

# Vestbjerg Byggecenter A/S

## Case for 1. semesters projekt

*Datamatiker*

*University College Nordjylland*

*14. december 2016*



Casper Froberg Andersen  
Tobias Andersen  
Stefan Krabbe Johansen  
Mikkel Lindstrøm Paulsen  
Arne George Ralston  
Christoffer Lund Sørensen





**University College Nordjylland**

Sofiendalsvej 60

9200 Aalborg SV

UCN

<http://www.ucn.dk/Default.aspx>

**Titel:**

Vestbjerg Byggecenter A/S  
Case for 1. semesters projekt

**Tema:**

Semester projekt

**Projekt periode:**

25. november - 14. december 2016

**Projekt gruppe:**

Gruppe 5  
dmab0916

**Deltagere:**

Casper Froberg Andersen  
Tobias Andersen  
Stefan Krabbe Johansen  
Mikkel Lindstrøm Paulsen  
Arne George Ralston  
Christoffer Lund Sørensen

**Vejleder:**

Gunhild Andersen  
Lise Klitsgaard  
Istvan Knoll

**Antal kopier:**

6

**Antal sider:**

79

**Afsluttet:**

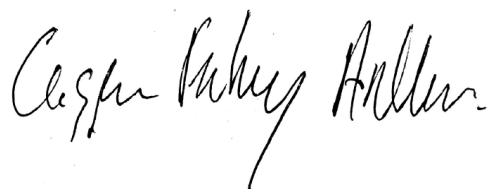
14. december 2016

**Abstract:**

This report is a product written by group five from dmab0916 at University College of Northern Denmark. The report is based on an assignment which purpose was to analyse, design and develop an entirely new system. Therefore the content of this report is based on the principles taught in the first semester of the AP graduate education on UCN.

This includes a preliminary investigation of the company, which comes from the business, system design, from system development, and the programming – from the programming course.

*Rapportens indhold er frit tilgængeligt, men offentliggørelse (med kildeangivelse) må kun ske efter aftale med forfatterne.*



---

Casper Froberg Andersen  
1061832@ucn.dk



---

Tobias Andersen  
1061912@ucn.dk



---

Stefan Krabbe Johansen  
1062358@ucn.dk



---

Mikkel Lindstrøm Paulsen  
1061928@ucn.dk



---

Arne George Ralston  
1061679@ucn.dk



---

Christoffer Lund Sørensen  
1061855@ucn.dk



# Forord

Rapporten er udarbejdet af en gruppe 1. semesters studerende på datamatikeruddannelsen ved University College Nordjylland (UCN). Formålet med denne rapport har været at analysere en virksomheds behov og ud fra dette designe og programmere et nyt system, som opfylder de krav der identificeres.

Der rettes en tak til Gunhild Andersen, Lise Klitsgaard og Istvan Knoll for god vejledning gennem projektperioden.

Der er gjort brug af UCN's SVN server til versionsstyring. Linket til denne er [https://kraka.ucn.dk/svn/dmab0916\\_1Sem\\_Project\\_5](https://kraka.ucn.dk/svn/dmab0916_1Sem_Project_5) og den endelige rapport er på revisionsnummer 176.

## Læsevejledning

Der vil gennem rapporten fremtræde kildehenvisninger efter Vancouver-metoden, hvilket vil sige at i teksten kan der f.eks. stå [1], hvilket refererer til kilde nummer 1 i litteraturlisten. I litteraturlisten er bøger og artikler angivet med forfatter, titel og forlag, mens internetsider er angivet med forfatter, titel og dato. Figurer og tabeller er nummereret i henhold til kapitel. Det vil sige i kapitel 2 har figur nummer 1 figurnummer 2.1. Derudover findes der under hver figur og tabel forklarende tekst.

University College Nordjylland, 14. december 2016





# Indhold

<b>1</b>	<b>Indledning</b>	<b>3</b>
<b>2</b>	<b>Metode</b>	<b>5</b>
2.1	Adizes' lederroller . . . . .	5
2.2	SWOT-analyse . . . . .	6
2.3	McGregor's X- og Y-teori . . . . .	7
2.4	Ansoffs vækstmatrix . . . . .	7
2.5	Information Economics . . . . .	8
2.6	Forandringsstrategier . . . . .	10
2.7	Medarbejder - opgave - mål tabel . . . . .	10
2.8	Workflow . . . . .	11
2.9	Use case . . . . .	12
2.10	Domænemodel . . . . .	13
<b>3</b>	<b>IT-forundersøgelse</b>	<b>15</b>
3.1	Analyse af nuværende situation . . . . .	15
3.2	Strategianalyse . . . . .	22
3.3	IT-strategi . . . . .	23
3.4	Applikationer og information . . . . .	26

3.5	Organisering af IT-funktionerne . . . . .	27
3.6	IT handlingsplan . . . . .	27
3.7	Cost-benefit analyse - Parker Benson . . . . .	30
3.8	Brugerdeltagelse . . . . .	31
3.9	Konklusion på forundersøgelsen . . . . .	31
<b>4</b>	<b>Problemdefinition</b>	<b>33</b>
4.1	Problemafgrænsning . . . . .	33
4.2	Problemformulering . . . . .	33
4.3	Succeskriterier . . . . .	34
<b>5</b>	<b>Inception</b>	<b>35</b>
5.1	Interview . . . . .	35
5.2	Mock-ups . . . . .	35
5.3	Medarbejder - opgave - mål tabel . . . . .	36
5.4	Workflow . . . . .	41
5.5	Use case diagrammer . . . . .	43
5.6	Use case beskrivelser . . . . .	45
5.7	Domænemodel . . . . .	47
5.8	Use-case prioritering . . . . .	48
5.9	Use case - Fully dressed . . . . .	49
<b>6</b>	<b>Elaboration</b>	<b>53</b>
6.1	Arkitektur . . . . .	53
6.2	System-sekvensdiagrammer . . . . .	54
6.3	Operationskontrakter . . . . .	56
6.4	Kommunikationsdiagrammer . . . . .	57

Indhold	1
6.5 Designklassediagrammer . . . . .	59
6.6 Kode . . . . .	62
<b>7 Diskussion</b>	<b>69</b>
7.1 Use cases . . . . .	69
7.2 Systemet . . . . .	69
<b>8 Konklusion</b>	<b>71</b>
8.1 Gruppeevaluering . . . . .	71
<b>Litteratur</b>	<b>73</b>
<b>A Udregninger til cost benefit</b>	<b>75</b>
<b>B Designklassediagram</b>	<b>77</b>
<b>C Brugervejledning</b>	<b>79</b>



# 1 | Indledning

Denne rapport er udarbejdet i forbindelse med et projekt på første semester på datamatiker uddannelsen på UCN af gruppe 5 fra klassen dmab0916. Projektet handler om at designe og implementere et IT-system til en fiktiv kunde ved navn Vestbjerg Byggecenter A/S. Der er på forhånd blevet opstillet en case, som beskriver de fleste af virksomhedens ønsker og krav til det nye system, samt en generel beskrivelse af virksomheden og dens opbygning.

Projektet er opbygget om en iterativ arbejdsproces ved navn Unified Process (UP), som er en agil tilgang til objekt-orienteret software design og -udvikling. Rapporten er således opbygget om de forskellige iterationer som projektet har gennemløbet. De forskellige iterationer er: it-forundersøgelse, som indeholder en virksomhedsanalyse hvor der fastlægges og beskrives krav for det ønskede system, inceptionfasen, hvor de præliminære systemkrav brainstormes og identificeres samt elaborations, hvor de højest prioriterede use case behandles.

Den iterative cyklus er baseret på løbende udvidelse og tilpasning af systemet gennem projektets iterationer. Hver iteration vil give feedback, som bruges til at udbygge systemet samt revidere tidligere beslutninger.

På baggrund af forundersøgelsen udarbejdes en problemformulering, som vil specificere projektets udfordringer og tilsigtede læringsmål. Sluttelig vil der i konklusionen vurderes, hvorvidt disse er opfyldt, samt en generel evaluering af projektet.



## 2 | Metode

I dette kapitel vil der være en kort beskrivelse og forklaring af de metoder og teorier der benyttes i rapporten.

### 2.1 Adizes' lederroller

Adizes' teori om lederroller bygger på, hvilke personlige egenskaber der kræves for at få virksomheden til at fungere, samt udvikling af virksomheden på længere sigt. Effektiv ledelse kræver, ifølge Adizes, at alle rollerne er dækket.

De fire roller Adizes betegner er:

- Producentrollen
- Administratorrollen
- Entreprenørrollen
- Integratorrollen

En oversigt over disse rollers egenskaber kan ses på figur 2.1.

		Fokuserer på	
		Produkt	Proces
Fokuserer primært på	Langt sigt	<b>Entrepreneurrollen</b> <ul style="list-style-type: none"> <li>• Kreativ og innovativ</li> <li>• Finder nye produkter og nye metoder</li> <li>• Tænkter strategisk</li> <li>• Stiller spørgsmål til det bestående</li> <li>• Risikovillig</li> <li>• Udvikling</li> </ul>	<b>Integratorrollen</b> <ul style="list-style-type: none"> <li>• Integrerer i et fællesskab</li> <li>• Indgår kompromiser</li> <li>• Skaber motivation og korpsånd</li> <li>• Leder gennem teamwork</li> <li>• Skaber udvikling hos medarbejderne</li> <li>• Skaber sammenhold</li> </ul>
	Kort sigt	<b>Producentrollen</b> <ul style="list-style-type: none"> <li>• Resultat- og handlingsorienteret</li> <li>• Stort præstationsbehov</li> <li>• Tager beslutninger</li> <li>• Flittig og travl</li> <li>• Medarbejderne bliver hjælpere</li> <li>• Faglig viden</li> </ul>	<b>Administratorrollen</b> <ul style="list-style-type: none"> <li>• Opstiller mål og regler</li> <li>• Kontrollerer og evaluerer</li> <li>• Skaber systematik</li> <li>• Analyserer sig frem til den rigtige løsning</li> <li>• Bureaukrati</li> <li>• Ordenssans</li> </ul>

Figur 2.1: Adizes' 4 lederroller [1, s. 208].

For at kunne tildele "roller" til den enkelte leder bruges der et *kodesystem* med tre niveauer.

- Stort bogstav betyder stor varetagelse af rollen
- Lille bogstav betyder nogen varetagelse af rollen
- Et O betyder, at lederen overhovedet ikke dækker rollen

Ud fra dette kan der tildeles fire bogstaver til en leder, eks. PAEI, PaEi, eller pOeI. Det er som regel umuligt at have store bogstaver ved alle rollerne, da nogle er i strid med hinanden [1, s. 208-209].

## 2.2 SWOT-analyse

En SWOT-analyse består af fire dele: stærke sider (*Strengths*), svage sider (*Weaknesses*), muligheder (*Opportunities*) og trusler (*Threats*).

Delene består af både interne og eksterne forhold. De interne forhold tager udgangspunkt i stærke og svage sider, hvor der dannes et overblik over, hvilke styrker og svagheder virksomheden selv har. De interne forhold karakteriseres ved, at det er noget, som virksomheden selv har indflydelse på.

De eksterne forhold, som består af muligheder og trusler, benyttes til at belyse, hvordan eksterne elementer kan påvirke firmaet. Modsat de interne forhold har virksomheden ingen indflydelse på de eksterne forhold [1, s. 362-363].



Et eksempel på hvordan en SWOT-analyse kan sættes op er vist på figur 2.2.

Interne forhold	
Stærke sider (Strengths)	Svage sider (Weaknesses)
Eksterne forhold	
Muligheder (Opportunities)	Trusler (Threats)

Figur 2.2: SWOT eksempel

## 2.3 McGregor's X- og Y-teori

Ved en leders menneskesyn forstås, hvordan en leder opfatter og tror på almene menneskelige egenskaber hos medarbejderne. På figur 2.3 ses de to forskellige menneskesyn, som McGregor mener, der findes [1, s. 205].

McGregors X- og Y-syn		
Lederens antagelser	X-syn	Y-syn
Hvilket forhold har de til det at arbejde?	Føler ubehag ved at arbejde, og forsøger derfor at undgå det	Arbejde er lige så naturligt for mennesket som fx leg og hvile
Hvilke ønsker har de til arbejdets indhold?	Foretrækker at blive dirigeret til at udføre simple rutine-opgaver	Frihedsgrader til at udføre udviklende arbejdsopgaver sådan, at medarbejderen kan styre sig selv i retning mod mål, som han selv går ind for
Hvilke behov ønsker de at få dækket via arbejdet?	Tryghed og sikkerhed samt de fysiske behov	Ego og selvrealisering ved at udvikle sig via jobbet
Er de villige til at påtage sig et ansvar?	Nej – vil hellere have tryghed ved at andre tager ansvaret	Ja – hvis det er i forbindelse med et meningsfyldt arbejde

Figur 2.3: McGregors X- og Y-syn [1, s. 205].

## 2.4 Ansoffs vækstmatrix

Igor Ansoffs vækstmatrix, illustreret i figur 2.4, bruges til at analysere, hvilke muligheder en virksomhed kan bruge for at vækste. Ifølge Ansoff er der fire følgende muligheder [1, s. 370-371]:

- Markedspenetrering
  - Ved denne strategi søger virksomheden at sikre et øget salg af eksisterende produkter på det nuværende marked ved at fortrænge konkurrenterne.
- Markedsudvikling
  - Denne strategi betegner, at virksomheden forsøger at afsætte sine nuværende produkter til nye markeder.
- Produktudvikling
  - Ved produktudvikling sker der en udvikling af nye varer til eksisterende markeder. Når denne strategi bruges, tilstræber virksomheden at udnytte "first mover-effekten", som handler om at være den første med et nyt produkt.
- Diversifikation
  - Ved diversifikation udvikles nye produkter til helt nye kundegrupper og markeder. Det vil sige, at virksomheden ikke kun bevæger sig ind på et nyt marked, men også med et helt nyt produkt.

Ansoffs vækstmatrix			
		Produkter	
		Nuværende	Nye
Markeder	Nuværende markeder	Markedspenetrering	Produktudvikling
	Nye markeder	Markedsudvikling	Diversifikation

Figur 2.4: Ansoffs vækstmatrice [1, s. 370].

## 2.5 Information Economics

Information Economics, som er udviklet af Parker og Benson, benyttes til at definere virksomhedens relationer mellem investering og forbedring af deres fremdrift og systemets udvikling. Dette gøres ved, at der eksplicit bliver evalueret mulige faktorer.

I figur 2.5 vises en tabel over fordelingen mellem de positive og negative faktorer.

	Forretningsområde (Business Domain)	IT – område (IT – domain)
Positive faktorer, der understøtter valgmuligheder mellem de udvalgte projekter	Strategic match Competitive advantage Management information Competitive response	Strategic IS architecture
Negative faktorer, der forringer valgmuligheder mellem de udvalgte projekter	Organisational risk	Definition uncertainty Technical uncertainty IS infrastructure risk

Figur 2.5: Diverse faktorer som tages i betragtning [2, s. 3]

De positive faktorer forstås, som noget virksomheden anser som værende af værdi, hvorimod de negative faktorer repræsenterer en usikkerhed eller risiko for firmaet [2, s. 3].

Disse faktorer opstilles i en tabel, hvor der tildeles points til hvert enkelt faktorer og dens vigtighed med henblik på tilknytning til virksomheden. Et eksempel på en tabel er vist på figur 2.6.

	Business domain						Technology domain				To tal
	ROI	SM	CA	MI	CR	OR	SA	DU	TU	IR	
Vægt	+10	+2	+2	+2	+1	-1	+3	-2	-2	-2	
Debitorstyring	40	8	4	6	2	0	3	0	0	0	63
E-handelssystem	30	10	8	6	5	-4	6	-6	-6	-6	43

Vurderingsfaktorer:

ROI = Return of investment  
 SM = Strategic match  
 CA = Competitive advantage  
 MI = Management information  
 CR = Competitive response  
 OR = Organizational or project risk  
 SA = Strategic IS architecture  
 DU = Definitional uncertainty  
 TU = Technical uncertainty  
 IR = IS infrastructure risk

Figur 2.6: Diverse faktorer som vurderes ud af vigtighed [2, s. 10]

## 2.6 Forandringsstrategier

Forandringsstrategi kendetegnes ved hvilke indsatsområder og metoder, som bliver anvendt i forhold til de grundlæggende antagelser, som findes i organisationen, og hvilken form for forandring organisationen ønsker [1, s. 397].

### Ekspertstrategi

Karakteriseres ved, at en eller flere eksperter gennemgår en analyse og diagnose, som ender ud i en problemformulering og et design for systemet [1, s. 397].

### Socioteknisk strategi

Socioteknisk strategi går på kompromis mellem de interne modsigende interesser for at udforme det nye system. Strategien tager udgangspunkt i at få brugerens sociale mål med systemet opfyldt. Dette gøres ved at interviewe brugeren [1, s. 398].

### Repræsentationsstrategi

Repræsentationsstrategi inddrager medarbejderne til udviklingen af programmet. Dette gøres ved at udvælge repræsentanter for de involverede brugere [1, s. 398].

### Deltagelsesstrategi

Deltagelsesstrategi ses som den humanistiske forandringsstrategi til udarbejdelse af systemet, da den varetager de brugere, som normalt ikke får så meget indflydelse i udviklingsfasen til systemet [1, s. 398].

I figur 2.7 ses en visuel fremvisning af diverse forandringsstrategier.

