



AARHUS UNIVERSITET
BUSINESS AND SOCIAL SCIENCES
INSTITUT FOR MARKETING OG ORGANISATION

Gruppeopgave, vintereksamen 2011 Modellering af informationssystemer, HA(it)

(UDFYLDES MED BLOKBOGSTAVER)

Navne (fulde navn) i gruppen:

NIELS CHRISTIAN MUNCK

SOFIE MARIE DAM MIKKELSEN

ANNE JUSTESEN

Aflevering tirsdag d. 20. december 2011 inden kl. 12.00

- **2 papirkopier til Vibeke Frisenvang på flg. adresse:**

Kontor HG102 - Haslegårdsvej 10

> [Find vej](#) (ind gennem kantinen, og videre ud i c-fløjen)

- **1 upload på CampusNet, under "Opgaver"**

Klik på 'Opgaver' i venstremenuen og klik evt. på  i øverste højre hjørne for at se en beskrivelse af, hvordan du upload'er din besvarelse

Indholdsfortegnelse

Del A: Kravspecifikation	3
1) Introduktion	3
2) Formål	3
Afgrænsning:	3
3) Omgivelser	4
a. Problemområde	4
b. Anvendelsesområde	4
Det rige billede.....	4
Forbilleder	5
Metaforer	6
4) Systemdefinition	6
A. Problemområde.....	8
1) Klasser	8
2) Hændelser	10
3) Hændelsestabel	13
4) Adfærdsmønster	15
Vare til salg.....	15
Udlejningsvare	15
Udlejningsordre.....	16
Ordre	17
Kunde	18
Medarbejder.....	19
Ordrelinje.....	19
5) Klynger	20
6) Struktur.....	20

7) Klassediagram	22
B. Anvendelsesområde	23
1) Brug	23
a) Oversigt:	23
2) Aktører	24
b) Brugsmønstre	25
3) Funktioner	29
a) Komplet funktionsliste	29
4) Brugergrænsefladen	31
5) Ikke-funktionelle krav	38
C. Databaser	40
Den tekniske platform:	40
Overordnet arkitektur:	43
Del B: Logbog	45
Foranalyse	45
Analyse af problemområdet	47
Analyse af anvendelsesområdet	53
Etablering af teknisk platform	54
Etablering af overordnet arkitektur	54

Del A: Kravspecifikation

1) Introduktion

Formålet med kravspecifikation er at udvikle et it-system, som har til formål at kunne løse lagerstyring og udlejningsproblemet som en samlet enhed.

Der anvendes OOA&D metoden til at belyse problemstillinger for derved at opnå den bedst mulige løsning. Denne metode er der skabt bekendtskab med i bogen "Objektorienteret, analyse og design"¹.

På baggrund af denne metode er der først udarbejdet en analyse af problemområdet og dernæst en analyse af anvendelsesområdet. Derefter er arkitekturen blevet designet ved hjælp af non-funktionelle krav og komponenter, som yderligere uddybes.

Derudover er den tekniske platform beskrevet i opgaven, som viser databasestrukturen.

2) Formål

Norup Tømmerhandel & Byggecenter er en lokal detailvirksomhed bestående af 2 ledere, der hver især står for tømmerhandlen og butikken. Hver leder har henholdsvis 15 og 11 ansatte under sig.

Ledelsen har tilkaldt et hold bestående af tre systemudviklere, idet de har rod på lageret og mangler overblik over udlejning. Derfor har de svært ved at yde den ønskede kundeservice. Ledelsen har indset, at der skal ske en radikal ændring i virksomheden. Ledelsen ønsker derfor at få udarbejdet en kravspecifikation for et it-system og har i den forbindelse ansat os, som værende de tre systemudviklere.

Afgrænsning:

Denne kravspecifikation er udarbejdet for at tage hånd om lagerstyring og udlejning. I rapporten ses der bort fra handel over nettet samt rykkere og reparationer ved udlejning.

Lederne for Norup Tømmerhandel og Byggecenter har ytret et ønske om, at de gerne vil have, at alle kunder er oprettet i systemet, hvilket indebærer, at hver kunde får tildelt et kundenummer. Dette er for at skabe en relation mellem kunden og butikken.

¹ Objektorienteret analyse og design, Matthiasen Lars, Munk-Madsen Andreas, Axel Nielsen Peter, Stage Jan, 3.udgave, Marko ApS 2001.

Når der skrives "vi" i opgaven, skal det forstås, at det ses fra systemudviklernes synsvinkel.

3) Omgivelser

Problemområdet og anvendelsesområdet er her kort beskrevet, men vil blive yderligere uddybet i det videre forløb.

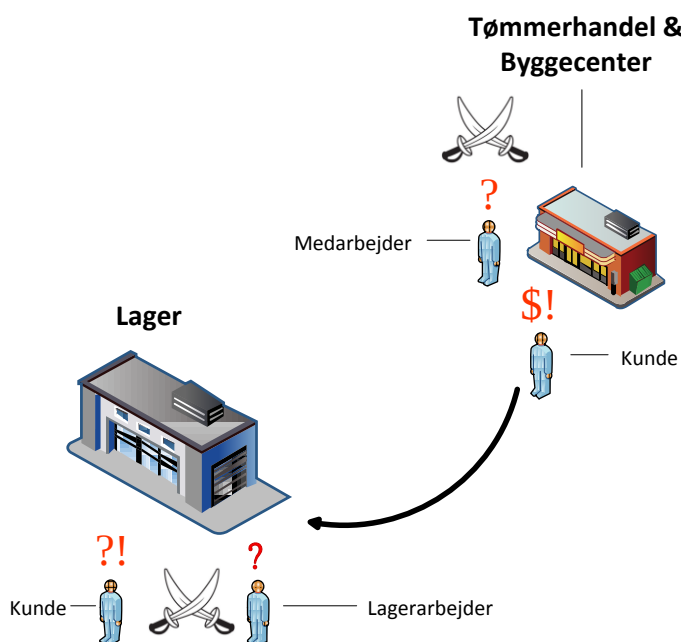
a. Problemområde

Systemet skal kunne registrere varebeholdningen. Ved køb af varer og udlejning af maskiner, skal denne varebeholdning løbende opdateres, så systemet indeholder information om antal enheder af varer og udlejningsvarer, der er på lager. Ved køb af varer og udlejning af en vare skal systemet kunne registrere kundeoplysninger.

b. Anvendelsesområde

Systemet skal fungere ved hjælp af en computer. Denne computer skal styres af medarbejderne og ledelsen for derved at kunne hente den nødvendige information og foretage de nødvendige funktioner.

Det rige billede



Det rige billede er udarbejdet ved, at systemudviklerne har været ude og observere og interviewe medarbejderne samt lederne i Norup Tømmerhandel og Byggecenter. På baggrund af observationer og interviews har systemudviklerne dannet sig et overblik over elementer, processer, strukturer samt problemerne i virksomheden. Det rige billede fungerer dermed som et kommunikationsredskab, som skaber fælles forståelse blandt systemudviklerne. Dermed er systemudviklerne klar over, hvilke problemer systemet skal gå ind og supportere.

Det starter med, at der foregår et salg eller en udlejning mellem elementerne kunde og medarbejder, som er en form for struktur. Det ses, at medarbejderen er i tvivl, om de solgte varer er på lager, som er illustreret med et spørgsmålstegn. Derefter sker der en proces ved, at kunden henvender sig på lageret for at få udleveret varerne. Kunden står og venter, og kan ikke finde en lagermedarbejder. Samtidig ses en struktur ved, at en lagermedarbejder løber fortvivlet rundt, og leder efter varer til en anden kunde. Dette giver lang ventetid, som ender ud i dårlig kundeservice.

For at give overblik er de ovenstående problemer illustreret med krydsende klinger på billedet.

Systemudviklerne vil ud fra to teknikker (metaforer og forbilleder) generere og vurdere idéer til løsning af problemer.

Forbilleder

SAP - Business One er et ERP system, der er lavet til mindre virksomheder. Derudover er der også Dynamics NAV. Begge ERP systemer er store forbilleder, da de indeholder styringselementer. Dynamics NAV indeholder muligheden for en applikation til udlejning². Elementerne lagerstyring og en udlejningsapplikation ville være gode ideer til systemet. Dog skal der ikke anvendes et helt ERP system, men kun dele af det. Derfor skal systemet indeholde en applikation til udlejningen, der tilpasses lagerstyring. Dog skal forbillederne ikke fastlåse andre muligheder.

² <http://pinpoint.microsoft.com/da-dk/applications/udlejnings-system-til-dynamics-nav-4295021605>

Metaforer

Virksomheden har ofte svært ved at overskue deres udlejning af varer til kunder. Udlejningen foregår ved, at kundens navn og telefonnummer skrives ned på et "udlejnings-papir" ud for den enkelte maskine. Det sker ofte, at dette papir ikke bliver opdateret:

- Når udlejningsvaren afleveres tilbage.
- Hvis kunden ringer ind for at bestille flere udlejningsdage.
- En udlejningsvare bliver udleveret uden, at det registreres nogen steder.

Dermed sker det, at virksomheden ikke aner, hvem der har udlejningsvaren, og hvor længe de har haft den. For at løse problemet anvendes metaforisk design, hvor brugerorganisationen ses med nye øjne. Her ses på, hvilket system Blockbuster anvender, og hvordan systemet arbejder i forhold til at overskue deres udlejninger. Blockbusters udlejning foregår ved, at hver kunde tildeles et udlejningskort. Dette kort fremvises samt der skal oplyses en kode ved udlejning. Kunden får oplyst, hvornår materiellet skal afleveres tilbage, og hvis dette ikke sker indenfor det angivne tidsrum, påføres der bøder. Kunden kan dermed ikke leje andet materiel, før denne bøde er betalt. Samlet set giver systemet overblik over, hvem der har materiellet, hvilket materiel de har, samt hvornår det afleveres tilbage. Dette system vil Norup Tømmerhandel og Byggecenter kunne udnytte i forhold til at sammensætte et mere overskueligt udlejningssystem, og det vil systemudviklerne have i tankerne for at løse udlejningsproblemet.

4) Systemdefinition

For at sammenfatte IT-systemets helhedsegenskaber anvendes BATOFF-kriteriet, som er med til at kvalitetssikre og skabe en konsistent formulering af systemdefinitionen.

Betingelser:	Udvikles i tæt samarbejde med de ansatte.
Anvendelsesområde:	Medarbejderne
Teknologi:	PCere forbundet med et sikkert netværk.
Objekter:	Varer, udlejningsvare, personale, kunder, PCere, opgaver
Funktioner:	IT-systemet skal bruges til at styre lageret og udlejningen
Filosofi:	Planlægningsværktøj

Et it-system, der bruges til at styre lageret og udlejningen af varer, der anvendes af medarbejderne. Systemet skal være et planlægningsværktøj i forhold til lagerstyring og udlejning af varer, så virksomheden har styr på hvilken kunde, der har hvilken udlejningsvare. Systemet skal håndtere kundeoplysninger. Systemet skal baseres på PC'ere forbundet med et internt, sikkert netværk. Systemet skal implementeres i tæt samarbejde med Norup Tømmerhandel og Byggecenter, hvor udviklerne er interesseret i en løbende feedback fra brugerne.

A. Problemområde

Udgangspunktet for analysen af problemområdet er systemdefinitionen.

Modelleringen af problemområdet tager udgangspunkt i klasseaktiviteten, som er fundamentet i problemområdet. Derefter ses der på struktur, som beskriver sammenhængen mellem klasser og objekter. Til sidst i problemområdet kigges der på objekterne, hvor egenskaberne og attributter for hver klasse beskrives.

