University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

Table of Contents

1st Semester Projekt, 2025.
Matias Munk Persson, Computer Science.

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

-

IT Transformation Project for Vestbjerg Byggecenter A/S

Introduction

Denne rapport beskriver udviklingsprocessen for det nye IT system til Vestbjerg Byggecenter A/S.
Denne case er en del af et 3 ugers forløb til 1st Semester Computer Science studerende, som skal integrere teoretisk viden i en applikation i den virkelige verden. Studerende skal agere som konsulenter som skal levere en indledende undersøgelse, analyse, system design og en delvis implementering af et IT system ved brug af den Unified Proces (UP) udviklingsmodel.

Preliminary Investigation

Strategic Analysis

Vestbjerg Byggecenter A/S opererer imidlertid med uddateret IT infrastruktur. Salg er behandlet gennem fire terminaler og interne opgaver som bogføring, yderligere så er kundeforespørgsler handlet manuelt. Manglen på integration og automation leder til ineffektivitet, især når det kommer til kontrol af inventar og håndtering af kunderelationer. Firmaet har forespurgt et mere moderne, centraliseret system som forbedrer den operationelle effektivitet, salgsprocesser og inventar kontrol m/ barcode teknologi.

Business Analysis

- ***Structure***

Virksomheden er struktureret med en central ledelse, der fører tilsyn med driften. Salgspersonale håndterer daglige transaktioner, mens specialiserede medarbejdere administrerer pris- og rabatpolitikker.

- ***Leadership***

Ledelsen er aktivt involveret i at definere priser og kræver adgang til salgsstatistikker på højt niveau til beslutningstagning. Deres støtte til it-transformation er stærk, hvilket indikerer parathed til organisatoriske forandringer.

- ***People***

Medarbejderne spænder fra sælgere til administrativt personale. De har brug for et brugervenligt system, der er tilgængeligt på tværs af alle afdelinger. IT-færdigheder

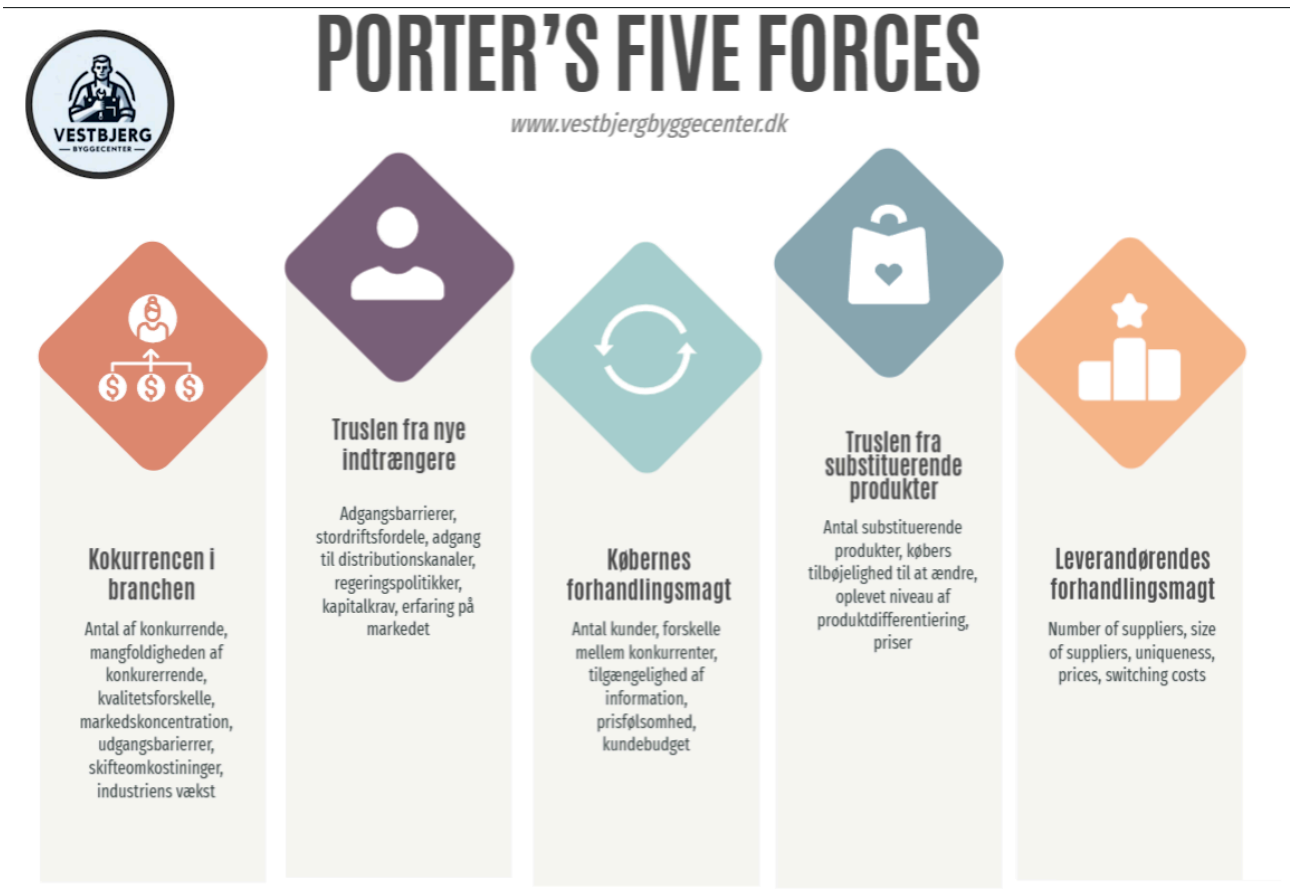
forventes at variere, hvilket kræver intuitivt design og tilstrækkelig træning.