Forandringsstrategier i forhold til menneskesyn og organisationsopfattelse			
		Menneskeopfattelse	
		X-teori	Y-teori
Organisationsopfattelse	Harmoni	Ekspertstrategi	Repræsentationsstrategi
	Konflikt	Socioteknisk strategi	Deltagelsesstrategi

Figur 2.7: Forandringsstrategier i forhold til organisationsopfattelse og menneskesyn [1, s. 401]

## 2.7 Medarbejder - opgave - mål tabel

For at specificere et systems afgrænsninger og funktionalitet skal det overvejes, hvem de primære brugere er. Derefter skal det vurderes, hvilke mål der skal opfyldes af systemet. Disse mål brainstormes og uddybes i form af opgaver og steps

i opgaven.

Disse opgaver og mål identificeres, ved at tage udgangspunkt i sin IT-forundersøgelse eller selve interviewet med kunden. Ud fra dette identificeres kundens nuværende opgaver og de målbare resultater [3].

Resultatet af disse overvejelser opstilles i en medarbejder - opgave - mål tabel, som vist på tabel 2.1.

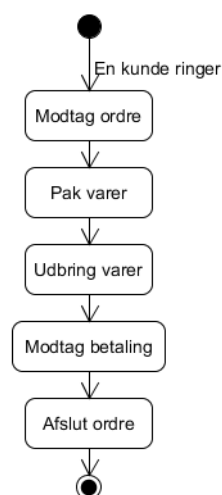
Medarbejder	Opgave	Mål	Step i opgave
Ekspedient	Modtage ny ordre	Registrer ordre	Finde varer kunde ønsker Tilføje varerne til ordren Tilføje kundeoplysninger Færdiggøre ordren Bekræftelse af ordren
...	...	...	...

Tabel 2.1: Eksempel på en medarbejder - opgave - mål tabel

## 2.8 Workflow

Et workflow diagram bruges til at vise den tidsmæssige rækkefølge over de opgaver, der er blevet identificeret i medarbejder - opgave - mål tabellen.

Et eksempel på et workflow diagram vises på figur 2.8.



Figur 2.8: Eksempel på et workflow.

## 2.9 Use case

En use case beskriver en række handlinger, som bliver udført af en aktør for at udføre et bestemt mål. Use cases beskrives med henblik på at identificere, klarlægge og organisere krav til systemet samt finde ud af, hvordan aktøren interagerer med systemet [4].

### 2.9.1 Use case beskrivelser

I rapporten bruges der to former for beskrivelser af use cases. Den korte form, brief, og den mere beskrivende, som kaldes fully dressed.

#### Brief

En brief beskrevet use case beskrives overordnet men komplet. Dette vil sige, at der beskrives, hvem der initierer use casen, de forventede systemhandlinger og responsen herpå, som giver systemet værdi for aktøren [4].

#### Fully-dressed

Fully dressed use cases beskriver, hvordan en opgave udføres med henblik på at nå til mål med use casen. Der bliver benyttet to forskellige stadier i en fully dressed: *Happy days* og *alternative flows*. Happy days beskriver, hvordan opgaven udføres, når alle forudsætninger for opgaven er som det forventes. Alternative flows beskriver situationer, som forhindrer direkte udførelse af opgaven.

En fully dressed beskrivelse kommer både med en præ- og en postbetingelse. Præbetingelsen beskriver hvilke betingelser, der skal være opfyldt før handlingen er mulig, og postbetingelsen beskriver hvad, der skal være opfyldt [5, s. 126].

### 2.9.2 Use-case diagram

Use-case diagrammer benyttes for at illustrere, hvilken aktør der interagerer med bestemte use-cases i systemet. Dette gøres for at skabe et overblik over, hvilke aktører der skal have adgang til bestemte funktioner i systemet [5, s. 158].

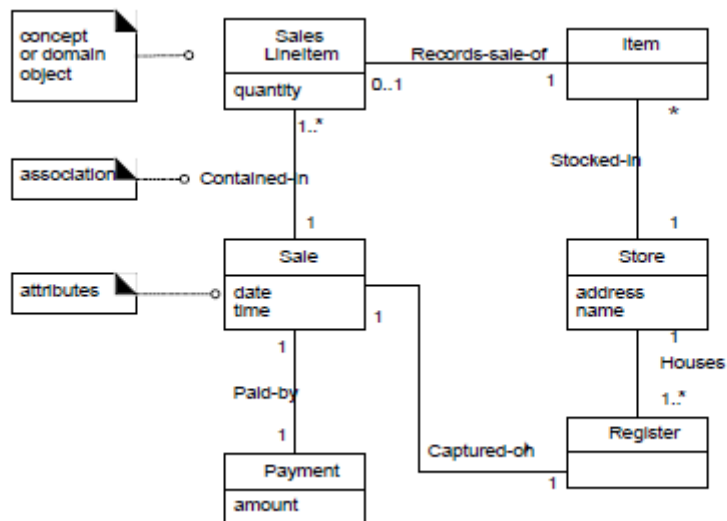
## 2.10 Domænemodel

En domænemodel er en visuel repræsentation af konceptuelle klasser. Domænemodellen benyttes som det første objekt-orienterede analyseredskab i systemudvikling [5, s. 128].

En domænemodel illustreres ved hjælp af et klassediagram som indeholder:

- attributter tilhørende klasserne
- konceptuelle klasser
- association mellem klasserne

Figur 2.9 viser et eksempel på en domænemodel med visuelle fremvisning af klasser, attributter og associering mellem klasser.



Figur 2.9: Domænemodel eksempel [5, s. 129]





## 3 | IT-forundersøgelse

I dette kapitel analyseres hvordan Vestbjerg Byggecenter A/S' nuværende interne systemer fungerer. Derudover analyseres strategiske muligheder, virksomhedens mål og strategi, it-strategi og -handlingsplan.

### 3.1 Analyse af nuværende situation

For at belyse hvordan den nuværende situation er for Vestbjerg Byggecenter A/S, gives der en kort beskrivelse af firmaets nuværende situation, navnlig med henblik på virksomhedens karakteristik, organisationsstruktur og kultur samt en lederkarakteristik.

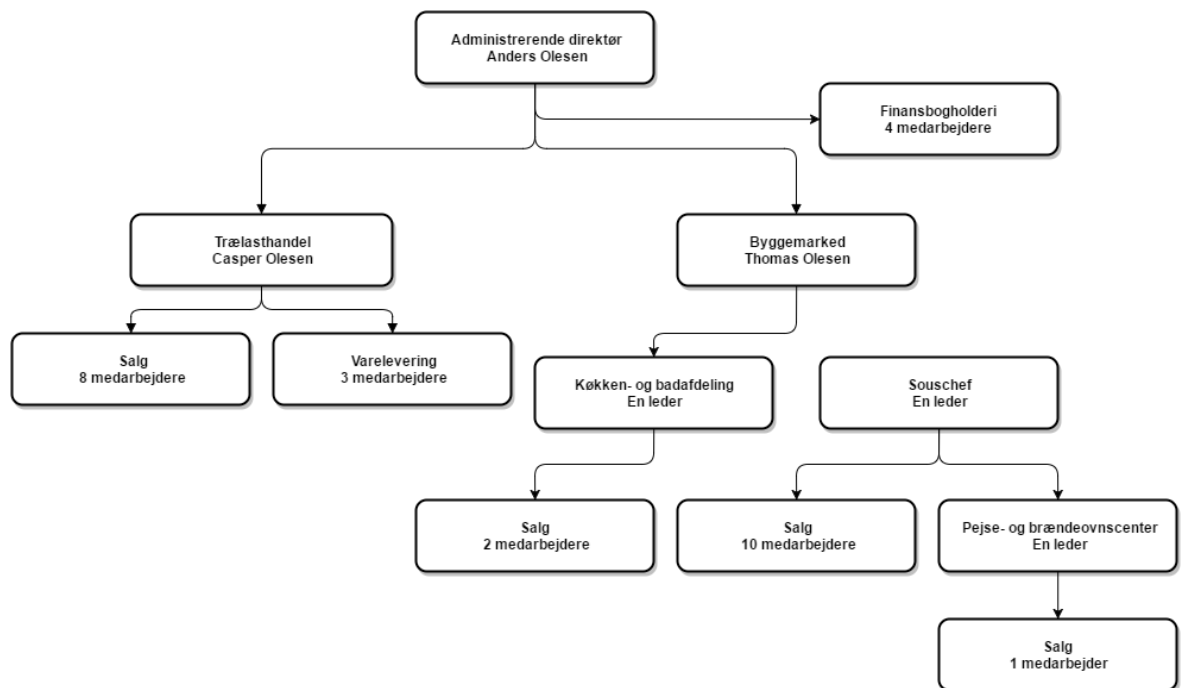
#### 3.1.1 Virksomhedskarakteristik

Vestbjerg Byggemarked blev stiftet i 1987 af Anders Olesen. I 2003 indtrådte hans sønner, Thomas og Casper, som en del af ledelsen. Samtidigt blev Vestbjerg Byggemarked til et aktieselskab. Vestbjerg Byggecenter A/S består af 33 fastansatte fordelt på to afdelinger. Den ene afdeling består af trælasthandel og den anden et byggemarked. Vestbjerg Byggecenter A/S har 1.000 faste kunder fordelt på 22 kundegrupper, som afgør hvor meget rabat en kunde får. Byggebranchen har sammenlagt 100.000 varenumre til rådighed, hvoraf Vestbjerg Byggecenter A/S lagerfører cirka 28.000 af disse. Disse varer er fordelt på 100 forskellige varegrupper. Ligesom kunderne er varerne fordelt på grupper, baseret på hvilken type vare det er.

### 3.1.2 Organisationsstruktur

I dette afsnit vil der blevet udarbejdet et overblik over Vestbjerg Byggecenter A/S' afdelinger. Derudover vil virksomhedens afdelingsprincip blive analyseret.

I figur 3.1 ses organisationsdiagrammet for Vestbjerg Byggecenter A/S som viser den vertikale og horisontale arbejdsdeling. Den vertikale arbejdsdeling viser, hvordan ledelsesansvaret er fordelt på virksomhedens fem hierarkiske niveauer. Det fremgår, at virksomheden er inddelt efter linje-stabs princippet [1, s. 280], hvor der er en klar ansvarsplacering i kraft af, at hver medarbejder kun har én direkte overordnet. Ydermere er basisorganisationen suppleret af en finansbogholderi stab. Den øverste ledelse (Anders Olesen) har et kontrolspænd på to, i kraft af de to hovedafdelinger; trælasthandel og byggemarked, som ledes af hans to sønner Thomas og Casper.



Figur 3.1: Organisationsplan over Vestbjerg Byggecenter A/S.

På den horisontale arbejdsdeling kan det ses, hvordan arbejdsopgaverne er fordelt i organisationen. Her ses det, at Vestbjerg Byggecenter A/S gør brug af objekt-princippet [1, s. 285], hvor virksomheden er opdelt efter produkter, hvilket i dette tilfælde er byggemarked, trælasthandel, køkken- og badafdeling samt pejse- og brændeovnscenter. Disse individuelle afdelinger (objekter) står hver især selv for

salg.

### 3.1.3 Vurdering af organisationsstruktur

Objektprincippet uddelgering af opgaveløsningen til de enkelte afdelinger gør det muligt at øge fokuseringen på et bestemt område, hvilket kan øge effektiviteten i kraft af, at det bliver lettere at fastsætte og overvåge mål i virksomhedens afdelinger. Denne specialisering ud fra objekter har dog den konsekvens, at virksomheden spreder salgsspecialisterne ud på forskellige afdelinger, hvilket medfører et tab i videnssynergien.

For at finde ud af hvordan ledelsen og medarbejderne omgås på arbejdspladsen, er det nødvendigt at undersøge de sociale dannelser og deres adfærd i Vestbjerg Byggecenter A/S. Dette gøres her, ved at analysere lederne i virksomheden.

#### Lederkarakteristik & menneskesyn

Dette afsnit gør brug af metoden beskrevet i afsnit 2.3. Det kan siges, at Anders Olesen har et y-syn, da han forsøger at motivere folk i hans arbejdsområde og giver meget frihed under ansvar til del-ledere og personer i kontorfunktionen. Casper Olesen virker også til at have et y-syn, da han interesserer sig for medarbejdernes velbefindende og holder samtaler, samtidigt går han også meget op i, at skabe en stærk kultur i virksomheden. Thomas Olesen kan siges at have en blanding af et x og y syn, da han er mere styrende end Thomas, men han vil også gerne hjælpe sine medarbejder med at udvikle sig.

#### Adizes lederroller

I dette afsnit vil der blive taget udgangspunkt i Adizes fire lederroller. Metoden til dette er beskrevet i afsnit 2.1. Anders og hans to sønner er blevet tildelt en profil ud fra de fire roller: *producentrollen*, *administratorrollen*, *entreprenørrollen* og *integratørrollen*.

Adizes 4 lederroller	
Anders Olesen	PaEI
Thomas Olesen	paeI
Casper Olesen	pAeI

**Tabel 3.1:** Adizes lederroller

Anders Olesen er blevet tildelt et stort bogstav i producentrollen, fordi han forventer, at alle er arbejdsomme og positive - især over for kunder og ikke er bange for at fyre en medarbejder, hvis vedkommende ikke opfylder disse krav. Der gives et stort bogstav i entreprenør-rollen fordi han inkluderer sine medledere i beslutningsprocessen. Derudover tiltaler det nye tiltag til salget af pejse og brændeovne også for entreprenørrollen. Desuden får han et stort bogstav i integratørrollen, fordi han motiverer folk i hans arbejdsområde og giver dem stor frihed til selv at udvikle sig.

Thomas Olesen får et stort bogstav i integratørrollen, fordi han gør meget for det sociale i virksomheden.

Casper Olesen får et stort bogstav i administratorrollen, fordi han er mere styrende end Thomas. Derudover tildeles Casper også et stort bogstav i integratørrollen, da han gerne vil hjælpe sine medarbejdere til at løse opgaverne så godt som muligt.

Ud fra disse vurderinger kan det konkluderes, at virksomheden har udfyldt alle fire lederroller. Dette er en fordel for virksomheden da der så vil blive fokuseret på alle fire aspekter af Adizes lederroller.

### 3.1.4 Problemer, hypoteser og løsningsmuligheder

I dette afsnit vil der blive identificeret problemer, hypoteser og løsningsmuligheder, som vises i tabel 3.2.

Problemer	Hypoteser	Løsningsmuligheder
Kunde køber varer på udsalg og/eller med mængderabat, så den samlede rabat bliver for stor	Medarbejderne har ikke styr på, hvor meget der gives i rabat til den enkelte	IT til rabatstyring
Stort kundekartotek	Det er svært at holde styr på, hvilke kundegrupper de forskellige kunder tilhører	IT til kundekartotek
En vare er udsolgt	Der er dårlig oversigt over antal af en given vare	IT til varehåndtering
Medarbejder kan ikke finde vare på lager	Oversigt over vareplacering er uoverskuelig	IT til lagerstyring
Udbringning af varer er uorganiseret	Leveringen af varer til kunde er ikke organiseret	Optimering af ruteplanlægning
Der er ikke styr på, om de ansatte sælger for mere eller mindre forventet	Der bliver på nuværende tidspunkt ikke ført salgsstatistikker på den enkelte medarbejder	IT til registrering af salg
Virksomheden har ikke 100 % styr på udlejningen	Med det manuelle system er den enkelte medarbejder nødt til at tjekke om noget udlejningsmateriel er afleveret eller lejet ud	IT til udlejningsstyring
En solgt vare på nettet bliver ikke registreret	Det nuværende system registrerer ikke noget ift. lager og bogholderi, når en kunde køber en vare på nettet	Implementering af XL-Bygs hjemmeside

**Tabel 3.2:** Tabel over identificerede problemer, hypoteser og løsningsmuligheder

### 3.1.5 SWOT-analyse

I dette afsnit anvendes analysemetoden SWOT fra afsnit 2.2 til at få et overblik over, hvilke problemstillinger Vestbjerg Byggecenter A/S står over for.

Interne forhold	
Stærke sider (Strengths)	Svage sider (Weaknesses)
<ul style="list-style-type: none"> <li>• Solid egenkapital</li> <li>• 23 års erfaring</li> <li>• Mange kundegrupper</li> <li>• Ansvaret er spredt ud på flere medarbejdere</li> <li>• Flere forskellige afdelinger</li> <li>• Tilfredse medarbejdere</li> <li>• To ledere har gennemført lederuddannelse</li> <li>• Udlejning</li> <li>• Stordriftsfordele gennem XL-Byg</li> <li>• Bredt sortiment</li> </ul>	<ul style="list-style-type: none"> <li>• Udbringning er uorganiseret <sup>1</sup></li> <li>• Mange kundegrupper (Uorganiseret kundearchiv)</li> <li>• Har ikke overblik over lageret</li> <li>• Gammelt IT-system</li> <li>• Uorganiseret rabatsystem</li> <li>• Virksomheden er fordelt på to matrikler</li> <li>• Har ikke styr og debitorer og tilgodehavender</li> </ul>
Eksterne forhold	
Muligheder (Opportunities)	Trusler (Threats)
<ul style="list-style-type: none"> <li>• Nære kunderelationer</li> <li>• Aktiemarkedet</li> <li>• Nye produkter</li> <li>• Nye og flere behov hos kunder ift. udlejningsmateriale</li> </ul>	<ul style="list-style-type: none"> <li>• Konkurrenter</li> <li>• Stigende brændstofpriser</li> <li>• Aktiemarkedet</li> </ul>

Tabel 3.3: SWOT-analyse for Vestbjerg Byggecenter A/S.

### 3.1.6 Stærke sider

Hos Vestbjerg Byggecenter A/S fremstår det, at deres egenkapital og deres store kundekartotek holder firmaet i velstand. Deres kundegrupper, som består af 1000 faste kunder, har bl.a. givet Anders rig mulighed for at lægge noget af ansvaret over på sine sønner og sine medarbejdere. Humøret holdes højt blandt de ansatte, og dette styrkes bl.a. ved diverse arrangementer, som firmaet holder.

