



## Forord

Softcross er utviklet av 3 studenter ved Høgskolen i Gjøvik våren 2001. Arne Sunde har vært oppdragsgiver. Han har skrevet hovedfags oppgave om kundeorienterte websystemer og ville lage et kundebehandlingssystem med grunnlag i hovedoppgaven. Arne Sunde bor i Oslo og jobber i EITeleØst.

Vi valgte oppgaven Softcross som hovedprosjekt fordi vi syntes det var en utfordrende og inspirerende problemstilling. Vi er godt fornøyde med å ha gjennomført et slikt prosjekt. Det har vært lærerikt og spennende.

Vi ønsker å rette en takk til Arne Sunde som har vært oppdragsgiver. Sunde har stor kompetanse innen fagfeltet og har kommet med gode innspill og tilbakemeldinger underveis. Han har også kurset oss i ColdFusion som vi har utviklet det meste av produktet i. Vi vil også takke Tom Røise som har vært veileder. Røise har kompetanse innen systemutvikling og har vært til stor hjelp under oppgaven. Vi takker også IT-tjenesten for lån av PC som vi satte opp som serveren Thecross. Ønsker også å rette en takk til alle som har vært med som testere av systemet.

Gjøvik 23.mai 2001

Jørn Arild Andenæs  
(Prosjektleder)

Børge Johnsen  
(Utviklingsmiljø og  
rapport ansvarlig)

Kristoffer Jørdre  
(Dokumentansvarlig)

## Innholdsfortegnelse:

<b>Forord</b> .....	<b>3</b>
<b>1 Innledning</b> .....	<b>5</b>
1.1 Oppgavedefinisjon:.....	5
1.2 Målgruppe.....	5
1.3 Egen kompetanse.....	5
1.4 Arbeidsformer:.....	6
1.5 Ord og uttrykk brukt i rapporten:.....	6
1.6 Organisering av rapporten .....	7
<b>2 Kravspesifikasjon</b> .....	<b>8</b>
2.1 Orginal kravspesifikasjon.....	8
2.1.A Introduksjon .....	8
2.1.B Brukerbeskrivelse.....	11
2.1.C Detaljert kravspesifikasjon.....	15
2.2 Analyse av arbeidet ved å komme frem til kravspesifikasjon. ....	24
<b>3 Design</b> .....	<b>25</b>
3.1 Systemarkitektur.....	25
3.2 Databasestruktur .....	27
3.3 Filstruktur .....	28
3.4 Layout .....	29
3.5 Programmering av designet.....	32
3.6 Navigasjonsstruktur.....	33
<b>4 Implementering</b> .....	<b>35</b>
4.1 Utviklingsmiljø .....	35
4.2 Valg av verktøy.....	36
4.2.A Programmeringsspråk.....	36
4.2.B Database.....	38
4.2.C Prosjektstyringsverktøy .....	38
4.2.D Systemutviklings verktøy.....	38
4.2.E Versjons kontroll system.....	38
4.3 Kode og Filstruktur.....	38
4.3.A Prinsipper ved kode.....	38
4.3.B Overordnet kode .....	39
4.3.C Filstruktur.....	41
<b>5 Testfasen - evaluering</b> .....	<b>44</b>
<b>6 Diskusjon av resultater</b> .....	<b>46</b>
6.1 Prosessen .....	46
Tidlig kode.....	46
Koden slik vi kodet det til slutt .....	46
6.2 Produktet .....	47
Forbedringer / videreutvikling:.....	47
6.3 Gruppearbeidet.....	48
6.4 Oppdragsgiver.....	49
6.5 Veileder.....	50
6.6 Veileder og Oppdragsgiver.....	50
<b>7 Konklusjon</b> .....	<b>51</b>
<b>8 Litteraturliste</b> .....	<b>52</b>
<b>9 Vedlegg</b> .....	<b>53</b>

## 1 Innledning

### 1.1 Oppgavedefinisjon:

Oppgaven har gått ut på å utvikle en prototype av et system som skal kunne brukes av flere små og mellomstore bedrifter samtidig. Hver bedrift skal kunne konfigurere systemet slik at de kan bruke det som sitt eget saksbehandlingsystem mot sine kunder. Systemet skulle ligge sentralisert på en webserver, som er [www.softcross.com](http://www.softcross.com). Modellen baserte seg på ulike brukernivå (kunde, bedrift, adm) som ett B2B2C system, der bedriften leier systemet av administratoren for å bruke det mot sine kunder.

Kunden kan sende inn en sak til bedriften og deretter følge saksgangen på den. Systemet skulle være enkelt å bruke, så enkelt at de ansatte hos bedriftene ikke har behov for noe særlig opplæring. Oppdragsgiver (hovedadministrator) skulle kunne administrere systemet fra web, f. eks opprette nye brukere (bedrifter), og fjerne kunder fra systemet.

På grunn av prosjektets tidsrammer ville fokus være på funksjonaliteten, og det gradfiskebrukergrensesnittet vil derfor ikke være prioritert.

Prototypen skulle bli testet opp mot et begrenset antall brukere for så å bli modifisert etter brukernes ønsker.

### 1.2 Målgruppe

#### **Produktmålgruppe:**

Små og mellomstore bedrifter som trenger kundebehandlingsystem og som har tilgang til Internett.

#### **Rapportens målgruppe:**

Oppdragsgiver, Sensor, Veileder og andre som måtte være interessert i å se hva hovedprosjektet Softcross går ut på.

### 1.3 Egen kompetanse

Etter 3 år på HIG har gruppa fått kunnskap innen mange programmeringsspråk. Webprogrammering hadde alle 3 drevet med i div prosjekter og jobber. Vi måtte lære oss å Coldfusion som vi utviklet mesteparten av produktet i, å sette opp egen webserver med Apache, MySQL, Oracle, WinCVS, MyODBC og Coldfusion server og div Javascript.

#### 1.4 Arbeidsformer:

Gruppen har holdt til på et grupperom på HIG. Vi har flyttet maskinene våre til dette rommet og jobbet der. Vi har også fått lånt en maskin av skolen til serverbruk. Vi har hatt kontinuerlig kontakt med Oppdragsgiver og har vært flere turer i Oslo og hatt møter her på HIG.

Gruppen har holdt mandagsmøter hvor vi har evaluert forrige ukes arbeid, samt planlagt kommende uke ved å sette opp en ukeplan (se vedlegg L). Hver onsdag har vi hatt møte med veileder der vi har diskutert prosjektet. Fra alle møtene har vi skrevet møtereferat (se vedlegg G)

#### 1.5 Ord og uttrykk brukt i rapporten:

Ord/uttrykk	Beskrivelse
Owner	En bedrift som abonnerer på systemet.
Employer	Er en ansatt hos en Owner
Department	Er en avdeling hos en owner, alle ansatte har en tilknytning til en department.
Customer	En kunde av en Owner
Ticket	Er en saken som Customer melder inn til Owner.
Log	Er meldinger som hører til en Ticket.
Administrator	En som har tilgang til hele systemet, som kan aktivere og deaktivere Ownere og Customer.
B2B2C	Buisness to Business to customer. Et system som eies av en bedrift, som leies ut til en annen bedrift, som så bruker det mot sine egne kunder.

## 1.6 Organisering av rapporten

### **Kapittel 1 – Innledning**

Gir en helt kort introduksjon til prosjektet, og inneholder definisjoner av uttrykk brukt senere i rapporten.

### **Kapittel 2 – Kravspesifikasjon**

Her beskrives hva som ble forventet løst i prosjektet. Denne delen ble skrevet før vi begynte på utviklingen.

### **Kapittel 3 – Design**

Med utgangspunkt i det totale designdokumentet (se vedlegg A), fremhever vi en del essensielle valg gjort i forhold til design og vurderer disse.

### **Kapittel 4 - Implementering av kode**

Begrunner valg av programmeringsspråk, database og systemutviklingsverktøy. Beskriver også utviklingsmiljøet og filstrukturen.

### **Kapittel 5 - Testing**

Beskriver prosessen hvordan vi planlagte, utførte og evaluerte testfasen.

### **Kapittel 6 – Diskusjon av resultater**

Inneholder diskusjon av gruppen, hvordan forholdet til veileder og oppdragsgiver var, og hvordan resultatet ble.

### **Kapittel 7 – Konklusjon**

Gruppen ser prosjektet under ett og former en konklusjon på grunnlag av kap 6.

### **Kapittel 8 - Underskrifter**

Underskriftene fra deltagerne i prosjektet.

### **Kapittel 9 - Litteraturliste**

Inneholder litteraturliste for prosjektet

### **Vedlegg**

Oversikt over alle vedleggene til rapporten

## 2 Kravspesifikasjon

Kravspesifikasjonen ble utarbeidet tidlig i prosessen og den har ikke blitt endret på siden da. Enkelte krav har blitt endret underveis, noen moduler har blitt lagt til, noen blitt fjernet. Mer om dette i diskusjon av resultater i kap. 6

I kravspesifikasjonen beskriver vi de generelle krav vi og den målgruppen vi har stiller til kundebehandlingssystemet. Vi beskriver dessuten funksjonelle og operasjonelle kravene vi stiller til systemet, og gir en vurdering av hvordan vi ser livssyklusen til et slikt konsept.

### 2.1 Orginal kravspesifikasjon

#### Dokumentets struktur

Del A: Er beregnet på beslutningstakerne i prosjektet. Denne delen er svært overordnet og forklarer i korte trekk retningslinjene for prosjektet.

Del B Er beregnet på brukeren av systemet. I denne delen spesifiserer vi systemet som kunden skal motta. Dette innebærer krav til materiell, kunnskap og lignende. Vi forsøker også å beskrive brukermiljøet og momenter omkring systemets livsløp.

Del C Er beregnet på oss og leverandør av det endelig systemet. Her spesifiseres de krav som stilles til systemet.

#### 2.1.A Introduksjon

##### 2.1.A.1 Bakgrunn

Etter å ha jobbet i flere år med utvikling og tilpasning av et kundesupportsystem for EITeleØst AS og skrevet en hovedoppgave om kundeorienterte websystemer, har oppdragsgiver sett behov for:

- Billigere og mindre avanserte systemer for kundesupport i små og mellomstore bedrifter
- Et Web-basert system for å fange opp, behandle og melde tilbake på henvendelser fra kunder.
- Effektivisering av kundepleie i egen bedrift. La kunden selv følge opp egne forespørsler for å avlaste eget supportapparat.
- Å kunne dele kostnadene på tvers av kundemassen ved å Sentralisere driften over Internett.

Oppdragsgiver mener det er behov for et slik system i markedet, men pga tidsmangel har han ikke hatt mulighet for å realisere prosjektet selv. Derfor har han tatt kontakt med HIG for å videreutvikle og teste ut ideen.

### 2.1.A.2 Kort om krav til systemet

Oppdragsgiver ønsker et kundebehandlingssystem som skal kunne brukes av flere bedrifter mot sine kunder. Systemet skal ligge på en sentralisert webserver. Alle bedriftene skal selv kunne sette opp sitt eget preg på siden (sette inn logo, velge font etc). Særpreget gjelder ikke bare grafikk men også enkelte funksjonelle ting som menyvalg og egne felter i meldingen.

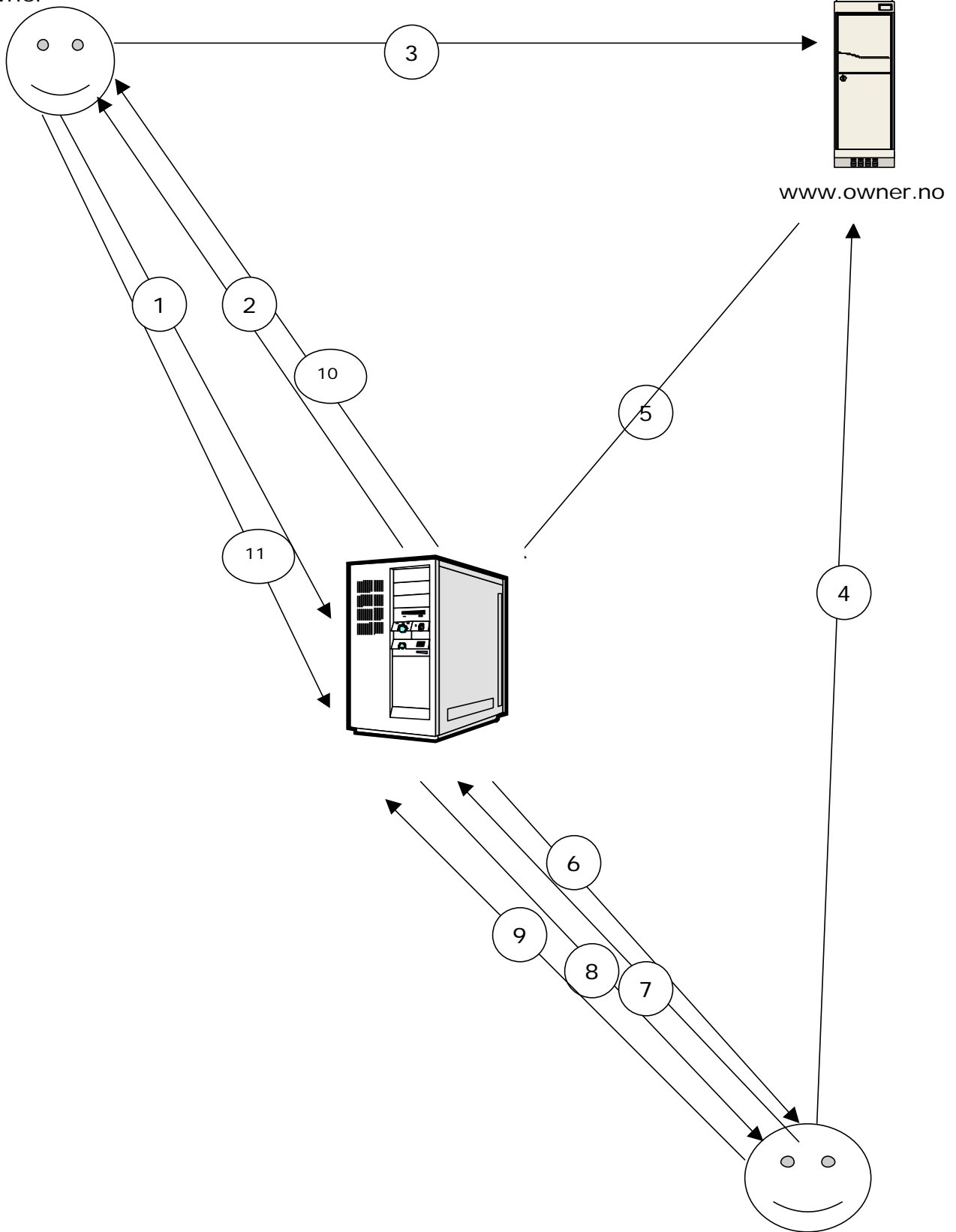
Ønsker til systemet:

- En bedrift må opprette en bruker for å få benytte systemet.
- Hver bedrift skal ha et brukernavn og passord
- En kunde må opprette en bruker hos den bedriften den ønsker kontakt med
- Hver kunde skal ha et brukernavn og passord hos den bedriften de ønsker kontakt med
- Når en kunde melder inn en "ticket", skal bedriften få beskjed om dette
- Bedriften skal kunne sette opp "prioritet" på en "ticket", dette skal ikke kunden kunne se.
- Når en bedrift har lest en "ticket" skal status settes til "lest".
- Både bedriften og kunden skal kunne sende meldinger tilhørende en "ticket" til hverandre.
- Bedriften skal kunne få ut rapporter etter behov.
- Hver måned skal det kunne skrives ut en rapport med "fakturagrunnlag" hvor det står hvor mye hver bedrift skal faktureres for.
- Det skal være en "Administrator" som skal kunne slette og endre systemet
- Systemet skal være sikkert
- Systemet skal være brukervennlig
- Systemet skal utvikles i Cold Fusion
- Systemet skal være koblet opp mot en database(MYSQL).

Avgrensninger:

- Det skal ikke være mulig for en kunde og ha kontakt med flere bedrifter med en og samme bruker.

Owner



`www.owner.no`

Customer

**Figur 2-1 Moduler i systemet**



Kommentar til Figur 2.1 :

1. Owner registrere seg og setter opp sine valg for siden på [www.softcross.com](http://www.softcross.com)
2. Den ferdige siden blir returnert til owner. Owner får en url til å legge inn som link fra [www.owner.no](http://www.owner.no) til [www.softcross.no/innlogging.cfm?owner=owner](http://www.softcross.no/innlogging.cfm?owner=owner)
3. Owner legger inn linken på deres hjemmeside [www.owner.no](http://www.owner.no), og evt. Sender invitasjoner til deres kunder
4. Customer går inn på [www.owner.no](http://www.owner.no). og
5. Customer trykker på linken [www.softcross.no/innlogging.cfm?owner=owner](http://www.softcross.no/innlogging.cfm?owner=owner) , og blir videresendt til owner sin side på [www.softcross.com](http://www.softcross.com)
6. Customer blir bedt om å skrive brukernavn og passord, hvis ikke customer er registrert fra før blir den bedt om å registrere seg.
7. Customer registrerer seg
8. Customer blir logget inn, og er klar til å bruke systemet
9. Customer sender inn en ny ticket/skriver i loggen på en ticket
10. Owner får den nye ticketen/beskjed i om det er lagt til en loggmelding på en ticket
11. Owner kan sette prioritet, status, velge avdeling og skrive i loggen på ticketen.
12. Customer få beskjed om at det er en endring på ticketen.

Når en owner har registrert seg vil punktene fra 4-12 gjenta seg.

### 2.1.A.3 Kort om systemets omgivelser

Systemet skal ligge på en webserver([www.softcross.com](http://www.softcross.com)). Bedrifter skal kunne benytte systemet fra denne serveren, og altså ikke legge det inn på sin egen server. Bedrifter skal kunne implementere(ved hjelp av frames) kundebehandlingssystemet på sine egne sider. De skal kunne sette sitt eget preg på sidene.

### 2.1.A.4 Referanser

Utviklingsmodellen er hentet fra "Kundeorienterte Web-systemer", hovedprosjekt på UIO November 1996 av Werner Grini og Arne Sunde..

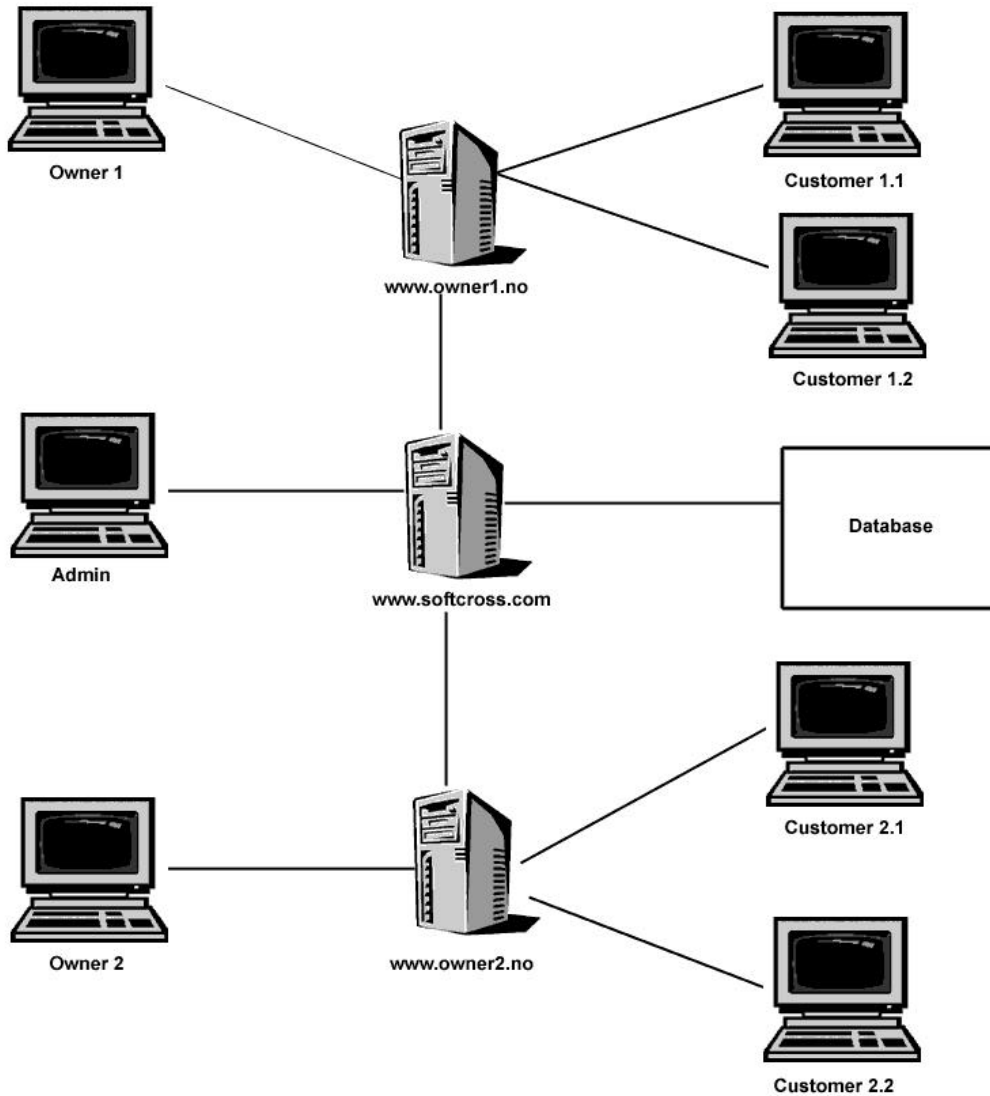
## 2.1.B Brukerbeskrivelse

### 2.1.B.1 Omgivelser

Systemet skal fungere over Internett. Det som kreves for at systemet skal fungere er:

- 1 webserver hvor systemet er sentralisert
- maskin(er) hvor Owner har tilgang til Internett
- maskin(er) hvor Customer har tilgang til Internett

Nettverksskisse:



Figur 2-2 - Nettverksskisse

### 2.1.B.2 Systemets brukere

Både Owner og Customer skal bruke systemet over Internett, og de må da kunne bruke en nettleser. For å kunne benytte systemet fullt ut bør man også ha tilgang til e-post.

Systemet skal være så enkelt at enhver ”vanlig” bruker av Internett, skal kunne benytte det uten vanskeligheter. Det skal også være en hjelp funksjon på alle sider, slik at en kan bli guidet gjennom siden etter ”wizard”-metoden.

For at systemet skal blir så enkelt, skal det lages en prototype som skal testes mot Owner og Customer. Etter denne testen skal det tas hensyn til deres ønsker om endringer.

Ved eventuelle problemer ved systemet skal det tas kontakt med Administrator, som er en kompetent person som skal arbeide med videreutvikling av systemet, som så skal hjelpe til med problemet.

### 2.1.B.3 Funksjon

Tabellen viser en oversikt over modulene og input og output til disse:

Moduler	Input	Output	Nr
Registrere ny Owner	Ownerinformasjon	Reg. employerside	1
Registrere ny Customer	Customerinformasjon	Innlogget side hos owner	7
Registrere ny Employee	Employerinformasjon	Owner sin egen side	
Invitasjon av Customer	Epostadresser	Epost m/invitasjon	3
Endre profil	Ønsket endring	Bekreftelse på endring	
Sende inn en Ticket	Ticketinnhold	Ticketinnhold	9
Tilegne en ticket avdeling	Avdeling	Ticket med avdeling	11
Sette status på en ticket	Status	Status	11
Sette en prioritet	Prioritetsnivå	Ingen for customer/ Pri.nivå for Owner	11
Sende en melding på en ticket	Melding	Ticket med log	12
Generere rapporter/søk	Rapportkriterier	Rapporten	
Generere fakturagrunnlag	Fakturaperiode	Fakturagrunnlag	
Innlogging	Brukernavn og passord	Den rette siden	8

Nr henviser til hvilken modul på figur 2.1

### 2.1.B.4 Operasjon

I normal tilstand skal systemet fungere på følgende måte:

Når en Customer/Owner bruker systemet, vil han/hun sitte å arbeide i en nettleser på Internett. For å benytte systemet må man være ”logget inn”. Når man er logget inn skal man kunne kalle på ulike moduler ved å klikke på valg i menyer.

Systemet skal være oppe og gå kontinuerlig.

Dersom webserveren, databasen eller forbindelsen til webserveren går ned, skal ikke noen brukere kunne benytte systemet. Dette er fordi all data som skal behandles/vises er avhengig av databasen. Dersom systemet går ned, skal alle data i databasen automatisk bli bevart. Data som brukeren arbeider med og som ikke er lagret i databasen når dette skjer, vil gå tapt. Systemet skal lagre alle data i databasen etter hvert som de legges inn og når noe endres.

Systemet skal generere feilmeldinger for de mest sentrale feil som brukeren gjør. Dersom brukeren prøver å gjøre noe som vil være uheldig for systemet, vil brukeren få en advarsel om dette og bli nektet å utføre operasjonen. Dersom brukeren skal utføre en operasjon av større betydning, som for eksempel å slette, vil brukeren få opp en melding om han/hun virkelig vil utføre dette.

### 2.1.B.5 Aspekter omkring livvsyklus

Hovedmålet for prosjektet er å utvikle en kjørbart prototype av systemet. Derfor er det viktig å kommentere i koden, slik at det blir lett for oppdragsgiver å endre systemet ved oppdateringer, endringer og feil.

Under utviklingen av prosjektet bruker vi et versjonskontrollsystem, winCVS. Dette systemet tar vare på alle gamle versjoner av koden, slik at man lett kan hente frem gammel kode. Dette systemet anbefales å bruke videre, det vil gjøre det enkelt og sikkert å oppdatere/redigere systemet. Dersom det viser seg at systemet blir veldig populært, bør systemet flyttes til en raskere server med en raskere tilkobling. Databasen bør også byttes ut slik at flere brukere kan jobbe mot databasen samtidig. Kan også vurdere å gjøre om systemet til en firelagsstruktur, hvor man bruker f.eks EJB`S som kobling mot databasen. Dette vil minke antall spørringer mot databasen. For å flytte dataen fra en database til en annen må man konvertere dataen til det rette formatet.

### 2.1.B.6 Ytelse

Det som begrenser ytelsen på systemet er:

- Databasen må kunne takle flere brukere på en gang
- Webserveren må kunne takle flere brukere på en gang

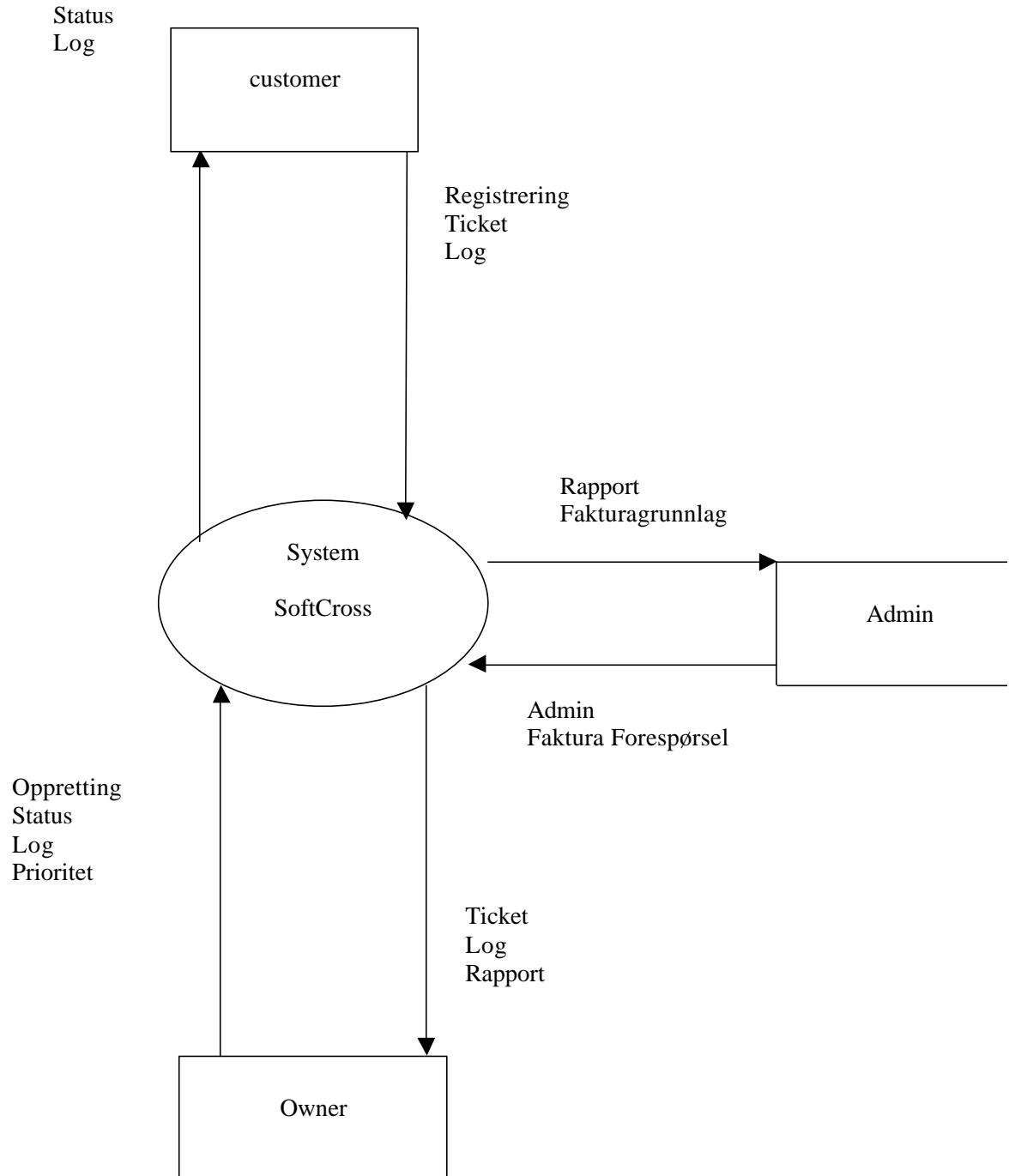
### 2.1.B.7 Antagelser

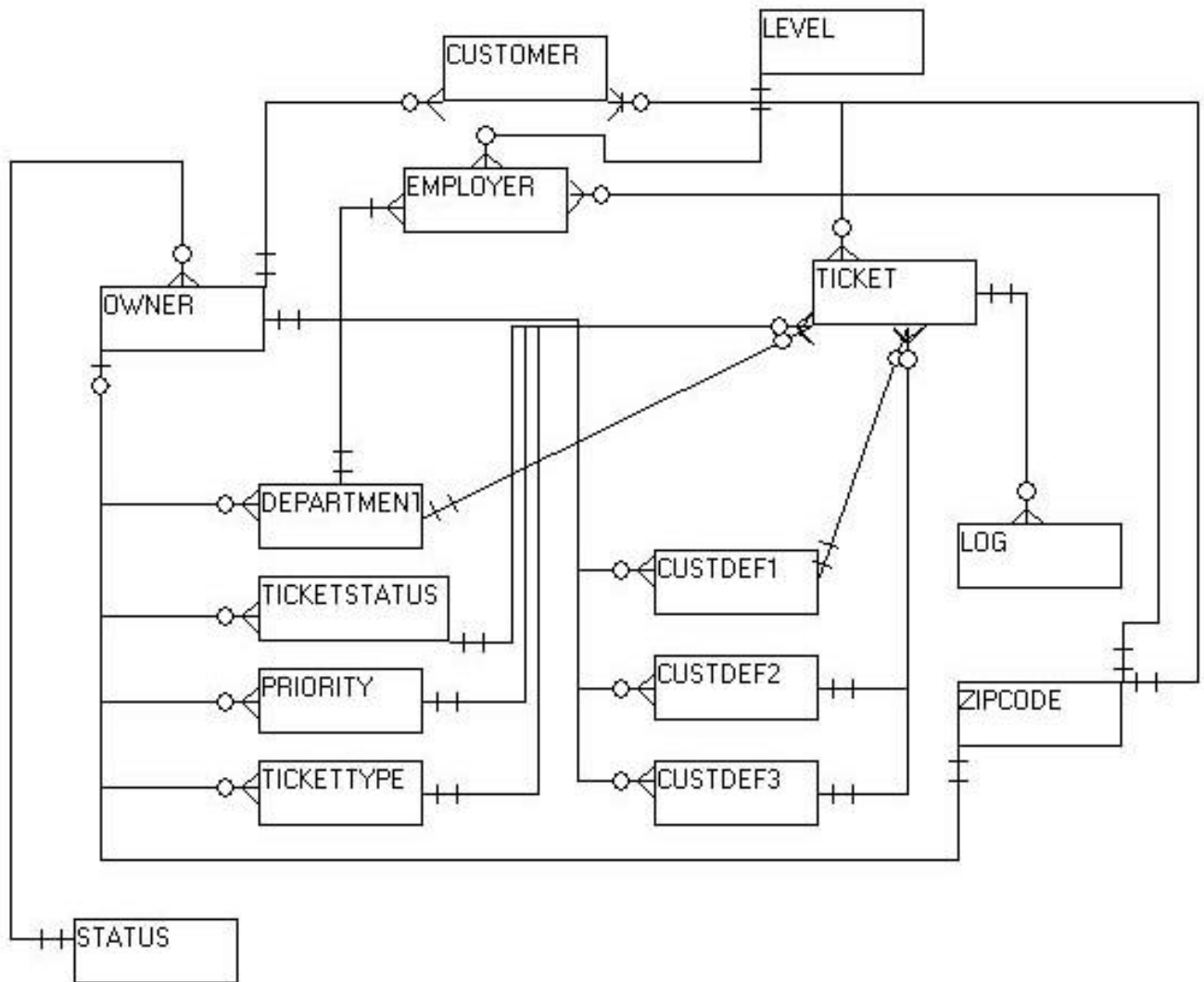
Vi antar at alle nye Owners skal kunne registrere seg selv, og at dette ikke er en jobb som skal utføres av Administrator.

## 2.1.C Detaljert kravspesifikasjon

### 2.1.C.1 Funksjonell struktur

Figur 2-3 Kontekstdiagram





Figur 2-4 Datamodell

## 2.1.C.2 Data spesifikasjon og dataordliste

### 2.1.C.2.1.1 Modus og kontroll

Systemet har tre nivåer(customer, owner, admin). Systemet knytter seg direkte mot databasen for input og output.

### 2.1.C.2.1.2 Ytelse

Det som begrenser ytelsen på systemet er:

- Databasen må kunne takle flere brukere på en gang
- Webserveren må kunne takle flere brukere på en gang
- Stilig design på sidene, men samtidig lett å bruke.

### 2.1.C.2.1.3 Sikkerhet

Alle passord skal bli vist kryptert på skjerm. For at man skal bruke systemet så må man logge seg på med brukernavn og passord. Siden påloggingssiden vil ligge på Internett vil det til en viss grad være fare for innbrudd. Bruker auto increment for å sikre at ingen brukere av systemet vil få samme ID nr ved registrering av ny bruker (tildele entydig ID). Session variablene skal vare i 30 min. Sensitive data skal ikke sendes i URL'en. Bør opprettes backup rutiner for databasen når systemet blir lagt ut for allmennheten.

### 2.1.C.2.1.4 Oppstart og nedtagning

Systemet skal settes opp på en webserver av typen apache. Coldfusion serveren skal installeres og kobles opp mot en database, MySQL via en MYODBC kobling. Når systemet blir flyttet til EITeleØst vil det bli brukt en MySQL database. Systemet kan bli flyttet hvis det viser seg at systemet blir veldig populært.

### 2.1.C.2.1.5 Tilgjengelighet

Når systemet blir satt opp på EITeleØst skal det være oppe 247 (hele tiden). Det er veldig liten mulighet for at systemet skal være nede når det ligger på serveren til EITeleØst. De har veldig bra sikkerhetsrutiner. Tilgjengeligheten for kundene varierer med kvaliteten på oppkoblingsmulighetene.

### 2.1.C.2.1.6 Innebygde tester

Alle operasjoner skal testes slik at brukeren får beskjed om feil input, manglende input. Det står beskrevet hvilke felt som er nødvendige (merket med symbol). Denne sjekken skal hovedsakelig ikke gjøres på serveren. Bør gjøres ved hjelp av javascript og på selve HTML siden (required).

