

dEkspSys
Anden delaflevering
Analysedokument og refleksion

Gruppe 3:
Jeppe Warberg Larsen - 20012571
Lars Vadstrup Hansen - 19993737
Bent Bisballe Jensen - 20001467

5. maj 2004

Indhold

1	Opgaven	3
1.1	Formål	3
1.2	Systemdefinitionen	3
1.3	Omgivelser	3
1.3.1	Problemområder	3
1.3.2	Anvendelsesområder	4
2	Problemområdet	4
2.1	Klynger	4
2.2	Struktur	5
2.3	Klasser	5
2.4	Hændelser	9
3	Anvendelsesområdet	9
3.1	Brug	9
3.2	Aktører	9
3.2.1	Ekspedient	9
3.2.2	Brugsmønstre	10
3.3	Funktioner	15
3.3.1	Komplet funktionsliste	15
3.3.2	Specifikation af funktioner	15
3.4	Brugergrænsefladen	15
3.4.1	Dialogform	15
3.4.2	Oversigt	20
4	Anbefalinger	21
4.1	IT-Systemets nytte og realiserbarhed	21
5	Reflektion	21
6	Referencer	24

1 Opgaven

1.1 Formål

Systemet skal assistere ekspedienterne i en mindre virksomhed, der sælger slagtøjsartikler. Salgsområdet; dels webbutikken og dels den fysiske butik. Der skal være en nem og overskuelig måde at holde styr på det omfangsrige lager (10.000 varenumre). Anvendelse af stregkoder bliver ikke taget med i betragtning. Systemet betragter ikke butikkens øvrige bogholderi, dvs. udgifter og indtægter som ikke har noget med slagtøjsvarer at gøre (ex. udgifter til el).

1.2 Systemdefinitionen

Med hjælp fra BATOFF kriterierne vil vi nu beskrive vores systemdefinition.

Betingelser	Brugerne af systemet skal have almindelig kendskab til arbejde med computere, dvs. være fortrolige med en filbrowser o. lign.
Anvendelsesområde	Butikken har et specialiseret varesortiment. Det skal assistere ekspedienterne i deres daglige arbejde ved at gøre det lettere, hurtigere og mere effektivt at arbejde med varelager, fakturering og tilbudsgivning. Der er brug for automatisering i forbindelse med indkøb og salg, således alle tal let kan samles i regnskabet. Herudover er der en række opgavelister, som ekspedienterne selv vedligeholder, der også kunne gøres tilgængelige i elektronisk form.
Teknologi	Standard PC, samt mindre server, internt netværk, samt internetforbindelse.
Objekter	Kunde, Vare, Varekatalog, Ordre, Lager, Kreditor, Indkøb, Ansæt, Opgave, Arbejdsplan, Forretning.
Funktionalitet	Støtte til drift af fysisk butik og online butik.
Filosofi	Gennem automatisering og systematisering vil vi minimere de menneskelige fejl og effektivisere de daglige procedurer i butikken.

1.3 Omgivelser

1.3.1 Problemområder

Lageret er uoverskueligt, der findes ikke noget online salg, der er ikke noget samlet kundekartotek, de mangler overblik over den daglige omsætning, der er ikke et gennemgående system, men derimod flere systemer (deraf en del som fungerer i kraft af papir og blyant) som ikke kan snakke sammen.

1.3.2 Anvendelsesområder

De ønsker et mere struktureret salgsapparat, for derved at lette deres arbejdsgang i hverdagen. En række af de problemer som systemet skal afhjælpe, er at overskueliggøre lager, være let at tilføje varer til lageret og samtidig holde styr på udgifterne forbundet med vareindkøb. Ved salg skal det være let at holde styr på indtægterne, samtidig skal systemet hjælpe dem med at holde styr på arbejdsopgaverne i processen at sælge en vare (rykke for betaling, bestille nye varer hjem, når de er udsolgt o.lign.). Systemet skal være integreret, så det gør det nemt at samarbejde med andre systemer (ex. deres revisors regnskabssystem). Systemet skal kunne snakke sammen med et onlinesalgssystem som vil blive beskrevet på et senere tidspunkt.

2 Problemområdet

2.1 Klynger

Der er 4 klynger i vores objektmodel: salg, køb, lager og arbejde. De er markeret ved de grå felter på klassediagrammet.

Salg

Salg repræsenterer alle indgående transaktioner i forretningen, dvs. kundekartoteket og ordresystemet. Denne klynge har kendskab til lager klyngen, med henblik på ordreoprettelse.

Køb

Køb repræsenterer alle udgående transaktioner i forretningen, dvs. kreditor-kartoteket, samt bestillingssystemet, som vi har valgt at kalde *indkøb*. Køb har naturligvis, ligesom salg, kendskab til lager.

Lager

Lager holder styr på eksisterende varer, hvad enten de findes fysisk i butikken (hvis der rent faktisk er indkøbt nogen, som endnu ikke er solgt), eller de blot eksisterer listet i et varekatalog over varer, som forretningen kan bestille hjem, og derfor potentielt har mulighed for at sælge til en kunde. Lageret indeholder derudover en række informationer omkring en given vare, analogt med den tykke mappe med varekataloger.

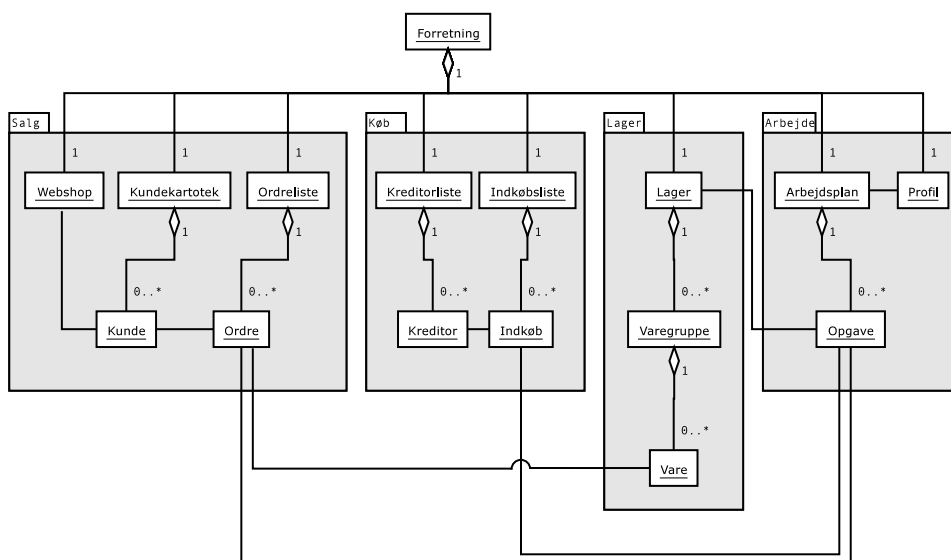
Arbejde

Arbejde indeholder informationer om igangværende processer i forretningen, samt kommende opgaver.