1) Klasser

Nedenfor har vi lavet en kandidatliste over mulige klasser. Vi har hentet inspiration fra det rige billede og systemdefinitionen. Derefter har vi udvalgt hvilke klasser systemet skal bevare informationer om. De udvalgte klasser er markeret i højreside med ✓

Beskrivelse af klasser i systemet

1. Kunde	Denne klasse indeholder informationer om kundens Navn, Adresse, PostNr, By, Telefonnummer, E-mail og Dato for oprettelse. Derudover skal der også oplyses CVR og EAN nummer (fremgår som Erhvervstype), hvis det er virksomheder eller offentlige instanser.	✓
2. Faktura	Indeholder information omkring: navn, adresse, e-mail og telefonnummer. Derudover skal der også oplyses CVR og EAN nummer hvis det er virksomheder eller offentlige instanser. Der skal være oversigt over varenummer, antal stykker og samlede beløb. Ved kontant køb står der kun varenummer, antal styk og samlede beløb.	
3. Ordrekladde	Indeholder information omkring: navn, adresse, e-mail og telefonnummer. Derudover skal der også oplyses CVR og EAN nummer hvis det er virksomheder eller offentlige instanser. Der skal være oversigt over varenummer, antal stykker og samlede beløb. Ved kontant køb står der kun	

	varenummer, antal styk og samlede beløb.	
4. Tilbud	Indeholder information omkring: navn, adresse, e-mail og telefonnummer. Derudover skal der også oplyses CVR og EAN nummer hvis det er virksomheder eller offentlige instanser. Der skal være oversigt over varenummer, antal stykker og samlede beløb. Ved kontant køb står der kun varenummer, antal styk og samlede beløb.	
5. Følgeseddel	Indeholder information omkring: navn, adresse, e-mail og telefonnummer. Derudover skal der også oplyses CVR og EAN nummer hvis det er virksomheder eller offentlige instanser. Der skal være oversigt over varenummer, antal stykker og samlede beløb. Ved kontant køb står der kun varenummer, antal styk og samlede beløb.	
6. Ordre	Indeholder information omkring: OrdreNr, KundeNr, Dato og MedarbejderNr.	✓
7. Udlejningsordre	Indeholder information om: UdlejningsordreNr, KundeNr, SerieNr, Udlejningsperiode og MedarbejderNr.	✓
8. Medarbejder	Klassen Medarbejder indeholder information omkring: MedarbejderNr, Navn, Adresse, PostNr, By, Telefon, Afdeling, Køn, CprNr.	✓
9. Vare til salg	Optræder ligesom et lager. Det vil sige, at systemet skal registrere, hvor mange varer, vi har i butikken og på det fysiske lager. VareNr, Varenavn, Antal enheder på lager, Indkøbspris, Varespecifikation.	✓
10. Udlejningsvare	Optræder som et lager for hvor mange udlejningsvarer, der er ledige til udlejning. Klassen information om: UdlejningsordreNr, SerieNr (hver eneste udlejningsvare får tildelt et specifikt	✓

	SerieNr.), Produktnavn, Antal enheder på lager og Udlejningspris pr. dag.	
11. Computer	Skal indeholde det IT-system, systemudviklerne er ved at udvikle.	
12. Ordrelinje	Denne klasse er indsat for at undgå mange-til-mange relation, og er dermed en interaktion. Dette er for at undgå redundant data. Klassen indeholder information om: OrdrelinjeID, OrdreNR, VareNr, Antal solgte enheder og Salgspris.	✓

2) Hændelser

Ligesom med kandidatlisten til klasserne har vi også valgt at lave en kandidatliste til hændelserne, der kan indgå i modellen af problemområdet. De udvalgte hændelser er markeret med ✓

Beskrivelse af hændelser i systemet

Reservere	Denne hændelse indtræffer, når en medarbejder reserverer en udlejningsvare for en kunde.	✓
Udleje	Denne hændelse indtræffer, når udlejningsvare udlejes.	✓
Bestille	Denne hændelse forekommer, når en kunde bestiller en udlejningsvare.	✓
Sende	Denne hændelse er grundlaget i systemet, der skal kunne sende informationer rundt blandt medarbejderne.	
Udlevere	Det fremstår af fakturaen, ordrekladde, tilbuddet eller følgesedlen, hvilke varer medarbejderen skal udlevere.	
Vente	Denne hændelse sker når kunden står og venter efter en lagermedarbejder.	
Styre	Denne hændelse fremkommer, når en medarbejder styrer computeren.	

Oprette	Denne hændelse indtræffer, når der: - Oprettes en "vare til salg" - Oprettes en "udlejningsvare" - Oprettes en "medarbejder" - Oprettes en "udlejningsordre" - Oprettes en "kunde" - Oprettes et "tilbud" - Oprettes en "ordre"	✓
Slette	Denne hændelse indtræffer, når der i systemet: - Slettes en "kunde" - Slettes en "udlejningsvare", hvis den fx går i stykker. - Slettes en "vare til salg"	✓
Genoprette	Denne hændelse sker, når der: - Genoprettes en kunde, hvis den har været slettet - Genoprettes en "vare til salg", hvis den har været slettet	✓
Opdatere	Denne hændelse indtræffer, når en medarbejder opdaterer varebeholdning. Der sker både, når medarbejderen bestiller varer, returnerer varer fra kunder og modtager varer fra leverandøren. Derudover bliver varebeholdningen også opdateret, når medarbejderen sletter/genopretter varer. Denne hændelse optræder både i klassen "medarbejder" og "vare til salg" og er dermed en <i>fælles hændelse</i>. Ved "ordrelinje" skal hændelsen "opdatere" forstås som, at der tilføjes varer til ordrelinjen.	✓
Ændre	Hvis udlejningsperioden skal ændres på udlejningsordren, sker denne hændelse. Derudover optræder denne hændelse i "kunde", hvis der skal ændres kundeinformation om kunden. Det kan fx være et tlf. nr. eller en adresse, der skal ændres.	✓
Returnere	Denne hændelse indtræffer, når:	✓

	- En kunde returnerer varer, som ses i "kunde". - Udlejningsvaren bliver returneret.	
Godkende	Kunden godkender tilbuddet, sker under "ordre"	✓
Annullere	Denne hændelse sker, hvis kunden annullerer "ordren" eller "udlejningsordren". Hændelsen er sporadisk og kan ske på ethvert tidspunkt i livscyklussen.	✓
Tilføje	Kunden bliver tilføjet en ordre eller en udlejningsordre. Denne hændelse indtræffer også, når ordren eller udlejningsordren bliver tilføjet henholdsvis ordrelinjer og udlejningsvare.	✓
Gennemføre betaling	Denne hændelse optræder både på udlejningsordren, ordren og dermed også ordrelinjen.	✓
Afslutte	Optræder på "ordre" og "udlejningsordre".	✓
Indsætte	Kontokunde indsætter penge på sin konto, hvor dato og beløb skal registreres	
Udbetale	Hvis der er for mange penge på kontoen, er det muligt for kontokunden at få udbetalt sine penge	
Afskedige	Når en medarbejder afskediges fra jobbet, er han/hun ikke længere aktiv i systemet	✓
Opnå	Når kunden opnår kredit, som både indtræffer på udlejningsordren og ordren.	
Afhente	Opstår når kunden afhenter udlejningsvaren.	✓

3) Hændelsestabel

Nedenfor ses en hændelsestabel, hvor de udvalgte klasser er karakteriseret ved et bestemt mængde hændelser.

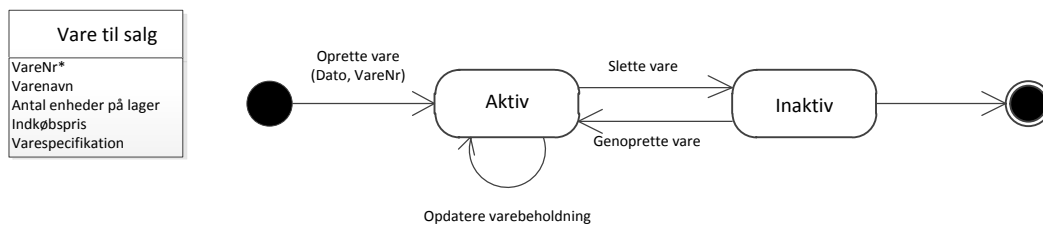
<i>Klasser</i>							
<i>Hændelser</i>	Kunde	Ordre	Udlejningsordre	Medarbejder	Vare til salg	Udlejningsvare	Ordrelinje
Reservere				✓			
Udleje						✓	
Bestille						✓	
Oprette	✓	✓	✓	✓	✓	✓	✓
Slette	✓				✓	✓	
Genoprette	✓				✓		
Opdatere				✓	✓		✓
Ændre	✓		✓				
Returnere	✓					✓	
Godkende		✓					
Annullere		✓	✓				
Tilføje		✓	✓				
Gennemføre betaling		✓	✓				✓
Afslutte		✓	✓				
Indsætte							
Udbetale							

Afskedige				✓			
Afhente						✓	

4) Adfærdsmønster

I følgende afsnit beskrives mulige hændelsesforløb for alle objekter i en klasse. Den primære nøgle for hver klasse er indikeret med en *. "Dato", som der står under hændelserne i tilstandsdiagrammerne registreres af systemet.

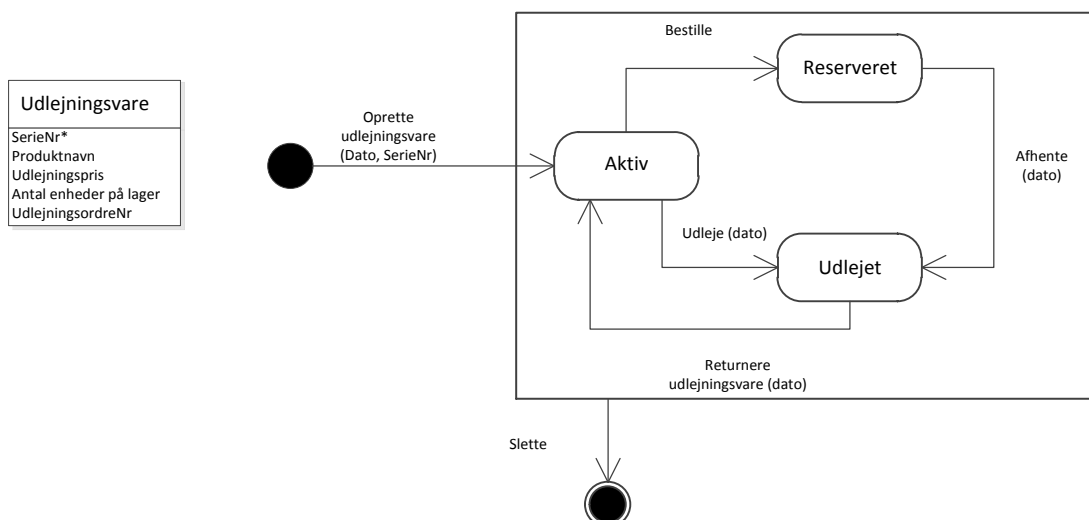
Vare til salg



Dette tilstandsdiagram viser, at der oprettes en vare i systemet, hvorefter den er aktiv. I tilstanden aktiv, kan varebeholdningen blive opdateret. Varen kan gøres inaktiv i systemet ved, at den slettes, og den er derfor ikke tilgængelig i systemet. Dog kan den genoprettes igen eller dø.

Kontrolstrukturen er sekvens, og i tilstanden aktiv er der brugt iterationer. Mellem tilstanden aktiv og inaktiv er der anvendt indirekte iteration.