<sup>1</sup>Information fra første interview med kunden

### 3.1.7 Svage sider

Trods de positive ting fremstår der mange tekniske udfordringer. Dette ses bl.a. ved deres rabatsystem, som ikke tager højde for kundens tilknytning til en kunde-gruppe, og kan derfor overskride det maksimale 20 % i rabat. Det fremgår også, at overblikket over diverse varer er uorganiseret. Dette skyldes, at Vestbjerg Byggecenter A/S' varer er fordelt ud over to matrikler, og håndteringen af disse varer ikke er systematiseret. Ydermere fremgår det af Vestbjerg Byggecenter A/S' resultatopgørelse, at en betragtelig del af aktiverne er bundet i debitorer, tilmed er tab på tilgodehavender også substantielle.

### Delkonklusion

Vestbjerg Byggecenter A/S har et godt samarbejds miljø, og deres store kundekartotek samt det store varesortiment er tegn på et velfungerende firma. Der forekommer dog problemer hos Vestbjerg Byggecenter A/S med hensyn til organisering af kundegrupper, lageret mm. Deres nuværende håndtering af varer og faste kunder forvolder problemer og dermed spildt arbejdstid. Dette tilkendes gives ved, at hverken salgs- eller lagermedarbejderne præcist ved, hvor én bestemt vare befinder sig. Der er ydermere observeret problemer ved rabatsystemet, hvor der ikke tages højde for en øvre grænse. Der er derfor plads til væsentlige forbedringer hos Vestbjerg Byggecenter A/S.

## 3.2 Strategianalyse

For Vestbjerg Byggecenter A/S gælder det om at have en strategi, der kan anvendes langt ud i fremtiden og sikre virksomhedens overlevelsesmuligheder på længere sigt. Derfor vil dette kapitel undersøge virksomhedens strategi for at se, hvor virksomhedens fokus ligger.

### 3.2.1 Valg af strategi

Ud fra SWOT analysen, som er beskrevet i afsnit 3.1.5, anbefales det at arbejde ud fra en mini-maxi strategi, hvor der fokuseres på at minimere svaghederne og maksimere mulighederne. Dette tilrådes, da svaghederne står i vejen for virksomhedens fremtidsplaner om at udnytte mulighederne for fortsat udvidelse af varesortiment og kunder. Tilmed foreligger der en oplagt og bred løsning til at forbedre mange af virksomhedens svagheder i form af implementering af et nyt IT-system.

### 3.2.2 Ansoffs vækstmatrice

For at kunne analysere Vestbjerg Byggecenter A/S' vækststrategi benyttes Ansoffs vækstmatrix, som er beskrevet i afsnit 2.4. Metoden giver et indblik i, hvor Vestbjerg Byggecenter A/S kan gøre en ekstra indsats for at forbedre sin vækst, og hvorvidt de nuværende strategier er nok til at sikre en fremtidig vækst.

		Produkter	
		Nuværende	Nye
Markeder	Nuværende	Markedspenetrering - Rabatter - Ekspansion mht. kunder og varer	Produktudvikling - Fleksibel udlejning
	Nye	Markedsudvikling - IT skal sikre konkurrenceevne og rentabilitet - Nethandel	Diversifikation - Pejse og brændeovne

**Figur 3.2:** Ansoffs vækstmatrice

Vestbjerg Byggecenter A/S' markedspenetreringsstrategi forekommer ved de mange faste rabatter, som kundegrupperne får glæde af. Dette vurderes til være en medvirkende årsag til, at Vestbjerg Byggecenter A/S har cirka 1.000 faste kunder.



Udvidelse i form af nethandel ville kunne udvide deres kundegruppe, da nogle kunder foretrækker at handle via internettet, samt at Vestbjerg Byggecenter A/S ville kunne nå ud til flere nye kunder som ikke er lokale.

### **3.2.3 Idé og mission**

Ideen bag Vestbjerg Byggecenter A/S var, at den almindelige ufaglærte håndværker kunne købe professionelle materialer og værktøj til fornuftige priser. Dette har videreudviklet sig til, at de professionelle håndværkere også har benyttet muligheden for at handle i Vestbjerg Byggecenter A/S, da de professionelle øjnede muligheden for at købe billige materialer med god og kompetent vejledning.

### **3.2.4 Vision og målsætning**

Vestbjerg Byggecenter A/S' vision er at ekspandere sin markedsandel ved at udvide sin kundebase og varesortiment. Vestbjerg Byggecenter A/S vil udvide sin rækkevidde ved at satse på at digitalisere virksomheden, så den forbliver konkurrencedygtig. For at blive klar til disse mål skal virksomheden tilegne sig et fremtidssikret IT-system til at procesoptimere de daglige arbejdsopgaver.

### **3.2.5 Tilpasning af organisationen**

Umiddelbart vurderes det, at organisationen ikke skal ændres i høj grad grundet det nye system. Dog skal medarbejderne lære, hvordan systemet skal bruges samt huske at opdatere informationer i systemet, hvis f.eks. en vare bliver flyttet til en ny plads. Dette betyder for Vestbjerg Byggecenter A/S' medarbejdere, at de skal huske at bruge og opdatere oplysninger, som ligger i det nye system.

## **3.3 IT-strategi**

I dette afsnit vil der på baggrund af Vestbjerg Byggecenter A/S' forretningsstrategi og -vision udlægges en hensigtsmæssig IT-strategi. Først vil virksomhedens nuværende IT-system kort beskrives. Dernæst drøftes hvilke muligheder et nyt IT-system har for at understøtte Vestbjerg Byggecenter A/S i at udfylde sin vision og strategi samt specificere, hvilke applikationer sådan et system måtte indeholde.

### 3.3.1 Nuværende IT-system

Vestbjerg Byggecenter A/S' UNIX-system er af ældre dato og opfylder ikke virksomhedens behov. IT-systemet bliver hovedsageligt brugt til registrering af salg og bogholderi, hvilket sker uafhængigt af hinanden. Til at understøtte disse funktioner bruges en simpel løsning i form af simpel tekstbehandling og regneark. Desuden er lagersystemet, kundehåndtering, ordrehåndtering og udlejning ikke IT-understøttet. Derudover er XL-Bygs nethandel ikke integreret ordentligt i Vestbjerg Byggecenter A/S' nuværende system.

### 3.3.2 Fastlæggelse af IT-muligheder

Vestbjerg Byggecenter A/S' strategi, som står beskrevet i afsnit 3.2.1, beskriver, hvorledes virksomheden ønsker at fokusere på at udbedre sine svagheder og muligheder. For at beskue virksomhedens svagheder mere detaljeret er det nødvendigt at se på de problemer og hypoteser der er opstillet i tabel 3.2, som findes i sektion 3.1.4.

Et af problemerne er at kunder kan få for store rabatter, i forhold til hvad Vestbjerg Byggecenter A/S egentlig ønsker. Idet rabatter er salgsrelateret, vurderes det nødvendigt, at der bliver udviklet et *salgssystem*. Her foreligger der tilmed en oplagt mulighed for procesoptimering både i form af mindre ekspederingstid og lagring af salgsdata.

Derudover er der problemer i forhold til styring af, hvilke varer der er på lager, og hvilke der ikke er. Det kan også være svært at finde forskellige varer, da virksomheden er på to matrikler. Disse problematikker står i vejen for virksomhedens vision om udvidelse af varesortiment. Dette ville kunne løses ved hjælp af et *lagerstyringssystem*.

Ydermere har Vestbjerg Byggecenter A/S ikke ordentligt styr på deres udlejning, som bliver håndteret manuelt. Dette er også et punkt, som kan forbedres vha. IT, hvilket giver anledning til et *udlejningssystem*.

For at understøtte Vestbjerg Byggecenter A/S' vision om at udnytte mulighederne for at udvide kundebasen, skal virksomheden nemt og overskueligt kunne håndtere forskellige kunder og kundegrupper. Af denne årsag vil en ny IT-løsning skulle indeholde et velstruktureret *kundesystem*.

Når en vare sælges på XL-Bygs hjemmeside bliver der ikke registreret noget i Vestbjerg Byggecenter A/S' system. Derfor er der også identificeret et behov for inte-

gration i et nyt system af det allerede eksisterende *nethandelssystem*.

Da de fleste af medarbejderne har computere hjemme og dermed grundlæggende kendskab til Windows, men formentlig ikke meget mere erfaring med computere, skal systemet være brugervenligt. Det er desuden værd at bemærke, at løsningen bør være et samlet system, og ikke fem forskellige delsystemer.

### 3.3.3 Nøgleapplikationer

Ud fra de IT-muligheder, som er beskrevet i afsnit 3.3.2, kan der udledes følgende nøgleapplikationer:

- Lagerstyring
- Hurtig og nem håndtering af ordrer og salg
- Håndtering af kunder
- Håndtering af udlejning
- Nethandel

### 3.3.4 Implementering

Der er flere muligheder for implementering. Det er vigtigt at lægge fokus på, at overgangen skal være så glidende som muligt. Overordnet kan det siges, at implementering skal tage hensyn til virksomhedens struktur, teknologi, opgaver og personer.

Med struktur menes der, at implementering af det nye system ikke må komme i konflikt med virksomhedens arbejdsdeling, kommunikation og beslutningsstruktur. Det nye system må derfor ikke give systemrettigheder til at ændre i rabatordninger for en salgsmedarbejder.

Den nye teknologi må ikke komme i konflikt med den eksisterende teknologi. Med dette menes der redskaber, der bruges til at løse arbejdsopgaver som f.eks. systemer, maskiner og arbejdsprocesser.

Arbejdsopgaver kan beskrives som udvikling i virksomheden, det er den eksisterende produktionslinje og tjenesteydelser. Det vil sige, at der ved implementering ikke må opstå situationer, hvor en levering bliver slettet fra et eksisterende system.

Under implementering skal medarbejdere fortsat have mulighed for at benytte det eksisterende system. At der opgraderes, opsættes og implementeres må ikke komme i konflikt med den almindelige arbejdsgang hos personalet.

### Implementerings muligheder

Implementering kan være en *total udskiftning* over en weekend eller helligdag, hvor butikken er lukket, da det giver ro til opstilling af back-up systemer, computere der skal anvendes af medarbejdere og ledere samt installation af soft- og hardware.

Alternativt kan der implementeres *sideløbende*, hvor det nuværende system stadig er i brug samtidig med, at det nye bliver installeret. Dette betyder, at implementeringen også kan foregå i virksomhedens almindelige åbningstid, og at der ikke skal kaldes en medarbejder ind uden for den almindelige åbningstid.

## 3.4 Applikationer og information

I dette afsnit beskrives, hvad der skal bruges i forhold til materiale, samt hvordan systemet skal bruges på kort sigt og langt sigt.

### 3.4.1 På kort sigt

I første omgang er det nødvendigt for Vestbjerg Byggecenter A/S at få det tiltænkte system op at køre. Dette gøres ved på kort sigt at indkøbe hardware.

Det betyder, at virksomheden skal investere i f.eks. en lille server til lagring af information. Altså er det på kort sigt ikke en fornuftig beslutning at indkøbe plads til 10.000 kunder, når det er konstateret, at Vestbjerg Byggecenter A/S har 1000. Ydermere er det ikke en nødvendighed for systemet at have plads til samtlige varenumre der findes i byggebranchen. Da Vestbjerg Byggecenter A/S har et katalog på 28.000 kan der indkøbes serverplads til 30.000.

For at sikre data er det for virksomheden nødvendigt at have en back-up. På kort sigt kan det overvejes at have en såkaldt "on site" back up.

### 3.4.2 På længere sigt

Om fem år, når Anders Olesen ønsker virksomheden totalt digitaliseret kan det betale sig for virksomheden at opgradere deres hardware. For Vestbjerg Byggecenter A/S betyder dette større server til opbevaring af lokale og internetbaserede kunder.

For den lokale virksomhed er det nødvendigt at overveje, hvor mange varenumre der skal lagerføres, men i princippet kan der indkøbes hardware til de 100.000 der findes i branchen.

For at sikre data er det nødvendigt at gøre dette på en anden måde. Der skal ikke længere være en "on site" back-up, men i stedet to til flere "remote site" lokationer, der kan opbevare og sikre data mod lokale katastrofer.

### 3.5 Organisering af IT-funktionerne

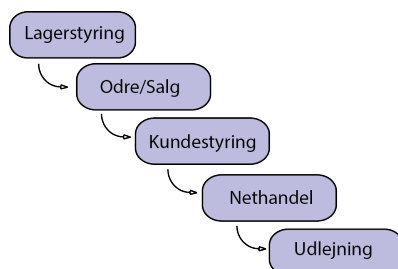
Ansvar for at vedligeholde og opdatere systemerne kan placeres ved samme firma som udviklede systemet. Dette er en fordel, da systemets opbygning så allerede er kendt. Dermed vil eventuelle ønsker om opgradering af systemet ikke skabe store udfordringer, da det ikke er nødvendigt for nye udviklere at sætte sig ind i systemet.

En anden mulighed er at ansætte et andet firma til denne opgave. Dette kan medføre, at yderligere arbejde med systemet bliver billigere, da der så vil være større konkurrence om prisen på opgaven.

### 3.6 IT handlingsplan

I dette afsnit vil der fastsættes en rækkefølge for implementering af de enkelte applikationer, som står beskrevet i afsnit 3.3.3. Dernæst laves en cost benefit analyse, samt en Parker Benson analyse af de enkelte applikationer for at fastlægge en prioritering ud fra hvilken der giver størst forretningsværdi.

#### 3.6.1 Rækkefølge af applikationer



Figur 3.3: Rækkefølge af IT-projekter

Nøglepunkterne, som beskrevet i afsnit 3.3.3, for Vestbjerg Byggecenter A/S er bl.a. lagerstyring og salgsstatistikker, og der skal lægges et væsentligt fokus på disse to emner for at imødekomme virksomhedens ønsker.

Da det, at vide hvilke varer der findes på lageret, er en essentiel del af at administrere et byggemarked, betragtes dette delsystem som en grundlæggende forudsætning for resten af systemets udvikling. Af denne årsag vil lagerstyring blive prioriteret først og vægtet som kritisk.

Det kan samtidig ikke vides, hvad der findes på lageret uden et system, der kan assistere med salg. Når der sælges en vare, er det nødvendigt, at denne vare trækkes fra beholdningen. Ydermere er en vigtig del af at drive forretning at kunne se, hvad der sælges og i hvilke mængder. Dette problem kan afhjælpes med et salgssystem, der automatisk danner statistikker og rapporter til medarbejderne og ledelsen. Af denne årsag vægtes et salgssystem som yderst vigtigt og vurderes til at komme umiddelbart næst i rækkefølgen, hvis ikke sideløbende med lagerstyring.

Et kundekartotek er vigtig men ikke lige så essentielt, som lager- og salgs-delsystemerne, for virksomhedens konkurrencedygtighed. Af denne årsag vægtes dette system som mindre vigtigt.

Når disse basale delsystemer er på plads er det derefter vigtigt at tænke på integration af nethandel for at kunne imødekomme virksomhedens vision.

Udlejning er en service der tilbydes, fordi konkurrenterne tilbyder den. Af denne årsag vægtes det som det mindst vigtige system.

### 3.6.2 Cost-benefit

I dette afsnit foretages en cost benefit analyse af det tiltænkte system. En cost-benefit analyse bruges til at beslutte, hvorvidt et projekt skal gennemføres. Måden analysen laves, er ved at sammenholde den forventede indtjening med de forventede udgifter. Af denne årsag er cost-benefit forholdet mellem omkostninger og udbytte [6].

De forventede udgifter kan ses i tabel 3.4.

Direkte omkostninger	Omkostninger (kroner)	
Datamatiker løn	192.000	engangsydelse
Server opstart	5.000	engangsydelse
Implementeringsomkostninger	5.000	engangsydelse
Driftsomkostninger		
Server leje	1.500	pr. mdr.
Netværk	300	pr. mdr.
Direkte indtægter/besparelser	Besparelse	
Sparet løn lager (ca. 15 min pr.gang)	56.250	pr. mdr.
Sparet løn kontor	10.500	pr. mdr.
Sparet udlejning	840	pr. mdr.

Tabel 3.4: ROI regnskab.

ROI beregnes ud fra direkte målelige- såvel som indirekte målelige omkostninger og besparelser. Ved det målelige forstås det mere konkrete, som kan regnes i kroner og øre. Her kan man på baggrund af den allerede tilegnede viden draget af foregående analyser og interviews gøres sig et løst estimat af potentielle omkostninger og besparelser, som vist på tabel 3.4. Igen skal det understreges, at dette bygges på vurderinger og estimerer som findes i bilag [7]. Omkostninger kan inddeles i engangsydelser (direkte omkostninger) og driftsomkostninger. Engangsydelsen "Datamatiker løn" beregnes ud fra den antagelse, at projektet vil tage ca. en måned at implementere. Regnestykket ser derfor således ud: 6 datamatikere gange 32.000 (gennemsnitlig månedsløn for en nyuddannet datamatiker [8]) . Ydermere vurderes implementerings- og teknologi-omkostninger ikke til have nogen nævneværdig værdi, jf. afsnit 3.3.4 og 3.5. De kombinerede daglige driftsbesparelser "sparet løn lager", "sparet løn kontor" og "udlejning" må derimod siges at være signifikante. De estimeres ud fra en effektiviseret arbejdsproces, som systemet vil medføre. Bl.a. vil et lagerstyringssystem mindske tiden det tager for en medarbejder at finde en vare på lageret.