- ***Culture***

Virksomheden viser vilje til at modernisere og investere i langsigtet effektivitet. Kulturen understøtter samarbejde og innovation, men kan kræve forandringsledelse for at sikre smidig tilpasning til nye teknologier.

■ Porter's Five Forces

Konkurrencen: Høj – Konstruktion og gør-det-selv (DIY) centre er forekommende.
Leverandør Kraft: Medium – Afhængig af leverandørernes bygningsmaterialer.
Købekraft: Høj – Kunder har mange alternativer.
Truslen fra substitutter: Medium – Online platforme tilbyder alternativer.
Truslen fra nye indtrængende: Medium – Barrierer eksisterer, men er ikke uoverkommelige.



Vestbjerg Byggecenter opererer i en branche med høj konkurrence, hvor både byggemarkeder og gør-det-selv-centre er udbredte og let tilgængelige for forbrugerne.

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

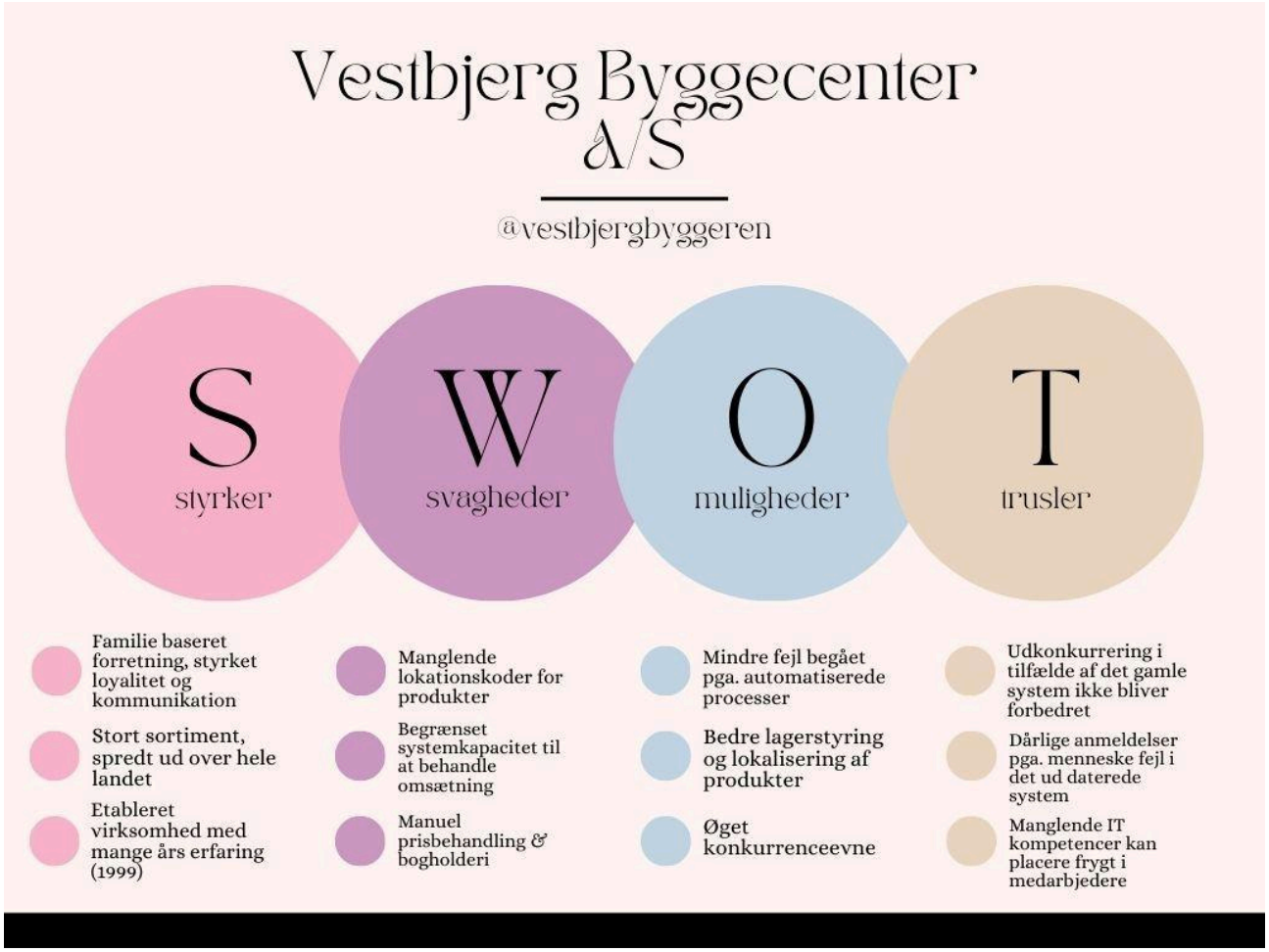
Dette giver kunderne stor købekraft, da de har mange alternativer at vælge imellem. Samtidig har virksomheden en vis afhængighed af sine leverandører af byggematerialer, hvilket placerer leverandørernes forhandlingsstyrke på et mellemniveau.