Der er her en repræsentation af de ansatte, samt en liste med opgaver, som

skal udføres og er blevet udført. Disse opgaver udgør tilsammen alt hvad der foregår i forretningens daglige arbejde, eksempelvis en bestilling, en betaling, et salg eller lign.

2.2 Struktur



Figur 2.1

Den overordnede objektstruktur er en træstruktur (forbundet med aggregeringer), idet forretningsklassen agerer rodknude med de fire klyngerstrukturer som undertræer. Hver klynge er igen en samling træer, (kundekartotek, ordreliste, kreditorliste, etc...) som igen har dataknuder/dataklasser under sig.

Disse er beskrevet vha. aggregeringer.

Der er en del kendskabsrelationer mellem klasser i forskellige klynger (eksempelvis mellem opgave i arbejdeklyngen og lager i lagerklyngen).

2.3 Klasser

Forretning

Forretningsklassen repræsenterer helheden. Den er den fysiske ramme omkring alt hvad der foregår i forbindelse med handel. Den har kendskab til alle transaktioner, på alle niveauer og kan på den måde skabe et overblik med henblik på statistik eller regnskab.

I den fysiske verden manifesteres forretningsklassen ved chefens skrivebord, med bunker af papirer, indeholdene informationer og rapporter over hvad der *er* foregået, *skal* foregå og *kan* foregå.

Webshop

Webshopklassen repræsenterer den hjemmeside hvorpå online kunder har mulighed for bestilling/køb af varer, som findes i lageret og er valgt til at figurere her (En del varer egner sig ikke til salg over nettet).

Kundekartotek

Kundekartoteksklassen er stedet hvor eksisterende kunder kan findes. Der er en række metoder stillet til rådighed til at søge efter dem, eksempelvis ved gennembladring eller en alfabetisk sortering, afhængig af repræsentationen. I den eksisterende forretning er kundekartoteket repræsenteret ved hhv. en minimalistisk database (Navision Concorde), samt et hav af gule post-it notes.

Kunde

Kundeklassen er den faktiske repræsentation af kunden, i form af navn, adresse, telefonnummer, etc.

Nogle kunder handler hyppigt i forretningen og skal derfor være nemme at finde frem til, hvorimod andre kunder kun handler een gang, og ikke udleverer personoplysninger i processen.

Eksempler på de to typer kunder:

Et eksempel på den første type er en skole som jævnligt køber nye dele til musiklokalets trommesæt. Her er en lang række informationer inkl. en eventuel rabat eller betalingsordning som skal huskes. Denne type kunde findes i den eksisterende forretning i den simple database.

Et eksempel på den anden type kunde er den mere supermarked-agtige kunde, som kommer ind i forretningen, finder sin(e) vare(r), går hen til kassen, betaler og forlader forretningen igen. Denne type kunde kræver ingen informationer overhovedet og kan systemmæssigt betragtes som en *John Doe* kunde.

Ordreliste

Ordrelisteklassen er en samling ordrer. Man har her mulighed for at danne sig et overblik over hvilke ordrer som er lavet hvornår.

Ordrelisten er i dag repræsenteret ved en bunke post-it notes sat på spyd (igangværende ordrer), samt en mappe med håndskrevne kvitinger (afsluttede ordrer).

Ordre

Ordreklassen repræsenterer en igangværende eller afsluttet handel.

Den knytter en kunde sammen med en mængde varer, samt eventuelle rabatter eller betalingsaftaler.

En ordre kan være i een af tre tilstande: *Tilbud*, som blot er en samling varer, med en tilknyttet pris (og eventuel rabat), *Ordre*, som er når kunden har sagt ja tak til et tilbud og sidst *faktura*, når kunden har modtaget varerne i ordren og betalt dem.

Kreditorliste

Kreditorlisteklassen er stedet hvor eksisterende kreditorer kan findes. Der er en række metoder stillet til rådighed til at søge efter dem, eksempelvis ved gennembladring eller en alfabetisk sortering, afhængig af repræsentationen. I den eksisterende forretning er kreditorkartoteket repræsenteret ved mappen med varekataloger, hvor hvert varekatalog viser hvilke varer der er tilknyttet hvilken distributør.

Kreditor

Kreditorklassen er den faktiske repræsentation af en kreditor. Denne har ligesom kunden en række attributer som oplyser om lokalisering og kontaktmuligheder. Derudover har en kreditor en række varenumre tilknyttet, som er den mængde varer, som den givne kreditor har mulighed for at importere. En kreditor er således en importør, distributør eller producent.

En kreditor udsender med mellemrum et varekatalog med de varer, som vedkommende udbyder, samt et opdateret priskatalog.

Indkøbsliste

Indkøbslisteklassen er en samling af indkøb. Der er her mulighed for at skabe sig et overblik over hvilke indkøb som har været foretaget hvornår, samt den samlede udgift inden for et givent tidsmæssigt interval (med henblik på regnskab og kasseopgørelse).

Indkøb

Indkøbsklassen repræsenterer en igangværende eller afsluttet handel med en kreditor. Den knytter en kreditor sammen med en mængde varer, samt eventuelle rabatter eller betalingsaftaler.

Et indkøb kan ligesom en ordre være i tre tilstande, som dog ikke håndteres af forretningen, førend tilstanden *faktura* er nået (Man må antage at der findes en lignende struktur hos distributøren).

Lager

Lagerklassen repræsenterer den overordnede lagerstruktur. Man kan her overskue hvor mange varer man har hjemme af en given slags, samt søge efter

varer, både fysisk (lokaliseringmæssigt) og informationsmæssigt. Lagerklassen udgør i dag to dele, nemlig hylderne og kasserne i forretningen, sammen med varekatalogets informationer.

Varegruppe

Varegruppeklassen er en lagerabstraktion som er nødvendig for nemmere at kunne finde en given vare. Derudover knytter den en række varer sammen, som har mange fællestræk. Et eksempel på en varegruppe kan være *storetrommeskind*. En varegruppe repræsenteres i dag via varens fysiske placering i butikken (den står et sted hvor der er flere varer af samme slags), samt i varekatalogerne, idet en ditributørs katalog og prisliste som regel kun indeholder een type varer.

En varegruppe kan være mere eller mindre specifik, der kan således være en varegruppe som hedder *trommeskind*, som så at sige indeholder *storetrommeskind*, en varegruppe kan således specialiseres, indtil man når den faktiske vare.

Vare

Vareklassen indeholder en mængde informationer om en given vare, den kan betragtes som det nederste blad i varegruppetræet. En vare har således en lang række fællesattributer med andre varer, hvilket knytter dem logisk sammen. Udover alle disse fællesattributer har en vare een eller flere attributer, som gør den unik. En vare har desuden en række attributer, som er nødvendige for at kunne handle med den (Et antal, en indkøbspris, en pris o. lign.). En vare repræsenteres i dag i kraft af dens fysiske tilstedeværelse eller figurerende på en linie i et varekatalog (evt. med billede).

