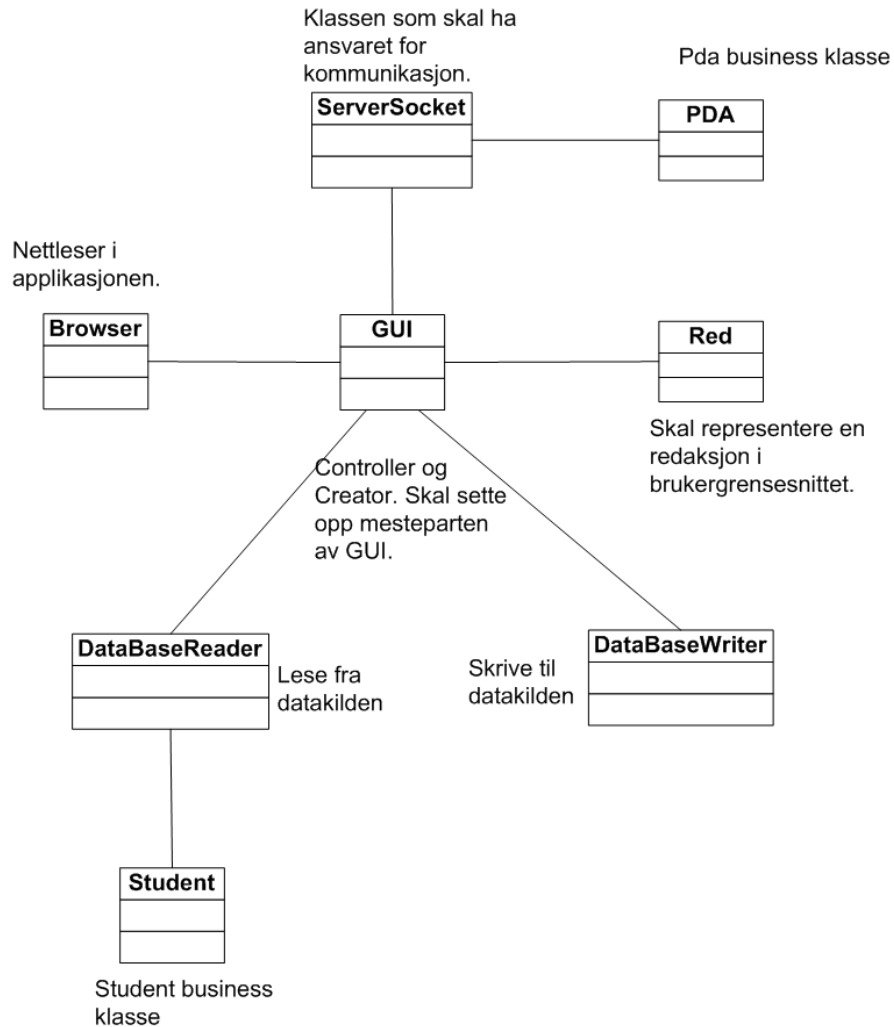
Under følger et klassediagram over redaksjonsmodulen i læringssimulatoren. På figur 3.10 er det illustrert hvordan redaksjon modulen er planlagt med klasser og ansvar.



**Figur 3.10 Klassediagram over redaksjonen**

### 3.5.3 Klassediagram for administratormodulen

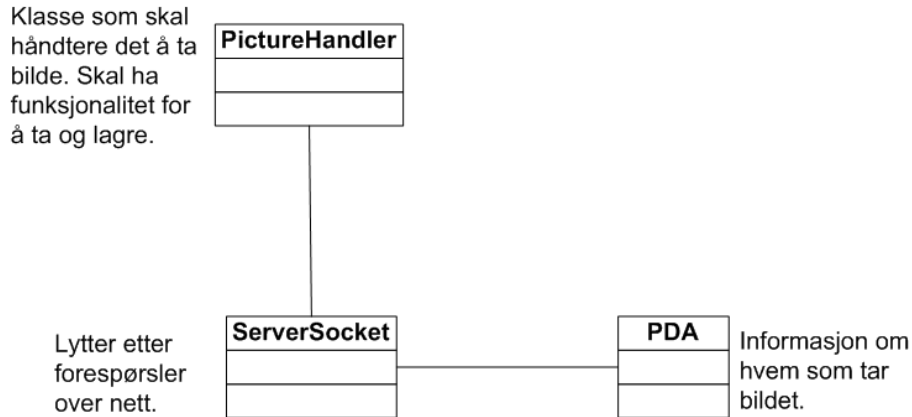
Under følger klassediagrammet over administratormodulen i læringssimulatoren. Med business klasse menes at det er en klasse som holder informasjon og som ikke utfører noe arbeid. På figur 3.11 er PDA tenkt som et slik business objekt. Klassen skal holde informasjon som har med den unike PDA å gjøre.



**Figur 3.11 Klassediagram over administratormodul**

### 3.5.4 Klassediagram for bildemodulen

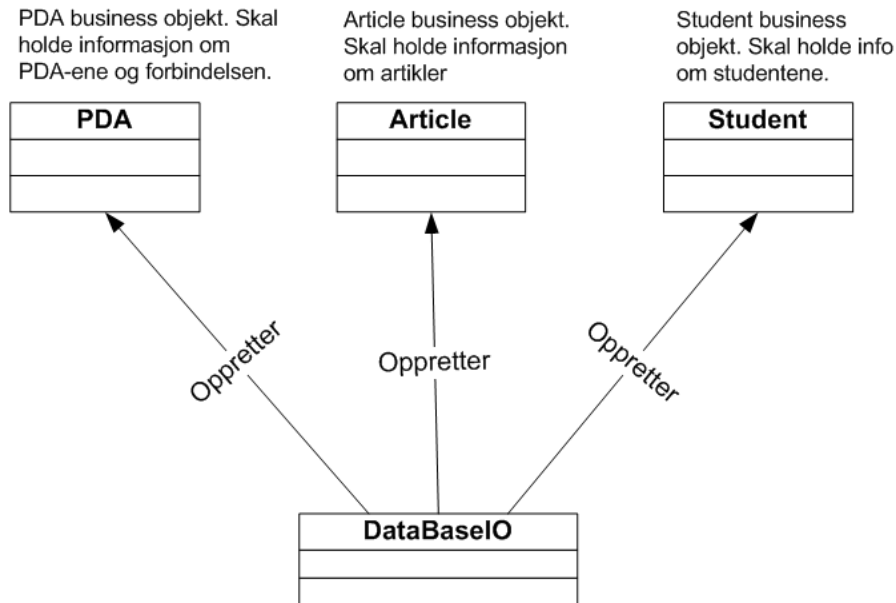
Det mest sentrale i denne modulen er   f  p  plass en klasse som tar seg av selve bildetakingen. Videre trengs en klasse for   ta imot foresp rsler fra PDA-er om   ta et bilde.



**Figur 3.12 Klassediagram over bildemodulen**

### 3.5.5 Klassediagram for pakken Data

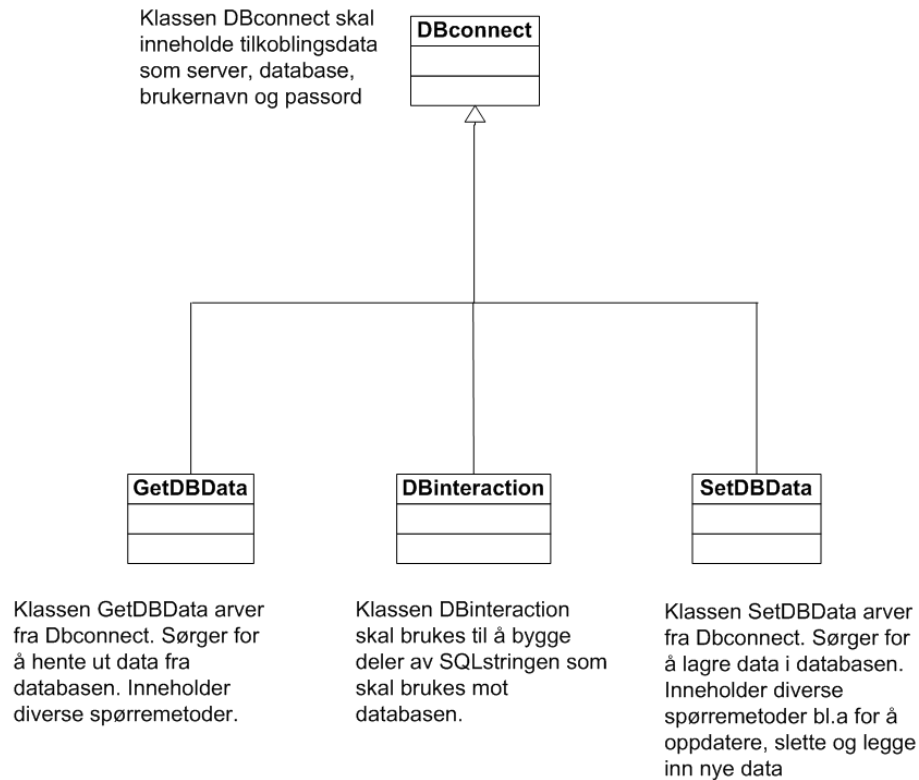
Pakken er planlagt   bli en del av applikasjonslaget p  serveren. I denne pakken er det skilt ut klasser med funksjonalitet flere av modulene p  serveren felles har brukt for. Applikasjonslaget skal v re et bindeledd mellom brukergrensesnittet og I/O-laget. Business objekt er ment som informasjonsekspert eller informasjonsholder.



**Figur 3.13 Klassediagram over pakken Data**

### 3.5.6 Klassediagram for pakken DB

Under følger klassediagrammet over pakken der det skal plasseres funksjonalitet til interaksjon med databasen. Pakken DB vil bli en del av I/O laget i systemet. Både redaksjonsmodulen, nettmodulen og administratormodulen har behov for denne funksjonaliteten og gruppen finner det derfor naturlig å skille dette ut i en egen pakke.

