

**HOVEDPROSJEKT:**



**GetAJob**

**FORFATTERE:**

BJØRN FOSSEN  
THOMAS MEUM  
JON ANDERS ØVERN

Dato: 10.mai 2005

## SAMMENDRAG AV HOVEDPROSJEKT

Tittel:	GetAJob	Nr: 4
		Dato: 10.05.05
Deltakere:	Bjørn Fossen	
	Thomas Meum	
	Jon Anders Øvern	
Veileder:	Rune Lossius	
Oppdragsgiver:	Maxus Media & Software	
Kontaktperson:	Sven Berg Ryen	
Stikkord (4 stk.)	CV, PHP, CV-maleditor, Java	
Antall sider: 64	Antall bilag: 5	Tilgjengelighet: Åpen
Kort beskrivelse av hovedprosjekt:		
<p>Prosjektgruppen har utarbeidet en online CV-generator. De som vil ha en CV registrerer seg på websiten og kan fylle ut sine CV-data. Deretter kan de se gjennom maler og kjøpe rettighet til en de liker. Denne malen blir nå tilgjengelig for brukeren og kan brukes så mange ganger vedkommende vil. Ut fra tilgjengelige maler kan bruker få generert en CV og tilsendt denne per e-post som et PDF-dokument. CV-malene i systemet lages av designere med et editorprogram som kjøres fra websiten. Hver gang det kjøpes en rettighet til en CV-mal får designeren av malen en provisjon. Administrasjonen av systemet håndteres av en administrator og en eller flere redaktører. Redaktøren skal kunne administrere designere og de brukere som ønsker en CV, samt kunne godkjenne nye CV-maler som er laget og godkjenne eller avslå søknader om å bli designer. Administratoren har samme ansvarsområde som redaktøren, men kan i tillegg administrere andre redaktører og se økonomiske rapporter.</p>		

## Forord

Hovedprosjektet er gjennomført for Maxus Media & Software. Prosjektet er en avsluttende oppgave ved dataingeniørstudiet ved Høgskolen i Gjøvik.

Vi vil takke Sven Berg Ryen og Håvard Fjær ved Maxus Media & Software, samt veileder Rune Lossius og Øivind Kolloen ved Høgskolen i Gjøvik for veiledning og hjelp under arbeidet.

Gjøvik 10. mai 2005

---

Bjørn Fossen

---

Thomas Meum

---

Jon Anders Øvern

## Innhold

<b>1</b>	<b>Innledning</b>	<b>1</b>
1.1	Problembeskrivelse . . . . .	1
1.2	Omfang og oppgavedefinisjon . . . . .	1
1.3	Målgruppe . . . . .	1
1.4	Egen faglige bakgrunn og kompetanse . . . . .	1
1.5	Rammer . . . . .	2
1.5.1	Gjennomføring og arbeidsmetoder . . . . .	2
1.5.2	Prosjektorganisering . . . . .	2
1.5.3	Prosjektoppfølgning og planlegging . . . . .	2
1.5.4	Systemutviklingsmodell . . . . .	3
1.6	Organisering av rapporten . . . . .	3
<b>2</b>	<b>Kravspesifikasjon</b>	<b>4</b>
2.1	Introduksjon . . . . .	4
2.1.1	Bakgrunn . . . . .	4
2.1.2	Kort om krav til systemet . . . . .	4
2.1.3	Referanser . . . . .	4
2.2	Brukerbeskrivelse . . . . .	4
2.2.1	Omgivelser . . . . .	4
2.2.2	Systemets brukere . . . . .	4
2.2.3	Operasjon . . . . .	5
2.2.4	Aspekter omkring livssyklus . . . . .	5
2.2.5	Ytelse . . . . .	5
2.3	Detaljert kravspesifikasjon . . . . .	5
2.3.1	Funksjonelle krav . . . . .	5
2.3.2	Operasjonelle krav . . . . .	14
2.4	Begrensninger . . . . .	14
2.4.1	Software design . . . . .	14
2.5	Aspekter omkring livssyklus . . . . .	14
2.5.1	Dokumentasjon . . . . .	14
2.5.2	Integrasjonstesting . . . . .	15
2.5.3	Krav til utvidelser . . . . .	15
2.6	Aspekter omkring installasjon . . . . .	15
2.6.1	Hardware installasjon . . . . .	15
2.6.2	Overgang . . . . .	15
2.6.3	Opplæring . . . . .	15
2.7	Utgivelser underveis . . . . .	15
2.8	Akseptansekrav . . . . .	16
2.9	Prosjektstyring . . . . .	16
2.9.1	Kvalitetssikring . . . . .	16

---

<b>3</b>	<b>Design</b>	<b>17</b>
3.1	Database . . . . .	17
3.1.1	Tabellen «bruker» . . . . .	17
3.1.2	Tabellen «designsoknad» . . . . .	17
3.1.3	Tabellen «mal» . . . . .	17
3.1.4	Tabellen «malrettighet» . . . . .	20
3.1.5	Tabellene «utdanning», «erfaring», «referanser» og «yrke»	20
3.1.6	Tabellen «post» . . . . .	20
3.1.7	Tabellen «grafikk» . . . . .	20
3.2	Website . . . . .	20
3.3	CV-maleditor . . . . .	22
<b>4</b>	<b>Implementering</b>	<b>23</b>
4.1	Utviklingsmiljø og programvare . . . . .	23
4.2	Koding . . . . .	24
4.2.1	Database . . . . .	24
4.2.2	CV-mal . . . . .	24
4.2.3	Websiten . . . . .	26
4.2.4	CV-maleditor . . . . .	30
4.2.5	CV-generering . . . . .	33
4.2.6	Paynet . . . . .	34
<b>5</b>	<b>Testing</b>	<b>35</b>
<b>6</b>	<b>Vurderinger og konklusjon</b>	<b>37</b>
6.1	Vurdering av resultatet . . . . .	37
6.2	Kritikk av oppgaven . . . . .	37
6.3	Alternative muligheter og valg underveis . . . . .	37
6.4	Fremtidige utvidelser . . . . .	37
6.5	Egenevaluering . . . . .	38
6.6	Konklusjon . . . . .	38
<b>7</b>	<b>Litteraturliste</b>	<b>39</b>
<b>A</b>	<b>Definisjoner</b>	<b>40</b>
<b>B</b>	<b>Planlagt og faktisk fremdrift</b>	<b>41</b>
B.1	Gantt-diagram for planlagt fremdrift . . . . .	41
B.2	Gantt-diagram for faktisk fremdrift . . . . .	42
<b>C</b>	<b>Statusrapporter</b>	<b>43</b>
<b>D</b>	<b>Logg</b>	<b>49</b>
D.1	Dagslogg . . . . .	49
D.2	Møtereferater . . . . .	53

**E CD-rom**

**57**

## Figurer

1	Use Case diagram for website. . . . .	6
2	Use Case diagram for editor. . . . .	12
3	Tabellene i databasen. . . . .	18
4	ER-modell for databasen. . . . .	19
5	Klasser brukt på websiten. . . . .	21
6	Klasser i CV-maleditoren. . . . .	22
7	Eksempel på en del av enkel CV-mal. . . . .	25
8	Bilde av editoren . . . . .	31
9	Bilde av editoren med preview . . . . .	32
10	Gantt-diagram for planlagt fremdrift. . . . .	41
11	Gantt-diagram for faktisk fremdrift. . . . .	42

# 1 Innledning

## 1.1 Problembeskrivelse

Mange har ikke helt øye for design og estetikk når det gjelder utforming av dokumenter. Slike faktorer er viktig i forbindelse med utforming av en CV for å gi et godt førsteinntrykk til en eventuell arbeidsgiver. Det er i dag standard å bruke dataverktøy, men det finnes ingen enkeltstående program for å lage en profesjonell CV, som er lett å bruke for alle og enhver. Man kan leie en designer privat til å utforme CV-en sin, men dette kan være dyrt og vanskelig tilgjengelig.

## 1.2 Omfang og oppgavedefinisjon

Prosjektet har som mål å få utviklet et CV-genereringssystem. Det er 4 brukerroller av systemet, vanlige brukere, designere, redaktører og administratorer. Førstnevnte brukerrolle er personer som har behov for å få generert en CV med profesjonelt utseende. Til dette skal det benyttes CV-maler som designere lager via en editor i systemet. Disse personene som ønsker seg en CV registrerer seg og får en kundekonto i systemet, og fyller inn sine relevante data til sin CV. Deretter kan de kjøpe rettighetene til en CV-mal, og få sin CV generert ut fra denne og tilsendt per e-post.

En designer får provisjon hver gang en bruker av systemet kjøper rettigheten til en CV-mal. Redaktørene i systemet har ansvaret for å administrere brukerne av systemet. Dette innebærer å godkjenne nye CV-maler som lages og kunne oppgradere/nedgradere brukertypen til brukere av systemet, da bare kunne endre status fra bruker til designer og omvendt.

Vanlige brukere som ønsker å bli designere i systemet sender en søknad, og det er også redaktørens rolle og kunne godkjenne eller avslå søknaden.

Administrator av systemet skal kunne gjøre alt som redaktørene kan i tillegg til å utnevne andre redaktører eller administratorer. Administratorer skal også kunne se en økonomisk rapport over systemet, der han ser hvilke CV-maler hvis rettighet er kjøpt og hvilke designere som skal ha provisjon.

## 1.3 Målgruppe

Målgruppen for det endelige produktet Maxus Media & Software, samt fremtidige brukere av systemet. Målgruppen for rapporten er veileder og oppdragsgiver. Andre som er interessert i å lese om systemet kan ha nytte av rapporten, for eksempel ved videreutvikling av systemet.

## 1.4 Egen faglige bakgrunn og kompetanse

Samtlige 3 medlemmer på gruppen går siste året på Bachelor i Ingeniørfag, data, og har valgt programvareutvikling som studieretning. Programvareutvikling



innebærer et programmeringsfag, Java, og objektorientert systemutvikling. Siden alle hadde valgt samme studieretning hadde vi mye av den samme kompetansen. Dette medførte at det måtte læres nye programmeringsspråk. Ellers har Bjørn jobbet ett år før han startet her og Jon Anders studerte ett år ved it-akademiet på Hamar. Thomas startet direkte etter fullført førstegangstjeneste.