Der eksisterer også en middel trussel fra substituerende løsninger, særligt i form af online platforme, som kan tilbyde både produkter og inspiration til gør-det-selv-projekter.

Truslen fra nye aktører i markedet vurderes ligeledes som moderat. Selvom der er adgangsbarrierer såsom etableringsomkostninger og branchekendskab, er de ikke uoverkommelige og kan overvindes af nye konkurrenter med den rette strategi.
En stor trussel kunne være en koncern som kommer ind på markedet, bygger kæmpe lagerhuse og stort produktsortiment rundt omkring i Danmark og udvikler et super moderne system som overgår Vestbjerg Byggecenters, til at udkonkurrere dem.

- **SWOT Analysis**

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV



Vestbjerg Byggecenter A/S blev grundlagt i 1999 og er en familieejet virksomhed. Virksomheden har en stærk intern struktur i og med at den er familieejet – kommunikationen er god og loyaliteten blandt medarbejderne kan man ikke sætte spørgsmålstegn til.

Vestbjerg Byggecenter A/S er en veletableret familieejet virksomhed med mange års erfaring siden grundlæggelsen i 1999. Den familiære struktur har bidraget til stærk intern kommunikation og loyalitet blandt medarbejderne, og virksomhedens brede sortiment er tilgængeligt over hele landet, hvilket styrker dens position på markedet.

Virksomheden står dog over for en række interne udfordringer, herunder et forældet IT-system, der ikke understøtter moderne lokationskoder eller automatiseret lagerstyring. Dette resulterer i manuel håndtering af priser og bogholderi, hvilket begrænser effektiviteten og øger risikoen for fejl.

På den positive side giver netop disse svagheder mulighed for vækst og forbedring. Ved at

modernisere IT-systemerne og automatisere arbejdsgange kan Vestbjerg Byggecenter opnå færre fejl, bedre lagerstyring og en stærkere konkurrenceevne.

Dog skal virksomheden også være opmærksom på eksterne trusler. Konkurrenter med mere moderne systemer kan opnå en fordel, og der er risiko for dårlige kundeanmeldelser, hvis fejl i de gamle processer fortsætter. Samtidig kan manglende IT-kompetencer blandt medarbejdere skabe modstand og usikkerhed i forbindelse med nødvendige forandringer.

■ Strategic Alignment

● *Mission*

Vestbjerg Byggecenter har som mission at levere bygningsmaterialer af høj kvalitet kombineret med professionel og personlig service til både entreprenører og private kunder. De bestræber os på at være en pålidelig samarbejdspartner, der skaber værdi gennem faglig ekspertise, tilgængelighed og troværdige leverancer.

● *Vision*

Virksomhedens vision er at udvikle os til at være en moderne og effektiv leverandør i byggebranchen ved at gennemgå en digital transformation. Gennem teknologisk opgradering ønsker de at styrke både deres interne processer og den samlede kundeoplevelse – og dermed positionere dem selv som en fremtidsorienteret aktør i markedet.

● *Values*

Virksomheden bygger deres arbejde og beslutninger på fire kerneværdier:

- **Integritet** – Handler ansvarligt, ærligt og med respekt for kunder og samarbejdspartnere.
- **Effektivitet** – Stræber efter at optimere processerne og sikre høj produktivitet.
- **Kundefokus** – Kunden er i centrum, og løsningerne tilpasses efter deres behov.
- **Innovation** – Der søges løbende nye og bedre måder at drive forretning på gennem teknologi og nytænkning.

● *Goals and Objectives*

For at understøtte virksomhedens vision har vi defineret følgende mål og konkrete målsætninger:

- Modernisere og opgradere interne systemer for at effektivisere arbejdsgange og reducere fejl.
- Forbedre kundeservice gennem hurtigere adgang til opdateret kunde- og

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

produktdata.

- Øge salget ved hjælp af dataanalyse, salgsstatistik og prædiktiv indsigt.
- Skabe en mere datadrevet kultur, hvor beslutninger træffes på et bedre informeret grundlag.

- ***Suggested Strategic Focus***

For at realisere denne strategi anbefales det, at Vestbjerg Byggecenter investerer i et integreret og brugervenligt ERP-system (Enterprise Resource Planning). Systemet skal:

- Centralisere produkt-, kunde- og salgsdata for bedre overblik og beslutningskraft.
- Automatisere lagerstyring og reducere afhængigheden af manuelle processer.
- Skabe grundlag for avanceret rapportering og realtids indsigt i forretningsdata.
- Øge medarbejdernes effektivitet og mindske risikoen for menneskelige fejl.