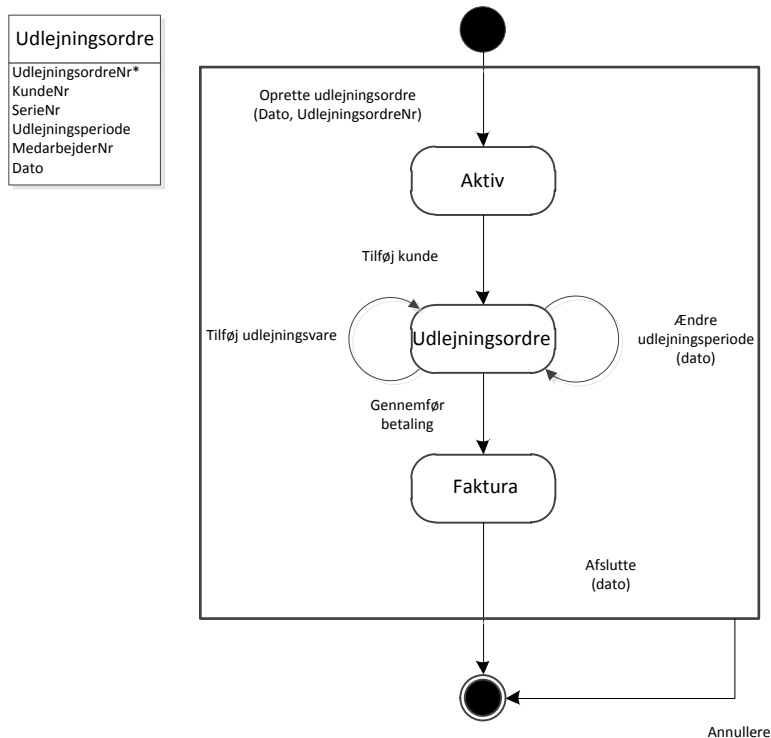
Udlejningsvare



Dette tilstandsdiagram starter ved, at udlejningsvaren bliver oprettet i systemet, hvorefter udlejningsvaren er aktiv i systemet. Herefter er der to muligheder. 1. Kunden

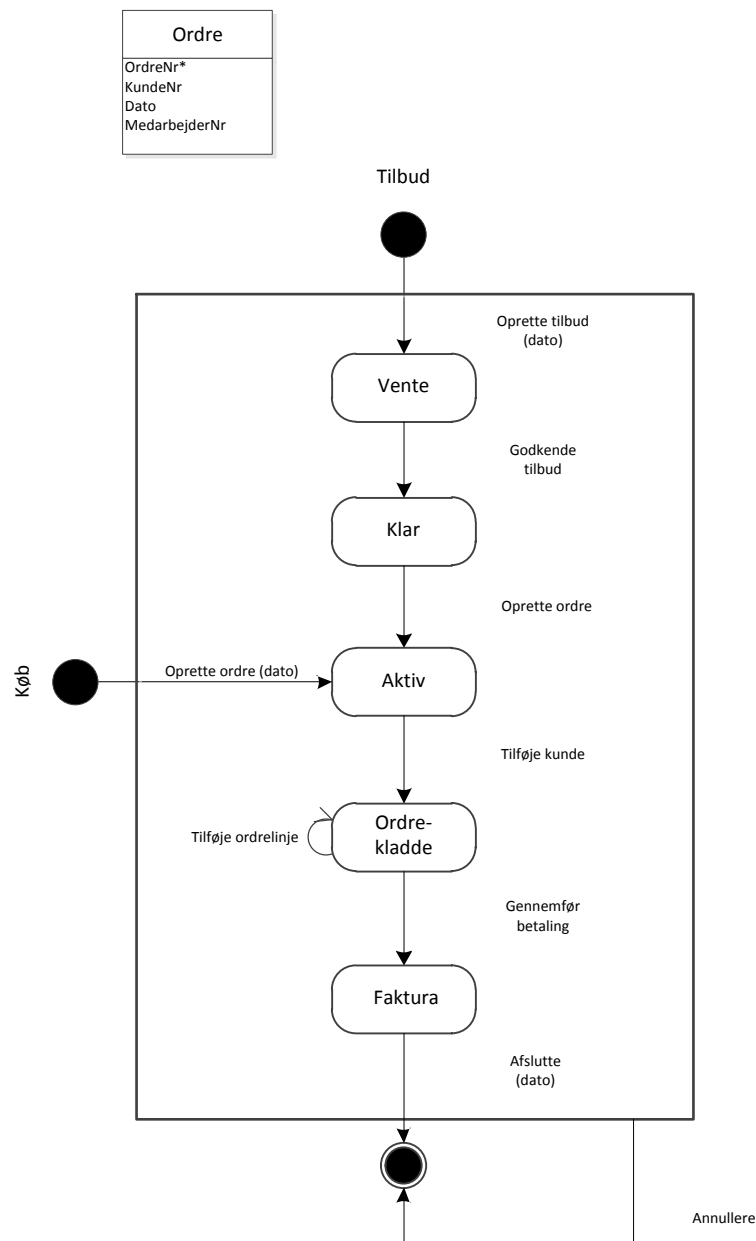
kan bestille udlejningsvaren, og den er dermed reserveret og udlejet, når kunden afhenter varen. 2. Udlejningsvaren bliver udlejet uden af den forinden er blevet reserveret. Når udlejningsvaren er udlejet, bliver den igen aktiv i systemet ved, at udlejningsvaren returneres, og den er dermed tilgængelig for at blive udlejet på ny. Kontrolstrukturen er sekvens. Derudover ses det, at i tilstanden "aktiv", kan der forekomme to situationer enten "bestille" eller "udleje", dette kaldes selektion. Derudover ses det, at en udlejningsvare kan slettes på ethvert tidspunkt i sin livscyklus ved den hierarkiske tilstand, hvis den skal skrotes.

Udlejningsordre



Udlejningsordren oprettes og er dermed aktiv i systemet. I tilstanden aktiv bliver der tilføjet en kunde, hvorefter den skifter til tilstanden "udlejningsordre". I denne tilstand tilføjes der en udlejningsvare til udlejningsordren, og her er det også muligt at ændre udlejningsperioden. Dernæst gennemføres betalingen og ordren afsluttes. Kontrolstrukturen er sekvens, og der anvendes to iterationer i tilstanden "udlejningsordre".

Ordre



Dette tilstandsdiagram har to startpunkter. Enten starter det ved, at der bliver oprettet et tilbud eller, at der bliver oprettet en ordre. Vi beskriver først, hvordan forløbet er, hvis der bliver oprettet et tilbud.

Det starter med, at der bliver oprettet et tilbud, hvorefter tilstanden går i venteposition, da kunden enten først skal godkende tilbuddet. Hvis kunden godkender tilbuddet oprettes der en ordre, hvorefter ordren er aktiv i systemet. Der tilføjes en kunde, hvorefter det bliver til en ordre. I tilstanden "ordrekladde" tilføjes der

ordrelinjer til ordren. Det antages, at hver gang der oprettes en ordre tilføjes der implicit ordrelinjer. Dernæst gennemføres betaling og ordren afsluttes.

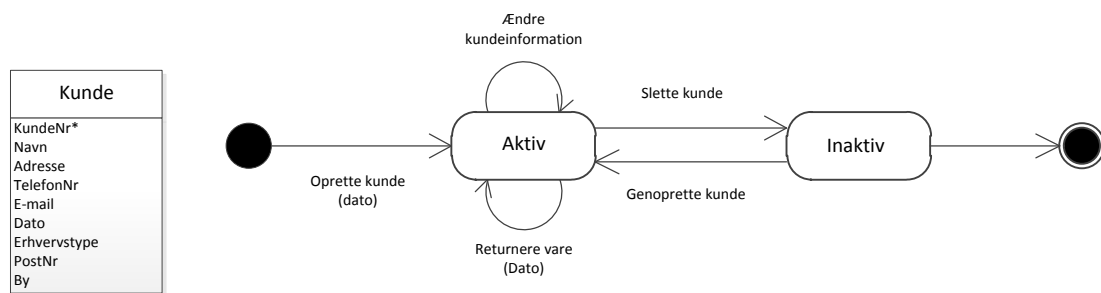
Hvis kunden ikke ønsker et tilbud, starter købet med, at der oprettes en ordre og derefter er forløbet det samme som beskrevet ovenfor.

Det antages, at lige så snart tilbuddet eller ordren bliver genereret, har tilbuddet, ordren og fakturaen det samme ordrenummer hele vejen igennem. Det vil sige, at ordrenummeret er "reserveret" hele vejen igennem.

Kontrolstrukturen er sekvens. Derudover ses det i tilstanden "ordrekladde", at der forekommer en iteration.

Ordre er en sporadisk proces, hvilket vil sige, at hændelsen "annullere" kan afslutte et objekt på et vilkårligt tidspunkt i forløbet. Derfor ses en hierarkisk tilstand.

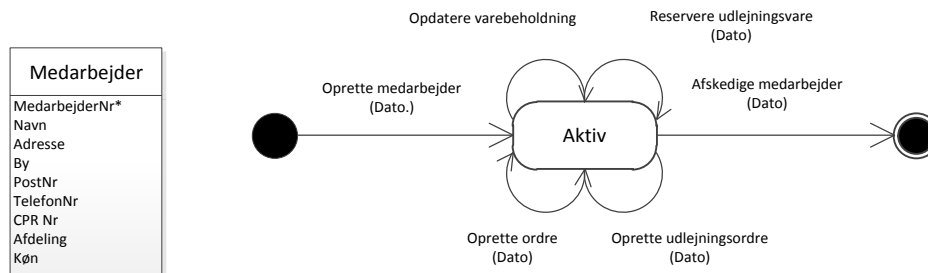
Kunde



Tilstandsdiagrammet starter med, at kunden bliver oprettet i systemet og har dermed en aktiv konto i systemet. Det er muligt at ændre kundeinformation, hvis kunden eksempelvis har fået ny adresse eller telefonnummer. Derudover kan kunden returnere varer i tilstanden aktiv. Hvis kunden ikke betaler sin regning eller ønsker at få sin konto lukket, slettes kontoen i systemet og den er dermed inaktiv, hvorefter den dør eller genoprettes.

Kontrolstrukturen er sekvens med iterationer i tilstanden aktiv. Derudover ses der indirekte iteration mellem tilstandene aktiv og inaktiv.

Medarbejder



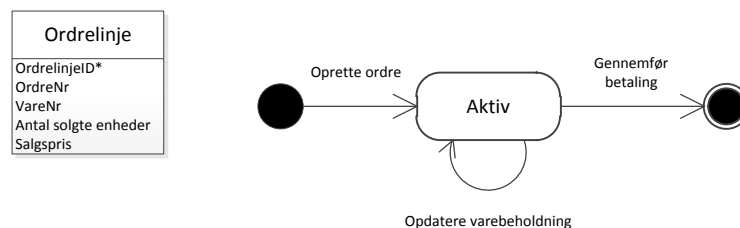
Dette tilstandsdiagram starter ved, at medarbejderen oprettes og bliver dermed aktiv i systemet ved, at han/hun får tildelt et medarbejdernummer. Når medarbejderen er aktiv, kan han/hun foretage følgende:

- Oprette ordre
- Oprette udlejningsordre
- Reservere udlejningsvare
- Opdatere varebeholdning

Hvis medarbejderen afskediges, dør tilstanden.

Kontrolstrukturen i tilstandsdiagrammet er sekvens, og derudover er der iterationer, når tilstanden er "aktiv".

Ordrelinje



Dette tilstandsdiagram starter med, at der oprettes en ordre, hvorefter ordrelinjen er aktiv i systemet. Når ordrelinjen er aktiv i systemet, bliver varebeholdningen opdateret ved, at der tilføjes "vare til salg" på ordrelinjen. Tilstand dør, når betalingen gennemføres.

5) Klynger

Efter nærmere gennemgang af vores klassediagram, er der vurderet, at behovet for et klyngediagram ikke er til stede. Dels er det, fordi det ikke gør, at problemområdet bliver mere anskueligt for læseren. Men også grundet, at der ikke er mange oplagte klyngestrukturer i problemområdet. Derfor ville indførelsen af et klyngediagram ikke være relevant.

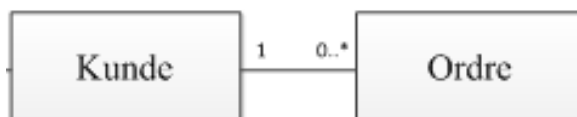
6) Struktur

I klassediagrammet er der associeringsstrukturer, som bliver uddybet nærmere nedenfor.

Associeringsstrukturer

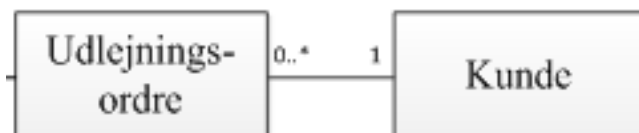
I klassediagrammet er der i alt 7 forskellige associeringer, hvor 3 klasser går igen ved 2 associeringer. Nedenstående bliver de enkeltes klassers kardinaliteter og strukturgrundlag forklaret.