Ud fra disse estimerer vil de samlede månedlige driftsbesparelser blive  $((56.250 + 10.500 + 840) - (1500 + 300)) = 65.790$  kr. Dette betyder, at de samlede engangsydelser vil blive tjent hjem i løbet af cirka fem måneder  $(202.000 / 65.790 \approx 3)$ .

Ved de indirekte målelige værdier vurderes der at forekomme et positivt afkast i form af øget arbejdsglæde, da den enkelte medarbejder, grundet en effektiviseret

salg- og ordreproces samt mindre spildtid på lageret, vil der opleves færre frustrationer i den daglige arbejdsgang. Derudover vil det nye system understøtte statistikker af salg og kunder, hvilket vil hjælpe ledelse med planlægge nye, samt optimere nuværende, marketingsstrategier.

### 3.7 Cost-benefit analyse - Parker Benson

I dette afsnit opstilles en Parker Benson cost-benefit analyse. Metoden til dette er beskrevet i afsnit 2.5.

Return of Investment (ROI)	Strategic Match (SM)
Competitive Advantage (CA)	Management Information (MI)
Competitive Response (CR)	Organisational Risk (OR)
System Architecture (SA)	Definition Uncertainty (DU)
Technological Uncertainty (TU)	Infrastructure Risk (IR)

	Forretningsdomæne						Teknologidomæne				Total
	ROI	SM	CA	MI	CR	OR	SA	DU	TU	IR	
Vægt	5	3	4	4	1	-1	2	-1	-2	-1	
Lagerstyringsdelsystem	25	15	12	16	4	-3	8	-2.5	-7	-3	64.5
Salgsdelsystem	20	15	8	20	5	-2	8	-2	-7	-3	62
Udlejningsdelsystem	10	3	12	8	1	-2	4	-1	-7	-1	27
Kundedelsystem	20	15	8	16	4	-2	8	-2	-7	-3	57
Nethandelsintegration	20	12	12	16	4	-2	8	-5	-8.5	-3	53.5

Figur 3.4: Parker Benson cost benefit analyse organisation

#### Resultat af Parker Benson analysen

I figur 3.4 fremgår det, at virksomheden har vægtet forretningsdomænet højest, og at teknologidomænet er blevet vægtet lavest. I forretningsdomænet er delkategorierne ROI, CA og MI også vægtet højt, mens CR og OR tildeles lavere vægtning. Dette indikerer, at virksomheden prioriterer afkast af målelig og svært-målelige værdier, samt det at kunne forblive konkurrencedygtige. Den lave vægtning af CR indikerer, at ledelsen ikke vurderer noget umiddelbart pres for hurtigt at reagere på det konkurrerende marked. Ydermere mener ledelsen tilsyneladende at have en velformuleret plan for implementering af systemet, og risikoen vurderes dermed lav jvf. den lave OR vægtning.

De to højeste scorende applikationer er lagerstyrings- og salgs-delsystemet, som klarer sig godt i det højt-vægtede forretningsdomæne. Udlejning er blevet tildelt



en lav score i forretningsdomænet, dog er risici ikke vurderet store, men da virksomheden ikke har vægtet teknologidomænet højt, har det ikke en stor betydning.

Det modsatte kan siges om nethandelsintegration. Denne har fået en høj score i forretningsdomænet men er derimod vurderet til at have høj risici, men igen fordi virksomheden ikke vægter teknologidomænet højt, har den fået tildelt en høj score.

Ud fra tabellen kan det konkluderes at lagerstyrings- og salgs-delsystemerne har fået de højeste scorere. Det kan derfor udledes, at disse to applikationer bedst kan betale sig, og det er derfor disse som har størst forretningsværdi.

### **3.8 Brugerdeltagelse**

Der bliver i forbindelse med udviklingen af dette system brugt ekspertstrategi fra figur 2.7 (ses i metodekapitel 2.6). Virksomhedsejeren Anders vil være den ekspert, der er involveret i systemudviklingen. Dette gøres på trods af, at der kan mistes vigtige informationer fra de ansatte, i forhold til det gamle system

Det er muligt, at eksperten på et senere tidspunkt i forløbet inddrager viden fra andre brugere.

### **3.9 Konklusion på forundersøgelsen**

Baseret på de opdagelser, der er gjort i IT-forundersøgelsen, kan det konkluderes, at Vestbjerg Byggecenter A/S kunne have gavn af at udskifte deres IT-system. De nøgleapplikationer, der er nævnt i afsnit 3.3.3, er alle vurderet nødvendige for et nyt system. Dog konkluderes det, at der hovedsageligt skal være fokus på lager og salgsstyring ved hjælp af Parker Benson analysen beskrevet i afsnit 3.7.



## 4 | Problemdefinition

I dette kapitel vil det analyserede problemområde blive afgrænset. Resultatet heraf er en problemformulering samt nogle succeskriterier som kan bruges til at vurdere, hvorvidt systemet lever op til forudbestemte krav.

### 4.1 Problemafgrænsning

I kapitel 3 blev der identificeret eksisterende udfordringer ved Vestbjerg Byggecenter A/S' eksisterende system. Det er eksempelvis ikke muligt at finde en vares lokation gennem IT-systemet. Derudover er det ikke muligt for ledelsen at generere detaljerede salgsstatistikker for afdelinger, sælgere og produkter, hvilket gør det besværligt for virksomheden at danne overblik over den nuværende situation.

Det vurderes derfor ved hjælp af afsnit 3.7, at det for Vestbjerg Byggecenter A/S vil være økonomisk fordelagtigt at gennemføre en udskiftning og opgradering af deres IT-system. Denne beslutning vil også bringe virksomheden tættere på deres vision om at blive mere digitaliseret. Som nævnt i konklusionen for IT-forundersøgelse, afsnit IT-forundersøgelse, afsnit 3.9, er de mest relevante problemområder salg og lager.

### 4.2 Problemformulering

Denne afgrænsning resulterer i en problemformulering som lyder:

*Hvordan kan der udvikles en digital løsning, som opfylder Vestbjerg Byggecenter A/S' krav til et nyt system, hvor det skal være muligt bl.a. at håndtere salg og lager samt oprettelse af nye kunder, varer og leverandørdata, således at dette hjælper både ledelsen og de ansatte i virksomheden?*

### 4.3 Succeskriterier

For at sikre at problemformuleringen løses på en tilfredsstillende måde, vil der i denne sektion opstilles en række succeskriterier, som kan bruges til at vurdere, hvorvidt systemet vil være brugbart for Vestbjerg Byggecenter A/S:

1. Systemet er stabilt og indeholder ingen kritiske fejl samt pludselige nedbrud.
2. Systemet hjælper de ansatte med salg, lagerstyring mv.
3. Systemet hjælper lederne med at fastsætte priser og rabatter.
4. Systemet gør det nemmere at registrere varer og færdiggøre f.eks. tilbud.

## 5 | Inception

I dette kapitel startes selve analysen af, hvordan systemet skal fungere. Indledningsvist beskrives der, hvad der er nødvendigt for systemet. Derefter designes hvordan systemet skal fungere ved hjælp af forskellige systemudviklingsværktøjer.

### 5.1 Interview

De problemstillinger, som Vestbjerg Byggecenter A/S dagligt oplever, er blevet identificeret og beskrevet i kapitel 3. For at opsummere, hvilke problemstillinger der er relevante, vil Anders Olesen gerne kunne føre statistik over salg samt have en bedre styring af sit lager.

Før det er muligt at opfylde Anders Olesens ønske om at vise statistik, er det nødvendigt først at have et system til at registrere salg. Derefter skal der implementeres et lagersystem, der kan vise, hvad der findes på lageret. Uden lagerstyring og automatisk genbestilling er der risiko for at sælge varer, der ikke findes i butikken. Dette vil give et falsk salg og kan derfor give butikken en omsætning der reelt ikke findes og derved give et falsk billede af indtjeningen.

### 5.2 Mock-ups

I dette afsnit vises og beskrives nogle af de første udkast af mock-ups. Formålet med mock-ups er at give kunden en fornemmelse af, hvordan systemet vil se ud og fungere. For at sikre at systemet er brugbart for kunden, vil de udarbejdede mock-ups testes på kunden.



Figur 5.1: Mock-up af hovedmenu

På figur 5.1 vises en mock-up for startskærmen til systemet. Da Anders Olesen skulle teste systemet stod det hurtigt klart, at det eneste, der for ham er relevant, er statistik-delsystemet. Da selve testen egentlig handlede om at oprette en ordre var det derfor ikke optimalt at bruge Olesen som testperson. Af denne årsag var de udarbejdede mock-ups ikke relevante for den videre udvikling af systemet.

### 5.3 Medarbejder - opgave - mål tabel

Dette afsnit er skrevet med udgangspunkt i metoden beskrevet i afsnit 2.7. Den tabel, som er blevet udarbejdet i forbindelse med systemet til Vestbjerg Byggecenter, A/S er vist i tabel 5.1

Tabel 5.1: Medarbejder - Opgave - Mål tabel

Medarbejder	Opgave	Mål	Step i opgave
Salgs- medarbejder	Oprette ny or- dre	Opret ordre	Find modul til salg Find undermodul 'Opret ordre' Find eller tilføj kunde Tilføj de ønskede varer Tilføj eventuelle rabatter Vælg levering eller afhentning
	Oprette nye tilbud	Opret tilbud	Find modul til salg Vælg undermodul 'Opret til- bud' Find eller opret kunde Tilføj de ønskede varer til til- buddet Tilføj eventuelle rabatter
	Annuller eksis- terende ordre	Annuller or- dre	Find modul til salg Vælg undermodul 'Annuller or- dre' Find kunde Find ordre Slet ordre
	Lav tilbud om til ordre	Skift tilbuds- type	Find modul til salg Find undermodul 'Eksisterende tilbud' Find kunde Vælg det aktuelle tilbud Vælg om tilbud skal godkendes eller annulleres
	Lav kassesalg	Sælg vare(r)	Find modul til salg Vælg undermodul 'Kassesalg' Tilføj de ønskede varer til salget Tilføj eventuelle rabatter
	Modtag penge for ordrer	Betal ordre	Find modul til kunde Vælg undermodul 'Udestående' Find kunde Afkryds ordrer til betaling Modtag penge

Fortsættes på næste side...

Tabel 5.1 – Fortsat

Medarbejder	Opgave	Mål	Step i opgave
	Lav faktura til kunde	Udskriv faktura	Find modul til kunde Vælg undermodul 'Fakturaer' Find kunde Find ordre Udskriv faktura
	Opret nye kunder	Opret kunde	Find modul til kunder Vælg undermodul 'Opret kunde' Indtast oplysninger om kunde Tildel kundegruppe
	Find eksisterende kunde	Find kunde	Find modul til kunder Vælg undermodul 'Find kunde' Søg på kunde
	Opdater kundeoplysninger	Opdater kunde	Find modul til kunder Vælg undermodul 'Ændr kunde' Find kunde Indtast nye oplysninger om kunde
	Find information om vare	Find vare	Find modul til lager Vælg undermodul 'Find vare' Søg på vare
	Print prisskilt til hyldeforkant	Udskriv hyldeforkant	Find modul til lager Vælg undermodul 'Skilte' Find vare Vælg størrelse Udskriv hyldeforkant
	Opret ny udlejningsmateriel i systemet	Opret udlejningsmateriel	Vælg modul til udlejning Vælg undermodul 'Opret udlejningsmateriel' Indtast oplysninger om udlejningsmateriel

Fortsættes på næste side...



Tabel 5.1 – Fortsat

Medarbejder	Opgave	Mål	Step i opgave
	Registrer udlejning af udlejningsmateriel	Opret udlejning	Find modul til udlejning Vælg undermodul 'Opret udlejning' Find eller tilføj kunde Find og vælg udlejningsmateriel
	Hjemkald udlejet udlejningsmateriel	Hjemkald udlejningsmateriel	Find modul til udlejning Vælg undermodul 'Hjemkald udlejning' Find den specifikke udlejning Send hjemkaldelse
Lagermedarbejder	Registrer oplysninger om ny leverandør	Opret leverandør	Find modul til lager Vælg undermodul 'Leverandører' Vælg undermodul 'Opret leverandør' Indtast oplysninger om leverandør
	Opdater oplysninger om eksisterende leverandør	Opdater leverandør	Find modul til lager Vælg undermodul 'Leverandører' Vælg undermodul 'Opdater leverandør' Indtast nye oplysninger om leverandør
	Opret ny vare	Opret vare	Find modul til lager Vælg undermodul 'Opret vare' Vælg leverandør Indtast oplysninger om vare
	Ændr oplysninger på vare	Opdater vare	Find modul til lager Vælg undermodul 'Opdater vare' Find vare der skal opdateres Indtast nye oplysninger om vare

Fortsættes på næste side...

Tabel 5.1 – Fortsat

Medarbejder	Opgave	Mål	Step i opgave
	Find information om vare	Find vare	Find modul til lager Vælg undermodul 'Find vare' Søg på vare
	Modtag vare	Registrer modtagelse	Find modul til lager Vælg undermodul 'Bestilte varer' Find den specifikke ordre Meld ordre ankommet
	Bestil varer hjem til lager	Genbestil varer	Find modul til lager Vælg undermodul 'Opret bestilling' Godkend aktuelle bestillinger systemet automatisk har oprettet Send bestilling videre til leverandør
Chauffør	Find aktuelle leverancer	Find leverancer	Find modul til lager Vælg undermodul 'Leveringer' Find aktuelle leverancer i liste Udskriv følgesedler
	Find varer tilknyttet et bestemt køkken	Find køkken-dele	Find modul til lager Vælg undermodul 'Find vare' Find og vælg aktuelle køkken
	Registrer varer leveret	Opdater vareantal	Find modul til lager Vælg undermodul 'Leveringer' Find den specifikke leverance Meld varer leveret
Leder	Fastsæt pris på vare	Administrer pris	Find modul til lager Vælg undermodul 'Opdater vare' Find den ønskede vare Opdater pris på vare

Fortsættes på næste side...

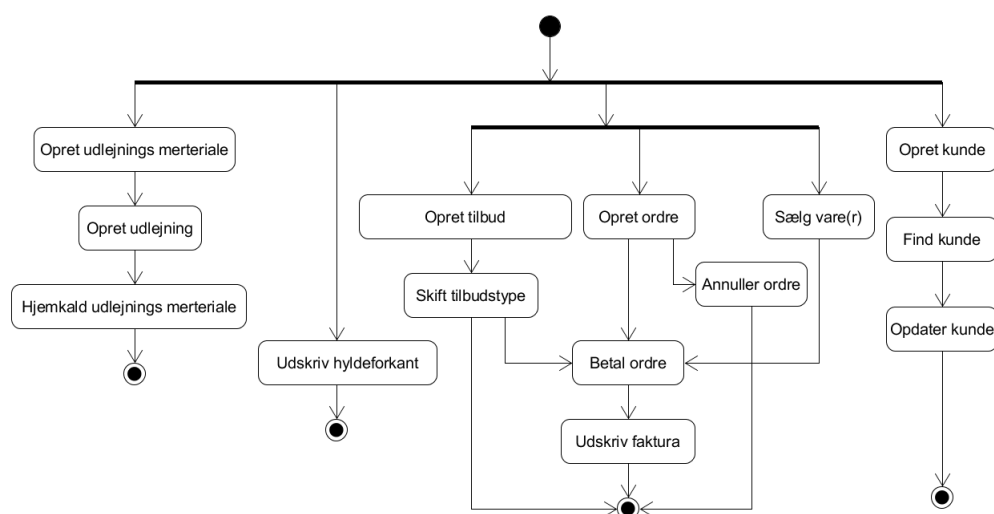
Tabel 5.1 – Fortsat

Medarbejder	Opgave	Mål	Step i opgave
	Aflæs statistik for ønskede område	Aflæs statistik	Find modul til statistik Vælg ønskede statistik- undermodul Vælg periode

Målene fra medarbejder - opgave - mål tabellen bruges videre i rapporten til at udarbejde use case navne og formål.

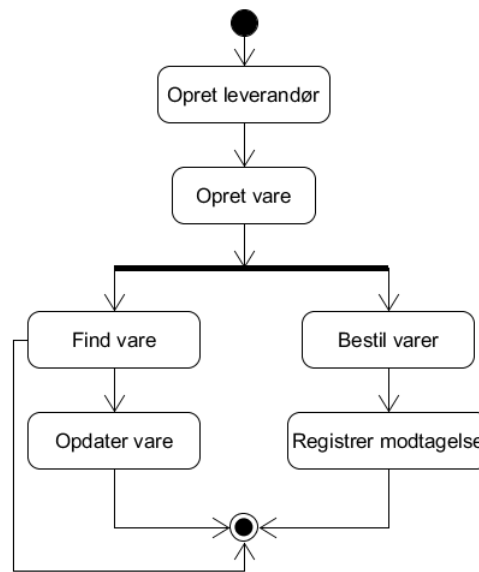
## 5.4 Workflow

I dette afsnit vil der blive taget udgangspunkt i medarbejder - opgave - mål tabellen fra afsnit 5.3, tabel 5.1, for at opstille workflows for de aktuelle aktører. Metoden, der bruges til dette, er beskrevet i afsnit 2.8.



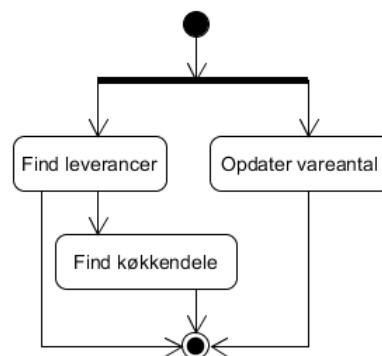
Figur 5.2: Workflow for Salgsmedarbejder

Figur 5.2 viser målene som salgsmedarbejderne står overfor i sin hverdag. Disse mål har en association afhængig af hvilke mål, salgsmedarbejderen skal løse. Denne aktør er den som, står over for de fleste opgaver i løbet af en arbejdsdag, de fleste af disse opgavers mål har en associering til en anden opgaves mål.