Arbejdsplan

Arbejdsplanklassen er en overordnet abstraktion af opgaver. Arbejdsplanen er sorteret efter prioritet og indeholder opgaver som skal løses, eller er blevet løst, af en medarbejder.

Opgave

Opgaveklassen er en repræsentation af en opgave som skal løses. En opgave vil typisk være knyttet til en ansat i forretningen, som i objektmodellen er repræsenteret vha. en profil. Opgaven (ligesom opgavelisten) findes i dag kun i hovederne på de ansatte, samt i en lille udstrækning i form af post-it notes, med memoer om opgaver som skal udføres.

Profil

Profilklassen er en abstraktion over den ansattes hukommelse og arbejdsområde. En ansat har således et kompetenceområde, som gør vedkommende i stand til at udføre een eller flere typer opgaver.

2.4 Hændelser

Den følgende liste viser de overordnede hændelser.

Alle samlingsknuderne er undladt, idet de ikke indgår direkte i hændelserne.

Hændelse	Kunde	Ordre	Kreditor	Indkøb	Lager	Opgave	Profil
Vare salg	X	X			X		X
Vare forespørgsel	X				X		X
Vareforsendelse	X	X			X	X	X
Varebestilling			X	X		X	X
Varemodtagelse			X	X	X	X	X
Tilbudsgivning	X	X			X		X
Registrer kunde	X						X
Registrer kreditor			X				X
Ansættelse af medarbejder							X
Opsigelse af medarbejder							X

3 Anvendelsesområdet

3.1 Brug

Vi har foreløbig to aktører, nemlig ekspedienter og web-kunder, hvoraf vi ikke har analyseret web-kunder endnu.

Vi har 31 brugsmønstre fastlagt ud fra de første tre møder vi har haft med vores kunde.

3.2 Aktører

3.2.1 Ekspedient

Formål: En person, der udfører salg, tilbudsgivning, fakturering, statistik samt lageropdatering.

Karakteristik: Der er en varierende grad af erfaring med brug af IT blandt ekspedienterne.

Eksempler:

- Lars (ekspedient i Music Aid) der har stor erfaring med brug af IT og stor forståelse for IT koncepter generelt.
- Ad hoc arbejdskraft der skal træde til v. sygdom, ferie o.lign., kan ikke forventes at have samme erfaring som de fastansatte.

3.2.2 Brugsmønstre

Nedenstående brugsmønstre (i tekstuel form) er lavet i samarbejde med Lars fra Music Aid. Vi anvendte flere forskellige metoder bl.a. interviews, observation og mockups. Således har vi både benyttet nogle af vejledningerne fra [Mathiasen et al], men også anvendt yderligere teknikker til bestemmelse af brugsmønstre.

FØLGENDE ER BEDRE I REFLEKTION Observation af ekspedienten gav os et indblik i, at der kommer til at ske nogle ændringer i, hvordan Lars bevæger sig i butikken i dag og hvordan han kommer til at arbejde frem over. Fx, skal han i dag hen til det fysiske lager for at afgøre om en bestemt vare er på lager, hvor han fremover både kan gå hen til lageret og lede efter varen, men også gå hen til systemet og søge efter den pågældende vare. Gennem interview og samtale med Lars fik vi ham til at sætte ord på hvordan han gerne ville arbejde. Ud fra det kendskab han har til de nuværende arbejdsopgaver, havde han en idé om hvad der ville gøre hans arbejde i dagligdagen nemmere. Ved at lade Lars prøve vores prototype af, fik vi ham til at specificere brugsmønstrene yderligere omkring oprettelse af varer, fakturaer/tilbud samt opgave / arbejdsplan områderne. Det vi kan se på vores nuværende status af tingene er, at vi ikke kan nå at få lavet en mere detaljeret analyse af hvordan Lars vil arbejde med Opgave / Arbejdsplan - til gengæld har vi fået rigtig god respons i form af vores seneste møde, hvor Lars for første gang benyttede vores prototype af vare / katalog sektionen af programmet. Ydermere fik Lars sat flere ord på den funktionalitet han ønsker i forbindelse med fakturaer og tilbud via varekataloget. Han fik også slået fast at han ønskede hele tre indgange til de mest brugte funktioner. Nemlig både i context-sensitive popup menuer, toolbars samt i top-menuer. Vi fik også Lars til at indse hvor meget strukturen på et website der indeholder varekataloget vil afspejle den struktur han opbygger via administrationen. Vi fandt også ud af at Lars ville have så få dialog-vinduer, som muligt for at gøre det mere enkelt og hurtigt at arbejde med de forskellige daglige opgave der skal løses.

Indkøb

Oprettelse af bestilling af varer hos leverandører / Opfølgning på modtagelse af varer / Redigering af bestilling, hvis varer udgået eller i restordre.

Disse mønstre skal assistere til at dels holde styr på udgifter i forbindelse med indkøb, samt sørge for at korrespondencen mellem virtuelt lager og fy-

sik lagerbeholdning forbliver opretholdt. Ved en oprettelse, vil man måske vente med at indtaste pris på bestilte varer, til de faktisk er modtaget. Når de er modtaget vil lageret blive opdateret. Der skal være mulighed for at afslutte dele af indkøbet dvs. hvis en vare er i restordre, så kan man nøjes med at lave lageropdatering på de varer der faktisk kom ind ad døren og vente med at opdatere resten indtil de faktisk kommer på lager.

Kreditor

Oprettelse: Her skal man kunne indtaste information omkring en kreditor (leverandør).

Redigering: Mulighed for at redigere en kreditors information

Slette: Slet en kreditor (kun muligt, hvis der ikke er indkøb knyttet til kreditor).

Disse mønstre sikrer at man har et overblik over sine kreditorer (leverandører) og at man kan differentiere de forskellige indkøb. Desuden er det her man angiver det antal der er bestilt - man kan ikke redigere antal af varer direkte på den enkelte vare. Dette skal sikre at der er god dokumentation af dels indkøb og dels krediteringer på defekte og returnerede varer.

Opdatering af varekatalog - nyt produkt fra bunden

Vores aktør (ekspedienten) tager følgende skridt for at oprette et nyt produkt fra bunden: Først oprettes eller vælges varekategori samt evt. tilføjes attributter. Dernæst vælges producent evt. opretter en ny producent. Produktkoden indtastes. Herefter kommer særlige attributter for netop denne vare i denne kategori. Endeligt skal ekspedienten indtaste information omkring de bestilte varer. Disse er: antal, salgspris, betalingsbetingelser, indkøbspris og leveringsdato.

Opdatering af varekatalog - nyt produkt fra andet lignende produkt

For at komme lidt hurtigere gennem oprettelsesprocessen kan vores ekspedient vælge at kopiere et eksisterende produkt for derefter at lave passende tilretninger. Først vælges et lignende produkt (evt. i samme kategori - dette er ikke noget krav). Dernæst kopieres produktet og rettes til, så det har de korrekte specifikationer. Endeligt skal ekspedienten indtaste information omkring de bestilte varer. Disse er: antal, salgspris, betalingsbetingelser, indkøbspris og leveringsdato.