## **1.5 Rammer**

### **1.5.1 Gjennomføring og arbeidsmetoder**

#### **Gjennomføring, arbeidsmetoder, prosjektorganisering, fremdriftsplan**

Vi har hatt jevnlig kontakt med oppdragsgiver og veileder for å underrette om progresjonen i arbeidet. Vår arbeidstid var fra 09.00 - 16.00 fra mandag - torsdag, og 09.00 - 14.00 på fredager. Unntaket ble tirsdag da det gikk et fag, WWW-teknologi, hele dagen, med teori før lunsj og labaktivitet etter. Jon Anders tok dette faget, mens Bjørn og Thomas tok et ekstra fag før jul slik at de manglet bare hovedprosjektet for å fylle opp gjenstående studiepoeng. De har likevel til dels fulgt faget. Dette medførte at tirsdagen ble litt amputert som arbeidsdag, men var det tid til overs på tirsdager så jobbet man med prosjektet. Vi jobbet i samlet gruppe på skolen i stedet for hver for oss slik at arbeidet ble mer effektivisert. Dokumentasjon av arbeidet, herunder dagslogg, møtereferater og rapporten generelt ble det jobbet kontinuerlig med.

### **1.5.2 Prosjektorganisering**

#### **Ansvarsforhold**

- Gruppeleder og loggansvarlig : Bjørn Fossen
- Webansvarlig og kontaktperson : Jon Anders Øvern
- Driftsansvarlig : Thomas Meum

I tillegg til dette hadde samtlige roller som programmerer, planlegger og rapportskriver.

**Øvrige roller** Veileder er Rune Lossius. Vi benyttet oss også av andre lærere med de rette kunnskaper til å hjelpe oss med vanskelige problemer vi kom over.

### **1.5.3 Prosjektoppfølgning og planlegging**

Planleggingen av prosjektets fremdrift er vist i Gantt-diagrammet på side 41.

#### 1.5.4 Systemutviklingsmodell

Vi valgte inkrementell systemutvikling av flere grunner. Oppgaven var lett å dele inn i inkremerter og det positive med inkrementell utvikling er at man er sikker på å få ferdig de første inkrementene, som kan testes ut, for så å bygge på videre.

### 1.6 Organisering av rapporten

Prosjektrapporten er delt inn i 7 kapitler, nummerert fra 1 til 7. Vedleggene kommer etter disse, indeksert fra A til E.

**Kapittel 1 Innledning** Gir en beskrivelse av prosjektet og oppbyggingen av resten av rapporten.

**Kapittel 2 Kravspesifikasjon** Beskriver kravene som stilles til systemet.

**Kapittel 3 Design** Beskriver oppbyggingen av systemet, viser overordnet arkitektur.

**Kapittel 4 Implementering** Beskriver utviklingsmiljøet og implementasjon av systemet.

**Kapittel 5 Testing** Beskriver hva og hvordan vi har testet.

**Kapittel 6 Vurderinger og konklusjon** En oppsummering av arbeidet med ulike vurderinger.

**Kapittel 7 Litteraturliste** En oversikt over kilder vi har brukt i rapport-skrivingen.

**Vedlegg** Diverse dokumenter som støtter opp om selve rapporten, for eksempel Gantt-diagram, logg og statusrapporter.

## 2 Kravspesifikasjon

### 2.1 Introduksjon

Introduksjonsdelen av kravspesifikasjonen er ment å gi en overordnet beskrivelse av systemet som skal utvikles.

#### 2.1.1 Bakgrunn

Prosjektet utføres for Maxus Media & Software ved Sven Berg Ryen. Dette firmaet har erfaring med hjemmesider og har siden 2001 arbeidet med vedlikehold og publisering av slike. Maxus Media & Software vil ha utviklet en website for generering av CV med profesjonelt utseende, og håper med dette produktet å fylle et hull i markedet.

#### 2.1.2 Kort om krav til systemet

Det skal lages et webbasert CV-genereringssystem. Brukere av systemet skal kunne logge inn, og legge inn sine CV-data og kunne ajourføre disse. Videre kan de velge ut en mal for en CV, kjøpe rettigheten til denne og få tilsendt sin CV per e-post. Disse malene er laget av designere via en CV-maleditor i systemet. Hver gang en bruker kjøper rettigheten til en CV-mal skal designeren av malen få provisjon. Det er også ønskelig å lage systemet så fleksibelt at det senere vil være mulig å enkelt utvide systemet til å støtte andre dokumenter enn CV-er, f.eks visittkort.

#### 2.1.3 Referanser

Noe av kravspesifiseringen vil ta utgangspunkt i prosjektrapporten fra prosjektet GetAJob(Øvern et al. 2004) i faget objektorientert systemutvikling fra høsten 2004.

## 2.2 Brukerbeskrivelse

### 2.2.1 Omgivelser

Systemet skal kjøre på en webserver som kan koble til en databaseserver. Det eneste kravet til webserveren er at den må ha støtte for PHP og ha installert Pear og Image Magick. Den må også ha tilstrekkelig ram.

### 2.2.2 Systemets brukere

Primærbrukerne av systemet skal være jobbsøkere som ikke nødvendigvis har stor kompetanse innen data. Systemet skal være så enkelt å bruke at dette ikke er noe problem. Der det er behov for det, skal systemet tilby hjelp til brukerne, f.eks ved hjelp av eksempler på hvordan ting skal fylles

ut. Brukerne aksesserer systemet ved hjelp av en nettleser. Andre brukere av systemet er administrerende brukere og CV-maldesignere.

### **2.2.3 Operasjon**

Systemet skal være tilgjengelig 24 timer i døgnet. Ved feil skal det sendes beskjed til hovedadministrator. Transaksjoner ved betaling av mal skjer via et eksternt betalingssystem. Dette systemet ivaretar sikkerheten rundt betalingen, så vi trenger ikke å tenke noe på det.

### **2.2.4 Aspekter omkring livssyklus**

Vi skal ikke vedlikeholde systemet siden det er et avsluttende hovedprosjekt i mai 2005. Systemet skal kunne flyttes til en annen server slik at oppdragsgiver kan få nytte av systemet. Rapporten vil fungere som dokumentasjon for systemet, slik at andre eventuelt kan vedlikeholde og utvide systemet ved en senere anledning.

### **2.2.5 Ytelse**

Systemet bør ha en responstid som samsvarer med det som er normalt på lignende webløsninger. Generering av CV-en i PDF-format kan ta noe tid, men da skal bruker få beskjed om dette underveis.

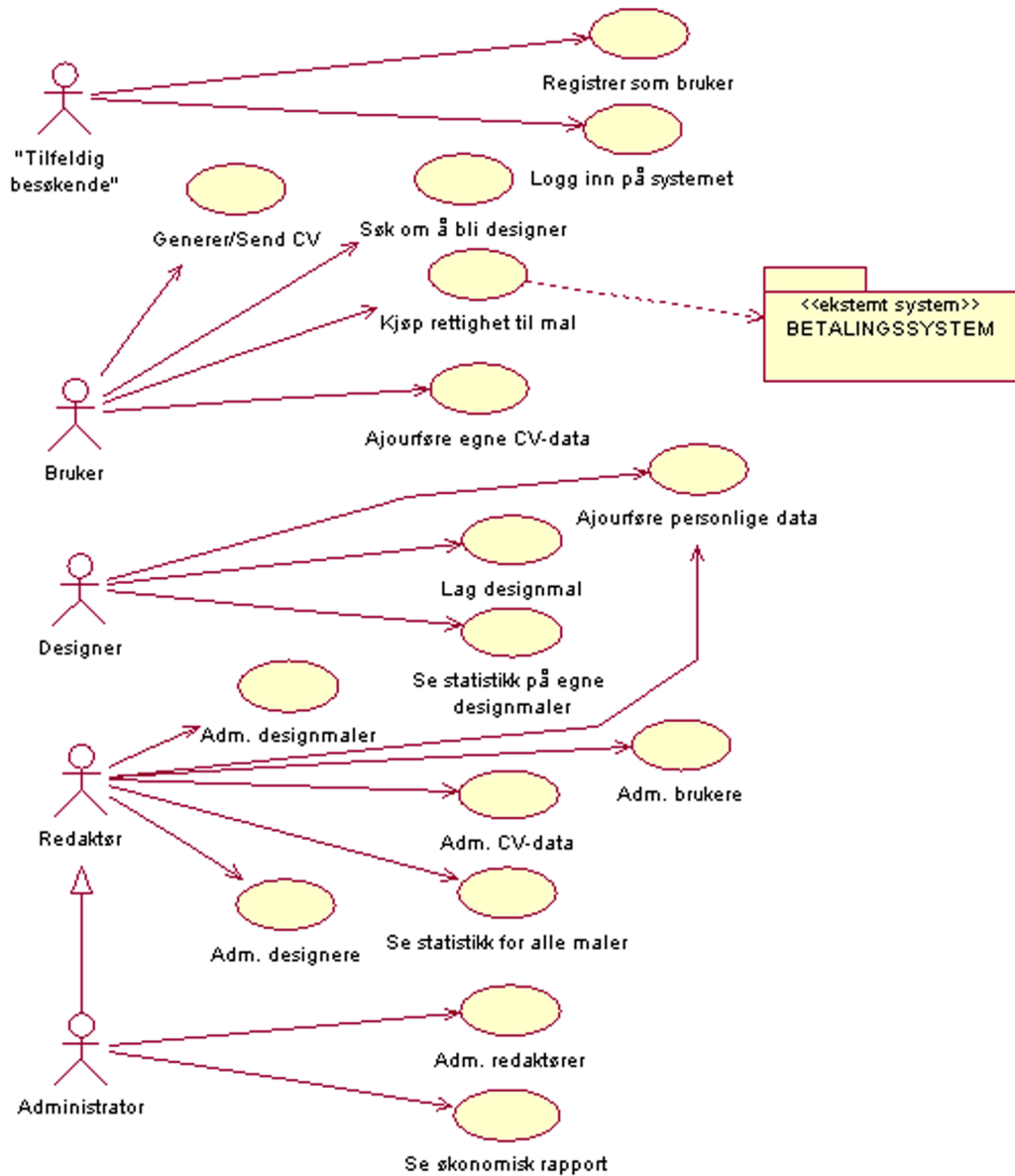
## **2.3 Detaljert kravspesifikasjon**

### **2.3.1 Funksjonelle krav**

Vi mener det er best å vise de funksjonelle krav gjennom Use Case-diagram med tilhørende beskrivelser av hvert Use Case. Vi har kategorisert hvert Use Case under den modulen vi mener det best passer inn under. Use Case-diagrammet og Use Casebeskrivelsene er hentet fra rapporten som er nevnt under punkt 2.1.3 side 4. Disse er gjennomarbeidet på nytt og modifisert.

Modulene som hvert Use Case tilhører vises her.

- Database: Designe og opprette database.
- Login og admin-modul: Alt som har med registrering, innlogging og administrering av brukere å gjøre.
- CV-modul: Lagring av CV-mal i database og ajourføring av disse dataene.
- Betalingsmodul: Tilkobling mot eksternt betalingstjeneste.
- CV-genereringsmodul: Genering av CV, sammenfletting av mal sammen med CV-data.
- CV-maleditor: Lage malen for hvordan en CV skal se ut.



Figur 1: Use Case diagram for website.

**Database modul** Ingen av Use Casene er direkte knyttet til oppbyggingen av database. Designet for databasen kommer vi tilbake til under designseksjonen.