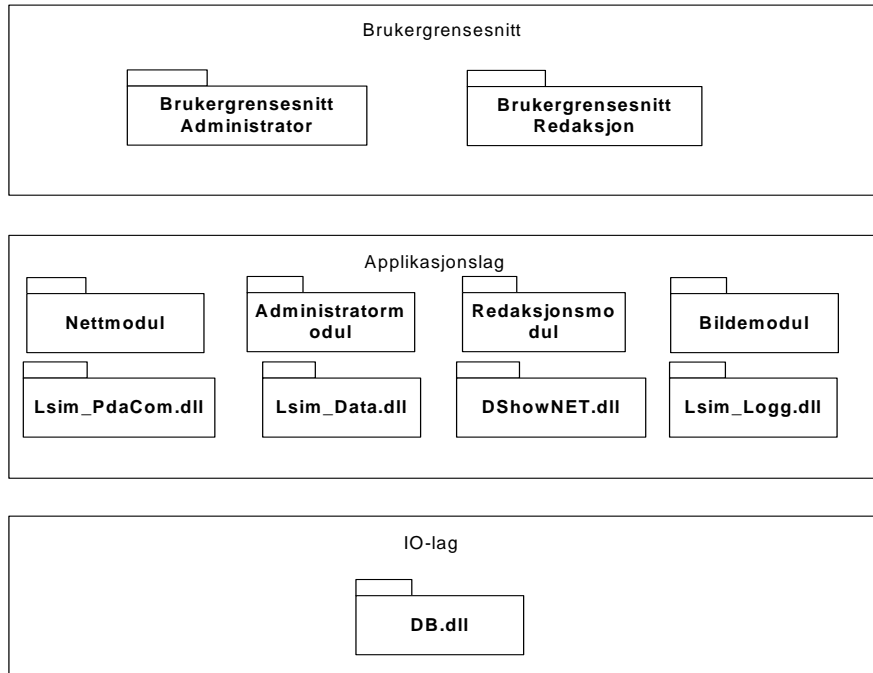


**Figur 3.14 Klassediagram over database pakken kalt DB**



### 3.6 Pakkediagram for løsningen på server

I dette diagrammet er det illustrert hvordan pakkene og modulene er lagdelt i systemet på server. De to modulene som skal ha direkte dialog med brukere, Administratormodul og redaksjonsmodul, skal skilles ut til brukergrensesnittlaget. De to andre modulene, nettmodulen og bildemodulen, vil brukerne kun benytte indirekte via andre moduler i systemet. Derfor skal disse plasseres i applikasjonslaget. I det nederste laget er det plassert en pakke som skal ha ansvaret for å ta imot forespørsler fra lag ovenfor om å snakke med databasen.



**Figur 3.15** Pakkene plassert i de lag de tilhører

### 3.7 Verktøy i designprosessen

Til tegning av diagrammer i designprosessen har prosjektgruppen benyttet Microsoft Visio. Visio er et tillegg til Microsoft Office. Verktøyet har mange ulike muligheter til å tegne diagrammer, alt fra byggetekniske tegninger til UML-diagrammer for systemutvikling.

Siden rapporten er skrevet i Microsoft Word, har det vært praktisk at Visio er en del av Office. Dersom man legger diagrammer fra Visio inn i Word-dokumentet, kan man enkelt oppdatere det ved å dobbeltklikke på diagrammet.

I kravspesifiseringen brukte gruppen Rational Rose til utvikling av Use Case-diagrammer og flytdiagrammer. I forhold til Visio, er Rational Rose tungvint. Grunnen til at vi brukte Rose til å begynne med skyldes at gruppen ikke hadde tilgang til Visio innledningsvis.

## **4 Implementasjon**

### **4.1 Utviklingsmiljø**

#### **4.1.1 Valg av utviklingsmiljø**

For prototypen til læringssimulatoren, bestemte oppdragsgiver at prosjektgruppen skulle utvikle systemet i .NET. Grunnen er at videreutviklingen av systemet skal utføres av oppdragsgiver. Det var derfor naturlig å tilpasse seg oppdragsgivers ønsker selv om ingen av medlemmene i gruppen hadde noe erfaring med utviklingsmiljøet. Det ble derfor en del av prosjektet å sette seg inn i Microsoft Visual Studio og .NET plattformen.

Innledningsvis ønsket prosjektgruppen å utvikle i VJ# .NET. Prosjektgruppen regnet med å dra nytte av erfaringer fra tidligere kurs i Java. VJ# .NET er nemlig et Java-tilpasset .NET utviklingsspråk. Utviklingen i VJ# skulle imidlertid vise seg å by på problemer. VJ# .NET er langt mindre utbredt enn alternativet VC# .NET. Dette gjelder spesielt tilgang til dokumentasjon og utbredelse av forum.

Til utviklingen av programvare på PDA med Windows operativsystem tilbyr Microsoft noe som heter .NET Compact Framework. Dette rammeverket er en begrenset versjon av hele .NET rammeverket. Utviklingsplattformen .Net Compact Framework støttet ikke VJ#. Her fantes ikke alternativer og vi ble nødt til å utvikle software til PDA i VC# .NET. Dermed besluttet prosjektgruppen å benytte VC# .NET både til utvikling av program for PDA og PC for å unngå konflikter.

#### **4.1.2 Egenskaper ved VC# .NET**

Felles for de aller fleste Microsoft produkter er at brukervennlighet har h y prioritet. Programvaren, MS Visual Studio, som nyttes til utvikling i dette prosjektet, er intet unntak. For eksempel er det lekende lett   lage brukergrensesnitt. Dette kan enkelt gj res ved   klikke og dra elementer fra en meny til det omr det du vil bygge p . I prinsippet virker det p  samme m te som et tegneverkt y. Man tegner alts  brukergrensesnittet og koden blir autogenerert. I motsetning til v r erfaring med Java, var det i dette tilfellet kj pt   sette opp GUI. Det finnes ogs  liknende verkt y for Java, men det er ikke noe som er innebygd fra leverand ren p  sammen m te.

Verkt yet for generering av GUI h res flott ut, noe det ogs  er i de fleste sammenhenger. Andre ganger kan riktignok designerverkt yet by p  irriterende problemer. La oss si at man har "tegnet" mesteparten av GUI-en som man  nsker vha Visual Studio sin designer. Deretter legger man til noen GUI-komponenter som man selv har kodet. I disse tilfeller er det viktig   holde orden p  koden og ikke blande egen kode inn i den autogenererte koden. Gruppen har smertelig erfart at den autogenererte koden helt p  egenh nd fjerner de hardkodete GUI-komponenter uten   gi beskjed. Dette fenomenet har v rt roten til noen fortvilte situasjoner.

P  sammen m te som Java, har VC# .NET verkt y for   generere nettsider av dokumenteringer av koden. I VC# .NET pakker den kommentarene til koden inn i XML. Fra menyen i Visual Studio kan man f  generert nettsider som da henter informasjon fra XML datakilden med kommentarer. Enkelt og veldig praktisk.

En sv rt nyttig egenskap ved .NET er at det er enkelt   implementere Windows komponenter inn i egne program. For eksempel har prosjektgruppen i dette prosjektet hentet Microsoft sin nettleser, Internett Explorer, inn i administratormodulen. Dette gj res ved at man henter en referanse fra operativsystemet inn i programvaren man jobber med.

#### **4.2 Databaseplattform**

Databasen kj rer p  en Windows SQL Server 2003 Enterprise Edition. Databasen er av typen MS SQL 2000 Standard Edition og prosjektgruppen valgte denne plattformen i samarbeid med oppdragsgiver. Databasen kj rer ogs  veldig bra sammen med operativsystemet da begge er produkter av Microsoft. Generelt er MS SQL en veldig stabil og robust database. Prosjektgruppen hadde ingen betenkeligheter med   bruke denne typen da evt. andre databaser ville v re mindre gode og koste penger. Oppdragsgiver hadde ogs  allerede lisenser p  MS SQL Server.

### 4.3 Kodestandard

Gruppen bestemte seg tidlig for   skrive koden med engelske variabelnavn. Dette ble bestemt fordi all koden ellers er engelsk, og prosjektgruppen f ler derfor det er naturlig   gi engelske variabelnavn. Dette er ogs  en fordel ved internasjonalisering.

N r det gjelder kommentering, er denne forfattet p  norsk. N r vitensenteret skal sette seg inn i v rt arbeid, fant vi det fornuftig   skrive kommentarene p  norsk. I ettertid ser vi at vi like gjerne kunne ha valgt   skrive b de variable og metodenavn p  norsk for dokumenteringens del. Nettsidene med dokumentasjon best r nemlig av engelske variabel – og metodenavn, mens forklaringen er skrevet p  norsk.

Dersom man plasserer tre slash-er, ///, over enheten man  nsker   kommentere i Visual Studio, genereres et omr de hvor man skal fylle inn kommentarer. Det autogenererte omr det varierer siden genereringen tar hensyn til om for eksempel metoden har parametere og/eller returverdier.

I v r kode har gruppen benyttet denne form for kommentering p  klasser, konstruktorer, metoder og variable. Eksempel p  kommentering for en metode kan se ut som f lger:

```
/// <summary>
/// Returnerer status for
/// socketforbindelse med server.
/// </summary>
/// <returns>
/// TRUE dersom tilkoblet, FALSE dersom ikke tilkoblet
/// </returns>
public override bool getConnectionStatus()
{
    return pdaIsConnected;
}
```

Visual Studio er hj lpesom i forbindelse med struktur p  koden. Korrekt innrykk, avstander og oppsett for kommentering autogeneres. Dette er sv rt nyttig og gir den som skal lese eller jobbe med koden god oversikt. Visual Studio er ogs  hj lpesom ved autofullf ring av kode, det vil si at programmet kommer opp med forslag til fullf ring av det man skriver. Dersom man skriver et klassenavn og avslutter med tegnet ”.” kommer det automatisk opp en liste med innhold i denne klassen. Det er for eksempel kj pt   f  oversikt over metoder og events som finnes i en klasse.

### 4.3.1 Eksempel p  kode

Under f lger kode hentet fra programvaren p  PDA. Kildekoden er hentet fra klassen Browser.cs som finnes i topppakken i PDAModulen.

```
using System;
using OpenNETCF.Windows.Forms;
using OpenNETCF;
using System.Windows.Forms;
using System.Drawing;

namespace Pda
{
    /// <summary>
    /// Klassen Browser arver fra OpenNETCF sin WebBrowser.
    /// Klassen OpenNETCF.Windows.Forms.WebBrowser inneholder en
    /// rekke metoder og events som er greie   benytte seg av.
    /// </summary>
    public class Browser : OpenNETCF.Windows.Forms.WebBrowser
    {
        public Browser()
        {
            // Setter innstillinger for browseren
            this.BackColor = System.Drawing.Color.LightSkyBlue;
            this.BorderStyle = System.Windows.Forms.BorderStyle.None;
            this.EnableClearType = true;
            this.ForeColor = System.Drawing.SystemColors.ActiveCaption;
            this.Location = new System.Drawing.Point(0, 0);
            this.Size = new System.Drawing.Size(240, 224);
            this.Text = "webBrowser";
            this.Url = null;
            this.DocumentCompleted += new
                webBrowserDocumentCompletedEventHandler(Browser_DocumentCompleted);
        }

        /// <summary>
        /// Metoden navigerer webBrowseren til en angitt adresse
        /// </summary>
        /// <param name="url">Adresse browseren skal lete opp</param>
        public void goToPage(String url)
        {
            this.Navigate(url);
        }

        /// <summary>
        /// Metode som setter cursor til normal hver gang en side er hentet
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void Browser_DocumentCompleted(object sender,
            WebBrowserDocumentCompletedEventArgs e)
        {
            Cursor.Current = Cursors.Default;
        }
    }
}
```

**Figur 4.1 Eksempel p  kode**

## 4.4 Koding av ulike komponenter i systemet

Mangel på erfaring med programmeringsspråket gruppen har benyttet, har gjort at gruppen har brukt mye tid på å prøving og feiling. Prosjektgruppen har implementert løsninger som etter en tid har blitt erstattet av bedre løsninger. Tempo i programmeringen til å begynne med var et problem siden samtlige i prosjektgruppen var ukjent med utviklingsmiljøet både for PDA og PC. Vi hadde riktignok satt av noe tid til å lære oss utviklingsmiljøet i fremdriftsplanen, men man blir ikke utlært i løpet av 14 dager.