Focusing

- ***System Vision***

(skriv om hvad system vision er, hvilke behov og krav der skal til, og hvad der godt ved det nye implementeret system, hvem er målgruppen, hvad er hovedfunktionerne, fordele og forretningsværdi hvilke problemer løser det, og hvorfor er det vigtigt)
Et centraliseret, rollebaseret salgs- og lagerstyringssystem, der strømliner driften, forbedrer datatilgængelighed og understøtter stregkode baseret på lagersporing.

- ***Risk Analysis***

Risk ID	Risk description	Risk type	Potential indicators	Likelihood (1-5)	Impact (1-5)	Risk score (Likelihood x impact)	Mitigation plan
R1	Modstand fra medarbejdere ved implementering	Organisatorisk	Negative kommentarer, lav systembrug, mange spørgsmål	3	4	12	Involver brugerne tidligt, tilbyd træning og understreg fordelene ved systemet
R2	Fejl ved overførsel af data fra det gamle system	Teknisk	Manglende eller forkerte kundedata/varer efter migrering	4	5	20	Test dataoverførsel grundigt, lav backups, brug erfarne udviklere
R3	Barcode-scannere virker ikke korrekt med nyt system	Teknisk/operationel	Systemet registrerer ikke scannede varer korrekt	3	4	12	Test hardware integration tidligt, samarbejd med leverandør, lav

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

							fallback-løsning
R4	Uklar kravspecifikation fra ledelsen	Kommunikation smæssig	Modstridende krav, ændringer undervejs, utilfredshed	3	3	9	Afhold regelmæssige møder med ledelsen, brug use cases og prototyper
R5	Systemnedbrud ved lancering	Teknisk	Systemet fejler ved første brug, lang ventetid, utilgængelighed	2	5	10	Gennemfør tests og simuleringer, hav en fall back-plan og support klar

Risk type kan være følgende:

- Estimation
- Organizational
- People
- Requirements
- Technology
- Tools

A risk analysis looks at risks, i.e. something that can happen, and if it does - how will it impact the business / project etc?

Steps:
identify risks
analyze likelihood and consequence
plan to address or minimize effect
monitor the risks in some way

Employee resistance: Medium likelihood, High impact
Mitigation: Training and gradual rollout.

Technical issues: Medium likelihood, Medium impact
Mitigation: Phased development and testing.

Data loss: Low likelihood, High impact
Mitigation: Regular backups.

- ***Stakeholder Analysis***

I forbindelse med udviklingen af IT-systemet til Vestbjerg Byggecenter er der foretaget en

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

stakeholderanalyse for at identificere nøgleinteressenter. Formålet er at sikre inddragelse og hensyn til alle relevante parter gennem projektet.

Analysen omfatter både interne og eksterne aktører. Ledelsen har stor interesse og indflydelse, da systemet påvirker driften og beslutningsgrundlaget. Medarbejderne, især sælgerne, er primært brugere og kan give vigtig feedback. Kunderne påvirkes indirekte gennem forbedret service, mens eksterne udviklere og konsulenter kræver klare krav og løbende dialog. Leverandører påvirkes i mindre grad via ændringer i lager- og bestillingssystemet.

Analysen hjælper med at planlægge kommunikation og forudse potentielle udfordringer i projektets faser.

Stakeholder name	**Contact person**	**Impact**	**Influence**	**What is important to the stakeholder?**	**How could the stakeholder contribute to the project?**	**How could the stakeholder block the project?**	**Strategy for engaging the stakeholder**
Ledelse (Ejere)	Anders	High	High	Effektiv drift, vækst, bedre, beslutningsgrundlag	Godkendelse af investeringer, beslutningstagning	Nedprioritere projektet, udskyde beslutninger	Ugentlige statusmøder, økonomisk rapportering
Sælgere/Medarbejdere	Casper/Thomas	High	Medium	Brugervenligt system, lavere fejlrate, nem kundesøgning	Deltage i test og feedback, komme med input til brugergrænseflade	Modstand mod ændringer, ignorerer det nye system	Workshops og træningsdage før og under implentering
Kunde(private)	Stamkunder	Medium	Low	Hurtigere espedition, korrekt rabat og kundedata	Give feedback via support eller fysisk i butikken	Skifter til konkurrent hvis oplevelse er dårlig	Brug af spørgeskeamer og mundtlig feedback i butikken
IT-konsulent/udvikler	Ekstern partner	High	High	Klare krav, god kommunikation og adgang til nøglepersoner	Designe, implementere og teste det tekniske system	Dårlig kommunikation, misforstå krav	Tydelig projektplan og kontaktperson i Vestbjerg
Leverandører	Eksterne firmaer	Low	Medium	Præcis varebestilling og lageropdatering	Justere integrationer og ordreformater hvis nødvendigt	Ignorerer integration eller ændringer	Tidlig dialog og evt. API-specifiktion

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

Management: High interest, High influence. Sales Staff: High interest, Medium influence. Customers: Medium interest, Low influence. IT Consultants (Students): High interest, Medium influence.

- ***Project Planning***

Inception: Week 1 Elaboration: Week 1–2 Construction: Week 2–3 Transition: End of Week 3.

- ***Cost / Benefit Analysis***

Costs: Development time, Training sessions, Hardware upgrades. Benefits: Reduced time spent on sales and stock control, Fewer manual errors, Faster customer service, Strategic insights through data analytics.