**Figur 5.3:** Workflow for Lagermedarbejder

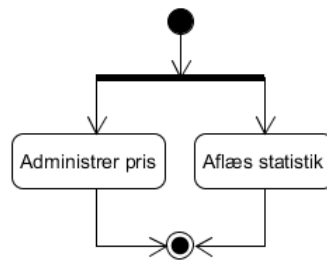
Figur 5.3 viser målene for de forskellige opgaver lagermedarbejderen står overfor i sin hverdag. Disse mål har en association afhængig af hvilke mål, som lagermedarbejderen skal løse.



**Figur 5.4:** Workflow for chauffør

I figur 5.4 ses de mål chaufføren oplever i sin arbejdsdag, som er stillet op i et

workflow diagram.



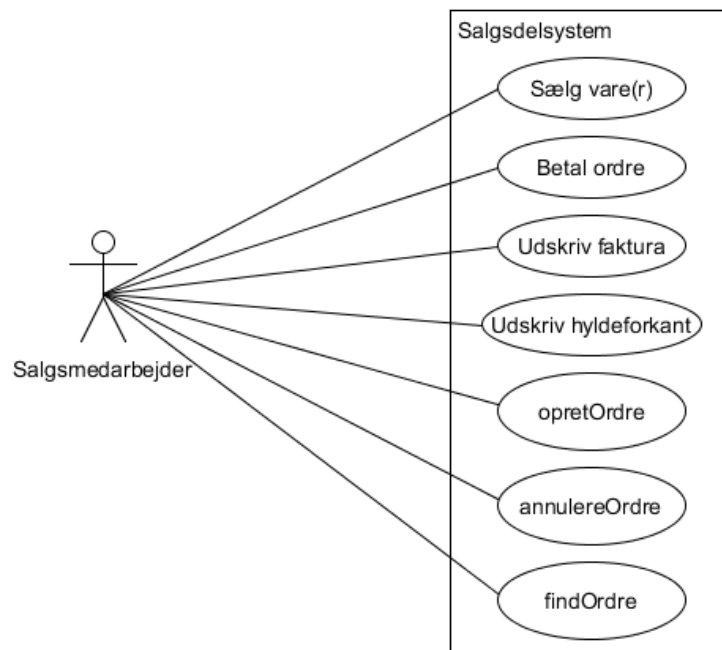
Figur 5.5: Workflow for Leder

Figur 5.5 illustrerer målene i et workflow for lederne.

## 5.5 Use case diagrammer

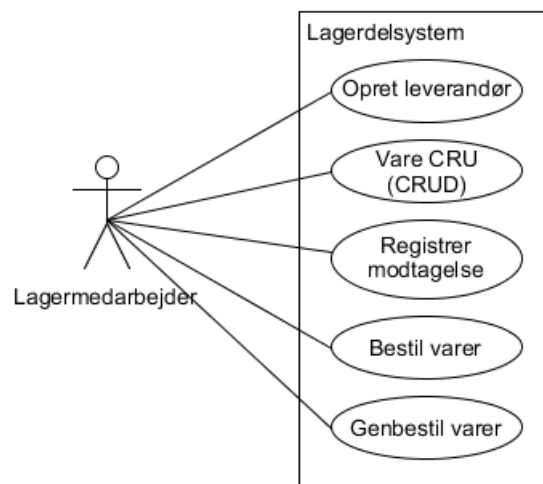
I dette afsnit opstilles use case diagrammer. Metoden til at udarbejde et use case diagram er beskrevet i afsnit 2.9.2.

På figur 5.6 vises use case diagrammet for salgsmedarbejdere. Det ses, at det er muligt for en salgsmedarbejder at Create (oprette), Read (finde) og Delete (annullere) ordrer. Det er derfor ikke muligt at opdatere en ordre. Hvis en kunde ønsker at tilføje noget til en eksisterende ordre, oprettes der en ny ordre. I forhold til "Kunde CRU" er 'D' fjernet, da eksisterende kunder ikke skal fjernes fra systemet. Angående 'Tilbud CD' er 'R' og 'U' fjernet, fordi et tilbud kun kan oprettes, og slettes når det bliver lavet om til en ordre eller annulleres.



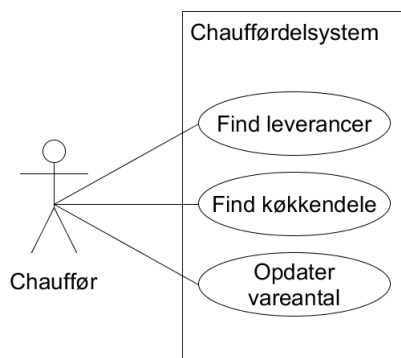
Figur 5.6: Use case diagram for salgsmedarbejdere.

Figur 5.7 illustrerer use case diagrammet for lagermedarbejdere. Årsagen til at 'D' er fjernet fra "Vare CRU" er, at når først en vare er i systemet, skal den ikke kunne fjernes igen. For at undgå at systemet hele tiden giver genbestillings-advarsler, kan lagerets minimum og maksimum beholdning ændres for den specifikke vare.



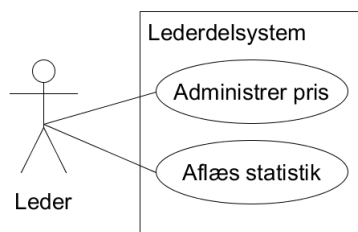
Figur 5.7: Use case diagram for lagermedarbejdere.

Figur 5.8 viser use case diagrammet for chauffører i Vestbjerg Byggecenter A/S.



Figur 5.8: Use case diagram for chauffører.

Det sidste use case diagram er figur 5.9, som viser de ting en leder kan gøre i systemet. Årsagen til, at der ikke kan gøres mere i systemet, er, at Anders Olesen har udtrykt et ønske om kun at have mulighed for at administrere priser og aflæse diverse statistikker.



Figur 5.9: Use case diagram for ledere.

## 5.6 Use case beskrivelser

I dette afsnit bliver nogle af use cases, som er blevet identificeret i afsnit 5.3, beskrevet på formen *brief*. Denne metode er blevet beskrevet i afsnit 2.9.1. Dette gøres for at vise, hvordan en aktør bruger systemet til at opnå et bestemt mål.

### 5.6.1 Salgsmedarbejder

Use case: Opret ordre

En kunde henvender sig til virksomheden for at afgive en ordre. Salgsmedarbejderen benytter IT-systemet til at danne en ordre til kunden, tilføjer kundeinforma-

tioner og varer. Salgsmedarbejderen gemmer ordren, udskriver bekræftelse, følgeseddel og faktureringsoplysninger til kunden, lager og bogholderi. Salgsmedarbejderen kan i alle trin af bestillingen se hvilke varer, pris på disse og total for hele ordren.

Use case: Betal ordre

En kunde har fået eller købt nogle varer, og skal betale for dem. Salgsmedarbejderen bruger systemet til at finde ordren og registrere, at den er blevet betalt.

Use case: Udskriv faktura

En kunde har betalt sine varer, og de er enten afsendt eller afhentet. Salgsmedarbejderen bruger derfor systemet til at udskrive en faktura på ordren, så kunden har et købsbevis.

Use case: Udskriv hyldeforkant

Der er kommet nye varer eller flyttet rundt på nogle, som allerede findes i Vestbjerg Byggecenter A/S. Salgsmedarbejderen bruger systemet til at finde varen der skal have en hyldeforkant samt at printe den.

### 5.6.2 Lagermedarbejder

Use case: Bestil varer

Lagerbeholdningen for en vare er kommet under minimumsgrænsen, og derfor er det tid til at bestille nye varer. Lagermedarbejderen bruger systemet til at godkende automatisk genererede ordrer og sende disse bestillinger til de respektive leverandører.

Use case: Registrer modtagelse

Nogle bestilte varer er ankommet til lageret. Lagermedarbejderen bruger systemet til at opdatere lagerbeholdningen, sådan de ankomne varer bliver tilføjet til lageret.

### 5.6.3 Leder

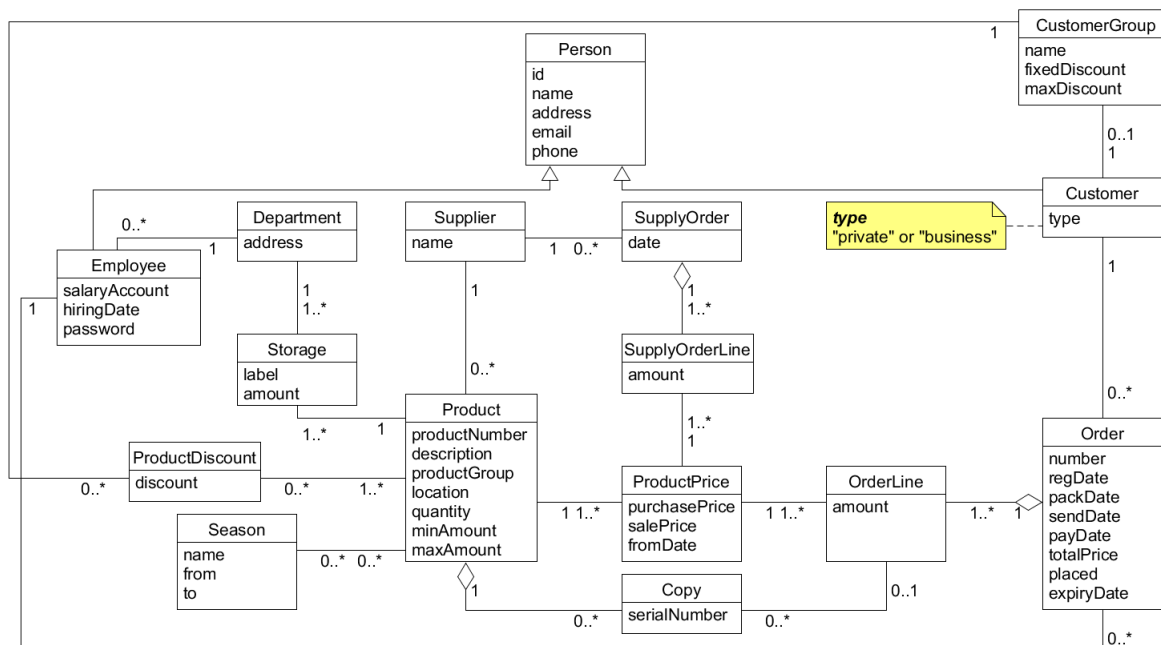
Use case: Aflæs statistik

En leder ønsker at se, hvordan det går den enkelte medarbejder. Lederen bruger systemet til at vise statistikker fordelt på den enkelte medarbejder over en valgt periode.



## 5.7 Domænemodel

I dette afsnit bliver domænemodellen for systemet til Vestbjerg Byggecenter A/S beskrevet. Domænemodellen er blevet udarbejdet ved hjælp af den metode, som er beskrevet i afsnit 2.10



Figur 5.10: Domænemodel for systemet til Vestbjerg Byggecenter A/S.

Figur 5.10 viser domænemodellen for systemet til Vestbjerg Byggecenter A/S. Her ses det, at klasserne Employee og Customer begge er specialiseringer af superklassen Person. Dette skyldes, at de har flere fælles attributter. Begge klasser har desuden en såkaldt 'nul-til-mange' relation til Order. Dette skyldes, at man som kunde kan lave nul til mange ordrer. Dog er det også nødvendigt med en ansat til at registrere ordren i systemet. Derudover er Customer også associeret til CustomerGroup, da nogle kunder får nogle faste rabatter af varierende størrelse. Mellem CustomerGroup og ProductDiscount er der også en associering. Dette skyldes, at eksempelvis tømrermester Michael kan have en aftale om, at han får ekstra rabat på en bestemt Product. Netop Customer-klassen har forbindelse til mange andre klasser, da denne er en meget central klasse i systemet. For at holde styr på gamle varepriser er Product associeret med ProductPrice. Denne vare pris er associeret til OrderLine, da der på den måde er mulighed for at holde styr på tidligere priser. Derudover er OrderLine aggregeret til Order, da en bestilling består

af flere bestillingslinjer. Da en *Order* altid gerne skulle have en *Supplier* er disse to klasser associeret. Desmere er *Product* også aggregeret til *Storage* og *Season*. Årsagen til at *vare* er associeret til *Storage* er for at kunne holde styr på, hvor en specifik vare befinder sig i en specifik *Department*. *Season* og *Product* er associeret med hinanden, da nogle varer kun bør lagerføres i de aktuelle sæsoner. Den sidste klasse, *Product* er direkte forbundet med, er *Copy*. Dette skyldes, at nogle varer har serienumre, og det er der også nødt til at være styr på. De tidligere nævnte klasser *ProductPrice* og *Copy* har begge associeringer til klassen *OrderLine*, da en *Order* kan bestå af flere forskellige varer. *Copy*-klassen er dog kun relevant, hvis der er tale om en vare, som har et serienummer.

## 5.8 Use-case prioritering

Use cases	Kompleksitet	Dækningsgrad	Forretningsværdi
Lav kassesalg	Lav	7	Høj
Betal ordre	Lav	2	Høj
Udskriv faktura	Lav	6	Høj
Udskriv hyldeforkant	Lav	4	Lav
Opret Ordre	Høj	9	Høj
Kunde CRU (CRUD)	Lav	2	Høj
Tilbud CD (CRUD)	Lav	2	Høj
Opret leverandør	Lav	1	Høj
Vare CRU (CRUD)	Høj	7	Høj
Modtag vare	Lav	4	Høj
Genbestil vare	Høj	8	Høj

Figur 5.11: Use-case prioritering

Ud fra domæneklassediagrammet og Parker Benson analysen er der udarbejdet et diagram over de forskellige use-cases kompleksitet og dækningsgrad. De to use cases, *Opret ordre* og *Genbestil vare*, er begge valgt, fordi deres dækningsgrad er de højeste, og derfor er deres kompleksitet også vurderet højt.

I forbindelse med *Genbestil vare* vil der også arbejdes med use casen *Modtag vare*, da det vurderes at disse to use cases har en tæt samhørighed og en logisk sammenhæng.

## 5.9 Use case - Fully dressed

I dette afsnit vil de mest komplekse use cases beskrevet i afsnit 5.6 blive gennemgået fully dressed. Metoden, til hvordan dette gøres, er beskrevet i afsnit 2.9.1.

Tabel 5.2 viser use casen *Opret ordre* beskrevet fully dressed.

**Tabel 5.2:** Fully dressed beskrivelse for *Opret ordre*.

Use case navn	<b>Opret ordre</b>	
Aktører	Salgsmedarbejder	
Præbetingelse	Varen findes i systemet.	
Postbetingelse	En ny ordre er gemt. Til ordren er tilføjet vare- og kundeoplysninger. Ordreseddel og faktura er udskrevet.	
Frekvens	20-100 gange dagligt fordelt på alle salgsmedarbejdere	
Flow of events	<b>Aktør</b>	<b>System</b>
	1. En kunde henvender sig for at købe varer	
	2. Ekspedienten påbegynder en ny ordre	3. Systemet opretter en ny ordre
	4. Ekspedienten indtaster varenummer på den vare der skal tilknyttes ordren.	5. Systemet viser varebeskrivelse, lagerbeholdning, pris og antal.
	6. Salgsmedarbejderen tilføjer varen til ordren	7. System associerer varen til ordren
	8. Trin 4-7 gentages indtil alle varer er tilføjet.	
	9 Salgsmedarbejderen indtaster kundenummer	10. Systemet finder kunden ud fra kundenummer
	11. Salgsmedarbejderen tilføjer kunden til ordren	12. System associerer kunden til ordren
		13. Systemet tilføjer ordrenummer, registreringsdato og total beløb til ordren.
	14. Salgsmedarbejderen afslutter ordren	15. Systemet gemmer ordren og udskriver følgeseddel og faktura

Fortsættes på næste side...

Tabel 5.2 – Forstsat

	<b>Aktør</b>	<b>System</b>
Alternative flow		5a. En vare kan ikke findes ud fra det indtastede varenummer. Systemet viser en fejlmeddelelse.
Flow of events	11a. Kunden findes ikke i systemet.	11b. Systemet prompter for at kunden skal oprettes.
	14a. Salgsmedarbejderen annullerer ordren	14b. Systemet sletter ordren.

På tabel 5.3 vises use casen *Genbestil vare* fully dressed. Ud fra nogle præ- og postbetingelser bliver der opstillet nogle rammer for, hvor funktionaliteten i selve systemet skal ligge. Dette giver mulighed for at kigge på, hvilke funktioner aktøren skal instantiere, og hvordan systemet derefter skal agere. I denne specifikke use case er det mest systemet, der er aktør, da de fleste funktioner bliver automatisk udført af systemet. Det alternative flow i denne analyse, er ikke mulig at udrede på nuværende tidspunkt, da det er en use case for sig.

Tabel 5.3: Fully dressed beskrivelse for *Genbestil vare*.

Use case navn	<b>Genbestil vare</b>	
Aktører	Lagermedarbejder	
Præbetingelse	Varen er oprettet i systemet, leverandøren er kendt, varen er under minimum antal og indenfor sæson	
Postbetingelse	Varen er tilføjet til bestillingslisten	
Frekvens	Når lagerbeholdningen for en vare rammer minimum antal.	
Flow of events	<b>Aktør</b>	<b>System</b>
	1. Use casen starter med at en lagermedarbejder ønsker at få bestillingslister	
		2. Systemet finder varer under minAntal
		3. Systemet finder leverandør ud fra varen
		4. Systemet opretter bestillingslister til de pågældende leverandør.