---

### **Inkrement Login og admin**

**Navn:** Registrer som bruker

**Mål:** Å bli gjenkjent av systemet som godkjent bruker

**Primæraktør:** Tilfeldig besøkende

**Beskrivelse:** En hvilken som helst person som kommer inn på websiten må logge seg inn for å benytte seg av tjenestene som tilbys. En ny bruker må derfor registrere personlig informasjon, velge brukernavn og passord for så å bli lagt til i databasen over brukere. En brukerprofil blir opprettet.

---

**Navn:** Logg inn på systemet

**Mål:** Å få tilgang til tjenesten som systemet tilbyr

**Primæraktør:** Tilfeldig besøkende

**Beskrivelse:** En registrert bruker/designer/administrator må logge seg inn på systemet. Først etter at en tilfeldig bruker har logget inn, blir han gitt tilgang til systemets kjernefunksjonalitet. Dette gjør han ved å skrive inn sitt brukernavn og passord.

---

**Navn:** Administrere designmaler

**Mål:** Å kunne moderere hva som legges ut av designmaler.

**Primæraktør:** Redaktør

**Beskrivelse:** Når en redaktør logger inn på systemet vil ha få en liste over hvilke designmaler som er lagt ut siden sist han var innlogget. Redaktøren skal da ha muligheten til å fjerne de malene han ikke syntes passer eller er gode nok.

---

**Navn:** Administrere designere

**Mål:** Å kunne godkjenne en designersøknad, og dermed legge til designeren til systemet. Han skal også ha mulighet til å fjerne designere han mener har brutt reglene eller ikke gjør en god nok jobb.

**Primæraktør:** Redaktør

**Beskrivelse:** En redaktør får en oversikt over siste forandringer siden forrige innlogging, og eventuelle søknader som har kommet inn. Hvis denne personen møter kravene som settes til å være designer, blir han godtatt og lagt til som designer. Hvis det skulle være slik at det er en og annen designer som bryter reglene for systemet, eller viser seg å være inkompetent, skal redaktøren ha mulighet til å fjerne denne designeren fra systemet. Det skal være mulig å endre designerens personlige data ved behov.

---

**Navn:** Administrere CV-data

**Mål:** Å kunne gå inn på databasen for CV-data for å fjerne useriøse instanser.

**Primæraktør:** Redaktør

**Beskrivelse:** En redaktør skal få en oversikt over hver brukers CV-data som ligger i databasen. Her skal han også ha mulighet til å fjerne useriøse/upassende CV-data.

---

**Navn:** Lag designmal

**Mål:** Å kunne lage en designmal ved hjelp av editoren.

**Primæraktør:** Designer

**Beskrivelse:** Designer bruker editoren til å lage en designmal for så å lagre denne. Dette Use Case er splittet opp og beskrevet mer grundig nedenfor.

---

**Navn:** Se statistikk på egne designmaler

**Mål:** Å kunne få oversikt over hvor mange som har benyttet seg av egne maler.

**Primæraktør:** Designer

**Beskrivelse:** En designer skal kunne gå inn på en rapportside hvor han har oversikt over hvilken av sine maler som brukere har benyttet seg av, og hvor populære de er. Han kan der også se hvor mye han har tjent på disse malene.

---

**Navn:** Se statistikk for alle maler

**Mål:** Å kunne få en rapport over hvor mye hver designmal i systemet benyttes av bruker

**Primæraktør:** Redaktør

**Beskrivelse:** En redaktør får en oversikt over hvor mange ganger hver designmal har blitt kjøpt.

---

**Navn:** Administrere brukere

**Mål:** Å kunne slette brukere som ønskes fjernet

**Primæraktør:** Redaktør

**Beskrivelse:** Administratoren/redaktøren har tilgang til en side som viser en oversikt over registrerte brukere, der kan han fjerne brukere som av en eller annen grunn ønskes fjernet. Redaktør og administrator skal også kunne endre en brukers data, samt kunne endre en brukers brukertype.

---

**Navn:** Administrere redaktører

**Mål:** Å kunne utnevne en bruker til redaktør.

**Primæraktør:** Administrator

**Beskrivelse:** En administrator velger en person han vil ha som redaktør. Denne personen må være registrert som bruker av systemet og administrator oppgraderer statusen til redaktør. Hvis redaktøren ikke gjør en god nok jobb kan administrator slette vedkommende.

---

**Navn:** Se økonomisk rapport



**Mål:** Å kunne få en oversikt over hvilke designere man skylder penger, og registrere betaling.

**Primæraktør:** Administrator

**Beskrivelse:** Administratoren får se en rapport hvor systemet har regnet ut hvor mye man skylder til designerne. Etter at overføringene er gjort (går utenfor dette systemet) så går administratoren inn og haker av at betalingene er gjort, og systemet nullstiller gjelden.

---

**Navn:** Søk om å bli designer

**Mål:** Å få levert søknad om å bli godkjent som designer, og forandret sin status i systemet deretter.

**Primæraktør:** Bruker

**Beskrivelse:** En bruker eller en tilfeldig besøkende skal ha mulighet til å søke om å bli godkjent som designer. Et søknadsskjema skal legges ut hvor brukere/besøkende fyller ut ønsket informasjon og søknaden blir lagret i systemet.

---

## CV-modul

**Navn:** Ajourføre egne CV-data

**Mål:** Lagre og endre informasjon som brukeren senere skal kombinere med en designmal for å generere en CV.

**Primæraktør:** Bruker

**Beskrivelse:** Brukeren ajourfører all relevant informasjon inn i et skjema som er tilknyttet brukeren, og dette lagres unna til brukeren skal generere CV.

---

**Navn:** Administrere personlig data

**Mål:** Å kunne forandre på personlig data tilknyttet designeren, redaktøren

**Primæraktør:** Designer, redaktør

**Beskrivelse:** Designer, redaktør får en oversikt over personlig data, og kan forandre dette og lagre endringer.

---

### **Betalingsmodul**

**Navn:** Kjøp rettigheter til mal

**Mål:** Brukeren betaler for rettighet til å bruke ønsket mal, og kan derfra benytte seg av den.

**Primæraktør:** Bruker

**Beskrivelse:** En bruker får en oversikt over alle maler i systemet, samt en forhåndsvisning. Forhåndsvisningen består av en ikke-redigerbar mal med dummy data. Brukeren kan derfra velge en ønsket mal, og betale for den gjennom betalingsløsningen. Hvis betalingen går i orden blir designmalen gjort tilgjengelig for brukeren å bruke så mange ganger han vil.

---

### **CV-genereringsmodul**

**Navn:** Generer/ Send CV

**Mål:** Å få en CV generert ut i fra CV-data og valgt CV-mal, og tilsendt som en fil i PDF-format.

**Primæraktør:** Bruker

**Beskrivelse:** En bruker velger en tilgjengelig mal og får en oversikt over CV-data, og merker av for data som skal være med. Han får deretter generert en CV ut fra dette. CV-en sendes til brukeren over e-post.

---

### **CV-maleditor**

**Navn:** Lagre mal

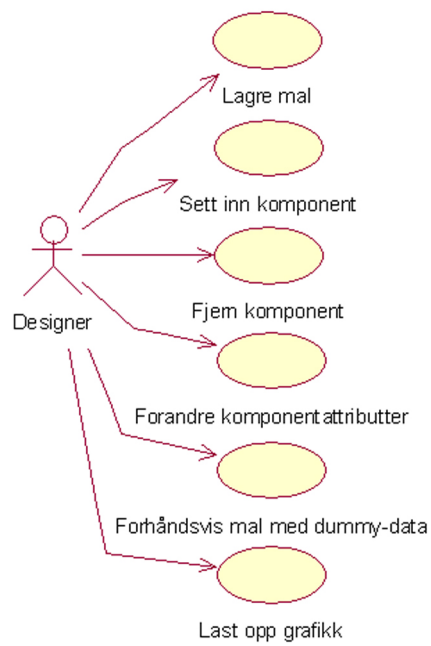
**Mål:** Få generert en CV-mal ut i fra dokumentet i editoren.

**Primæraktør:** Designer

**Beskrivelse:** En designer har gjort seg ferdig med CV-malen, og vil nå lagre denne. Komponentverdiene skal danne grunnlaget for en CV-mal i PDF-format. Malen vil bli generert og lagret midlertidig i systemet til en redaktør eller admin godkjenner eller underkjenner den.

---

**Navn:** Sett inn komponent



Figur 2: Use Case diagram for editor.

**Mål:** Få lagt til en komponent til gjeldene dokument.

**Primæraktør:** Designer

**Beskrivelse:** En designer vil sette inn en komponent i malen. En komponent er et generelt objekt, slik som en tekstboks, grafikk etc. Designeren vil velge hvilken type han vil sette inn, plassere denne og sette ønsket størrelse og eventuelt passende «tag». Hvis dette er en tekstboks, kan han skrive inn tekst.

---

**Navn:** Fjern komponent

**Mål:** Få fjernet en valgt komponent fra dokumentet.

**Primæraktør:** Designer

**Beskrivelse:** En designer er misfornøyd med plasseringen av en komponent og ønsker å fjerne denne.

---

**Navn:** Forandre komponentattributter

**Mål:** Å få flyttet/endret på allerede plasserte komponenter.

**Primæraktør:** Designer

**Beskrivelse:** Designeren vil tilpasse komponenten som han har plassert i editorvinduet. Dette vil innebære forandring av posisjon, størrelse og «tag». Hvis dette er en tekstboks kan han forandre teksten.

---

**Navn:** Forhåndsvisning med dummy-data

**Mål:** Å få en forhåndsvisning av designmalen med dummy-data fra databasen.

**Primæraktør:** Designer

**Beskrivelse:** Designeren har lyst til å se hvordan designmalen tar seg ut med faktiske data, og skal ha muligheten til å få en forhåndsvisning. Systemet skal generere en CV ut i fra malen som er under utvikling med dummy-data fra databasen. Dummy-data er fiktiv informasjon om en bruker som ikke eksisterer i virkeligheten.

---

**Navn:** Last opp grafikk

**Mål:** Å kunne hente inn grafikk og legge den til en mal i editoren.

**Primæraktør:** Designer

**Beskrivelse:** Designeren skal kunne laste opp grafikk og legge den til i editoren. Han skal få en oversikt over tilgjengelig grafikk og få en forhåndsvisning av denne i form av en liste. Han skal lett kunne legge disse til CV-malen, og kunne skalere og flytte dem etter ønske.

---

### 2.3.2 Operasjonelle krav

**Krav til mal for design** Designmalene skal lagres i PDF, og CV-data skal hentes fra databasen og settes inn i riktige steder i malen.