Det sjekkes om brukernavn finnes fra før, dette er en sjekk som må gjøres på serveren.

## 2.1.C.2.2 Operasjon i feilsituasjoner

### Feilrapportering

Feil som oppstår skal rapporteres til brukeren via skjerm. Feilmelding som systemet lager skal ikke vises - bare feilmeldinger som er laget av oss.

### Gjenervelse etter feil

Etter (system) feil må brukeren taste inn opplysninger på nytt. Disse blir ikke lagret ved systemfeil. Ved feilmeldinger der brukeren har tastet inn feil, manglende opplysninger vil data fortsatt bli værende i skjemaet, men man må trykke lagre en gang til + manglende opplysninger for at data skal bli lagret.

### 2.1.C.3 Moduler

#### FUNKSJONELLE KRAV TIL MODUL 1. – REGISTRERE NY OWNER

##### Input

Data som fylles ut på registreringssiden. Data om owneren som vil begynne å bruke systemet:

- Navn
- Adresse
- Postnr
- E-post
- Telefon nr
- Fax
- Land
- Passord
- Bekreft Passord
- Farge
- Skrift
- Bilde
- Customize felter(3 stk)
- Sette opp hvilken status man kan velge i systemet
- Sette opp hvilke avdelinger som finnes i bedriften
- Sette opp employee(username, password, department)
- Sette opp hvilke prioritets valg man har i bedriften
- Sette opp ticket type

##### Prosessering

Legger inn inntastede data inn i databasen i owner, employee, department, tickettype, ticketstatus, priority, customize.

##### Output

Man får beskjed om at data er lagret. Får beskjed om at enkelte opplysninger må fylles ut før man kan starte å bruke systemet. Kommer inn på hovedmenyen på systemet.

#### FUNKSJONELLE KRAV TIL MODUL 2. – REGISTRERE NY CUSTOMER

##### Input

Input her er data som man fyller inn i formene på registreringssiden. Noe er valgfritt. Generell data om cusomer:

- Navn
- Adresse
- Postnr
- Tlf
- Passord
- Bekreft passord
- Fax
- Land
- E-post





### **Prosessering**

Modulen legger inntastede data inn i databasen. Oppretter en ny kunde tilknyttet bedriften.

### **Output**

Man får beskjed om at data er lagret. Kommer inn på hovedmenyen på systemet.

## **FUNKSJONELLE KRAV TIL MODUL 3. – ENDRE DATA I PROFIL FOR OWNER - DATABASEN**

### **Input**

Input her er endring på data i profilen til owner.

Endre feltene:

- Navn
- Adresse
- Postnr
- Tlf
- Passord
- Bekreft passord
- Fax
- Land
- E-post

### **Prosessering**

Modulen legger inntastede data inn i databasen. All data blir lagt inn slik at gammel data blir skrevet over, selv om det er gjenlagring av gammel data.

### **Output**

Man får beskjed om at data er lagret. Får opp hovedmenyen etterpå.

## **FUNKSJONELLE KRAV TIL MODUL 4. – ENDRE DATA I PROFIL FOR CUSTOMER - DATABASEN**

### **Input**

Input her er endring på data i profilen til customer.

Endre feltene:

- Navn
- Adresse
- Postnr
- Tlf
- Passord
- Bekreft passord
- Fax
- Land
- E-post

### **Prosessering**

Modulen legger inntastede data inn i databasen. All data blir lagt inn slik at gammel data blir skrevet over, selv om det er gjenlagring av gammel data.

### **Output**

Man får beskjed om at data er lagret. Får opp hovedmenyen etterpå.

## FUNKSJONELLE KRAV TIL MODUL 5. – SENDE INN EN TICKET

### Input

Input data blir hva man oppgir i ticketen. Varierer litt fra om man er Customer eller Owner.

- Overskrift
- Saksbeskrivelse
- Tickettype

### Prosessering

Modulen legger inntastede data inn i databasen

### Output

Man får beskjed om at ticketet er sendt.

## FUNKSJONELLE KRAV TIL MODUL 6. – SETTE STATUS PÅ EN TICKET

### Input

Input er hvilken status man setter på ticketen.

F. eks:

- Vente på deler
- Behandles
- Sendes til annen avdeling
- Utført

### Prosessering

Modulen legger inntastede data inn i databasen. Omgjør fargen på teksten, slik at man kan se at statusen er forandret. Begge brukere(owner, customer) får se endret status.

### Output

Man får beskjed om at Status er satt. Viser med at teksten forandrer farge.

## FUNKSJONELLE KRAV TIL MODUL 7. – SETTE PRIORITET PÅ EN TICKET

### Input

Input er hvilken prioritet man setter på ticketen.

F. eks:

- Lav
- Høy
- Hastesak
- Nedprioritert

### Prosessering

Modulen legger inntastede data inn i databasen.

### Output

Man får beskjed om at Prioritet er satt.



## **FUNKSJONELLE KRAV TIL MODUL 8. – SENDE INN EN MELDING PÅ EN TICKET**

### **Input**

Input er meldingen man setter inn i loggen på ticketen. Skrives inn i ett tekstfelt nederst i ticketen.

### **Prosessering**

Modulen legger inntastede data i ticketen inn i databasen.

### **Output**

Man får beskjed om at man har skrevet en melding inn i ticketen. Blir lagt som output i loggen på ticketen.

## **FUNKSJONELLE KRAV TIL MODUL 9. – GENERERE RAPPORTER/SØK**

### **Input**

Input er søkekriteriet. Angir hvilken rapport det søkes etter.

### **Prosessering**

Gjør en Query i databasen med de angitte parametrene, og den henter ut de postene etter søkekriteriet

### **Output**

Output er enten forhåndsdefinerte rapporter, eller etter hva man angir som søkekriterier.

F. eks:

- Saker behandlet forrige mnd.
- Faktura grunnlag pr bedrift forrige mnd.
- Hvilke Customer man har vært i kontakt med siste mnd.
- Saker i en avdeling

## **FUNKSJONELLE KRAV TIL MODUL 10 – GENERERE FAKTURA GRUNNLAG**

### **Input**

Taster inn faktura kriterier man søker etter.

### **Prosessering**

Gjør en query i databasen med de angitte parametrene, henter ut ønsket data. Registrerer i ett felt i databasen at faktura er skrevet ut.

### **Output**

Output er enten forhåndsdefinerte rapporter, eller etter hva man angir som søkekriterier.

F. eks

- Faktura på sakene bedriften har behandlet den siste mnd
- Års avgift for bruk på systemet.



## **FUNKSJONELLE KRAV TIL MODUL 11. – INNLOGGING I SYSTEMET**

### **Input**

Taster inn brukernavn og passord, trykker på submit eller på slett.

### **Prosessering**

Sørger for at brukeren kommer inn på systemet og at riktig side vises. Startsidene kommer an på hvilken department man tilhører.

### **Output**

Er data korrekt får man beskjed om man er innlogget på systemet, man kommer inn på riktig department hvis man er owner. Hvis data er ukorrekt vil man få beskjed om å prøve igjen.

## **FUNKSJONELLE KRAV TIL MODUL 12. – INVITASJON AV CUSTOMERS VIA EPOST.**

### **Input**

E-post adressen til customers som skal ha invitasjon.

### **Prosessering**

Sørger for at e-post blir sendt til customers. Sender ett standard brev.

### **Output**

Får opp en liste over hvilke customers man har sendt e-post til.

## **FUNKSJONELLE KRAV TIL MODUL 12. – REGISTRERE EMPLOYER**

### **Input**

Skriver inn :

- brukernavn
- passord
- telefonnr
- mobil
- adresse
- navn
- department
- email
- postnr
- brukernivå

### **Prosessering**

Legger inn employee inn i databasen slik at man kan logge seg på med eget passord og brukernavn med riktige rettigheter..

### **Output**

Output er en bekreftelse på at man har registrert aktuell employee

#### 2.1.C.4 Begrensninger

Kunden skal kunne sette opp sitt eget preg på sidene. Det skal være mulig for kunden å implementere løsningen på sine egne websider. Kunden skal få opp rapporter i nettleseren over alle "ticketene" kunden har hatt.

Programmerings språkene blir Html, JavaScript, Sql og Coldfusion.

Systemet skal støtte en helt standard moderne nettleser. Nettleseren må støtte Java-script. Systemet stiller ingen krav til software pakker/verktøy hos owner/customer.

Brukeren må ha Internett tilgang og operativsystemet hos owner/customer må støtte bruk av nettlesere og Internett.

Det er minimale hardware krav. Så lenge maskinen fungerer til normal bruk av Internett. Bør ha rask Internett tilgang.

#### 2.1.C.5 Dokumentasjon

Det vil bli utarbeidet en dokumentasjon av arbeidet.

Vår dokumentasjon vil grovt sett inneholde:

- Innholdsfortegnelse
- Kravspesifikasjon
- Design
- Koding
- Testing
- Navn på personer som har utviklet og som har blitt brukt til support
- Evaluering

##### 2.1.C.5.1 Modul og integrasjonstesting

Ved testingen skal vi sende ut spørreskjema til de aktuelle testerne. For at vi skal gjennomføre en vellykket testfase må vi ha tilstrekkelige testere på customer og owner siden. På owner siden bør det være folk som har vært borte i slike systemer før.

##### 2.1.C.5.2 Krav til support, service og vedlikehold

Administrator skal kunne vedlikeholde systemet for kunden på web. Det er derfor lett for administrator å utføre support.

##### 2.1.C.5.3 Krav til utvidelser

Systemet vi skal utvikles er en prototype. Det skal være mulig å endre systemet til neste prototype, helt tilbake på kravspekk nivå.

Hver bedrift som benytter systemet skal ha mulighet til å endre på noe av designet på websidene. Det skal også være mulig å legge til egendefinerte attributter i databasen.

## 2.2 Analyse av arbeidet ved å komme frem til kravspesifikasjon.

For å kunne utforme kravspesifikasjonen måtte vi prøve å finne mest mulig ut om hva som kunde behandlingssystemer skulle inneholde. Vi hadde ett møte med Oppdragsgiver hvor vi gjennomgikk hvilke moduler som var vanlige ved slike systemer, og hvilke moduler Oppdragsgiver mente var kritiske for ett slikt system. Vi satt av en uke til research hvor vi skulle se på lignende systemer, prøve å finne ut via artikler og undersøkelser hva som var viktig ved slike systemer. Oppdragsgiver sendte oss noen linker til lignende systemer. Her fikk vi endel gode ideer for noen moduler og en litt bedre forståelse av oppgaven. Vi fikk også sett på en kundesystem som oppdragsgiver hadde lagd. Gruppen brukte også news groups for å få flere impulser.

Problemene ved å lage ett slikt system er at vi ikke viste hvilke moduler som er viktige og hvilke som kan utelukkes. Vi hadde vanskeligheter for å finne ut hvilke vi skulle ha med, men ved samarbeid med oppdragsgiver kom vi fram til en løsning.

For å utforme kravspesifikasjonen brukete vi en kavspesifikasjonsmal. Den er basert på «The STARTS Purchasers' Handbook» kap.4 og Appendix B, oversatt til norsk av Frode Haug, HiG.

Vi valgte å bruke SA modelering, dette valgte vi fordi SA modelering er et fint verktøy i kombinasjon med Datamodelering. Vi kunne valgt et objektorientert modeleringsverktøy som "Rational rose". Grunnen til at vi ikke valgte dette var at vi hadde nok med å lære et nytt programmeringspråk og ikke tid til å lære en ny modeleringsverktøy.

### 3 Design

I denne delen av rapporten viser vi de essensielle valgene vi har tatt med hensyn på designet av systemet.

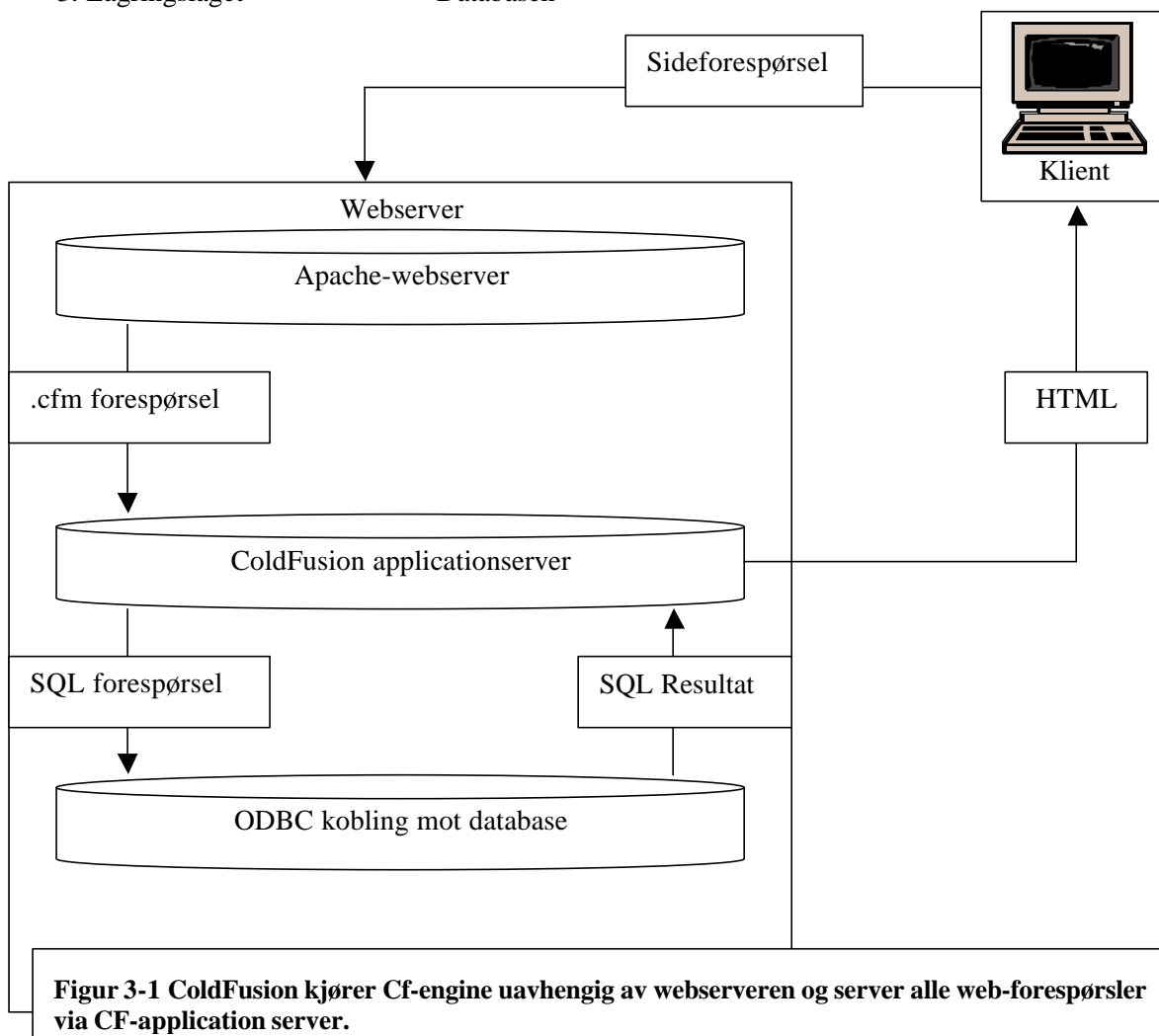
#### 3.1 Systemarkitektur

Oppgaven vår har gått ut på å utvikle en internetbasert løsning. Vi var derfor avhengig av å lage en løsning som blir gjort tilgjengelig ved hjelp av en webserver. Da det er veldig mye data som skal lagres i systemet, var vi avhengig å bruke en database. For at sidene skulle bli dynamiske, og snakke med databasen, var vi avhengig å bruke et annet programmeringsspråk enn html. Vi valgte å bruke ColdFusion som programmeringsspråk (Mer om valg av programmeringsspråk i kapittel 4).

Vi fant så ut at når vi skal bruke ColdFusion kan vi bruke to forskjellige systemarkitekturer. Valgmulighetene stod mellom å bruke EnterpriseJavaBeans eller lignende som et ekstra lag mellom hovedlaget og lagringslaget, eller koble hovedlaget direkte mot lagringslaget. Ved å bruke EJB's ville vi redusert databaseaktiviteten ved at man får alt innholdet i databasene tilgjengelig i objekter, i stedet for at man må kjøre databaseforespørslers. Dette ville da blitt en firelags arkitektur. Siden vi syntes tiden ble knapp nok til å lære oss et helt nytt språk, og at valget mellom 3 og 4 lagsarkitektur ikke ville ha noen virkning på om prototypen skulle fungere, valgte vi å bruke den enkleste arkitekturen.

**3 lagsarkitektur:**

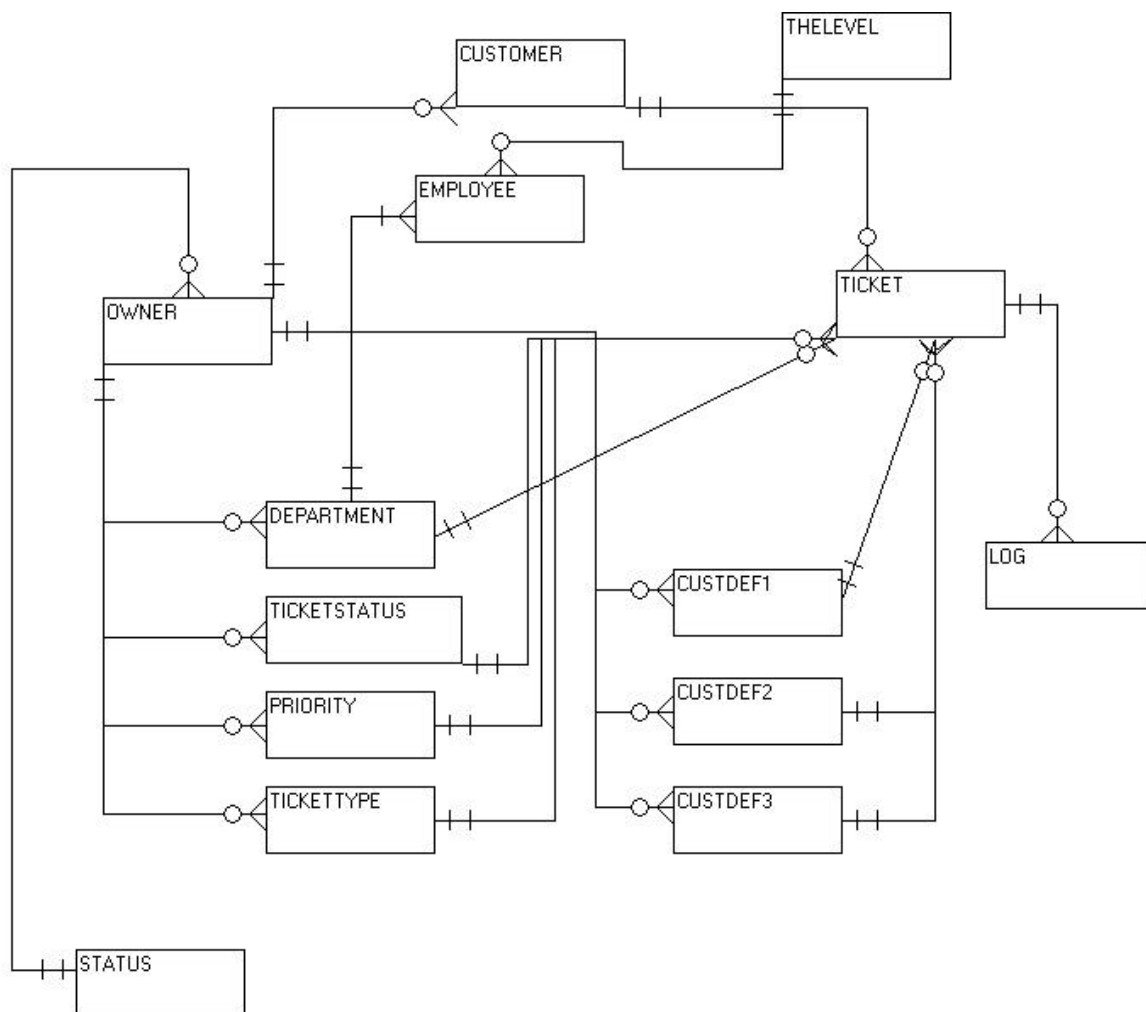
1. Presentasjonslaget - Grensesnitt som blir vist i browseren
2. Hovedlaget - Selve applikasjonen på serveren
3. Lagringslaget - Databasen





### 3.2 Databasestruktur

For at vi skulle klare å lage en applikasjon som skulle kunne brukes av flere bedrifter tilpasset deres behov, var vi avhengig av å ha en database med tilpassningsmuligheter. Vi valgte derfor å lage egne tabeller for de forskjellige valgmulighetene hver bedrift skulle kunne sette opp. Disse valgmulighetene som ble kartlagt i kravspesifikasjonen er avdelinger, statuser, prioriteter, meldingstyper samt muligheten for å sette opp 3 egendefinerte felter. Poenget er at hver bedrift skulle ha sin egen "owner\_id", denne attributten skulle så ligge i alle rader i tabellene med valgmulighetene. Så ville bedriften og deres kunder kun få tilgang til radene med de rette "owner\_id"ene. Denne "owner\_id"en brukes også til å holde rede på hvilken bedrift hver kunde tilhører. Alle kunder har så sin egen "customer\_id". Attributten "customer\_id" brukes så i alle meldinger, slik at vi alltid vet hvilken kunde og bedrift meldingene tilhører. Se Figur 3-2 og vedlegg ??? for beskrivelse av datamodellen.



Figur 3.2 Datamodell, se vedlegg A for detaljert beskrivelse av datamodellen.

### 3.3 Filstruktur

For at det ikke skulle bli så mye rot på serveren og for at filarkitekturen skulle bli mer oversiktlig, valgte vi å lage noen underkataloger på serveren.

De katalogene som er valgt er:

- / - Dette er rotkatalogen, her ligger filer som har med registrering av nye kunder å gjøre.
- /customer - Her ligger alle filene som blir brukt når en er logget inn som customer.
- /images - Her ligger alle bilder som er brukt til designet av sidene.
- /login - Her ligger loginside.
- /owner - Her ligger alle filene blir brukt når en er logget inn på ownersiden.
- /upload - Her blir alle filer som er opplastet av brukerne lagt.

I alle katalogene med coldfusion filer ligger det en Application.cfm fil. Dette er filer som ColdFusion serveren kjører før noen av de andre filene blir kjørt. Hvis alle disse filene hadde vært like, kunne vi byttet dem ut med kun en fil som vi hadde lagt i roten. Hvis ColdFusion Serveren ikke finner en Application.cfm i katalogen hvor den forespurte filen ligger, bruker den filen som ligger i katalogen over. Finner den ikke filen her heller så letes det etter den i katalogen over osv.

Grunnen til at vi har forskjellige typer av denne filen er at vi i Application.cfm filene i /customer og /owner sjekker at brukeren virkelig er logget inn. Hvis ikke brukeren er logget inn, blir dem sendt til /login/login.cfm og må logge inn. På denne måten hindrer vi at man kan gå inn på sidene uten og logge inn, noe som ville vært veldig dårlig med hensyn på sikkerhet.



Vi har valgt og dele opp skjermbilde i 3 deler som vist Figur 3-4 Vi kunne valgt å laget systemet med frames, og fortsatt fått den samme inndelingen. Grunnen til at vi ikke valgte frames var at vi ville unngå å få flere scrollbarer på siden og at vi heller ville bruke ColdFusions muligheter til å inkludere sider.

De 3 delene er:

### Del 1: Info del

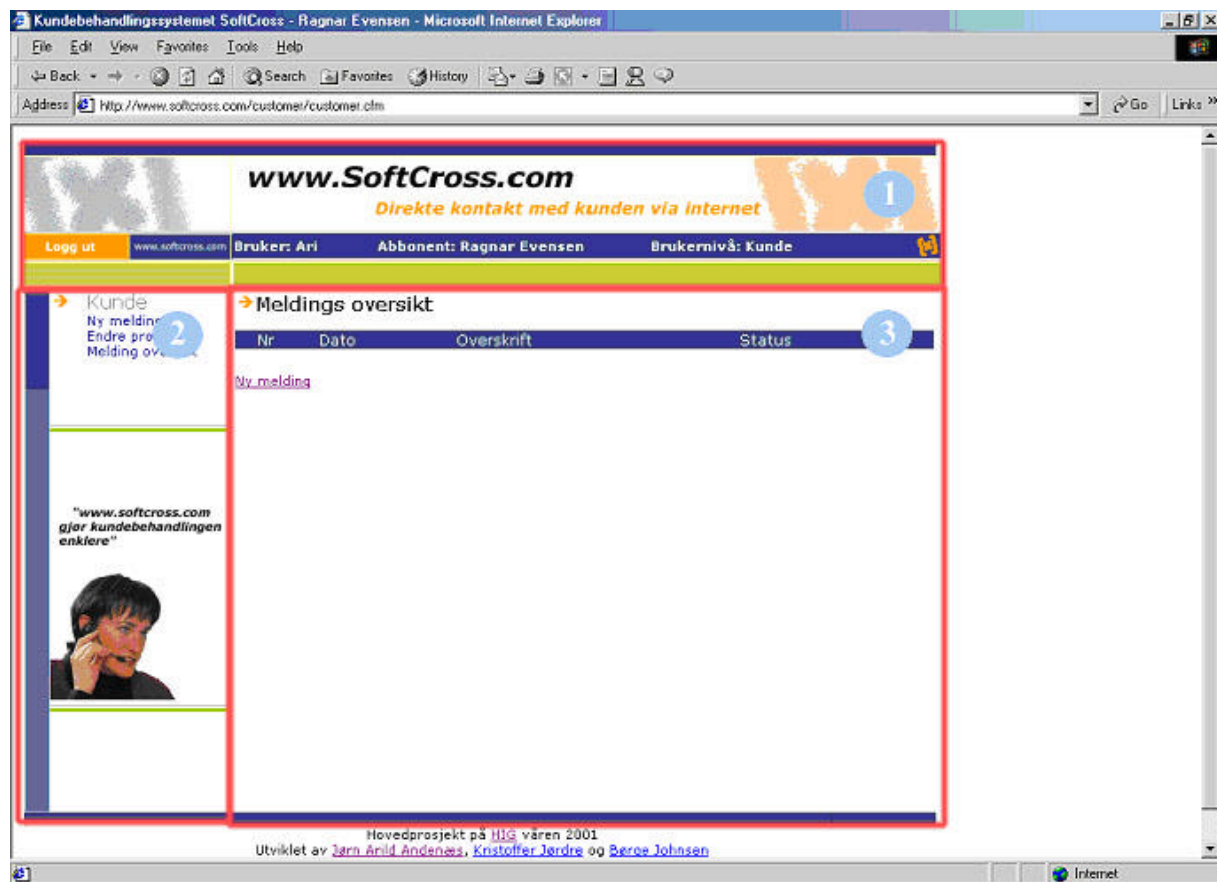
I denne delen av skjermbildet vil det stå info om hvem som er logget inn, hvilken abonnent owner man er logget inn under, og hvilket brukernivå man er logget inn på. Hvis abonnenten har satt opp at logoen deres skal vises i systemet så vil den vises her. "Logg ut" knappen er også lagt i denne delen.

### Del 2: Meny

Denne delen består av menyen til systemet. Menyene vil variere etter hvilket brukernivå man er logget inn som. Hvilket nivå man er logget inn som vil alltid stå som overskrift i menyen. Når man er logget inn vil man på alle sidene få se denne menyen. Når det klikkes på et valg i menyen blir man sendt til en ny side. Det ser for brukeren som om det kun er del 3 som lastes på nytt.

### Del 3: Hoveddelen

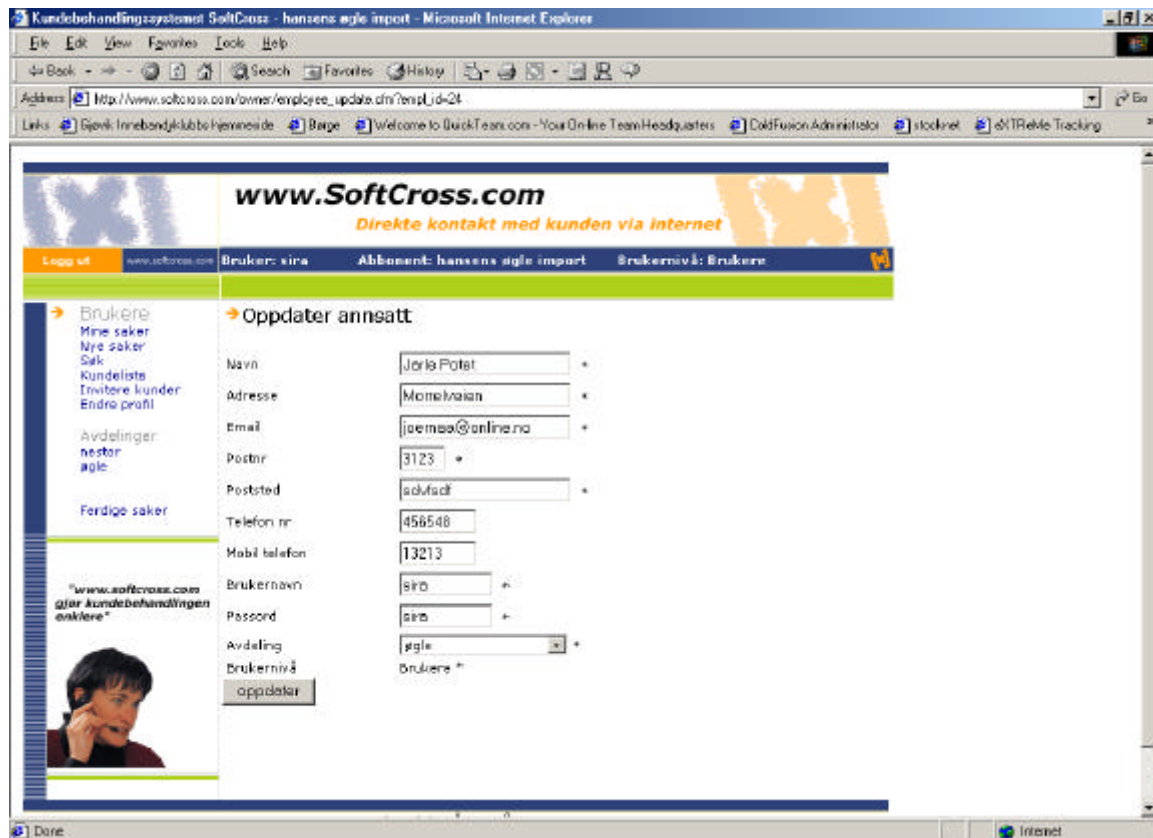
Dette er hoveddelen av skjermbildet og det er denne delen som endrer seg når man klikker seg inn på en ny side. For at brukergrensesnittet skal bli best mulig, vil vi at denne delen skal ha en fast bredde. Det skal alltid i denne delen stå en overskrift som forteller brukeren hvilken side som vises.



Figur 3-4 Skjermbildet er delt opp i 3 deler

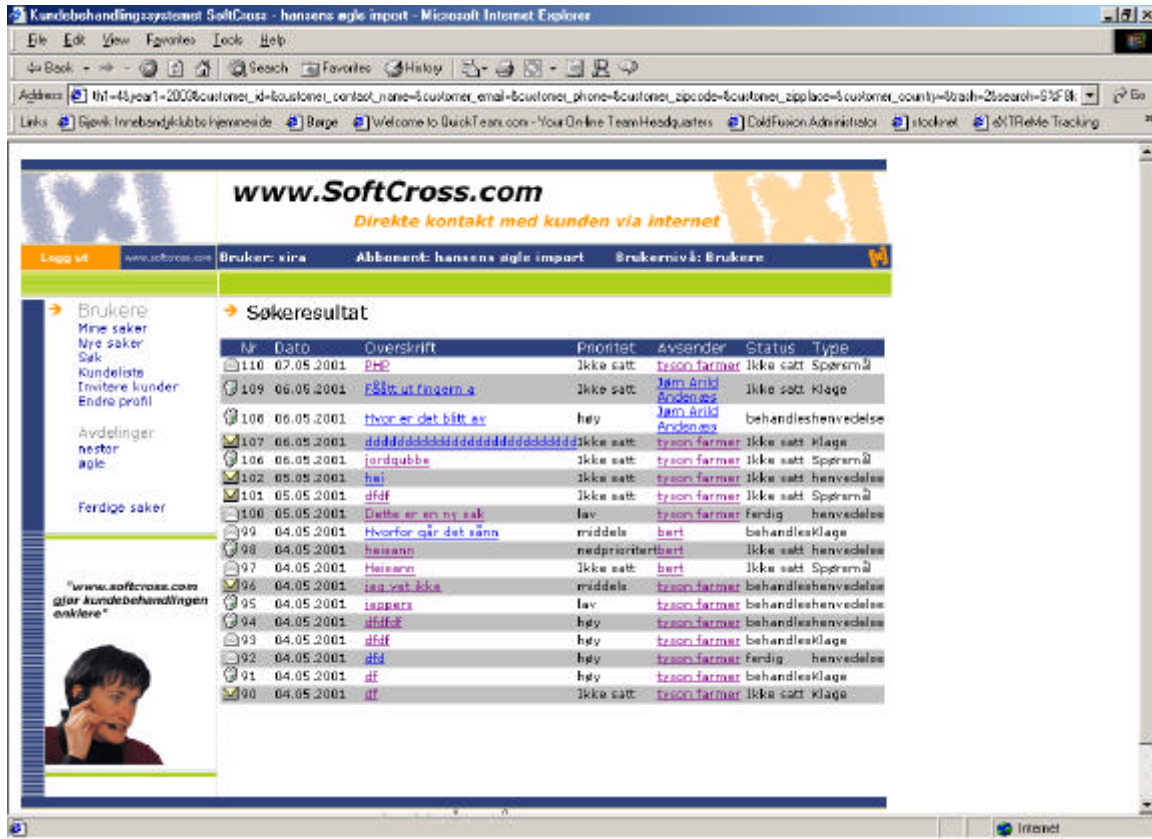
Navigeringen mellom sidene foregår hovedsakelig at man klikker seg inn på sidene ved via menyen. På noen sider vil man også få opp et valg på selve siden som kan sende deg til en ny side. Dette kan f. eks være når man klikker seg inn på en melding eller lignende. Man er selvfølgelig nødt til å bruke mus.

Vi har også brukt en del forms på sidene, for å navigere seg mellom felter i formsene kan man bruke internettstandarder som "Tab" og "Shift"+"Tab". En side med forms vi typisk se ut som Figur 3-5.



Figur 3-5 Skjerm bilde av side med forms

De sidene som er uten forms er sider med tabeller med mange rader og kolonner. For å lage et klart skille mellom hver rad har vi valgt å ha forskjellig farge på annen hver rad i tabellene. Se Figur 3-6.



Figur 3-6 Skjerm bilde av en side med en tabell med mange rader og kolonner.

I tabellene med oversikt over meldinger har vi brukt 3 forskjellige ikon for å vise status på meldingene. Disse ikonene er kjente for de fleste brukere, da de er de samme som i Microsoft Outlook Express.