Opdatering af varekatalog - eksisterende produkt fra kreditormodul

For at få varer hjem, der allerede er oprettet, skal ekspedienten blot vælge den eller de varer, han ønsker at bestille. Endeligt skal ekspedienten indtaste information omkring de bestilte varer. Disse er: antal, salgspris, betalingsbetingelser, indkøbspris og leveringsdato.

Alle ovenstående mønstre har indirekte med oprettelse og redigering af varer at gøre. Dermed har de også med administration af foldere der anvendes til logisk at inddеле de enkelte varegrupper. Der er ligeledes mønstre for oprettelse af foldere og varer samt redigering af de enkelte attributter der indgår i folderne og varerne. De ser ud, som følger:

Opdatering af varekatalog - Oprettelse af vare

Vores aktør (ekspedienten) tager følgende skridt for at oprette et nyt produkt fra bunden: Først oprettes eller vælges varekategori, hvorefter han vælger producent evt. opretter ny. Han indtaster dernæst produktkode samt særlige attributter for netop denne vare i denne kategori.

Opdatering af varekatalog - nyt produkt fra andet lignende produkt

For at komme lidt hurtigere gennem oprettelsesprocessen kan vores ekspedient vælge at kopiere et eksisterende produkt for derefter at lave passende tilretninger. Han begynder med at vælge lignende produkt (evt i samme kategori - dette er ikke noget krav). Så kopierer han produktet og laver passende tilretninger på produktet.

Opdatering af varekatalog - ny folder

Her skal ekspedienten først indtaste foldernavn, hvorefter han opretter relevante attributter der er specielle for varer i denne folder.

Opdatering af varekatalog - rediger folder

Her kan ekspedienten redigere foldernavn samt oprette og slette attributter der er specielle for varer i denne folder. Man kan ikke fjerne globale attributter i en underfolder - så skal man gå op i foldertræet og finde den folder der "ejer"attributterne. Der er også nogle helt faste attributter, som pris og produktkode der er systemattributter og derfor ikke kan fjernes af vores ekspedient (hvilket han i realiteten nok heller aldrig kunne ønske sig at gøre).

Opdatering af varekatalog - slet folder

Her kan ekspedienten slette en folder.

Salg

Oprette | Redigere | Slette (Tilbud, Faktura).

Der er mulighed for at oprette både tilbud, som ikke figurerer som salg før de har ændret status til faktura. Fra salgsmodulet skal man ligeledes kunne udskrive en ordrebekræftelse, som ikke er andet end et dokument der ligner en faktura på nær at overskriften er ordrebekræftelse. Det er ikke altid muligt eller nødvendigt at notere en kundes navn på en faktura, som fx ved kontant salg i butikken. Derfor skal der være mulighed for at udelade kundenavn ved fakturaoprettelse.

Herunder præsenteres et par konkrete eksempler på mere detaljerede brugsmønstre.

Mønster for oprettelse af faktura / tilbud:

Her skal ekspedienten enten via Drag and Drop tage en eller flere varer fra kataloget og tilføje til fakturaen. Dernæst vælger han om det skal være et tilbud eller en faktura. Han enten opretter eller finder kunden blandt eksisterende kunder. Ekspedienten har også mulighed for at give rabatter enten pr vare eller på det samlede tilbud. Her skal man enten give et tal for rabatten eller en procentsats. Ligemeget hvad man gør, skal systemet selv regne den anden form ud. Dvs. hvis ekspedienten indtaster en procentsats skal systemet regne rabatten ud i kroner og ører og vice versa. Til sidst kan ekspedienten vælge betalingsform - kontant, kredit x dage osv.

Mønster for reparationer

Her opretter vores ekspedient en speciel faktura med mulighed for ekstra meget tekst til beskrivelse af hvad der faktisk er foretaget under reparationen. Timelønnen trækker ekspedienten ind på fakturaen - det er oprettet, som et varenummer. Kommentarerne skal løbende kunne skrives til, efterhånden, som arbejdet skrider frem.

Ved fakturaer skelnes mellem kontant faktura og faktura med betalingsfrist.

Kunde

Oprette | Redigere | Slette (Kun muligt hvis kunden ikke har foretaget bestillinger).

Disse mønstre sikrer at vi kan opretholde et kunderegister, over kunder der ofte handler i butikken.

Opgave

Modtage opgaver fra systemet / Oprette opgaver / Slette opgaver.

Disse mønstre skal assistere ekspedienterne i deres daglige arbejde fx med at holde lageret fyldt og bestille varer hjem. De skal også fungere som en slags huskeliste, hvor ekspedienten kan oprette opgaver til sig selv eller en anden ekspedient.

Arbejdsplan

Lav oversigt over opgaverne samt indgang til at håndtere de enkelte opgaver.

Ansæt

Oprette / Slette / Redigere / Log på / Log af.

Disse mønstre skal sikre at vi dels kan administrere brugere af systemet og holde styr på hvilken bruger der har adgang til hvad, samt hvad den enkelte bruger af systemet foretager sig i systemet - fx foretager et salg eller indkøber varer og opdaterer lagerstatus.

Statistik

Følgende brugsmønster er kun et af mange eksempler på statistikker der kan trækkes fra systemet, men det har været et væsentligt punkt for vores kunde, derfor er det taget med.

Kasseopgørelse ved fyraften

Her kan man fx lave export af data til Navision Concorde, samt se statistikker over dagens indtægter og udgifter samt netto fortjeneste.

3.3 Funktioner

Nedenstående funktioner er udarbejdet på baggrund af ovenstående liste af brugsmønstre.

3.3.1 Komplet funktionsliste

Beskrivelse	Objekt	Kompleksitet	Type
Opret,	Indkøb	Medium	Opdatering
Rediger	Indkøb	Medium	Opdatering
Fjern bestilling hos leverandør	Indkøb	Medium	Opdatering
Opret	Kreditor	Simpel	Opdatering
Rediger	Kreditor	Simpel	Opdatering
Fjern kreditor	Kreditor	Simpel	Opdatering
Opret,	Salg	Medium	Opdatering
Rediger	Salg	Medium	Opdatering
Fjern tilbud/faktura	Salg	Medium	Opdatering
Opret	Kunde	Simpel	Opdatering
Rediger	Kunde	Simpel	Opdatering
Fjern kunde	Kunde	Simpel	Opdatering
Tilføj	Vare	Medium	Opdatering
Rediger	Vare	Medium	Opdatering
Fjern vare	Vare	Medium	Opdatering
Tilføj	Folder	Medium	Opdatering
Rediger	Folder	Medium	Opdatering
Fjern folder	Folder	Medium	Opdatering
Tilføj	Opgave	Kompleks	Opdatering
Fjern opgave	Opgave	Kompleks	Opdatering
List opgaver evt. filtreret efter forskellige kriterier valgt af ekspedienten	Arbejdsplan	Medium	Aflæsning / Beregning
Opret	Ansæt	Simpel	Opdatering
Rediger	Ansæt	Simpel	Opdatering
Fjern	Ansæt	Simpel	Opdatering
Log på, Log af		Medium	Aflæsning / Beregning
Skriv hændelse for bruger X på objekt Y	Log	Medium	Opdatering