Det har vært en overgang å tilpasse seg Windows plattformen. Spesielt med tanke på klassebiblioteker, dll-filer, og hvordan disse inngår i Microsoft Windows baserte programmer.

### 4.4.1 Koding av Administratormodulen

Programmet som utgjør administratormodulen har gruppen gitt navnet AdminServer. Dette programmet er en sentral del av prosjektgruppens system. Hovedoppgaven til applikasjonen er å sette opp kommunikasjon med PDA-ene på nettverket og sikre korrekt registrering. I tillegg til å være et styringsverktøy, starter AdminServer prosessene som kjører bildemodulen og redaksjonene.

Administratorprogrammet var det første gruppen startet å utvikle. Tidlig ønsket vi å få på plass en form for kommunikasjon mellom PDA-ene og serveren. På bakgrunn av erfaring med socketprogrammering i Java, besluttet gruppen å løse kommunikasjonen vha socket. Socketforbindelsen dekket også behovet for å sende kommandoer fra server til klientene. Gruppen utviklet dermed en serverapplikasjon for PC og en klientapplikasjon på PDA. På denne måten fikk vi testet om våre ideer virkelig lot seg gjennomføre.

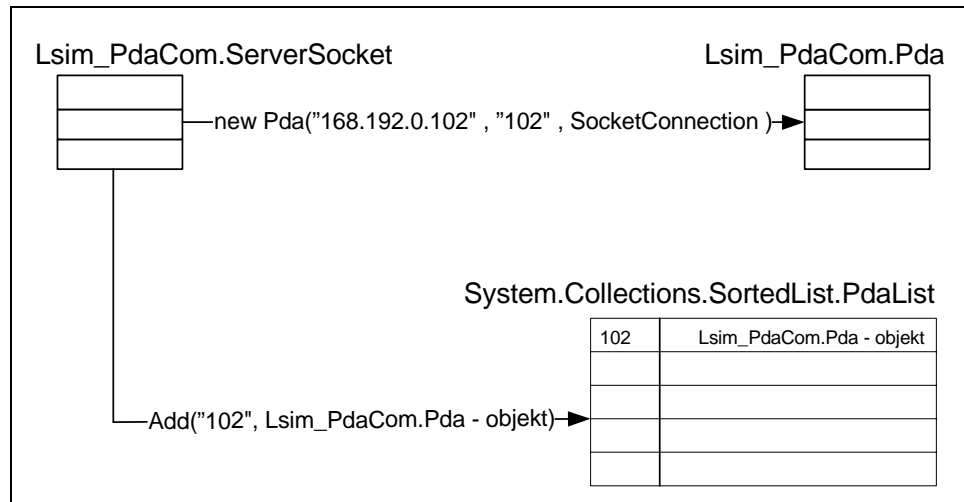
Det finnes mange ulike måter å implementere socketforbindelsen i .NET. Nettopp av den grunn gikk det mye tid til å finne en måte som virket. Det begrensede rammeverket for PDA, .NET Compact Framework, har ikke de samme mulighetene som det fulle rammeverket. Det oppsto derfor en del problemer da disse to rammeverkene skulle snakke sammen.

Gruppen fant etter hvert en løsning som tilfredstilte den funksjonaliteten vi hadde sett for oss. Til socketforbindelsen mellom klient og server har gruppen brukt en klasse som heter System.Net.Socket.TcpListener. Denne brukes ved at man oppgir porten serveren lytter på og den lokale IP-adressen til serveren. Denne lyttingen er implementert som en egen tråd, og lytter gjennom hele registreringsfasen.

Når forbindelsen er oppe, brukes klassene System.Net.Sockets.NetworkStream til å jobbe mot datastrømmen over forbindelsen. På denne nettverksstrømmen brukes både System.IO.StreamWriter og System.IO.StreamReader for å lese og skrive over socketforbindelsen.

Hver gang når serveren mottar en tilkobling, oppretter serverprogrammet et objekt av PDA informasjonseksperterklassen. En informasjonseksperter er en klasse som holder informasjon og som selv gjør minimalt med databehandling. Dette objektet lagrer informasjon om PDA-en og socketforbindelsen og alle objektene legges i en System.Collections.SortedList. En slik liste fungerer etter hashmap prinsippet. Det vil si at hvert element i listen består av en unik nøkkel og en verdi. Listen sorteres etter nøkkelen.

PDA-ens ID brukes som verdiens nøkkel, og verdien i dette tilfellet er PDA objektet som inneholder all nødvendig informasjon om PDA-en. På denne måten kan man i programmet hente ut for eksempel PDA med ID 105 fra listen og kjøre kommandoer til denne enheten over dens forbindelse med server. Løsningen er skissert i figur 4.2.



**Figur 4.2 Håndtering av tilkobling fra PDA**

Til å sjekke om forbindelsen mellom PDA og server er oppe, sjekker server om den får svar fra PDA ved å sende "PING" til alle klientene. PDA-ene svarer med "PONG" dersom de er på nett. Dersom server ikke mottar "PONG", fjernes den aktuelle PDA fra listen over aktive PDA på nettverket. Denne kontrollen gjentas hvert 5. sekund. Grunnen til at vi valgte 5 sekunder er at vi ønsket at listen over påloggede PDA-er skulle være så oppdatert som mulig.

Annen funksjonalitet i AdminServer dreier seg om informasjon til Rollespilleder. Brukeren av programmet får informasjon om hvem og hvor mange av deltakerne som har greid å registrere seg. Prosjektgruppen har implementert en nettleser i applikasjonen AdminServer. Ideen bak dette skyldes krav fra oppdragsgiver om å gjøre programvaren så nettbasert som mulig. Så når rollespilleder ønsker en oversikt over hvem som har registrert seg, skjer dette ved at man klikker på en knapp som navigerer nettleseren til en ASPX side på nettverket som fysisk kommuniserer med databasen og presenterer data derfra. Til oppgradering av systemet i fremtiden vil det være enklere å vedlikeholde nettsidene som applikasjonen leser, enn å videreutvikle selve programmet.

Gruppen har i programmet valgt å implementere to redaksjoner. Dette er implementert kun for testing siden gruppen ikke har hatt teknisk utstyr til rådighet til å kjøre mer enn én redaksjon.

Ved å klikke og dra, kan rollespilleder knytte registrerte elever til redaksjoner. Når rollespilleder starter rollespillet, skrives informasjon til databasen vha klassen `DataBaseIO` som befinner seg i pakken `Lsim_Data`. Etter at skrivingen er fullført, starter så AdminServer nye prosesser for hver redaksjon som er benyttet. For å kunne starte en redaksjon, kreves det at det er knyttet minst to elever til redaksjonen. Dersom dette er gjort på en korrekt måte, kan rollespillet begynne.

#### 4.4.2 Koding av PDAModulen

Det var knyttet mange usikkerhetsmomenter med programmering mot håndholdte enheter. Ikke nok med at utviklingsmiljøet var nytt, sto gruppen ovenfor programmering mot en helt ny type hardware.

Siden det tok noe tid før gruppen fikk tildelt utstyr fra oppdragsgiver, ble vi nødt til å lære oss utviklingen for PDA vha Visual Studio sin emulator for lomme PC-er. Dette har stort sett fungert utmerket. Emulatoren var svært nyttig for prosjektgruppen, selv om vi har erfart at det er stor forskjell på emulatoren og den ekte varen når det kommer til stabilitet. Emulatoren kjører som et program på PC-en og låner prosessorkraft fra dens CPU. I og med at PC-er er langt mer stabile enheter enn PDA-er, gjør dette at emulatoren oppfører seg langt mer stabilt enn PDA-en, som til stadighet låser seg.

En bakdel med emulatoren er at den er langt dårligere i testsammenheng. Vi fikk ikke testet trådløs kommunikasjon og rekkevidde før vi fikk låne det tekniske utstyret av oppdragsgiver. Allikevel har vi mye å takke emulatoren for i opplæringsfasen og emulatoren kan anbefales som testverktøy.

Etter krav fra oppdragsgiver skulle programvaren på PDA, altså klientene, være så tynne som mulig. Det var et poeng å belaste den svake prosessoren på PDA-ene minst mulig. Av den grunn har gruppen implementert en nettleser i applikasjonen. Denne har gruppen hentet fra OpenNETCF sitt åpne bibliotek. I det begrensede rammeverket for PDA-ene, .NET Compact Framework, fant vi ingen muligheter for å implementere en nettleser direkte. Valget falt på OpenNETCF sin WebBrowser. En annen grunn til at vi valgte å implementere en nettleser er at det var nødvendig for å oppfylle de arkitekturelle kravene fra oppdragsgiver.

En annen funksjonalitet på PDA-ene er lydavspilling. En viktig del av rollespillet er intervjuer med personer for å innhente informasjon. Intervjuene fungerer som monologer og er lydfiler som spilles i den rette sammenhengen. Til dette formål er det implementert et P/Invoke bibliotek der vi har plukket ut det PDA-ene trenger for å spille av lyd.

Det første som skjer når programvaren kjøres på PDA, er at den forsøker å koble seg til serveren. Dette gjøres ved hjelp av klassen System.Net.Socket.TcpClient. Tilkoblingen skjer som kjent via det trådløse nettverket for systemet som baserer seg på IEEE sin standard, 802.11b. På hver PDA finnes et innebygd trådløst nettverkskort. Innstillingene på nettverket settes opp på forhånd og IP-adressene til hver enkelt PDA er statiske.