## 2.4 Begrensninger

### 2.4.1 Software design

**Software standarder og språk** Websiten vil utvikles slik at det er enkelt å kunne endre utseende senere. Funksjonaliteten på sidene vil bli kodet i PHP. Dette har flere fordeler, blant annet at kommunikasjon mot en database er veldig enkelt i PHP. Vi vil ta i bruk et utviklingsverktøy som heter Zend Studio for å programmere i PHP. Når det gjelder å generere CV-en så vil vi bruke et PHP-bibliotek som heter fpdf.

**Software kommunikasjonsstandard og grensesnitt** Kommunikasjon mot databasen fra websiten vil skje med PEAR, PHP Extension and Application Repository. Dette forenkler kommunikasjonen og gjør den mer robust. Det er ikke klarlagt enda fra oppdragsgivers side hvilket eksternt betalingssystem som skal benyttes ved betaling av en CV-mal.

**Database** Vi kommer til å bruke en MySQL-database. Grunnen til at vi velger akkurat denne typen database er fordi den er gratis, utbredt og generelt veldig god. Den er veldig godt dokumentert og grensesnittet mot PHP er enkelt å implementere.

**Operativsystem** Server vil kjøre på Linux i testfasen, mens websystemet kan aksesseres fra hvilket som helst operativsystem som har en nettleser.

## 2.5 Aspekter omkring livssyklus

### 2.5.1 Dokumentasjon

Websystemet skal kunne gi brukere hjelp om det er nødvendig.

### **2.5.2 Integrasjonstesting**

Vi utvikler inkrementelt og vil teste og integrere hvert inkrement etter hvert som det blir ferdig. Testingen gjennomfører vi fra en PC som kjører Windows XP plassert innenfor HIG-nettverket, siden serveren websiten ligger på er plassert på en studentbolig innenfor skolens nettverk. Vi vil legge ut versjoner på en ekstern server med jevne mellomrom slik at de som er interessert kan teste ut systemet.

### **2.5.3 Krav til utvidelser**

Websystemet skal lages slik at det siden er enkelt å bytte ut selve utseendet. Når det gjelder format for designmal er dette også mulig å bytte ut. Formatet skal være i PDF, og en eventuell utskifting av dette formatet vil medføre en del endringer ved at man må bygge en modul for å flette sammen den nye typen designmal sammen med CV-data. Vi har valgt å bruke PEAR i forbindelse med tilkobling til database, noe som gjør det enkelt å skifte database siden.

## **2.6 Aspekter omkring installasjon**

### **2.6.1 Hardware installasjon**

Det er ikke klargjort fra oppdragsgivers side når han vil sette systemet ut i drift. Hardware som trengs er en server med MySQL-database og en web-server der websiten skal ligge.

### **2.6.2 Overgang**

Dette systemet skal ikke erstatte et allerede eksisterende system hos oppdragsgiver.

### **2.6.3 Opplæring**

Bruken av systemet skal være lett for alle parter. Opplæring er heller ikke en vesentlig del av prosjektet. Oppdragsgiver som muligens vil ikle rollene som hovedadministrator og redaktør i første omgang, har god innsikt i hva systemet så skal gjøre, så han kan lære bort dette videre siden hvis han skal sette bort driften av systemet. Det vil ikke skje opplæring av vanlige brukere siden bruken av systemet skal være enkelt, og det er mulig å få hjelp på websiten.

## **2.7 Utgivelser underveis**

Websiten vil ligge ute etter hvert som vi tester systemet, men offisielle utgivelser underveis vil ikke forekomme.

## **2.8 Akseptansekrav**

Hver modul vil bli ansett som fullført når Use Casene som tilhører en modul er fullført, se punkt 2.3.1.

## **2.9 Prosjektstyring**

Kommunikasjon med veileder vil i utgangspunktet skje via møte annenhver uke. Vi vil også ha jevnlig kontakt med oppdragsgiver for å underrette om progresjonen i prosjektet.

### **2.9.1 Kvalitetssikring**

All kode skal dokumenteres i den grad det er nødvendig med PHPDoc- og Javadoc-standard. Det vi anser som triviell og selvforklarende kode blir ikke kommentert. Vi vil underveis gi veileder noe av dokumentasjonen for prosjektet, til gjennomlesing for å kunne gi kommentarer.

Vi har en CVS-server der vi vil legge alt arbeidet underveis. Denne vil bli tatt daglig backup av. I tillegg vil alle kjøre en update fra serveren ved dagens slutt, slik at arbeidet alltid vil være på minst 3 maskiner i tillegg til CVS-serveren.

## 3 Design

I denne delen av rapporten skal vi forklare hvordan vi har tenkt for oss systemet med hensyn på generell arkitektur og informasjonsbehandling. På dette tidspunkt utviklet vi en database, beskrevet nedenfor med forklaringer og ER-modell, og senere klassediagram for de klassene vi brukte på websiten. Etter hvert som vi jobbet med prosjektet ble det klart at det var fornuftig å inkludere en editor-modul i systemet, hvis klassediagram er beskrevet nedenfor.

### 3.1 Database

Før vi satte i gang designet av databasen hadde vi forestilt oss hvordan denne skulle se ut. Vi brukte en top-down teknikk der vi satte opp de tabeller vi syntes skulle være med og deretter sjekket vi om databasen tilfredstilte kravene til normalisering. Data i databasen er lagret i klartekst unntatt passordene som er hashet i MD5-form. Figuren på side 18 viser databasens tabeller, vist med primærnøkler i fet kursiv. Figuren på side 19 viser sammenhengen mellom tabellene. Kun primærnøkler er vist i tabellene på denne figuren.

#### 3.1.1 Tabellen «bruker»

Denne tabellen tar vare på en brukers personlige data og et bilde av vedkommende. Attributtet «aktiv» er det gjort en del diskusjon rundt. Det dreide seg om hvordan vi skulle utestenge useriøse brukere og å kunne gjenopprette brukere etter at de har slettet sin konto og vil ha den opprettet igjen. En nyregistrert bruker av systemet må aktivere sin brukerkonto via en e-post som inneholder en aktiveringslink som er unik for den brukeren. En bruker som ikke oppfyller de krav som settes vil kunne få sperret sin konto midlertidig ved å sette kontoen til å ikke være aktiv. Når en bruker vil ha seg selv slettet fra systemet så settes kontoen til ikke aktiv, det vil med andre ord si at brukeren kan angre og få kontoen åpnet igjen såfremt administrator ikke har slettet inaktive kontoer i mellomtiden. Prøver man å logge inn på en konto som ikke er aktiv får man da beskjed om at kontoen er deaktivert.

#### 3.1.2 Tabellen «designsøknad»

Her tas søknader om å bli designere vare på, og en søknad slettes når denne blir innvilget eller slettet.

#### 3.1.3 Tabellen «mal»

CV-maler som designere lager med hjelp av CV-maleditoren vil bli lagret her. Et preview av malen tas også vare på her. Attributtet «godkjent» ble



bruker	
<b>brukerid</b>	integer
opprettet	date
brukernavn	varchar(30)
brukertype	enum('1','2','3','4')
passord	varchar(32)
fornavn	varchar(30)
etternavn	varchar(30)
adresse	varchar(30)
postnr	varchar(4)
epost	varchar(30)
fodselsdato	date
kjonn	enum('M','K')
yrkesid	integer
tlfnr	varchar(15)
mobnr	varchar(15)
bilde	longblob
bildemime	varchar(128)
bildestorrelse	integer
bildefilnavn	varchar(64)
aktiv	char(1)

designsoknad	
dato	timestamp(14)
<b>brukerid</b>	integer
soknad	text

malrettighet	
<b>brukerid</b>	integer
<b>malid</b>	integer
betalt	char(1)

post	
<b>postnr</b>	varchar(4)
poststed	varchar(30)

yrke	
<b>yrkesid</b>	integer
yrkesgruppe	varchar(50)

erfaring	
<b>erfaringsid</b>	integer
brukerid	integer
fradato	varchar(10)
tildato	varchar(10)
arbeidsgiver	varchar(50)
stilling	varchar(50)
beskrivelse	text

mal	
<b>malid</b>	integer
brukerid	integer
kjopt	integer
mal	longblob
malmime	varchar(128)
malstorrelse	integer
malfilnavn	varchar(64)
dato	date
previewmalliten	longblob
previewmal	longblob
previewmalmime	varchar(128)
godkjent	char(1)

utdanning	
<b>utdanningsid</b>	integer
brukerid	integer
fradato	varchar(10)
tildato	varchar(10)
sted	varchar(50)
grad	varchar(50)

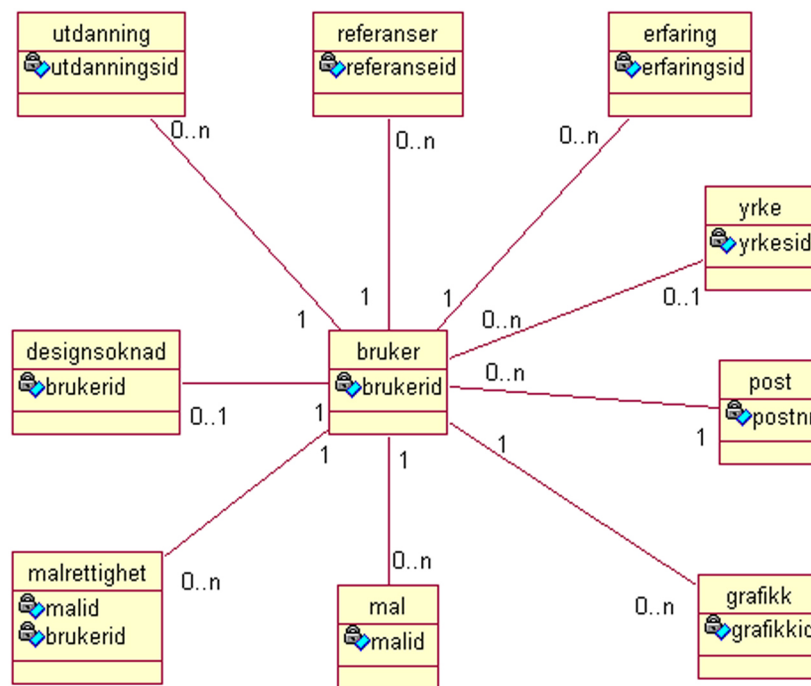
  

referanser	
<b>referanseid</b>	integer
brukerid	integer
navn	varchar(50)
arbeidssted	varchar(50)
stilling	varchar(50)
tlfnr	varchar(10)
epost	varchar(50)
beskrivelse	text

grafikk	
brukerid	integer
<b>grafikkid</b>	integer
grafikkbilde	longblob
grafikkpreview	longblob
grafikknavn	varchar(60)
brukt	char(1)

Figur 3: Tabellene i databasen.