- Kunde - Ordre:



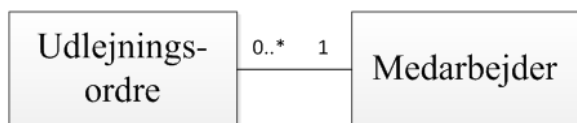
En kunde har nul eller flere ordrer, da kunden sagtens kan være registeret i systemet uden at skulle have købt noget. Derimod kan en ordre, kun have én kunde.

- Kunde - Udlejningsordre:



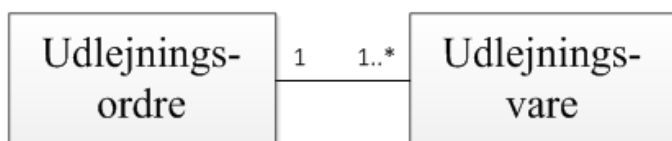
En kunde kan have nul eller flere udlejningsordrer, da kunden godt kan være registeret i systemet uden at have lejet noget. Dog kan en udlejningsordre kun have én kunde.

- **Udlejningsordre - Medarbejder:**



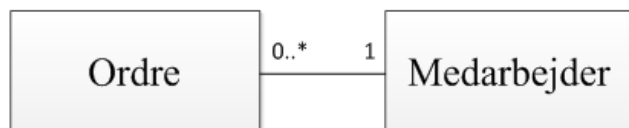
En udlejningsordre kan have én medarbejder og kun én. Klassen "Medarbejder" kan oprette nul eller flere udlejningsordrer.

- **Udlejningsordre - Udlejningsvare:**



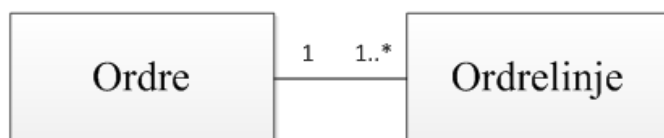
En udlejningsvare kan kun være på én udlejningsordre og kun én. Dog kan en udlejningsordre godt have én eller flere udlejningsvarer. Strukturen er anvendt, da det kun skal være muligt for én specifik udlejningsvare at være på én udlejningsordre.

- **Ordre - Medarbejder:**



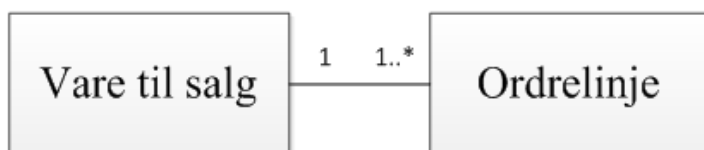
En ordre kan kun indeholde én medarbejder, men en medarbejder kan godt oprette 0 eller flere ordrer.

- **Ordre - Ordrelinje:**



En ordre kan indeholde én eller flere ordrelinjer, men én ordrelinje kan kun indgå på én ordre.

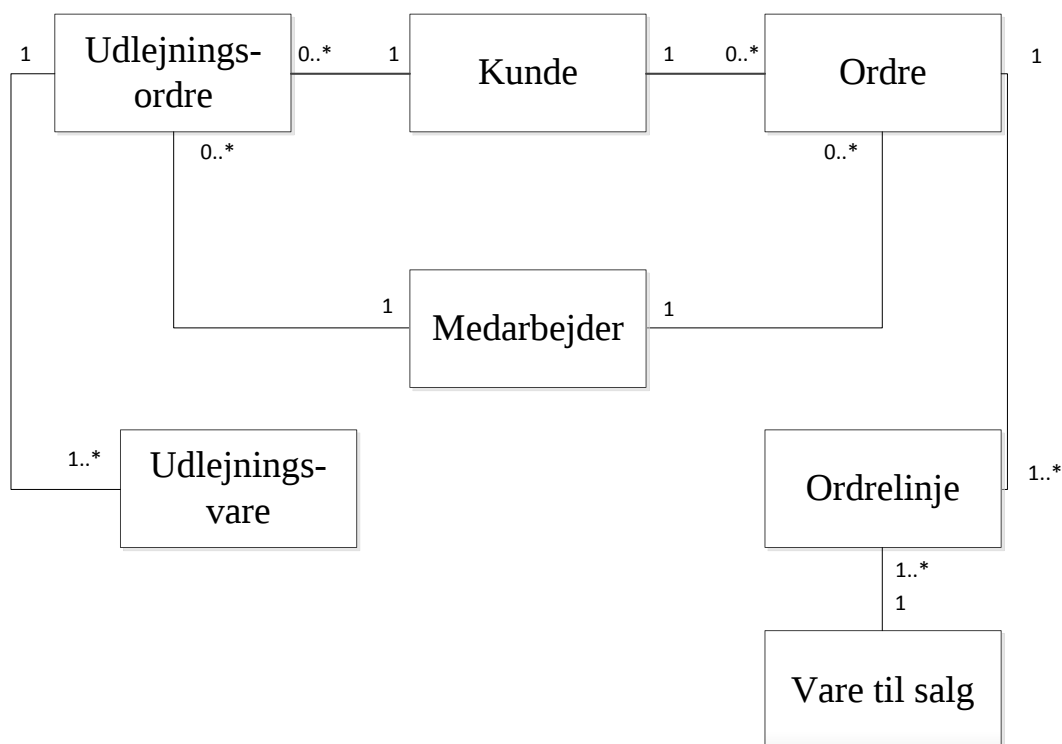
- Vare til salg - Ordrelinje:



En "vare til salg" kan være på én eller flere ordrelinjer, men en ordrelinje kan kun bestå af én og kun én "vare til salg".

7) Klassediagram

Her ses vores klassediagram, hvor de enkelte dele er sammensat til et hele.



B. Anvendelsesområde

1) Brug

a) Oversigt: Aktørtabel for Byggecenter og Tømmerhandelssystemet

Nedenfor ses en oversigt over analysen af systemets brug, som beskriver hvilke aktører, der kan udføre hvilke brugsmønstre.

	Butiks- medarbejder	Tømmerhandels- medarbejder	Kasseassistent	Lager- medarbejder	Administrator
Brugsmønstre					
Ordre- administration	✓	✓	✓		
Udlejnings- administration		✓			
Kunde- administration		✓			✓
Medarbejder- administration					✓
Modtagelse af betaling		✓	✓		
Bestilling af varer	✓	✓			
Modtagelse af varer				✓	
Oprettelse af en vare	✓	✓			

2) Aktører

Hver aktør er beskrevet i følgende afsnit.

Aktørspecifikationer:

Butiksmedarbejder

Formål: En person, som arbejder i butikken. Butiksmedarbejderens arbejdsområde er at kunne oprette en ordre, oprette varer og bestille varer.

Karakteristik: Systemets brugere omfatter mange medarbejdere med forskellige erfaringer.

Tømmerhandelsmedarbejder

Formål: En person, som arbejder i tømmerhandel. Tømmerhandelsmedarbejderens opgaver er at oprette ordrer, modtage betaling, lave reservationer af udlejningsvarer og bestille varer hjem til tømmerhandlen. Derudover skal medarbejderne i Tømmerhandlen kunne oprette kunder i systemet, da de har indsigt i hvilke rabatter, der er rentabelt at give de forskellige kundegrupper.

Hvis der kommer nye varer hjem til Tømmerhandel, skal medarbejderne være i stand til at oprette varerne.

Karakteristik: Systemets brugere omfatter mange medarbejdere med forskellige erfaringer.

Kasseassistent

Formål: En person, som arbejder i kassen. Kasseassistentens opgaver er at oprette ordrer samt modtage betaling.

Karakteristik: Systemets brugere omfatter mange medarbejdere med forskellige erfaringer.

Lagermedarbejder

Formål: En person, som arbejder på lageret. I forhold til systemet er lagermedarbejderens eneste opgave at opdatere varebeholdningen i form af at scanne hjemkomne varer ind i systemet.

Karakteristik: Systemets brugere omfatter mange medarbejdere med forskellige erfaringer.

Administrator

Formål: I Norup Tømmerhandel og Byggecenter er der to ledere, som står for administrationen. Administratorernes opgaver er kunde- og medarbejderadministration, hvor de kan tilføje, slette og ændre kunde- og medarbejderinformationer. Dog kan administratoren også foretager andre arbejdsopgaver, og dermed fungerer som aktøren Butiksmedarbejder og Tømmerhandelsmedarbejder.

Karakteristik: Systemets brugere omfatter mange medarbejdere med forskellige erfaringer.

b) Brugsmønstre

Brugsmønsteret ”modtagelse af betaling” er uddybet, da denne er kompleks.

Brugsmønsterspecifikationer:

Ordreadministration

Brugsmønster: Ordreadministration indeholder brugsmønstre oprettelse, sletning og ændring af en ordre eller et tilbud, og denne igangsættes af en butiksmedarbejder, tømmerhandelsmedarbejder eller en kasseassistent.

Navn: Ordreadministration	
Formål: At behandle et salg	
Frekvens: Aperiodisk	
Aktør(er): Butiksmedarbejder, tømmerhandelsmedarbejder og kasseassistent	
Startbetingelser:	
Beskrivelse:	
Aktør	System
1. Medarbejderen modtager kundens varer og medarbejderen opretter en ordre	2. Systemet åbner et ordre-vindue med et ordrenummer. 4. Kundeoplysninger fremkommer

3. Medarbejderen indtaster kunde-nummer. [1.Kunde ikke oprettet i systemet eller ønsker at betale kontant]	6. Systemet tilføjer varerne og implicit ordrelinjer på ordren.
5. Medarbejderen scanner varerne	8. Systemet viser det samlede beløb på varerne
7. Medarbejderen oplyser det samlede beløb og anmoder om betaling [2. Kunde vil betale på konto]	10. Systemet håndterer betalingen
9. Medarbejderen modtager betaling fra kunden - <i>se brugsmønsteret for modtagelse af betaling</i>	11. Systemet udskriver kvittering
Undtagelser: 1. Hvis kunden ikke er oprettet i systemet eller han/hun ønsker at betale kontant, skrives 999 på "kundennummer-feltet". 2. Gå til step 11.	
Sluttilstand: Betalingen godkendes.	Version: 1
Initialer: MedarbejderNr.	

Objekter: Ordre, kunde, vare, medarbejder

Funktioner: Opret ordre, Opret ordrelinje

Udlejningsadministration

Brugsmønster: Udlejningsadministration igangsættes af en tømmerhandelsmedarbejder. Dette brugsmønster starter med, at der er en kunde, der kontakter Tømmerhandlen for at lave en reservation af en udlejningsvare. Reservationen kan senere blive slettet, ændret eller annulleret.

Objekter: Kunde, udlejningsordre, udlejningsvare, medarbejder

Funktioner: Opret reservation, Opret udlejningsordre

Kundeadministration

Brugsmønster: Kundeadministration starter med, at en kunde ønsker at få en konto, hvorefter kunden bliver oprettet af enten en tømmerhandelsmedarbejder eller en administrator, som tilføjer de personlige informationer.

Objekter: Kunde, medarbejder, administrator

Funktioner: Opret kunde

Modtagelse af betaling

Brugsmønster: Modtagelse af betaling sker, når kasseassistenten eller tømmerhandelsmedarbejderen modtager betaling fra kunde. Dette kan ske i form af betaling med kontanter eller med betalingskort. Forløbet for Dankort ses ud fra beskrivelsen nedenfor.

<i>Navn:</i> Dankortterminal		
<i>Formål:</i> At modtage betaling		
<i>Frekvens:</i> Aperiodisk		
<i>Aktør(er):</i> Tømmerhandelsmedarbejder og kasseassistent		
<i>Startbetingelser:</i>		
<i>Beskrivelse:</i>		
Aktør	System	
2. Kort indlæses	1. Anmod om betaling	
3. Pinkode verificeres	4. Kode godkendt	
[Pinkode ikke korrekt]	5. Betaling gennemføres, og der printes en kvittering	
<i>Undtagelser:</i>		
Betalingen bliver ikke gennemført, hvis pinkoden ikke er korrekt. Der skrives en fejlmeddelelse i displayet; forkert pinkode.		
<i>Sluttilstand:</i> Betalingen godkendes.	<i>Version:</i> 1	<i>Initialer:</i> Medarbejdernr.