Dersom oppkoblingen er vellykket, startes resten av programmet. Innledningsvis gjøres brukergrensesnittet klart til registrering. I bakgrunnen vil en egen tråd til enhver tid lytte etter kommandoen fra server. Stort sett vil all aktivitet på PDA-en basere seg på ASPX-nettsider som hentes frem av nettleseren. Derfor er styringen hvilke sider PDA-en skal vise som et viktig moment.

For å ta bilde til pressekort, må brukeren av PDA-en ha registrert seg. Når det er registrert på serveren at en PDA har registrert seg, sendes en kommando til PDA som gjør at brukeren får mulighet til å ta bilde. Brukeren får samtidig instruksjoner om hvordan dette skal foregå. Da registreringen er over, får brukeren beskjed om å møte i sin redaksjon. Herfra vil funksjonaliteten til PDA-en variere ut fra hvilken rolle brukeren velger i den virtuelle redaksjonen.



#### 4.4.3 Koding av Bildemodulen

Applikasjonen som skal ta seg av knipsing av bilder i l ringssimulatoren, har f tt navnet PictureGrabber. Tjenesten er implementert som en Windows Service og bruker et webkamera til   ta bilder med. En Windows service er en tjeneste som kj rer i Windows. Tjenesten er et eget program som st r alene eller brukes av andre programmer. Disse ligger i bakgrunnssystemet og kan startes og avsluttes av andre programmer som kjenner til dens navn. Dette er en vanlig m te   implementere tjenester p  i Windows. PictureGrabber startes og stoppes automatisk via AdminServer.

Gruppen har gjenbrukt kode i denne sammenhengen for   sikre fremgangen i arbeidet med systemet. Det ville tatt for mye tid dersom vi p  egenh nd skulle utviklet klasser og funksjoner for interaksjon med webkamera. Prosjektgruppen fant et  pent program funnet p  nettet kalt DirectShow .NET [3] som bruker DirectX 8.1. Programmet hadde den funksjonaliteten gruppen hadde behov for og mer til og er derfor tilpasset bruk i dette prosjektet.

I systemet prosjektgruppen har utviklet, er det kun PDA-ene som tar bilder. P  lignende m te som AdminServer, har PictureGrabber en socket som lytter etter foresp rsel om   ta bilde fra PDA-er. Prosjektgruppen har nok en gang valgt   l se lyttingen ved hjelp av System.Net.Socket.TcpListener klassen. Siden tjenesten ogs  kj rer p  server, har begge de to lytterne naturlig nok den samme IP-adressen. Det som skiller dem er porten.

N r en PDA  nsker   ta et bilde, kobler den seg til tjenesten ved   bruke en System.Net.Socket.TcpClient og bruke portnummeret til bildetjenesten og IP-adressen til serveren. N r dette gj res, tas bildet og det lagres til disk i JPG-format med bredde 198 og h yde 162. Det er valgt sm  bilder slik at de passer skjermen p  PDA-ene. Navnet p  filen blir presspic\_ID.jpg der "ID" er ID-en til PDA-en som tar bildet.

Bildene brukes i systemet b de til   presentere pressekortet p  PDA-ene og til valg av roller i redaksjonen. Bildemodulen er laget med tanke p  det endelige systemet. Det vil v re en enkel sak   utvide bruken av den.

#### 4.4.4 Koding av Nettmodulen

Prosjektgruppen har utviklet en del enkeltsider i webprogrammeringsspr ket ASP.NET og standard HTML. Sidene er nettsider som inneholder tekst og kode som gj r at man f r presentert informasjonen man  nsker til brukerne av systemet p  en enkel m te. Sidene av administratormodulen og PDA-modulen. N  de to modulene har bruk for   hente fram en spesifikk side, sendes en foresp rsel til webserveren. For   vise sidene riktig m  det brukes en nettleser som henter fram nettsidene. Dette er implementert i modulene som har bruk for dette. Alle nettsidene ligger p  serveren og kan nyttes av modulene via nettverket de er koblet opp mot.

Det skiller mellom to forskjellige sider i dette prosjektet, ASP.NET utviklede sider og rene HTML sider. HTML sidene brukes der det ikke trengs noe prosessering fra serverens side. Disse sidene er lik for alle brukerne og det trengs ikke noe validering eller prosessering.

I ASP.NET f r man laget nettsider som krever validering og prosessering uten at klienter trenger noen form for st tte for dette. I tillegg kan man hente ut data fra ulike datakilder og f  tilgang til en rekke ulike funksjonaliteter p  serveren. All prosessering skjer p  serveren, og p  denne m ten flytter man arbeidet bort fra enhetene med mindre prosessorkraft til serveren. Spesielt p  PDA-modulen er dette nyttig da en PDA har liten prosesseringskraft og batterikapasitet. Hvis det trengs   endre funksjonalitet trenger man bare   gj re dette p  nettsidene.

Nettmodulen brukes for eksempel i registreringen. Denne flyten fungerer på følgende måte:

PDAModulen ber nettleseren om å hente fram registreringssiden på server. Webserveren mottar forespørselen og starter eksekveringen av siden. Webserveren leser igjennom all kode hvor nettmodulen oppdager at den må koble seg til en SQL-database for å hente ut alle tilgjengelige grupper. Tilkobling til databasen skjer via pakken System.Data.SqlClient. Gruppene vises så i en nedtrekksliste. Resten av siden med tekstfelt og knapper genereres og hele siden vises til eleven.

Når eleven trykker på registreringsknappen så vil informasjonen i tekstfeltene først valideres. Deretter sjekker nettmodulen om eleven virkelig har tastet inn fornavn, etternavn og valgt en av gruppene fra nedtrekkslisten. Hvis noe av denne informasjonen mangler, vises informasjonen på nytt sammen med en feilmelding som forteller hva som mangler.

Når alt er i orden kobler nettsiden seg opp mot SQL-databasen og lagrer informasjonen som eleven tastet. Dersom lagringen går greit vises en tekst om at registreringen er vellykket. Dersom det oppsto en feil vil eleven motta en feilmelding.

#### **4.4.5 Koding av Redaksjonsmodulen**

Redaksjonsprogramvaren er viktig når selve rollespillet starter. Hovedoppgaven til redaksjonen er å la elever sette sammen en nettavis.

Prosjektgruppen hadde to alternativer til løsning for hvordan nettavisen kunne implementeres. Dette var å integrere nettleser i redaksjonsapplikasjonen og la den hente fram ASP.NET sider, eller å integrere nettavisen inn i selve applikasjonen. Gruppen gikk for å integrere nettavisen inn i selve applikasjonen. Se kapittel 5.6.3 for bakgrunn for valg av redaksjons som applikasjon.

Det første som ble utviklet i redaksjonen var at elevene som var tilknyttet en gitt redaksjon fikk presentert sitt pressekortet på skjermen. Pressekortet består av et bilde og navnet til eleven. Dataene for å generere pressekortet blir hentet fra databasen. Etter hvert laget gruppen en løsning slik at elevene kunne få tildelt roller ved å trykke på pressekortet sitt. Dette ble løst ved at det over pressekortene står en tekst for hvilken rolle som nå skal velges, og ber eleven som ønsker denne rollen om å trykke på pressekortet. Når pressekortet blir trykket på forsvinner kortet slik at det ikke er mulig å velge flere roller pr elev.

Når alle rollene er fordelt starter selve nettavisen opp i redaksjonen. Dette er løst ved å dele skjermbildet opp i tre deler, en topp og resten av programmet under toppen er delt i to, høyre og venstre side. Nettavisen inneholder bl.a. en tråd som sjekker hvert 5. sekund om det er registrert noen nye påstander i databasen. Disse blir så skrevet ut til en liste som vises på toppen av skjermbildet. Det er lagt inn slik at kun nye og ikke tidligere viste påstander legges til denne listen som er synlig på skjermen.

På venstre del av hovedskjermbildet vil det bli vist alle de artiklene som elevene kan velge mellom å publisere til nettavisen. Disse blir lest inn fra databasen og legges inn som informasjonseksperter. Objektene blir så lagt til en SortedList. Programmet går igjennom alle objektene og oppretter label av dem, en hovedlabel, en tittel-label og en label for selve det tekstlige innholdet. De to sistnevnte labelene legges på hovedlabelen. Hovedlabelene plasseres så på venstre del av skjermbildet. Hver av disse artiklene har en knapp tilknyttet seg slik at man kan legge artikkelen til i selve nettavisen. Når dette skjer legges artikkel ID-en i en SortedList til alle publiserte artikler og nettavisen tegnes opp på nytt ut i fra denne listen.

Artiklene i nettavisen plasseres på labeler etter samme prinsipp som ovenfor, bare at det er tre andre knapper som legges på hovedlabelen, flytt opp, flytt ned og fjern fra nettavis. Ut fra antallet artikler i nettavisen og dens plassering avgjør om hvilken knapp hver av de artiklene skal ha. Slik som siste artikkel skal ikke ha mulighet for å flyttes ned og første artikkel skal ikke kunne flyttes opp.

Redaksjonene startes opp når rollespillet startes fra Administratormodulen. Det opprettes en ny prosess for hver redaksjon og ID-en til redaksjonen sendes med som parameter. På denne måten vet redaksjonen hvilket nummer den er i rollespillet.

#### 4.4.6 Logging av hendelser i systemet

Prosjektgruppen har utviklet en egen loggerpakke kalt Lsim\_Logg. Denne pakken består av en klasse som tar seg av å skrive statusmeldinger, vanlige feilmeldinger og større programfeilmeldinger som Exceptions. Loggerpakken inkluderes i alle andre pakker og applikasjoner som har bruk for å logge meldinger.

Det finnes allerede et godt logger-system i .NET. Siden dette var et nokså ufarlig prosjekt, besluttet prosjektgruppen å lage en egen logger som en del av det å lære seg .NET.

Når man oppretter en instans av logg-klassen sendes det med en parameter som angir navnet på loggfilen. Dette har vi gjort for å kunne gjøre enkelttester av klasser og lagre disse på egne filer. Ellers har prosjektgruppen lagt opp til at det skal logges til en fil for hver kjørbare applikasjon. Det vil si at pakker som inngår i en applikasjon logges alle til applikasjonens loggfil. Pakker som inngår i flere applikasjoner, slik som DB.dll, logger meldinger til en egen fil.