### 3.5 Programmering av designet

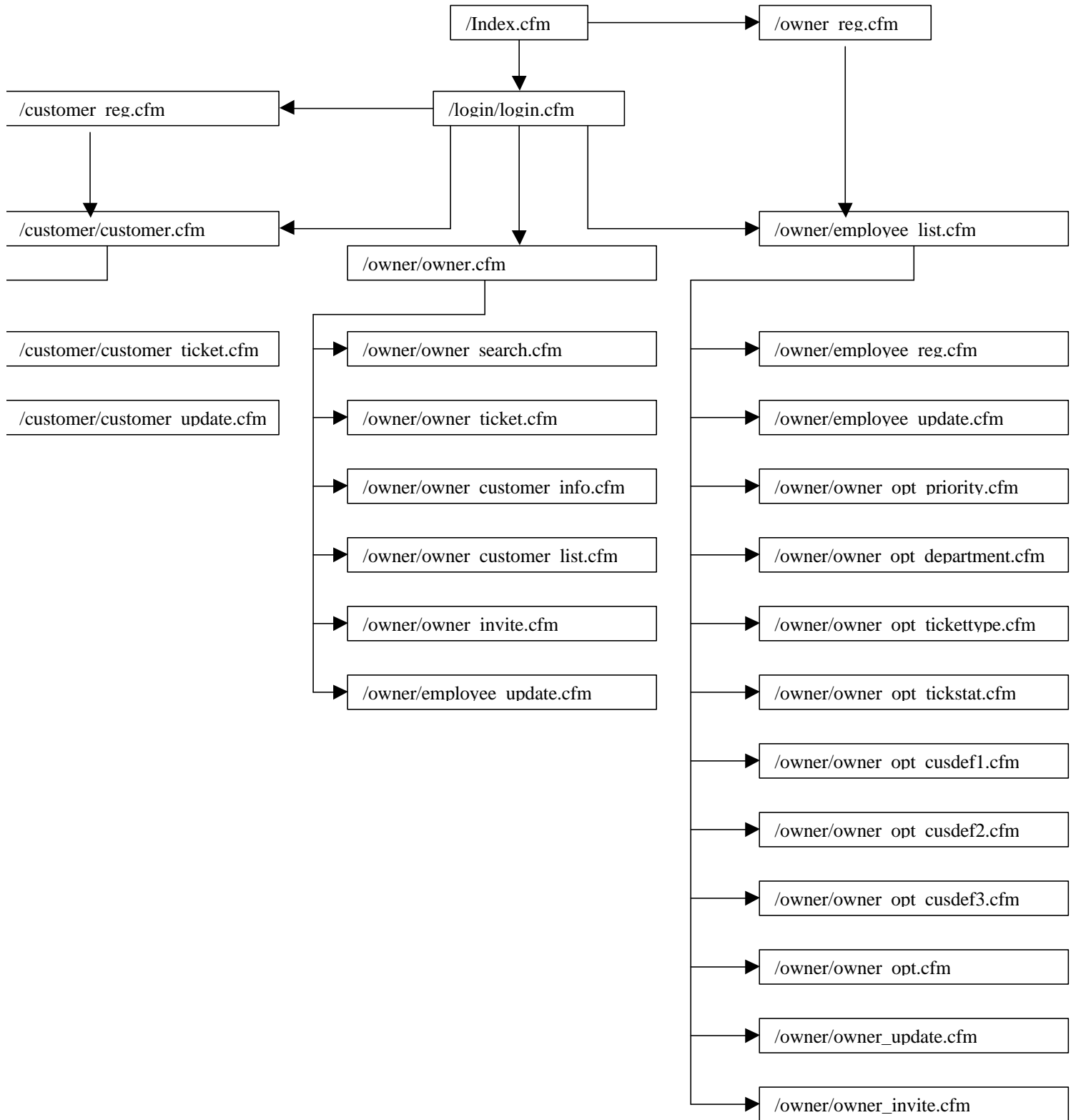
Vi har brukt stylesheets/CSS for å få et gjennomført design på systemet. Vanligvis ville vi inkludert stylesheets på vanlig HTML måte. Men siden det skulle være mulig for de forskjellige bedriftene å kunne sette sitt eget preg på siden med font og bakgrunnsfarge, har vi valgt og lage en egen stylesheets.cfm fil som henter variablene for font og bakgrunnsfarge i databasen. I stylesheets.cfm blir ting som fontstørrelser, tabellbakgrunnsfarger, tabellstørrelser, fontfarger og selectboksbredder satt.

Det meste av designet i del 1 og del 2 er bilder. Disse ligger ikke i stylesheets.cfm men direkte i header.cfm. De fleste bildene er programmert og komprimert mest mulig. Dvs. At vi isteden for å lage et bilde av en rett strek som er 200 piksler langt, har laget et bilde som 1 piksel bredt som gjentas 200 ganger. Dette vil gjøre siden mye raskere med det samme utseende.



### 3.6 Navigasjonsstruktur

For å vise navigasjonsstrukturen har vi laget et navigasjonskart se figur 3.7. Vi har delt opp systemet i 3 brukernivåer på ownersiden og 1 brukernivå på customersiden. Når en person logger inn i systemet sjekkes det først om brukeren er en customer eller employee. Hvis brukeren er en customer vil den få vist /customer/customer.cfm siden med en meny som er beregnet for kunden. Hvis brukeren er en employee sjekkes det hvilket nivå brukeren skal logges inn på. De tre nivåene er mottaker, bruker og administrator. Mottaker og bruker har de samme rettighetene, forskjellen er at en bruker vil bli logget rett inn på sin avdeling mens en mottaker blir logget rett inn til listen over nye meldinger. Begge kommer inn på /owner/owner.cfm, men det er forskjellige url-parametere som blir sendt med. Administrator vil bli logget inn til oppsettet av systemet og vil bli vist /owner/employee\_list.cfm siden hvor han/hun skal kunne legge til og endre brukere.



Figur 3.7 Navigasjonskart, for detaljert beskrivelse se vedlegg A

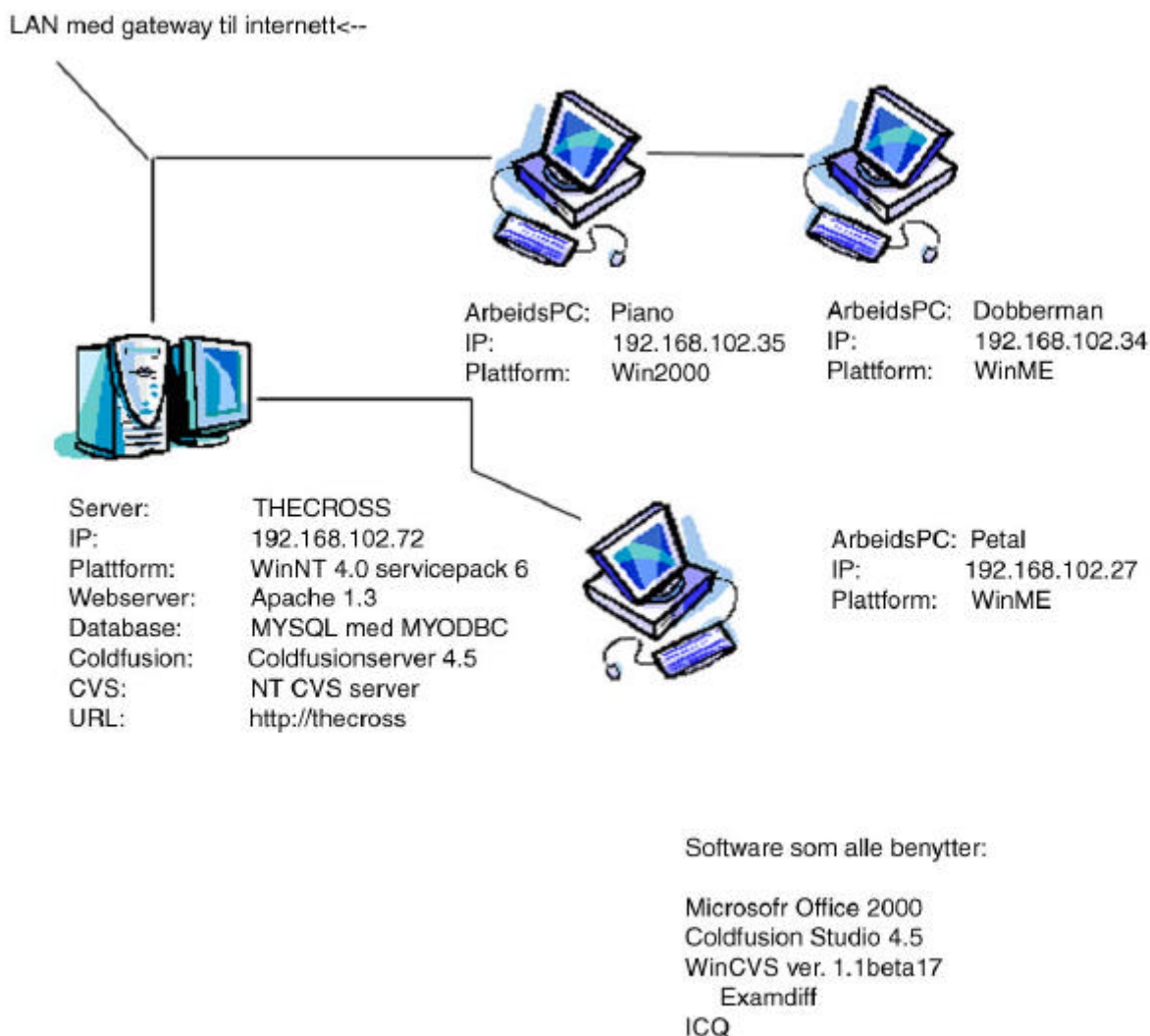


## 4 Implementering

I dette kapitlet tar vi for oss hvordan vi har utviklet applikasjonen. Vi starter med å gi en beskrivelse av utviklingsmiljøet. Vi går så videre med å fortelle hvilke verktøy vi har brukt, hvorfor vi har valgt dem og hvilken erfaring vi har fått av å bruke dem. Til slutt i kapitlet forteller vi om hvordan vi har kodet og viser noen eksempler på kode.

### 4.1 Utviklingsmiljø

Gruppen flyttet våre private PC til det tildelte grupperommet på skolen. Vi fant ut at det var lurt å sette opp en server, så vi tok kontakt med it-tjenesten. De var behjelpelig og vi fikk låne en PC. Vi installert NT server på maskinen selv om sterke krefter innad i gruppen ville ha en Linux server. Grunnen til at vi valgte NT var fordi fullverdig ColdFusion server ville bli mye dyrere. Koblet PC'ene sammen via nettverket på skolen. Se Figur 4.1.



4.1 Figur - Utviklingsmiljø



### **Detaljert beskrivelse av hvordan serveren(thecross) er satt opp:**

Da prosjektet vårt krever at vi har tilgang til en server med ColdFusion server installert, valgte vi å sette opp en egen server. Serveren er en P166MMX med 96MB RAM og 1,7BG HD.

#### **Webserver:**

Valgte å legge inn NT på den, da ColdFusion server for Unix er meget dyr å anskaffe. Vi satt opp Apache webserver. Produktet vårt består av en database knyttet mot web. Vi benytter MYSQL database med MYODBC som bindeledd mellom databasen og ColdFusion sidene.

#### **ColdFusion server 4.5**

La inn installasjonen av ColdFusion server 4.5 for NT. For å få ColdFusion serveren til å fungere sammen med Apache webserver måtte vi legge inn følgende linje i HTTPD.conf fila (konfigurerings fila til Apache):

```
LoadModule coldfusion_module modules/ApacheModuleColdFusion.dll
```

#### **MySQL Server and Clients 3.23:**

La inn MySQL for NT. Måtte legge inn brukere for å få tilgang til databasen fra andre maskiner. Dette gjøres ved å legge inn brukere i passordfila manuelt.

#### **MyODBC 2.5**

La inn MyODBC for NT for å knytte ColdFusion koden til MySQL databasen. Trengte ikke å konfigureres noe.

Grunnen til at vi har valgt å sette opp et utviklingsmiljø på skolen, er at vi mener at det er lettere å jobbe som en gruppe når vi sitter samlet. Vi har derfor flyttet pc-ene våre til skolen.

## **4.2 Valg av verktøy**

### **4.2.A Programmeringsspråk**

Vi har valgt å utvikle systemet i programmeringsspråket ColdFusion da oppdragsgiver helst så at vi brukte dette språket. Etter som oppdragsgiver fikk overbevist oss om ColdFusion's muligheter, utfordringen med å lære enda et programmeringsspråk og at dette er et språk som mange ledende firmaer bruker nå, var dette et enkelt valg. Alternativt kunne vi brukt asp, php, jsp m.fler. Det som kjennetegner alle disse språkene er at koden blir kjørt på serveren og vist som ren html til klienten. Javascript som derimot blir kjørt på klienten vil vi også bruke til enkle "fancy" ting. Felles for ColdFusion, asp og jsp er at det opprettes en ny tråd for hver gang en side blir kalt i stedet for at det startes en ny prosess, som det gjøres i CGI og php. Dette krever mindre ressurser.

#### **ASP**

ASP kjøres fritt på Windows NT/2000 Server med ISS, men er også oversatt så det kan kjøres på noen andre servere. Kildekoden består av ASP kode og HTML kode.

ASP sidene blir kjørt på serveren og sendt til klienten som vanlig html kode, det kreves derfor ikke noen ekstra ressurser av klienten. I motsetning til PHP

ASP - koden er basert på Visual Basic script. I Asp kan man lage sine egne komponenter med C++, Visual Basic eller Java. Asp krever mye maskinkraft av serveren

## PHP

PHP er veldig mye brukt nå, og er et godt alternativ som kan kjøres både på UNIX-baserte og Windows servere. Kildekoden består av PHP kode og HTML kode.

PHP sidene blir kjørt på serveren og sendt til klienten som vanlig html kode, det kreves derfor ikke noen ekstra ressurser av klienten. En ulempe her i motsetning til ASP er at det opprettes en ny prosess for hver forespørsel, ikke en tråd. PHP er opensource og det er derfor lett å finne relaterte kodesnutter til problemer. Det er også en fordel hvis man har vært borte i Perl. Finnes muligheter for Assosiative Arrays og regulære uttrykk.

## COLDFUSION

Allaire ColdFusion kan kjøres både på Windows og UNIX server. Og som php og asp kjøres koden på serveren for så å bli vist til klienten som ren html. Vi skal bruke en utviklingsutgave av ColdFusion for Windows. Denne utgaven er en billig utgave og server delen er sperret for mer enn en bruker. En fordel med ColdFusion er at hele administrasjonsverktøyet kan styres enkelt via Internett, gjennom hvilken som helst nettleser. Her kan man enkelt styre bla. Database aliaser og lignende, slik at hvis man bytter database. Endrer man kun her og slipper å gjøre noen endringer i selve koden. Det er nettopp denne databasehåndteringen som skal være en av de sterke sidene hos ColdFusion. En ulempe er at en komplett server utgave er ganske dyr i motsetning til php som er gratis

I tillegg til serverdelen følger det også med ColdFusion studio, som er et eget program beregnet for å utvikle ColdFusion applikasjoner. Man har her mulighet for å koble seg direkte opp mot databasen, for å kunne endre og ha innblikk i den. Andre fordeler med dette programmet skulle være at man kan bruke design vindu for å lette designet på siden(wysiwyg), men denne delen har vi funnet så ustabil at vi har valgt å ikke bruke den. Vi måtte derfor skrive også all html koden selv.

ColdFusion er også raskere enn både PHP og ASP. Det er også mer komprimert enn PHP og ASP. Mindre kode for samme funksjonalitet.

En ulempe i starten var at dette var ett språk vi ikke hadde noe kunnskap til og at det er litt annerledes syntaks enn andre språk vi er vant til.

Som kilder til disse sammenligningene har vi brukt: Websidene til Kurset Klient og Serversideprogrammering på HIG, Allaire Forums, egne erfaringer og Arne Sunde.

## Javascript

Siden vi ville ha minst mulige spørringer mot ColdFusion serveren ville vi benytte Javascript som begrenser disse spørringene. Vi brukte Javascript for å sjekke om forms hadde riktig input. Sjekket mot E-post adresse, og om passordene var skrevet inn likt når man skulle angi passord og deretter bekrefte det.

Vi bruker også Javascript for å laste ned en side på nytt. Dette for å slippe å ha flere forms på en side.

#### 4.2.B Database

Oppdragsgiver ville at vi skulle bruke MySQL siden det var gratis. Vi hadde minimalt med kjennskap til denne databasen, men gruppen hadde kjennskap til lignende databaser. Det fungerte bra å sette opp databasen, men vi fikk litt problemer med å sette opp koblingen mot ColdFusion serveren. Hadde ingen kjennskap med MyODBC koblingen mellom serveren og databasen, men etter å ha undersøkt saken nærmere gikk det greit. Etter hvert ble det bestemt at vi skulle begynne å bruke Oracle på den eksterne serveren. Databasestrukturen ble ikke endret, men det skapte enkelte syntaks problemer i kodingen. For nærmere detaljer om databaserstruktur se vedlegg A

#### 4.2.C Prosjektstyringsverktøy

Vi valgte å bruke Microsoft Project for å sette opp Gant-skjema vårt. Hadde ingen erfaring med slike verktøy fra før, men det ble anbefalt av veileder. Det var ingen problemer knyttet opp mot bruken av Microsoft Project, selv om det tok litt tid å sette seg inn i verktøyet.

Gruppen har også satt opp ett excel dokument som holder styr på hvor mange timer gruppen har brukt, og hva vi har gjort. For mer detaljer se vedlegg I

#### 4.2.D Systemutviklings verktøy

For å sette opp SA diagrammer i utviklingsfasen brukte vi Silverun. Silverun er et program som er mye brukt i systemutvikling sammenheng. Gruppen hadde også kjennskap til dette programmet fra før.

#### 4.2.E Versjons kontroll system

For å få kontroll over filene har vi brukt versjons kontroll systemet WinCVS. Dette gjør at man hele tiden får tilgang på siste versjon av filene og at man kan hente frem tidligere versjoner hvis nødvendig. Valgte å bruke WinCvs(betaversjon) siden dette var ett gratis program, kunne også brukt Microsoft Visual Sourcesafe men dette produktet er ikke gratis. En av medlemmene hadde også brukt WinCVS i forbindelse med jobbsammenheng slik at det ikke var helt ukjent.

Vi satte opp CVS-NT server. Alle i gruppa ble registrert som brukere. Det blir liggende siste versjon på alle arbeidsstasjonene og alle versjonene ligger på THECROSS.

### 4.3 Kode og Filstruktur

#### 4.3.A Prinsipper ved kode

Siden vi er 3 medlemmer i gruppen kan det være vanskelig å få full oversikt. Det kan være vanskelig å sette seg inn i koden med en gang. For å bedre dette har vi satt enkelte standarder ved programmeringen. Vi har brukt prefiksing ved variabel bruk, f. eks form.customer\_name, url.customer\_id. Vi har også brukt variabel navn som er lik navnene i databasen. Ved navngiving av queryer begynner alle navnene med "q", arrayer begynner alle navnene med "a". Vi har også kommentert koden.

## 4.3.B Overordnet kode

### 4.3.B.1 ColdFusion

Alle filene består av html- og ColdFusion-kode. ColdFusion koden kjennetegnes ved at den starter og slutter med ColdFusion-tager. Alle ColdFusion-tagene starter med <CF og avsluttes med </CF. Man kan ofte skrive html kode mellom disse, men det kommer an på hvilken ColdFusion-tag man bruker. Det kan også skrives attributter i ColdFusion-tagene. Noen attributter kan være obligatoriske mens andre er frivillige, det kommer an på hvilken ColdFusion-tag man skal benytte og hva hensikten med den skal være. For å utføre en spørring mot en database i ColdFusion bruker man <cfquery>-tagen. <cfquery>-tagene har 2 obligatoriske attributter, det er "name=" og "datasource=". "datasource="-attributten sier hvilken databasealias som skal benyttes. Databasealiasene og deres tilhørende databaser er definert på ColdFusion

Administrator sidene. "name="-attributten er variabelnavnet på listen som resultatet av sql-spørringen blir lagt i. Får å få tak i en spesiell kolonne må man skrive "*listenavn.kolonnenavn*".

#### Eksempel på kode:

Query:

```
<cfquery name="qtickettype" datasource="softcross" dbtype="ODBC">
select * from tickettype where owner_id = #session.owner_id#
order by tickettype_name
</cfquery>
```

Henter ut all data fra en spesifikk owner fra tabellen tickettype sortert etter navn.

Legger inn queryen i en array:

```
<cfset session.atickettype=ArrayNew(2)>
<cfoutput query="qtickettype">
<cfset
session.atickettype[ArrayLen(session.atickettype)+1][1]=#qtickettype.ticket
type_id#>
<cfset
session.atickettype[ArrayLen(session.atickettype)][2]=#qtickettype.ticketty
pe_name#>
<cfset
session.atickettype[ArrayLen(session.atickettype)][3]=#qtickettype.ticketty
pe_description#>
</cfoutput>
```

Legger inn all data fra queryen inn i en array. Dette gjøres i Login filen.  
Arrayen brukes til å skrive ut verdier fra databasen inn i en selectboks:  
Legger output fra array inn i en select boks:

```
<TR>
<TD colspan="2" >Meldingstype:</TD>
<TD colspan="2" >
  <SELECT size=1 name=tickettype_id >
    <cfif #qticket.tickettype_id# EQ "">
      <option value=""> Ikke satt</option>
    </cfif>
    <cfloop index="Counter" from=1
to="#ArrayLen(session.atickettype)#">
      <cfoutput>
        <option value="#session.atickettype[Counter][1]#"
          <cfif #qticket.tickettype_id# EQ
#session.atickettype[Counter][1]#>
            selected
          </cfif>
          >#session.atickettype[Counter][2]#</option>
        </cfoutput>
      </cfloop>
    </select>
  </td></tr>
```

Her sjekkes det om tickettype er satt, hvis ikke settes den til "ikke satt". Så går man gjennom en loop som går helt til arrayen er tom. Skriver ut verdiene fra arrayen ut i en select boks.

Ved å legge inn all data inn i arrayer i session når man logger seg inn, gjør at antall spørringer mot databasen blir redusert.

#### 4.3.B.2 Javascript

Vi brukte javascript på enkelte sider. Dette for validering og for innlasting av sider.

Eks på javascript:

```
function validate(){
  x=document.test
  alfa=x.owner_email.value.indexOf("@")
  punktum=x.owner_email.value.indexOf(".")
  pass1=x.owner_password1.value
  pass2=x.owner_password2.value
  number=x.owner_phone.value
  submitOK="True"

  if (alfa===-1 || punktum===-1 ) {
    alert("Ikke korrekt email adresse")
    submitOK="False" }

  if (pass1!=pass2) {
    alert("Ikke samsvar mellom passordene")
    submitOK="False" }
```

```
if (submitOK=="False") {  
    return false }  
}
```

Javascriptet sjekker om email adressen inneholder @ og . , det sjekker også om passordene er like. Returner true hvis all data er av riktig format.

```
<form method="post" action="owner_reg.cfm" ONSUBMIT="return validate()" name="test" >
```

Javascriptet kalles når man trykker på knappen og validate() funksjonen kjøres.

### 4.3.C Filstruktur

#### Sider med <form>s

Det som kjennetegner disse sidene er at det er felter som må fylles ut i dem. Når man trykker submit knappen, har vi valgt å ha den samme siden som action-side, Dette gjør at vi får færre sider og mindre kode. For å sjekke at de nødvendige feltene er fylt ut bruker vi et ekstra felt i formen for hvert felt. Dette feltet kan se ut som:

```
<input type="hidden" name="empl_name_required" value ="Du må fylle ut navn!" >
```

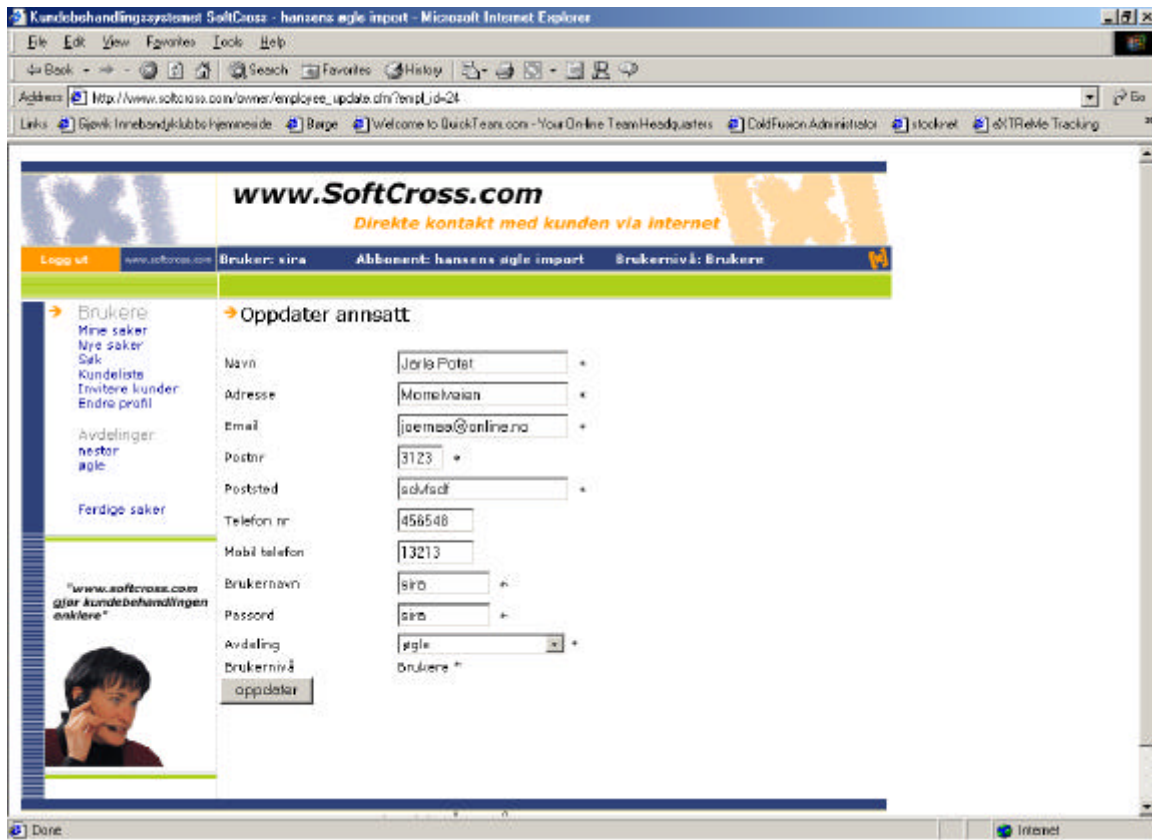
Vi kan lage den samme funksjonen med litt mer sexy utskrift i ColdFusion, men det krever at dataen blir sendt til serveren først. For å spare ressurser på serveren har vi derfor ikke valgt denne metoden.

For å sjekke om dataen er på riktig format vil vi bruke javascript. Dette kan også gjøres med ColdFusion-tags, men for å spare mer ressurser fra serveren gjør vi også dette med javascript.

Når nå all dataen er fylt ut blir den lagt til i databasen via en <cfquery>-tag og brukeren vil få opp en beskjed om at endringen er lagret.



Eksempel på en side med forms:



The screenshot shows a Microsoft Internet Explorer browser window displaying the SoftCross website. The address bar shows the URL: `http://www.softcross.com/owner/employee_update.cfm?empl_id=24`. The page header includes the SoftCross logo and the tagline "Direkte kontakt med kunden via internet". Below the header, there is a navigation bar with the text "Bruker: sine", "Abonnement: hansens egne import", and "Brukernavn: Brukere". The main content area is divided into two columns. The left column contains a menu with options like "Brukere", "Mine saker", "Nye saker", "Sak", "Kundeliste", "Invitere kunder", "Endre profil", "Avdelinger", "nestor", "egle", and "Ferdige saker". The right column is titled "Oppdater ansett" and contains a form with the following fields: "Navn" (Joris Potet), "Adresse" (Monselvaian), "Email" (joemea@online.no), "Postnr" (3123), "Poststed" (schvisd), "Telefon nr" (456548), "Mobil telefon" (13213), "Brukernavn" (sids), "Passord" (sids), "Avdeling" (egle), and "Brukernavn" (Brukere). A "Oppdater" button is located at the bottom of the form. The browser's status bar at the bottom shows "Done" and "Internet".

### Sider uten forms

Det som kjennetegner disse siden er at de som regel bli kalt med variable. Det vil si at den samme siden kan tilsynelatende se forskjellige ut etter som hvilke variable som blir sendt med. På f. eks login.cfm vil man få opp logoen til firmaet som abonnerer på siden når man sender med owner\_id demmes. Et annet eksempel er ticket sidene, der man sender med ticket\_no i urlen, det er nettopp denne variabelen som sier hvilken melding som skal vises på siden. Vi har selvfølgelig gjort dette så sikkert at man ikke kan skrive ticket\_no inn her som ikke er ens eget. Dette er gjort ved at Owner\_id er med i hver spørring mot databasen.





## 5 Testfasen - evaluering

For å kunne gjennomføre testfasen måtte vi flytte systemet vårt fra HIG til ElTeleØst for å gjøre systemet tilgjengelig på internett. Dette skapte enkelte problemer. Vi hadde i utgangspunktet fått beskjed om at vi skulle benytte MySQL database, men det ble gjort om av oppdragsgiver til Oracle. Dette medførte at vi måtte gjøre om endel på sql setninger og sql script for databasen. Vi kunne ikke ha auto increment, dobbelt frutter i sql setningene, og det var forskjellig måte å lage Outer Join setninger på. Denne jobben ble gjort både av oppdragsgiver og av gruppen selv. Heldigvis hadde vi avsatt tid til dette slik at de ikke gikk utover selve testingen.

Uken før påske satte gruppen i gang med selve testfasen. Vi sendte invitasjoner rundt om til utvalgte personer som skulle være med å teste systemet. Dette var meddstudenter, bekjente, og personer som kjente til slike systemer fra før. Personene som skulle teste på owner siden var personer som oppdragsgiver skulle kontakte og veileder.

Sammen med invitasjonen fulgte det med et spørreskjema som personene skulle fylle ut. Spørsmålene dreide seg om hvordan designet så ut, hvordan funksjonaliteten virket (se vedlegg B). Meningen med testfasen var å få vite hva som fungerte mindre bra og hva som var bra ved systemet..

Selve testfasen fungerte bra. Vi fikk testet systemet mot customere og mot owner siden. Selv om vi testet owner siden så burde vi kanskje fått testet med flere personer enn det som ble gjort. Antallet customer testere var vesentlig større enn på owner siden.

Ved testingen satt vi som Owner/customer, mens den andre parten satt som Customer/owner. Sendte meldinger frem og tilbake til hverandre samtidig som testeren testet andre funksjonaliteter.

Dette medførte at gruppen også oppdaget endel feil under testingen.

Fikk tilbake evalueringsskjemaene fra de fleste testerne selv om enkelte av evalueringsskjemaene ikke ga oss de tilbake meldingene vi ville ha. Noen var litt for sjenerøse med ros og ga litt for lite kritikk. Enkelte av skjemaene var veldig bra utfylt og med kommentarer som vi absolutt hadde behov for. Disse fikk oss til å få øynene opp for ting vi ikke hadde tenkt på før.

Generelle kommentarer til systemet:

- Funksjonalitet som ikke fungerte
- Designet
- Lite intuitive ord

Fikk overraskende positive kommentarer til systemet på owner siden. Brukerne var positive til wizard når man skulle registrere seg som owner og begynne å bruke systemet. Det bør nevnes at noen av testpersonene ikke hadde veldig erfaring med slike systemer før, og derfor kanskje ikke så behovet for funksjoner som ikke var der.

Etter vi hadde gjennomgått alle evalueringsskjemaene avtalte vi ett møte med oppdragsgiver for å gjennomgå disse feilene med han og deretter sette opp hvilken prioritet de skal ha. Under dette møtet satte vi opp feil som måtte rettes opp, trivielle endringer og deretter endringer av prioritet 1, 2 og 3 (se vedlegg D). Endringene som hadde prioritet 1 og 2 ble ansett som de viktigste for at systemet skulle fungere tilfredsstillende. Endringer av prioritet 3 kunne gjøres hvis vi fikk tid.

Oppdragsgiver gikk også gjennom endel forbedringer som kunne gjøres ved programmeringen også. Han hadde gått gjennom litt av koden som hadde blitt gjort i forkant av testfasen, og det var noen endringer som



## Hovedprosjektet SoftCross



måtte gjøres med hensyn på kapasitet/hastighet ved å bruke systemet. Vi måtte prøve å begrense bruken av SQL spørringer mot databasen, og heller prøve å legge variable i session når man logget/registerer seg i systemet. Etter dette møtet satte gruppen i gang med vildreutviklingen. Vi fikk rettet alle feil, trivielle endringer og endringer av prioritet 1. På prioritet 2 var det bare en endring vi ikke fikk gjort, på prioritet 3 ble det ikke gjort noe endringer. Oppdaget også enkelte småfeil underveis som måtte gjøres. En av endringene med prioritet 1 var designet. Under testfasen var designet nedprioritet, men etter litt rådslagning ble dette forandret helt.

## 6 Diskusjon av resultater

### 6.1 Prosessen

Vi valgte å bruke utviklingsmodellen oppdragsgiver utviklet i forbindelse med hovedoppgaven han skrev på UIO. Se vedlegg J . Fordelen med denne modellen er at den går i flere iterasjoner. Modellen er beregnet for å utvikle prototyper, for så å teste disse og da videreutvikle i ny iterasjon. Vi diskuterte å bruke bl.a. fossefall modellen, men denne egner seg dårlig siden prosessen etter testfasen er viktig. Å kjøre en testfase var noe vi syntes virket veldig interessant.

Vi satte opp Gantskjema for gjennomføring av prosjektet under forprosjektet. Gruppen har ved få unntak fulgt dette skjemaet. Kravspesifikasjonen ble en uke forsinket, videreutviklingen måtte vi kutte ut en uke for heller å satse på rapporten og planleggingen av testfasen ble flettet inn sammen med kodingen før testfasen. Gruppen hadde to eksamener før påske. Disse ble flyttet en uke og arbeidet med hovedprosjektet like dann.

Dersom vi hadde hatt lengre tid på prosjektet, ville vi kjørt en iterasjon til. Etter denne iterasjonen ville produktet vært tilnærmet ferdig. Vi ville da få testet brukergrensesnitt og design mot brukere, utviklet ferdig administrator delen og hjelpefunksjoner.

Gruppen har gjennom kodenfasen blitt bedre og bedre på koding i ColdFusion og avanserte SQL setninger mot databasen. Dette har medført at mye av kodingen har blitt gjort flere ganger i stadig bedre versjoner. Vi har også blitt mer beviste på å kode med hensyn på serverhastighet og kapasitet.

Et eksempel på dette kan være:

#### Tidlig kode

Her har vi en Query for å hente ut tickstat\_name til en ticket. Tickstat\_id er en attributt i ticket. For å få ut tickstat\_name kjørte vi da en Query inne i en Query slik at vi brukte tickstat\_id'en vi hadde funnet i ticket i den andre queryen slik at vi fant riktig tickstat\_name.

```
<cfquery name="qticket" datasource="softcross" dbtype="ODBC">
    select tickstat_id
    from ticket
    where ticket.customer_id = #session.customer_id#
    order by ticket_no desc
</cfquery>

<cfoutput query="qticket">
    <cfquery name="qticketstat" datasource="softcross" dbtype="ODBC">
        select tickstat_name, tickstat_id
        from ticketstatus
        where tickstat_id = #qticket.tickstat_id#
    </cfquery>
    #qticketstat.tickstat_name#</TD>
</cfoutput>
```

#### Koden slik vi kodet det til slutt

Etter å ha programmert ColdFusion en stund fant vi ut at det gikk an å få ut tickstat\_name til en ticket med en SQL Query. Vi bruker her outer join slik at vi får where argumentet i SQL setningen til å gi oss riktig tickstat\_name. Outer join gjør også at du får med alle ticketene også de der tickstat\_id ikke er satt.

```
<cfquery name="qticket" datasource="softcross" dbtype="ODBC">
select ticket_no, tickstat_name,
from ticket, ticketstatus
where
  ticket.customer_id = #session.customer_id# and
  ticketstatus.tickstat_id(+) = ticket.tickstat_id
order by ticket_no desc
</cfquery>

<TABLE>
<TR><TD>ticket nr</TD><TD>Ticket status</TD></TR>
  <cfoutput query="qticket"><TR><TD>#qticket.ticket_no#</TD>
    <TD>#qticket.tickstat_name#</TD></TR>
</cfoutput>
```

Vi klarte å gjennomføre nesten alle målene fra forprosjektet. Målene ble noe forandret underveis da oppdragsgiver så potensialet i prototypen vår. Administrator delen for systemansvarlig ble nedprioritet til fordel for utvidelser som bra bl.a. brukergrensesnitt og søkefunksjoner.