Fortsættes på næste side...

Tabel 5.3 – Fortsat

	Aktør	System
Flow of events		5. Systemet tilføjer varen til bestillingslisten
		6. Systemet sender bestillingslisten til godkendelse
	7. Medarbejderen godkender bestillingslisten	8. Systemet sender bestillingslisten til leverandøren
Alternative flow	7a. Medarbejderen har en ændring til bestillingslisten	

Tabel 5.4 viser fully dressed for use casen *Modtag vare*. Denne use case er blevet udviklet i forbindelse med *Genbestil vare*, da disse følges ad, når handlinger i forhold til behandling af vare som skal bestilles og modtages. Diagrammet viser hvad der skal være opfyldt, før denne use case kan køre igennem. Dette vises i præ-betingelser. Post betingelserne viser hvad use casen ender med, når den kører igennem som den skal.

Use case navn	Modtag vare	
Aktører	Lagermedarbejder	
Præbetingelse	Varen er tidligere bestilt, og varen er nu ankommet på lager. Det samlede antal bestilte varer bliver leveret	
Postbetingelse	Varen er registreret som modtaget og lagerbeholdningen af den givne vare opdateres	
Frekvens	1-10 gange dagligt	
Flow of events	Lagermedarbejder	System
	1. En tidligere bestilt vare bliver leveret til lageret	
	2. Lagermedarbejderen bruger systemet til at finde bestilte varer, som endnu ikke er modtaget	3. Systemet returnerer bestilte varer, som endnu ikke er modtaget
	4. Lagermedarbejderen udvælger fra listen den vare som er blevet leveret	5. Systemet registrerer den valgte vare som modtaget og opdaterer lagerbeholdningen
	Trin 4-5 gentages afhængig af antal varer modtaget	

Tabel 5.4: Fully dressed beskrivelse for *Modtag vare*.

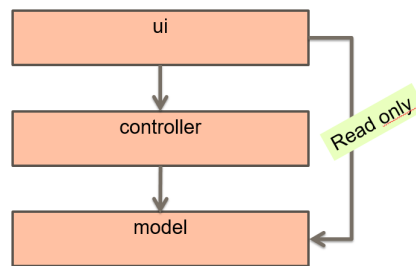


## 6 | Elaboration

I dette kapitel vil der, som viderebygning på inception-fasen, blive udarbejdet system-sekvensdiagrammer, operationskontrakter, interaktionsdiagrammer og designklassediagrammer. Ydermere vil der være kode eksempler på, hvordan programmet er blevet implementeret.

### 6.1 Arkitektur

Et systems arkitektur er en konceptuel model, som definerer struktur og opførsel for et system. For at designe en forståelig og fleksibel arkitektur bruges packages. Desuden er der også fokus på low coupling og high cohesion (lav kobling og høj samhørighed). Cohesion refererer til graden af samhørighed. Coupling siger noget om, hvor meget de forskellige komponenter afhænger af hinanden. Fordelen ved dette fokus er, at det er nemt at udskifte dele i systemet. En normal og logisk arkitektur inddeler systemet i følgende tre lag: grænseflade (ui), controller og model. Der findes to former for tre-lags arkitektur – åben og lukket. Den åbne arkitektur tillader at ui-laget også kan læse fra model-laget. Et eksempel på en sådan arkitektur kan ses på figur 6.1. Den lukkede tillader kun lagene at kigge i det nærmest underliggende lag. Det vil sige at ui kun har adgang til controlleren, som har adgang til model-laget [9].

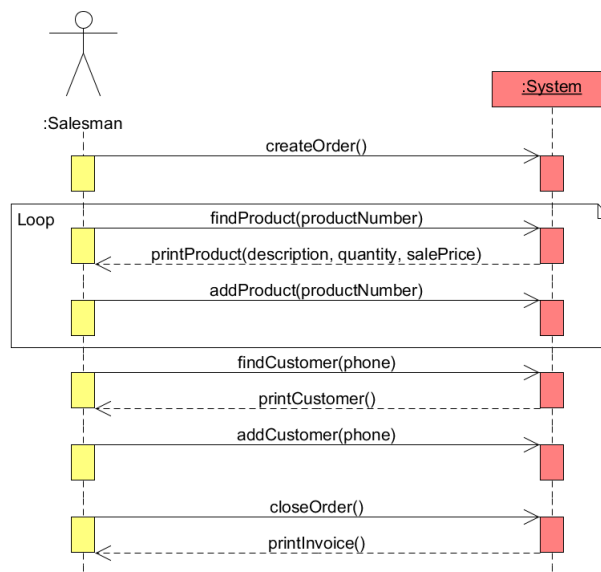


Figur 6.1: Eksempel på tre-lags åben arkitektur [9].

## 6.2 System-sekvensdiagrammer

I dette afsnit fremvises de udarbejdede system-sekvensdiagrammer (SSD) for use cases *Opret kunde*, *Genbestil vare* og *Modtag vare*. Dette gøres for at vise et sæt af handlinger der er i forbindelse med den pågældende use case og for at vise, hvordan aktøren og systemet interagerer med hinanden [5, s.118].

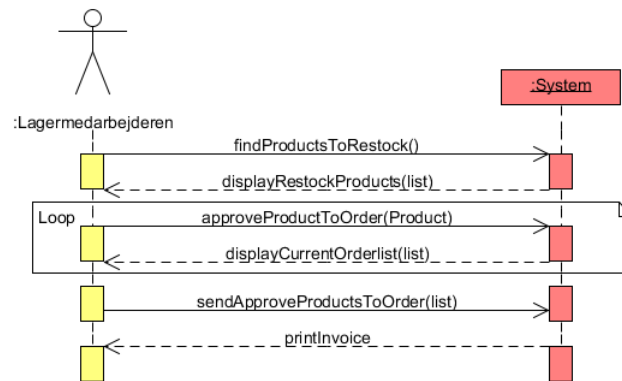
I figur 6.2 ses SSDet for use casen *Opret ordre*.



Figur 6.2: SSD'et for use casen *Opret ordre*.

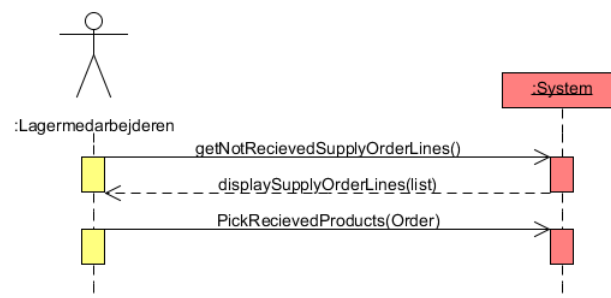
I figur 6.3 ses SSDet for use casen *Genbestil vare*.





**Figur 6.3:** SSD'et for use casen *Genbestil vare*.

På figur 6.3 ses SSDet for use casen *Genbestil vare*. Lagermedarbejderen starter med at bruge systemet til at få vist en liste over de varer, hvis beholdning er nået under den angivne minimumsgrænse. Dernæst godkendes de varer fra listen, som ønskes genbestilt enkeltvis, hvor systemet løbende viser en foreløbig ordreliste med de godkendte vare. Sluttelig afsendes ordrerne på ønskede vare og systemet udskriver en ordrebekræftelse.



**Figur 6.4:** SSD'et for use casen *Modtag vare*.

På figur 6.4 ses SSDet for use casen *Modtag vare*. Når en bestilt vare ankommer til lageret, bruger lagermedarbejderen systemet til at finde den modtagne vare på en liste over bestillinger, hvorefter den registreres som modtaget.

### 6.3 Operationskontrakter

I dette afsnit vil der blive udarbejdet operationskontrakter. Dette gøres med henblik på at beskrive et detaljeret stadie af objekterne tilhørende use casen efter operationerne er blevet gennemført [5, s.177].

Operation: createOrder
Use case: Opret ordre
Præbetingelse:
- En kunde og vare findes i systemet
Postbetingelse:
- En instans o af Ordre blev oprettet
- o blev tilskrevet aktuelle værdier
- En instans k af Kunde er blevet associeret til o

**Tabel 6.1:** Operationskontrakt for Opret ordre.

På tabel 6.1 vises operationskontrakten for *Opret ordre*. Det ses, at ordre skal have associeret en kunde og have tilskrevet aktuelle værdier.

Operation: sendApprovedProductsToOrder(approvedProducts)
Use case: Genbestil vare
Præbetingelse:
- Listen approvedProducts af objekt typen 'Product' er kendt.
- Objektet soCon af typen SupplyOrderContainer er kendt.
Postbetingelse:
- Objektet sol af typen SupplyOrderLine er blevet oprettet.
- sol.product og sol.amount er blevet givet værdier.
- Objektet so af typen SupplyOrder er blevet lavet, og so.regDate, so.supplier er blevet givet værdier.
- sol er blevet associeret med so.
- so er blevet associeret med soCon.

**Tabel 6.2:** Operationskontrakt for Genbestil vare.

I operationskontrakten på tabel 6.2 er amount det antal, som skal genbestilles. Hvilket udregnes ud fra varens attributter, maxAmount minus quantity. maxAmount er det maksimum antal af varen som må være på lageret. quantity er det aktuelle antal på lageret.

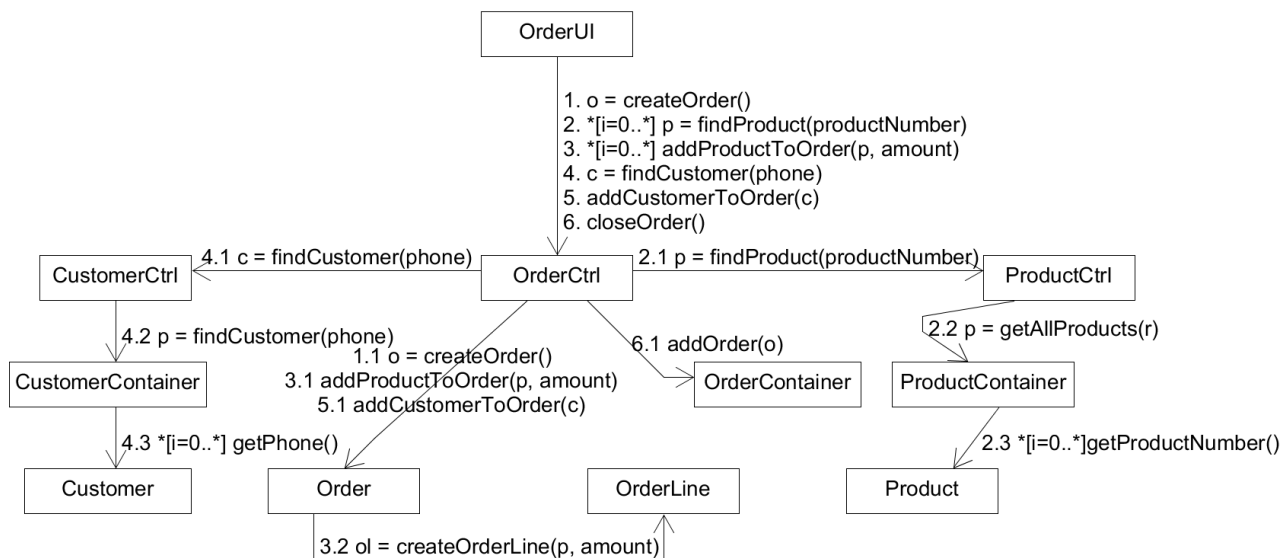
Operation: setSupplyOrderListToRecieved()
Use case: Modtag vare.
Præbetingelse:
- Objektet so af typen SupplyOrderLine er kendt.
- Objektet p af typen Product eksistere og er associeret til so.
Postbetingelse:
- so.isRecieved er blevet sat til true værdi.
- p.quantity er sat til værdien = p.quantity + so.amount.

Tabel 6.3: Operationskontrakt for Modtag vare.

I tabel 6.3 ses operationskontrakten for *Modtag vare*. Denne viser, at en SupplyOrderLine so skal være kendt, og denne skal have et associeret Product tilknyttet for at blive executed. Der ses desuden, at når use casen er løbet, som den skal, er so.isRecieved sat til true for at vise, at varen er modtaget, og at p.quantity er blevet opdateret, så varemængden på lager vises korrekt.

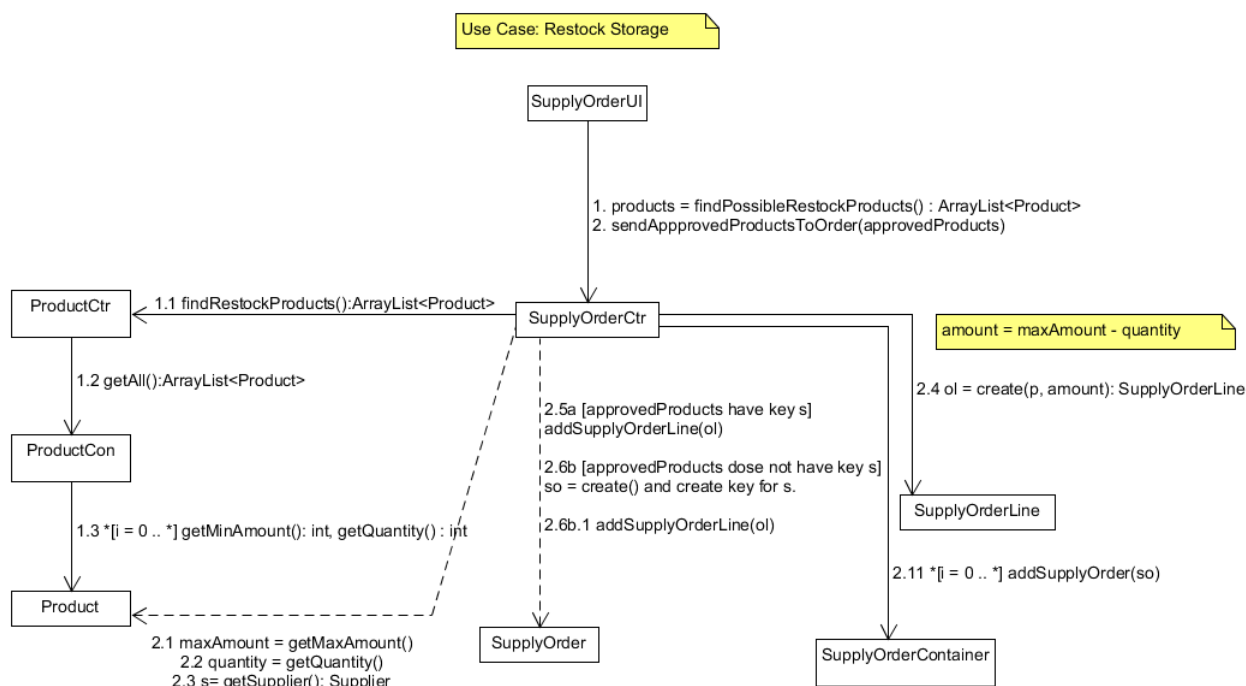
## 6.4 Kommunikationsdiagrammer

I dette afsnit opstilles interaktionsdiagrammer. Dette gøres med henblik på, at illustrere hvordan objekterne interagerer med hinanden ved hjælp af metoder [5, s. 197].



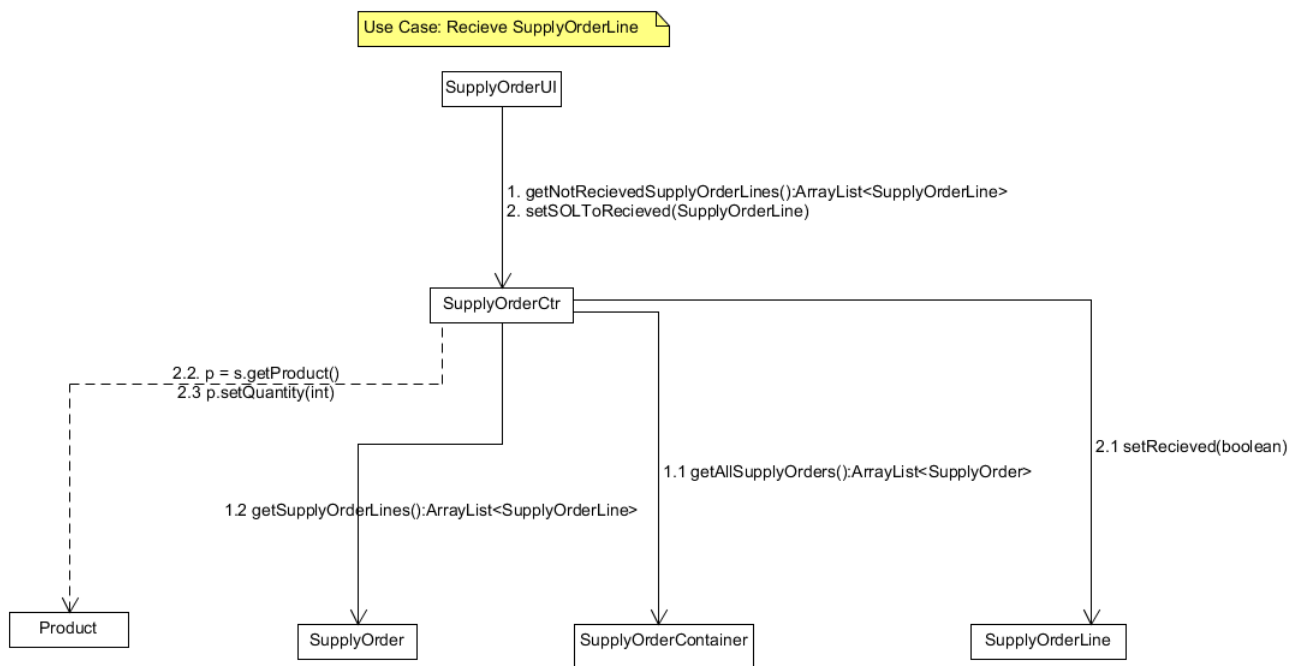
Figur 6.5: Kommunikationsdiagram for Opret ordre.