Eks på hvordan man implementerer logging:

```

/// <summary>
/// Oppretter logger objekt
/// </summary>
protected Logger logg = new Logger("Database");

/// <summary>
/// Metoden tar seg av selve eksekveringen mot datakilden, og åpner tilkoblingen. Initierer Reader
også.
/// </summary>
/// <returns>true eller false</returns>
private bool ExecuteStatement()
{
    try
    {
        // Initierer tilkobling mot datakilden
        myCommand = new SqlCommand(sql, connection);
        if(connection.State.ToString() == "Closed")
            connection.Open(); // Åpner tilkobling mot databasen

        myCommand.ExecuteNonQuery();

        return true;
    }
    catch(Exception e)
    {
        // Logger Exception feilmelding
        logg.exceptionMsg("Klarer ikke å kjøre sql kommando mot databasetilkoblingen", e);
        return false;
    }
}

```

Eksempel på utskrift fra loggeren:

```
// Logg av Exception uten egendefinert feilmeldingsforklaring
12.05.2005 13:56:20      Exception Message : Object reference not set to an instance of an object.
                        at AdminServer.Gui..ctor() in c:\lsim\adminserver\adminserver\gui.cs:line 135

// Logg av Exception med egendefinert feilmeldingsforklaring
12.05.2005 13:58:37      Exception Message : Object reference not set to an instance of an object.
                        Feil i initialiseringen av AdminServer
                        at AdminServer.Gui..ctor() in c:\lsim\adminserver\adminserver\gui.cs:line 135
```

## 4.5 Utvikling av Databasen

Grappa startet tidlig i prosjektperioden med å lage en enkel database som skulle dekke de viktigste elementene i systemet. Til å begynne med var registreringsprosessen det viktigste. Fra begynnelsen inneholdt databasen kun grunnleggende tabeller som gjorde at grappa fikk testet modulene. Etter hvert som flere moduler kom på plass måtte også databasen utvides. Grappa endte opp med en databasemodell som ligger som vedlegg H i rapporten. Med veiledning fra lærere fikk vi opp en modell som fungerer tilfredsstillende til vårt bruk.

I første del av prosjektperioden var formålet med databasen å lagre fornavn og etternavn for å lage en profil av hver og en elev. Senere fikk vi behov for å ta vare på informasjon for grupper, artikler, holde informasjon om redaksjoner, storyer, intervjuobjekter osv.

Grappa har valgt å navngi tabellene med korte og konsise navn. Databasemodellens tabeller kunne blitt navngitt ved hjelp av navnekonvensjon. Dette gjøres med et prefiks, slik at alle tabeller som hører til samme database starter med samme prefiks. Det er også mulig å bruke flere prefikser hvis tabellene i databasen er organisert etter et hierarki. Siden databasen er så liten som den er og ingen klare hierarkier, har grappa valgt å se bort i fra navnekonvensjon.

## 4.6 Gjenbruk av kode i prosjektet

Noe av fordelen med .NET rammeverket er nettopp gjenbruk. Man kan på svært enkelt vis hente funksjonalitet som ligger i Windows og integrere denne inn i sin egen applikasjon. På denne måten får man raskt på plass masse funksjonalitet og kan tilpasse et program til sitt eget behov. Innen gjenbruk av kode er også objekt orientert programmering en viktig faktor.

Flere steder i prosjektet har det vært naturlig for oss å se etter ferdige løsninger for komponenter i det totale systemet. For gruppen handlet det om å få ferdig ett system i henhold til avtale med oppdragsgiver. Av den grunn følte vi det var nødvendig for fremgangen i arbeidet å benytte seg av enkelte åpne kilder (Open Source). Gjenbruk av kode i forbindelse med den fysiske forbindelse med webkamera, var noe prosjektgruppen så for seg tidlig i prosjektets fase. Det er som kjent ikke nødvendig å finne opp kruttet på nytt, hvilket også gjelder i programmeringssammenheng.

Gruppen har benyttet seg av et åpent bibliotek for interaksjon med webkamera. Programmet gruppen har hentet funksjonalitet fra et program kalt DirectShow .NET. Dette er et program for bruk av webkamera hvor man kan lagre filmsnutter og ta bilder vha et webkamera. Pakken inneholder mye mer enn hva vi hadde brukt for i dette prosjektet, dermed valgte vi å strippe ned koden en del. Vi lagde derfor en egen modifisert utgave av DirectShow med den funksjonaliteten vi trengte.

Den funksjonaliteten gruppen har gjenbrukt er i korte trekk oppkobling mot kamera, sjekk av korrekt versjon av DirectX og ikke minst det å ta bilde fra videostrømmen til webkameraet. Funksjonaliteten har vi brukt sammen med vår egen kode og plassert det inn i en Windows Service kalt PictureGrabber.

Til å utvikle programvaren på PDA har vi også gjenbrukt noe kode. Dette gjelder spesielt til avspilling av lyd. Her har vi benyttet oss av lydavspilleren som finnes i OpenNETCF, se <http://www.opennetcf.com>. OpenNETCF-biblioteket er omfattende og inneholder åpne kilder til mange formål. NETCF står for .NET Compact Framework og er ment som en tilvekst til den originale .NET Compact Framework som har svært begrenset funksjonalitet. Det er også fra samme bibliotek vi har hentet nettleseren som er integrert som en del av programvaren på PDA.

## 4.7 Versjonshåndtering

Versjonshåndtering brukes for å holde orden på dokumenter og de ulike versjonene av dem. I et prosjekt hvor flere aktører jobber med samme dokument, er det behov for å sikre at for eksempel ikke to stykker jobber på samme dokument nøyaktig samtidig. Dette sikres ved at versjonshåndteringssystemet "eier" alle dokumentene og brukerne må låse opp dokumentene for å få lov til å skrive til dem. Kun en aktør kan låse opp et dokument av gangen. Ved å låse dokumentet, lagres dokumentene en ny versjon.

Vi valgte å utelate dokumentering og rapport skriving fra versjonshåndtering. Siden vi bare er fire stykker, og vi jobbet sammen innen for et område på noen få kvadratmeter, følte vi ikke at vi fikk noe utbytte av det. Vi førte en manuell versjonshåndtering av dette.

Til kodingen er det brukt noe som heter Jalindi Igloo versjonshåndtering. Denne tilleggstenesten kan lastes ned gratis fra <http://www.jalindi.com/igloo>. Softwaren kan installeres og integreres i Visual Studio .NET.

Når man jobber med en applikasjon i Visual Studio, kalles samlingen av filer et prosjekt. Hvert av disse prosjektene knyttes til versjonshåndteringen. Versjonshåndteringssystemet tar da automatisk styring over de ulike filene i prosjektet og lagrer automatisk nye versjoner av filene gjennom utviklingen.

En fordel med å bruke versjonshåndtering, er at man kan sammenligne de ulike versjonene. La oss si at man har en versjon av programvaren som virker uten feil. To uker senere oppstår problemer man ikke finner ut av. Ved hjelp av versjonshåndteringen kan man sammenligne den versjonen med feil mot den som virket. Det er da enklere å finne feilkilden og man kan spare mye tid.

## 4.8 Installasjon

Til installering av programmer utviklet for Windows, har prosjektgruppen benyttet et program kalt Inno Setup. Programmet kan lastes ned gratis fra <http://www.jrsoftware.org/>. Inno Setup lager enkelt en installasjonsveiviser for Windows programmer. Inno Setup har eksempelvis støtte for å sette opp snarveier, la bruker velge installasjonssted, kopiere filer og å avinstallere programmet.

Til installasjon på PDA har gruppen sett etter tilsvarende løsninger som Inno Setup for PC. Det finnes en god utgave som heter Easy Pocket PC Installer. Ulempen er prisen. Fullversjonen koster penger og gruppen har av den grunn valgt å legge programmet til PDA i en katalog som ganske enkelt må flyttes over til hver enkelt PDA i systemet. Se installasjonsveiledning i vedlegg L.

## **5 Diskusjon av resultater**

### **5.1 Fremdriftsplan**

Store prosjekter er vanskelige å planlegge. Databransjen har gjennom mange år hatt rykte på seg for å sprengte både tidsrammer og økonomiske rammer. Prosjektgruppen har gjennom arbeid med dette prosjektet, fått forståelse for at utviklingsprosjekter innen IT kan sprengte rammer.

Det ble besluttet å bruke gantt-skjema til å planlegge fremdriften. I begynnelsen av prosjektet satte gruppen seg ned for å lage en plan for de neste fem månedene. Dette var ingen enkel oppgave. Sett fra vårt perspektiv, var usikkerhetsmomentene mange. Spesielt var vi usikre på hvor mye tid vi hadde behov for til opplæring i .NET rammeverket. I tillegg ble det ren gjetning fra vår side hvor lang tid det ville ta og utvikle hver modul. Første utkast til fremdriftsplan finnes i vedlegg N.

Senere i prosjektperioden har gruppen utarbeidet to nye fremdriftsplaner. Disse planene er laget i forbindelse med statusmøter. På det tidspunkt vi hadde siste statusmøte, passet den sist oppdaterte planen så godt for videre arbeid, at det ikke var behov for å utarbeide en ny.

Selv om prosjektplaner kan virke håpløse å lage, har prosjektgruppen erfart at prosessen med å lage en plan øker bevisstheten rundt arbeidsoppgavene. Gruppen har hatt diskusjoner og mange ulike forslag til fremdriftplan. I tillegg oppleves milepæler og delmål som gode verktøy å jobbe etter.

Tidlig i prosjektets fase ble gruppen liggende på etterskudd i forhold til planen. Først ved utgangen av påske greide vi å komme à jour. Totalt sett føler prosjektgruppen at det har vært nyttig å jobbe etter en fremdriftsplan. Dette til tross for at den virkelige fremdriften ikke har vært helt i tråd med planen.

## 5.2 Arbeid med kravspesifisering

I og med at vår oppdragsgiver skal dra nytte av vårt arbeid, hadde vi mye kontakt gjennom utarbeidelsen av kravspesifikasjon. Oppdragsgiver var naturlig nok interessert i innholdet i kravspesifiseringsdokumentet. Dette medførte en del endringer av innholdet dokumentet. Avsatt tid til kravspesifisering slo derfor feil.

Innlandets Vitensenter hadde før vårt prosjekt startet, utarbeidet en innholdskisse for det endelig systemet som skal dekke medieeksperimentet. Prosjektgruppen foretok sammen med oppdragsgiver et utvalg av funksjonalitet prototypen skulle inneholde. I denne oppgaven var altså mye funksjonalitet forhåndsbestemt. Vår utfordring ble å tilpasse prototypen. Gruppen valgte å utvikle den funksjonaliteten vi mente det lot seg gjøre innenfor våre rammer.

Gruppen har lagt mye vekt på kravspesifiseringen og revidert innholdet en rekke ganger. Noe av grunnen til at det tok lengre tid å bli ferdig enn antatt, skyldtes at gruppen i samråd med oppdragsgiver stadig hadde forslag til forbedringer. I tillegg følte vi det var nødvendig å endre funksjonalitet i takt med tilegnet kunnskap innenfor utviklingsmiljøet. Arbeidet var tidkrevende, men gav samtlige gruppemedlemmer nyttig erfaring som gruppen ikke ville vært foruten. Prosjektgruppen har tatt lærdom av hendelsen og erfart at kravspesifikasjon er vanskelig.