Gruppen har laget et system som ligger på en sentral webserver. Mange bedrifter kan bruke dette systemet samtidig og tilpasse det slik de vil ha det. Farger, skrifttyper og størrelser, logo og åpningstekst kan tilpasses hver enkelt bedrift. Bredriften kan også sette opp egendefinerte felter i forbindelse med meldingssidene.

Systemet skulle også være så lett å bruke at ansatte hos bedrifter skulle kunne benytte systemet uten videre opplæring. Dette ser vi at vi har oppnådd til en viss grad. For at en Owner skal klare å registrere seg i systemet kreves det noe kunnskap om internett og at man har lest brukerveiledningen.

#### VEDLEGG

Gruppen har utviklet en prototype og kjørt to itterasjoner dvs. kodet en prototype, testet systemet mot både Owner og Customer og rettet feil.

## 6.2 Produktet

Vi er godt fornøyde med resultatet. Prosjektet har resultert i en kjørbar prototype som har vært gjennom et iterasjon. Prototypen har også blitt forbedret og klargjort for ny testing.

Prosjektet har også resultert i mye kunnskap innen systemutviklingsprosesser for gruppa i henhold til målsetningen.

### Forbedringer / videreutvikling:

Da tiden ble noe knapp på slutten av prosjektet, valgte vi å ikke legge til disse modulene. Dette bør gjøres før produktet lanseres på markedet.

Avtale om kunde med responstid:

Dette vil være en avtale med mellom owner og customer om hvor lang tid owner skal ha til å svare på en sak. I den forbindelse ville et system med trafikklys hvor de tre fargene ville indikere hvor lang tid det er igjen før man må svare være aktuelt.



Administrator del:

For å kunne administrere systemet på en lettere måte enn å skrive SQL setninger rett mot databasen trenger man å utvikle en administrator del. Tanken var å lage et enkelt oppsett med moduler for sletting og oppretting av nye data, rapportutskrifter og fakturagrunnlag. Muligheten ligger i databasen, slik at mye av dette kan gjøres ved spørringer direkte mot databasen. Bør lages nye sider for dette.

Tilsending av e-post hvis man har glemt passord kan også være en fin forbedring.

Bedre "help" funksjoner på sidene er også noe som bør implementeres. Vi har laget et lite brukerhefte VEDLEGG, men slik hjelp burde ligge på sidene.

### 6.3 Gruppearbeidet

Gruppen har samarbeidet bra. Grupperommet på HIG har fungert flott. Gruppen har hatt med seg egne maskiner og sittet i et tett og nært utviklingsmiljø. Det har også være to andre grupper og en annen student på grupperommet. Generelt sett har dette gått bra, men i innspurten har det vært eksamener på grupperommet slik at vi ikke kunne befinne oss på rommet. Dette har ført til litt komplikasjoner i innspurten.

Det har blitt holdt ukentlige mandagsmøter der vi har diskutert forrige ukes resultater og videre arbeid. Vi har hatt ukeplaner med hver enkelts arbeidsoppgaver. Vi har også hatt workshops hvor vi har diskutert diverse problemer ved hjelp av bl.a. brainstorming. Gruppen har klart å komme frem til løsninger som alle kan stå inne for.

De ulike medlemmene i gruppen har utfylt hverandre bra. Alle gruppe medlemmene har deltatt i alle fasene av prosjektet. Som de fleste grupper i slike prosjekter har vi hatt situasjoner der noen har hatt bedre bakgrunnskunnskaper og derfor sterkere meninger og syn på hvordan de ulike delene skal gjennomføres. I enkelte tilfeller har kommunikasjonen sviktet noe og har ført til noe dobbeltarbeid. Tonen mellom gruppemedlemmene har vært bra. Gruppen har også møttes ofte til en pizza og rødbrus for sosialt samvær.

Gruppen har ført loggbok over hvor mange timer vi har jobbet, og hva vi har gjort. Gruppen har sammenlagt jobbet ca 985 timer fordelt slik:

- Børge 281.5 timer
- Jørn 377.5 timer
- Kristoffer 326 timer

<b>12-Mar</b>	Børge	Programmering customer	3
	Kristoffer	programering av ticket, update og insert	3.5
	Jørn Arild	programering owner options	3.5
<b>13-Mar</b>	Kristoffer	Ticket	1.5
<b>14-Mar</b>	Kristoffer	Ticket + litt på owner_reg/update customer_reg/update	7
	Jørn Arild	Masse forskjellig	7.5
	Børge	Design, header, footer	5.5
<b>15-Mar</b>	Kristoffer	Møte med tom, ticket, testing	5.5
	Børge	Møte med tom, ticket, testing	4.5
	Jørn Arild	Møte med tom, ticket, testing	5.5
<b>19-Mar</b>	Kristoffer	Mandagsmøte, programmering	6
	Børge	Mandagsmøte, programmering	6
	Jørn Arild	Mandagsmøte, programmering	6
<b>21-Mar</b>	Kristoffer	Statusmøte med Tom	1
	Børge	Statusmøte med Tom	1
	Jørn Arild	Statusmøte med Tom, klargjøring for web, programmering	5

Figur 6-1 – Utdrag fra loggbok

## 6.4 Oppdragsgiver

Helt fra planleggingsfasen har Arne Sunde vært en flott støttespiller. Han har vært ivrig på å møte oss både på Gjøvik og i Oslo. Den faglige kompetansen innen ColdFusion og database/web har vi satt stor pris på. Arne gjennomførte et to dagers kurs innen ColdFusion som gav oss en bra start. Vi har også hatt god kommunikasjon via e-post og telefon.

I begynnelsen av prosjektet fikk vi tilnærmet frie tøylere innen utforming av prosjektet. Men etter 1. iterasjon ble kravene fra oppdragsgiver mer spesifikke. Under testfasen skulle han skaffe og teste mot Owner siden med folk fra ElTeleØst, men det ble ikke noe av og det meste av testingen på Owner siden ble utført av Arne selv. Feedbacken var allikevel konstruktiv og god.

Arne er en hyggelig kar.

## 6.5 Veileder

Tom Røise har vært en bra veileder. Tross at vi glemte det 1. veiledermøte tilga han oss og møtte ivrig opp. Vi har hatt stor nytte av veileders faglige kompetanse innen systemutviklingsfaget. Han har også vært behjelpelig ved gjennomgang av dokumenter og påpeke mangler og forbedringer. Den faglige kunnskapen innen programmering har han ikke kunnet bidratt så mye med.

Tom Røise var også en ivrig tester av Softcross under testfasen og kom med masse positiv og oppmuntrende kritikk i en noe slitsom periode.

Tom er en hyggelig kar.

## 6.6 Veileder og Oppdragsgiver

Veileder og Oppdragsgiver har passet gruppen glimrende. De har utfylt hverandre på en fin måte. De hadde et noe ulikt syn når det gjaldt fordelingen på arbeidsmengden mellom rapporten og koding på slutten av prosjektet. Veileder mente vi burde satse på en bra rapport, mens Oppdragsgiver mente vi heller skulle prioritere koding. Vi valgte å legge mest vekt på rapporten i slutfasen av prosjektet.



## 7 Konklusjon

Gruppen har jobbet tett sammen i ett halvt år. Ingen hadde på forhånd vært borte i lignende prosjekter før. Selv om det har vært mye usikkerhet angående tidsrammer, kvalifikasjoner innen enkelte fagfelt og hvordan kommunikasjonen både innad i gruppen og med eksterne aktører (veileder og oppdragsgiver) har prosjektarbeidet fungert nesten knirkefritt. Vi mener vi har kommet fram til gode løsninger på problemene ved hjelp av gode arbeidsmetoder. Gruppen har lært veldig mye innen systemutviklingsprosesser, lært ett nytt programmeringsspråk og synes selv vi har kommet frem til en bra testbar prototype. For å gjøre systemet testbart ble konfigurasjonsmulighet fra owner siden, det at det er en generell applikasjon hvor hver bedrift tror at man har sitt eget system og mer vekt på layout, prioritert foran hjelpefunksjoner og administratorsider. Tiden ble derfor for knapp til at vi fikk lagd noen hovedadministratordel i det hele tatt. Produktet fungerer fint uten denne delen, men å generere fakturagrunnlag og administrasjon av systemet blir mye tyngre da dette må gjøres rett mot databasen.

Å utvikle en prototype og gjennomføre en fullstendig testfase mot simulerte kunder av systemet, var spesiels lærerikt. Det er kunden som bestemmer om et produkt er bra og utviklere har ikke likt syn på produktet som kundene. Vi er derfor glade for at vi valgte å bruke den utviklingsmodellen vi gjorde og ikke den tradisjonelle "fossefall" modellen.

Da vi valgte prosjektet softcross var vi som alle andre usikre på hvilket hovedprosjekt vi skulle velge. Vi er i ettertid glade for at vi valgte prosjektet softcross. Vi føler vi har tilegnet oss nyttig kunnskap som vi med stor sannsynlighet kommer til å få bruk for i en fremtidig jobb situasjon.

Vi håper oppdragsgiver vil benytte prototypen vår til videre utvikling av Softcross!

## 8 Litteraturliste

- [GrSu96] Werner Grini og Arne Sunde: *Kundeorienterte Web-systemer*, Hovedsfagoppgave november 1996 Ved UIO
- [GAAJAC99] Gutzman, Alexis & Ashenfelter, John & Arehart, Charles, *ColdFusion for dummies 4*, 1999
- [DAMK00] Danesh, Arman & Motlagh, Kristin: *Matering ColdFusion 4.5*, 2000
- [MoCh00] Mohnike, Charles: *Sams teach yourself ColdFusion in 21 days*, 2000
- [Brukermanual] Alaire, *Brukermanualer til ColdFusion 4.5*
- [BEHVW96] Bjune, Tove E & Holmstedt, Viggo & Weiseth, Per Einar : *Systemutvikling*, 1996
- [Som95] Ian Sommerville: *Software Engineering*, 1995.
- [Har99] Don Harper: *WinCvs Version 1.1*, 1999.
- [WebA] ColdFusion Support Forum  
<http://forums.allaire.com/coldfusion/>
- [WebB] W3Schools Online Web Tutorials  
<http://www.w3schools.com/>
- [WebC] Høgskolen i Gjøvik  
<http://www.hig.no>
- [WebD] EarthWeb.com: The IT Industry Portal - JavaScripts  
<http://www.javascripts.com>
- [IRC] IRC kanaler  
EfNet (#java, #javascript, #sql, #mySQL, #oracle, #kallerud)

## 9 Vedlegg

<a href="#"><u>Vedlegg A</u></a> <a href="#"><u>Designdokument</u></a> .....	54
<a href="#"><u>Vedlegg B</u></a> <a href="#"><u>Evaluerings skjemaer</u></a> .....	75
<a href="#"><u>Vedlegg C</u></a> <a href="#"><u>Resultat av testfasen</u></a> .....	87
<a href="#"><u>Vedlegg D</u></a> <a href="#"><u>Endringsskjema</u></a> .....	90
<a href="#"><u>Vedlegg E</u></a> <a href="#"><u>Gantskjema</u></a> .....	92
<a href="#"><u>Vedlegg F</u></a> <a href="#"><u>Eksempler på kildekode</u></a> .....	95
<a href="#"><u>Vedlegg G</u></a> <a href="#"><u>Eksempel på møtereferater</u></a> .....	115
<a href="#"><u>Vedlegg H</u></a> <a href="#"><u>Statusrapporter</u></a> .....	119
<a href="#"><u>Vedlegg I</u></a> <a href="#"><u>Tidsbruk</u></a> .....	126
<a href="#"><u>Vedlegg J</u></a> <a href="#"><u>Utviklingsmodell</u></a> .....	131
<a href="#"><u>Vedlegg K</u></a> <a href="#"><u>Brukerveiledning</u></a> .....	133
<a href="#"><u>Vedlegg L</u></a> <a href="#"><u>Eksempel på ukeplan</u></a> .....	145



**Vedlegg  
A  
Designdokument**

## 1 Introduksjon

### 1.1 Formål

Formålet med dette it-systemet er å forenkle bedrifters kontakt med deres kunder, ved å utnytte potensialet for kommunikasjon over Internett. Særlig interessant er det at flere bedrifter skal kunne bruke systemet på samme webserver, slik at dette er en tjeneste alle bedrifter kan bruke uten å måtte installere det hos dem selv. Systemet skal derfor være generelt, med mulighet for at bedriftene selv skal kunne det sette opp.

### 1.2 Kvalitetsmål

Tabell 1-1 viser vår prioritering av generelle designkriterier. Det er lagt stor vekt på flyttbarhet, vedlikeholdbart, testbart og gjenbrukbart da, fordi systemet vi skal utvikle er en prototype som skal kunne videreutvikles. Siden systemet skal videreutvikles av oppdragsgiver, har vi latt han velge hvilket programmeringsspråk som skal brukes. Det har derfor blitt satt som et kriterium at systemet skal lages i en kombinasjon av HTML og ColdFusion.. Brukbarhet og sikkerhet er ikke lagt så stor vekt på siden nå, men bør legges vekt på i det ferdige produktet.

Prioritering av designkriterier

Kriterium	Meget viktig	Viktig	Mindre viktig	Irrelevant	Trivielt oppfylt
Brukbart		X			
Sikkert		X			
Effektivt					X
Korrekt			X		
Pålitelig					X
Vedlikeholdbart	X				
Testbart	X				
Fleksibelt		X			
Forståelig		X			
Gjenbrukbart	X				
Flyttbart	X				
Integrerbart					X

Tabell 1-1

### 1.3 Teknisk plattform

Systemet skal være mest mulig uavhengig av teknisk plattform. Dette vil vi oppnå ved å benytte oss av ColdFusion og HTML. Eksisterende utstyr, basisprogramvare og systemer skal utnyttes. Brukerne forventes å kjøre de meste brukte nettleserne.

#### Utstyr

Systemet planlegges utviklet ved bruk av grafiske arbeidsstasjoner mot en NT-server, men dette er ikke et krav. Klientene må ha maskiner kraftige nok til å kunne kjøre de mest brukte nettleserne.

#### Systemer og apparater

Som nevnt tidligere benytter dette systemet Internett som kommunikasjonsmediet. Det er derfor viktig at systemet fungerer godt sammen med ulike nettlesere. Skjermbildene og kommunikasjonen med klientene må derfor baseres på standarder, som kan tolkes av de mest brukte nettleserne.

Tanken er at systemet på en måte skal kunne kommunisere med oppdragsgivers regnskapssystem. Denne delen av systemet er blitt nedprioritert i denne omgang, siden vi først og fremst skal lage en brukbar prototype som skal kunne testes av kunder.

## 2 Kode og datastruktur

### 2.1 Metode

Vi har valgt å utvikle systemet i programmeringsspråket ColdFusion da oppdragsgiver helst så at vi brukte dette språket. Etter som oppdragsgiver fikk overbevist oss om ColdFusion's muligheter, utfordringen med å lære enda et programmeringsspråk og at dette er et språk som mange ledende firmaer bruker nå, var dette et enkelt valg. Alternativt kunne vi brukt asp, php, jsp m.fler. Det som kjennetegner alle disse språkene er at koden blir kjørt på serveren og vist som ren html til klienten. Javascript som derimot blir kjørt på klienten vil vi også bruke til enkle "fancy" ting.

### 2.2 ColdFusion

Allaire ColdFusion kan kjøres både på Windows og UNIX server. Og som php og asp kjøres koden på serveren for så å bli vist til klienten som ren html. Vi har brukt en utviklingsutgave av ColdFusion for Windows. Denne utgaven er en billigutgave og server delen er sperret for mer enn en bruker. En fordel med ColdFusion er at hele administrasjonsverktøyet kan styres enkelt via Internett, gjennom hvilken som helst nettleser. Her kan man enkelt styre bla. Database aliaser og lignende, slik at hvis man bytter database. Endrer man kun her og slipper å gjøre noen endringer i selve koden. Det er nettopp denne databasehåndteringen som skal være en av de sterke sidene hos ColdFusion.

I tillegg til serverdelen følger det også med ColdFusion studio, som er et eget program beregnet for å utvikle ColdFusion applikasjoner. Man har her mulighet for å koble seg direkte opp mot databasen, for å kunne endre og ha innblikk i den. Andre fordeler med dette programmet skulle vært wysiwyg, men denne delen har vi funnet så ustabil at vi har valgt og ikke bruke den.

### 2.3 Overordnet kode/data struktur

Alle filene består av html- og ColdFusion-kode. ColdFusion koden kjennetegnes ved at den starter og slutter med ColdFusion-tager. Alle ColdFusion-tagene starter med <CF og avsluttes med </CF. Man kan ofte skrive html kode mellom disse, men det kommer an på hvilken ColdFusion-tag man bruker. Det kan også skrives attributter i ColdFusion-tagene. Noen attributter kan være obligatoriske mens andre er frivillige, det kommer an på hvilken ColdFusion-tag man skal benytte og hva hensikten med den skal være. For å utføre en spørring mot en database i ColdFusion bruker man <cfquery>-tagen. <cfquery>-tagene har 2 obligatoriske attributter, det er "name=" og "datasource=". "datasource=" attributten sier hvilken databasealias som skal benyttes. Databasealiasene og deres tilhørende databaser er definert på ColdFusion Administrator sidene. "name=" attributten er variabelnavnet på listen som resultatet av SQL-spørringen blir lagt i. Får å få tak i en spesiell kolonne må man skrive

”listenavn.kolonnenavn”.

Kode	Kommentar
<pre>&lt;cfoutput&gt; &lt;b&gt; Hei verden, med fet tekst &lt;/b&gt; &lt;/cfoutput&gt;</pre>	<p>Kode for utskrift på skjerm: Utskrift: Hei verden, med fet tekst</p>
<pre>&lt;cfquery name="qowner" datasource="softcross" maxrows="5"&gt; select * from owner &lt;/cfquery&gt;</pre>	<p>ColdFusion-kode med attributter og sql. &lt;cfquery&gt;-tagen er en tag for kjøre et sql uttrykk på en database. Her er "name" og "datasource" to obligatoriske attributter, mens "maxrows" er en frivillig attribut. "qowner" er variabelnavnet man må bruke for å få tak i resultatet av sql-spørringen. "softcross" er aliasnavnet til databasetilkoblingen. Dette er som nevnt satt opp på administrasjonssidene til serveren. "maxrows" er hvor mange rader som skal hentes ut, hvis ikke denne er satt hentes alle ut.</p>
<pre>&lt;cfoutput query="qowner"&gt; #qowner.owner_name# &lt;/cfoutput&gt;</pre>	<p>Utskrift av ovenstående sql-spørring. Dette vil skrive kolonnen "owner_name" som er hentet fra sql-spørringen over. Attributten query="qowner" kan sammenlignes med "for each" i php. Den sørger for at &lt;cfoutput&gt;-tagen looper lik antall ganger det er linjer som kommer fra spørringen. #qowner.owner_name# er en variabel som sier at resultatet fra kolonnen "owner_name" i spørringen "qowner" skal skrives ut på skjermen.</p>
<pre>&lt;cfif isDefined("navn")&gt; &lt;cfoutput&gt; Du heter #navn#  &lt;/cfoutput&gt; &lt;cfelse&gt; Jeg vet ikke hva du heter &lt;/cfif&gt;</pre>	<p>ColdFusion-kode med variabler og if/else statement: Her sjekkes det om variabelen <i>navn</i> finnes, hvis den f. eks inneholder "Jørn" skrives det ut på skjermen: Du heter Jørn Hvis variabelen ikke finnes skrives det ut: Jeg vet ikke hva du heter Grunnen til at det er med &lt;cfoutput&gt;tag ved den ene hendelsen og ikke ved den andre, er at man trenger denne taggen for å skrive ut en variabel til skjerm.</p>

**Tabell 2-1**

## 2.4 Katalogstruktur

For at det ikke skal bli så mye rot på serveren og for at filarkitekturen skal bli mer oversiktlig, har vi valgt og lage noen underkataloger på serveren.

De katalogene som er valgt er:

- / - Dette er rotkatalogen, her legges filer som har med registrering av nye kunder å gjøre. *Mulig også sider som har med info om softcross å gjøre.*
- /admin - Denne katalogen er satt av til admin-sidene for selve systemet (Denne delen har blitt nedprioritert)
- /customer - Her ligger alle filene blir brukt når en er logget inn som customer .
- /images - Her ligger alle bilder som er brukt til designet av sidene
- /login - Her ligger loginsiden
- /owner - Her ligger alle filene blir brukt når en er logget inn på ownersiden.
- /upload - Her blir alle filer som er opplastet av brukerne lagt.

I alle katalogene med coldfusion filer ligger det en Application.cfm fil. Dette er filer som ColdFusion serveren kjører før noen av de andre filene blir kjørt. Hvis alle disse filene hadde vært like, kunne vi byttet dem ut med kun en fil som vi hadde lagt i roten. Hvis ColdFusion Serveren ikke finner en Application.cfm i katalogen hvor den forespurte filen ligger , bruker den filen som ligger i katalogen over. Finner den ikke filen her heller så letes det etter den i katalogen over osv.

Grunnen til at vi har forskjellige typer av denne filen er at vi i Application.cfm filene i /customer og /owner sjekker at brukeren virkelig er logget inn. Hvis ikke brukeren er logget inn, blir dem sendt til /login/login.cfm og må logge inn. På denne måten hindrer vi at man kan gå inn på sidene uten og logge inn, noe som ville vært veldig dårlig med hensyn på sikkerhet.



## 2.5 Filstruktur

Da det er store funksjonelle forskjeller på sidene, har vi benyttet litt forskjellige strukturer i dem.

### Sider med <form>s

Det som kjennetegner disse sidene er at det er felter som må fylles ut i dem. Når man trykker submit knappen, har vi valgt å ha den samme siden som action-side, Dette gjør at vi får færre sider og mindre kode. For å sjekke at de nødvendige feltene er fylt ut bruker vi et ekstra felt i formen for hvert felt.

Dette feltet kan se ut som:

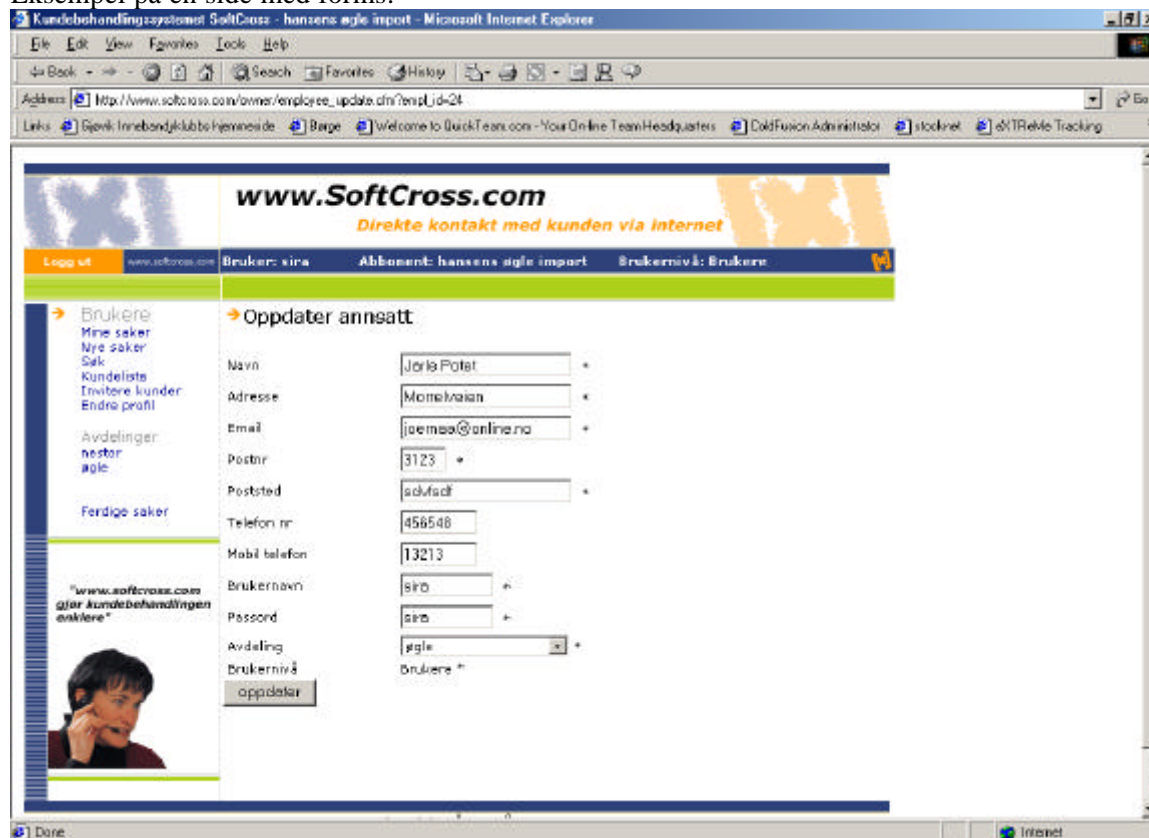
```
<input type="hidden" name="empl_name_required" value="Du må fylle ut navn!" >
```

Vi kan lage den samme funksjonen med litt mer sexy utskrift i ColdFusion, men det krever at dataen blir sendt til serveren først. For å spare ressurser på serveren har vi derfor ikke valgt denne metoden.

For sjekke om dataen er på riktig format vil vi bruke javascript. Dette kan også gjøres med ColdFusion-tags, men for å spare mer ressurser fra serveren gjør vi også dette med javascript.

Når nå all dataen er fylt ut blir den lagt til i databasen via en <cfquery>-tag og brukeren vil få opp en beskjed om at endringen er lagret.

Eksempel på en side med forms:



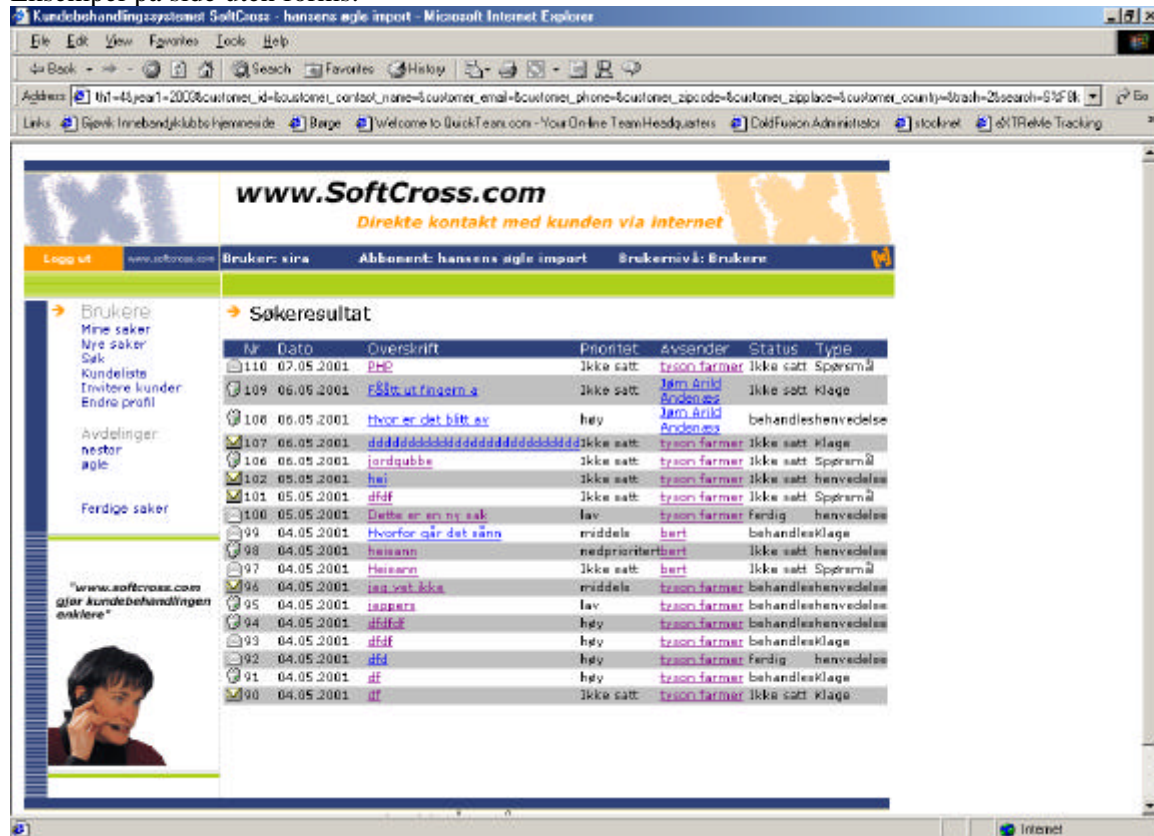
Figur 0-1

### Sider uten <form>s

Det som kjennetegner disse siden er at de som regel bli kalt med variable. Det vil si at den samme siden kan tilsynelatende se forskjellige ut etter som hvilke variable som blir sendt med. På f. eks login.cfm vil man få opp logoen til firmaet som abonnerer på siden når man sender med owner\_id demmes. Et annet eksempel er ticket sidene, der man sender med ticket\_no i urlen, det er nettopp denne variabelen som sier hvilken melding som skal vises på siden. Vi har selvfølgelig gjort dette så sikkert at man ikke kan skrive ticket\_no inn her som ikke er ens eget.

Alle sidene som en ser når man er logget inn inkluderer sidene header.cfm og footer.cfm. I header.cfm ligger toppen og menyene til alle sidene. Hvilken meny som vies kommer ann på hva man er logget inn som. Hvis man ikke er logget inn vises et loginfelt. I footer.cfm avsluttes alle tabeller som begynnes på i header.cfm.

### Eksempel på side uten forms:

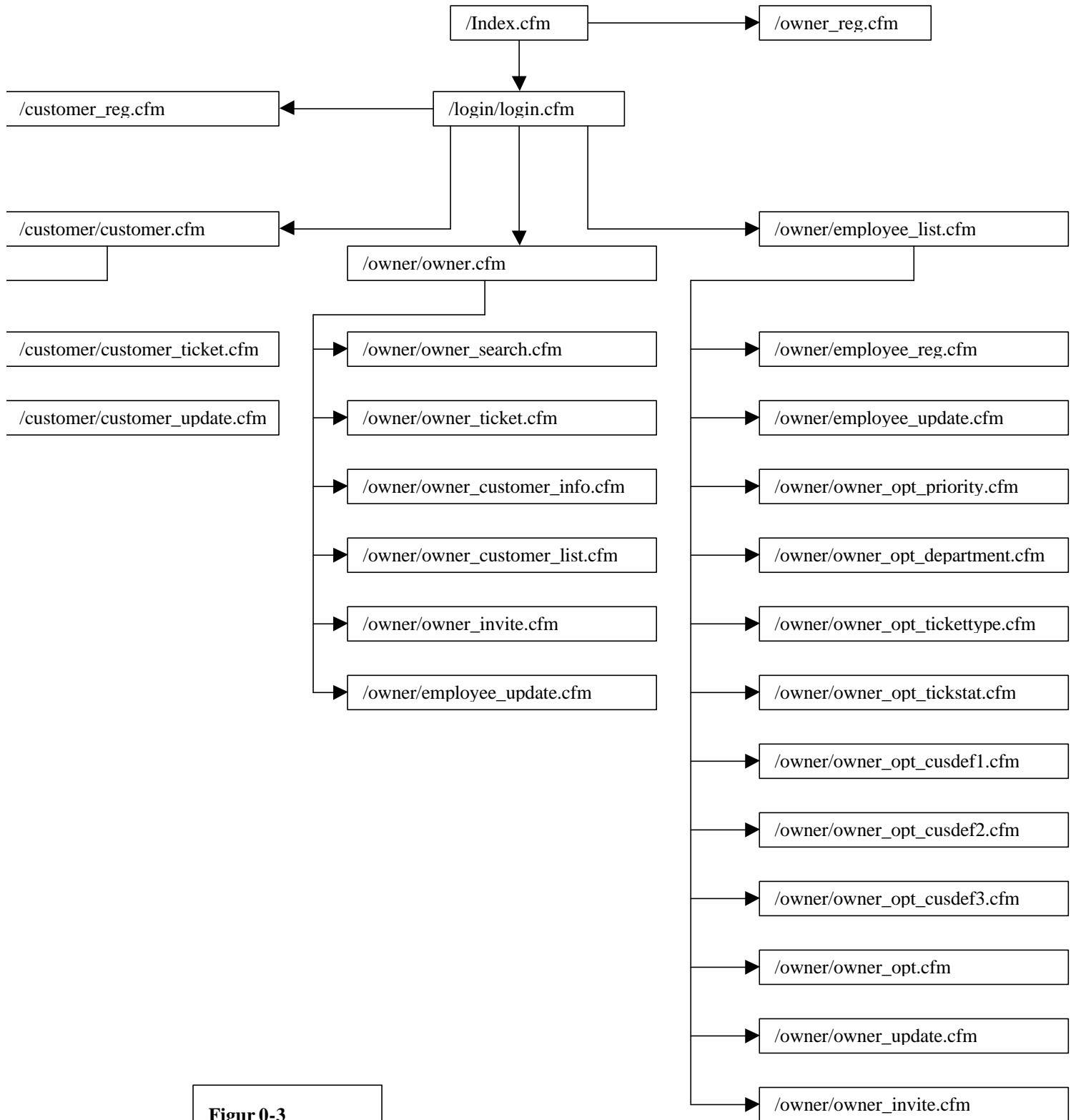


The screenshot shows a web browser window displaying the SoftCross website. The page title is "Kundebehandlingssystemet SoftCross - hansens egne import - Microsoft Internet Explorer". The address bar shows a URL with search parameters. The website header includes the SoftCross logo and the tagline "Direkte kontakt med kunden via internet". Below the header, there is a navigation bar with "Logg ut", "Bruker: vira", "Abonnement: hansens egne import", and "Brukernavn: Brukere". The main content area is titled "Søkeresultat" and displays a table of search results. The table has columns for "Nr", "Dato", "Overskrift", "Prioritet", "Avsender", "Status", and "Type". The results show various tickets with their respective dates, titles, priorities, senders, and statuses.

Nr	Dato	Overskrift	Prioritet	Avsender	Status	Type
110	07.05.2001	BHE	Ikke satt	tyson-farmer	Ikke satt	Spørsmål
109	06.05.2001	F&S utringem a	Ikke satt	Jørn Arild Andenæs	Ikke satt	Klage
108	06.05.2001	Hvor er det blitt av	høy	Jørn Arild Andenæs	behandles	henvendelse
107	06.05.2001	ddddd	Ikke satt	tyson-farmer	Ikke satt	Klage
106	06.05.2001	jordgubbe	Ikke satt	tyson-farmer	Ikke satt	Spørsmål
102	05.05.2001	ha	Ikke satt	tyson-farmer	Ikke satt	henvendelse
101	05.05.2001	dff	Ikke satt	tyson-farmer	Ikke satt	Spørsmål
100	05.05.2001	Dette er en ny sak	lav	tyson-farmer	Ferdig	henvendelse
99	04.05.2001	Hvorfor går det sånn	middels	bert	behandles	Klage
98	04.05.2001	heisann	nedprioritert	bert	Ikke satt	henvendelse
97	04.05.2001	Heisann	Ikke satt	bert	Ikke satt	Spørsmål
96	04.05.2001	ja, det loka	middels	tyson-farmer	behandles	henvendelse
95	04.05.2001	isappers	lav	tyson-farmer	behandles	henvendelse
94	04.05.2001	dff	høy	tyson-farmer	behandles	henvendelse
93	04.05.2001	dff	høy	tyson-farmer	behandles	Klage
92	04.05.2001	df	høy	tyson-farmer	Ferdig	henvendelse
91	04.05.2001	df	høy	tyson-farmer	behandles	Klage
90	04.05.2001	df	Ikke satt	tyson-farmer	Ikke satt	Klage

Figur 0-2

## 2.6 Navigasjonsstruktur



Figur 0-3

## 2.7 Filbeskrivelse

### Fellesfiler:

#### Login.cfm:

For å få tilgang til systemet må man fylle ut brukernavnet og passord i formene som man registrerte seg med på owner/customer registreringen. Alle passord og brukernavn er unike. Hvis man logger seg på men feil passord eller brukernavn vil man få beskjed om at det er feil, og man kan prøve å logge seg på nytt. Det er samme login sider for owner og customer.

Designet for login sidene er dynamisk, slik at hver enkelt owner kan velge tekst og bilde.

Login siden navigerer deg inn på riktig side(owner eller customer).