Figur 4: ER-modell for databasen.

opprettet etter en diskusjon i gruppa. Attributtet er satt til false(0) som default. En mal som lastes opp blir ikke tilgjengelig for kjøp, ei heller vises den på designeres liste over maler før en redaktør eller administrator har godkjent malen, det vil si at attributtet er satt til true(1). Dette er gjort slik fordi malen skal være godkjent før man skal kunne kjøpe rettigheten til den. Dette medfører at en mal som er blitt godkjent kan ikke slettes fra systemet, da dette vil være ødeleggende for brukere som har kjøpt rettigheten til malen.

#### **3.1.4 Tabellen «malrettighet»**

Brukere av systemet kan kjøpe rettighet til en CV-mal. Dette lagres i denne tabellen.

#### **3.1.5 Tabellene «utdanning», «erfaring», «referanser» og «yrke»**

I de 3 første tabellene lagres en brukers utdanning, erfaring og referanser, men i den siste er en liste over yrkesgrupper en bruker kan velge at han arbeider innen.

#### **3.1.6 Tabellen «post»**

Denne tabellen inneholder samtlige postnr og poststeder i Norge.

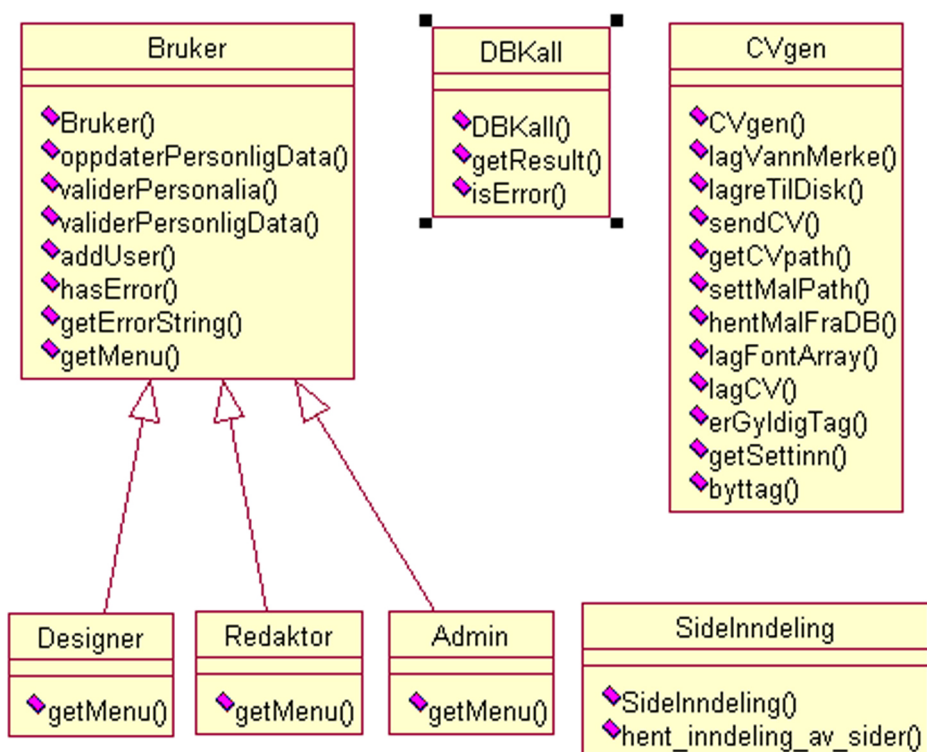
#### **3.1.7 Tabellen «grafikk»**

Grafikk som designere skal bruke til CV-malen ligger lagret her.

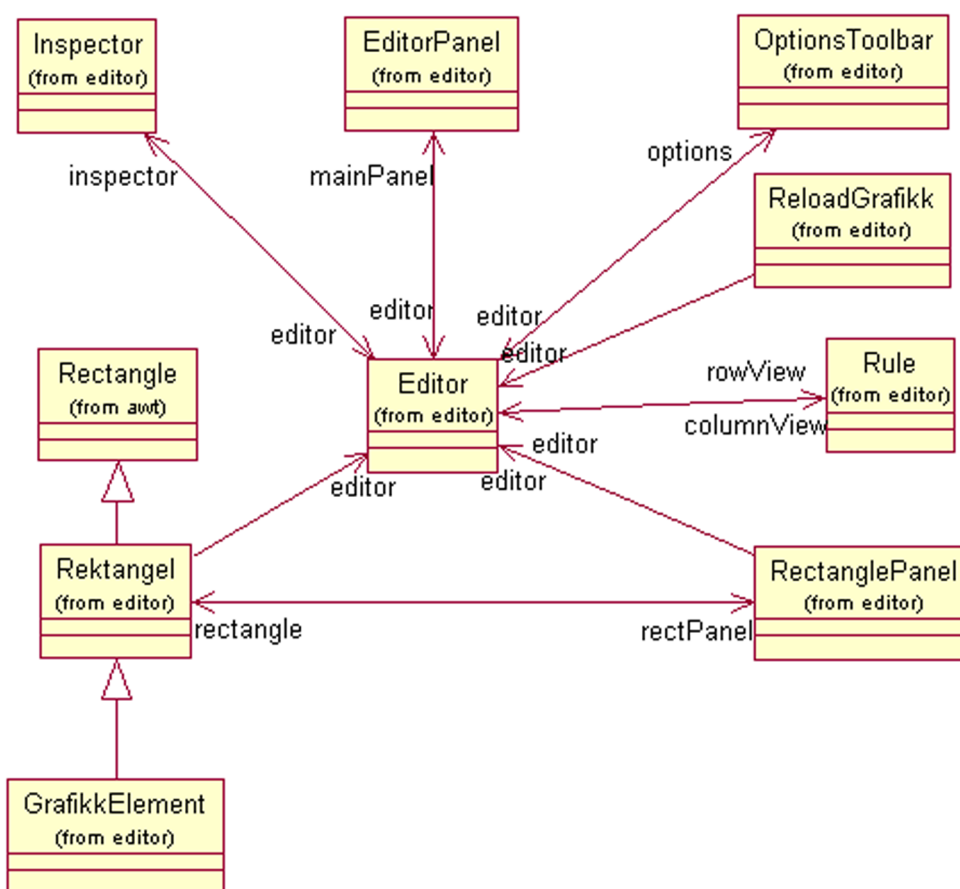
En bruker som får endret sin brukertype, for eksempel fra bruker til designer mister ikke sine CV-data, men de er ikke synlige lenger. Hvis han vil gå tilbake til å være en vanlig bruker, er dette mulig siden man da endrer brukertypen og brukeren får da tilgang til sine CV-data igjen.

### **3.2 Website**

Med utgangspunkt i kravspesifikasjonen, prøvde vi å kartlegge hva vi trengte av forskjellige sider for å dekke kravene vi hadde utarbeidet. Det vi gjorde var å lage en mockup på en «sitemap» av systemet. Det vil si å skissere et diagram med alle websidene vi kom til å trenge, og deres sammenhenger. På figur 5 side 21 vises klasser vi selv har laget i forbindelse med webiten. Mer om hva disse brukes til kommer vi tilbake til i implementasjonskapitlet. Vi har ikke vist attributter som funksjonene tar imot, dette fordi noen av funksjonene tar imot veldig mange attributter som ville sprengte figuren.



Figur 5: Klasser brukt på websiten.



Figur 6: Klasser i CV-maleditoren.

### 3.3 CV-maleditor

På figur 6 side 22 er klassene i CV-maleditoren, som er en Java Applet. Vi har ikke tatt med funksjoner siden det i noen av klasse var ganske mange, og dette ville gjøre klassediagrammet mer uoversiktlig.

## 4 Implementering

Her skal vi beskrive utviklingsmiljøet, ulike verktøy vi har benyttet og hvordan kodingen har foregått.

### 4.1 Utviklingsmiljø og programvare

Webserver, databaseserver og CVS-server har vært på samme maskin under utviklingen, men ulike versjoner av systemet har ligget ut på flere forskjellige servere, deriblant phpweb på HIGs nettverk. Grunnen til dette er at førstnevnte server er stasjonert på en hybel på Kallerud og er kun tilgjengelig fra student-PCer på skolen.

Kodingen av websiten og kommunikasjon mot database har foregått med PHP. Et bibliotek som heter PEAR ble brukt i forbindelse med kommunikasjon mot databasen, siden det da er enklere å skifte ut databasetypen siden.

#### **L<sup>A</sup>T<sub>E</sub>X**

L<sup>A</sup>T<sub>E</sub>X er et skriveverktøy der det man skal skrive kodes i L<sup>A</sup>T<sub>E</sub>Xkode. Det er et kraftig, robust og velutviklet verktøy som håndterer det som de fleste andre skriveverktøy kan gjøre. Det fine med L<sup>A</sup>T<sub>E</sub>X er at man ikke trenger å bry seg så mye om utseendet på det resulterende dokument, siden dette tas hånd om av verktøyet, men heller tenke mer på innhold. 2 av gruppemedlemmene hadde vært borti L<sup>A</sup>T<sub>E</sub>X før og da falt valget på dette når det gjaldt skriving av rapport.

#### **MsProject**

Dette er et velkjent prosjektstyringsprogram, som vi har benyttet oss av for å lage et Gantt-diagram. Et slikt diagram er en tidstabell som viser hva som må gjøres, når det skal gjøres, og hvor lang tid det tar. Siden vi har valgt inkrementell utvikling som modell, fant vi ut hvilke moduler systemet ville bestå av og definerte dem som inkremitter. Disse ble, i tillegg til øvrige arbeidsoppgaver, satt opp som blokker i rekkefølge, fra prosjektstart til det ferdige produktet. Dette inkluderer da også milepæler og deadlines. Man kan se det planlagte Gantt-diagrammet på side 41.

#### **CVS**

Concurrent Version System er et verktøy for versjonshåndtering. Dette er nødvendig for at alle skal ha tilgang til de nyeste versjoner av samtlige filer.

### **Adobe Photoshop CS og GIMP**

Bildebehandlingsprogrammet Photoshop ble brukt for å skalere bildene som skulle inn i rapporten til riktig størrelse og til å lage grafikk, både for websiten og prosjekt-presentasjonssidene. GIMP ble brukt til å konvertere bildene til rapporten til eps-format, som er formatet som L<sup>A</sup>T<sub>E</sub>X støtter best.

### **Adobe GoLive**

Dette programmet ble brukt til bygge CSS-filen som websiden bruker til å formatere utseendet.

### **Zend Studio**

Zend er et komplett utviklingsmiljø for PHP. Versjonen vi brukte hadde innebygd CVS, noe som gjorde det enkelt for oss ved at vi lett kunne legge det vi hadde laget ut på webserveren for så å teste det. Vi fikk lisenser av oppdragsgiver til dette programmet.

### **Rational Rose Enterprise Edition**

En del av Rational Software som er et systemutviklingsverktøy med støtte for UML. Vi har brukt dette til å lage Use Case- og klassediagram.

### **TextPad**

Redigerings- og kodingsverktøy som støtter syntaks highlighting. Vi brukte dette til å kode editoren.