Objekter: Kunde, medarbejder

Funktioner: Modtagelse af betaling

Bestilling af varer

Brugsmønster: Bestilling af varer igangsættes, når der mangler varer i butikken eller på lageret. Dette brugsmønster udføres af butiksmedarbejderen og tømmerhandelsmedarbejderen, som har hver deres ansvar for bestemte varegrupper.

Objekter: Vare, kunde, medarbejder

Funktioner: Bestilling af vare(r)

Modtagelse af varer

Brugsmønster: Modtagelse af varer foregår på lagret, hvor lagermedarbejderen scanner varerne ind i systemet.

Objekter: Medarbejder, vare

Funktioner: Modtagelse af vare(r)

Oprettelse af varer

Brugsmønster: Oprettelse af varer igangsættes ved, at der er kommet nye varer hjem, som skal oprettes i systemet. Derudover kan dette brugsmønster også ske, hvis en vare skal genoprettes.

Objekter: Medarbejder, vare(r)

Funktioner: Oprettelse af vare(r)

Medarbejderadministration

Brugsmønster: Medarbejderadministration indeholder brugsmønstre som oprettelse, sletning og ændring af medarbejdere, og denne igangsættes kun af en administrator. Administratoren starter med at oprette en medarbejder. Derefter kan der ændres på medarbejderinformationen, hvis det er nødvendigt. Når den pågældende medarbejder ikke arbejder hos tømmerhandelen mere, kan administratoren slette personen.

Objekter: Administrator, medarbejder

Funktioner: Opret medarbejder

Bemærkning: Ikke alle funktioner er sat på, da listerne over funktionerne ville blive for komplekse. De fremgår i afsnittet under Funktioner.

3) Funktioner

Funktioner er udarbejdet med udgangspunkt i brugsmønstre og systemdefinitionen. For at klarlægge hvilke funktioner, systemet skal indeholde stilles spørgsmålet: Hvad skal systemet gøre? Ud fra dette spørgsmål er der opstillet følgende liste af funktioner, hvor hver især er vurderet ud fra deres kompleksitet, det vil sige, hvor kompliceret funktionen vil være at udvikle. Der skelnes mellem simpel, middel, kompleks og særdeles kompleks. Derudover vælges hvilken type funktionen er, og her sondres mellem fire typer signalering, opdatering, beregning og aflæsning.

a) Komplet funktionsliste

Funktion	Kompleksitet	Type
Opret ordre/tilbud	Medium	Opdatering
Opret udlejningsordre (reservation)	Medium	Opdatering
Ændre udlejningsperiode	Simpel	Opdatering
Reserver udlejningsvare	Medium	Signalering
Tilføj kunde (kundenummer eller 999)	Simpel	Opdatering
Tilføj varer (ordrelinjer)	Simpel	Opdatering
Modtagelse af betaling	Simpel	Opdatering
Annullere ordre/tilbud/udlejningsordre	Simpel	Opdatering
Opret kunde	Medium	Opdatering
Ændre kundeinformation	Simpel	Opdatering
Slet kunde	Simpel	Opdatering
Opret vare/udlejningsvare	Medium	Opdatering
Bestilling af varer	Medium	Opdatering
Slette vare/udlejningsvare	Simpel	Opdatering
Opret medarbejder	Medium	Opdatering
Ændre medarbejderinformation	Simpel	Opdatering
Slette medarbejder	Simpel	Opdatering

Modtagelse af varer	Simpel	Opdatering
---------------------	--------	------------

Specifikation af hvilke funktioner ordreadministrationen skal understøttes af:

Det vælges at lave en beskrivelse af hvilke funktioner der understøtter ordreadministrationen, for netop at skitsere overfor de uerfarne brugere, hvad det er systemet skal gøre, når de interagerer med systemet. Beskrivelsen kan give en forståelse for, hvordan systemet arbejder, når der oprettes en ordre. Førhen havde Norup Tømmerhandel og Byggecenter problemer med at have et overblik over varebeholdningen, da de ikke havde lagerstyring. Dette skal dette system løse. Derfor er det vigtigt at præcisere, at når brugerne indtaster en ordre og tilføjer varer til systemet, opdaterer systemet automatisk varebeholdningen. Medarbejderne må derfor ikke som i "de gode gamle dage" give kunden en ekstra varer med uden beregning uden at tæste varen ind i systemet, da systemet dermed ikke vil vise et retvisende billede af, hvad der ligger på lager af den pågældende vare.

Dermed beskrives der, hvilke funktioner systemet foretager, når medarbejderne indtaster en ordre i systemet.

"Opret ordre" er en funktion i tabellen, der understøtter brugsmønstret ordreadministration, hvor en butiksmedarbejder eller tømmerhandelsmedarbejder interagerer med systemet. Når butiksmedarbejderen eller tømmerhandelsmedarbejderen ønsker at indtaste en ordre, trykker de på en tast, hvorefter systemet skal indeholde funktionen, at den automatisk opretter en ordre. Når ordren er oprettet, skal systemet indeholde en funktion, hvor der automatisk bliver oprettet ordrelinjer, når medarbejderen scanner varen. Når disse varer bliver tilføjet til ordren, skal systemet automatisk opdatere varebeholdningen, og dette fremgår dermed også som en funktion. Derudover skal systemet indeholde en funktion, at der skal kunne tilføjes en kunde til ordren, så når medarbejderen indtaster et kundenummer eller 999 skal systemet automatisk tilføje kunden til ordren. Dernæst skal systemet kunne afslutte ordren ved at betalingen gennemføres. Det skal også være muligt for medarbejderen at

slette ordren, så når medarbejderen trykker på knappen "annuller" skal systemet annullere ordren.

Vurder systematisk

For at sikre, at funktionslisten dækker systemets samlede funktionalitet, og at funktionerne er indbyrdes konsistente, er der tjekket op med systemdefinitionen og modellens klasser og hændelser. I følgende afsnit gives en beskrivelse af funktionslisten sammenholdt med systemdefinitionen. Fra systemdefinition: *"Systemet skal være et planlægningsværktøj i forhold til lagerstyring og udlejning af varer, så virksomheden har styr på hvilken kunde, der har hvilken udlejningsvare"*. Dette opfyldes ved, at når medarbejderne udfører følgende: sælger varer (i form af "opret ordre"), bestiller varer, returnerer varer fra kunder og modtager varer fra leverandører, indeholder systemet den funktion, at den automatisk opdaterer varebeholdning, så medarbejderen altid er klar over, hvad der er på lager. Derudover ses det for at håndtere udlejningsproblemet, indeholder systemet funktionerne: "opret udlejningsordre", "ændre udlejningsperiode" og "reserver udlejningsvare". Dermed er alle udlejningsordre inde i systemet, og systemet indeholder derudover denne funktion, at hvis en bestemt udlejningsvare er udlejet, er den reserveret i systemet, som vises ved signalering, og den kan dermed ikke udlejes til andre. *"Systemet skal håndtere kundeoplysninger"*. Det ses i funktionslisten, at systemet kan udføre funktioner "opret kunde", "ændre kundeinformation" og "slet kunde".

4) Brugergrænsefladen

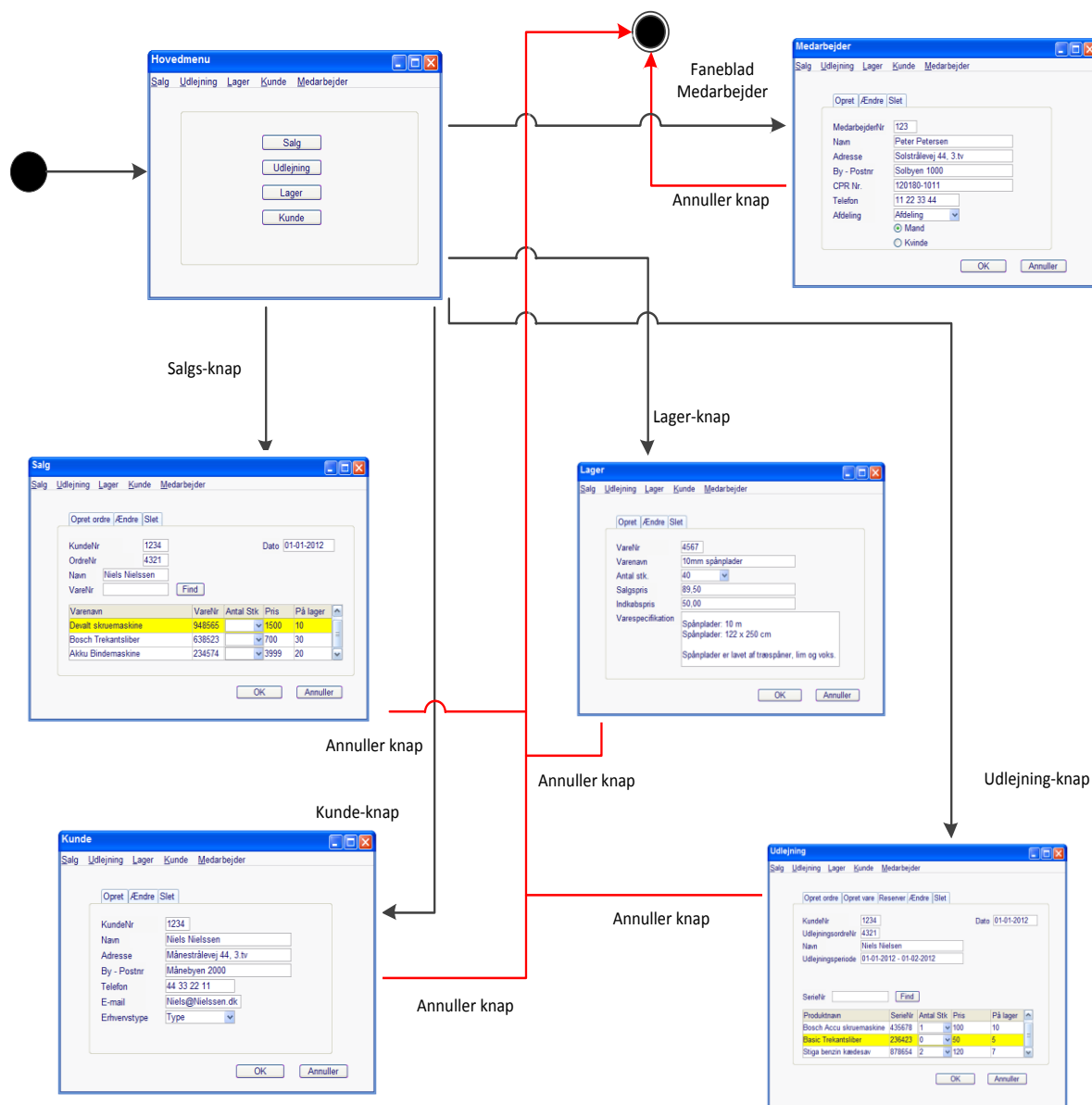
For at kunne fastlægge grænsefladens elementer tages der udgangspunkt i funktioner, brugsmønster, og modellen af problemområdet. Efter fastlæggelsen bestemmes grænsefladens elementer, og det ender ud i en vurdering af disse. Overordnet er det vigtigt, at brugergrænsefladen har et let og overskueligt udseende. Der skal helst ikke være for mange forskellige elementer, som kan distrahere brugere og derved forårsage fejl. Derudover skal det helst ikke tage for lang tid at lære og styre. Valg brugergrænsefladedialogform er: *Mønster for Menuvalg og Skemaudfyldelse*. *Menuvalg* er til navigeringen og *Skemaudfyldelsen* til indtastning af data.

Liste af grænsefladeelementer*Vinduer*

Hovedmenu
Salg
Udlejning
Kunde
Lager
Medarbejder

Navigeringsdiagram

For at skabe et overblik over relationerne mellem de forskellige brugergrænseflade vinduer, er der blevet lavet et navigeringsdiagram. De forskellige vinduer bygger meget på den samme skabelon, men er tilpasset til det enkelte grænsefladebehov. Mest relevante funktioner kan aflæses direkte i de enkelte vinduer.