- # Unified Process

Dette projekt følger Unified Process metodikken, som strukturerer udviklingen ind i 4 hovedfaser: Indledningsfasen, Afviklingsfasen, Konstruktionsfasen og Overgangsfasen.

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

■ Inception

Actor-goal table:

Actor	Goal	Steps
Customer	Place order	Gennemse kataloget Tilføj varer til indkøbskurven, kassen, modtager bekræftelse
Customer	Cancel order	Log ind, vælg ordre, vælg annullerings mulighed, bekræft annullering
Customer	Receive refunds	Anmod om tilbagebetaling, vent på godkendelse, modtag betaling
Sales assistant	Register customer	Indtast kundeoplysninger, valider data, indsend formular
	Register order	Tag kundeanmodning, indtast ordreoplysninger, bekræft
Sales assistant	View order	Søg ordre-id, åbn ordredetaljer
Sales assistant	Modify order	Få adgang til ordre, rediger detaljer, gem ændringer
Sales assistant	Handle product	Opdater produktoplysninger, tjek lager, flag problemer
Warehouse employee	CRUD product	Create, read, update, delete produkt forekomster
Warehouse employee	Pack order	Find produkter, pak sikkert, etiketter pakken

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

Warehouse employee	Dispatch order	Planlæg afhentning, overlever til kurer
Warehouse employee	Order products	Tjek lager, vælg leverandør afgiv ordre.
Accountant	Process payment	Verificer transaktion, registrer betaling, bogfør købet i systemet.
Accountant	Manage invoices	Opret faktura, send til kunden, track status
Accountant	Generate reports	Vælg tidsramme, kompilér data, eksporter data.
Manager	Manage users	Tilføj/rediger/fjern brugerkonti.
Manager	Set pricing	Lur markedet, juster priserne, publicer ændringer
Manager	View reports	Tilgå dashboard, vælg filtre, vis/print output
Finance system	Process transactions	Modtag data, validér, eksekver transaktion.
Finance system	Manage invoice	Synkronisér med interne data, opdater status.
Finance system	Apply discounts	Kontroller berettigelse, anvend discount.
Finance system	Set pricing	Modtag input, opdater prisdatabasen.

Workflow

For at kunne visualisere forløbet på ordrebehandlingen, anvender vi standardiserede symboler i workflow-diagrammet.

Workflow-diagrammet anvender en række standardiserede symboler til at strukturere og

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

visualisere forløbet i ordrebehandlingen. Aktiviteter og handlinger som f.eks. "Tjek kunde", "Pak ordre" og "Send faktura" er repræsenteret som rektangler, hvilket tydeligt angiver, at der er tale om operationelle trin i processen, som enten kræver brugerhandling eller systemaktivering. Disse udgør grundelementerne i det trinvise forløb, hvor hver opgave udføres én efter én.

Beslutningspunkterne fremgår som diamanter og er anvendt flere steder i forløbet, eksempelvis ved vurdering af lagerstatus, leveringsform og betalingsstatus. Disse steder udgør logiske forgreninger, hvor processen ændrer retning afhængigt af givne betingelser. Det giver mulighed for at tilpasse arbejdsgangen baseret på hændelser som "Hvis kunden ikke eksisterer" eller "Modtager en annullering".

Der er også brugt parallelle stier, adskilt af sorte bjælker, som muliggør samtidig behandling af flere aktiviteter – fx pakning af ordre og udsendelse af faktura – uden at de skal vente på hinanden. Dette giver mulighed for effektiv udnyttelse af ressourcer, da flere opgaver kan udføres parallelt.

Slutpunkter og startpunkter markeres med henholdsvis sort cirkel og målskive-symboler, der indikerer hhv. begyndelsen og afslutningen på processen. Derudover optræder et tidsbaseret symbol (timeglas) til at håndtere overskridelse af betalingsfrist, hvilket fungerer som en event drevet afvigelse fra den normale arbejdsgang.

Samlet set viser diagrammet en struktureret og styret arbejdsproces, hvor kombinationen af konkrete opgaver, valg undervejs og mulighed for samtidige handlinger sikrer en effektiv og fleksibel ordrebehandling med indbygget håndtering af undtagelser og afslutning.

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

Fig 1: Dette diagram er et flowdiagram for forretningsprocesser, der repræsenterer arbejdsgangen for behandling af en produktforespørgsel og ordre, herunder håndtering af kunder, checks, salg, levering, fakturering og potentielle betalingsforsinkelser eller annulleringer.

Her er en trin-for-trin forklaring af flowet:

Start:
"Vare Forespørgsel"

Lagerkontrol:
"Vare ikke på lager" → "Bestil vare hjem".

Ellers → "Tjek kunde".

Kundehåndtering:
Hvis kunden ikke findes, skal du gå til "Opret kunde".

Efter kunden er bekræftet eller oprettet → "Opret salg". Opdelt i to stier:
Efter oprettelse af salget opdeles processen i:

1. Ordreopfyldelse Sti:
"Pak ordre"

Vælg derefter mellem:

"Afhentning"

"Levering"

2. Faktura Sti:

"Send faktura"

Derefter:

"Modtag betaling"

Eller hvis betalingen er forsinket → "Betalingsfrist på 30 dage er overskredet"

→ "Send rykker"

1st Semester Projekt, 2025.
Matias Munk Persson, Computer Science.