## **4.2 Koding**

### **4.2.1 Database**

Implementering av database gjorde vi med phpMyAdmin. phpMyAdmin er et PHP-basert databasehåndteringssystem for MySQL. Dette er et veldig nyttig verktøy, for det gjør jobben med å ajourføre databasen mye lettere enn å skrive all SQL-koden manuelt.

### **4.2.2 CV-mal**

Vi hadde litt problemer med å lese en CV-mal. Det var problematisk å finne igjen de ulike tagger i CV-malen. Hvordan malen ble bygget opp var avhengig av hvilket program den ble laget i. Se for øvrig figur 7 side 25 på hvordan en CV-mal ser ut. Vi prøvde å lage en CV-mal i Adobe Acrobat og Adobe Indesign. Målet var å kunne finne igjen posisjonene til tagger i malen slik at CV-data fra databasen kunne skrives til disse. Siden PDF-kildekoden var avhengig av hvilket program som bygget opp malen så bestemte vi oss for

<b>Personalia</b>				<b>!bilde!</b>
<b>!navn!</b>				
<b>!adresse!</b>				
<b>!postnrsted!</b>				
<b>!tlf!</b>				
<b>!mob!</b>				
<b>!epost!</b>				
<b>!fdato!</b>				
<b>Utdanning</b>				
<b>!utdsted!</b>		<b>!utdgrad!</b>	<b>!utdtildato!</b>	<b>!utdfradato!</b>
<b>Erfaring</b>				
<b>!erffradato!</b>	<b>!erftildato!</b>	<b>!erfarbeidsgiver!</b>		
		<b>!erfstilling!</b>		
		<b>!erfbeskrivelse!</b>		

Figur 7: Eksempel på en del av enkel CV-mal.



å kun bruke et program til å lage CV-malen. Med en mal laget i fpdf fant vi ganske greit igjen taggene og vi bestemte oss for at malen skulle lages med fpdf. Hver tag er inne i et rektangel siden det var mye enklere å få ta i x og y-koordinaten til rektangelet enn selve taggen ved lesing av malen. Problemet som da oppstod er at malen må kodes i PHP med hjelp av fpdf. Det ble derfor bestemt at vi skulle lage en enkel editor for å kunne lage en slik designmal. Editoren sender da sine data til et PHP-script som bygger opp malen ved hjelp av fpdf. Vi fant det også hensiktsmessig å lage en editor siden vi lå en del foran tidsskjemaet som var satt opp.

### 4.2.3 Websiten

Det har blitt lagt vekt på utseende på websystemet, siden det er viktig at de som tilbyr profesjonelle CV-maler har en side som gir uttrykk for at de kan noe om design og estetikk. Det betyr at vi har gjort oss flid med å tilpasse hvert grafikk-element i websiten til å passe inn med hverandre. Men det har seg slik at hva som er «hot, and what's not» forandrer seg veldig fort på Internett. Dette gjelder i hovedsak selve utseendet, så vi har prøvd å sentralisere innstillinger for grafikken gjennom bruk av CSS. Vi har her plassert det meste av formatering av html og bruk av grafikk. En slik sentralisering av innstillingene vil gjøre det lett for oppdragsgiver å gjøre både store og små forandringer gjennomførende på hele websiten. Vi benyttet oss av Zend Studio for å kode websiten. De fleste filene er i PHP-format.

Autentisering og autorisering av brukere er viktig på websiten slik at de riktige sidene vises til riktige personer. Når en bruker logger inn kontaktes databasen og man får svar hvilken brukertype vedkommende er. Dette lagres i en session. Når en side skal vises så sjekker man om brukertypen i session-et er riktig, hvis ikke gis beskjed om at man ikke har tilgang til siden. Vi har brukt en klasse som heter login for å forenkle sjekkingen rundt dette. Login-klassen har en metode for å hente ut brukertypen fra session-et. Et eksempel på hvordan dette er gjort vises under.

```
if($login->getBrukerType() == DESIGNER){
    ...
} else {
    echo 'Kun innloggede designere kan aksessere denne siden';
}
```

Vi har opprettet klasser av de forskjellige brukertyper som arver fra en hovedklasse som heter bruker, se figur 5. Det opprettes et objekt av en slik klasse når en bruker logger inn. Hver av klassene har en funksjon for å returnere en meny. Ved hjelp av polymorfisme henter man enkelt ut menyen som hver brukertype skal få etter innlogging avhengig av hvilken klasse objektet er opprettet av.

DBkall-klassen benyttes hver gang man skal kommunisere med databasen.

Ved opplisting av elementer på en side som gjør at siden blir lang slik at man må scrolle har vi løst dette med en klasse som heter SideInndeling. Denne deler siden slik at et bestemt antall elementer vises på en side, samt gir navigasjon til videre sider for å flere elementer.

CVgen-klassen brukes når man skal lage en CV. Det som skjer i denne klassen er at CV-malen som skal brukes lastes ned fra databasen. Deretter leser man CV-malen med hjelp av fpdi, finner igjen de ulike taggene med tanke på posisjon, størrelse, font og hvilken tag den inneholder. Deretter kontaktes databasen for å hente ut riktig data i forhold til taggen og denne dataen settes inn i CV-en der taggen stod. Når det gjelder grafikk fungerer dette på samme måte som for en vanlig teksttag. I stedet for tekst så henter man ut bilde fra databasen og setter dette inn i CV-en.

Under følger en oversikt over filene på websiten vi har laget selv. Disse har blitt navngitt ut fra hva de skal gjøre eller inneholder.

<b>Filnavn</b>	<b>Beskrivelse av fil</b>
admbrukere.php	Her skjer administrasjon av brukere. Brukeren kan her gis nye rettigheter i systemet, ved å gi dem status som en annen brukertype. Å sette brukerkontoer aktive eller inaktive skjer også her.
aktiver.php	Aktivering av nyregistrerte brukere.
bruker.php	Inneholder klassene bruker, designer, redaktør og admin.
config.php	Konfigurasjonsfil som gjør flytting til en annen server lettere.
cvdef.xml	Inneholder CVdata-tagger til bruk i editor.
cvgen.php	Genereringen av CV-PDF-en.
db_init.php	Klasse om oppretter en tilkobling til databasen.
dbkall.php	Klasse som kommuniserer med databasen.
endrepassord.php	Brukere av systemet kan endre sitt passord her.
endrepersonligedata.php	En bruker kan endre sine personlige data. Dette gjelder også designere, redaktører og administrator.
epost.php	Klasse for å sende epost.
erfaring.php	Bruker kan fylle inn/slette sine arbeidserfaringer.
finnesikke.php	Inneholder feilmelding hvis bruker prøver å aksessere en side som ikke finnes.
faanyttpassord.php	Brukere av systemet kan her få et nytt passord ved å oppgi sin e-postadresse.
generermal.php	Generer en CV-mal ut fra data som fåes fra mal-editoren.
hentgrafikk.php	Henter grafikk fra databasen og sender til editor.
index.php	Hovedsiden for websiten.
innlogget.php	Hovedside for innloggede brukere.
kontomeny.php	Den innloggede brukers meny. Hver brukertype har ulike menyer. Disse menyene fåes fra filen og klassene i bruker.php.
kvittering.php	Tar imot kvittering etter betaling av CV, setter malrettighet.
lastoppbilde.php	Her kan man laste opp et bilde til CV-en.
lastoppmal.php	Laster editoren til å lage en CV-mal.
loggut.php	Logger brukeren ut av systemet og sletter tilhørende sessions.

<b>Filnavn</b>	<b>Beskrivelse av fil</b>
login.php	Tar seg av innlogging av brukere. Inneholder klassen login.
maloversikt.php	Designere kan se oversikt over sine maler, samt se hvor mye de har tjent.
nyemaler.php	Lar designer se en oversikt over og kan godkjenne evt nye designmaler.
okorapport.php	Viser en økonomisk rapport for administrator.
previewmal.php	Sammen med vismaler vises her et preview av en CV-mal.
referanser.php	Bruker kan fylle inn/slette sine arbeid-referanser.
registrer.php	Innfylling av data for nyregistrering av brukere skjer her.
sedesignsoknader.php	Redaktør/administrator kan lese design-søknader og innvilge eller slette disse.
sideinndeling.php	Inneholder en klasse med metoder som kan dele en side i flere sider og gir navigering mellom disse.
skrivinncvdata.php	Side for brukere som omfatter personlige data og utdanning, referanser og erfaring.
slett.php	Ved sletting av utdanning, referanser eller erfaring brukes denne filen.
slettmeg.php	Bruker kan slette seg selv og blir logget ut. Brukerkontoen settes inaktiv.
sokomdesigner.php	En bruker kan sende inn søknad om å bli designer.
textbox.php	Brukes når man lager en mal.
utdanning.php	Bruker kan fylle inn/slette sine utdan-ninger.
velgcvdata.php	Lar bruker velge hvilke CV-data som skal være på CV-en.
velkommen.php	Velkomstsider som vises for alle brukere.
vellykket.php	Etter vellykket kjøpt av mal videresendes man hit.
visallemaler.php	En redaktør eller administrator kan se en oversikt over alle designmaler.
visbilde.php	Viser et bilde av brukeren sammen med personlig data.
viscv.php	Viser et preview av en CV med brukerd-data.
viscvpreview.php	Sammen med viscv.php viser denne et pre-view av en CV.

Filnavn	Beskrivelse av fil
visgrafikk.php	Viser grafikk som er tilgjengelig for designere å bruke i editoren.
viskjoptemaler.php	Viser oversikt over maler en bruker har kjøpt rettigheten til.
vismal.php	Viser en forhåndsvisning av en designmal en bruker kan kjøpe rettigheten til.
vismaler.php	Viser selve previewet av en mal.
xmlparser.php	Parser en XML-fil.