Detaljespecifikation

For at få et mere dybdegående indblik i de forskellige vinduer, er der lavet detaljespecifikation af nogle centrale grænsefladeelementer. I detaljer beskrives vinduernes indhold og hvilke forskellige funktioner, der er blevet brugt.

Inden brugerne kommer ind i vinduet "Hovedmenu", er der et log-in vindue, hvor medarbejderne bliver bedt om at køre sit magnetkort igennem og følgende vindue fremkommer:



1. Ved drop-down menuerne kan brugerne vælge mellem de forskellige menuer. Dog er det kun administratorerne, som kan komme ind på Medarbejder (password beskyttet).
2. De forskellige menuer, som vælges ved et enkelt klik. Dog fremgår Medarbejder ikke med en hurtig knap, da den kun er betinget administratorerne.

1. Ved fanebladene kan medarbejderne ved hjælp af et klik, skifte mellem: Opret ordre, ændre en ordre og sletning af en ordre.
2. Her ses stamoplysninger om en ordre. OrdreNr bliver genereret automatisk. Ved feltet KundeNr indtastes kundens nummer, hvorefter systemet automatisk generer kundens navn. Det er også i dette felt, at der indtastes 999, hvis kunden ønsker at betale kontant. Derudover er det muligt at scanne varerne eller søge efter varerne.
3. Ved afslutning af en ordre klikker medarbejderne på OK.
4. På hvilket som helst tidspunkt kan man annullere en ordre, hvis den ikke ønskes gennemført.

Udlejning

Salg Udlejning Lager Kunde Medarbejder

Opret ordre Opret vare Reserver Ændre Slet

KundeNr 1234 Dato 01-01-2012

UdlejningsordreNr 4321

Navn Niels Nielsen

Udlejningsperiode 01-01-2012 - 01-02-2012

SerieNr Find

Produkt navn	SerieNr	Antal Stk	Pris	På lager
Bosch Accu skruemaskine	435678	1	100	10
Basic Trekantsliber	236423	0	50	5
Stiga benzin kædesav	878654	2	120	7

OK Annuller

1. Ved fanebladene kan medarbejderne ved hjælp af et klik, skifte mellem: Opret udlejningsordre, ændre, reservere, ændre sletning af en udlejningsordre.
2. Stamoplysninger som er tilknyttet en udlejningsordre. Under SerieNr har medarbejderne mulighed for at lede efter en bestemt udlejningsvare. De funde varer fremkommer i tabellen, hvor det ses, hvor mange antal stk. kunden ønsker at udleje og pris pr. udlejningsdag.

Lager

Salg Udlejning Lager Kunde Medarbejder

Opret Ændre Slet 1

h 2

VareNr 4567

Varenavn 10mm spånplader

Antal stk. 40

Salgspris 89,50

Indkøbspris 50,00

Varespecifikation

Spånplader: 10 m
Spånplader: 122 x 250 cm
Spånplader er lavet af træspåner, lim og voks.

OK Annuller

Denne funktion optræder som vareinformation. Det vil sige, at det er her, at alle varer bliver oprettet, ændret og slettet, og det fremgår på "Salg" og "Udlejning", hvor mange enheder, der er på lager, så medarbejderne hurtigt kan oplyse kunden hvor mange enheder, der er tilgængelig på lageret, og kunden dermed kan købe.

1. Her kan der vælges mellem fanerne "oprette", "ændre" eller "slet" en vare.
2. Information om varen

5) Ikke-funktionelle krav

I dette afsnit ses der på non-funktionelle krav, som definerer de overordnede kvaliteter, der lægges vægt på ved systemet. Nedenfor ses en prioriteringsliste over non-funktionelle mål, som er prioriteret ud fra systemdefinitionen. Systemdefinitionen indeholder betingelsen, at der skal være et tæt samarbejde med de ansatte, og netop dette tages der udgangspunkt i, i forhold til de non-funktionelle krav.

	Meget vigtigt	Vigtigt	Mindre vigtigt	Irrelevant
Brugervenlig	X			
Sikkert	X			
Effektivt		X		
Korrekt		X		
Pålideligt		X		
Vedligeholdbart			X	
Testbart				X
Fleksibelt	X			
Forståeligt	X			
Genbrugbart				X
Flytbart				X
Integrerbart				X

Tabellen tager udgangspunkt i produktkrav indenfor non-funktionelle krav, idet der ses på de specifikke, ønskede egenskaber, som systemet skal have som helhed. Nedenfor er der uddybet de mål, der har fået topprioritet i tabellen.

Systemet skal være brugervenligt, idet Norup Tømmerhandel og byggecenter har alle aldersgrupper som kommende brugere, hvilket gør, at hver enkelt medarbejder skal kunne håndtere systemet. Dette mål retter hen imod mål, som lyder, at systemet også skal være forståeligt og fleksibelt, hvilket er hovedreglen for et godt system. I dette system er det vigtigt, at det er forståeligt i den grad, at det skal være enkelt formuleret og et enkelt sprog, der bliver brugt, sådan alle generationer kan følge med. Derudover skal det være fleksibelt, hvilket gør, at der videre i forløbet kan ændres i nuværende beslutninger. Ved hjælp af indkapsling kan der forsøges at begrænse konsekvenserne af sådanne ændringer i beslutningstagen, og det derfor er muligt at udskifte egenskaber med en ny tilsvarende egenskab.

Derudover står der i systemdefinitionen, at det interne netværk skal være sikkert. Derfor skal systemet være sikkert, sådan der sikres imod systemets integritet ved uheld eller beviste forsøg på at forårsage skade, hvor en løsning kunne være en virus-kontrol. Norup Tømmerhandel og Byggecenter har fortrolige oplysninger om kunder, leverandør og medarbejder, som ikke skal kunne misbruges.

Principperne betyder, at ovennævnte kriterier altid skal prioriteres og, at ingen helt kan ignoreres, med mindre det i den konkrete sammenhæng er irrelevant.

C. Databaser

Den tekniske platform:

For at vise forståelsen for at kunne oprette en database ud fra vores klassediagram, er der fremvist nogle enkelte SQL kommandoer. De forskellige SQL koder er for oprettelsen af nogle af klasserne, hvor vi har udvalgt følgende: Kunde, Ordre, Ordrelinje og Vare til salg. Derudover er der også vist de to SQL forespørgsels kommandoer. Til sidst er der vist relationsdiagrammet.

Oprettelse af klassen Kunde:

```
CREATE TABLE Kunde
(
  KundeNr int NOT NULL PRIMARY KEY,
  Navn VarChar(100) NOT NULL,
  Adresse VarChar(100) NOT NULL,
  TelefonNr int NOT NULL,
  Email VarChar(100) NOT NULL,
  By VarChar(20) NOT NULL
  Postnummer int Char (4)
  Dato Date,
  CVR int Char (8),
  EAN int Char (13)
)
```

Oprettelse af klassen Ordre:

```
CREATE TABLE Ordre
(
  OrdreNr int NOT NULL PRIMARY KEY,
  KundeNr int NOT NULL FOREIGN KEY REFERENCES Kunde(KundeNr)
  Dato Date NOT NULL,
  MedarbejderNr int
)
```

Oprettelse af klassen Ordrelinje:

```
CREATE TABLE Ordrelinje
(
  OrdrelinjeID int NOT NULL PRIMARY KEY
  OrdreNr int FOREIGN KEY REFERENCES Ordre(OrdreNr)
  VareNr int FOREIGN KEY REFERENCES Vare til salg(VareNr),
  Antal solgte enheder int NOT NULL,
  Salgspris int NOT NULL
)
```

Oprettelse af klassen Vare til salg:

```
CREATE TABLE Vare til salg
(
VareNr int NOT NULL PRIMARY KEY,
Varenavn TEXT,
Antal enheder på lager int NOT NULL,
Indkøbspris int NOT NULL,
Varespecifikation TEXT
)
```

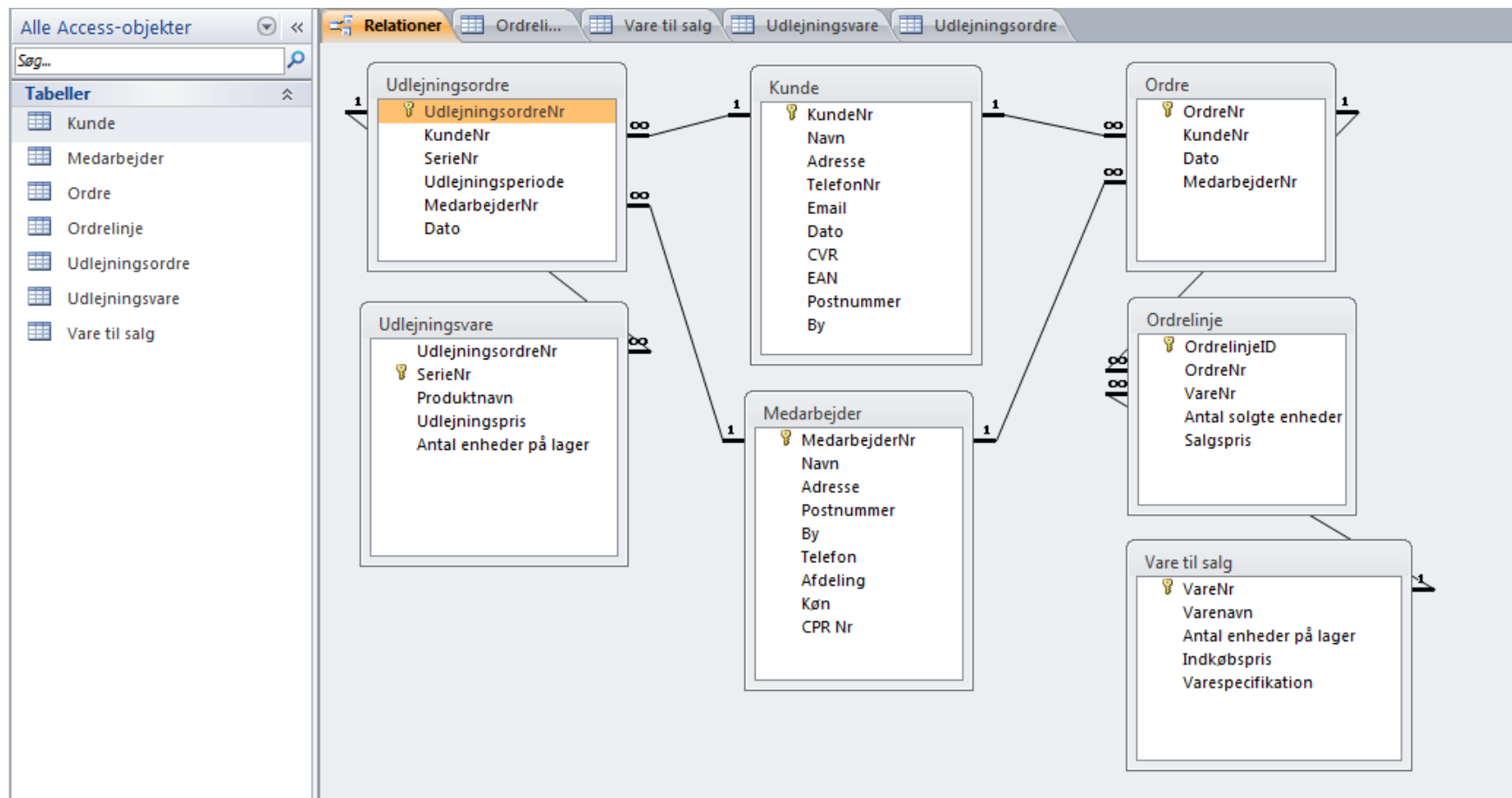
Forespørgsel A: Denne forespørgsel udarbejdes for "vare til salg". Der ses bort fra "udlejningsvare".

```
SELECT [Vare til salg].VareNr, [Vare til salg].Varenavn, SUM([ordrelinje].[Antal Solgte enheder])
FROM [Vare til salg] INNER JOIN (Ordre INNER JOIN Ordrelinje ON Ordre.OrderNr = Ordrelinje.OrderNr) ON
[Vare til salg].VareNr = Ordrelinje.VareNr
GROUP BY [Vare til salg].VareNr, [Vare til salg].Varenavn
ORDER BY SUM([ordrelinje].[Antal solgte enheder]) DESC;
```

Forespørgsel B:

```
SELECT Ordre.OrderNr, Sum(Ordrelinje.[Antal solgte enheder]*[Ordrelinje].[Salgspris]), Ordre.dato,
Kunde.Kundenr, Kunde.navn
FROM Kunde INNER JOIN ([Vare til salg] INNER JOIN (Ordre INNER JOIN Ordrelinje ON Ordre.OrderNr =
Ordrelinje.OrderNr) ON [Vare til salg].Varenr = Ordrelinje.VareNr) ON Kunde.Kundenr = Ordre.Kundenr
GROUP BY Kunde.Kundenr, Kunde.navn, Ordre.OrderNr, Ordre.dato
HAVING (((Sum([Ordrelinje].[Antal solgte enheder]*[Ordrelinje].[Salgspris]))>=2000))
ORDER BY Ordre.dato, Ordre.OrderNr;
```

Her ses relationsdiagrammet, som er udarbejdet ved hjælp de kommandoer, der er beskrevet ovenfor.