3.3.2 Specifikation af funktioner

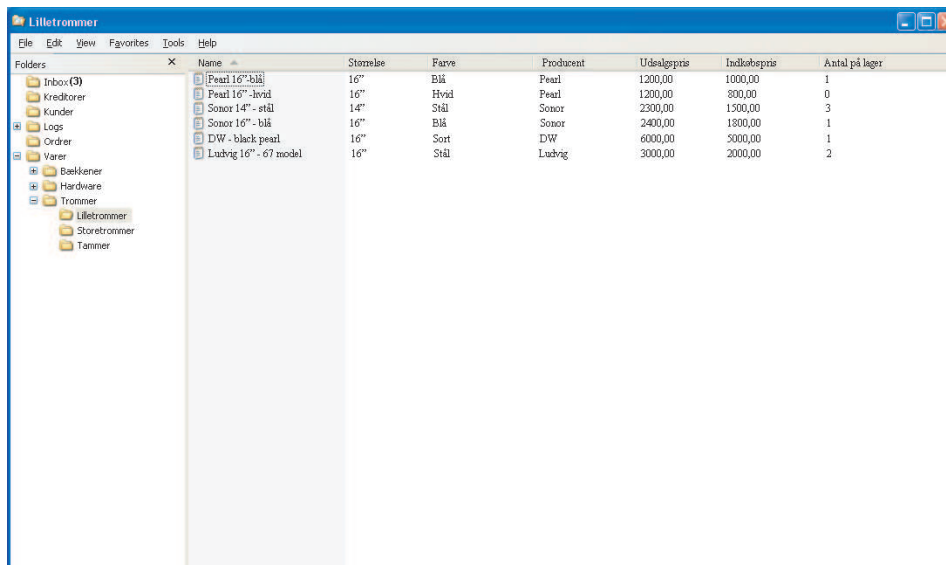
Den komplekse funktion "Tilføj/fjern opgave", skal knytte en opgave til et bestemt objekt og dermed muliggøre et direkte link til den pågældende opgave der skal udføres. dvs ekspedienten let skal kunne finde det sted i systemet, hvor den pågældende opgave skal findes. Næste skridt er så at vi automatisk vil detektere, hvornår opgaven er udført. Derfor skal de enkelte objekter der bliver tilknyttet en opgave selv sige til opgavemodulet, hvornår opgaven er udført. Herefter skal opgaven automatisk fjernes fra brugerens inbox.

3.4 Brugergrænsefladen

3.4.1 Dialogform

Brugergrænsefladen i vores system er inspireret programmer til filhåndtering og e-mails. Vi har således valgt dialogmønstre som er velkendte fra moderne

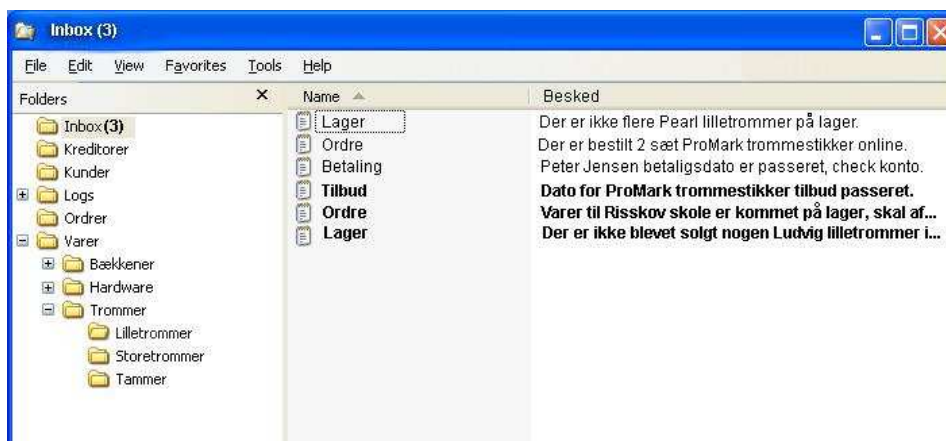
operativsystemer; menuvalg, skemaudfyldelse og direkte manipulation. Følgende billede er fra vores mockup, skabt vha. billedmanipulation.



Figur 3.1

Navigationen i programmet tager udgangspunkt i en mappestruktur, som vist på figur 3.1. Skærbilledet er delt i to, hvor vinduet til venstre viser en række mapper. Ved klik på en mappe udfoldes eventuelle undermapper, og mappens indhold vises i højre vindue. Dette indhold vil for alle mappers vedkommende vises som lister af "filer". Menubaren øverst vil dynamisk ændres til at fremvise de menupunkter der er mulige for den pågældende mappe.

Der er seks mapper:



Figur 3.2

Inbox

Vi har valgt en email-metafor for systemets kommunikation til brugeren. Således afspejler den klassetklyngen "Arbejde". I inboxen ligger beskeder om hændelser og tilstande i systemet, som brugeren skal bekendtgøres med og tage stilling til. På figur 3.2 ses et eksemplificeret indhold i vores mock-up. En besked er dels kendetegnet ved en titel som angiver hvilken type information der er tale om, eksempelvis ordrer, og dels en kort tekst der fortæller hvilke objekter der er involveret og hvorledes. Ved åbning af beskeden skal der via linkning kunne springes til et relevant sted i systemet, eksempelvis en bestemt varegruppe.

Inbox skal have diverse e-mail funktionaliteter som markering af læst/ulæst, arkivér m.m.

Kreditorer

Denne mappe svarer til objektet "Kreditorliste". Her ses en liste over virksomheder som butikken foretager indkøb hos. På samme måde som en adressebog.

Kunder

Listen over registrerede kunder, som objektet "Kundekartotek". Ligeledes som en adressebog.

Logs

Systemet skal løbende bogføre alle hændelser med konsekvenser. Som det ses i objektetdiagrammet, har vi endnu ikke taget stilling til grænsefladedesign af et sådant log-system.

Ordre

Mappen Ordre, som afspejler "Ordreliste", skal vise en liste over alle aktuelle ordrer i systemet. En ordre er kendetegnet ved en række attributter.

Varer

I vores prototype har vi koncentreret os om varelageret, som er essentielt i vores system, j.f. klassediagram og brugsmønstre.

Ved klik på Varer udfoldes mappetræet med varegrupperne. De forskellige varetyper har tilknyttet tekniske data, som er hensigtsmæssige at angive som attributter, frem for blot at indgå i den tekstuelle beskrivelse af produktet. Et eksempel er et trommeskind hvis diameter angives i tommer. 5 af attributterne er statiske og globale og findes i alle varegrupper. Det drejer sig om Navn, Varegruppe, Pris, Indkøbspris og Antal.