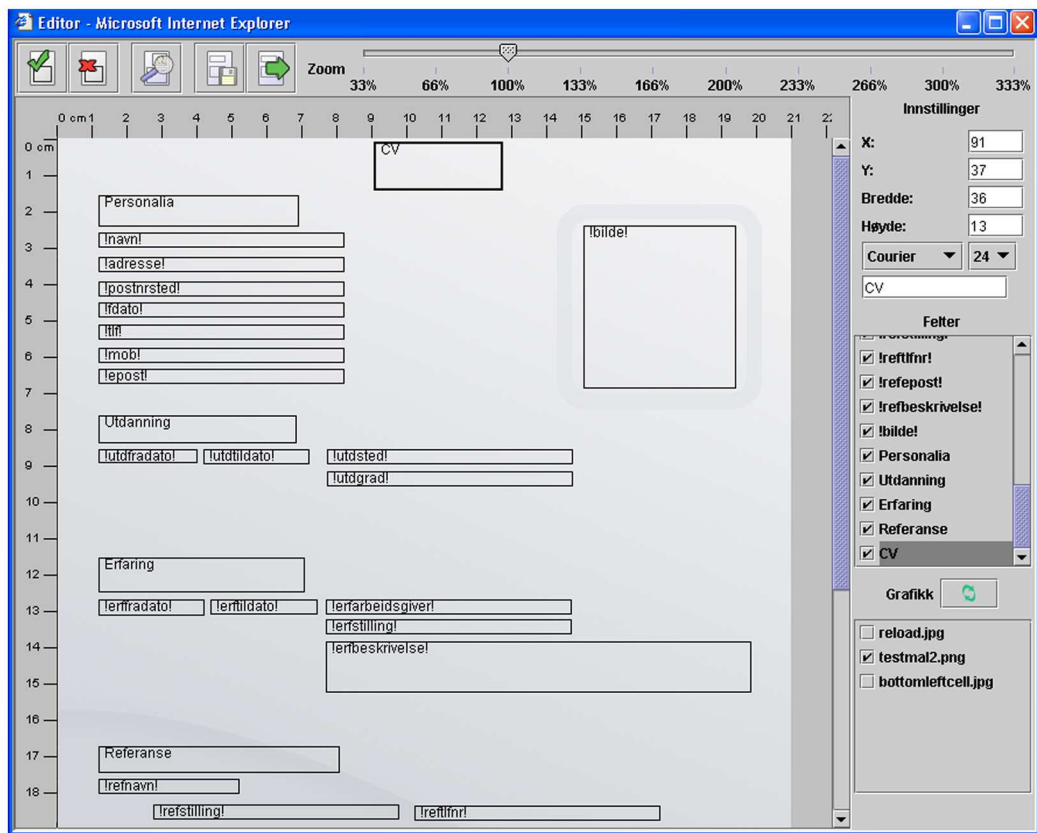
#### 4.2.4 CV-maleditor

Vi var litt usikre på hva vi skulle kode editoren i. Etter møte ved oppdragsgiver kom frem det at valgene stod mellom Javascript og Java i en Java Applet. Oppdragsgiver hadde mest erfaring med Javascript og foreslo dette. Ingen av oss hadde noe erfaring med Javascript, men vi kunne alle Java. Dette talte for at vi burde velge en Java Applet. I tillegg skulle vi ha en del klasser, samt arv. Dette er ting som lar seg løse lettere i en Java Applet enn med Javascript, da arv og klasser er noe tungvint i sistnevnte språk. I tillegg var vårt personlige synspunkt at vi kunne klare å lage en bedre editor med en Java Applet. Valget falt derfor på en Java Applet.

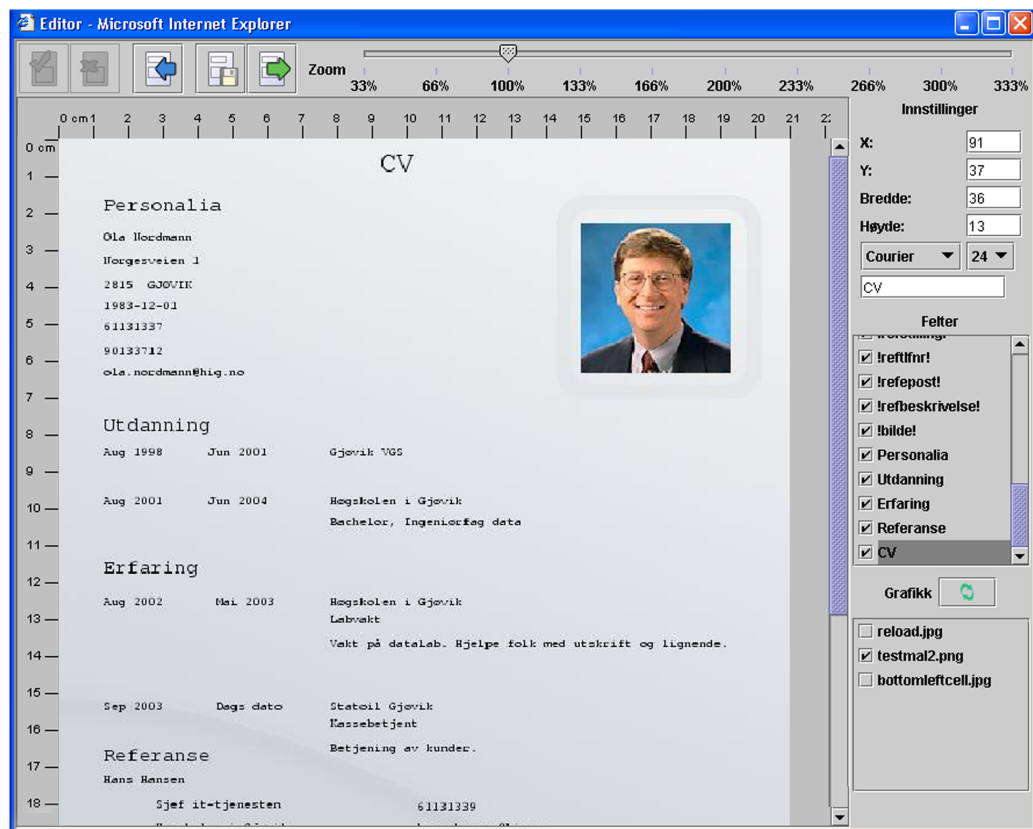
Et bilde av editoren er vist på figur 8 side 31. Alle tagger som er tilgjengelig er plassert i editoren inni rektangler. Disse taggene leses fra en XML-fil. Disse kan flyttes rundt etter eget ønske om hvordan malen skal se ut. Man kan merke en eller flere rektangler og så sette egenskaper i inspectorfeltet på høyre side. Med inspectorfeltet menes det området der man ser x, y, bredde, høyde, font og tekststørrelse for et rektangel. Egenskaper som et rektangel har er x- og y-posisjon, bredde, høyde, font, størrelse, samt en tag. En tag er for eksempel «!navn!» Hvert rektangel har en avkrysningsboks, plassert i høyre del av editor, som avgjør om rektangelet er synlig eller ikke. Det kan hende at designere velger å ikke ha med alle rektangler som er tilgjengelig. Det er mulig å legge til egne rektangler med egendefinert tag. Dette er da en tag som ikke vil byttes ut med CV-data fra databasen, men blir til tekst som for eksempel «Navn:». Disse egendefinerte taggene er tagger som ikke har «!» før og etter. Det er valgt å sette på «!» på tagger som skal byttes ut med CV-data slik at man ser forskjell på disse taggene og de egendefinerte taggene. Det er brukt forkortelser i taggene, «utd», «erf» og «ref» står for henholdsvis utdanning, erfaring og referanser.

Man kan se et preview av malen man lager. Det lages da en CV med dummy-data fra en bruker som er fiktiv, og det dannes et bilde av CV-en og dette vises i editorvinduet. Se figur 9 side 32.

Når det gjelder dette med flere utdanninger, erfaringer og referanser må designeren sette av plass til dette. Det vil si at for eksempel utdanningstaggene står disse bare oppført en gang og det er tom plass et stykke



Figur 8: Bilde av editoren



Figur 9: Bilde av editoren med preview

under før de neste taggene kommer som på figur 7. Når CV-en genereres vil man så kunne sette inn flere utdanninger der det er satt av plass. Hvor mange utdanninger som settes inn er avhengig av hvor mange brukeren har og velger å sette inn.

Lagring av en mal underveis skjer i en cookie. Ved oppstart av editoren sjekkes det om denne cookien er satt. Hvis det er tilfelle så leses malen fra cookien.

Når designer er ferdig med malen sender han den inn til godkjenning. Det som skjer er at alle rektanglene som ligger i en vektor settes sammen til en lang tekststreng og sendes til et PHP-script som tar seg av oppbyggingen av CV-malen.

Grafikk som er tilgjengelig vises på en liste i nederst til høyre i editoren. Når en designer skal sette inn grafikk krysser han av i avkrysningsboksen og grafikken kommer opp i malen. Det som skjer er at man poster grafikfilnavnet til et PHP-script, som leser grafikken fra databasen og sender dette til editoren. Dette er gjort slik fordi hvis all tilgjengelig grafikk skal sendes ved oppstart av editor så ville dette være unødvendig tidkrevende.

De mest sentrale klassene i editoren er vist på figur 6 side 22. Hvert synlig rektangel i en mal er et objekt av klassen `Rektangel`. Denne arver fra Java klassen `Rectangle`. Vi har lagt til en del attributter. De viktigste er font, tekststørrelse og tag. Plassering av rektangel, samt bredde og høyde tas vare på i morklassen `Rectangle`. Hvert grafikkelement arver fra `Rektangel` igjen fordi grafikkelement også trenger en tag samt posisjon. Taggen til et grafikkelement består av `gfx:designerid:filnavn`. Når en CV skal genereres setter man inn riktig grafikkelement der taggen stod. Grafikkelementet hentes her også fra databasen. Mer om hvordan CV-en genereres kan leses i kapittel 4.2.5.

`RectanglePanel`-klassen viser hvert rektangel på høyre side i editor sammen med avkrysningsboksen. `OptionsToolbar`-klassen viser toolbaren med knapper helt øverst i editoren. `Editor`-klassen er hovedklassen som utgjør selve Appleten. `Rule` og `Corner` klassen utgjør til sammen linjalene som er synlig rundt malen. `EditorPanel`-klassen er arbeidsområdet for malen. `ReloadGrafikk`-klassen reloader listen over tilgjengelige grafikkelementer hvis det lastes opp mer grafikk til databasen mens editoren kjøres.

#### 4.2.5 CV-generering

Når bruker skal generere sin CV ut fra en mal får han listet opp sine utdanninger, erfaring og referanser, som har en unik id hver. Bruker krysser av for hvilke utdanninger og så videre som skal være med. Disse id-ene og id-en til malen postes til `cvgen.php` som tar seg av selve CV-genereringen. Malen som skal brukes hentes fra databasen og lagres midlertidig til disk. I constructoren til `cvgen`-klassen opprettes det en array over gyldige tagger. Dette er tagger som skal byttes ut med CV-data, og disse leses fra en XML-



fil, den samme som editor leser inn for å få overensstemmelse. For øvrig kan man lett utvide XML-filen til å inneholde flere tagger om ønskelig. Deretter brukes fpdf og fpdi til å parse malen og generere CV-en. Man finner igjen rektangelet i en mal, fonten som er brukt og taggen, for eksempel «!navn!». Deretter sjekkes det om taggen er gyldig og hvis den er det, så byttes den ut med CV-data. Hvis ikke blir den stående. All tekst som skal inn i CV skrives over der rektangelet stod. All personlig info kommer med på CV-en, men det er valgfritt hvilke utdanninger og så videre som skal være med. Sistnevnte er grupper som skal repeteres hvis flere er valgt. Dette skjer ved at man detekterer når man kommer til eksempelvis en utdanning. Da må man hente ut de utdanninger fra databasen som skal være med og sette inn disse i henhold til malen. I malen er det bare satt opp en utdanning, men det skal være satt av tom plass til flere.