Overordnet arkitektur:

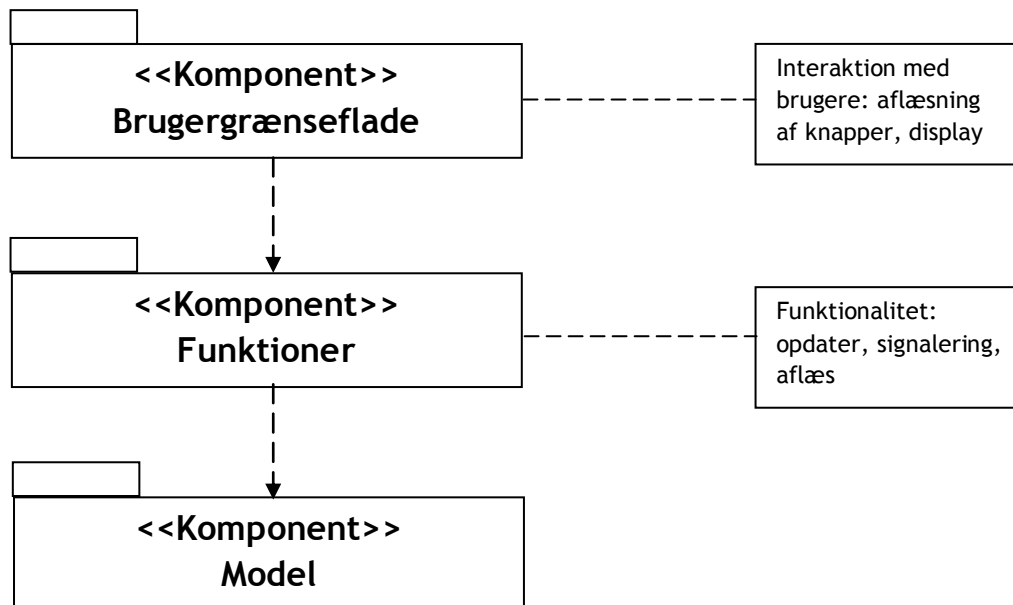
I designet af komponentarkitekturen tages der udgangspunkt i non-funktionelle krav. Først udforskes arkitekturmønstre. I vores gennemgang af arkitekturmønstrene har vi valgt at anvende en enkel grundarkitektur, som er et lagdelt mønster. Denne arkitektur består af grænseflade-, funktions- og modelkomponenter. Modelkomponenten indeholder modellen af problemområdet, og på et niveau ovenover lægges et komponent med systemets funktioner og øverst en grænsefladekomponent, som er brugergrænsefladen.

Næste aktivitet i komponentarkitekturen er at definere delsystemet. Systemet i vores projekt er forholdsvis simpel, og dermed er det tilstrækkeligt at anvende grundarkitekturen.

Aktiviteten "opdel komponenter" udbygges grundarkitekturen, hvor vi ser på komponenterne: model, funktion og grænseflade. Herefter ses på komponenter til indkapsling af den tekniske platform.

- **Model:** Lagerstyringssystem skal adskille varebeholdning og salg (ordre). Reservationssystem skal adskille udlejningsordre (det vil sige de maskiner, der er udlejet) og reservationer. Der er behov for en databasekomponent under modelkomponenten.
- **Funktion:** Opdatering er en central funktion i vores system, da varebeholdningen af både vare til salg og udlejningsvare skal kunne give et retvisende billede. Derudover er opdatering vigtig i form af, at vi skal vide hvilke reservationer, der er lavet på hvilken udlejningsvare.
- **Grænseflade:** Interaktionen mellem medarbejderen (brugerne) og funktionaliteten af systemet eller andre systemer.

Efter arbejdet med disse aktiviteter får vi følgende klassediagram med komponentarkitektur:

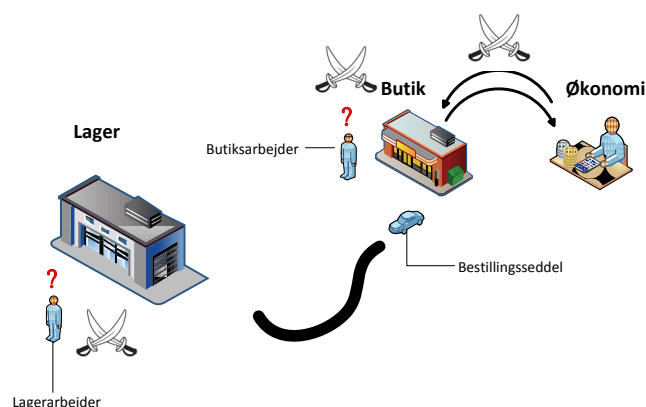


"Brugergrænsefladen" skal gøre brugeren i stand til at interagere med systemet. Det sker i form af, at medarbejderne (brugerne) i Norup Tømmerhandel og Byggecenter, kan aflæse hvilke knapper de skal trykke på for at kunne foretage sig en handling i systemet. Dette kan for eksempel være "oprette en ordre". Derudover skal displayet også opdateres, så det giver medarbejderne et retvisende billede af varebeholdningen. Det ses at komponenten "Funktioner" er forbundet med "Brugergrænsefladen" i form af en stiplede pil. Dette signalerer at komponenten "Funktioner" gør komponenten "Brugergrænsefladen" funktionel. Komponent "Modellen" er også forbundet med komponenten "Funktioner" i form af en stiplede linje. Komponent "Model" holder styr på de objekter, som skal understøttes af funktioner, og er dermed forbundet med komponenten "Funktioner".

Del B: Logbog

Foranalyse

For at forstå situationen startede vi med at liste de problemer op, som beskrevet i gruppeopgaven. Derefter snakkede vi om, hvilken type virksomhed, der reelt har sådan nogle problemer. Vi fandt frem til, at vi vil arbejde med en "Tømmerhandel og Byggecenter". For at forstå arbejdsgangen i virksomheden, har vi været i en lokal XL-Byg og set, hvordan det fungerer, hvorefter vi tegnede vores rige billede. Vores første rige billede som set herunder var vores første udkast, hvor vi tog udgangspunkt i en kunde, der købte en vare i tømmerhandlen for derefter at gå ud på lageret.



Da vi skulle beskrive vores rige billede, fandt vi frem til, at strukturerne i billedet manglede at blive præciseret, og vi redigeret dermed i strukturen inde i Tømmerhandlen og Byggecentret og på lageret.

Udvælgelse af kandidater til systemdefinitionen er gjort ud fra de generelle vurderingskriterier. Efter udvælgelsen har vi tilføjet "faktura" til vores systemdefinition, da den ligger indenfor problemområdet. Dermed har vi modificeret systemdefinitionen. Derudover havde vi en "bestillingsseddel" som kandidat, men den fandt vi irrelevant og slettede denne klasse.

Først beskrev vi klasserne enkeltvis og snakkede om, hvad de skal indeholde. Derefter lavede vi tabellen, for at få et overblik over, hvilke hændelser, der tilhører hvilke klasser. Dermed blev det mere overskueligt at beskrive hændelserne.

Den første:

Systemdefinition

Et it-system, der bruges til at styre lageret, økonomien og udlejningen af vare, der anvendes af medarbejderne, ledelsen og de administrative personer. Systemet skal primært være et planlægningsværktøj i forhold til lagerstyring og økonomistyring og sekundært et kommunikationsværktøj i forhold til udlejning af vare, så virksomheden har styr på hvilken kunde, der har hvilken udlejningsvare. Systemet skal baseres på PC'ere forbundet med et netværk. Systemet skal implementeres i tæt samarbejde med de ansatte, hvor udviklerne er interesseret i en løbende feedback fra brugerne.

Analyse af problemområdet

Den 22. september

Efter arbejdet i tirsdag (den 20. september) har vi hver især tænkt over, hvordan vi kan løse problemet med hændelserne "at bestille", "at reservere", "at udleje" og "at sende". Vi valgte i tirsdag at tilføje en klasse mere, der hed "ordre", som skulle sammensættes med hændelsen "at bestille", idet Tømmerhandlen modtager mange opkald fra kunder, der skal bestille varer. Vi har fundet frem til, at det vil give bedre overblik at lave en samlet klasse, der hedder "Ordre", der skal indeholde Faktura, Ordrekladde, Følgeseddel, Tilbud. Udlejning skal være en klasse for sig.

	Klasser							
Hændelser	Kunde	Reservation	Faktura	Ordre	Medarbejder	Udlejnings-vare	Vare	Computer
Reservere	✓	✓		✓	✓	✓		✓
Udleje					✓	✓		✓
Bestille	✓			✓	✓		✓	✓
Sende			✓		✓			✓
Udlevere					✓			
Vente	✓							
Styre								✓

Den 4. oktober 2011

Vi fandt frem til, at Tilbud, Ordrekladde, Faktura og Følgeseddel indeholder information om det samme, og har derfor samlet alle klasserne til én klasse - Ordre. Vi vil derfor modellere hvilke tilstande en ordre kan have i form af tilbud, ordrekladde, faktura og følgeseddel længere nede i opgaven.

	Klasser							
Hændelser	Kunde	Faktura	Ordrekladde	Tilbud	Følgeseddel	Udlejning	Medarbejder	Vare

Reservere	✓	✓	✓	✓	✓	✓	✓	✓
Udleje		✓	✓	✓	✓	✓	✓	✓
Bestille	✓	✓	✓	✓	✓			✓
Sende		✓	✓	✓	✓		✓	✓
Udlevere		✓	✓	✓	✓	✓	✓	
Vente	✓							
Styre							✓	✓

20. oktober: Udarbejdelse af tilstandsdiagrammer

Ved udarbejdelse af vores tilstandsdiagrammer, har vi fundet frem til, at vi mangler en masse hændelser i vores hændelsestabel - den er derfor blevet opdateret. Ved udarbejdelsen af hændelserne har vi haft en del problematikker, om hvilke hændelser, der skal involveres i hvilke klasser. Fx hvor betalingen skal indgå - om det skal være ved kunden eller ved ordren eller kan det indgå begge steder. Vi har fundet frem til en løsning, som ses i opgaven. Derudover har vi fundet frem til, at vi mangler en klasse "Lager", da systemet skal kunne opdatere varebeholdningen og samtidig holde styr på, hvor mange og hvilke varer, der ligger på lager. Denne klasse skal tilføjes.

Tilstandsdiagrammerne har virkelig givet os en masse arbejde, men samtidig har det gjort opgaven mere overskuelig for os, da vi har fundet frem til problemstillinger, som vi havde overset.

Den 28. oktober 2011

Vi har fundet frem til, at vi definerer klassen "Vare" som vores "Lager", og dermed vil vi ikke have en klasse der hedder "Lager", som skrevet i afsnittet ovenfor.

Vi har gennemgået vores klassesdiagram og opdaget, at vi har en mange-til-mange relation mellem ordre og vare. Vi har valgt at ophæve mange til mange relationen, og derfor kan der kun indgå ét varenummer på én ordre. Det vil sige at hvert enkelt søm har ét bestemt varenummer. Praktisk set vil det være mere realistisk at lave en overordnet varegruppe for den specifikke varegruppe.