The screenshot shows the 'StoreAid v1.0' application window. On the left is a tree view under 'Varer' with categories like Skind, Trommestikker, Trommer, Bongo, Storetrommer, Tom, Snare, Cymbals, Crash, China, Ride, and Effekt. The 'Trommer' category is selected. On the right is a table with columns: Navn, Varegruppe, Pris, Indk. Pris, Antal, ProdVare..., Grossist, and Producent. The table lists various drum products and their details.

Navn	Varegruppe	Pris	Indk. Pris	Antal	ProdVare...	Grossist	Producent
Starclassi...	Storetrom...	0	0	0	SCP-BD-...		Tama
Paiste Po...	China	398	169	0	712001p...		Paiste
HHX Evol...	Crash	2100	1250	0	sc367803		Sabian
Rockstar ...	Storetrom...	0	0	0	RSC-BD-...		Tama
Deep Forest	Storetrom...	0	0	0	MPX-BD-...		Mapex
Rockstar ...	Storetrom...	0	0	0	RSCEFX-B...		Tama
Paiste	Effekt	250	104	0	702001p...		Paiste
Zildjian	Crash	1600	750	0	z04cr18		Zildjian
Zildjian A...	Crash	2590	1394	0	1820517		Zildjian
V Series	Storetrom...	0	0	0	MPX-BD-...		Mapex
Swingstar ...	Storetrom...	0	0	0	SSC-BD-BLS		Tama
Saturn Pro	Storetrom...	0	0	0	MPX-BD-...		Mapex
Rockstar	Storetrom...	0	0	0	RS-BD-BRG		Tama
Pro M Seri...	Storetrom...	0	0	0	MPX-BD-...		Mapex
Starclassi...	Storetrom...	0	0	0	SCM-BD-...		Tama
Meinl Clas...	Ride	1795	890	0	me0003ri...		Meinl
Rerno	Skind	155	47	0	rem-011...		Slagtøjcen...
Sabian	Crash	1498	520	0	sc234502		Sabian
M Series	Storetrom...	0	0	0	MPX-BD-		Mapex

Figur 3.3 (NB: Der skal ikke skelnes mellem mappe/fil-ikoner)

Figur 3.3 er et screenshot af prototypen. Den markerede mappe er her den generelle Varer, som har fået 3 ekstra attributer. Visningen er rekursiv, dvs. alle varer i undermapperne vises, og det kun med den aktuelle mappes attributer, som har mening for alle underliggende varer. Derfor angives Varegruppe, som henviser til den meste specifikke undermappe. For eksempel er den øverste vare på figuren "Storetrommer", som, lidt indforstået, hører ind under "Trommer".

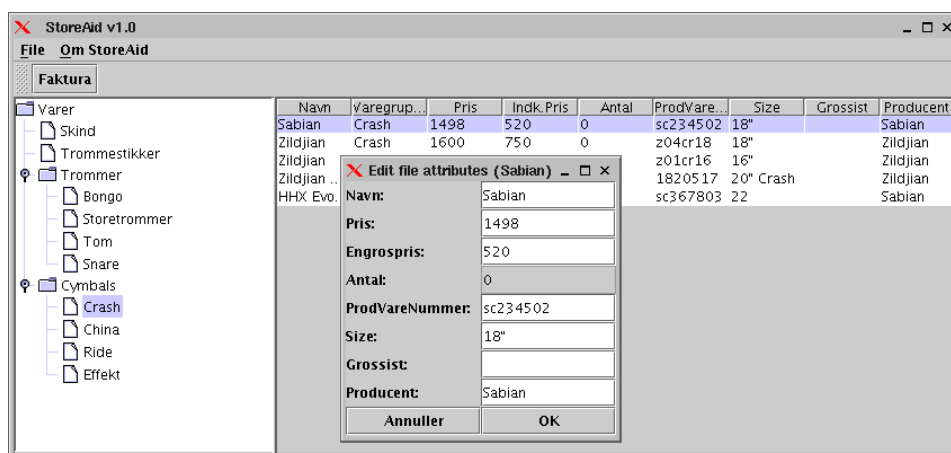
Attributernes navne vises øverst i vinduet, og kan således bruges som en nøgle til hurtig sortering af en varegruppe, på samme måde som man i et filsystem kan vælge at vise mappendhold sorteret alfabetisk eller efter filstørrelse m.m.

The screenshot shows the 'StoreAid v1.0' application window. The 'Cymbals' category is selected in the tree view. The table on the right displays cymbal products with columns: Navn, Varegrup..., Pris, Indk. Pris, Antal, ProdVare..., Size, Grossist, and Producent.

Navn	Varegrup...	Pris	Indk. Pris	Antal	ProdVare...	Size	Grossist	Producent
Sabian	Crash	1498	520	0	sc234502	18"		Sabian
Paiste Po...	China	398	169	0	712001...	18"		Paiste
Meinl Ra...	Ride	1695	840	0	me0003...	20"		Meinl
Paiste	Effekt	250	104	0	702001...	12"		Paiste
Zildjian	Crash	1600	750	0	z04cr18	18"		Zildjian
Zildjian ...	Crash	2590	1394	0	1820517	20" Crash		Zildjian
Zildjian	Crash	1500	450	0	z01cr16	16"		Zildjian
HHX Evol...	Crash	2100	1250	0	sc367803	22		Sabian
Meinl Cla...	Ride	1795	890	0	me0003...	22"		Meinl
Istanbul ...	Effekt	350	120	0	i02ef04...	10"		Istanbul
Stagg - S...	Effekt	198	62	0	st200208	8"		Stagg

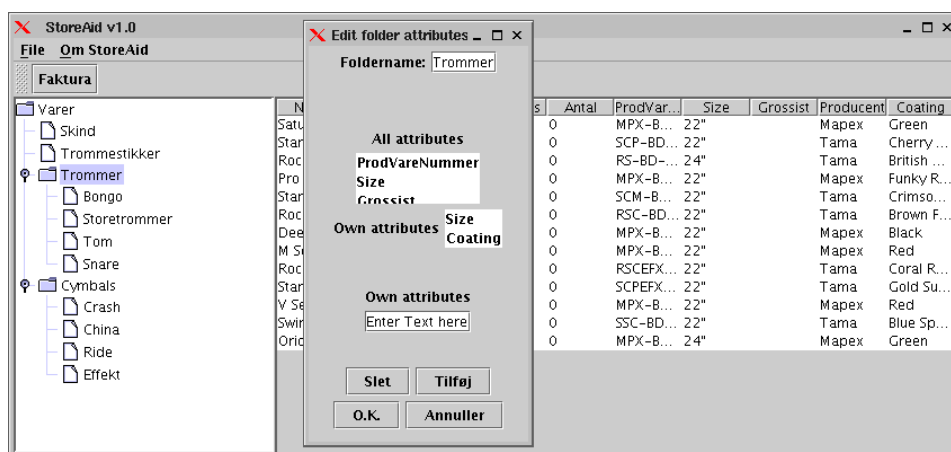
Figur 3.4

På figur 3.4 er der klikket på Cymbals (bækkener). Denne undergruppe er udvidet med attributen "Size".