#### **4.2.6 Paynet**

For å kode betalingstjenesten måtte vi sette oss inn i dokumentasjon for å implementere en Paynet-terminal. Når en bruker skal betale sendes han videre til Paynet sin server og foretar betalingen der. Etter fullført betaling sendes han så tilbake til websiten. På samme tidspunkt sendes det en kvittering til websystemet om betalingen gikk i orden. Dersom den gjorde det oppdateres malrettigheten for brukeren i databasen og han kan nå generere sin CV. Hvis bruker avbryter betaling underveis kommer han tilbake til en oversikt over maler han kan kjøpe. Slik betalingen foregår i dag benyttes kun en testkonto hos Paynet. Det vil si at det ikke belastes fra den betalendes konto. Skal man benytte seg av en reell konto hos Paynet må det opprettes en slik. Konfigurasjonen av en konto hos Paynet er beskrevet i config.php.

## 5 Testing

Testing har vært en gjennomgående prosess gjennom hele prosjektet for å sikre at systemet handler som forventet. Testingen har blitt utført av gruppen selv og noe av oppdragsgiver. Vi fikk etter hvert serverplass av arbeidsgiver og la ut versjoner av systemet. På denne måten kunne arbeidsgiver teste og kvalitetssikre arbeidet, samt komme med tilbakemeldinger om ting som skulle endres eller løses på en annen eller bedre måte.

Utviklingen av websiten var det vi startet med, og testingen gikk ut på å sjekke om det virket tilfredsstillende i forskjellige nettlesere, herav Internet Explorer 6.0, Mozilla FireFox, Opera. I forbindelse med Javascript ble det noen problemer underveis. Det var 2 av nettleserne som ikke registrerte riktige museklikkhendelser i avkryssningbokser på en html-side. Dette lot seg løse ved å skrive om koden litt. Integrasjonstesting har vært viktig på websiten. Vi sjekket at PHP-scriptene fungerte som de skulle hver for seg, og til slutt som en helhet. I Zend studio er det en funksjon som sjekker PHP-koden og gir råd hvis det skulle vise seg å være nødvendig. Denne benyttet vi oss av iblant.

CV-maleditoren ble også testet ut fortløpende, og feil man fant ble rettet opp der og da hvis det lot seg gjøre.

Utbetaling av provisjon til designere, samt at disse kan se hvor mye som de har tjent på sine maler og hvor mye de har tilgode er godt uttestet. Dette har foregått ved at vi har simulert flere brukere og kjøpt rettighet til maler og loggført dette med penn og papir og så sjekket om systemet kommer frem til samme resultat.

Testing av betalingsfunksjonen for kjøp av CV-mal foregikk på en testserver hos Paynet. Der kunne man sette opp en testkonto, slik at man det virket som man betalte, men det ble ikke trukket noe fra kontoen man betalte med.

Editoren virker med IE 6, Opera og Mozilla Firefox i Windows. Lagring av en mal i cookie virker ikke på de to sistnevnte. På Mac virker editoren i safari. For å få den til å virke i Mozilla Firefox på Mac må man laste ned <http://sourceforge.net/projects/javaplugin/>  
På Linux får vi ikke editoren til å starte opp.

Testing av website og editor tok utgangspunkt i Use Casene. Status for testene nedenfor gjelder for IE6 og Opera 8 på Windows XP.

<b>Teste hva</b>	<b>Status</b>
Registrere seg som ny bruker	OK
Opprette ny bruker som administrator	OK
Logge inn/ut	OK
Endre passord	OK
Få nytt passord	OK
Skrive inn CV-data og endre disse	OK
Laste opp bilde til bruk i CV	OK
Slette seg selv om bruker	OK
Kjøre rettighet til mal	OK
Generere og sende CV	OK
Administrere CV-maler	OK
Administrere brukere	OK
Administrere designsøknader	OK
Se statistikk for egne maler	OK
Se økonomisk rapport	OK
Sende CV-mal til godkjenning	OK
Zoom i editor	OK
Preview av mal i editor	OK
Lagring av CV-mal, samt innlesning igjen	OK
Grafikk i editor	OK
Setting av egenskaper, x,y, bredde osv for tagger	OK

## 6 Vurderinger og konklusjon

### 6.1 Vurdering av resultatet

Vi er fornøyd med utviklingen av prosjektarbeidet. Vi føler at vi har utviklet et system som er i henhold til kravspesifikasjonen. Underveis i arbeidet har vi mottatt ønsker om nye funksjoner til systemet fra oppdragsgiver, både i starten og mot slutten. Dette påvirket litt hva vi prioriterte å gjøre, blant annet brukte vi mye tid på editoren på slutten.

Vi har lagt vekt på å lage et funksjonelt system der websiten skal være selvforklarende for brukerne. Vi utvidet også systemet til å omfatte en CV-maleditor på grunn av at vi fikk bedre tid enn det vi først hadde beregnet.

Implementasjonen av PHP på websiten har gått bra. Ingen av oss hadde noe erfaring fra PHP fra før, men dette gikk veldig bra. Kommunikasjonen mot databasen gikk veldig greit med PHP, som har god støtte for kommunikasjon for MySQL.

### 6.2 Kritikk av oppgaven

Vi ser nå når vi er ferdig at ting kunne vært gjort annerledes. Slik som systemet er i dag lagres malen i en PDF-fil. Det hadde vært mer hensiktsmessig om denne ble lagret i en XML-fil da dette hadde medført mindre tid til å lese malen ved CV-generering. Vi så ikke før mot slutten da vi hadde fått mer erfaring at vi kunne lagret malen annerledes. Hadde vi hatt mer tid hadde vi muligens gjort om dette.

### 6.3 Alternative muligheter og valg underveis

Mye av funksjonaliteten for systemet var klarlagt av oppdragsgiver og nedfelt i kravspesifiseringen. Hva som skulle benyttes til å utvikle websiten kunne vi velge selv. Oppdragsgiver foreslo EZ-publish, men dette var et veldig stort og omfattende rammeverk så vi besluttet å bygge opp websiten helt fra bunnen av selv, både med tanke på innlogging, autentisering og resten av websiten.

### 6.4 Fremtidige utvidelser

Websiten er bygd opp slik at databasen enkelt kan byttes ut. Det samme med utseendet på websiten, dette bestemmes av en CSS-fil og vi har skilt funksjonalitet og utseende. Dette på grunn av at det er programmeringsmessig pent å gjøre dette, og for at oppdragsgiver kan bytte ut utseende senere om han ønsker.

Hvilke felter som er med i CV-en, bestemmes av en XML-fil. Dette gjør at man kan utvide editoren til å kunne inneholde andre felter enn kun felter som passer til en CV. Det som muligens kan forbedres siden er CV-maleditoren.

«Snap to Grid» funksjonalitet kan nevnes. Med det menes at man trekker opp en rett linje og elementer kan lett justeres etter denne.

Per i dag får man generert en CV som er på en side. Dette burde være tilstrekkelig for de fleste nyutdannede, men for de med flere mange års erfaring burde man muligens hatt flere sider.

Lagring av malen i XML er også noe å ta fatt i ved videre utvidelser. Når nye brukere registrerer seg i dag kan de velge en yrkesgruppe de tilhører. Dette er gjort med tanke på at det kan lages en søkefunksjon for arbeidsgivere siden.

## 6.5 Egenevaluering

Samarbeidet innad i gruppa har fungert meget bra. Ingen av oss hadde jobbet sammen i gruppe før. Det har ikke vært noen konflikter, bare faglige diskusjoner som hører til et slik gruppearbeid. Det var spennende i starten å se om gruppa fungerte bra. Vi har jobbet sammen 5 dager i uka som planlagt. I forprosjektrapporten tildelte vi forskjellige roller til gruppedeltakerne, og disse fulgte vi bra. Jon Anders og Bjørn hadde hovedansvaret for web-siten, mens Thomas hadde ansvaret for editoren. Vi valgte å fordele ansvarsområder på denne måten slik at arbeidet ble mer effektivt. Hadde vi latt ansvarsområdene gå på omgang kunne dette bli ineffektivt.

Valget av systemutviklingsmodell har vi vært fornøyd med. Inkrementell utvikling gjør at man hele tiden har kontroll over progresjonen, og det er et moderat papirarbeid til denne modellen. Vi har daglig ført logg over arbeidet som er gjort og notert timer. Antall timer som er jobbet til sammen ble 1286, og dette består av research, læring av nye programmeringsspråk, nettsøk, planlegging, koding og skriving av rapport.

## 6.6 Konklusjon

Vi har nå jobbet siden januar med prosjektet. Det har vært en interessant og lærerik periode. Vi har fått god erfaring i det å jobbe i gruppe mot et samlet mål. Den prosessen man gjennomgår ved gjennomføring av et hovedprosjekt og de erfaringer man opparbeider seg er nyttige å ta med seg videre da prosjektarbeid er noe vi kommer til å møte i arbeidslivet eller ved videre studier.

Vi anser prosjektet som vellykket.

## 7 Litteraturliste

Jon Anders Øvern et al. *GetAJob*. Studentarbeid  
Gjøvik: Høgskolen i Gjøvik, 2004

Kodesnutter og kode vi har benyttet oss av er referert til i kildekoden der det er blitt brukt. Se forøvrig kilder.txt på CD-rom.

## A Definisjoner

Bruker	En bruker av systemet. Det vil si en vanlig bruker, designer, redaktør eller en administrator.
Cookie	En fil som lagres på klientsiden.
CSS	Cascading Style Sheets. En utvidelse av HTML for å lettere formatere et dokument.
Designmal	Her en CV-mal. Generelt en mal for hvordan et dokument skal se ut.
fpdf	Open-source bibliotek for å generere en PDF.
fpdi	Open-source bibliotek for å lese en PDF.
MD5	Message Digest Algorithm #5. Protokoll for å generere en hash.
Mockup	En enkel skisse.
Normalisere	Formell metode for design av database for å unngå dobbeltlagring og selvmotsigende data.
Parse	Bryte ned en datamengde til dens grunnleggende elementer. Eksempelvis er alle ordene i denne teksten slike elementer.
Polymorfisme	En variabel kan referere til forskjellige objekter i et klassehierarki. Denne variabelen kan velge ut en redefinert metode i en av objektene.
Session	En session er kommunikasjon med bevaring av tilstand med en start og slutt, som for eksempel inn- og utlogging.
Sitemap	Kart som viser sammenhengen mellom ulike sider på en website.
System	Hele systemet, det vil si database, website og editor.
Top-down teknikk	Her brukt som å bestemme tabellene i databasen før attributtene. Man ser litt større på problemet.
Website	Selve websidene for systemet.