## 5.3 Realisering av krav til systemet

Det finnes en rekke krav til systemet, se kapittel 2. Gruppen har løst de aller fleste av kravene og har et system som dekker mesteparten av flyten i rollespillet som ble avtalt mellom prosjektgruppen og oppdragsgiver. Riktignok finnes det enkelte avvik. I ettertid ser gruppen at det ville ha vært gunstig å merke mindre viktige krav med "dersom tid" i kravspesifikasjonen.

Årsaken til avvik forklares for hver modul nedenfor.

### 5.3.1 Krav til administratormodul og bildemodul

Kravene til modulene finnes i kapittel 2.6.1 i denne rapporten. I kravspesifikasjonen finnes det til sammen fem krav til disse modulene. De fire første kravene har gruppen innfridd. Krav nummer 3 er riktignok kun delvis innfridd.

Til krav 3 har gruppen fått på plass det viktigste, nemlig funksjonalitet for å knytte elever til redaksjoner. Muligheten til å sette opp rollespillet på forhånd har gruppen ikke innfridd. Dette skyldes prioritering fra gruppens side. Mot slutten av prosjektperioden fant ikke gruppen det hensiktsmessig å bruke tid på denne funksjonaliteten da det var andre oppgaver som ble vurdert viktigere. Gjennom utviklingsfasen er systemet testet med to PDA-er. Av den grunn syntes gruppen funksjonaliteten var lite interessant å prioritere. Funksjonaliteten vil ha langt mer for seg dersom det er 32 PDA-er, dermed vil funksjonaliteten være langt mer nyttig i den endelige læringssimulatoren.

Krav 5 er ikke implementert. Gruppen har ikke vurdert kravet om å vise hvor langt en redaksjon har kommet, som kjernefunksjonalitet. Det ble nedprioritert mot slutten av prosjektet på grunn av mangel på tid.



### **5.3.2 Krav til redaksjonsmodulen**

Denne delen av systemet består av åtte krav, se 2.6.2. I all hovedsak er disse kravene innfridd.

I krav nummer 3 til denne modulen står det at eleven i som har valgt rollen som utøvende journalist, skal kunne velge oppdrag. Dette er implementert slik at rollespillmotoren tildeler oppdrag automatisk. Systemet leter etter elever som har valgt denne rollen og tildeler oppdrag.

### **5.3.3 Krav til PDAModulen**

Det finnes tre hovedkrav til systemet på PDA-ene. Viser her til kapittel 2.6.3. Riktignok er krav nummer 2 splittet i to deler. Systemet til prosjektgruppen dekker alle krav som er satt til denne modulen og mer til. PDAModulen er også den modulen som er blitt prioritert mest. Dette er også i henhold til avtale med oppdragsgiver.

## **5.4 Problemer underveis**

Gjennom prosjektet har gruppen støtt på en rekke problemer. Mesteparten av problemene har dukket opp i forbindelse med kodingen. Innledningsvis støtte vi også på vanskeligheter med nettverket systemet skulle bygges innenfor.

### **5.4.1 Problemer med oppsett av nettverk**

Da gruppen fikk utlevert utstyr av oppdragsgiver, startet arbeidet med å sette opp nettverket.

I systemet er det en bærbar PC som har rollen som tjener. På PC-en er det installert Windows Server 2003 operativsystem. Ved installasjon av denne softwaren, gjorde gruppen seg en nyttig erfaring. Tidlig i installasjonsprosessen, fikk vi meldinger av installasjonsprogrammet om at det manglet filer. Vi prøvde igjen og igjen, men tilsvarende feil oppsto. Gruppen syntes dette var snodig siden både PC-en og installasjons CD-en var nye. I tillegg hadde vi formatert disken, slik at det ikke skulle ligge igjen noe rusk. Til slutt gikk gruppen til det skritt å bytte ut CD-spillere på den bærbare PC-en for å se om feilen kunne ligge der. Vi fikk byttet CD-spiller med en tilsvarende IBM maskin, men ingen endring i oppførsel.

Senere skulle det vise seg at feilen skyldtes måten vi hadde formatert disken på. Innledningsvis når man installerer Windows operativsystem, kan man velge ulike måter å formatere disken(e) på. Prosjektgruppen hadde valgt hurtigformatting og dermed lå det igjen filer fra det forrige operativsystemet. Dette stakk kjepper i hjulene for vår installasjonsprosess.

Problemer dukket også opp med de håndholdte PC-ene. Operativsystemet på PDA-ene er fra fabrikken konfigurert til å skåne batteriet på PDA-en. Til å begynne med hadde gruppen noe bry med at PDA-ene gikk i hvilemodus. Dermed ble forbindelsen mellom PDA og tjeneren brutt. Da prosjektgruppen ble mer kjent med operativsystemet på PDA, fikk vi imidlertid bukt med problemet. Løsningen var ganske enkelt å endre innstillinger for strømsparing på PDA-ene.

Det trådløse nettverkskortet på PDA-ene har hatt en lei tendens til å velge nettverk av fri vilje. I det lokalet hvor prosjektgruppen har satt opp systemet, er det mange trådløse aksesspunkt tilgjengelig. Ved testing av funksjonalitet har det enkelte ganger vært vanskelig å forstå hvorfor kommunikasjon ikke fungerte. Dette skyldte i mange tilfeller at PDA-en var tilkoblet et annet nettverk. Etter hvert ble vi oppmerksomme på problemet og fikk også dette under kontroll.

#### 5.4.2 Problemer med koding

Programmering kan være frustrerende. I dette prosjektet har ikke gruppen greid å unngå å møte på større og mindre problemer med koding. Spesielt hadde vi mye trøbbel med bildemodulen.

I henhold til kravspesifikasjonen skulle bildemodulen plasseres på server. Gruppen startet arbeidet å diskutere løsning og sondere terrenget for ferdiglagde løsninger. Gruppen valgte å bruke gjenbruk av en åpen kilde vi fant på nettet. Gruppen fant en løsning som så ut til å passe. Problemet med løsningen vi fant, var at den var altfor omfattende i forhold til vårt behov. Litt tid gikk til å strippe kildekoden til det vi hadde bruk for. Selve jobben ble å plassere programmet inn i vårt system.

Etter litt tilpassning, fikk gruppen implementert bildesoftwarenn inn i systemet. Gruppen valgt til å begynne med å legge koden inn som en del av Administratormodulen. Vi plasserte en test knapp i administratormodulen. Her kunne vi ta bilder uten problem og lagret dem dit vi ville. Alt virket veldig lovende til nå.

Da dette var gjort, startet arbeidet med å legge funksjonalitet ut på PDA-ene. Kravet til løsningen er blant annet at bilder skal virtuelt kunne tas fra PDA. Gruppen besluttet å sette opp en socketforbindelse til bildemodulen fra PDA. Ideen var at koden som da kjøres skulle være nøyaktig lik koden vi testet direkte innad i serverapplikasjonen. Gruppen tok det som en selvfølge at dette ville fungere siden gruppen nå hadde opparbeidet noe erfaring med socketforbindelser fra registreringen.

Av en eller annen grunn fungerte ikke vår plan. Vi kunne ta bilder fra serverapplikasjonen, men ikke fra PDA. Ekstra frustrerende var det at samme kode som feilet ved oppkobling via socket, virket når vi la koden i en knappelytter direkte i applikasjonen. Vi visste dermed at det ikke var selve koden det var noe i veien med, men at feilen må ligge på et annet plan.

Heldigvis for oss er Visual Studio et utmerket verktøy når det kommer til debugging. Det vil si at man kan gå igjennom koden linje for linje og finne ut hvor det oppstår feil mens programmet kjører. Ved å gjøre nettopp dette, greide vi å spore kilden til feilen. Systemet gav oss en feilmelding vi ikke fant ut av umiddelbart. Feilmeldingen lød som følger: "System.Engine.Excecution.Error". Ved å oppsøke nyhetsgrupper på nettet, fant vi straks ut at dette var en alvorlig feil. Vi lyktes imidlertid ikke å finne noen med tilsvarende problem, slik at vi ble nødt til å søke hjelp andre steder.

I skolens fagmiljø for data finnes det dessverre for oss ikke mange med erfaring innenfor .NET rammeverket. Det tok derfor litt tid for oss å komme i kontakt med en person som kunne hjelpe oss. Vi fikk heldigvis hjelp og greide å løse problemet ved å skille ut bildemodulen som en egen prosess. Det viste seg at problemet skyldtes trådkonflikt. I systemet er bildemodulen nå implementert som en Windows Service og kjører i bakgrunnen på operativsystemet. Bildemodulen startes fremdeles av administrator applikasjonen.

## **5.5 Videre arbeid med systemet**

Videre arbeid med læringssimulatoren må baseres på spesifikasjonen til Innlandets Vitensenter. I dette dokumentet fra vår oppdragsgiver er det endelige systemet beskrevet, se vedlegg P.

### **5.5.1 PDAModulen**

Løsningen til PDA er godt rustet til å passe inn i en helhet. Programvaren er implementert som en utvidet nettleser. PDA-ene skal spille av lyd og fungerer som nikkedukker for server. Server befaler og PDA gjør som server sier. Det er enkelt å utvide funksjonaliteten til PDA-ene ved å lage flere nettsider. For prosessmotoren i det fremtidige systemet er det bare å dirigere PDA til de nettsider som er ønskelige.

På PDA-ene skal det under registrering skives tekst. Som kjent finnes det ikke tastatur på en PDA. For å løse problemet med å skrive tekst, finnes det et ferdiglaget virtuelt tastatur i operativsystemet. Prosjektgruppen har valgt å benytte dette tastaturet til registrering for å spare tid. Ved testing fikk gruppen tilbakemeldinger på at tastaturet er lite brukervennlig.

En annen mulighet er å lage et virtuelt tastatur på nettsidene der dette er nødvendig. Dette kan baseres på en type tastatur vi kjenner igjen fra mobiltelefoner. Brukergruppen til læringssimulatoren vil som sagt være ungdomsskoleelever og denne gruppen har i alle fall erfaring med bruk av tastatur på mobiltelefoner. Dette kan bør man kanskje dra nytte av.

Et problem prosjekt gruppen ikke har rukket å ta så mye hensyn til, er hva som skal skje dersom PDA-en låser seg og må restarter midt i rollespillet. Dette skyldes at gruppen ikke har rukket i prosjektet å utvikle selve rollespilldelen av simulatoren langt nok. Det har derfor ikke blitt noe tema.