Figur 3.5

På figur 3.5 er der højreklikket på en bestemt vare og valgt "Rediger vare". En dialogboks popper op, og de forskellige attributværdier kan ændres ved direkte indtastning. Dog ikke "Antal" som kun kan ændres ved salg/indkøb.



Figur 3.6

En væsentlig overvejelse omkring brugerfladen er muligheden for manipulation af indholdet i Varer. Det færdige system kan leveres med en varekategorisering der afspejler det nuværende sortiment, men det skal være muligt og nemt at ændre dette. På figur 3.6 er der højreklikket på Trommer og valgt "Rediger mappe". Under "Alle attributer" vises nedarvede + egne attributer. Ved at indtaste og "Tilføj" udvider man attributmængden i forhold til den ovenstående mappe. Dette påvirker således også mappens undermapper. Man kan slette en attribut ved at markere den og klikke "Slet". Dette bør gøres med omtanke. Ændringerne træder i kraft, når der klikkes "Ok".

En ny undermappe skabes ved at højreklikke på en mappe og vælge "Ny mappe". Dialogboksen er den samme som for "Rediger mappe".

Tilføjelser

Brugerfladen skal have en lang række tilføjelser i forhold til det der er beskrevet ovenfor:

- En tilbudsmappe i roden af træet, som man så kunne trække varer over i, for at sætte dem på tilbud.
- Der skal være et søgefelt i toolbaren, som virker på den måde, at man kan indskrænke de viste elementer i filvinduet, udfra et regulært udtryk.
- Der skal overvejes en måde at bestille nye varer hjem på, evt. gennem kreditor, evt. direkte på varen i lageret.
- Der skal være bruger-login med password, når klienten opstartes. Når man først er logget på, skal man have mulighed for at skifte bruger, uden nogen som helst form for besvær.
- Der skal være en toolbar, som afspejler de mulige funktioner i den givne kontekst. Disse skal kunne tilpasses som i f.eks. Word.
- Når man er i mappen Ordre, skal man kunne vælge "Ny ordre". I et vindue/dialogboks kan man indtaste informationer om handlen og angive kunde, gerne valgt fra kundedatabase. Ved at vælge "Tilføj varer" får man mulighed for at vælge varer fra vare-mapperne.
- Inbox'en skal angive om den enkelte opgave er udført, og i så fald skal beskeden kunne slettes.

Efter MusicAid blev præsenteret for den første prototype, blev følgende bemærket:

- Popup dialogbokse opleves hurtigt som irriterende. Det foreslås at de vises i et fast vindue under fileview'et.
- Der ønskes alternativer til højreklikket. På toolbaren kan der sættes knapper for de hyppige handlinger.
- Fleksibel omstrukturering af varer. Opret hurtigt en ny vare, læg den på et generelt niveau, og placér den specifikt på et senere tidspunkt. Markér en eller flere varer og drag-drop dem til en anden mappe. Flyt hele mapper.
- Fed skrift i de attributkolonner som kunden finder vigtigst.
- Ændring af vareattributer ved multipel markering. Eksempel: ny leverandør af varemærket Zildjian -> alle Zildjian-produkter markeres og redigeres samtidig.
- Hemmeliggørelse af data, først og fremmest indkøbspris, ved visning af skærm i en kundesituation.

3.4.2 Oversigt

Navigationen i et filsystem er dynamisk, og derfor er der ikke medtaget et navigationsdiagram med skærm-til-skærm billeder. Når der ikke aktive dialogbokse på skærmen, kan man som udgangspunkt navigere i mapperne vilkårligt.

4 anbefalinger

4.1 IT-Systemets nytte og realiserbarhed

Systemet her er kun nyttigt for MusicAid såfremt det overgår de eksisterende værktøjer, og er i stand til at integrere de mange funktionaliteter. Hvis systemet ikke indfrier de høje krav til fleksibilitet og hurtighed, vil man i mange arbejdssituationer gøre tingene på "den gamle måde" og derved svigte hele konceptet i elektronisk lagerstyring.

MusicAid har ambitioner om internet-handel, og dertil anser vi elektronisk lagerstyring for en nødvendighed. Ideen er netop, at systemet rummer både butik- og internet-handel, så man undgår parallelle systemer.

Et fuldt udviklet system vil desuden kunne lette arbejdet med regnskab og årlig revision.

At systemet er realiserbart, må antages ud fra at mange lignende detailforretninger anvender IT til lagerstyring og e-handel, heriblandt nogle af MusicAid's konkurrenter.

Hardware-aspektet i systemet bør ikke give anledning til hindringer, idet vi antager, at det kan køre på en standard PC.

De økonomiske rammer for en realisering af projektet har vi ikke overvejet. Med systemets mange funktioner og kompleksitet, må man forvente at der ligger mange timers arbejde i programmeringen. Til gengæld kan vi håbe på små udgifter til software, licenser o.l. da vi i høj grad kan benytte ikke-kommercielle værktøjer.

5 Refleksion

6. februar regnede vi med at have et første møde med slagtojsforretningen DrumCity. Til det interview ville vi støtte os til kapitel 1 i OOA&D. Vi ville da spørge til butikkens daglige rutine, og notere os hvordan deres nuværende teknik anvendes. Det ville lede os videre til kundens beskrivelse af problem- og anvendelsesområder, og hvilke generelle krav der kunne blive stillet til et nyt system. Vi kunne i samtalen nævne ideer og muligheder, uden at forsøge at lede kunden i en bestemt retning og derved bibeholde en stor åbenhed overfor kundens ideer og forslag.

Vi lagde meget vægt på ikke at fare afsted uden at få fundamentet i orden først. Derfor ville vi indledningsvis kun tale om problemområdet til det første møde. Det viste sig dog hurtigt at man ikke altid kan styre samtalen 100% hvorfor vi også hurtigt kom til at tale om anvendelsesområdet og eksempler på systemer. Alligevel fik vi en bedre ide om de krav vores kunder havde gennem mødet samt basis for at stille mere målrettede spørgsmål til det næste møde.

Endelig ville vi beskrive en plan for vores forløb og hvad den næste samtale skulle indebære. Desværre meddelte DrumCity i sidste øjeblik, at de ikke

var interesserede i projektet. Heldigvis fik vi hurtigt kontakt til en anden forretning, MusicAid, som havde præcist samme behov, hvorved vi kunne genbruge vores indledende forberedelser.