Figur 6.5 viser det udarbejdede kommunikationsdiagram for *Opret ordre*. Denne del af programmet er designet ud fra, at en kunde henvender sig, for at bestille en ordre. Der kan tilføjes nul til mange produkter på en *OrderLine* og nul til mange *OrderLines* på en *Order*. Til en *Order* bliver der tilføjet en *Customer*.



**Figur 6.6:** Kommunikations for Genbestil Vare.

På figur 6.6 ses kommunikationsdiagrammet for Genbestil Vare. Bemærk de to alternative interaktioner 2.5a og 2.6b. Dette er grundet at metoden `sendApprovedProductsToOrder` sorterer de givne ordre ud fra leverandør vha. et `HashMap`, som indeholder `SupplyOrder` objekter, med `Supplier` som key. Når `sendApprovedProductsToOrder` kaldes vil der instansieres `SupplyOrderLine` objekter ud fra de `Product` objekter medsendt i parameterlisten. Disse `SupplyOrderLine` objekter placeres på det `SupplyOrder` objekt vis key (`Supplier`) i `HashMap`'en matcher `Supplier` på `Product` objektet i `SupplyOrderLine` objektet (2.5a). Hvis der ikke forekommer noget match, betyder det at der endnu ikke er oprettet et `SupplyOrder` objekt ud fra `Product` objektets `Supplier` og der vil instansieres et nyt `SupplyOrder` objekt (2.6b - 2.6b.1).



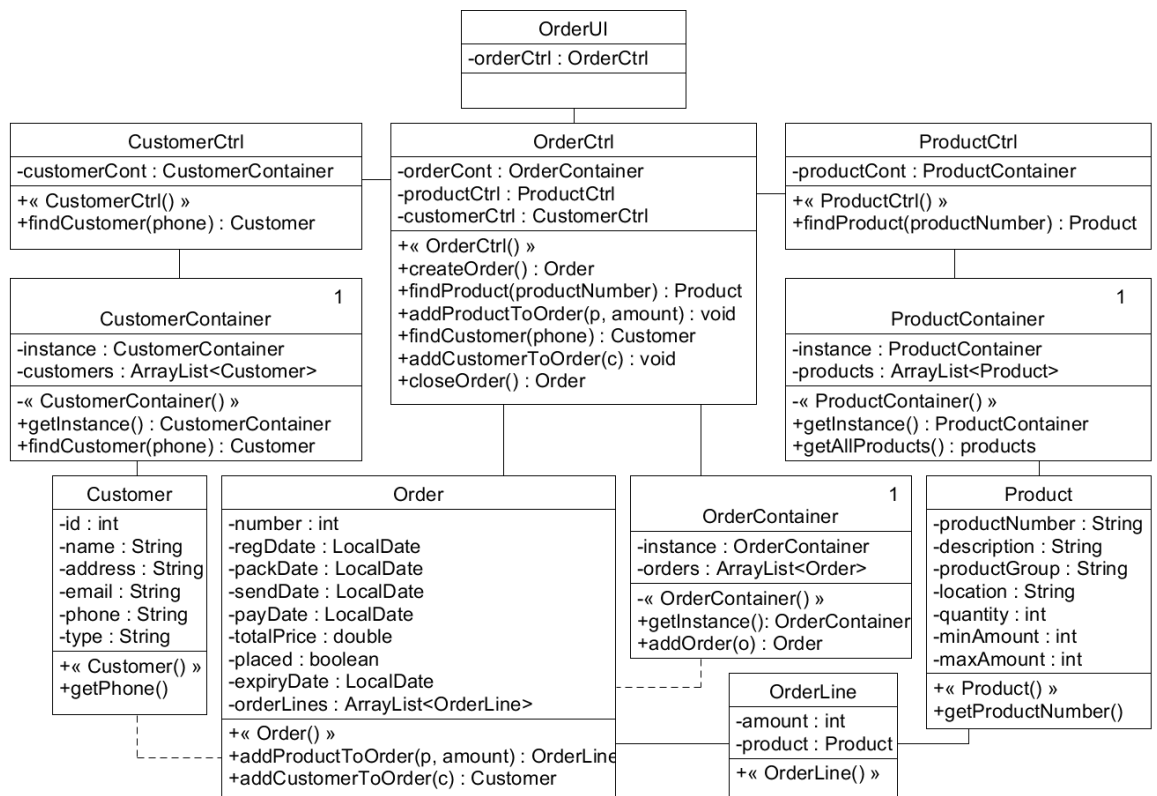
Figur 6.7: Kommunikationsdiagram for *Modtag vare*.

På figur 6.7 ses kommunikationsdiagrammet for *Modtag Vare*. Denne del af system er designet ud fra den antagelse at alle vare i en ordre nødvendigvis ikke leveres på samme tid. Denne løsning baseres på at registrere én vare ad gangen i stedet for hele ordren.

## 6.5 Designklassediagrammer

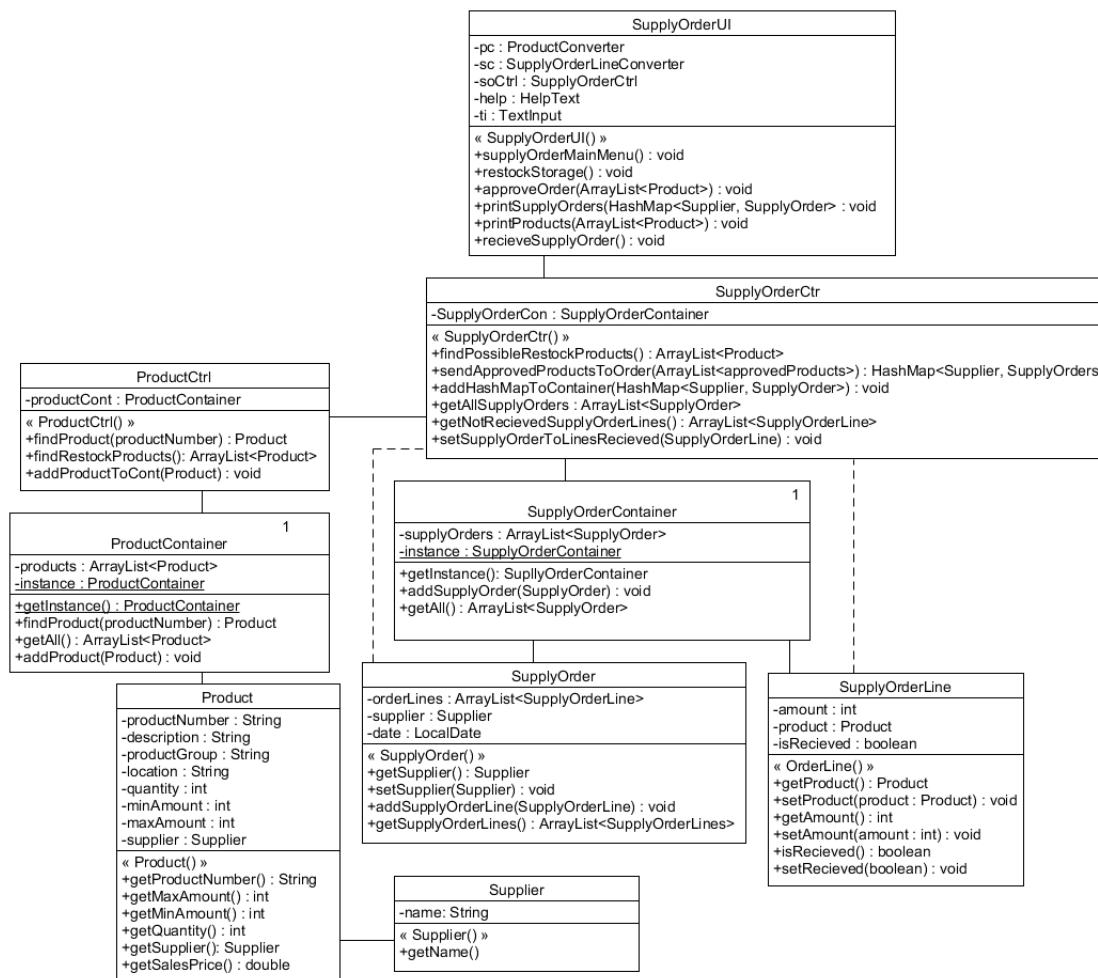
I dette afsnit vil der blive fremvist designklassediagrammer, som er udarbejdet på baggrund af interaktionsdiagrammerne fra afsnit 6.4. Et samlet designklassediagram findes i bilag B. Dette gøres for at give et samlet overblik over klasserne, hvor der illustreres de metoder og attributter, som tilhører de respektive klasser [5, s. 10].

På figur 6.8 vises designklassediagrammet for use casen *Opret ordre*.



**Figur 6.8:** Designklassediagram for *Opret ordre*.

På figur 6.9 vises designklassediagrammet for use casen *Genbestil vare og Modtag vare*.



Figur 6.9: Designklassediagram for Genbestil vare og Modtag vare.

Figur 6.8 og figur 6.9 repræsenterer de færdige design klasse diagrammer for use cases *Opret ordre*, *Genbestil varer* og *Modtag varer*. Diverse container-klasser, er lavet som singleton. Dette repræsenteres med ét 1-tal i øverste højre hjørne. Singleton mønsteret er med til, at sikre der kun instantieres et objekt af klassen. Dette ses ved den statiske metode `getInstance()` i container-klasserne.

## 6.6 Kode

I denne sektion vil der blive fremvist kodeeksempler på, hvordan diverse *use cases* er blevet implementeret i systemet. Dette gøres ved at vise udvalgte metoder og give en kort forklaring til disse metoders funktionalitet.

### 6.6.1 Tilføj produkt til ordre

```
1  public OrderLine addProductToOrder(Product p, int amount){
2      if(orderLines.size() != 0){
3          for(OrderLine orderLine : orderLines){
4              if(orderLine.getProduct().equals(p)){
5                  int temp = orderLine.getAmount() + amount;
6                  orderLine.setAmount(temp);
7                  return orderLine;
8              } else {
9                  return createOrderLine(p, amount);
10             }
11         }
12     }
13     else {
14         return createOrderLine(p, amount);
15     }
16     return null;
17 }
```

**Listing 6.1:** Metoden addProductToOrder .

På listing 6.1. Denne metode tager to parameter, bestående af typerne Product og int. Metoden starter med en if-else kæde, hvor der checkes efter om, hvorvidt ArrayListen har nogle ordrer tilknyttet til sig. Hvis det er tilfældet, benyttes et for-each loop til at se, om det produkt som ønskes tilknytning, allerede findes i ordren. Er dette tilfældet bliver amount lagt til den eksisterende amount. Ellers vil metoden createOrderLine(p, amount) blive kaldt, som står for at oprette en orderLine og derefter tilføjes ordrelinjen til listen.



### 6.6.2 Udregn totalpris for ordren

```
1 private double sumTotalPrice(int i) {  
2     double temp = 0;  
3  
4     if(i >= 0){  
5         OrderLine ol = orderLines.get(i);  
6         Product p = ol.getProduct();  
7         temp = p.getSalePrice() * ol.getAmount();  
8         return temp + sumTotalPrice(i - 1);  
9     }  
10    else {  
11        return temp;  
12    }  
13 }
```

**Listing 6.2:** Metoden sumTotalPrice(int i) .

```
1 public double sumTotalPrice(){  
2     return totalPrice = sumTotalPrice(orderLines.size() -  
3     1);  
}
```

**Listing 6.3:** Metoden sumTotalPrice .

På listing 6.2 og listing 6.3 ses de to metoder som bliver brugt til at udregne den totale pris ud for ordren. Metoden sumTotalPrice() returnerer totalPrice, hvor totalPrice er assigned sumTotalPrice(orderLines.size()). Dette gøres med henblik på at komme hele listen igennem med et rekursivt kald. Den rekursive metode ses i listing 6.2, hvor der tages en intsom parameter. Denne parameter repræsenterer størrelsen af listen med OrderLines.

Inputtet dekrementeres med én hver gang metoden kaldes. Dette sker ind til indeks 0 på ArrayListen er nået. Metoden starter med en if-else sætning, hvor der checkes om starten af listen er nået. Hvis ikke metoden har været hele listen igennem, så laves en midlertidig OrderLine, som assignes den nuværende position i listen, ved hjælp af metodekaldet get(i). Ud fra den fundne OrderLine findes det tilhørende Product og dets salgspris. Salgsprisen ganges derefter med antallet af produkter som så assignes til en ny variabel temp. Efterfølgende returneres temp plus sumTotalPrice(i - 1), dette gøres for at tælle listen ned, samt at den fundne fulde pris, for produktet returneres.

### 6.6.3 Genbestil vare

```
1 public HashMap<Supplier, SupplyOrder>
   sendApprovedProductsToOrder(ArrayList<Product>
   approvedProducts){
2     HashMap<Supplier, SupplyOrder> tempSupplyOrders = new
   HashMap<>();
3     for(Product p : approvedProducts){
4         Supplier sup = p.getSupplier();
5         int amount = amountToReorder(p);
6         SupplyOrderLine sol = new SupplyOrderLine(p, amount);
7         if(tempSupplyOrders.containsKey(sup)){
8             tempSupplyOrders.get(sup).addSupplyOrderLine(sol);
9         }
10        else{
11            SupplyOrder tempNewSupplyOrder = new SupplyOrder(sup);
12            tempNewSupplyOrder.addSupplyOrderLine(sol);
13            tempSupplyOrders.put(sup, tempNewSupplyOrder);
14        }
15    }
16
17    return tempSupplyOrders;
```

Listing 6.4: Metoden sendApprovedProductsToOrder .

På listing 6.4 ses metoden `sendApprovedProductsToOrder` som hører til *use casen Genbestil vare*. Metodens opgave er at tage en liste af varer, som er godkendt til genbestilling og knytte disse til genbestillingsordrer, som er sorteret efter leverandør. Metoden tager en `ArrayList` (`approvedProducts`) med objekter af typen `Product` i parameterlisten og returnerer et `HashMap` af `SupplyOrder` objekter med `Supplier` som key. Først instantieres et lokalt `HashMap`, `tempSupplyOrders`, som skal holde `SupplyOrder` objekterne. For-each loopet på linje tre itererer igennem `ArrayList`en `approvedProducts`, hvor den først finder objektets attributter `p.supplier` og `amount`. `amount` er det antal af den givne vare der skal genbestilles, som udregnes: `p.getMaxAmount() - p.getQuantity()`. `amount` findes via et indre metodekald til metoden `amountToOrder` som returnerer resultatet af førnævnte regnestykke. Dernæst instansieres et objekt `sol` af typen `SupplyOrderLine` ud fra det pågældende `Product` objekt i loopet og dens `amount`. I linje 7 tester if-sætningen om det aktuelle `Product` objekt i `sol` har en `supplier` der matcher en eksisterende key i `HashMap`'et. Er dette tilfældet betyder det, at der allerede er oprettet et `SupplyOrder` objekt ud fra den pågældende leverandør som `sol` så dernæst tilføjes. Findes der ikke et match oprettes der et nyt `SupplyOrder` objekt som så tilknyttes `sol` og lægges i `HashMap`'et `tempSupplyOrders`. En fordel ved denne metode er,

at den samler alle genbestillingsvarer til samme leverandør i én ordre i stedet for mange små.

```
1 public void addHashMapToContainer(HashMap<Supplier ,
2     SupplyOrder> tempSupplyOrders){
3     Collection<SupplyOrder> c = tempSupplyOrders.values();
4     Iterator<SupplyOrder> it = c.iterator();
5     while(it.hasNext()){
6         SupplyOrder temp = it.next();
7         supplyOrderCon.addSupplyOrder(temp);
8     }
9 }
```

**Listing 6.5:** Metoden addHashMapToContainer .

Når samtlige Product objekter er tilknyttet SupplyOrder objekter i tempSupplyOrders, skal brugeren bekræfte afsendelse af genbestillingen. Når bestillingen bekræftes kaldes metoden vist på listing 6.5 som tager et HashMap (i dette tilfælde tempSupplyOrders) og lægger alle dennes values i et Collection objekt c som tager typen SupplyOrders. Derefter itereres c via et Iterator objekt it og hver SupplyOrder objekt tilføjes SupplyOrderContainer objektet supplyOrderCon's ArrayListe.