Dersom en omstart av PDA-en inntreffer, vil det i prototypen fortsatt være lagret i databasen alt hva eleven har gjort. Problemet er at det vil være vanskelig komme tilbake i spillet. Problemet med omstart bør vurderes nøye fra Vitensenterets side, da PDA-ene er ustabile enheter.

En faktor som kan fremprovosere omstart, er en liten knapp som finnes på alle PDA-er. Denne knappen restarter PDA-ene. Med en hvilken som helst spiss gjenstand kan brukeren trigge omstart. Denne knappen bør derfor fysisk skjules på ett eller annet vis. I tillegg er PDA-ene egentlig ikke robuste nok til å bli mistet i gulvet og bli behandlet uforsiktig. Med elever fra ungdomsskolen er nok ikke dette til å unngå. En anbefaling fra gruppen side vil være å investere i en "rustning" for PDA-en. Denne kan bestå av gummiforsterkninger og en fysisk sperre for knappen som restarter PDA-en.

### **5.5.2 Administrasjonsmodulen**

Her er mulighetene for mer funksjonalitet mange. Et styringssystem kan inneholde styring for alt som foregår i rollespillet. Poenget med et slikt verktøy er å sikre at det for rollespilleder er enkelt å sette i gang et rollespill og styre det uten å ha detaljerte datakunnskaper.

Det bør for eksempel lyse "varselamper" hver gang elever gjøre noe galt skjer i rollespillet. En annen ide er å utvide med mulighet for å sette innstillinger i programmet. For eksempel endre portnummer, fjerne elever fra redaksjoner ved feil etc.

### 5.5.3 Redaksjonsmodulen

Den virtuelle redaksjonen kan utvides i flere retninger. Her kan man bygge på opplæring for hvordan lage TV, radio og andre medier. Kun fantasien setter begrensninger for videre arbeid.

Når det gjelder brukergrensesnittet i prosjektgruppens løsning, er dette laget i Visual Studio .NET. Utseende på redaksjonsprogrammet er i dag Windows basert. I en endelig løsning vil prosjektgruppen anbefale å bruke mer grafikk. Eksempelvis tror prosjektgruppen at Flash vil være et egnet utviklingsmiljø for brukergrensesnittet i redaksjonene. Flash er et verktøy for å generere interaktive og animerte nettsider. Siden gruppen ikke har noe erfaring med verktøyet fra før, anså gruppen risikoen til å være for stor til å lære seg dette verktøyet i tillegg til .NET.

Den virtuelle redaksjonen er laget med tanke på bruk av "touch-screen" eller trykkfølsom skjerm. Denne typen skjermer lar brukeren fysisk trykke på skjermen for å bruke et system. Trykkskjermer brukes i mange sammenhenger og erstatter musepekerens jobb. I dette prosjektet har vi laget brukergrensesnitte med tanke på trykkskjerm, men testet og utviklet med museklikk.

## 5.6 Diskusjon av retningslinjer fra oppdragsgiver

Til dette prosjektet har gruppen egentlig hatt to spesifikasjoner å forholde seg til. Den ene er kravspesifikasjonen som gruppen sammen med oppdragsgiver har utarbeidet. Den andre er den tekniske spesifikasjonen av den totale medielaben som vår oppdragsgiver har utarbeidet. I denne spesifiseringen var mange valg foretatt før vårt prosjektarbeid startet. For eksempel ble gruppen anmodet om å utvikle prototypen i .NET rammeverket på bakgrunn av hva som var bestemt i oppdragsgivers spesifikasjon. I mange tilfeller har ikke gruppen hatt mulighet til å velge teknologi fritt.

### 5.6.1 Utviklingsmiljø

Dersom gruppen i januar fritt kunne valgt utviklingsmiljø, ville nok valget falt på Java. Siden alle gruppemedlemmene har gjennomført kurs i Java, ville Java vært mest naturlig å velge for oss.

Det er mange likheter mellom .NET og Java. I begge rammeverk står objektorientering sentralt. Rammeverkene har gode klassebiblioteker. Det finnes også forskjeller. Java har i større grad rykte på seg for å ha god sikkerhet, et emne hvor Microsoft har et litt frynsete rykte. Når det kommer til grafisk brukergrensesnitt har nok Microsoft et lite overtak.

Gruppen hadde ingen kjennskap til .NET plattformen på forhånd. Etter å ha jobbet med rammeverket i flere måneder skjønner vi at .NET stadig blir mer og mer populært. Microsoft har langt på veg lyktes i å lage et godt rammeverk som støtter mange ulike teknologier og behov. Spesielt er gruppen godt fornøyd med mulighetene for utvikling for PDA. Rammeverket med navnet .NET Compact Framework er kraftig samtidig som det er forholdsvis enkelt å bruke. Nyhetsgrupper og støtten som finnes på <http://msdn.microsoft.com> er det heller ingenting å si på. For ferske utviklere som oss har tilgangen på hjelp og brukervennlighet i utviklingsmiljøet hjulpet oss mye.

Nå som gruppen har noe erfaring med begge rammeverkene, virker det fornuftig fra oppdragsgiver å benytte Microsoft .NET som utviklingsplattform. Et sentralt tema i læringssimulatoren er grensesnittet mot brukerne. Her vil, slik som gruppen opplever det, Microsoft ha bedre innebygd støtte enn Java. Et annet poeng er den gode støtten for utvikling mot PDA som finnes i Microsoft sitt rammeverk.

### 5.6.2 Trådløst nettverk 802.11b

IEEE, som er en organisasjon som lager en rekke ulike standarder for it-løsninger, har laget tre standarder for WLAN. Disse er 802.11a, 802.11b og 802.11g.

Standard 802.11a sin styrke er god støtte for mange brukere, høy hastighet og mulighet for å regulere frekvens. Rekkevidden er en svakhet for denne standarden. Maksimal overføringshastigheten er 54 mbps.

Standard 802.11g har god støtte for mange brukere med høy hastighet og har den beste signalstyrken av de tre alternativene. Ulempen er at teknologien er dyrere enn både 802.11a og b. Maksimal overføringshastigheten er 54 mbps.

Dette er den mest vanlige formen for WLAN er standarden 802.11b. Styrken til denne er god rekkevidde. En svakhet er at den ikke støtter like mange brukere med høy hastighet som 802.11a. Overføringshastigheten er lavere enn de to andre standardene, med 11 mbps.

Blåtann og infrarød er to andre former for trådløs teknologi. Spesielt er infrarød et dårlig alternativ. Overføringshastigheten og rekkevidden til infrarød er svært begrenset. Der det kan være mulig å bruke infrarød, kan være ved fremtidige dokkingstasjoner i systemet til å kartlegge hvor en PDA befinner seg i lokalene til læringssimulatoren.

Apple.com [4] skriver at "Bluetooth™-teknologien er en banebrytende, åpen spesifikkasjon som muliggjør trådløse forbindelser over korte avstander mellom stasjonære og bærbar datamaskiner, håndholdte enheter, mobiltelefoner, mobiltelefoner med kamera, skrivere, digitalkameraer og til og med datamaskintastatur og -mus."

For Blåtann operer man med en rekkevidde på omkring 10 meter, hvilket vil være i snaueste laget for læringssimulatoren. Når det gjelder overføringshastighet er også trådløst WLAN 802.11b overlegent i forhold til Blåtann. Bluetooth 2, som er siste versjon av denne teknologien, har maks overføringshastighet på 3 mbps. Til sammenligning har Vitensenterets valg av teknologi, WLAN 802.11b, en overføringshastighet lik 11 mbps.

Fra oppdragsiver er det besluttet at all kommunikasjon mellom klient og server skal foregå via trådløst WLAN 802.11b. Som forklart er dette den standarden til IEEE som har lavest overføringshastighet. Slik det er tenkt i læringssimulatoren skal klientene være lubne. Med det menes det at for eksempel PDA-ene skal ha lydfiler plassert lokalt. På den måten slipper man å sende store filer over nettverket. Dersom dette gjennomføres, vil det holde i massevis med den overføringshastigheten 802.11b tilbyr.

Gruppen var tidlig skeptisk til WLAN generelt på bakgrunn av usikkerhet rundt batterikapasitet på PDA-ene. Det er nemlig slik at trådløse nettverkskort på håndholdte enheter bruker mye strøm når de er aktivert. Gjennom testing har det vist seg å ikke være noe stort problem. Så lenge det ikke er noe problem for PDA-ene å bruke WLAN, ser gruppen dette som den beste løsningen for systemet siden man kan ha en type teknologi som dekker all kommunikasjon.

### 5.6.3 Klientene i systemet

Arkitekturisk skal både redaksjon og PDA være klienter i systemet basert på spesifiseringen av Innlandets Vitensenter sin medielab. I prosjektgruppens prototyp, er det kun PDA-en som er klient. Redaksjonsmodulen er implementert som en applikasjon på server.

I prosjektgruppens system kjører serveren to applikasjoner. Den ene applikasjonen er administratormodulen og den andre er nevnte redaksjonsmodul. Mot slutten av prosjektet valgte gruppen å implementere løsningen på denne måten hovedsakelig for å teste funksjonalitet på PDA. I ettertid ser vi at for å imøtekomme de overordnede kravene til læringssimulatoren, burde dette vært løst på en annen måte. Vi burde implementert en nettleser i redaksjonen og gjort den til en tynn klient i stedet for en del av serveren. Dette var en bevisst avgjørelse fra gruppens side.

På det tidspunkt gruppen valgte å lage redaksjonen som en intelligent applikasjon, hadde gruppen problemer med oppdatering av nettsidene i nettleseren gruppen hadde kjennskap til. Med oppdatering mener vi at samme side lastes på nytt for å vise eventuelle endringer. Gruppen var skeptisk til å implementer nettavisen med bruk av nettleser og ASPX sider, på bakgrunn av behov for å koble seg til databasen for hver gang siden lastes. I ettertid ser gruppen at redaksjonen ideelt kunne vært laget ved hjelp av ASP.NET og sessions slik at redaksjonen ble en klient i systemet.

En av utfordringene når det gjelder utvikling med web, er å lage en konsekvent applikasjon av flere uavhengige HTML sider. Dette problemet gjelder spesielt web utvikling fordi http er en statusløs protokoll. Hvert kall til en webserver er uavhengig, og serveren har ingen kunnskap om en nettlesers tidligere kall. For å overkomme denne begrensningen, brukes en teknikk for å få skaffe konsekvente brukersesjoner via web. ASP gir en kraftig og fleksibel websesjons håndtering som ikke krever noen spesiell form for programmering. Sesjonsobjektet, et av de vesentlige objektene som ASP støtter, gir en komplett web sesjon håndterings løsning.