18. februar, andet møde med Lars Hedemann. Igen benyttede vi en ganske uformel form for interview til at tale om problem- og anvendelsesområde, aktører og brugsmønstre. På baggrund af det forrige møde var vi denne gang lidt bedre rustede til at få spurgt ind til flere relevante områder. Desuden havde vi inden mødet under vores forberedelser besluttet at tage et par udskrifter fra et eksisterende system (OS-Commerce), for at inspirere Lars Hedeman til at stille nogle konkrete krav.

Præsentationen af de enkelte screenshots blev ledsaget af mindre forklaringer og scenarier for brug af systemet. Denne præsentation gav Lars Hedemann anledning til at nævne et af deres eksisterende systemer - Navision Concorde som eksempel på hvordan det nye system *ikke* skulle virke. Navision Concorde er i følge Lars Hedemann mindet mere på revisorer end på almindelige ekspedienter i butikker.

Efter mødet kunne vi udarbejde et ustruktureret overblik over nogle centrale objekter og så småt begynde på det første udkast til et design af brugergrænsefladen.

Efter vores lille succes med at inspirere Lars Hedemann fra forrige møde besluttede vi at lave et mere konkret mock-up af den brugergrænseflade, som vi sammen med kunden havde fået beskrevet. Dette mock-up blev udformet vha. HTML og billeder. I dette mock-up indgik en række modificerede screenshots af fil/folder screenshots fra Windows XP. Vi linkede fra passende steder til nye billeder, således det kunne blive så realistisk, som muligt og derved inspirere så meget som muligt og forvirre så lidt som muligt. Vi benyttede fil/folder metaforen, netop fordi den er så velkendt for de fleste brugere i dag. Ideen med denne mock-up var også at se om vores kunde synes at denne form for brugergrænseflade ville være intuitiv at arbejde med - på den måde kunne vi få Lars til mere aktivt at deltage i design processen ved faktisk at prøve noget design af i praksis. Vi afprøvede hermed vores evner til at benytte cooperative design / participatory design jvf. [Ehn & Kyng]

27. Februar præsenteres HTML mock-up. For at se, om vores kunde rent faktisk kunne finde hoved og hale i vores mock-up havde vi forbered nogle usecases, der skulle afsløre hvor godt og intuitivt det nye design virkede. Generelt var vores kunde meget tilfreds med at vi havde benyttet ovenstående metaforer, da det gjorde det hurtigt og let at anvende og forstå brugergrænsefladen. Vi benyttede også denne lejlighed til at få beskrevet nogle brugsmønstre, som vi kunne benytte til dels vores rapport og dels til det næste skridt, hvor vi ville lave en mere funktionel prototype der rent faktisk implementerede mange af de elementer, som vi gennem de indledende møder havde fået på plads med vores kunde.

Med ovenstående respons fra Lars Hedemann var vi klar til at producere første udkast til analysedokumentet. Her fandt vi virkeligt ud af at selvom vi havde skrevet dagbog og arbejdet med mock-up, så var der en del ting, som vi ikke havde nok information til at skrive om. Her tænker vi bl.a. på hændelsestabellen og funktionslisten samt brugsmønstrene. Der var der stadig nogle uklarheder, som vi ville være nødt til at få afdækket ved yderligere møder med vores kunde.

Efter respons på analysedokumentet fra vores responsgruppe, erfarede vi, at vores objektmodel mildest talt var mangelfuld og havde mange løse ender. Den reviderede vi derfor, som noget af det første.

Til den 3. iteration fandt vi det nødvendigt at lave en prototype med forholdsvis meget funktionalitet implementeret. At lave endnu en "papmodel" ville ikke bringe større klarhed over systemets design. Derfor dedikerede vi en del tid til programmering i uge 13-17.

uge 11. Valgte sprog til implementation af prototype.

uge 12 - 13 Kvarterspause (eksamener)

uge 14 - 16 Kodning af prototype

uge 17. Præsentation af prototype

Vi valgte at implementere vores prototype i Java, da vi alle i gruppen i forvejen havde kendskab til dette sprog.

Vi opdagede en del brugsmæssige problemer da vi implementerede prototypen. Hastigheden af prototypen ville ikke være acceptabel i det færdige system. Javas Look and Feel agerer ikke altid, som det forventes.

Prototypen skulle primært illustrere hvordan selve varekataloget og varerne kunne administreres. Sekundært var der en smule Drag and Drop funktionalitet, der kunne vise hvordan man kunne trække varer fra kataloget over på en faktura eller et tilbud.

D. 21. april præsenterede vi prototypen for Lars Hedemann. Til afprøvnings af systemet brugte vi en modificeret "tænke højt" test, hvor testpersonen (LH) blev stillet en række opgaver, som han så skulle løse i små logisk sammenhængende skridt. Mellem hvert skridt ville vi bede ham forklare hvorfor han gjorde, som han gjorde. Grunden til dette var, at vi mente at han ville blive for forstyrret for meget af en ideel "tænke højt" jvf [Bødker].

Inden han gik igang med de opgaver vi stillede ham, gjorde vi ham opmærksom på nogle u hensigtsmæssigheder, dels vedr. begrænsningerne i nogle af brugergrænsefladens elementer og dels omkring hastigheden af programmet. De 6 usecases var alle enkle og vedrørte varekataloget.

Vi afsluttede med en generel snak om forbedringer og udvidelser af GUI.

Observationen af LH's betjening af systemet belyste, at det som udgangspunkt er brugbart og overskueligt. LH bidrog desuden med en del forslag og konstruktiv kritik (j.f. Tilføjelser i afsnit 3.4)

Når vi nu kigger på vores tre iterationer af analysen, kan vi se, hvordan de enkelte værktøjer vi har benyttet til at afdække de forskellige områder har været gode på nogen punkter, mens de ikke har været tilstrækkelige på andre. For at tage et eksempel, så var det første mock-up ikke tilstrækkeligt til at vise, hvor meget LH synes det forstyrrer, når der er dialog vinduer i stedet for at næsten alle standard operationer foregår i samme vindue. Dette blev først rigtigt tydeligt, da vi præsenterede ham for vores funktionelle prototype.

Vi så også tydeligt, hvor vigtigt det er at der er så få distraktioner, som muligt, når man vil præsentere noget nyt for kunden. Fokus kan meget hurtigt skifte, når uhensigtsmæssigheder i en prototype flytter fokus fra vores usecase over til problemer med fx hastigheden. Det til trods for at vi inden vi gik igang havde informeret LH om problemer af den art kunne opstå.

På samme måde kan vi nu se at den viden vi har opnået gennem at først få udviklet mock-up og dernæst prototypen og få respons på begge har givet os en meget bedre forståelse af problem- og anvendelsesområdet for systemet. Det har gjort det lettere at få krystalliseret den viden i denne anden udgave af Analysedokumentet.

6 Referencer

Vores mockup kan ses på adressen:

<http://www.aasimon.org/dEkspSys/mockup1/>

Yderligere information om projektet og gruppen, inkl. dagbog kan læses på:

<http://www.aasimon.org/dEkspSys>