På login sidene blir det kjørt mange spørringer mot databasen som henter ut info om customer/owner og legger dette inn i arrays som legges i session.

#### Customer\_reg.cfm

Når en customer av en owner får innbydelse av owneren får den en link som går til login.cfm med.

Owner\_id blir da sendt med i URL og customer går inn på customer\_reg.cfm med denne owner\_id'en og registrerer seg som bruker av systemet til owner. Tekstfeltene som har en stjerne ved siden av seg er obligatoriske felter i databasen og må skrives inn. Om disse feltene ikke er fylt inn så får brukeren beskjed om hvem felter som ikke er fylt ut og må skrive disse inn for å få lagret.

#### Owner\_reg.cfm:

For å kunne ta systemet i bruk må man registrere seg som owner. Her må opplysninger om brukeren(firmaet) av systemet fylles ut. Må også fylle ut administrator opplysninger(navn, avdeling, brukernavn og passord) for administrator av systemet. Administrator vil få brukerrettigheter på høyeste nivå.

Alle feltene(unntatt ett: fax) er obligatoriske, og man vil få feilmelding hvis man ikke fyller ut disse, eller om dataene ikke er riktig(validering). Man vil også få beskjed hvis passord eller brukernavn finnes fra før. Denne sjekken vil skje når man trykker på submit knappen.

### Customerdel

#### Customer.cfm

Her får customer opp alle sakene sine med dato/klokkeslett, sak og status. Den siste saken som er customer har registrert kommer øverst. Når kunden har fått svar på en sak står det NY ved den saken.

#### Customer\_ticket.cfm

På denne siden registreres saken. Dropdownbox'ene med de forskjellige valg customer velger fra på denne siden er bestemt av owner. Owner setter altså opp hvilke valg customer har. Overskrift er obligatorisk felt i databasen og må være med. Customer får beskjed om å fylle ut feltet hvis det ikke er fylt ut.

#### Customer\_update.cfm

Her kan customer endre profilen sin. Opplysningene om customer kommer. Feltene med \* må være fylt ut, ellers får customer beskjed om å fylle dem inn.

## Ownerdel:

Beskrivelse av sidene:

Owner.cfm:

Denne siden består av flere moduler. Her vil man få opp sakene som hver enkelt avdeling har mottatt eller nye saker som skal behandles.

Nye saker:

Får opp de nye sakene som har blitt sendt til bedriften. Her får man opplysninger om når de er sendt, overskrift på saken, hvem det er fra og hvilken type sak det er. Hvis man vil ha fler opplysninger om den enkelte sak må man trykke på linken til saken.

Avdelinger:

Avdelingene er angitt av owneren. Antall avdelinger avhenger av hvor mange som har blitt satt opp av owneren.

Her kommer sakene som man har satt avdeling på. Har de samme mulighetene her som på nye saker.

Update\_empl.cfm:

På denne siden kan man oppdatere opplysninger som man har registrert om en ansatt. Denne modulen kommer opp uansett hvilke brukerrettigheter man har. Man sjekker om nødvendige data er med og om de er av riktig format. Nye data blir lagt til databasen.

Owner\_opt\_department.cfm:

På denne siden kan man legge til avdelinger som finnes i bedriften, slette dem eller endre dem. Får vist hvilke avdelinger som finnes. Hvis man skal legge til en ny avdeling så får man opp en ny side hvor man skal fylle inn avdelingsnavn og beskrivelse. Får opp samme side ved endring av avdeling, slik at man kan endre avdelingsnavnet.

Owner\_opt\_priority.cfm:

På denne siden kan man legge til prioritet som kan settes på saker, slette dem eller endre dem. Får vist hvilke prioriteter som finnes. Det er lagret 3 defaulte prioriteter( høy, middels og lav). Hvis man skal legge til en ny prioritet så får man opp en ny side hvor man skal fylle inn prioritets navn og beskrivelse. Får opp samme side ved endring av prioritet, slik at man kan endre prioritetsvalget.

Owner\_opt\_tickettype.cfm:

På denne siden kan man legge til meldingstyper, slette dem eller endre dem. Får vist hvilke meldingstyper som finnes. Det er lagret 2 defaulte meldingstyper( spørsmål, klage).Hvis man skal legge til en ny meldingstype så får man opp en ny side hvor man skal fylle inn meldingstype og beskrivelse. Får opp samme side ved endring av meldingstype, slik at man kan endre på meldingstypen.

Owner\_opt\_tickstat.cfm:

På denne siden kan man legge til statuser på saker, slette dem eller endre dem. Får vist hvilke statuser som finnes. Det er lagret 2 defaulte statuser( behandles, ferdig). Hvis man skal legge til en ny status så får man opp en ny side hvor man skal fylle inn status navn og beskrivelse. Får opp samme side ved endring av status, slik at man kan endre statusvalget.

Owner\_opt\_custdef1.cfm, Owner\_opt\_custdef2.cfm, Owner\_opt\_custdef3.cfm:

Her finnes det radiobuttons som ber owneren velge om feltene skal være aktive eller ikke(JA/NEI). Hvis man vil ha alternativer å velge mellom skrives det inn i formen på siden og trykker submit

button(Legg til). Alle alternativene listes opp under, annen hver med grå og hvit bakgrunn. Hvis man vil slette/endre disse alternativene må man trykke på link til dette. Ved endre får man opp alternativet opp i formen, endrer og trykker på submit knappen(Endre).

Hvis man vil slette ett alternativ får man opp en advarsel, hvis alternativet er valgt i en melding, om at feltet finnes i en melding og at valget vil bli borte i denne meldingen.

Owner\_opt.cfm

På denne siden kan man velge hvordan layouten på siden skal se ut. Man kan velge en logo som skal vises både på innloggings siden, customer siden og på owner siden. Kan bruke browse knapp for å finne ønsket bilde. Bildet blir lagret i en opplad katalog med unikt navn. Kan merke radio button på enten bevare bilde eller nytt bilde hvis ønskelig. Kan sette font type, bakgrunnsfarge og tekst til åpningssiden ved å angi dette i formene. Trykker submit knappen(lagre) for å lagre.

Owner\_update.cfm

Her kan owneren oppdatere profilen på firmaet sitt. Opplysningen om firmaet kommer opp i formen(Firmanavn, adresse, postnr, poststed, land, telefonnr, fax, email), så det er bare å endre disse og trykke på oppdater. Blir sjekket om data er av riktig format og at det ikke er blanke felt der data er obligatoriske.

Employee\_list.cfm

Her vises alle ansatte i firmaet i en liste med opplysninger om hver enkelt(Navn, adresse, email, avdeling, telefon, mobil, nivå, postnr, poststed, brukernavn, passord. Man kan trykke på link om man vil endre, slette ansatt. Er også link til registrere ansatt.

Employee\_reg.cfm

På denne siden kan man registrere ansatte ved bedriften. Skriver inn opplysninger om ansatte(Navn, adresse, email, postnr, poststed, telefon, mobil, brukernavn, passord, brukernivå(selectboks), avdelingsnivå(selectboks)), og deretter trykke på submitknappen(lagre). Sjekker om data er riktig og om at data er fylt ut.

Owner\_invite.cfm

For å sende invitasjon til kunder fyller man ut denne siden. Fyller ut email og Navn til kunden, deretter trykke send. Dermed blir det sendt en email til kunden med invitasjonen som vises på owner\_invite siden.

Owner\_ticket.cfm

Her vises meldingen. Øverst vises info om kunden(Navn, adresse, poststed, email, telefon, mobil, fax og kontaktperson). Deretter vises info om meldingen(Referansenr, type, avdeling, prioritet, status). Nedenfor vises meldingene med info(meldingsdato, tekst, forfatter). Nederst kan det legges til en kommentar til saken i textarea feltet.

Owner\_customer\_list.cfm

Her får man en oversikt over alle customerne til owneren. Det kommer opp en liste over hver kunde med følgende data: kunde nr, kundenavn, email adressen, kontaktpersonen, link til info side om cusotmer og link til meldingene som den spesifikke customer har sendt til owneren.

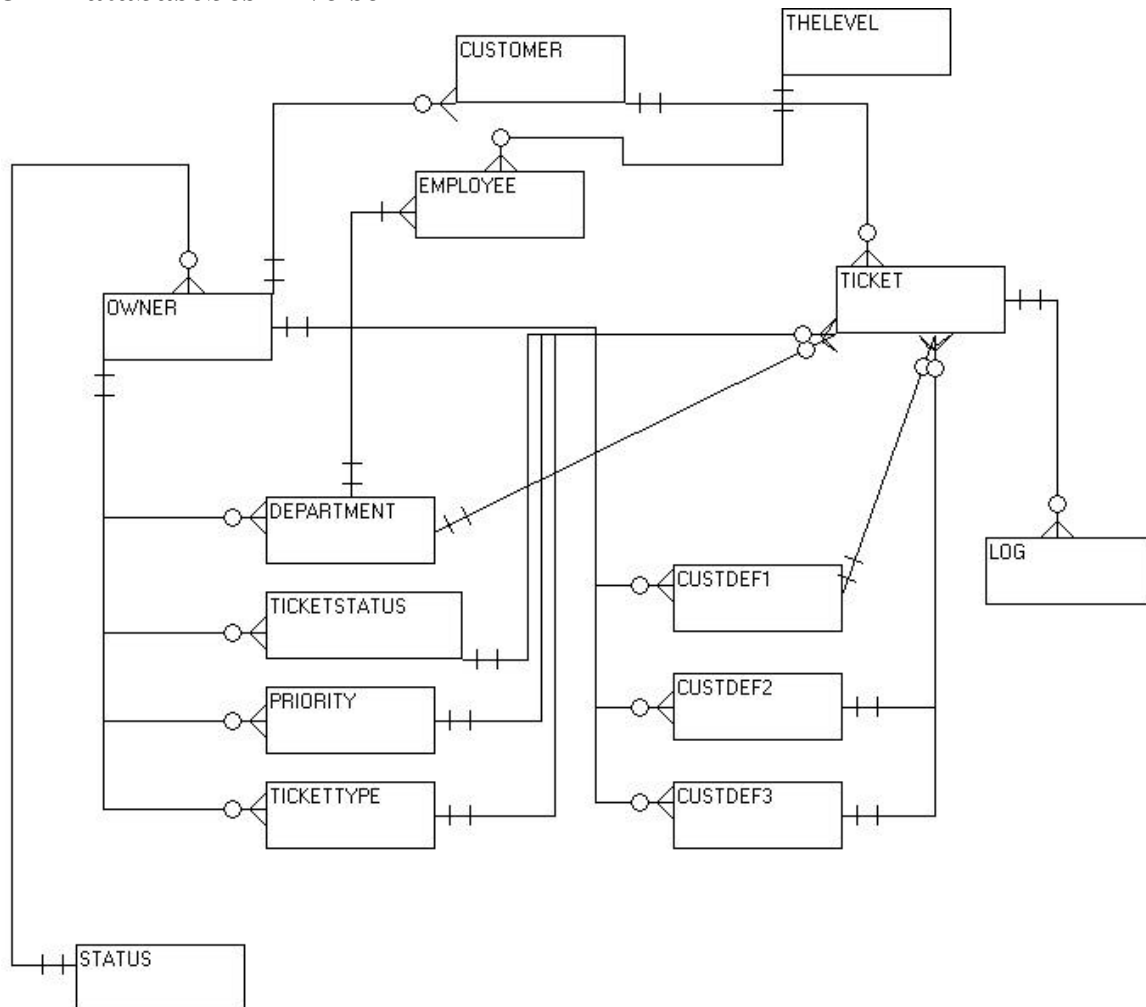
Owner\_customer\_info.cfm

Her listes det ut data om en customer. Kundenr, kundenavn, adresse, post nr, poststed, land, info, reg. Dato, telefon, mobil, email, fax og kontaktperson.

Owner\_search.cfm

På denne siden kan man søke etter meldinger. Man kan angi alle mulige kriterier for søk, søk på meldinger(avdeling, overskrift, prioritet), søk på dato, søk på kunde(navn, postnr, tlf) og man kan angi om det skal være aktive/ferdig meldinger eller alle meldinger. Noen av søkefeltene er select bokser som man kan bla gjennom får å finne ønsket søkekritere.

### 3 Databasebeskrivelse



**Figur 3-1**

Tar først for oss relasjonene mellom entitetstypene:

En OWNER har mange CUSTOMERS

Hver OWNER i systemer kan ha mange CUSTOMERS. OWNER er de som betaler for systemet. Vi har valgt at hver CUSTOMER kun har et brukernavn og passord. Dette betyr at alle som bruker kundebehandlingssystemet hos CUSTOMER må logge seg på med samme brukernavn og passord. Hver CUSTOMER kan kun ha en OWNER. Hvis en CUSTOMER skal bruke systemet mot flere OWNERS må han registrere seg flere ganger i systemet.

En CUSTOMER har flere TICKETS

En TICKET er en sak. Hver CUSTOMER kan ha mange saker mot OWNER.

En TICKET har Mange LOG'er

En LOGG er en melding under en sak. Både OWNER og CUSTOMER kan sende inn slike LOG'er.

En OWNER har mange DEPARTMENT'S





DEPARTMENT er en avdeling i firmaet til OWNER. F. eks Data, admin og slikt.

En DEPARTMENT har flere EMPLOYERS

En EMPLOYER er en ansatt i en avdeling. Grunnen til at vi har satt relasjonene på denne måten er at når man logger seg på så skal man få opp sakene som gjelder sin avdeling.

Et THELEVEL har flere EMPLOYERS

Alle EMPLOYERS har et LEVEL. Level betyr brukerrettighetene EMPLOYER har i systemet.

En STATUS har flere OWNERS

Status for en OWNER er om i hvilken grad OWNER er aktiv/passiv i systemet. Hvis en OWNER slutter å bruke systemet blir STATUS satt til passiv.

De neste entitetstypene er entitetstyper som blir satt av OWNER for å bli brukt som attributter under TICKET.

En OWNER har flere DEPARTMENT som har flere TICKET

Hver avdeling kan ha flere Tickets og OWNER setter opp hvilke DEPARTMENTS som skal ticketen skal gjelde.

En OWNER har flere TICKETSTATUS'er som har flere TICKET's.

Hver ticket har en ticketstatus som blir satt av OWNER. Ticketstatus kan typisk være "under bestilling" eller "lest". Hvilke ticketstatuser som skal kunne benyttes settes av OWNER

En OWNER har flere PRIORITY's som har flere TICKETS.

Owner setter prioritet på en TICKET. Denne prioriteten er intern hos OWNER og CUSTOMER skal ikke se denne.

En OWNER har flere TICKETTYPE's som har flere TIKCKET's

En OWNER definerer tickettyper. CUSTOMER setter tickettype på ticketen. Et eksempel på tickettype kan være "KLAGE på etc"

CUSTDEF:

OWNER definerer CUSTUM felter. Som skal være med på en ticket. CUSTOMER setter verdien på disse på ticketen.



Her er alle Entitetstypenes attributter:

OWNER
<u>OWNER_ID</u> OWNER_NAME *STATUS_ID OWNER_ADDRESS OWNER_PHONE OWNER_EMAIL OWNER_FAX OWNER_COUNTRY OWNER_REG_DATE OWNER_LAST_F_DATE OWNER_CUST_DEF1_VALUE OWNER_CUST_DEF2_VALUE OWNER_CUST_DEF3_VALUE OWNER_IMAGEPATH OWNER_BG_COLOR OWNER_TEXT OWNER_FONT owner_ZIPCODE owner_ZIPPLACE

CUSTOMER
<u>CUSTOMER_ID</u> CUSTOMER_NAME *OWNER_ID CUSTOMER_ADDRESS CUSTOMER_EMAIL CUSTOMER_FAX CUSTOMER_PHONE1 CUSTOMER_PHONE2 CUSTOMER_CONTACT_NAME CUSTOMER_USERNAME CUSTOMER_PASSWD CUSTOMER_REG_DATE CUSTOMER_DESCRIPTION CUSTOMER_COUNTRY customer_zipcode customer_ZIPPLACE

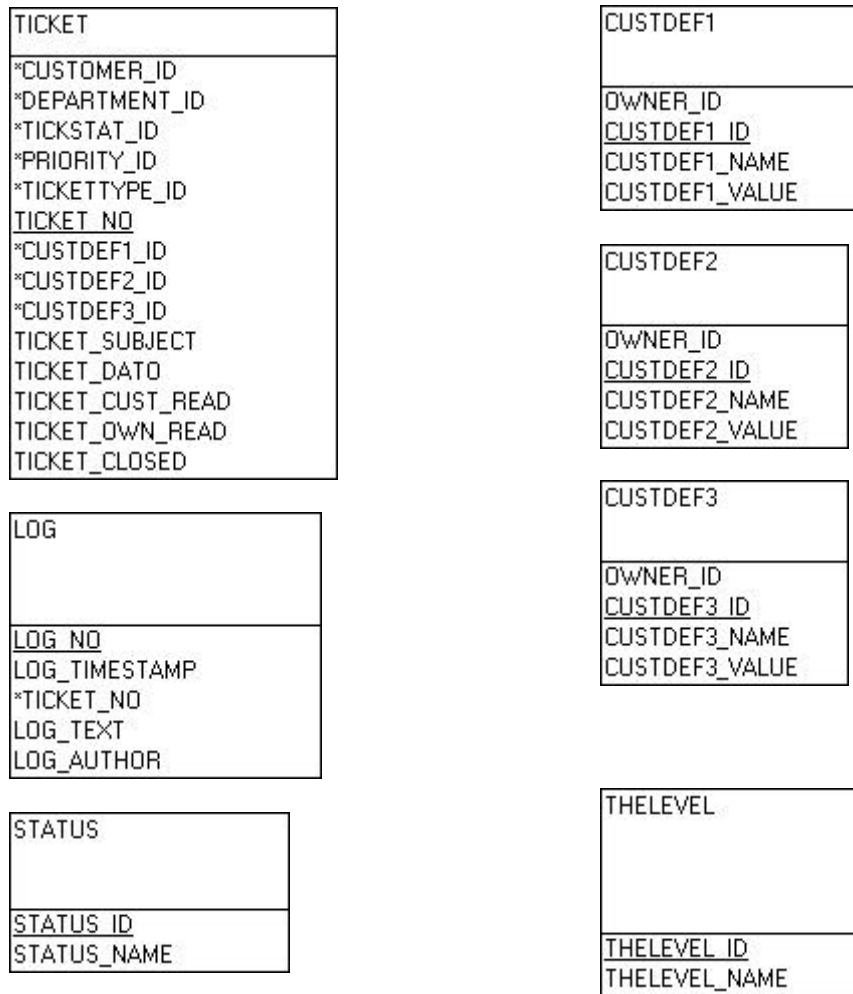
EMPLOYEE
<u>EMPL_ID</u> EMPL_NAME EMPL_ADDRESS EMPL_EMAIL *DEPARTMENT_ID EMPL_CELLPHONE *THELEVEL_ID EMPL_PHONE EMPL_USERNAME EMPL_PASSWD empl_zipcode empl_zipplace

DEPARTMENT
<u>DEPARTMENT_ID</u> DEPARTMENT_NAME *OWNER_ID DEPARTMENT_DESCRIPTION

TICKETSTATUS
<u>TICKSTAT_ID</u> TICKSTAT_NAME *OWNER_ID TICKSTAT_DESCRIPTION

PRIORITY
<u>PRIORITY_ID</u> PRIORITY_NAME *OWNER_ID PRIORITY_DESCRIPTION

TICKETTYPE
<u>TICKETTYPE_ID</u> TICKETTYPE_NAME *OWNER_ID TICKETTYPE_DESCRIPTION



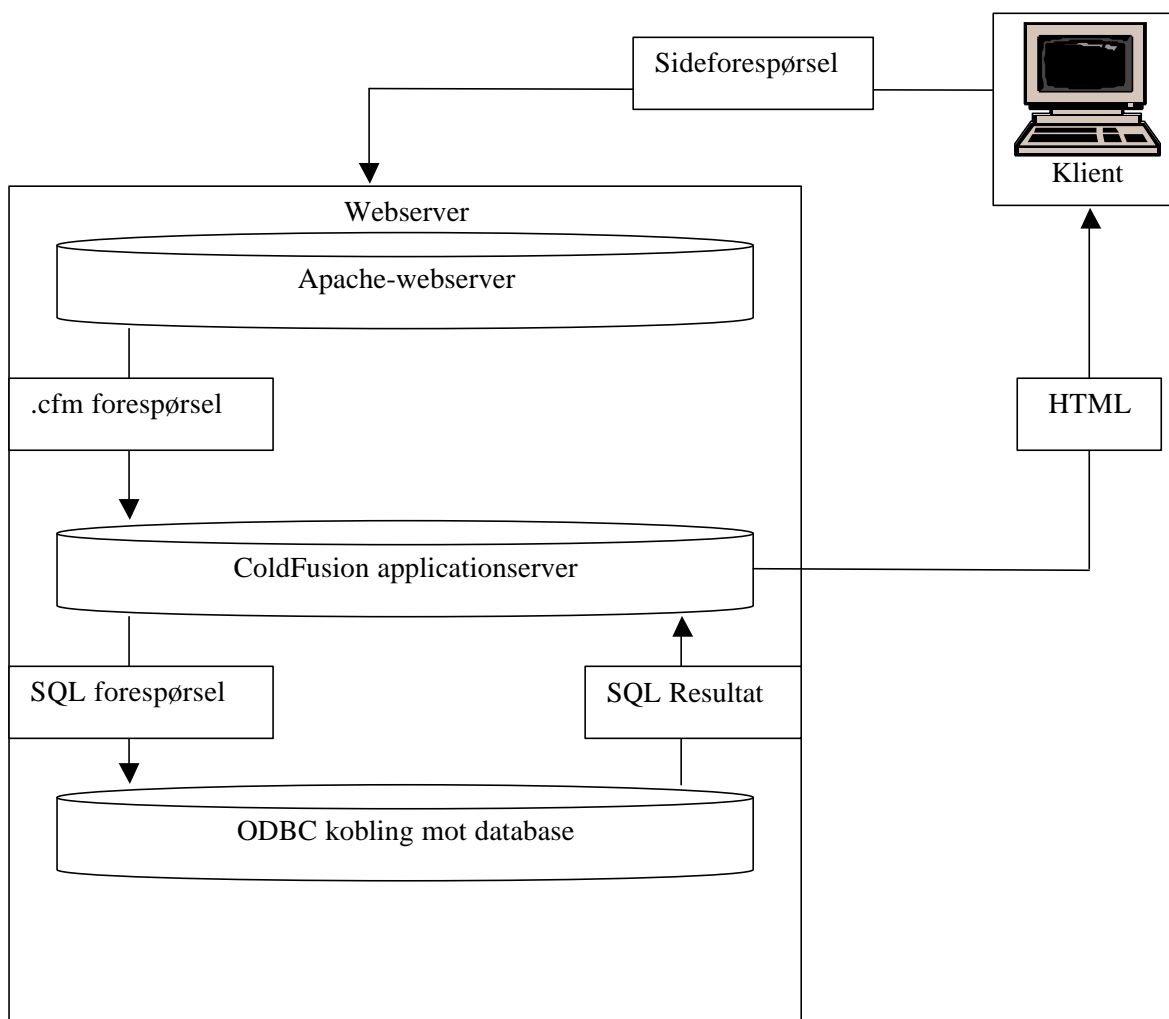
Figur 3-2

## 4 Systemarkitektur

Når vi bruker ColdFusion kan vi bruke forskjellige typer arkitekturer. Det er med ColdFusion mulighet for å bruke EnterpriseJavaBeans eller lignende som et ekstra lag mellom hovedlaget og lagringslaget. Dette vil redusere databaseaktiviteten ved at man får alt innholdet i databasene tilgjengelig i objekter, i stedet for at man må kjøre databaseforespørsler. Dette vil da bli en firelags arkitektur.

Siden vi har svært lite kunnskap og erfaring om en slik fire lags arkitektur, har vi valgt å bruke en tre lags arkitektur:

1. Presentasjonslaget - Grensesnitt som blir vist i browseren
2. Hovedlaget - Selve applikasjonen på serveren
3. Lagringslaget - Databasen

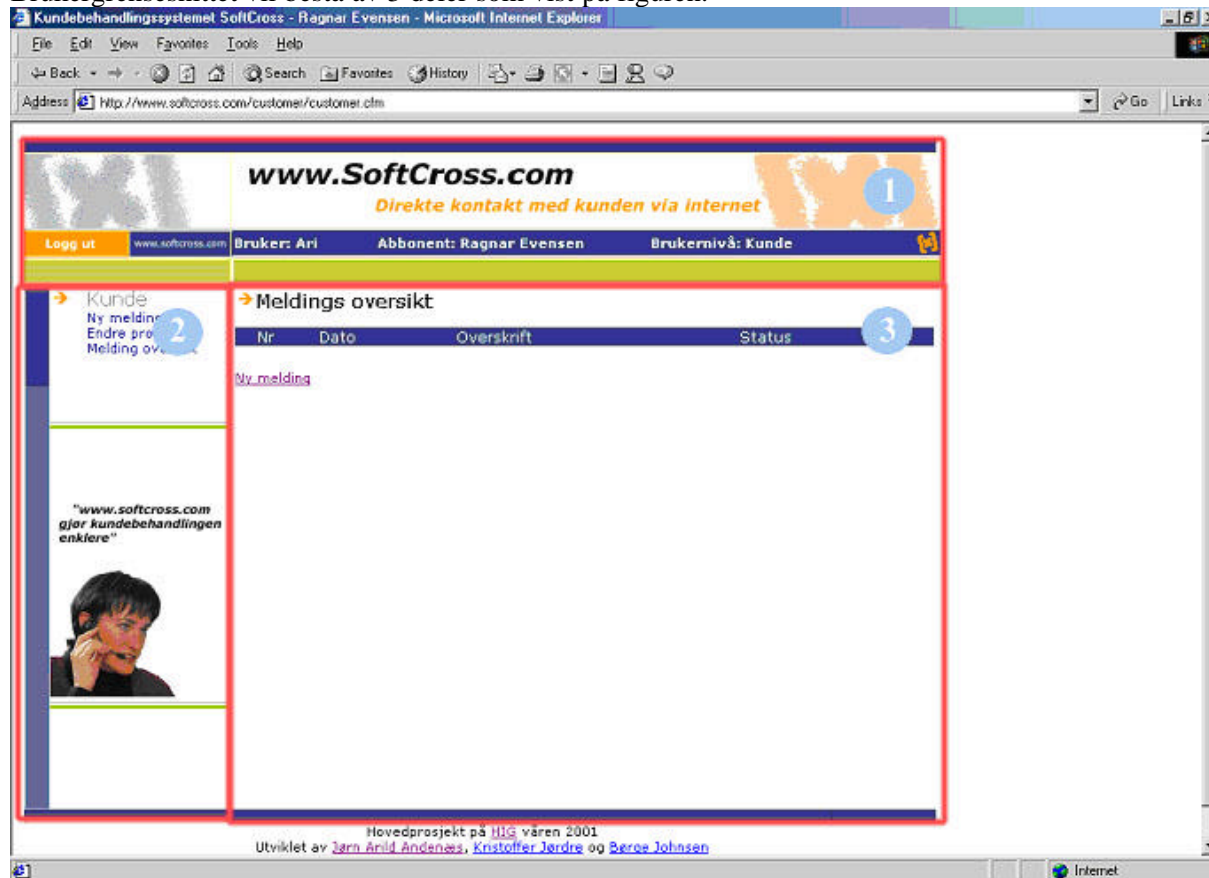


Figur 4-1 Systemarkitektur

## 5 Brukergrensesnitt

### 5.1 Beskrivelse av brukergrensesnitt

Brukergrensesnittet vil bestå av 3 deler som vist på figuren.



Figur 5-1

### Del 1: Info del

I denne delen av skjermbildet vil det stå info om hvem som er logget inn, hvilken abonnent(owner) man er logget inn under, og hvilket brukernivå man er logget inn på. Hvis abonnenten har satt opp at logoen deres skal vises i systemet så vil den vises her. "Logg ut" knappen er også lagt i denne delen.

### Del 2: Meny

Denne delen består av menyen til systemet. Menyene vil variere etter hvilket brukernivå man er logget inn som. Hvilket nivå man er logget inn som vil alltid stå som overskrift i menyen. Når man er logget inn vil man på alle sidene få se denne menyen. Når det klikkes på et valg i menyen blir man sendt til en ny side. Det ser for brukeren som om det kun er del 3 som lastes på nytt.

### Del 3: Hoveddelen

Dette er hoveddelen av skjermbildet og det er denne delen som endrer seg når man klikker seg inn på en ny side. For at brukergrensesnittet skal bli best mulig, vil vi at denne delen skal ha en fast bredde. Det skal alltid i denne delen stå en overskrift som forteller brukeren hvilken side som vises.



#### Ny melding

Dette symbolet betyr det at det er en ny melding enten er ny eller at det har kommet en ny melding fra kunden på denne meldingen



#### Melding lest

Dette symbolet betyr at meldingen er lest.



#### Melding avsluttet

Dette symbolet betyr at meldingen er avsluttet, det går da ikke an å skrive noe mer på den.

### *GUI Standard*

#### Distribusjon av nye utgaver

Siden gruppen jobber med en demo versjon av systemet vil det etter hvert bli lagt ut forbedret versjoner av systemet. Denne prosessen er ikke gruppen involvert i, den bestemmes av oppdragsgiveren.

#### Bruk av mal objekter

Det skal brukes en del standard skjemaer som kan lett kopieres. Registreringsskjemaene inneholder stort sett de samme formene. Egendefinertfelt 1,2,3 er tre omtrent like skjemaer som kan kopieres, og prioritet, status, type, avdelinger er skjemaer som har lik funksjonalitet og design.

#### Plassering , bredde og avstand

Bredde på tekstbokser skal ha en standard på 30 bokstavbredder. Enkelte felter skal avvike fra standarden hvis de er naturlig kortere (tlf, postnr. Etc..). Alle felter og tekstbokser skal plasseres inn i tabeller. Tabellene har default "lufting" mellom seg, og skal derfor ikke defineres. Se Figur 5-2.

Firmanavn	<input type="text"/>	*
Adresse:	<input type="text"/>	*
Postnr:	<input type="text"/>	*

Figur 5-2 Eksempel på form

### Knapper

Vi skal bruke følgende standardknapper:

Lagre

Avbryt

Endre

Søk

Send

Logg inn

Bruker standard størrelse som blir bestemt av teksten i knappene. Se Figur 5-3.



Figur 5-3 Knapper

### Ledertekster(overskrifter)

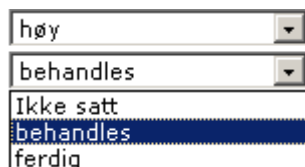
Ledertekster skal være venstrejustert, være skrevet med størrelse 11 pt og ha en oransje pil foran seg. Hvis det er en naturlig tilleggsoverskrift, skal denne skrives i samme størrelse, men i kursiv. Se Figur 5-4.

### ➔ Avdeling: *Admin*

Figur 5-4 Eksempel på overskrift

### Selectbokser

Størrelsen på disse avhenger av innputen som finnes i databasen. Mengde avhenger også av input fra owneren. All input i disse boksene skal være sortert alfabetisk. Selectboksene blir forskjellige i Netscape og Explorer se Figur 5-5 og Figur 5-6.



Figur 5-5 I explorer setter en fast bredde på alle selectbokser hvis de er forhåndsdefinert i stylesheets.



Figur 5-6 I Netscape blir selcboksene like brede som teksten inni er lang.

**Skrift:**

All vanlig skrift skal være størrelse 10px og type med standard font Verdana. Owner kan selv velge font men anbefalt er Verdana. I typiske tabeller med mange rader brukes skriftstørrelse 8px. Det skal bare være stor bokstav i første bokstav i første ord. Skriften vil variere i systemet siden owner kan velge type selv.

*Farger*

Element	Farge
Vindusbakgrunn	Hvit standard, kan på noen sider velges av owner
Knapper	Lys grå, sort tekst
Redigerbare felter Listebokser Rullegardinlister Avhkningsbokser	Hvit, sort tekst
Ledetekster	Mørk grå
Angivelse av obligatoriske felter	Merkes ved svart stjerne
Tabelloverskrifter	Hvit tekst med blå bakgrunn
Tabeller	Sort tekst med annen hver linje hvit og lysgrå.

**Tabell 5-1**

## 6 Testing

### 6.1 Beskrivelse av testfasen

Da vi planla prosjektet bestemte vi oss for å ha en iterasjon i systemutviklingsprosessen vår. Vi bestemte oss for å foreta en testfase og deretter videreutvikle systemet vårt. Dette for å få systemet vårt til å bli best mulig. Gruppen har ikke hatt erfaringer med slike kundebehandlingssystemer før, og ved hjelp av en testfase vil gruppen få tilbakemeldinger som forhåpentligvis vil være svært viktige for det endelige systemet.

For å utnytte testfasen best mulig må vi kontakte flest mulig personer både med erfaring fra slike systemer som kan teste på owner siden, og personer som ikke har erfaring som kan teste på customer siden. Det vanskeligste er å få tak i personer med erfaringer siden gruppen kjenner få personer med denne bakgrunnen. Oppdragsgiver har erfaring, og gruppen håper på at han kan være behjelpelig med å skaffe oss noen testbrukere på owner siden.

Vi tror det ikke vil være noe vanskelig å skaffe brukere til å teste på customer siden, siden det finnes ganske mange elever på HIG som har sagt seg villige til å bli med i en testfase. Gruppen kjenner også folk utenfor skolen som sannsynligvis blir med å teste.

Den kritiske fasen er å få ferdig systemet slik at testingen vil ha noe for seg. Det er ikke nødvendig at alt skal fungere smertefritt, men det er viktig at systemet fungerer slik at man får testet. Designet på sidene vil bli nedprioritert i forhold til funksjonaliteten, men hvis gruppen får tid vil vi gjøre designet så bra som mulig.

Det er også veldig viktig at vi lager bra spørreskjemaer som vi kan sende til testpersonene. Viktig å spørre riktige spørsmål og at det skal være lett å fylle ut, slik at vi får tilbake skjemaene med nyttige kommentarer. Spørreskjemaene blir i Word format siden dette formatet mest universalt. Kan også lage et spørreskjema på nettet, men dette kan ta alt for lang tid å lage og evaluere.

Gruppen må også avtale ett møte med oppdragsgiver etter testfasen, slik at vi kan evaluere skjemaene og finne hvilke feil som er nødvendig å rette opp, og hvilke ekstra funksjonaliteter vi må legge til systemet.

### 6.2 Testfasen punkt for punkt

Ta kontakt med testbrukere og avtale tid med dem for testing

Utarbeideevalueringsskjema et for owner og et for customer

Teste sammen med testbrukerne. Der testbrukeren er customer må vi sitte på ownersiden og svare.

Omvendt der testbrukeren er owner.

Gjennomgå alle evalueringsskjemaene, og notere ned alle kommentarer.

Møte med oppdragsgiver hvor vi går gjennom alle kommentarene punkt for punkt. Setter så opp en prioritert rekkefølge på de ønskede endringene.

Sette opp et skjema hvor alle endringene står i prioritert rekkefølge, hvem som skal utføre endringene og et signaturfelt for utført endring.

Utføre endringene og fylle ut endringskjema.





**Vedlegg  
B  
Evalueringsskjemaer**



## Evaluerings-skjema etter test av Softcross - Owner

Du har blitt plukket ut som en privilegert tester av hovedprosjektet Softcross. Softcross er et Hovedprosjekt på Høgskolen i Gjøvik.

Takk for at du sa ja til dette!