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

Går til sidst til "Modtag betaling" eller:

Hvis "Modtager en annullering" → "Annuller ordre"

Konklusion:
Når betalingen er modtaget, og ordren er opfyldt (afhentning eller levering), fortsætter processen til:

"Afslut ordre" (Luk ordre)

Slut på processen.

Annullerings Afdeling:
Hvis kunden annullerer på noget tidspunkt, markeres ordren som "Annuller ordre" (Annuller ordre).

Dette flow diagram viser, hvordan en virksomhed håndterer produktsalg - fra forespørgsel til opfyldelse, fakturering og opfølgning efter salg, inklusive undtagelser som forsinket betaling eller annullering.

Lad mig vide, om du ønsker en oversat version eller et forenklet flow.

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

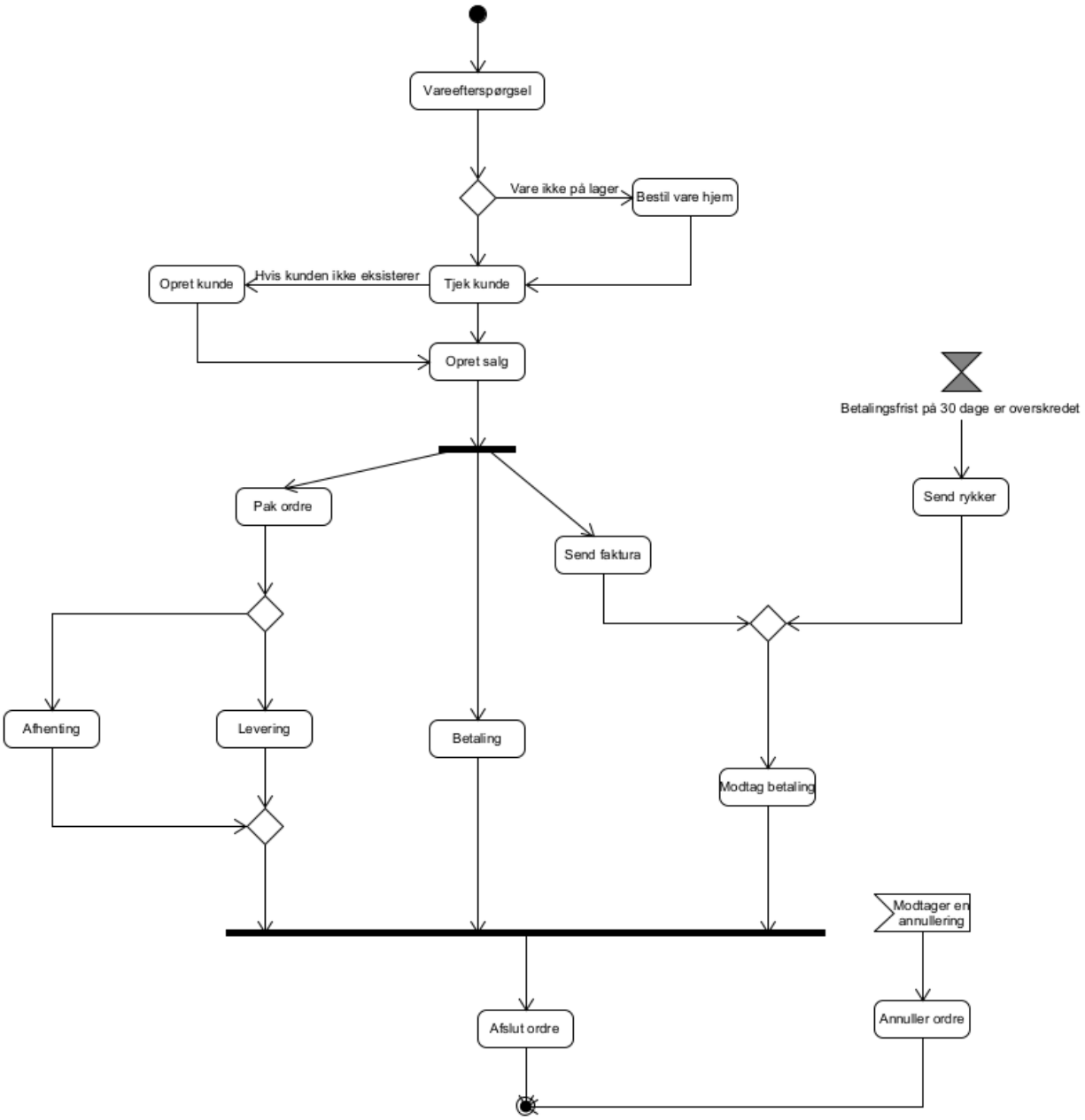


Fig 1

Diagrammet på figur 1 viser et klassediagram (UML – Unified Modeling Language), som visualiserer strukturen i et objektorienteret system i forbindelse med et lager- & ordrestyringssystem. Det viser klasser, attributter, metoder såvel som relationer til andre klasser med nedarvning.