#### 6.6.4 Modtag vare

```
1
2 public ArrayList<SupplyOrderLine>
3     getNotRecievedSupplyOrderLines(){
4     ArrayList<SupplyOrder> supplyOrders = getAllSupplyOrders();
5     ArrayList<SupplyOrderLine> sol = new ArrayList<>();
6     for (SupplyOrder s : supplyOrders){
7         for(SupplyOrderLine so: s.getSupplyOrderLines()){
8             if (!so.isRecieved()){
9                 sol.add(so);
10            }
11        }
12    }
13    return sol;
14 }
```

**Listing 6.6:** Metoden getNotRecievedSupplyOrderLines

På listing 6.6 ses metoden getNotRecievedSupplyOrderLines fra use casen *Modtag vare*. Metoden itererer igennem alle SupplyOrder objekter hentet fra SupplyOrderContainer

objektet via det indre metodekald `getAllSupplyOrders`. For hver iteration loopes der yderligere via et indre `for-each` loop som itererer igennem alle `SupplyOrderLine` objekter for det pågældende `SupplyOrder` objekt. I linje 7 testes det aktuelle `SupplyOrderLine` objekt for om dens attribut `isRecieved` er `false`. Er dette tilfældet bliver objektet tilføjet `ArrayList`en `sol` som returneres ved metodens afslutning.

```
1
2 public void setSOLToRecieved(SupplyOrderLine
3     supplyOrderLine){
4     ArrayList<SupplyOrderLine> sol =
5         getNotRecievedSupplyOrderLines();
6     for (SupplyOrderLine s: sol){
7         if (s.equals(supplyOrderLine)){
8             s.setRecieved(true);
9             Product p = s.getProduct();
10            p.setQuantity(amountToReorder(p) + p.getQuantity());
11        }
12    }
13 }
```

Listing 6.7: Metoden `setSOLToRecieved`

Metoden på listing 6.7 bruges til at registrere, når en ordrelinje fra en leverandørbestilling modtages, samt opdatere lagerbeholdningen af modtaget vare. Metoden gør brug af metoden fra listing 6.6 for at lave en lokal `ArrayList`, `sol`, af alle objekter af typen `SupplyOrderLine`, hvis `isRecieved` attribut er sat til `false`. Hvert objekt i `ArrayList`en `sol` testes for sammenlignelighed med det medgivne `SupplyOrderLine` fra parameterlisten. Findes et match, sættes det pågældende objekts `isRecieved` attribut til `true` og det tilknyttede `Product` objekt's attribut `quantity` sættes i linje 8 til `amountToReorder() + p.getQuantity`.

### 6.6.5 libTUI

Til at implementere brugergrænsefladen (TUI) er der gjort brug af et importeret library ved navn `libTUI`. I dette bibliotek ligger forskellige klasser som indeholder en masse logik til at implementere basale tui-metoder. Disse klasser er generiske således, at de er bredt anvendelige. Denne genbrug af kode er med til at sikre højere cohesion.

```
1 public T promptChoiceValidOnly(String title, String prompt) {
2     System.out.println(title);
```

```
3  displayOptions();
4  boolean selecting = true;
5  int choice = 0;
6  while (selecting){
7      choice = new TextInput().promptInt(prompt);
8      if (choice <= options.size() && choice > 0 ){
9          return options.get(--choice);
10     }
11
12     else if(choice == 0) {
13         selecting = false;
14     }
15     else {
16         System.out.println("Indtast venligst et gyldigt tal");
17     }
18 }
19
20 return null;
21 }
```

Listing 6.8: Metoden PromptChoiceValidOnly

På listing 6.8 ses metoden `PromptChoiceValidOnly` som findes i klassen `TextChoice` i biblioteket `libTUI`. Biblioteket er blevet opdateret med denne metode, da den giver yderligere anvendelsesmuligheder og er yderst relevant for netop dette system. Denne metode er blevet implementeret til anvendelse, hvis der er brug for at udvælge et objekt fra listen af options. Options er en instansvariabel i klassen og holder en `ArrayList` af objekter af den generiske type `T` som specificeres i parameterlisten ved instantieringen af `\verbTextChoice|` objektet. Brugen af denne metode forudsætter at options først fyldes med de objekter der ønskes at vælge fra. Først opstilles de givne valgmuligheder med tilhørende tal via metodekaldet `displayOptions()`. Brugeren vælger så det objekt som ønskes returneret ved at indtaste dennes sidestillede tal. Indtastes et ugyldigt tal fremvises en fejlmeddelelse (linje 16) og while-loopet (linje 6) starter forfra. Indtastet tallet 0 returnerer metoden `null` og loopet brydes. Ved et gyldigt tal returneres tallets tilhørende objekt.

Der gøres brug af denne metode i metoderne `restockStorage` og `recieveSupplyOrder` i `tui` pakken. Her bruges `null` til at indikere en afslutning af metoden.

Det vurderes, at disse kode eksempler er de mest komplekse, og derfor også dem der er mest interessante at diskutere.

### 6.6.6 Tidskompleksitet

Til udregning af tidskompleksitet benyttes store-O notation. Store-O viser hvordan løbe tiden af en algoritme vokser, baseret på størrelsen af inputtet [10].

Det forekommer i de fleste metoder, at der benyttes  $O(n)$ , da listen bliver gennemgået unsorted og der foretages  $n$  iterationer inden det ønskede element er fundet. Dog er der også en metode som er  $O(n^2)$  denne metode er `getNotRecievedSupplyOrderLines` som ses i listing 6.6. Dette skyldes at der foretages et nested for-loop som gør at vi skal gennemgå to lister for, at finde det ønskede element.

## 7 | Diskussion

I dette kapitel vil der blive diskuteret, hvad der kunne være været gjort anderledes undervejs i projektet. De enkelte diskussioner er delt op i underafsnit for bedre at kunne diskutere individuelle dele.

### 7.1 Use cases

Efter use casen Genbestil vare var blevet implementeret blev modtag varer implementeret som blev begrundet i 5.11. Der kunne argumenteres for, at den næste use case der skulle være arbejdet med, i stedet burde have været Vare CRU (CRUD). Denne use case er nemlig blevet vurderet som den tredje mest komplekse use case og vil derfor have været et logisk valg.

### 7.2 Systemet

Dette afsnit har til formål at belyse diskussionsemner i forhold til det udviklede system.

#### 7.2.1 Begrænsninger i programmet

Efter samtalen med Anders Olesen fremkom det, at ønsket fra ledelsen var noget andet, end hvad der er blevet implementeret. Anders Olesen ønskede at kunne holde overblik over diverse statistikker. Dette kontra de andre problemstillinger blev vægtet mindre vigtigt. Der foreligger dog den mulighed, at dette problem kunne have vist sig at være mere komplekst, end de der blev udvalgt til undersøgelse og implementering. Det blev dog skønnet under diskussion af cost-benefit i afsnit 3.7,

at den ønskede statistik var en del af salgsdelsystemet, og at salgsdelsystemet derfor skulle være implementeret først.

Ydermere blev der lagt vægt på at brugerfladen ikke vil være optimal for statistikker, da det ikke umiddelbart er muligt at fremvise på en professionel og overskuelig måde. Derfor har vi valgt ikke at arbejde med statistik use cases i dette projekt.

### 7.2.2 Kode-duplikering

I forhold til et endeligt system vil det være en idé, at de ansatte også kan fungere som kunder. I mange butikker får ansatte en fast rabat, hvilket er rimeligt at antage, at de også gør her. Derfor vil det på domænemodellen også være en mulighed at associere ansat til kundegruppe. Dette vil gøre, at der er mulighed for at undgå kode-duplikering.

### 7.2.3 Succeskriterierne

Succeskriterierne der blev nævnt i afsnit 4.3 vil her diskuteres. De fire kriterier samt deres begrundelse er:

1. Systemet er stabilt og indeholder ingen kritiske fejl samt pludselige nedbrud.
  - Det er forsøgt at gøre programmet stabilt ved, at bruge exceptions og -handling. Disse "fanger" fejl og beder brugeren om at genindtaste input. Ydermere er der brugt if-sætninger for at kontrollere at et kald til et objekt ikke returnere null.
2. Systemet hjælper de ansatte med salg, lagerstyring mv.
  - Det har ikke været muligt at undersøge og teste om systemet hjælper de ansatte med hverken salg eller lagerstyring.
3. Systemet hjælper lederne med at fastsætte priser og rabatter.
  - Dette punkt er ikke blevet implementeret, da det ikke havde relevans for de use-case som der blev taget udgangspunkt i.
4. Systemet gør det nemmere at registrere varer og færdiggøre f.eks. tilbud.
  - Dette punkt er heller ikke blevet implementeret og er derfor heller ikke relevant at diskutere.



## 8 | Konklusion

I dette afsnit vil der blive lavet en konklusion på projektet og forløbet som helhed. Ud fra interviewet med Anders Olesen fremgik det, at ønsket omkring salgsstatistik var det højst prioriterede ved et nyt system. Dette skæres fra tidligt, da det skal være muligt at sælge noget, før der kan laves eksempelvis salgsstatistikker. Den indledende analyse, IT-forundersøgelse, resulterede i den følgende problemformulering:

*Hvordan kan der udvikles en digital løsning, som opfylder Vestbjerg Byggecenter A/S' krav til et nyt system, hvor det skal være muligt bl.a. at håndtere salg og lager samt oprettelse af nye kunder, varer og leverandørdata, således at dette hjælper både ledelsen og de ansatte i virksomheden?*

Problemformuleringen anses for at være delvist opfyldt. De dele, der er opfyldt, er muligheden for at oprette ordrer, genbestille varer og modtage varer. Dog mangler der flere dele af systemet, før det endeligt kan benyttes.

### 8.1 Gruppeevaluering

Gruppearbejdet har generelt fungeret fint mht. opdeling af de forskellige opgaver. Dette vil sige, at arbejdsopgaverne har været forholdsvis ligeligt fordelt.

Samordning i gruppen har været fin, og der var blevet aftalt et ambitionsniveau ud fra arbejdskontrakten.

Gruppen har formået at strukturere og uddelegere opgaver på en forsvarlig måde, således at der også har været taget hensyn til projektets læringsmål. Alle gruppe-medlemmerne har arbejdet med de forskellige dele.

I de tilfælde, hvor det har været nødvendigt at indsamle information, er der blevet brugt det udleveret materiale og bøgerne fra undervisningen. Og når der har været

brug for at stille spørgsmål til underviserne, har det været muligt at kontakte dem via. mail.

Styring i gruppen har fungeret fint, og der er blevet anvendt de relevante metoder, og det interessante for det udviklet program er blevet dokumenteret og reflekteret over.

I starten af projektperioden blev der udarbejdet følgende tidsplan, som blev næsten helt overholdt til punkt og prikke.

- 30/11-2016: IT forundersøgelse.
- 07/12-2016: Højest prioriterede use cases designes og implementeres.
- 09/12-2016: Næst højest prioriterede use cases designes og implementeres.
- 13/12-2016: Rettearbejde færdig.
- 14/12-2016: Aflevering (senest klokken 15.00).

# Litteratur

- [1] Hans Jørgen Skriver, Erik Staunstrup, and Peter Storm-Henningsen. *Organisation*. Trojka / Gads Forlag A/S, 2012. ISBN 978-87-92098-65-8.
- [2] KOMP s. 62-73 *Costbenefit - udleveret af Lise Klitsgaard*, 2016.
- [3] *Systemudvikling slides fra lektion 1*, 2016. URL <https://ucn.instructure.com/courses/7790/modules/items/73454>.
- [4] *Systemudvikling slides fra lektion 2*, 2016. URL <https://ucn.instructure.com/courses/7790/modules/items/73572>.
- [5] Craig Larman. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Third Edition*. Addison Wesley Professional, 2004. ISBN 0-13-148906-2.
- [6] e conomic. Cost-benefit analyse - hvad er en cost-benefit analyse?, 2016. URL <https://www.e-conomic.dk/regnskabsprogram/ordbog/cost-benefit-analyse>.
- [7] Swifthosting. Swifthosting, 2016. URL <https://swifthosting.dk/bestil-dedicated.php>.
- [8] Prosa. Prosa, 2016. URL <https://www.prosa.dk/>.
- [9] *Programmering slides fra lektion 15*, 2016. URL <https://ucn.instructure.com/courses/7789/modules/items/84744>.
- [10] Devin Balkcom og Khan Academys computing curriculum team Thomas Cormen. Store-o notation, 2016. URL <https://da.khanacademy.org/computing/computer-science/algorithms/asymptotic-notation/a/big-o-notation>.



## A | Udregninger til cost benefit

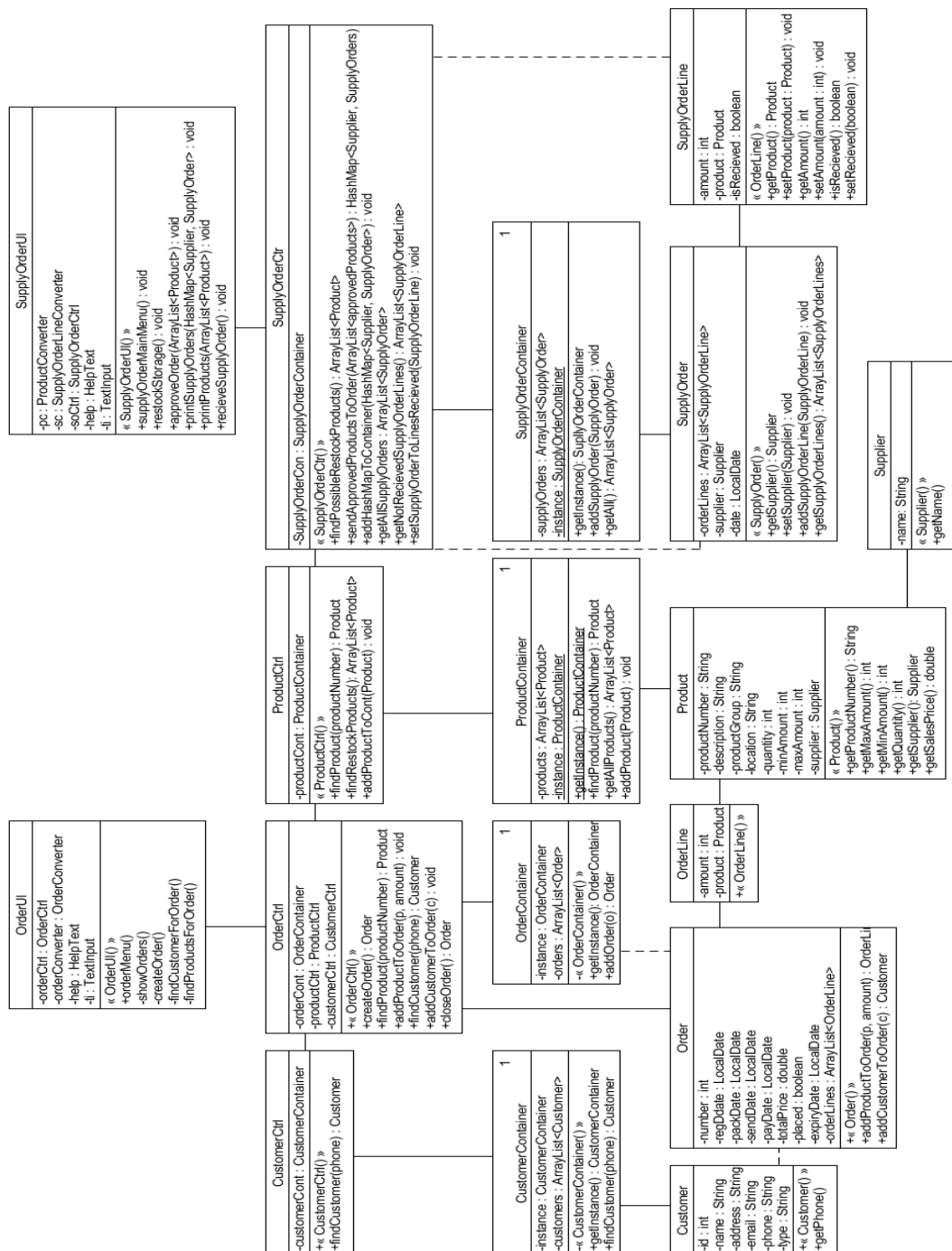
Arbejdsfunktion	Minutter	Antal gange	Tid brugt (minutter)	Tid brugt (timer)	Timeløn (kroner)	Kostpris pr. medarbejder (kroner)
Lager medarbejder pr. uge.	20	50	1000	16.66	150	2500
Kontor medarbejder pr. uge.	20	10	200	3.33	140	466.66
Udlejning pr. uge.	120	1	120	2	140	280
Lager medarbejder	15	50	750	12.5	150	1875
Kontor medarbejder	15	10	150	2.5	140	350
Udlejning	90	1	90	1.5	140	210

**Tabel A.1:** Udregninger til cost-benefit analysen.





# B Designklassediagram



Figur B.1: Samlet designklassediagram.



## C | Brugervejledning

Programmet er opbygget med et tekst user interface.

For at starte programmet køres derfor main metoden i MainMenuUI, klassen.

Opret ordrer:

Vælg undermenuen Ordrer, ved at indtaste 1 i consolen og tryk enter.

Vælg Opret ordrer, og dernæst Tilføj varer, hvor varenummeret indtastes.

Liste over mulige varenummer: 1, 2, 4, 5, 6, 102, 140, 13.

Når de ønskede varer er tilføjet til ordren skal der tilføjes en kunde.

Dette kan gøres ud fra eksisterende kunder ved at indtaste deres telefonnummer.

Telefonnummer for kunder der er oprettet i systemet:

"12345678", "12121212", og "77778888".

Eksisterende ordrer:

Indtast 2 i Ordrer menuen. Der er på forhånd gemt nogle ordrer ved brug af persistent data.

Genbestil varer:

Vælg undermenuen Bestillinger fra hovedmenuen.

Indtast 1 for at tjek lageret for genbestillinger.

En liste vil blive udskrevet af varer der skal genbestilles.

Varene kan vælges at blive godkendt ved at vælge dem ud fra listen.

Når alle varer der ønskes godkendt er valgt, indtastes 0 for done.

Herefter vil en bestillingliste udprintes som enten skal bekræftes eller afbegræftes.

Varemodtagelse:

Vælg undermenuen Bestilling fra hovedmenuen.

Indtast 2 for varemodtagelse.

En liste vil blive udskrevet af varer som er blevet bestilt (Genbestil varer skal udføres først).

Når alle varer der ønsket at sættes til modtager er valgt, indtastes 0 for done.