### Oppgavebeskrivelsen til Softcross:

Oppgaven går ut på å utvikle en prototype av et system som skal skape bedre kommunikasjon mellom bedrift og kunde når kunden tar kontakt med bedriften anngående en spesiell sak. Kundene skal kunne følge status til den aktuelle saken via Softcross, som skal ligge sentralisert på en webserver. Modellen baserer seg på ulike brukernivå(kunde, bedrift, adm) som ett B2B2C system. Prototypen skal bli testet opp mot et begrenset antall brukere for så å bli modifisert etter brukernes ønsker. Dette systemet skal kunne brukes av ulike bedrifter, som en kontaktforbindelse mellom bedriften og kunden(ene). Her skal kunden kunne følge opp status på en sak den har hos bedriften, og kunne kommunisere med bedriften under behandling av denne saken. Systemet skal ha en mulighet for å tilpasses hver enkelt bedrift og skal kunne brukes av bedrifter med ulike behov. Hver bedrift skal selv kunne administrere enkelte deler av systemet via web. Systemet skal være enkelt å bruke, så enkelt at de ansatte hos bedriftene ikke har behov for noe særlig opplæring. Oppdragsgiver(hovedadministrator) skal kunne administrere systemet fra web, f.eks opprette nye brukere(bedrifter), og fjerne kunder fra systemet. På grunn av prosjektets tidsrammer vil fokus være på funksjonaliteten, og det grafiskebrukergrensesnittet vil derfor ikke være prioritert.

**Oppdragsgiver:** Arne Sunde

**Veileder:** Tom Røyse

**Er det noe du lurer på med systemet så kontakt oss på:**

Mail: [borge@johnsen.com](mailto:borge@johnsen.com) (Børge)

Tlf: 93021930 (Jørn Arild)



Hvilken rolle har du som bruker

Systemet Softcross kan benyttes av mange bedrifter til kundebehandling. Du er en simulert bedrift som skal bruke systemet mot dine kunder. Vi vil da registrere oss som dine kunder og sende deg saker. Du kan da følge opp saken.

Hva skal du gjøre:

Gå inn på <http://softcross.com/index.cfm>

- Registrer ditt firma som bruker av systemet (trykk: opprett ny kunde)
- Følg så guiden videre for å sette opp systemet slik DU vil.
- Registrer en ansatt og logg inn med denne. Legg merke til at du har forskjellige muligheter om du er Administrator, Mottaker eller bruker.
- Send oss invitasjon til ditt system (bruk [joernaa@online.no](mailto:joernaa@online.no) som mailadresse)
  
- Så sender vi deg en sak
- Så svarer du oss og følger opp saken til den er avsluttet
  
- Fyll deretter ut dette skjemaet.

## Registrer deg i system:

Generelt

Veldig bra

Dårlig

5

4

3

2

1

--	--	--	--	--

Spesifikt:

Hva synes du om designet på denne siden?

*Skriv litt om det her:*

--

Er det noe funksjonalitet som mangler eller som du ikke skjønner?

*Skriv litt om det her:*

--

## Om sidene

Var de lette å forstå?

Generelt

Veldig bra

Dårlig

5

4

3

2

1

--	--	--	--	--

Hva synes du om navigeringen på sidene – var den god?

Generelt

Veldig bra

Dårlig

5

4

3

2

1

--	--	--	--	--

Hva med navnebruk på funksjoner – var disse enkle å forstå?

Generelt

Veldig bra

Dårlig

5

4

3

2

1

--	--	--	--	--



Har du brukt slike systemer før? JA/NEI

Hvis JA

Hvordan syntes du dette systemet fungerte i forhold til lignende systemer du har prøvd?

Generelt

Veldig bra

Dårlig

5

4

3

2

1

--	--	--	--	--

## Hva synes du burde forbedres med systemet:

Brukergrensesnittmessig:

*Skriv litt om det her:*

Navigeringsmessig:

*Skriv litt om det her:*

Funksjonalitetsmessig:

*Skriv litt om det her:*

Nevn 5 positive ting med systemet:

- 1.
- 2.
- 3.
- 4.
- 5.

Nevn 5 negative ting med systemet:

- 1.
- 2.
- 3.
- 4.
- 5.



Tusen takk for innsatsen!!!!

Du har vært til STOR hjelp!

Kristoffer Jørdre  
Jørn Arild Andenæs  
Børge Johnsen



Evaluerings-skjema etter test av Softcross - Customer

Du har blitt plukket ut som en privilegert tester av hovedprosjektet Softcross. Softcross er et Hovedprosjekt på Høgskolen i Gjøvik.

Takk for at du sa ja til dette!

Oppgavebeskrivelsen til Softcross:

Oppgaven går ut på å utvikle en prototype av et system som skal skape bedre kommunikasjon mellom bedrift og kunde når kunden tar kontakt med bedriften anngående en spesiell sak. Kundene skal kunne følge status til den aktuelle saken via Softcross, som skal ligge sentralisert på en webserver. Modellen baserer seg på ulike brukernivå(kunde, bedrift, adm) som ett B2B2C system. Prototypen skal bli testet opp mot et begrenset antall brukere for så å bli modifisert etter brukernes ønsker. Dette systemet skal kunne brukes av ulike bedrifter, som en kontaktforbindelse mellom bedriften og kunden(ene). Her skal kunden kunne følge opp status på en sak den har hos bedriften, og kunne kommunisere med bedriften under behandling av denne saken. Systemet skal ha en mulighet for å tilpasses hver enkelt bedrift og skal kunne brukes av bedrifter med ulike behov. Hver bedrift skal selv kunne administrere enkelte deler av systemet via web. Systemet skal være enkelt å bruke, så enkelt at de ansatte hos bedriftene ikke har behov for noe særlig opplæring. Oppdragsgiver(hovedadministrator) skal kunne administrere systemet fra web, f.eks opprette nye brukere(bedrifter), og fjerne kunder fra systemet. På grunn av prosjektets tidsrammer vil fokus være på funksjonaliteten, og det grafiskebrukergrensesnittet vil derfor ikke være prioritert.

**Oppdragsgiver:** Arne Sunde

**Veileder:** Tom Røyse

**Er det noe du lurer på med systemet så kontakt oss på:**

Mail: [borge@johnsen.com](mailto:borge@johnsen.com) (Børge)

Tlf: 93021930 (Jørn Arild)

## Hvilken rolle har du som bruker

Systemet Softcross kan benyttes av mange bedrifter til kundebehandling. Du som kunde vil da være kunde av en slik bedrift. Bedriften i dette tilfellet er hovedprosjektgruppa selv. Du vil derfor kunne sende oss meldinger gjennom vårt eget system. En del design instillinger er gjort av oss på sidene, som farger og skrifttyper, logo osv. Dette kan hver enkelt bedrift sette opp selv. Systemet ville derfor sett annerledes ut hvis du hadde vært registrert som kunde av en annen bedrift.

Du skal altså benytte systemet til å sende klager og forespørsler til bedriften du er kunde hos. Så send oss gjerne en klage ☺

Hva skal du gjøre:

I dette tenkte tilfellet er du kunde hos **webhouse.no**. Her hoster du ditt firmas hjemmeside, i denne forbindelse har det oppstått noen problemer.



**Webhouse.no** er bruker av "SoftCross kundebehandlingssystem" og vil at du som kunde hos dem, skal bruke dette systemet for klager og spørsmål.

Du mottar nå en invitasjon, fra webhouse.no på email. Gå inn på denne adressen og gjør følgende:

-Registrer deg som bruker

- Trykk på "Ny melding", skriv så inn en melding og trykk send
- Trykk så "Endre profil", og endr noe av dataen din.
- Trykk så på "Melding oversikt", og se om du har fått noe svar på din melding. Send nå gjerne inn flere meldinger eller skriv noe tillegg på noen av de gamle meldingane.
- Gjør dette flere ganger.

-Fyll deretter ut dette skjemaet.

## Registrer deg som ny kunde:

Generelt

Veldig bra

Dårlig

5

4

3

2

1

--	--	--	--	--

Spesifikt:

Hva synes du om designet på denne siden?

*Skriv litt om det her:*

--

Er det noe funksjonalitet som mangler eller som du ikke skjønner?

*Skriv litt om det her:*

--



## Innloggingssiden:

Generelt

Veldig bra

Dårlig

5

4

3

2

1

--	--	--	--	--

Spesifikt:

Hva synes du om designet på denne siden?

*Skriv litt om det her:*

--

Er det noe funksjonalitet som mangler eller som du ikke skjønner?

*Skriv litt om det her:*

--

## Første side etter innlogging:

Generelt

Veldig bra

Dårlig

5

4

3

2

1

--	--	--	--	--

Spesifikt:

Hva synes du om designet på denne siden?

*Skriv litt om det her:*

--

Er det noe funksjonalitet som mangler eller som du ikke skjønner?

*Skriv litt om det her:*

--



--

## Ny melding:

Generelt

Veldig bra

5

4

3

2

Dårlig

1

--	--	--	--	--

Spesifikt:

Hva synes du om designet på denne siden?

*Skriv litt om det her:*

--

Er det noe funksjonalitet som mangler eller som du ikke skjønner?

*Skriv litt om det her:*

--

## Endre profil:

Generelt

Veldig bra

5

4

3

2

Dårlig

1

--	--	--	--	--

Spesifikt:

Hva synes du om designet på denne siden?

*Skriv litt om det her:*

--

Er det noe funksjonalitet som mangler eller som du ikke skjønner?

*Skriv litt om det her:*

## Meldingsiden – når du klikker på en sak... ser meldingsforløpet...

Generelt

Veldig bra

Dårlig

5

4

3

2

1

--	--	--	--	--

Spesifikt:

Hva synes du om designet på denne siden?

*Skriv litt om det her:*

Er det noe funksjonalitet som mangler eller som du ikke skjønner?

*Skriv litt om det her:*

## Generelt:

Har du brukt lignende systemer før?

JA / NEI

Er det noe du synes bør forandres med systemet?

*Skriv litt om det her:*

Etter å ha prøvd systemet, nevnt et par positive ting med systemet:

*Her:*



Var det noen konkrete feil som oppsto?

*Her:*

Tusen takk for innsatsen!!!!

Du har vært til STOR hjelp!

Kristoffer Jørdre  
Jørn Arild Andenæs  
Børge Johnsen



**Vedlegg  
C  
Resultat av testfasen**

## Testfasen - evaluering

Uken før påske satte gruppen i gang med testfasen. Vi sendte invitasjoner rundt om til utvalgte personer som skulle være med å teste systemet. Dette var medstudenter, bekjente, og personer som kjente til slike systemer fra før. Sammen med invitasjonen fulgte det med et spørreskjema som personene skulle fylle ut. Spørsmålene dreide seg om hvordan designet så ut, hvordan funksjonaliteten virket (se vedlegg – spørreskjema). Meningen med testfasen var å få vite hva som fungerte mindre bra og hva som var bra ved systemet..

Etter å ha evaluert testskjemaene har gruppen oppdaget mange feil som må rettes opp. Selv om enkelte av evalueringsskjemaene ga lite kritikk og mye ros, var heldigvis enkelte kritiske til flere sider ved systemet.

### Kommentarer til systemet – Customer :

- Designet må bli bedre. Mer farger og liv på siden.
- Bruke bedre ord på ticketen. Bruke melding i stedet for ticket. Varierer enkelte steder mellom engelske og norske ord.
- Når man har valgt avdeling, meldingstype, og valgfrie feltene får man ikke opp det som er valgt(selected), men det som opprinnelig stod der.
- Når man går på oppdateringer står det bare ett ord i mange av feltene som skulle hatt flere ord. F.eks i navn, adresse. Adresse nr forsvinner når man trykker oppdater.
- 0 forsvinner i postnr, f.eks til Oslo (0321).
- Bør skrive hvilken grense det er på passord(min 1 tegn, max 8)
- Bedre plass til dato på meldingen. Blir for tett opp i meldingen.
- Ordet avbryt på oppdatere profilen er missledende. Bør bruke tilbakestill i stedet.
- Det er mangel på sikkerhet. Kan forandre på ticketnr i URL`en og du kan få opp ticketen til andre kunder.
- Ny melding i hovedmenyen bør forandres til ny sak. Ordet ”ny melding” er lite intuitivt.
- Når man legger HTML tagger i feltene oppstår det errormeldinger når man trykker submit knappen.
- Kommer ”ikke satt” på meldingstatus når man sender en ny melding.

Vi fikk generelt veldig mye kommentarer på designet på systemet. Dette var ingen overraskelse siden gruppen hadde kun prioritert funksjonaliteten ved systemet. Gruppen fikk også mange tilbakemeldinger på uventede feil. Dette er feil som skal bli rettet opp i videreutvikling fasen.

**Kommentarer til systemet – Owner :**

- Wizar gjør det lettere å begynne å bruke systemet..
- Bør komme rett inn på nye saker når man logger seg på.
- Går ikke å skifte nivå på brukerne av systemet(admin, ansatt).
- Bør være forskjellig farge på meldinger fra owner og customer.
- Textarea på meldingen bør være bredere.
- Ticketen skal bli satt til lest når man går inn på den, ikke når man trykker på submit knappen.
- Sette hvilken ansatt som har skrevet kommentar på saken, ikke firmaet.
- Registrering av ansatt- skrive feil.
- Kan ikke slette kundedefinerte verdier i feltene når man allerede har valgt noen av disse verdiene i en melding.
- På oppdatering av ansatt er det ingen sjekker på om data er av riktig format (f.eks ikke sjekk på nummer). Har ikke satt feltene til maxlengder.
- Når man har oppdatert profilen kommer feil link opp.
- Man bør ikke få endret brukernivået på oppdatere ansatt. En ansatt bør ikke kunne sette administrator rettigheter på seg selv.

Fikk overraskende positive kommentarer til systemet på owner siden. Brukerne var positive til wizard når man skulle registrerer seg som owner og begynne å bruke systemet. Det bør nevnes at noen av testpersonene ikke hadde veldig erfaring med slike systemer før, og derfor kanskje ikke så behovet for funksjoner som ikke var der.

Deltagerne av gruppen har også vært med å testet owner siden i og med at vi satt og kommuniserte med customerene når de testet systemet.

Før vi begynner å vidreutvikle må vi velge hvilke feil som er mest viktige for systemtet på customer og owner siden, og hvor nøye vi skal finpusse designet.



**Vedlegg  
D  
Endrings skjema**



Vi starter med de øverste endringene og arbeider oss nedover. Kviter etterhvert som endringen er gjort.		
<b>Feil</b>	<b>Utføres av</b>	<b>Kvittering</b>
Custdef* fungerer bare i IE, bør fungere i Netscape og Sletting endring av poster, sett til NULL	Jørn Arild Jørn Arild, Børge	Jørn Arild
Får ikke bytta brukernivå på bruker. Bare administrator skal kunne bytte brukernivå på andre. Hvis man er administrator og vil bytte brukernivå på seg selv, bør man få opp en beskjed om at man må være sikker på at det er en som er satt som administrator!	Børge	børge andre kristian per johnsen
Lest/ikke lest skal endres etter lest, ikke etter sendt	Kristoffer	Kristoffer
Det valget som er valgt bør vises i ticket, fjern også ordet selected	Kristoffer	Kristoffer
Når man endrer noe i employe, får ,man kunn opp første ordet i alle feltene, Legg til "". Og epost blir ikke lagret? Sjekk om dette gjelder employe_reg og..	Børge	børge andre kristian per johnsen
0 forsvinner i postnr.. På custom_update?	Kristoffer	fixa
Owner_id bør legges inn i alle query, slik at man ikke lan gå inn på andres ticketer etc..(på ticket bør custom_id også sjekkes)	Kristoffer	Kristoffer
I ticketstatus dukker > opp i netscape	Kristoffer	Kristoffer
<b>Trivielle endringer</b>	<b>Utføres av</b>	<b>Kvittering</b>
Kundeinfo bør vises på meldinger, når owner ser på dem	Kristoffer	Kristoffer
Sjekk om blankt felt bør gjøres i formen	Kristoffer, Jørn Arild, Børge	Børge, Jørn Arild, Krist
Textarea større, med wrapping	Kristoffer	Kristoffer
Firmanavn byttes med navn i ticket	Kristoffer	Kristoffer
Synlig ticket id	Kristoffer, Jørn Arild, Børge	Kristoffer, Jørn Arild, Børge
Bedre ord	Kristoffer, Jørn Arild, Børge	
Beskrivelse av alle felter( eks. Passord maks 8 tegn)	Kristoffer, Jørn Arild, Børge	
Passord skrives inn to ganger	Børge	børge andre kristian per johnsen
Bedre plass til dato på meldinger	Kristoffer	Kristoffer
Få dato fra database ikke fra CF	Kristoffer	fixa
<b>Endringer pri 1</b>	<b>Utføres av</b>	<b>Kvittering</b>
Listing av kunder hos owner	Jørn Arild	Jørn Arild
Avslutte saker		jønn, kriss
Forbedre all kode	Jørn Arild, Kristoffer og han	Børge har gjort hihi
Login, med application.cfm	Jørn Arild	Jørn Arild
Design	Børge	børge, jørn
Utvidet søk	Jørn Arild	Jørn Arild
DATETIME, istedet for varchar		jørn, børge andre
Kataloger	Jørn Arild	Jørn Arild
<b>Endringer pri 2</b>	<b>Utføres av</b>	<b>Kvittering</b>
Navn på person som har ansvar for saken		
Liste med mine saker		strax jønn
Mail?		
Sjekk format på data, sjekk om epost etc.		kris bj
Intern log, kune skrive meldinger som ikke vises til kunden		ikke Lagd.class
Info om hva man er logget inn som		jønn
<b>Endringer pri 3</b>	<b>Utføres av</b>	<b>Kvittering</b>
Sjekk om html i formfelter		BULL
Avtale om kunde med responstid		BULL
Admin del		BULL
Trafikklys		BULL



**Vedlegg  
E  
Gantskjema**



## Hovedprosjektet SoftCross





## Hovedprosjektet SoftCross





**Vedlegg  
F  
Eksempler på kildekode**



```
        update ticket
        set custdef1_id = NULL
        where custdef1_id=#url.custdef1_id#
    </CFQUERY>
    <CFQUERY dbtype="ODBC" datasource="softcross" name="qcustdef1delete">
delete from custdef1
where owner_id=#session.owner_id# and custdef1_id=#url.custdef1_id#
    </CFQUERY>

</cfif>

</cfif>
<!-- Hvis endre -->
<cfif isdefined("form.edit")>
<CFQUERY dbtype="ODBC" datasource="softcross" name="qcustdef1update2">
    update custdef1
    set custdef1_value = '#form.custdef1_value#'
    where custdef1_id=#form.custdef1_id# and owner_id=#session.owner_id#
</CFQUERY>
</cfif>
<!-- Hvis lagre -->
<cfif isdefined("form.new")>
<CFQUERY dbtype="ODBC" datasource="softcross" name="qcustdef1update">
    insert into custdef1(owner_id,custdef1_name, custdef1_value)
    values(#session.owner_id#,'#form.custdef1_name#','#form.custdef1_value#')
</CFQUERY>
</cfif>
<cfif isdefined("form.submit")>
<CFQUERY dbtype="ODBC" datasource="softcross" name="qcustdef1update">
    update custdef1
    set custdef1_name = '#form.custdef1_name#'
    where owner_id=#session.owner_id#
</CFQUERY>
</cfif>
<cfif isdefined("url.onof")>
<CFQUERY dbtype="ODBC" datasource="softcross" name="qownerupdate">
update owner
set owner_cust_def1_value = #url.onof#
where owner_id=#session.owner_id#
</CFQUERY>
<cfset session.owner_cust_def1_value=#url.onof#>
</cfif>

<cfif NOT isdefined("url.delete") or url.delete neq 3>

<CFQUERY dbtype="ODBC" datasource="softcross" name="qcustdef1">
select *
    from custdef1 where owner_id = #session.owner_id#
    order by custdef1_value
</CFQUERY>
<cfif qcustdef1.recordcount eq 0>
<!-- Hvis det ikke finne noen forekomster skal cusdefeltet være av -->
<CFQUERY dbtype="ODBC" datasource="softcross" name="qownerupdate">
```

```
update owner
set owner_cust_def1_value = 0
where owner_id=#session.owner_id#
</CFQUERY>
<cfset session.owner_cust_def1_value=0>
<div class="overskrift">Egendefinert felt nr 1
</div>
<br>
<table border="0" width="400">
  <tr>
    <td colspan=3> Du har ikke opprettet noe egendefinert felt nr. 1, gjør det gjerne nå <td>
  </tr>
  <form method="POST" action="owner_opt_cusdef1.cfm<cfif
isdefined("url.new")>?new=1</cfif">
  <tr>
    <td >Felt navn:</td>
    <td ><input name="custdef1_name" maxlength="30"
value="<cfoutput>#qcustdef1.custdef1_name#</cfoutput>"></td>
    <td ></td>
  </tr>
  <tr>
    <td >Første alternativ</td>
    <td ><input name="custdef1_value" maxlength="30" ></td>
    <td ></td>
  </tr>
  <tr>
    <td ></td>
    <td ><input type="Submit" value="&nbsp;&nbsp;&nbsp;Lagre&nbsp;&nbsp;&nbsp;" name="new"></td>
    <td ></td>
  </tr>
</form>
</table>
<cfelse>
<cfparam name="url.action" default="none">
<br>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td class="overskrift" colspan="2" >Egendefinert felt nr 1: <i><cfoutput> #qcustdef1.custdef1_name#
</cfoutput></i></td>
    <td > <a href="owner_opt_cusdef1.cfm?action=heading">Endre overskrift</a></td>
  </tr>
  <tr>
    <td colspan=3> &nbsp;&nbsp;&nbsp;</td>
  </tr>
</table>
```



```
<tr>
<td width=150>Skal feltet vises for kunden:</td>
<td width=200>
  <form action="">
  <input type="radio" name="onof" value="1"
    <cfif session.owner_cust_def1_value eq 1> <cfoutput> checked </cfoutput></cfif>
onclick="window.open('owner_opt_cusdef1.cfm?onof=1<cfif isdefined("url.new")>&new=1</cfif>',
'_self')">JA&nbsp;
    <input type="radio" value="0" name="onof"
    <cfif session.owner_cust_def1_value eq 0> <cfoutput> checked </cfoutput></cfif>
onclick="window.open('owner_opt_cusdef1.cfm?onof=0<cfif isdefined("url.new")>&new=1</cfif>',
'_self')">NEI
  </form>
  </td>
</tr>
<tr>
<td colspan=3> &nbsp;</td>
</tr>
<tr>
<td colspan=3> &nbsp;</td>
</tr>
<!-- Endre overskrift -->
<cfif url.action eq "heading">
<form method="POST" action="owner_opt_cusdef1.cfm<cfif isdefined("url.new")>?new=1</cfif>">
<tr>
<td >Endre overskrift:</td>
<td ><input name="custdef1_name" maxlength="30"
value="<cfoutput>#qcustdef1.custdef1_name#</cfoutput>">
  <input type="hidden" name="custdef1_name_required" value ="Du må fylle ut
overskrift!" ></td>
<td ><input type="Submit" value="&nbsp;&nbsp;&nbsp;Endre&nbsp;&nbsp;&nbsp;" name="submit"></td>
</tr>
</form>
</cfif>

<!-- Legge til en variabel -->
<cfif url.action eq "none">
<form method="POST" action="owner_opt_cusdef1.cfm<cfif
isdefined("url.new")>?new=1</cfif>">
<tr>

<td width=150>Legg til nytt alternativ</td>
<td width=200>
  <input type="hidden" name="custdef1_name"
value="<cfoutput>#qcustdef1.custdef1_name#</cfoutput>">
  <input name="custdef1_value" maxlength="30" >
  <input type="hidden" name="custdef1_value_required" value ="Du må fylle
ut alternativet!" ></td>

  <td><input type="submit" value="Legg til" name="new"> </td>
</tr>
```

```
</form>
</cfif>
  <!-- Endre en variabel -->
  <cfif url.action eq "edit">
    <form method="POST" action="owner_opt_cusdef1.cfm<cfif
isdefined("url.new")>?new=1</cfif">
  <tr>
    <td width=150>Endre alternativet:</td>
    <td width=200>
      <input type="hidden" name="custdef1_id" value="<cfoutput>#url.custdef1_id#</cfoutput">
      <input name="custdef1_value" maxlength="30"
value="<cfoutput>#url.custdef1_value#</cfoutput">
      <input type="hidden" name="custdef1_value_required" value ="Du
må fyller ut alternativet!" ></td>

    <td><input type="submit" name="edit" value="Endre"> </td>
  </tr>

</form>
</cfif>

  <tr>
    <td colspan=3> &nbsp;</td>
  </tr>

<tr >
  <td colspan="3" align="center" bgcolor="SILVER"><b>Alternativer:</b></td>
</tr>
<cfset count=0>
<cfoutput query="qcustdef1">
  <tr <cfif count eq 0> bgcolor="E0E0E0" <cfset count=1> <cfelse> bgcolor="White" <cfset
count=0></cfif>>
  <td >&nbsp;</td>
  <td >#qcustdef1.custdef1_value#</td>
  <td >
    <a
href="owner_opt_cusdef1.cfm?action=edit&custdef1_id=#qcustdef1.custdef1_id#&custdef1_value=#
qcustdef1.custdef1_value#<cfif isdefined("url.new")>&new=1</cfif">Endre</a>
    <a href="owner_opt_cusdef1.cfm?delete=1&custdef1_id=#qcustdef1.custdef1_id#<cfif
isdefined("url.new")>&new=1</cfif">Slett</a>
  </td>
</tr>
</cfoutput>
  <tr>
    <td colspan=3> &nbsp;</td>
  </tr>
</table>
</cfif>
```



## Hovedprosjektet SoftCross



```
<cfif isdefined("url.new")>
  <p align="center"> <a href="owner_opt_cusdef2.cfm?new=1">Neste side --> </a> </p>
</cfif>
</cfif>
<cfinclude template="../footer.cfm">
```



<!--

-----  
Copyright (c) 2001 by Softcross  
-----

File: owner\_ticket.cfm  
-----

Description : legger inn svar og delegerer en allerede laget ticket. Hvis ticketen settes til ferdig, kan det ikke sendes flere meldinger.

Date: 01-Mai-2001  
Author: Kristoffer Jørdrø  
Project: Softcross  
-----

-->

```
<cfinclude template="../header.cfm">
<cfif isdefined("form.posted")>
<!-- henter tid fra databasen(oracle)-->
<CFQUERY NAME="qtidspunkt" DATASOURCE="WOK_odbc">
SELECT sysdate from DUAL
</CFQUERY>
<CFSET todayDate = CreateODBCdate(#qtidspunkt.sysdate#)>
<cfquery dbtype="ODBC" name="qticket" datasource="softcross">
select department_id from ticket where ticket_no=#form.id#
</cfquery>

<cfquery dbtype="ODBC" name="qupdate_ticket" datasource="softcross">
UPDATE ticket
SET
<cfif #form.department_id# NEQ "">
department_id = #form.department_id#,
</cfif>
<cfif #form.tickstat_id# NEQ "">
tickstat_id = #form.tickstat_id#,
</cfif>
<cfif #form.priority_id# NEQ "">
priority_id = #form.priority_id#,
</cfif>
tickettype_id = #form.tickettype_id#,
ticket_cust_read=0,
<cfif #form.ticket_closed# EQ "ja">
ticket_closed = 1,
</cfif>
<cfif #form.ticket_description# EQ "" and #form.department_id# NEQ #qticket.department_id#>
ticket_own_read=0
<cfelse>
ticket_own_read=1
</cfif>
WHERE
ticket_no = #form.id#
</cfquery>
```



```

<cfif #form.ticket_description# NEQ "">
<cfquery dbtype="ODBC" name="qupdate_customer" datasource="softcross">
INSERT INTO log
(log_timestamp,
ticket_no,
log_text,
log_author)
values(
<cfoutput>#variables.todayDate#</cfoutput>,
#form.id#,
'#form.ticket_description#',
'#session.empl_name#'
)
</cfquery>

</cfif><!-- end sjekk om det er en tekst i meldingen som er sendt if 2.-->

<cflocation url="owner_ticket.cfm?ticket_no=#form.id#&ny=1">

<!--
*****-->
<!-- hvis ikke posted-->

<cfelse>
<cfquery dbtype="ODBC" name="qticket" datasource="softcross">
select
department_name, ticket.department_id ,
tickettype_name, ticket.tickettype_id,
tickstat_name, ticket.tickstat_id,
priority_name, ticket.priority_id,
ticket_subject, ticket_no, ticket_closed, ticket_own_read,
custdef1_name, ticket.custdef1_id, custdef1_value,
custdef2_name, ticket.custdef2_id, custdef2_value,
custdef3_name, ticket.custdef3_id, custdef3_value,
customer_name, customer_email, customer_phone1, customer_phone2, customer_fax,
customer_address, customer_contact_name, customer_zipplace
FROM customer,ticket, department, priority, ticketstatus,
tickettype, custdef1, custdef2, custdef3
where
ticket.ticket_no=#url.ticket_no# and
ticket.customer_id = customer.customer_id and
customer.owner_id = #session.owner_id# and
ticket.department_id = department.department_id(+) and
ticket.priority_id = priority.priority_id(+) and
ticket.tickstat_id = ticketstatus.tickstat_id(+) and
ticket.tickettype_id = tickettype.tickettype_id(+) and
ticket.custdef1_id = custdef1.custdef1_id(+) and
ticket.custdef2_id = custdef2.custdef2_id(+) and
ticket.custdef3_id = custdef3.custdef3_id(+)
</cfquery>
<cfquery datasource="softcross" dbtype="ODBC" name="qticket_log">
select LOG_TIMESTAMP, LOG_TEXT, LOG_AUTHOR
from log, ticket, customer

```



```
where log.TICKET_NO = #url.ticket_no# and log.ticket_no=ticket.ticket_no and ticket.customer_id
= customer.customer_id and customer.owner_id=#session.owner_id#
</cfquery>
```

```
<cfif not isdefined("url.ny")>
<!-- setter ticketen til lest når man går inn på den-->
<cfquery datasource="softcross" dbtype="ODBC" name="qcust_read">
update ticket SET
ticket_cust_read=0,
ticket_own_read=1
where ticket_no=#url.ticket_no#
</cfquery>
</cfif>
```

```
<div class="overskrift">Melding:
&nbsp;<i><cfoutput>#qticket.ticket_subject#</cfoutput></i></div>
<form action="owner_ticket.cfm" method="post">
  <TABLE width="100%" >
  <TR>
  <TD colspan="4" class="overskrift" class="table_over"><P align=center><font
color="white">Kunde info:</font> </TD></class>
  </TR>
  <TR width="100%">
  <TD width="25%">Navn: </TD>
  <TD width="25%"><cfoutput>#qticket.customer_name#</cfoutput> </TD>
  <TD width="25%">Telefon: </TD>
  <TD width="25%"> <cfoutput>#qticket.customer_phone1#</cfoutput></TD>
  </TR>
  <TR>
  <TD>Adresse:</TD>
  <TD><cfoutput>#qticket.customer_address#</cfoutput></TD>
  <TD>Mobil:</TD>
  <TD><cfif #qticket.customer_phone2# EQ "0"><cfoutput></cfoutput>
  <cfelse>
  <cfoutput>#qticket.customer_phone2#</cfoutput></cfif>
  </TD></TR>
  <TR>
  <TD>Poststed:</TD>
  <TD><cfoutput>#qticket.customer_zipplace#</cfoutput></TD>
  <TD>Fax:</TD>
  <TD><cfif #qticket.customer_fax# EQ "0"><cfoutput></cfoutput>
  <cfelse>
  <cfoutput>#qticket.customer_fax#</cfoutput></cfif>
  </TD></TR>
  <TR>
  <TD>Email:</TD>
  <TD><a
href="mailto:<cfoutput>#qticket.customer_email#</cfoutput>"><cfoutput>#qticket.customer_email#
</cfoutput></a></TD>
  <TD>Kontaktperson:</TD>
  <TD><cfoutput>#qticket.customer_contact_name#</cfoutput></TD></TR>
  <TR>
```



```
</select> </td></tr>
<tr>
<TD colspan="2" >Status:</TD>
<TD colspan="2" ><SELECT size=1 name=tickstat_id >
<cfif #qticket.tickstat_id# EQ ""><option value=""> Ikke satt </option></cfif>
<cfloop index="Counter" from=1 to="#ArrayLen(session.aticketstatus)#">
<cfoutput>
<option value="#session.aticketstatus[Counter][1]#"
<cfif #qticket.tickstat_id# EQ #session.aticketstatus[Counter][1]#>
selected
</cfif>
>#session.aticketstatus[Counter][2]#</option>
</cfoutput>
</cfloop>
</select> </td></tr>

<cfif #session.owner_cust_def1_value# EQ 1 and #qticket.CUSTDEF1_NAME# NEQ "">
<TR>
<TD colspan="2"><cfoutput>#qticket.CUSTDEF1_NAME#</cfoutput></TD>
<TD colspan="2"><cfoutput>#qticket.CUSTDEF1_VALUE#</cfoutput></TD>
</TR></cfif>

<cfif #session.owner_cust_def2_value# EQ 1 and #qticket.CUSTDEF2_NAME# NEQ "">
<TR>
<TD colspan="2"><cfoutput>#qticket.CUSTDEF2_NAME#</cfoutput></TD>
<TD colspan="2"><cfoutput>#qticket.CUSTDEF2_VALUE#</cfoutput></TD>
</TR></cfif>

<cfif #session.owner_cust_def3_value# EQ 1 and #qticket.CUSTDEF3_NAME# NEQ "">
<TR>
<TD colspan="2"><cfoutput>#qticket.CUSTDEF3_NAME#</cfoutput></TD>
<TD colspan="2"><cfoutput>#qticket.CUSTDEF3_VALUE#</cfoutput></TD>
</TR></cfif>
<TR>
<TD>&nbsp;</TD>
<TD>&nbsp;</TD>
<TD>&nbsp;</TD></TR>
</TABLE>

<table width="100%">
<TR >
<TD class="table_over" width="25%"><font color="white">Dato:</font></TD>
<TD class="table_over" width="50%"><P align=center><font
color="white">Melding:</font></p></TD>
<TD class="table_over" width="25%"><font color="white">Fra:</font></TD></TR></class>

<cfset count="0">
<cfoutput query="qticket_log">
<cfif count EQ "0">
```







<!--

-----  
SoftCross mainproject on HIG  
-----

File: login.cfm  
-----

Description : Login side.

Hvis owner\_id sendes med i urlen vises velkomstteksten, og logoen til den owneren som har denne iden.

Koden sjekker om personen som logger inn er employee eller customer og hvilket brukernivå de skal logge inn på.

Legger også alle variable i session for senere bruk. Legger felter som det ikke er standard antall av i arrays.