Lad os gennemgå hovedkomponenterne én ad gangen:

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

Kerneklasser og -relationer

Order

- Har en reference til:
    - `Customer` (kunden der laver ordren)
    - `Employee` (medarbejderen som håndterer ordren)
    - `Location` (Lokation på varelager)
    - `Product` (Produktet som sælges til kunden)
- Har også:
    - Liste over produkter (`List<Product>`)
    - Leveringsstatus, dato og rabat
- Metoder inkluderer fx `addProduct`, `getTotalPrice`, `getDiscount`

Controller-klasser

Disse styrer adgangen til og manipulation af de respektive domæne objekter:

OrderController

- Håndterer ordrer
- Metoder til at tilføje/fjerne produkter, ændre kunde, beregne total mv.

CustomerController

- Tilføjer/fjerner kunder
- Finder kunder

EmployeeController

- Samme som ovenstående, men for ansatte

1st Semester Projekt, 2025.
Matias Munk Persson, Computer Science.

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

LocationController

- Håndterer lokationer og produkter på lager

ProductController

- Håndterer produkter og søgninger i produktkatalog

Product (og underklasser)

- Attributter: `product_id`, `name`, `price`, `weight`, `dimensions`, mv.
- Underklasser:
    - `SimpleProduct`
    - `UnitProduct`
- Indeholder metoder til at få og sætte attributter, samt beregne vægt og pris

---

Customer og Employee

klasserne repræsenterer brugere i systemet:

- Attributter: navn, adresse, telefonnummer, mv.
- Har også metoder til at hente og ændre data

Location

- Bruges til at holde styr på, hvor produkter opbevares
- Attributter: `city`, `managerName`, `address`, mv.
- Indeholder metode til at finde lagerværdi og antal varer

Container-klasser

1st Semester Projekt, 2025.
Matias Munk Persson, Computer Science.

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

Hver controller arbejder sammen med en *Container*-klasse (en slags lager eller database i hukommelsen):

- `ProductContainer` → Holder liste af produkter
- `OrderContainer` → Holder liste af ordrer
- `CustomerContainer` → Kunder
- `EmployeeContainer` → Ansatte
- `LocationContainer` → Lokationer

Relationer

- Stærk **sammensætning og aggregation** mellem objekter. Eksempler:
    - `Order` er afhængig af både `Product`, `Customer` og `Employee`
    - Controllers arbejder med deres respektive containere
    - Produkt har to specialiserede former via arv: `SimpleProduct` og `UnitProduct`

Eksempel på arbejdsgang:

1. En kunde placerer en ordre.
2. `OrderController` opretter en `Order`, knytter en `Customer`, tilføjer `Product`.
3. `Order` bruger produktdata fra `ProductContainer`.
4. Når ordren skal pakkes, bruges `Location` til at finde produkterne.
5. Betaling og afslutning af ordren styres via metodekald i `Order`.