## 5.7 Kritikk av oppgaven

I prosjektet er det rom for forbedringer på mange plan. Det er lett i ettertid å sette fingeren på mange ting som burde vært gjort annerledes. Gruppen har gått opp mange nye stier og lært utrolig mye gjennom prosessen. Det sier seg selv at mye av tiden nettopp har gått med til læring. I dette avsnittet ønsker prosjektgruppen å være kritiske til eget arbeid.

### 5.7.1 Fremdriftsplan

Gruppen ser i ettertid at fremdriftsplanen ble noe grov. Prosjektgruppen hadde hodebry med å lage en plan på det tidspunktet dette skulle gjøres. Det var mange spørsmål omkring utviklingsmiljø og funksjonalitet som gjorde oss usikre. Det å planlegge en 6-måneders periode virket nytteløst.

Det ble laget en plan og gruppen har blitt noen erfaringer rikere. Slik vi ser det, er mye erfaring den viktigste forutsetningen for å lage en god fremdriftsplan. Det gruppen har fått mest ut av planen er faseinndelingen og innleveringsfrister. Gjennom prosjektet har det vært veldig greit å se hvor man bør ligge an og justere seg deretter. Hovedpoenget med planen var å sikre at dersom vi fulgte planen, ville vi komme i mål. Så i de tilfeller hvor vi var på overtid med en modul, brukte vi planen til å rette opp båten og få den på rett kurs.

I planen har vi delt inn arbeidet med hver modul i to inkremitter. Den første iterasjonen var ment å bestå av en utvikling og en integrering av det som ble utviklet i denne iterasjonen, for så at den andre iterasjon skulle være en videre utvikling av iterasjon en. I praksis har gruppen fulgt en stor iterasjon innefor hver av modulene som er listet opp. Det har ikke vært noe klart skille på de to iterasjonene og arbeidet med hver modul har gått i ett.

### **5.7.2 Dialog med oppdragsgiver**

Gjennom prosjektet har gruppen hatt løpende dialog med oppdragsgiver. I begynnelsen var møteaktiviteten hyppig. Utover i prosjektperioden avtok møteaktiviteten noe, hvilket begge parter var inneforstått med.

Nokså tidlig i prosjektet oppsto det et avvik med hva oppdragsgiver forventet av kravspesifikasjonen og hva gruppen hadde utarbeidet. Gjennom et møte fikk vi fanget opp problemet og endret kravspesifikasjonen en del. Blant annet ville oppdragsgiver ha med detaljerte flytdiagrammer som viste gangen i prototypen.

### **5.7.3 Versjonshåndtering**

Håndtering av ulike dokumenter ved hjelp av manuell versjonshåndtering har fungert bra. Gruppen kunne ha kommet tidligere i gang med disse rutinene. Denne måten å sikre dokumenter på ble tatt i bruk etter at forprosjektet var avsluttet. Vi burde ideelt sett har startet med dette helt fra starten av. Styringen ble merkbart bedre.

Til koding har vi brukt et gratis versjonshåndteringssystem. Dette har stort sett fungert bra. Arbeid med versjonshåndtering har vært er ny erfaring for samtlige gruppelemmer. I begynnelsen måtte man sette seg inn i hvordan dette fungerte og venne seg til bruken før man ble komfortabel. Gruppen har ikke greid å gjennomføre versjonshåndtering på alle modulene, men på mesteparten.

### **5.7.4 Koding**

Som videreutvikler av systemet ville oppdragsgiver ha koden lagdelt etter modell-view-controller (MVC)prinsippet. Gruppen har ikke hatt mye erfaring med utvikling av store programmer og har i all hovedsak kjennskap til MVC fra teori.

Gruppen har forsøkt å etterleve MVC, men har ikke lyktes til fulle. Vi har forsøkt å tilrettelegge for å skille mellom brukergrensesnittet, databehandlingen og arbeid mot datalageret. For at koden skal tilfredsstillende MVC fullstendig, er det fremdeles en jobb igjen å gjøre.

Generelt kan man si at koden i prosjektet ikke er ideelt oppdelt. Gruppen burde ha skilt ut mye av koden i enkelte klasser og laget flere nye klasser. Mindre i hver klasse og flere klasser. Enkelte klasser har i systemet litt mye ansvar og avhengigheten til noen av klassene er store.

Nå i ettertid har vi lært at måten å skille lagene på i MVC er å skille brukergrensesnittet og kontroll logikken. Med views i toppen, controllerene i midten og modellen i bunnen. Lagene kan kun bruke servicene til laget under seg. Eller med Relaxed-Layered-Architecture (RLA) måten å se det på, der alle lagene bruker servicene til alle lagene som befinner seg under seg selv i hierarkiet. Sistnevnte metode gir stor fleksibilitet, men er mye vanskeligere å vedlikeholde.

### **5.7.5 3D modellen**

Under oppstarten av prosjektet var oppdragsgiver veldig interessert i 3d modellen vi hadde planer om å lage. Vi visste på forhånd at det kom til å ta for lang tid å lage en fullstendig implementasjon av en hjelpefigur med tanke på den begrensede tiden vi har i hovedprosjektet. Et poeng var å vise potensialet av en slik figur ved å implementere den i deler av systemet. Det skulle vise vanskelig å demonstrere figuren i programmet.

Gruppen besluttet å benytte gif-filer til animasjonene. Dette bød på problemer. Det er slik at animerte gif-filer oppfører seg forskjellig ut fra hardware og software, og fungerer ikke særlig bra som et medium der film må være synkronisert med lyd. For å få dette til å fungere måtte vi ha implementert en animasjonsavspiller eller Windows MediaPlayer. Dette ble ikke prioritert med tanke på at vi ikke kom til å få fullført animeringen av figuren.

Vi har likevel riggen og modellen klar. Den har omgivelser, lyssetting, tekstur og materiale-oppsett klart som kan brukes videre. Selv om modellen kan trenge utseendemessige justeringer ut fra oppdragsgivers ønsker, så er den klar til å animeres. Figuren ligger vedlagt i vedlegg I.

## **5.8 Evaluering av prosjektet**

### **5.8.1 Organisering**

Gjennom prosjektarbeidet har gruppen hatt få uoverensstemmelser. Arbeidsfordeling har stort sett fungert bra og rollene som ble definert i forprosjektrapporten, se vedlegg O.

I prosjektet har Håvard hatt rollen som prosjektleder. Hovedansvaret til prosjektleder har vært å styre gruppen og holde orden på det praktiske. Vi har ikke hatt store uenigheter innad i gruppen.

### **5.8.2 Arbeidsfordelingen**

Gjennom hele prosjektet har gruppen stort sett arbeidet sammen om både dokumentasjon og utvikling av systemet. Vi har jobbet mye to og to. Gjennom kodingsfasen har det blitt gjort en del arbeid enkeltvis. Siden vi har hatt mange moduler, har det i mange tilfeller vært hensiktsmessig å sette en mann per modul.

Gruppemedlemmene har ulike kvaliteter. Fordelingen av ulike arbeidsoppgaver har fungert slik at hvert gruppemedlem selv har valgt gjøremål. Det har ikke manglet på oppgaver i dette prosjektet. Gruppen har benyttet oppgavelister der man kvitterer som ansvarlig for en oppgave. Oppgavelistene er utarbeidet i fellesskap. Da en oppgave er utført, har den ansvarlige krysset av oppgaven som utført.



### **5.8.3 Subjektiv opplevelse av hovedprosjektet**

Prosjekt som arbeidsform er svært aktuelt med tanke på møtet med arbeidslivet. Hovedprosjektet ved HiG er det første store prosjektet vi har jobbet med og det har vært en svært nyttig erfaring.

Vi har erfart det ikke alltid er like lett å samordne produksjon av dokumenter og kode i prosjekter. I dette tilfellet har vi vært fire ulike forfattere av både rapport og kode. Det er helt klart en utfordring å få en god flyt gjennom det man produserer slik at det ser helhetlig ut.

Arbeid med prosjekt forbereder oss på arbeidslivet. I forhold til å jobbe mot eksamen, føler vi at man har mer igjen for å jobbe med prosjekt. Man får kombinert teori og praksis på en gunstig måte.

Noe som gruppen gjennom dette prosjektet ser på som svært verdifullt, er jobben vi måtte gjøre for å sette oss inn i ny teknologi. Som dataingeniør må man regne med at man får bruk for evnen til å lære seg nye ting.

Gruppen synes at skolens organisering av hovedprosjektet har vært ryddig. Skolens lærere har vært positive og lett tilgjengelige. For oss har det imidlertid vært vanskeligere å finne lærere å diskutere .NET med enn det ville vært med for eksempel Java.

## 6 Konklusjon

Etter at gruppen har arbeidet med å lage en prototyp på et system som skal bli en del av Innlandets Vitensenter sin læringssimulator sitter vi igjen med mange gode erfaringer. I systemet skal brukerne spille et rollespill som skal gi innføring i hvordan en nettavis blir til ved bruk av moderne teknologi.

For gruppen var det vanskelig i kravspesifiseringsprosessen å si med hundre prosent sikkerhet hva vi kunne løse, siden vi var totalt ukjente med miljøet vi skulle utvikle i. Derfor har enkelte krav blitt løst på alternative måter i forhold til kravspesifikasjonen.

Gruppen ser det som spesielt verdifullt å ha lært seg å utvikle i et nytt rammeverk. I databransjen må man stadig være forberedt på å lære seg ny teknologi. Utbyttet av å gjøre store prosjekter slik som dette, føles vel så stort som å følge kurs.

Dersom gruppen skulle utviklet systemet på nytt med den kunnskap vi har i dag, kunne vi tatt på oss å løse langt flere krav til det totale systemet til Innlandets Vitensenter. Usikkerheten rundt utviklingsmiljøet har naturlig nok svekket fremgangen.

Etter å ha gjennomført det avsluttende hovedprosjektet ved HiG er prosjektgruppen godt fornøyd med utbyttet og resultatet av prosjektet, til tross for at enkelte krav ikke gjennomført. Gjennom prosjektet har gruppa møtt på mange nye og spennende utfordringer.

Gruppen er også svært fornøyd med veileder og oppdragsgiver. Vi har fått en rekke nyttige tips og vink og har hatt en god dialog prosjektperioden igjennom.

## Bibliografi

- [1] Craig Larman, 2002, "Applying UML and Patterns, second edition", Prentice Hall
- [2] "Mitt dataleksikon" hos Pcworld.no. Hjemmeside  
<http://www.pcworld.no>
- [3] Software for deler av bildemodulen, DirectShow. Hjemmeside  
<http://www.codeproject.com/cs/media/directshownet.asp>
- [4] Info om Bluetooth (Blåtann). Hjemmeside  
<http://www.apple.com/no/bluetooth/>