Date: 05-Mai-2001

Author: Jørn Arild Andenæs

Project: Softcross  
-----

-->

```
<CFSET temp = StructDelete(Session, "loggedin")>
<CFSET temp = StructDelete(Session, "empl_id")>
<CFSET temp = StructDelete(Session, "owner_id")>
<cfif IsDefined("url.owner_id")>
<CFQUERY NAME="qowner" DATASOURCE="softcross">
SELECT      *
FROM        owner
WHERE owner_ID='#url.owner_id#'
</CFQUERY>
</cfif>
<cfparam name="qowner.owner_bg_color" default="white">
<cfparam name="qowner.status_id" default="1">
<cfparam name="qowner.recordcount" default="1">
<cfparam name="qowner.owner_name" default="SoftCross">
<cfparam name="qowner.owner_text" default="">
<cfparam name="qowner.owner_imagepath" default="">
<cfparam name="qowner.owner_font" default="Verdana">

<html>
<head>
    <title>Softcross login</title>
<cfinclude template="../stylesheets.cfm">
</head>
<body bgcolor="<cfoutput>#qowner.owner_bg_color#</cfoutput>" >
<!-- Hvis ownerid finnes i databasen og ikke er satt til inaktiv-->
<cfif qowner.status_ID eq 1 and qowner.recordcount gt 0>

<!-- hvis brukernavn og passord er sendt med-->
<cfif IsDefined("form.username") and IsDefined("form.password") >
<!-- sjekke om employee eller customer -->
<cfquery name="qemployee" datasource="softcross" dbtype="ODBC">
SELECT      *
FROM        employee,department
WHERE EMPL_USERNAME = '#form.username#' and EMPL_PASSWD = '#form.password#' and
employee.department_id = department.department_id
```



```
</CFQUERY>
<cfif #qemployee.RecordCount# gt 0> <!-- hvis det er en employee som logger seg på-->
<cfset SESSION.empl_id = qemployee.empl_id>
<cfset SESSION.empl_name = qemployee.empl_name>
<cfset SESSION.empl_address = qemployee.empl_address>
<cfset SESSION.empl_email = qemployee.empl_email>
<cfset SESSION.department_id = qemployee.department_id>
<cfset SESSION.empl_cellphone = qemployee.empl_cellphone>
<cfset SESSION.empl_phone = qemployee.empl_phone>
<cfset SESSION.empl_username = qemployee.empl_username>
<cfset SESSION.empl_passwd = qemployee.empl_passwd>
<cfset SESSION.empl_zipcode = qemployee.empl_zipcode>
<cfset SESSION.empl_zipplace = qemployee.empl_zipplace>
<cfset SESSION.thelevel_id = qemployee.thelevel_id>
<cfquery name="qthelevel" datasource="softcross" dbtype="ODBC">
SELECT      *
FROM        thelevel
WHERE thelevel_id = #SESSION.thelevel_id#
</CFQUERY>
<cfset SESSION.thelevel_name = qthelevel.thelevel_name>
<!-- Henter ut liste med alle kunder og legger i en array -->
<cfquery name="qcustomer" datasource="softcross" dbtype="ODBC">
SELECT      *
FROM        customer, department, employee
WHERE employee.empl_id = #SESSION.empl_id# and employee.department_id =
department.department_id and customer.owner_id = department.owner_id
order by customer_name
</CFQUERY>
<cfset session.acustomer=ArrayNew(2)>
<cfoutput query="qcustomer">
<cfset session.acustomer[ArrayLen(session.acustomer)+1][1]=#qcustomer.customer_id#>
<cfset session.acustomer[ArrayLen(session.acustomer)][2]=#qcustomer.customer_name#>
<cfset session.acustomer[ArrayLen(session.acustomer)][3]=#qcustomer.customer_address#>
<cfset session.acustomer[ArrayLen(session.acustomer)][4]=#qcustomer.customer_email#>
<cfset session.acustomer[ArrayLen(session.acustomer)][5]=#qcustomer.customer_fax#>
<cfset session.acustomer[ArrayLen(session.acustomer)][6]=#qcustomer.customer_phone1#>
<cfset session.acustomer[ArrayLen(session.acustomer)][7]=#qcustomer.customer_phone2#>
<cfset session.acustomer[ArrayLen(session.acustomer)][8]=#qcustomer.customer_contact_name#>
<cfset session.acustomer[ArrayLen(session.acustomer)][9]=#qcustomer.customer_description#>
<cfset session.acustomer[ArrayLen(session.acustomer)][10]=#qcustomer.customer_country#>
<cfset session.acustomer[ArrayLen(session.acustomer)][11]=#qcustomer.customer_zipcode#>
<cfset session.acustomer[ArrayLen(session.acustomer)][12]=#qcustomer.customer_zipplace#>
</cfoutput>

<cfelse> <!-- hvis det ikke er en employee må det være en customer-->
<cfquery name="qcustomer" datasource="softcross" dbtype="ODBC">
SELECT      *
FROM        CUSTOMER
WHERE CUSTOMER_USERNAME = '#form.username#' and CUSTOMER_PASSWD =
'#form.password#'
</CFQUERY>
```

```
<cfif #qcustomer.RecordCount# gt 0> <!-- hvis det er en customer som logger seg på-->
<cfset SESSION.customer_id = qcustomer.customer_id>
<cfset SESSION.customer_name = qcustomer.customer_name>
<cfset SESSION.customer_address = qcustomer.customer_address>
<cfset SESSION.customer_email = qcustomer.customer_email>
<cfset SESSION.customer_fax = qcustomer.customer_fax>
<cfset SESSION.customer_phone1 = qcustomer.customer_phone1>
<cfset SESSION.customer_phone2 = qcustomer.customer_phone2>
<cfset SESSION.customer_contact_name = qcustomer.customer_contact_name>
<cfset SESSION.customer_username = qcustomer.customer_username>
<cfset SESSION.customer_passwd = qcustomer.customer_passwd>
<cfset SESSION.customer_description = qcustomer.customer_description>
<cfset SESSION.customer_country = qcustomer.customer_country>
<cfset SESSION.customer_zipcode = qcustomer.customer_zipcode>
<cfset SESSION.customer_zipplace = qcustomer.customer_zipplace>
<cfset SESSION.thelevel_name = "Kunde">
<cfset SESSION.thelevel_id = 3>
<cfelse> <!-- Hvis det verken er en customer eller employee, er det feil brukernavn-->
<cfset error=1>

</cfif><!-- end sjekk om det er en customer-->
</cfif><!-- end sjekk om det er en employee-->
<cfif not isdefined("error")>
<!-- legger felles variable for owner og customer i session -->
<cfif isdefined("SESSION.empl_id")>
<CFQUERY NAME="qowner" DATASOURCE="softcross">
SELECT      *
FROM        owner,department
WHERE
department.department_id = #qemployee.department_id# and department.owner_id = owner.owner_id
</CFQUERY>
<cfelse>
<CFQUERY NAME="qowner" DATASOURCE="softcross">
SELECT      *
FROM        owner,customer
WHERE
customer.customer_id = #qcustomer.customer_id# and customer.owner_id = owner.owner_id
</CFQUERY>
</cfif>
<cfset SESSION.loggedin = 1>
<cfset SESSION.owner_id = qowner.owner_id>
<cfset SESSION.owner_name = qowner.owner_name>
<cfset SESSION.owner_address = qowner.owner_address>
<cfset SESSION.owner_phone = qowner.owner_phone>
<cfset SESSION.owner_email = qowner.owner_email>
<cfset SESSION.owner_fax = qowner.owner_fax>
<cfset SESSION.owner_country = qowner.owner_country>
<cfset SESSION.owner_cust_def1_value = qowner.owner_cust_def1_value>
<cfset SESSION.owner_cust_def2_value = qowner.owner_cust_def2_value>
<cfset SESSION.owner_cust_def3_value = qowner.owner_cust_def3_value>
<cfset SESSION.owner_imagepath = qowner.owner_imagepath>
<cfset SESSION.owner_bg_color = qowner.owner_bg_color>
<cfset SESSION.owner_text = qowner.owner_text>
<cfset SESSION.owner_font = qowner.owner_font>
```

```
<cfset SESSION.owner_zipcode = qowner.owner_zipcode>
<cfset SESSION.owner_zipplace = qowner.owner_zipplace>
<!-- Henter ut liste med alle ansatte og legger i en array -->
<cfquery name="qemployee2" datasource="softcross" dbtype="ODBC">
    SELECT      *
    FROM        department, employee, thelevel
    WHERE       employee.department_id = department.department_id and department.owner_id =
#session.owner_id#
    and employee.thelevel_id = thelevel.thelevel_id
order by empl_name
</CFQUERY>
<cfset session.aemployee=ArrayNew(2)>
<cfoutput query="qemployee2">
<cfset session.aemployee[ArrayLen(session.aemployee)+1][1]=#qemployee2.empl_id#>
<cfset session.aemployee[ArrayLen(session.aemployee)][2]=#qemployee2.empl_name#>
<cfset session.aemployee[ArrayLen(session.aemployee)][3]=#qemployee2.empl_address#>
<cfset session.aemployee[ArrayLen(session.aemployee)][4]=#qemployee2.empl_email#>
<cfset session.aemployee[ArrayLen(session.aemployee)][5]=#qemployee2.department_id#>
<cfset session.aemployee[ArrayLen(session.aemployee)][6]=#qemployee2.empl_cellphone#>
<cfset session.aemployee[ArrayLen(session.aemployee)][7]=#qemployee2.thelevel_id#>
<cfset session.aemployee[ArrayLen(session.aemployee)][8]=#qemployee2.empl_phone#>
<cfset session.aemployee[ArrayLen(session.aemployee)][9]=#qemployee2.empl_username#>
<cfset session.aemployee[ArrayLen(session.aemployee)][10]=#qemployee2.empl_zipcode#>
<cfset session.aemployee[ArrayLen(session.aemployee)][11]=#qemployee2.empl_zipplace#>
<cfset session.aemployee[ArrayLen(session.aemployee)][12]=#qemployee2.department_name#>
<cfset session.aemployee[ArrayLen(session.aemployee)][13]=#qemployee2.thelevel_name#>
</cfoutput>
<cfquery name="qcustdef1" datasource="softcross" dbtype="ODBC">
select * from custdef1 where owner_id = #session.owner_id#
order by custdef1_name
</cfquery>
<cfset session.acustdef1=ArrayNew(2)>
<cfoutput query="qcustdef1">
<cfset session.acustdef1[ArrayLen(session.acustdef1)+1][1]=#qcustdef1.custdef1_id#>
<cfset session.acustdef1[ArrayLen(session.acustdef1)][2]=#qcustdef1.custdef1_name#>
<cfset session.acustdef1[ArrayLen(session.acustdef1)][3]=#qcustdef1.custdef1_value#>
</cfoutput>
<!-- Legger custdef2 i session -->
<cfquery name="qcustdef2" datasource="softcross" dbtype="ODBC">
select * from custdef2 where owner_id = #session.owner_id#
order by custdef2_name
</cfquery>
<cfset session.acustdef2=ArrayNew(2)>
<cfoutput query="qcustdef2">
<cfset session.acustdef2[ArrayLen(session.acustdef2)+1][1]=#qcustdef2.custdef2_id#>
<cfset session.acustdef2[ArrayLen(session.acustdef2)][2]=#qcustdef2.custdef2_name#>
<cfset session.acustdef2[ArrayLen(session.acustdef2)][3]=#qcustdef2.custdef2_value#>
</cfoutput>
<!-- Legger custdef3 i session -->
<cfquery name="qcustdef3" datasource="softcross" dbtype="ODBC">
select * from custdef3 where owner_id = #session.owner_id#
order by custdef3_name
</cfquery>
<cfset session.acustdef3=ArrayNew(2)>
```



```
<cfoutput query="qcustdef3">
<cfset session.acustdef3[ArrayLen(session.acustdef3)+1][1]=#qcustdef3.custdef3_id#>
<cfset session.acustdef3[ArrayLen(session.acustdef3)][2]=#qcustdef3.custdef3_name#>
<cfset session.acustdef3[ArrayLen(session.acustdef3)][3]=#qcustdef3.custdef3_value#>
</cfoutput>
<!-- Legger department i session -->
<cfquery name="qdepartment" datasource="softcross" dbtype="ODBC">
select * from department where owner_id = #qowner.owner_id#
order by department_name
</cfquery>
<cfset session.adepartment=ArrayNew(2)>
<cfoutput query="qdepartment">
<cfset session.adepartment[ArrayLen(session.adepartment)+1][1]=#qdepartment.department_id#>
<cfset session.adepartment[ArrayLen(session.adepartment)][2]=#qdepartment.department_name#>
<cfset
session.adepartment[ArrayLen(session.adepartment)][3]=#qdepartment.department_description#>
</cfoutput>
<!-- Legger priority i session -->
<cfquery name="qpriority" datasource="softcross" dbtype="ODBC">
select * from priority where owner_id = #session.owner_id#
order by priority_name
</cfquery>
<cfset session.apriority=ArrayNew(2)>
<cfoutput query="qpriority">
<cfset session.apriority[ArrayLen(session.apriority)+1][1]=#qpriority.priority_id#>
<cfset session.apriority[ArrayLen(session.apriority)][2]=#qpriority.priority_name#>
<cfset session.apriority[ArrayLen(session.apriority)][3]=#qpriority.priority_description#>
</cfoutput>
<!-- Legger tickettype i session -->
<cfquery name="qtickettype" datasource="softcross" dbtype="ODBC">
select * from tickettype where owner_id = #session.owner_id#
order by tickettype_name
</cfquery>
<cfset session.atickettype=ArrayNew(2)>
<cfoutput query="qtickettype">
<cfset session.atickettype[ArrayLen(session.atickettype)+1][1]=#qtickettype.tickettype_id#>
<cfset session.atickettype[ArrayLen(session.atickettype)][2]=#qtickettype.tickettype_name#>
<cfset session.atickettype[ArrayLen(session.atickettype)][3]=#qtickettype.tickettype_description#>
</cfoutput>
<!-- Legger ticketstatus i session -->
<cfquery name="qticketstatus" datasource="softcross" dbtype="ODBC">
select * from ticketstatus where owner_id = #session.owner_id#
order by tickstat_name
</cfquery>
<cfset session.aticketstatus=ArrayNew(2)>
<cfoutput query="qticketstatus">
<cfset session.aticketstatus[ArrayLen(session.aticketstatus)+1][1]=#qticketstatus.tickstat_id#>
<cfset session.aticketstatus[ArrayLen(session.aticketstatus)][2]=#qticketstatus.tickstat_name#>
<cfset session.aticketstatus[ArrayLen(session.aticketstatus)][3]=#qticketstatus.tickstat_description#>
</cfoutput>

<cfif isdefined("session.empl_id")>
<cfif qemployee.thelevel_id eq 2>
```

```
<cflocation
url="../owner/owner.cfm?department_id=#qemployee.department_id#&department_name=#qemployee.department_name#">
</cfif>
<cfif qemployee.thelevel_id eq 1>
<cflocation url="../owner/owner.cfm">
</cfif>
<cfif qemployee.thelevel_id eq 0>
<cflocation url="../owner/employee_list.cfm">
</cfif>
<cfelse>
<cflocation url="../customer/customer.cfm">
</cfif>
<cfelse>
<cfoutput>
Det har oppstått en av følgende feil: <br>
- Brukernavnet #form.username# finnes ikke i databasen. <br>
- Du tastet feil passord <br><br>

Klikk <a href="login.cfm<cfif isdefined("url.owner_id")>?owner_id=#url.owner_id#</cfif>"> her
</a> for å prøve på nytt, eller for å registrere deg som ny bruker.</br>
</cfoutput>
</cfif>
<cfelse><!-- hvis ikke sendt med noe brukernavn -->
<cfoutput>
<BR>
<h3>Velkommen som bruker av #qowner.owner_name#'s sitt kundebehandlingssystem</h3>
<br>
<cfif qowner.owner_imagepath neq "">
 <br>
</cfif>
<form action="login.cfm" method="post">
<!-- Ownerid sendes med i urlen -->
<TABLE>
<tr>
<td colspan="2"> #qowner.owner_text# <br><br></td>
</tr>
<TR>
<TD>Brukernavn:&nbsp;   </TD>
<TD><INPUT maxLength=10 size=10 name="username"><input type="hidden"
name="username_required" value ="Du må fylle ut brukernavn!" ></TD></TR>
<TR>
<TD>Passord:</TD>
<TD><INPUT type=password maxLength=10 size=10
name="password"><input type="hidden" name="password_required" value ="Du må fylle ut
passord!" ></TD>
</TR>
<TR>
<TD>
<TD>
<INPUT type=submit value="Logg inn"
name="submit"></TD></TD>
</td></TR></TABLE></form>
```



```
<cfif isdefined("url.owner_id")>
  <P>
    Er du ikke registrert som kunde i vårt system? Registrer deg <a
href=" ../customer_reg.cfm?owner_id=#url.owner_id#">her </a><BR></P>
</cfif>
</cfoutput>
</cfif> <!-- Hvis ikke brukernavn og passord finnes i databasen -->
<cfelse>
Det har oppstått en av følgende feil:
- Ownerid'en som ble sendt med fantes ikke i databasen <br>
- Bedriften sine sider er satt til innaktiv
</cfif> <!-- Hvis ikke brukernavn og passord er sendt med -->
```

```
</body>
</html>
```





**Vedlegg  
G  
Eksempel på møtereferater**

## 1 Møtereferat for SoftCross

**Dato: 31.01.01**

**Sted: Grupperommet**

**Deltagere: Jørn, Kristoffer, Børge og Tom**

### **Referat:**

**Vi snakket om innføringskurset i helgen med Arne Sunde.**

**Vi spurte Tom om han viste om noen gamle hovedprosjekter som vi kunne bruke og det gjorde han. Tom skal maile oss om dette.**

**Tom syntes også at SA modellering var en grei måte å jobbe på siden prosjektet dreier seg mye om kjernen i systemet (databasen).**

**Det blir ikke noe møte onsdag den 14 da gruppen skal til Oslo i møte med Arne Sunde.**

**Tom nevnte om vi skulle levere statusrapport før eller etter statusmøte. Dette må vi diskutere til neste møte.**

**Vi snakket også om hvem vi skal teste mot i testfasen. Dette skal vi forhøre oss med Arne Sunde om.**

## 2 Møtereferat for SoftCross

**Dato: 29.01.01**

**Sted: Gjøvik**

**Deltagere: Børge Johnsen, Jørn Arild Andenæs, Kristoffer Jørdre**

### Referat:

Oppsummerte uken som gikk. Har holdt det planlagte programmet i henhold til Gantskjemaet. Fikk oppfylt de forventningene gruppen hadde til møtet som ble holdt i helgen. Ble litt liten tid til å gå gjennom kravspesifikasjonen, men hoved målet var å få en innføring i ColdFusion.

Den følgende uken skal gruppen konsentrere seg om:

- Gjøre research
  - på internett(gå gjennom linker som Arne Sunde har sendt oss)
  - se gjennom tidligere hovedprosjekter
  - Sende mail til Arne Sunde for å få litt mer opplysninger om kundebehandling systemer(Jørn A Andenæs).
- Fikse koblingen mellom mellom MYODBC og CF – serveren. Denne jobben skal gjøres av Børge Johnsen
- Prøve hvis man får tid å fikse versjons kontroll system ( WinCVS ). Prøve å få dette til å fungere på serveren
- Gjøre ferdig forfallende arbeid fra tidligere perioder i prosjektet.
- Se litt på Rapport skrivning som er gjort på tidligere prosjekter. Skal begynne denne uken med dette( i henhold til Gant skjema) og gruppen har liten erfaring med dette før(Kristoffer Jørdre).
- Lese i CF bøkene slik at vi får en bedre forståelse for prog. I CF

### 3 Møtoreferat for SoftCross

**Dato: 26.02.01**

**Sted: Grupperom**

**Deltagere: Jørn Arild Andenæs, Børge Johnsen, Kristoffer Jørdre**

#### Referat:

Gjennomgikk forrige ukeplan. Gruppen har fått gjort alle målene som ble satt. Har fått rettet opp endel i forhold til gant – skjemaet. Gruppen er godt fornøyd med forrige ukes arbeid.

Vi fant ut at vi skal gjøre følgende denne uken:

Denne uken har gruppen satt av tid til programmering. Det er hovedoppgaven denne uken. Programmeringen må planlegges slik at vi får mest mulig oversikt over hvem som skal gjøre hva på forhånd.

Vi skal også ha statusmøte onsdag 28 feb. Gruppen skrev status rapport i slutten av forrige uke, men vi må rette opp noe på den denne uken.

Det gjenstår noen småting på databasen som må rettes opp.

Det er også noe som må rettes opp på kravspekken før vi kan si oss helt ferdige med den. Har fått en faks fra Arne med endel ting som han mener vi bør rette opp på kravspekken.



**Vedlegg  
H  
Statusrapporter**



## Statusrapport 28.februar 2001

### 1. Status

#### Planlegging

I begynnelsen av prosjektet lagde vi ett gantskjema. Dette har vi klart å holde bra, uten at vi har flyttet noen tidsfrister.. Vi har hatt ukemøter hver mandag der vi har planlagt hva vi skal gjøre neste uke. Her har vi klart å rettet opp mye av skjevheter i forhold til gantsjemaet. Vi har funnet ut at vi muligens må bruke litt mindre tid på forberedelser av testing, testing og evaluering for å få bedre tid til programmering. Vi har satt av forholdsvis god tid til Test perioden. Vi ser at programmeringsjobben muligens kan ta litt mer tid enn planlagt.

Gruppen har vært flinke til å si i fra om dager, helger der man blir borte slik at dette blir klargjort god tid på forhånd. Dette har lettet arbeidet for gruppen i forhold til arbeid mot tidsfrister.

#### Organisering av gruppens arbeid og ansvarsområder

Gruppen satte ansvarsområder i starten av prosjektet etter kunnskap og erfaring innen hvert enkelt område. Dette har fungert bra. Arbeid innad i gruppen har blitt deligert på hvert ukemøte og det har vært lite problemer med at arbeid ikke har blitt gjort, og at flere har gjort samme jobb. Gruppen har prøvd å fordele lik arbeidsmengde på hver gruppelem, selv om det er individuelle forskjeller på hvor lang tid man bruker på arbeidet.

#### Klargjøring av problemstillingen/systemering

Problemstillingen er satt i forprosjektet og vi har foreløpig ingen grunn til å forandre på denne enda.

#### Løsningsmetode/koding

Siden vi ikke har blitt ferdig med datamodellen enda, har vi ikke begynt med programmeringen enda.. Vi har begynt å sette opp skjermbilder og tenkt litt på funksjonaliteten bak dette. Har tenkt litt på hvordan kodingen blir i forhold til funksjonaliteten, men skal starte for alvor med koding den neste uken.

#### Rapportskriving

Rapportskrivingen er bare i start fasen enda.. Vi har dokumentert utviklingsmiljøet, datamodellen, møtereferater. Rapportskrivingen starter for fullt når vi begynner å programmere. Vi har valgt å følge rapportskriving i henhold til systemutviklingsmodellen.

### 2. Totalstatus for punktene over (oppsummering)

Gruppen er bra fornøyd med hvordan fordelingen av oppgavene har blitt løst og hvordan de har blitt gjort. Gruppen har funnet ut at enkelte deler av gantsjemaet kan ta litt mer tid enn planlagt. Derfor har vi sett gjennom dette god tid på forhånd, som er positivt for videre arbeid. Føler at generelt planlegging av arbeid har foregått over forventningene, det negative er heller utførelsen av det planlagte arbeidet.

### 2. Totalstatus for punktene over (oppsummering)

Gruppen er bra fornøyd med hvordan fordelingen av oppgavene har blitt løst og hvordan de har blitt gjort. Gruppen har funnet ut at enkelte deler av gantsjemaet kan ta litt mer tid enn planlagt. Derfor har vi sett gjennom dette god tid på forhånd, som er positivt for videre arbeid. Føler at generelt



planlegging av arbeid har foregått over forventningene, det negative er heller utførelsen av det planlagte arbeidet.

### 3. Hva er under arbeid?

Grappa har avsluttet kravspesifikasjonen og har så vidt begynt på utvikle prototypen. Grappa vil fremover mot testingen konsentrere seg om koding.

### 4. Er tidsfristene

- overholdt, Utviklingsmiljøet (men selvfølgelig er det stadig ting å rette opp), Opplæring(men vi lærer jo så lenge vi lever), Research

- overskredet, Kravspesifikasjon – 1 uke forsinket

#### Hva med motivasjon:

Gruppens samarbeid, arbeidsformer og organisering.

Gruppen har samarbeidet bra.

Arbeidsformer: Vi har hittil jobbet endel med de samme tingene alle sammen. Tenker da på kravspesifikasjon og forprosjekt primært. Vi vil dele arbeidsoppgaver mere mellom oss når vi nå setter i gang med koding for fullt.

Organisering: Organiseringen av rollene i grappa har fungert bra. Jørn har tatt ansvar som leder, Krisoffer som dokumentansvarlig og Børge som ansvarlig for utviklingsmiljøet.

Vi synes det er oppmuntrene å jobbe i gruppe på et slikt prosjekt. Dette minner om en senere jobbsituasjon og det liker vi!

### 5.Hvordan oppleves veilederkontakt:

Vi har hatt møte med veileder en gang i uka med få unntak. Dette opplever vi som veldig positivt. Da vi ikke har vært gjennom så store prosjekter før er det fint å spørre veileder.

Vi er fornøyde med veileder. Prosjektet vårt er et typisk systemutviklings prosjekt med datamodelering av database. Tom har god innsikt i dette!

Mellom disse momenter kan det være overlapping, og det kan være forhold som er uteglemt fra denne forslagslisten.

Gruppen og veileder kan avtale at det skal lages og benyttes standardkjemaer for rapporteringen. Uansett form skal rapportene dateres og signeres av gruppens leder.

## Statusrapport 21. mars 2001

### 1. Status

#### Planlegging

Siden forrige statusrapport, fulgt gantskjemaet til punkt og prikke. Vi har ukemøter hver mandag der vi evaluerer forrige uke og setter opp ukeplan for hva som skal gjøres.

#### Organisering av gruppens arbeid og ansvarsområder

Gruppen satte ansvarsområder i starten av prosjektet etter kunnskap og erfaring innen hvert enkelt område. Dette har fungert bra. Arbeid innad i gruppen har blitt deligert på hvert ukemøte og det har vært lite problemer med at arbeid ikke har blitt gjort, og at flere har gjort samme jobb. Gruppen har prøvd å fordele lik arbeidsmengde på hver gruppelem, selv om det er individuelle forskjeller på hvor lang tid man bruker på arbeidet.

#### Klargjøring av problemstillingen/systemering

Problemstillingen er satt i forprosjektet. Vi har nesten ferdigutviklet en prototype som samsvarer med problemstillingen, men selvfølgelig er kravspesifikasjonen i stadig utvikling.

#### Løsningsmetode/koding

Vi er i full gang med kodingen. Datamodellen har vi måtte korrigere flere ganger under kodingen, da det har dukket opp behov for nye attributter og entitetstyper. Vi er godt fornøyd med språket Coldfusion.

Vi skal så snart som mulig legge løsningen på en server på EI-teleøst. Løsningen vil da få url'en [www.softcross.com](http://www.softcross.com). I den forbindelse vil det bli gjort noen forandringer i datamodellen, da oppdragsgiver vil benytte Oracle database og ikke MYSQL som vi har brukt på vår server.

#### Rapportskriving

Vi er i gang med rapportskrivningen. Vi prøver å dokumentere mest mulig, slik av vi har et godt grunnlag til sluttrapporten. Vi har dokumentert utviklingsmiljøet, datamodellen, møtereferater og programkode. Vi har valgt å følge rapportskrivning i henhold til systemutviklingsmodellen. Har ikke satt i gang å dokumentere programmeringen enda.

### 2. Totalstatus for punktene over (oppsummering)

Vi er godt fornøyd med prosjektet hittil. Vi har ikke hatt store avvik fra gantskjema noe vi ser på som positivt. Vi var en periode litt nervøse for at vi ikke skulle klare å bli ferdige med programmeringen til testfasen, men det er vi godt på veg til å klare.

### 3. Hva er under arbeid?

Gruppen jobber nå med å få frem en et ferdig produkt til testing. Vi har kodet en kjørbare prototype og koder videre på denne mot testingen. Vi jobber også med planlegging av testfasen. Skal også jobbe



med designet på siden slik at det ser bedre ut og mer oversiktlig. Det gjenstår også å gjøre sidene mere oversiktelige for nye customere og ownere. Få litt mere logisk rettleiding ved registrering.

#### 4. Er tidsfristene

- overholdt, Utviklingsmiljøet (men selvfølgelig er det stadig ting å rette opp), Opplæring (men vi lærer jo så lenge vi lever), Research, implementering av datamodellen, programmering og design (dette jobber vi med videre...)

- overskredet, Kravspesifikasjon – 1 uke forsinket

#### Hva med motivasjon:

Gruppens samarbeid, arbeidsformer og organisering.

Gruppen har samarbeidet bra.

Arbeidsformer: I starten jobbet vi mye med planleggingen sammen. Vi har i den siden sist statusrapport fordelt oppgaver mellom oss og tatt for oss de ulike modulene.

Organisering: Organiseringen av rollene i gruppa har fungert bra. Jørn har tatt ansvar som leder, Krisoffer som dokumentansvarlig og Børge som ansvarlig for utviklingsmiljøet.

Motivasjonen er fremdeles på topp. Gøy å få ferdig en kjørbare utgave av systemet. Spesielt gøy på Thecross serveren som vi har satt opp selv i eget utviklingsmiljø. Prosessen hittil har vært meget lærerik for alle på gruppa!

#### 5. Hvordan oppleves veilederkontakt:

Vi har hatt møte med veileder en gang i uka med få unntak. Dette opplever vi som veldig positivt. Da vi ikke har vært gjennom så store prosjekter før er det fint å spørre veileder.

Vi er fornøyde med veileder. Prosjektet vårt er et typisk systemutviklingsprosjekt med datamodellering av database. Tom har god innsikt i dette!

Har også fått bra med tilbakemelding på hva som kan bli bedre etter at han testet prototypen. Fikk også vite hva som var positivt.



## Statusrapport 25. april 2001

### 1. Status

#### Planlegging

Siden forrige statusrapport har vi fulgt planen. Vi brukte noe lenger tid på koding enn beregnet, men tok oss inn igjen. Vi har kodet, planlagt testfasen, gjennomført testfasen og evaluert. Vi har hatt ukemøter hver mandag, møter med Veileder og oppdragsgiver. Lørdag 21.april hadde vi møte med oppdragsgiver. Sammen gjennomgikk vi feil ved systemet, og satte opp prioriterings rekkefølge på retting av disse feilene.

#### Organisering av gruppens arbeid og ansvarsområder

Gruppen satte ansvarsområder i starten av prosjektet etter kunnskap og erfaring innen hvert enkelt område. Dette har fungert bra. Arbeid innad i gruppen har blitt deligert på hvert ukemøte og det har vært lite problemer med at arbeid ikke har blitt gjort, og at flere har gjort samme jobb. Gruppen har prøvd å fordele lik arbeidsmengde på hver gruppelem, selv om det er individuelle forskjeller på hvor lang tid man bruker på arbeidet.

#### Klargjøring av problemstillingen/systemering

Problemstillingen er satt i forprosjektet. Vi er ferdige med første iterasjon i prosessen vår. Vi har fulgt kravspek'en så godt som mulig og gjort noen endringer.

#### Løsningsmetode/koding

Vi er i full gang med kodingen. Datamodellen har vi måtte korrigere flere ganger under kodingen, da det har dukket opp behov for nye attributter og entitetstyper. Vi er godt fornøyd med språket Coldfusion.

Vi har lagt systemet på softcross.com. Vi måtte endre noe på datamodellen og koden da serveren vi hadde fått beskjed om skulle benytte mysql benyttet oracle.

Etter testfasen har vi funnet endel feil og mangler i koden. Disse driver vi nå og retter opp.

#### Rapportskriving

Vi er i gang med rapportskrivingen. Vi prøver å dokumentere mest mulig, slik av vi har et godt grunnlag til sluttrapporten. Vi har dokumentert utviklingsmiljøet, datamodellen, møtereferater, programkode og testfasen. Vi har valgt å følge rapportskriving i henhold til systemutviklingsmodellen.

Vi har begynt på hovedrapporten. Vi vil i tiden som gjenstår bruke mer og mere tid på rapportskriving, da dette er en viktig del av prosjektet. Vi bruker nå ca 50% av ressursene på rapportskriving. Første steg er å bli ferdig med designdokumentet.

### 2. Totalstatus for punktene over (oppsummering)

Vi er godt fornøyd med prosjektet hittil. Vi har ikke hatt store avvik fra gantskjema noe vi ser på som positivt.

Vi er nå ferdige med testfasen. Vi hadde noe hastverk med å bli ferdige med prototypen til testfasen, men ble ferdige, selv om det selvfølgelig stadig dukker opp nye ting å endre på.

Vi er godt fornøyde med testfasen. Oppdragsgiver testet også systemet og skulle få noen på eltele-øst til å teste. Dette har blitt noe forsinket.

Vi håper vi fremdeles er i rute. Vi har ikke gjennomført noe slikt stort prosjekt før og er noe spente på det endelige resultatet og hvor lang tid vi vil bruke på de forskjellige delene. Så langt har dette gått bra og vi håper å klare alt etter planen.

### 3. Hva er under arbeid?

Vi jobber nå med å gjøre ferdig koding og sluttrapporten

### 4. Er tidsfristene

- overholdt, Utviklingsmiljøet (men selvfølgelig er det stadig ting å rette opp), Opplæring (men vi lærer jo så lenge vi lever), Research, implementering av datamodellen, programmering og design (dette jobber vi med videre...), testfasen.

- overskredet, Kravspesifikasjon – 1 uke forsinket

### Hva med motivasjon:

*Gruppens samarbeid, arbeidsformer og organisering:*

Gruppen har samarbeidet bra.

Arbeidsformer: I starten jobbet vi mye med planleggingen sammen. Vi har i den siden sist statusrapport fordelt oppgaver mellom oss og tatt for oss de ulike modulene og rapportene.

Organisering: Organiseringen av rollene i gruppa har fungert bra. Jørn har tatt ansvar som leder, Krisoffer som dokumentansvarlig og Børge som ansvarlig for utviklingsmiljøet.

Motivasjonen er fremdeles bra, selv om vi gleder oss til å bli ferdige. Vi håper på en grå og trist mai måned som oppfordrer til innendørs rapportskrivning og programmering ☺. Morro at systemet fungerer og ligger på WWW.

### 5. Hvordan oppleves veilederkontakt:

Vi har hatt møte med veileder en gang i uka med få unntak. Dette opplever vi som veldig positivt. Da vi ikke har vært gjennom så store prosjekter før er det fint å spørre veileder.

Vi er fornøyde med veileder. Prosjektet vårt er et typisk systemutviklings prosjekt med datamodellering av database. Tom har god innsikt i dette!

Har også fått bra med tilbakemelding på hva som kan bli bedre etter at han testet prototypen. Fikk også vite hva som var positivt.