Efter arbejdet med tilstandsdiagrammer og et gennemsyn af vores klassediagram, har vi valgt at lave en interaktion mellem "Ordre" og "Vare" for netop at undgå mange-til-mange relation, da det vil være urealistisk at have et specifikt varenummer for alle varer i butikken. Derfor har vi lavet klasse "Ordrelinje", som er et overordnet varenummer for et vis antal varer.

Vi har derfor tilføjet en klasse i vores klassediagram og lavet tilstandsdiagram over denne klasse.

Vi har valgt at lave to tilstandsdiagrammer for klassen "Ordre", da det giver bedre overblik. Dermed har vi lavet hvordan et tilstandsdiagram over hvordan det løber for henholdsvis kontokunde og kontantkunde.

Derudover har vi modelleret "udlejningsvare" og "ordre" både for kontantkunde og kontokunde med en hierarkisk tilstand, i hvilken hændelser "slette" og "annullere" kan afslutte et objekt på et vilkårligt tidspunkt af forløbet. Det vil sige at en "udlejningsvare" og en "ordre" kan henholdsvis "slettes" og "annulleres" på ethvert tidspunkt af sin livscyklus.

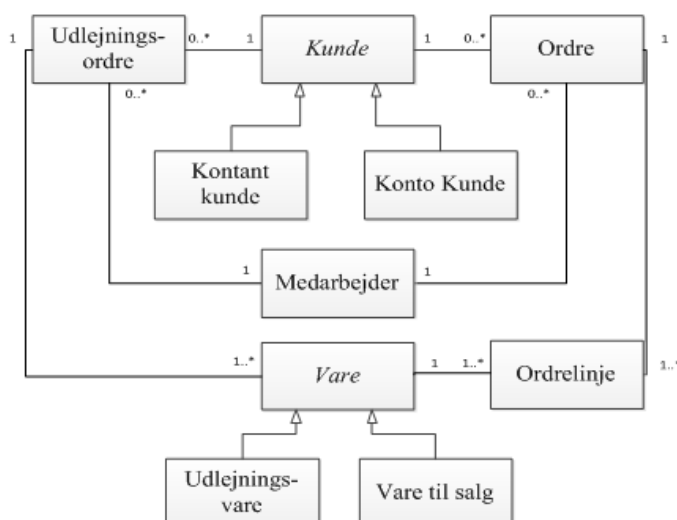
Den 7. november 2011

Efter nærmere vurdering har vi fundet frem til, at generaliseringsstrukturen med "Vare - Udlejningsvare" og "Vare til salg" skal slettes. I stedet har vi skabt to nye associeringsstrukturer:

Udlejningsordre - Udlejningsvare

Ordrelinje - Vare til salg.

Denne modificering af vores klassediagram er opstået, da vi indså, at "Udlejningsvare" har et serienummer, mens "Vare til salg" har et varenummer. Da de har forskellige attributter, skal de dermed optræde som selvstændige klasser.



Den 15. november:

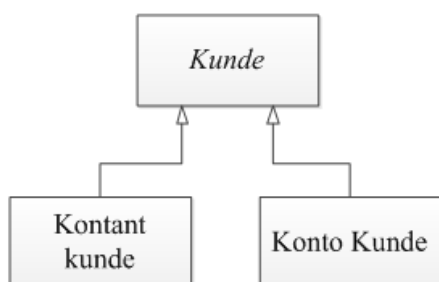
Efter klassepræsentationerne blev vi i dag gjort opmærksom på, at det ikke er realistisk at opdele klassen Kunde i subklasserne; Kontokunde og Kontantkunde, da en kunde både kan være en kontokunde og kontantkunde. Generaliseringsstrukturen viser, at kunden enten kan være kontokunde eller kontantkunde. Dette er en forkert skelnen, idet en kontokunde godt kan købe kontant og en kontantkunde kan godt blive oprettet i systemet, og dermed blive til en kontokunde.

Grunden til at vi havde opdelt klassen Kunde i to, var på grund af ved en kontokunde skal systemet kunne gemme dennes personlige oplysninger, hvorimod det skulle systemet ikke ved kontantkunden. Dermed samler vi disse to i en klasse, Kunde. Ved oprettelse af tilbud til en kontantkunde skal medarbejderen selv tilføje personlige oplysninger.

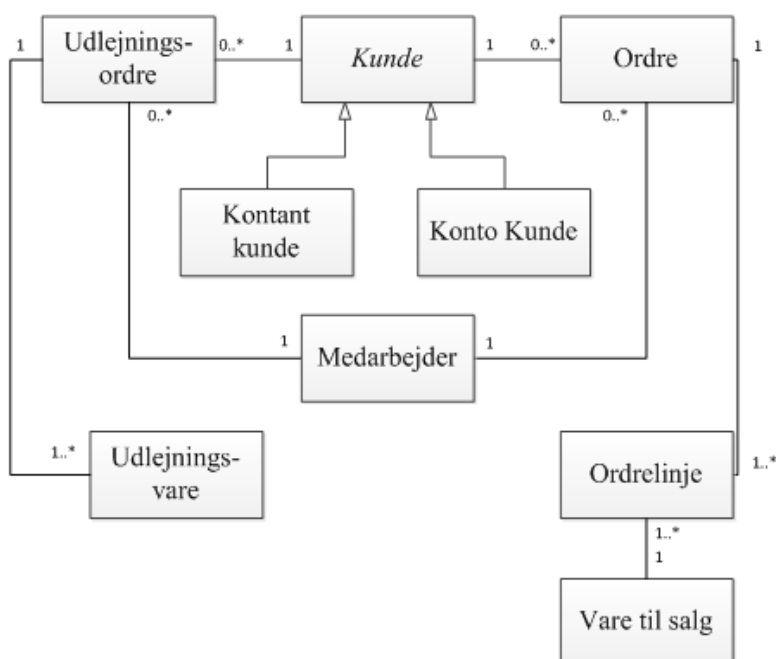
På baggrund af dette er følgende slettet:

Generaliseringsstrukturer: I klassediagrammet er der en generaliseringsstruktur.

- *Kunde*:



Kontantkunde og Kontokunde: Ved denne generaliseringsstruktur er klassen "Kunde", superklassen. Dernæst har vi nedenunder "Kontokunde" og "Kontantkunde" som subklasserne. Denne struktur er anvendt, da en kunde enten kan være en "Kontantkunde" eller en "Kontokunde".



Den 22. november 2011

I dag har vi ændret i vores tilstandsdiagrammer. Førhen havde klassen "Ordre" to tilstandsdiagrammer, der beskrev henholdsvis kontantkøb og kontokøb. Vi har valgt at samle disse to tilstandsdiagrammer i ét. Dette skyldes, at det giver et bedre overblik og nu ses der blot på hvilke to betalingsformer kunden kan vælge. Dermed skal

medarbejderen ikke selv tilføje oplysningerne til et tilbud (se logbog d. 15. november), da alle kunder skal oprettes i systemet før de kan foretage et køb eller modtage et tilbud. Det afgørende er om kunden vælger at betale kontant eller på konto - det vises ved selektionen.

Hændelserne "indsætte" og "udbetale" i tilstandsdiagrammet "Kunde" er slettet, da der ses bort fra administration af økonomi.

Analyse af anvendelsesområdet

Den 29. november 2011

Så blev der ændret i tilstandsdiagrammerne igen. Vi har valgt, at kunderne kun kan betale på konto, og vi vil først i arbejdet med anvendelsesområdet tage højde for kontantbetaling, da det praktisk talt skal være muligt i vores system.

Brugsmønstrene og aktørerne er blevet yderligere uddybet, og der er blevet lavet tilstandsdiagram over brugsmønstret "ordreadministration". Derudover er der blevet tildelt objekter og funktioner til hvert brugsmønster.

Den 6. december 2011

Vi har kigget hele opgaven igennem, og fået rettet småfejl hist og her. Vi har slettet "økonomi" i vores rige billede, da vores kravspecifikation ikke behandler dette problem. Dog vil Norup Tømmerhandel og Byggecenter forhåbentlig opnå en bedre økonomi ved at indføre lagerstyring. Vi er derfor klar til at gå direkte til funktioner og databaser.

Den 10. december 2011

Funktionslisten er opdateret, og i og med at systemet skal anvendes af uerfarne brugere er der lavet en specifikation af funktionen "opret ordre". Derudover er der skrevet et afsnit om sammenspillet mellem funktioner og systemdefinitionen.

Vi har også fået lavet non-funktionelle krav, hvor vi har taget udgangspunkt i systemdefinitionen. Vi har valgt at fokusere på produktkrav, da dette giver en god sammenhæng med systemdefinitionen.

Udover funktionslisten og de non-funktionelle krav, har vi også lavet grænsefladen i dag. Da vores system skal være nemt og overskueligt, har vi fokuseret på et design som ikke indeholder alt for mange unødige ting. Vi har valgt at vise de mest relevante vinduer, og derfor har vi undladt at vise vinduerne for fanebladene: opret..., ændre, slet osv. Derudover har vi også kun valgt at lave detailspecifikation af vinduerne: hovedmenu, salg og udlejning. Men kunne også sagtens have taget de resterende vinduer med: kunde, lager og medarbejder. Det har vi dog undladt, da de ligner meget hinanden, og derfor ville detailspecifikation bare blive en gentagelse. Sekvensdiagrammet har vi valgt ikke og medtage, da vi mener at den ikke hjælper yderligere til med forståelsen af grænsefladen. Navigeringsdiagrammet har en meget simpel opbygning, da alle vinduerne

tager udgangspunkt i hovedmenuen. Derfor er der ingen interaktion mellem de andre "undermenuer".

Eablering af teknisk platform

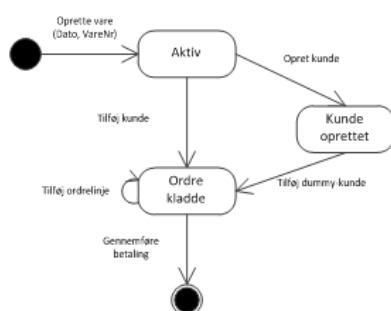
Med hjælp fra filen "RelationelleDatabaser_MIS2011" og hjemmesiden "http://www.w3schools.com/sql/default.asp" har vi været i stand til at oprette vores forskellige tabeller ud fra vores klassediagram. Måden, vi har transformeret vores klassediagram om til tabeller i databasen, var først ved at oprette tabellen med klassen som navn, så den fik det rigtige tabelnavn. Herefter har vi så taget udgangspunkt i hvilke attributter den enkelte klasse havde, og sat dem ind i tabellen. Derefter har vi så bestemt hvilken attribut, der var primærnøgle, og sat den nødvendige SQL kode på. Når vi havde styr på de grundlæggende elementer for en tabel, begyndte vi at sætte de forskellige: constraints og datatyper på. Efterfølgende fandt vi ud af hvilke forskellige relationer, der var mellem de enkelte tabeller, og satte derefter de for nødende constraints på. Det resulterede så i sidste ende med vores relationsdiagram.

Eablering af overordnet arkitektur

Eableringen af den overordnede arkitektur har vi taget udgangspunkt i problemområdet, funktioner og brugergrænsefladen. Vi har gået ind og kigget på hvert komponent og beskrevet, hvad det bidrager med til den overordnede arkitektur. Dernæst har vi beskrevet deres indbyrdes relationer.

Gennemgang af rapport inden aflevering - den 19. december

Tilstandsdiagram for brugsmønstret ordreadministration:



"Dette tilstandsdiagram er udarbejdet for at vise, hvad der sker, når kunden ikke er oprettet i systemet. Hvis kunden ønsker at betale kontant, skal kunden først oprettes som sker ved hjælp af en dummy kode (999). Dermed tror systemet, at kunden figurerer som en kontokunde i systemet. Dette er blot en kode, som tildeles kunderne for at gøre systemet mere realistisk, da det også skal kunne imødekomme kontantbetaling".

Vi har valgt at slette dette tilstandsdiagram, da vores specifikation af brugsmønstret "ordreadministration" beskriver forløbet bedre.