## **Vedlegg I Tidsbruk**

Dato:	Navn:	Oppgave:	Timer
[januar>	Børge	Workshop, Oppsetting av server, datamodelering,	75
	Jørn	Workshop, Oppsetting av server, datamodelering,	65
	Kristoffer	Workshop, datamodelering.	50
<b>27.01.01</b>	Kristoffer	Innføringskurs i CF	8
	Børge	Innføringskurs i CF	8
	Jørn Arild	Innføringskurs i CF	8
<b>28.01.01</b>	Kristoffer	Innføringskurs i CF/Diskusjon av kravspekk	5
	Børge	Innføringskurs i CF/Diskusjon av kravspekk	5
	Jørn Arild	Innføringskurs i CF/Diskusjon av kravspekk	5
<b>05.02.01</b>	Kristoffer	Møte/Ordnet opp i doc/ukeplan	3
	Børge	Møte/satt opp server/Kravspek start	4
	Jørn Arild	Møte/møtereferat sette opp cvs	3
<b>3/6/2001</b>	Børge	Brainstorming, kravspek	1
	Kristoffer	Brainstorming, kravspek	1
	Jørn Arild	Brainstorming, kravspek	1
<b>7-Feb</b>	Børge	Kravspek	2
	Kristoffer	Kravspek	2
	Jørn Arild	DFD0 skjema	0.5
	Jørn Arild	Kravspek	2
	Jørn Arild	Weboppset	1
<b>8-Feb</b>	Børge	Kravspek, utviklingsmiljø	5
	Jørn Arild	Kravspek	5
<b>11</b>	Jørn Arild	Kravspek	2
	Kristoffer	Kravspek	2
<b>12</b>	Børge	Kravspek/MØTER	6
	Kristoffer	Kravspek/MØTER	4
	Jørn Arild	Kravspek/MØTER	4
<b>13</b>	Kristoffer	Kravspek	4
	Børge	Kravspek	5
	Jørn Arild	Kravspek	3
<b>14</b>	Kristoffer	Omvisning hos Sunde	2
	Børge	Omvisning hos Sunde	2
	Jørn Arild	Omvisning hos Sunde	2
<b>17</b>	kristoffer	kravspek	0.5
<b>19</b>	Børge	Utviklingsmiljø + ukemøte datamodel	4
	Kristoffer	Møte, dfdskjemaer,ukeplan og datamodell	3
	Jørn Arild	Møte, dfdskjemaer, og datamodell	3
<b>20</b>	Børge	Datamodell	3
	Jørn Arild	Datamodell+skjermbilder	3
	Kristoffer	Datamodell+skjermbilder	3
<b>21</b>	Børge	Møte,	

	Jørn Arild	Møte, møteref, møteinnkalling, database	2
	Kristoffer	Møte, kravspek,skjermbilde	3.5
	<b>22</b> Børge	Kravspek, datamodel	5
	Jørn Arild	Kravspek, datamodel	5.5
	Kristoffer	Kravspek, datamodel	5
	<b>23</b> Jørn Arild	Programmert	4
	Kristoffer	Programmert	4
	<b>26</b> Jørn Arild	Programmert, kravspek, ukemøte	8.5
	Kristoffer	Programmert, kravspek, ukemøte	5
	Børge	Programmert, kravspek, ukemøte	5
	<b>27</b> Jørn Arild	Programmert	5
	Kris	Programmert	5
		Programmert, owner.cfmm,ticket.cfm og owner_opt.cfm, møte	7
	<b>28</b> Jørn Arild	Programmert, møte	7
	Kristoffer	Programmert, møte	4
	Børge	Programmert, møte	4
<b>1.03.01</b>	Jørn Arild	Programmering	5.5
		server oppsett av database + coldfusion hacking	5
	<b>5-Mar</b> børge	programering	2.5
	kristoffer	programmering og kløning med database	3
	Jørn Arild	Programmering og slikt	3
<b>06.03.01</b>	Børge	Programmering	4
	Kristoffer	Programmering av owner_opt og database	4.5
	Jørn Arild	programering av employee reg og ukemøte	4.5
<b>3/7/2001</b>	Børge	programering av ticket og ukemøte	8
	Kristoffer	programmering java og ukemøte	8
	Jørn Arild	Programmering employeer	5
<b>3/8/2001</b>	Børge	programmering av ticket	5
	Kristoffer	programering owner options	5
	Jørn Arild		
	<b>11-Mar</b> Kristoffer	Ticket, SQL setninger	3
	Jørn Arild	owner	2.5
	<b>12-Mar</b> Børge	Programmering customer	3
	Kristoffer	programering av ticket, update og insert	3.5
	Jørn Arild	programering owner options	3.5
<b>13-Mar</b>	Kristoffer	Ticket	1.5
		Ticket + litt på owner_reg/update	
<b>14-Mar</b>	Kristoffer	customer_reg/update	7
	Jørn Arild	Masse forskjellig	7.5
	Børge	Design, header, footer	5.5
<b>15-Mar</b>	Kristoffer	Møte med tom, ticket, testing	5.5
	Børge	Møte med tom, ticket, testing	4.5
	Jørn Arild	Møte med tom, ticket, testing	5.5
<b>19-Mar</b>	Kristoffer	Mandagsmøte, programmering	6

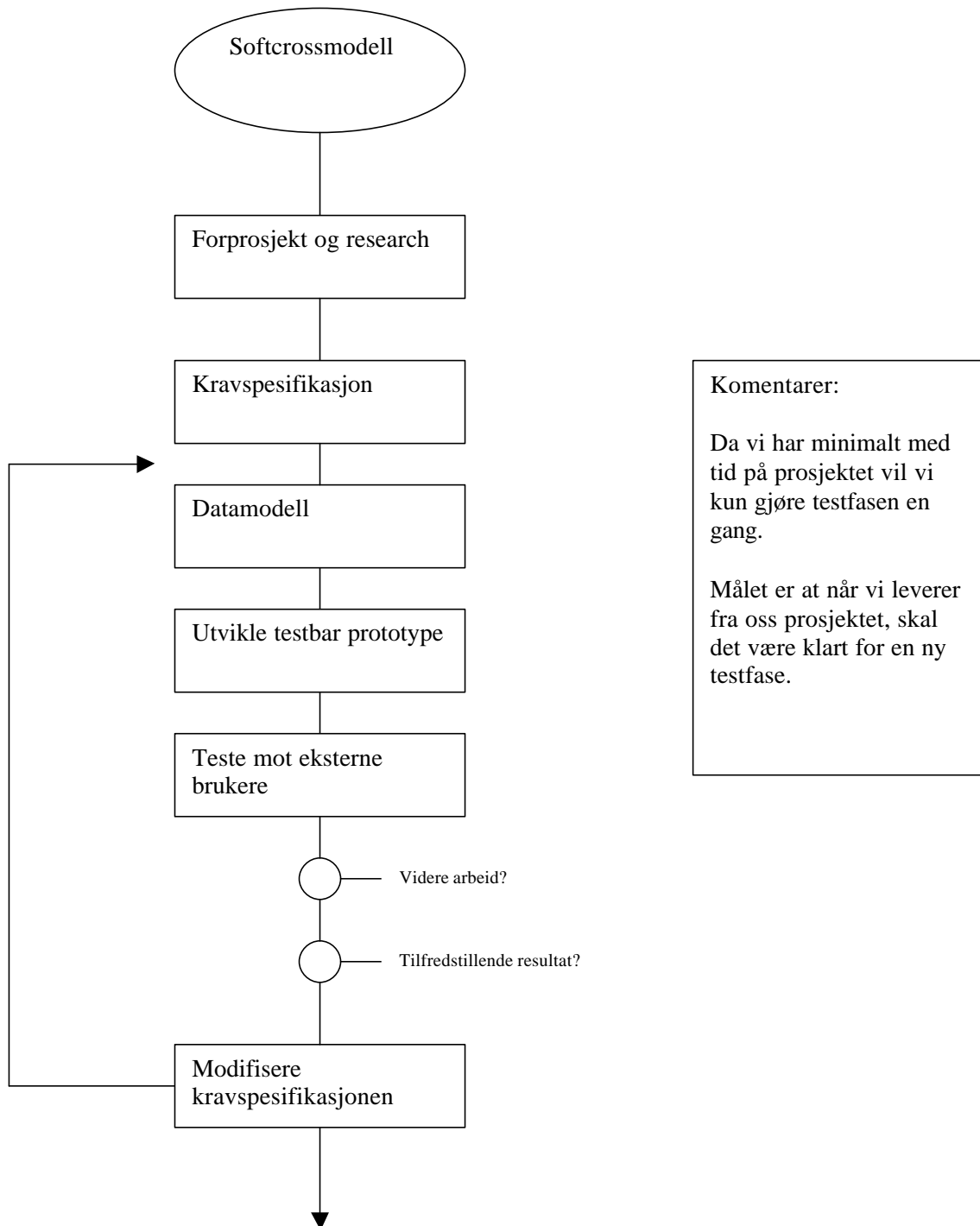
	Børge	Mandagsmøte, programmering	6
	Jørn Arild	Mandagsmøte, programmering	6
<b>21-Mar</b>	Kristoffer	Statusmøte med Tom	1
	Børge	Statusmøte med Tom	1
	Jørn Arild	Statusmøte med Tom, klargjøring for web, programmering	5
<b>22-Mar</b>	Jørn Arild	Programmering, tilrettelegging for www	5
<b>26-Mar</b>	Jørn Arild	Programmering	8
	Børge	Evalueringsskjema	4
	Kristoffer	Programmering	8
<b>2-Apr</b>	Kristoffer	Testing	3.5
	Jørn Arild	Testing	3.5
	Børge	Testing	1
<b>uke 14</b>	Kristoffer	Testing	10
	Jørn Arild	Testing	10
	Børge	Testing	10
<b>18-Apr</b>	Jørn Arild	evaluering	2
	Børge	evaluering	2
	Kristoffer	evaluering	4.5
<b>19-Apr</b>	Jørn Arild	designdokument	4.5
	Børge	rapportskriving - designdokument	4
	Kristoffer	Evalueringsskjema	4
<b>20-Apr</b>	Børge	designdokument	3
	Jørn Arild	designdokument	4
	Kristoffer	designdokument	3
<b>21-Apr</b>	Børge	Møte med oppdragsgiver, Evaluering av testfasen. Prioritering av feil.	4
	Kristoffer	Møte med oppdragsgiver, Evaluering av testfasen. Prioritering av feil.	4
	Jørn Arild	Møte med oppdragsgiver, Evaluering av testfasen. Prioritering av feil.	4
<b>23-Apr</b>	Børge	Koding. Retting av feil. Update_empl. Møte	5
	Kristoffer	Koding. Retting av feil. Ticket. Møte	6
<b>24-Apr</b>	Jørn Arild	Koding. Retting av feil. Custdef. Møte	6
	Jørn Arild	Koding. Retting av feil. Custdef, login	7
	Børge	Koding. Retting av feil. Employee	7
<b>25-Apr</b>	Kristoffer	Koding. Retting av feil. Ticket.Customer_reg	9
	Børge	Status møte + koding	2
	Jørn Arild	Statusmøte, koding og designdok	6
<b>26-Apr</b>	Kristoffer	Statusmøte, koding og designdok	6
	Børge	Forord, innledning, status	2
	Kristoffer	GUI	5.5
<b>27-Apr</b>	Jørn Arild	Designdokument og programmering	5.5
	Kristoffer	Programmering	3
	Jørn Arild	Programmering	5.5
<b>30-Apr</b>	Børge	Design, møte,	5
	Jørn Arild	owner	7
	Kristoffer	javascript	5.5

<b>1-May</b>	Kristoffer	javascript	3.5
	Børge	Design	2
	Jørn Arild	Javascript, alt mulig rart	6.5
<b>2-May</b>	Kristoffer	javascript og stæsj	7
	Børge	database og stæsj	7
	Jørn Arild	Design og implementering	7
<b>3-May</b>	Jørn Arild	Design, diverse småplukk	6.5
	Kristoffer	Ticket, sql, div småplukk	6.5
	Børge	Design, emploee	5
<b>4-May</b>	Kristoffer	Backup, uploading av sider på softcross	7.5
	Jørn Arild	Softcross.com problemer	5.5
	Børge	Softcross.com problemer	2.5
<b>5-May</b>	Kristoffer	Fiksing	4
	Jørn Arild	Fikfaks	5
<b>6-May</b>	Jørn Arild	Testing, feilretting, prog.	5.5
	Kristoffer	Testing, feilretting, prog.	5.5
<b>8-10.mai</b>	Jørn Arild	rapportskriving - designdokument	12
<b>14-May</b>	Jørn Arild	rapportskriving - designdokument	3
	Kristoffer	rapportskriving	3
		rapportskriving, testfasen, konklusjon, diskusjon av resultater	7.5
<b>15-May</b>	Kristoffer		
	Jørn Arild	rapportskriving, design, implementering.	7.5
	Børge	rapportskriving, testfasen, konklusjon, diskusjon av resultater	5.5
<b>18-May</b>	Børge	Rapport, møte med Tom	7
	Jørn Arild	Rapport, møte med Tom	7
	Kristoffer	Rapport, møte med Tom	7
<b>19-May</b>	Børge	Rapport, Brukerveileding	7
	Jørn Arild	rapport	7
	Kristoffer	rapport	7
<b>20-May</b>	Jørn Arild	Rapportskriving	7
	Kristoffer	Rapportskriving	7
<b>Resten</b>	Børge		25
	Kristoffer		25
	Jørn Arild		25





## **Vedlegg J Utviklingsmodell**



Idéen til modellen er hentet fra :

*Kundeorienterte Web-systemer, av Werner Grini og Arne Sunde, Hovedsfagoppgave november 1996 Ved UIO, Side S 55*



**Vedlegg  
K  
Brukerveiledning**

**Brukermanual for SOFTCROSS**  
Softcross er utviklet av  
Jørn Arild Andenæs  
Kristoffer Jørdre  
Børge Johnsen

**Innledning:**

Dette er brukermanualen for softcross systemet. Softcross er kundebehandlingssystemet på nett som gjør det enklere for bedrifter å kommunisere med kunder.

**Hva trenger du for å bruke softcross systemet?**

Det eneste du trenger for å bruke systemet er internett-tilkobling og en nettleser.

Systemet er to delt. Du er enten bruker av systemet som Kunde eller Bedrift. Brukermanualen har derfor to kapitler. Kapittel 1 for Kunde og kapittel 2 for Bedrift.

## Kapittel1 1- KUNDENS BRUK AV SOFTCROSS

Registrering:

*Du har mottatt en e-post fra Bedriften du er kunde hos som ser slik ut:*

Hei

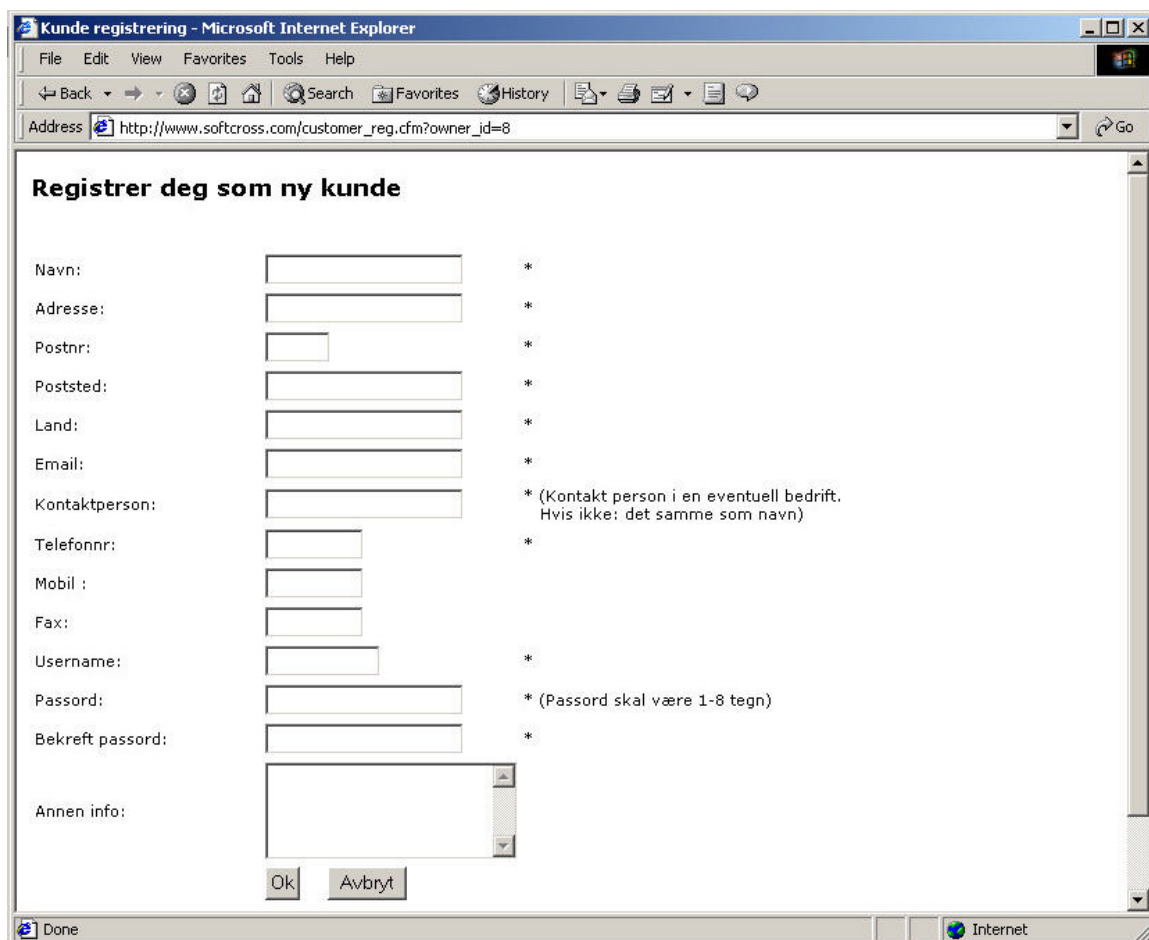
Vi har nå tatt i bruk et nytt kundebehandlingssystem; Softcross. Og vi vil gjerne at du som kunde av oss skal kunne bruke dette som en kontaktforbindele til oss.

Får å finne mer info eller registrere deg gå til [http://www.softcross.com/login/login.cfm?owner\\_id=8](http://www.softcross.com/login/login.cfm?owner_id=8)

Hilsen Børge

Webhouse.no

*Følg linken og du vil komme på en side der du kan veldge "registrer deg her" nederst. Velg dette og du vil komme til denne siden:*



The screenshot shows a Microsoft Internet Explorer browser window with the title 'Kunde registrering - Microsoft Internet Explorer'. The address bar contains 'http://www.softcross.com/customer\_reg.cfm?owner\_id=8'. The main content area displays a registration form with the following fields and labels:

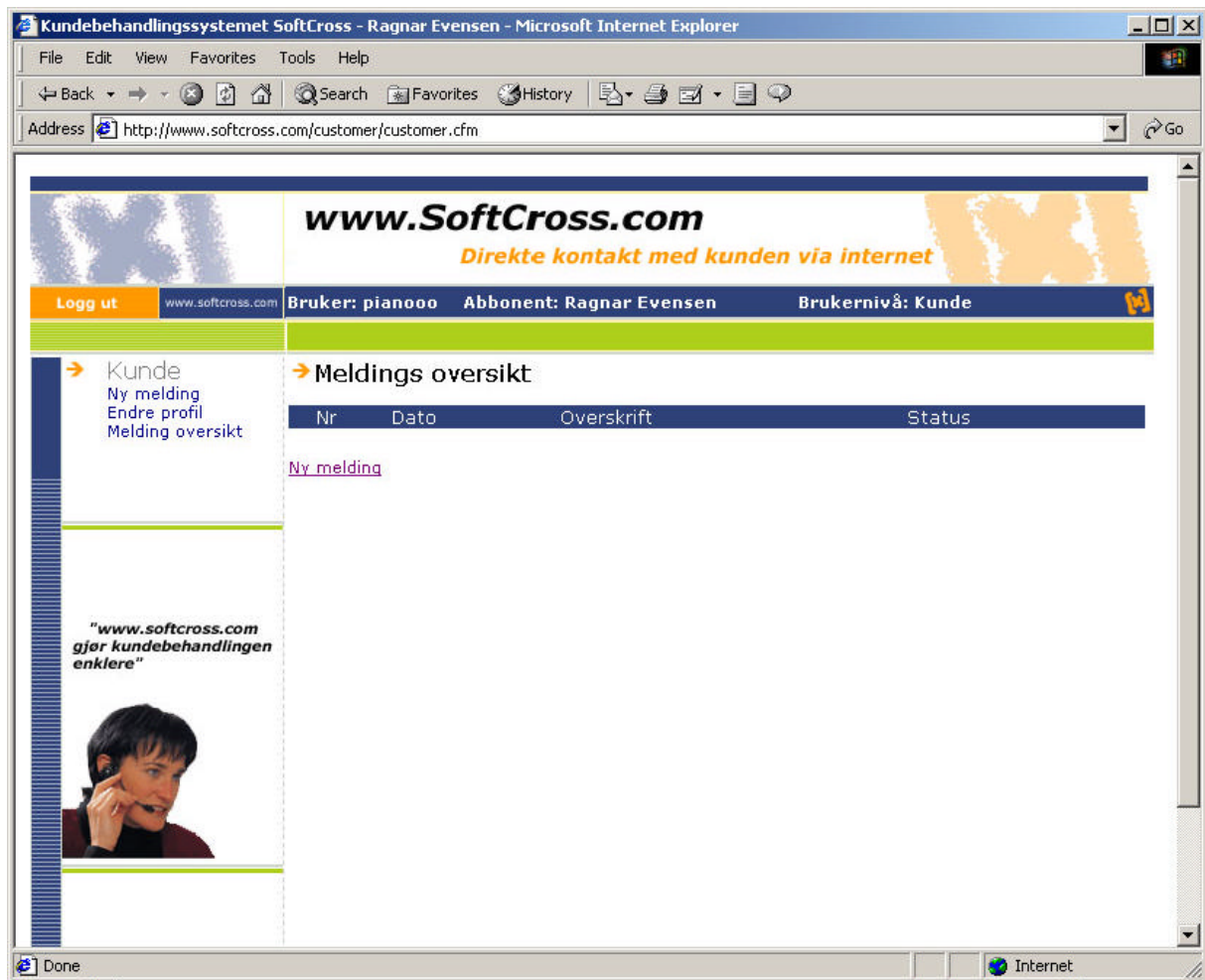
- Navn:  \*
- Adresse:  \*
- Postnr:  \*
- Poststed:  \*
- Land:  \*
- Email:  \*
- Kontaktperson:  \* (Kontakt person i en eventuell bedrift. Hvis ikke: det samme som navn)
- Telefonnr:  \*
- Mobil:
- Fax:
- Username:  \*
- Passord:  \* (Passord skal være 1-8 tegn)
- Bekreft passord:  \*
- Annen info:

At the bottom of the form are two buttons: 'Ok' and 'Avbryt'.

Navn er navnet på firmaet ditt og kontaktpersonen er deg.

***Feltene som er markert med \* må fylles inn for at du skal få registrert deg. Ta vare på brukernavnet og passordet ditt!!***

Fyll ut feltene og du kommer til denne siden:



Denne siden har 3 valg i menyen

Ny melding:

Trykk for å skrive ny melding til bedriften

Endre profil:

Trykk for å endre info om ditt firma/deg selv

Melding oversikt:

Her får du listet ut alle sakene du har mot bedriften du er kunde av.

## Kapittel 2 – Bedriften og Softcross


Som bedrift kan du registrere deg i Softcross-systemet og få kontroll over klager og forespørsler fra kunder på nett.

For å ta i bruk systemet må du registrere Bedriften i systemet. Her følger veiledning på hvordan du gjør dette. Lykke til!!

Gå inn på <http://softcross.com>

### Registrering

Trykk på "Registrer mitt firma i Softcross" og du kommer inn på denne siden:



Her fyller du ut Firmaopplysninger og info om deg som administrator. Den administratoren du registrerer her får tilgang til å opprette nye ansatte hos bedriften din og endre på layout og egendefinerte felter. **HUSK Å TA VARE PÅ brukernavn og passord!!!**

Feltene merket med \* my fylles ut!

Den avdelingen du som administrator skriver inn vil være den 1. registrerte avdelingen i bedriften. Du kan registrere flere senere.

Trkk på "send" og du kommer til denne siden

Opplysningene er lagret i databasen.

Du har blitt registrert som **administrator** av systemet softcross. For at du og andre personer i bedriften skal kunne bruke systemet må du registrere deg/dem som **brukere** av systemet. Du bør også konfigurere siden din for bruk. For å kunne gjøre dette trykker du på linken nedenfor.

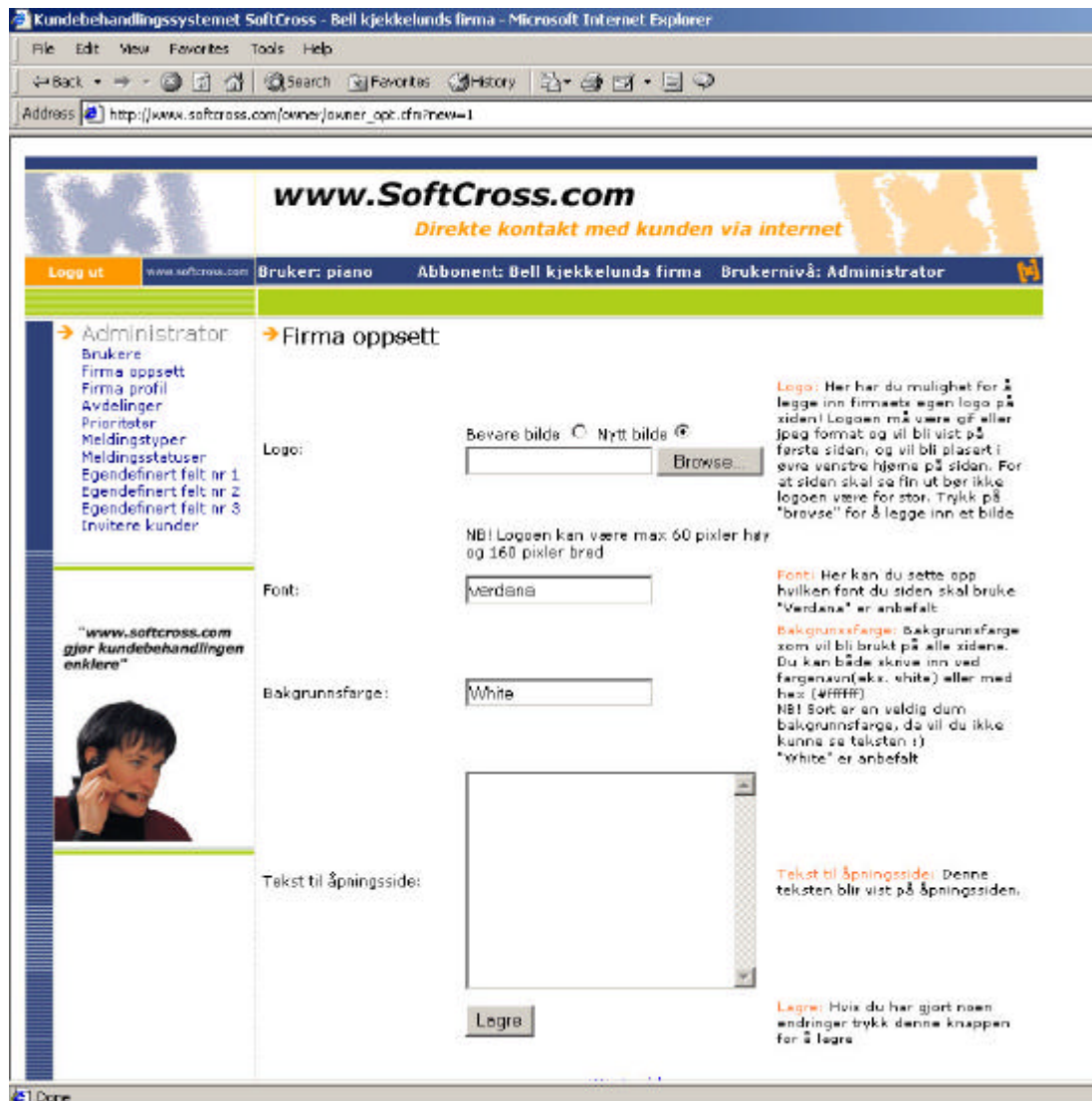
Denne linken fører deg inn i en guide som gjør dette automatisk for deg.

[Guide](#)

Du kan også gå rett inn på siden din og heller konfigurere valgene/opprette brukere senere

[Hovedsiden](#)

Du kan her velge å gå direkte til hovedsiden. Dette er ikke anbefalt, men brukes hvis du vil ha standard oppsett. Trykk på GUIDE. Og du kommer til denne siden:





Logo: velg browse og finn en logo lagret lokalt hos deg som du vil bruke på sidene.  
Her setter du også opp font, bakgrunnsfarge og tekst til åpningside.

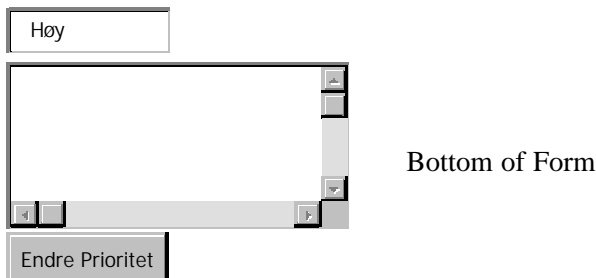
Trykk "Lagre" og velg "neste side →" og du kommer til denne siden:

## Prioriteter: *Endre*

Top of Form

Prioritetsnavn:

Prioritets beskrivelse:



### Prioritetstyper:

Høy	<a href="#">Endre</a> <a href="#">Slett</a>
Lav	<a href="#">Endre</a> <a href="#">Slett</a>
Middels	<a href="#">Endre</a> <a href="#">Slett</a>

### [Neste side -->](#)

Høy, middels og lav er prioriteter som du kan sette på en melding du får fra en kunde. Denne prioriteten er en intern prioritet som kunden ikke ser.

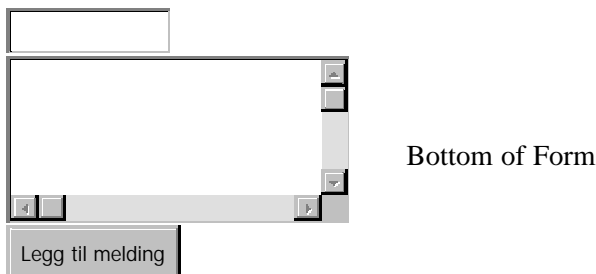
Trykk på "neste side →" og du kommer til denne siden

## Meldingstyper: *Legg til ny*

Top of Form

Meldingstypenavn:

Meldingstype beskrivelse:



### Meldingstyper:

Klage	<a href="#">Endre</a> <a href="#">Slett</a>
Spørsmål	<a href="#">Endre</a> <a href="#">Slett</a>

### [Neste side -->](#)

Klage og Spørsmål er standard meldingstyper.

Når en kunde sender inn en melding kan kunden velge mellom disse meldingstypene.

Du kan legge til nye og endre meldingstyper.

Trykk på "Neste side →" og du kommer til denne siden.

## Meldingstater: *Legg til ny*

Top of Form

Meldingsstatusnavn:

Meldingsstatus beskrivelse:



Bottom of Form

### Meldingsstater:

Behandles	<a href="#">Endre</a> <a href="#">Slett</a>
Ferdig	<a href="#">Endre</a> <a href="#">Slett</a>

[Neste side -->](#)

*Behandles og Ferdig er standard meldingstater.*

*Meldingsstatus er statusen du som Bedrift setter på en melding du får fra kunden. Kunden kan da se hvilken status du har satt og følge med på hvor i behandlingsprosessen Bedriften er.*

Du kan legge til, endre og slette meldingstater.

Trykk på "Neste side →" og du kommer til denne siden.

## Avdelinger: *Legg til ny*

Top of Form

Avdelingsnavn

Avdelings beskrivelse:



Bottom of Form

### Avdelinger:

piano	<a href="#">Endre</a> <a href="#">Slett</a>
-------	---

[Neste side -->](#)

Her legger du til, endrer og sletter avdelinger. Du MÅ ha en avdeling registrert for at systemet skal fungere.

*Når du mottar en melding kan du sette hvem avdeling som skal behandle denne.*

*Trykk på "Neste side -->" og du kommer til denne siden.*



**www.SoftCross.com**  
*Direkte kontakt med kunden via internet*

Logg ut [www.softcross.com](#) Bruker: piano Abbonent: Bell kjekkelunds firma Brukernivå: A

→ Administrator  
Brukere  
Firma oppsett  
Firma profil  
Avdelinger  
Prioriteter  
Meldingstyper  
Meldingsstatuser  
Egendefinert felt nr 1  
Egendefinert felt nr 2  
Egendefinert felt nr 3  
Invitere kunder

→ **Egendefinert felt nr 1**

Du har ikke opprettet noe egendefinert felt nr. 1, gjør det gjerne nå

Felt navn:

Første alternativ

[Neste side -->](#)

*"www.softcross.com  
gjør kundebehandlingen"*

Du kan sette opp egendefinerte felter som kunden skal kunne velge når han/hun skriver en melding. Har her som eksempel skrevet inn "pianostrenger" og "shimmel"

Trykk på lagre og du får opp denne siden for å skrive inn flere alternativer det egendefinerte feltet skal ha.



Logg ut [www.softcross.com](#) Bruker: piano Abbonent: Bell kjekkelunds firma Brukernivå: Administrator

→ Administrator  
Brukere  
Firma oppsett  
Firma profil  
Avdelinger  
Prioriteter  
Meldingstyper  
Meldingsstatuser  
Egendefinert felt nr 1  
Egendefinert felt nr 2  
Egendefinert felt nr 3  
Invitere kunder

→ **Egendefinert felt nr 1: pianostrenger** [Endre overskrift](#)

Skal feltet vises for kunden:  JA  NEI

Legg til nytt alternativ

**Alternativer:**

andre	<a href="#">Endre</a>	<a href="#">Slett</a>
shimmel	<a href="#">Endre</a>	<a href="#">Slett</a>

[Neste side -->](#)

*"www.softcross.com  
gjør kundebehandlingen  
enklere"*



Her kan du altså legge til så mange alternativer du vil ha. Disse alternativene vil komme i en dropdownboks på skjemaet der kunden fyller inn meldingen.

Du må også velge at feltet skal vises for kunden for at feltet skal bli satt til aktivt.

Trykk på "Neste side -->" og du kommer til 2 nye sider for å legge til egendefinerte felter. Du kan altså til sammen ha 3 egendefinerte felter som kunden kan velge ved registrering av melding.

Den neste siden deretter er:

## ➔ Brukere

	Navn	Avdeling	Brukernavn	Brukernivå
<a href="#">endre</a> <a href="#">slett</a>	Kjell bekkelund	piano	per	Administrator
<a href="#">Registrer ny ansatt</a>				

Den eneste brukeren du har til nå er administrator. Det eneste administrator kan gjøre er å endre på innstillinger og legge til nye ansatte.

**Du MÅ** registrere en ansatt som er "mottaker" denne ansatte kan motta og svare på meldinger fra kunden

Trykk på "registrer ny ansatt"

Feltene merket med \* er obligatoriske.

## Registrere ny ansatt

Top of Form

Navn	<input type="text"/>	*
Adresse	<input type="text"/>	*
Email	<input type="text"/>	*
Postnr	<input type="text"/>	*
Poststed	<input type="text"/>	*
Telefon nr	<input type="text"/>	
Mobil telefon	<input type="text"/>	
Brukernavn	<input type="text"/>	*
Passord	<input type="text"/>	*
Bekreft passord	<input type="text"/>	*
Avdelingå	piano <input type="text"/>	*
Brukernivå	Mottakere <input type="text"/>	*

**Litt om de forskjellige typen brukere:**

Mottaker: -når mottaker logger inn kommer han til siden med nye saker og kan fordele disse til de ulike avdelingene.

Bruker: -når bruker logger seg inn kommer han til siden med saker som tilhører avdelingen han jobber i. En bruker kan også se nye saker og fordele disse på avdelinger.

Administrator: -Setter opp design og registrerer/sletter/endrer ansatte.

Administrator: - Setter opp layout på sidene og legger til/endrer ansatte.

Trykk på "Neste side ->" og du er ferdig registrert som bruker av Softcross systemet!!

Du kan nå velge fra menyen til venstre for å endre og legge til nye poster i de overnevnte emnene

[Brukere](#)  
[Firma oppsett](#)  
[Firma profil](#)  
[Avdelinger](#)  
[Prioriteter](#)  
[Meldingstyper](#)  
[Meldingsstatuser](#)  
[Egendefinert felt nr 1](#)  
[Egendefinert felt nr 2](#)  
[Egendefinert felt nr 3](#)  
[Invitere kunder](#)

---

**"www.softcross.com  
gjør kundebehandlingen  
enkler"**

**Gratulerer!**

Du er nå ferdig å sette opp systemet, og det eneste som gjenstår er å legge til en link til systemet fra hjemmesiden til bedriften din og/eller sende ut innvitajoner til dine kunder. Du kan enten sende invitasjoner med vanlig mail, eller gå inn op siden "Invitere kunder"

## Bruk av systemet

Når du logger deg på får du opp oversikt over alle meldingene som har kommet fra kundene dine. Du kan svare på meldingen,e skjemaet for dette ser slik ut:

### Melding: tv

Top of Form

#### Kunde info:

Navn:	Trond Trondsen	Telefon:	61139392
Adresse:	vangsveien 34	Mobil:	
Poststed:	Vang	Fax:	
Email:	trond@tronsen.com	Kontaktperson:	Trond trondsen

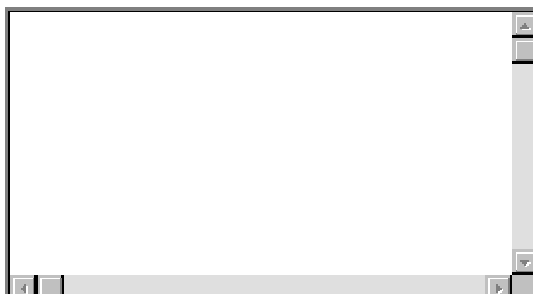
#### Meldings info:

Referansenr:	90
Meldingstype:	Klage
Avdeling:	øgle
Prioritet:	Ikke satt
Status:	behandles

Dato:	Melding:	Fra:
04.05.2001	Tv'en min fungerer ikke	Trond Trondsen
04.05.2001	Kan du si noe mere om hva som ikke fungerer?	Bell kjekkelund

Avslutt saken:  ja  Nei

Kommentar til saken:





**Vedlegg  
L  
Eksempel på ukeplan**



Ukeplan for SoftCross

Uke 8

Jørn	Diskutere datamodell Definere datamodell Legge inn datamodell i MySQL Sende mail til Sunde for å høre respons på kravspek Planlegge programmering. (sette opp skjermbilder) Rette opp kravspek
Børge	Diskutere datamodell Gjøre ferdig dokumentasjon om utviklingsmiljøet Definere datamodell Legge inn datamodell i MySQL Planlegge programmering. (sette opp skjermbilder) Rette opp kravspek
Kristoffer	Diskutere datamodell Ukeplan Definere datamodell Legge inn datamodell i MySQL Planlegge programmering. (sette opp skjermbilder) Rette opp kravspek