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

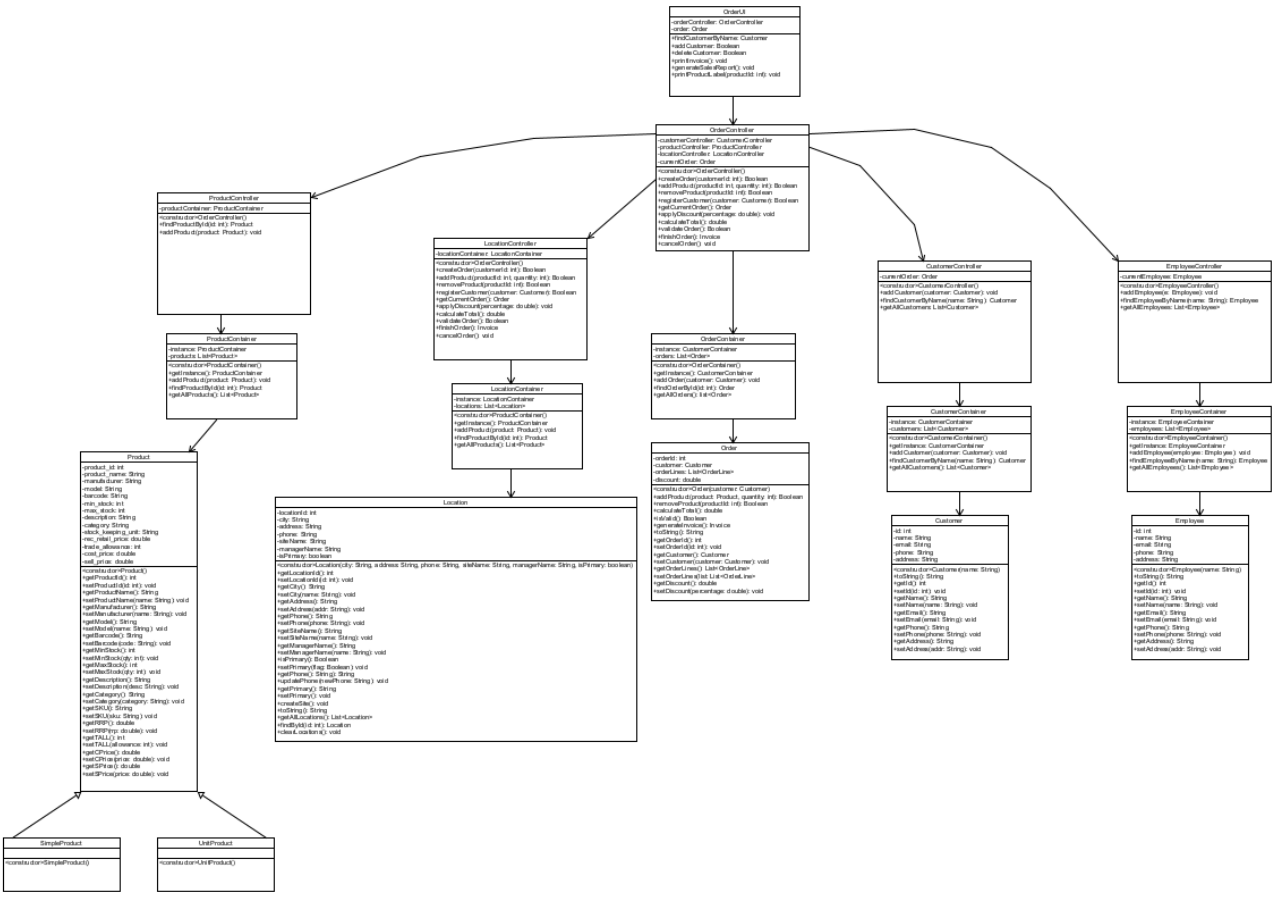


Fig 2

Fig 3

Kundeoprettelse og Ordreoprettelse (venstre side)

Aktører og objekter:

- `:CustomerUI` (Brugerfladen)
- `CustomerController`
- `:Customer`
- `:CustomerContainer`

Forløb:

1. `CustomerUI` kalder `createOrder()` → `Order` oprettes.
2. Derefter kaldes `addProduct(...)` og `findCustomerByName(...)`

1st Semester Projekt, 2025.
Matias Munk Persson, Computer Science.

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

3. Hvis kunden ikke findes, kaldes:
    - `create(...)` på `Customer` with ID, navn, adresse, e-mail.
    - `addCustomer(...)` på `CustomerController`
4. Kunden gemmes i `CustomerContainer`
5. `getName()` bruges til at vise/valider kundeoplysninger

Pointen: Brugeren opretter en ordre, og hvis kunden ikke findes, oprettes og gemmes kunden.

---

2. Ordreoprettelse og Håndtering (midten)

Aktører og objekter:

- `:OrderUI`
- `OrderController`
- `:Order`
- `:OrderLine`
- `:OrderContainer`

Forløb:

1. `createOrder()` kaldes → nyt `Order`-objekt oprettes
2. `addProduct(...)` og `findCustomerByName(...)` kaldes
3. Ordren gemmes i `OrderContainer`
4. Produkter og kundeoplysninger tilføjes

Pointen: Viser hvordan en ordre oprettes og kobles med kunde og ordrelinjer.

Produktoprettelse og Søgning (højre side)

Aktører og objekter:

- `:ProductUI`
- `CustomerController` (ser ud til at være brugt fejlagtigt i stedet for fx `ProductController`)
- `:Customer` (burde måske være `Product`)

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

- `:CustomerContainer` (igen, burde nok være `ProductContainer`)

Forløb:

1. `createProduct()` og `findProductById(...)` kaldes
2. Produkt gemmes i containeren
3. `getName()` kaldes

OBS: Diagrammet har sandsynligvis en fejl — `CustomerController` og `Customer` burde være `ProductController` og `Product`.

Opsummering:

Disse sekvensdiagrammer beskriver **hvordan controllers, containers og objekter samarbejder** i tre typiske brugsscenarier:

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

Diagram	Hovedfunktion	Objekter i fokus
1	Opret kunde og ordre	`Customer`, `CustomerController`
2	Opret og tilføj ordre	`Order`, `OrderLine`, `OrderController`
3	Opret og find produkt	(Fejl: bruger `Customer` i stedet for `Product`)

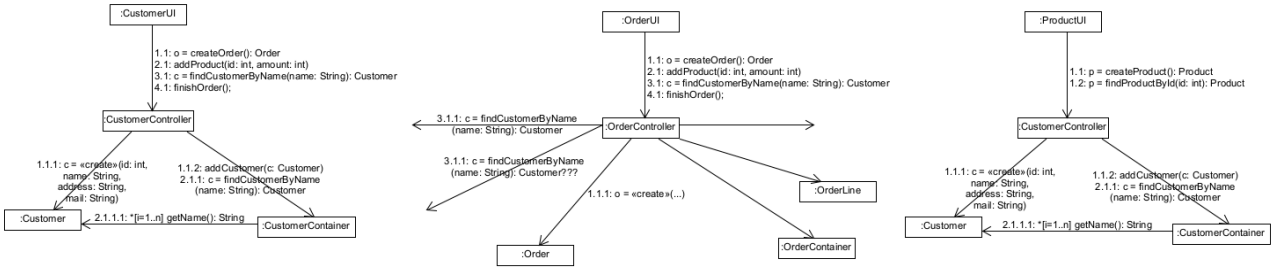
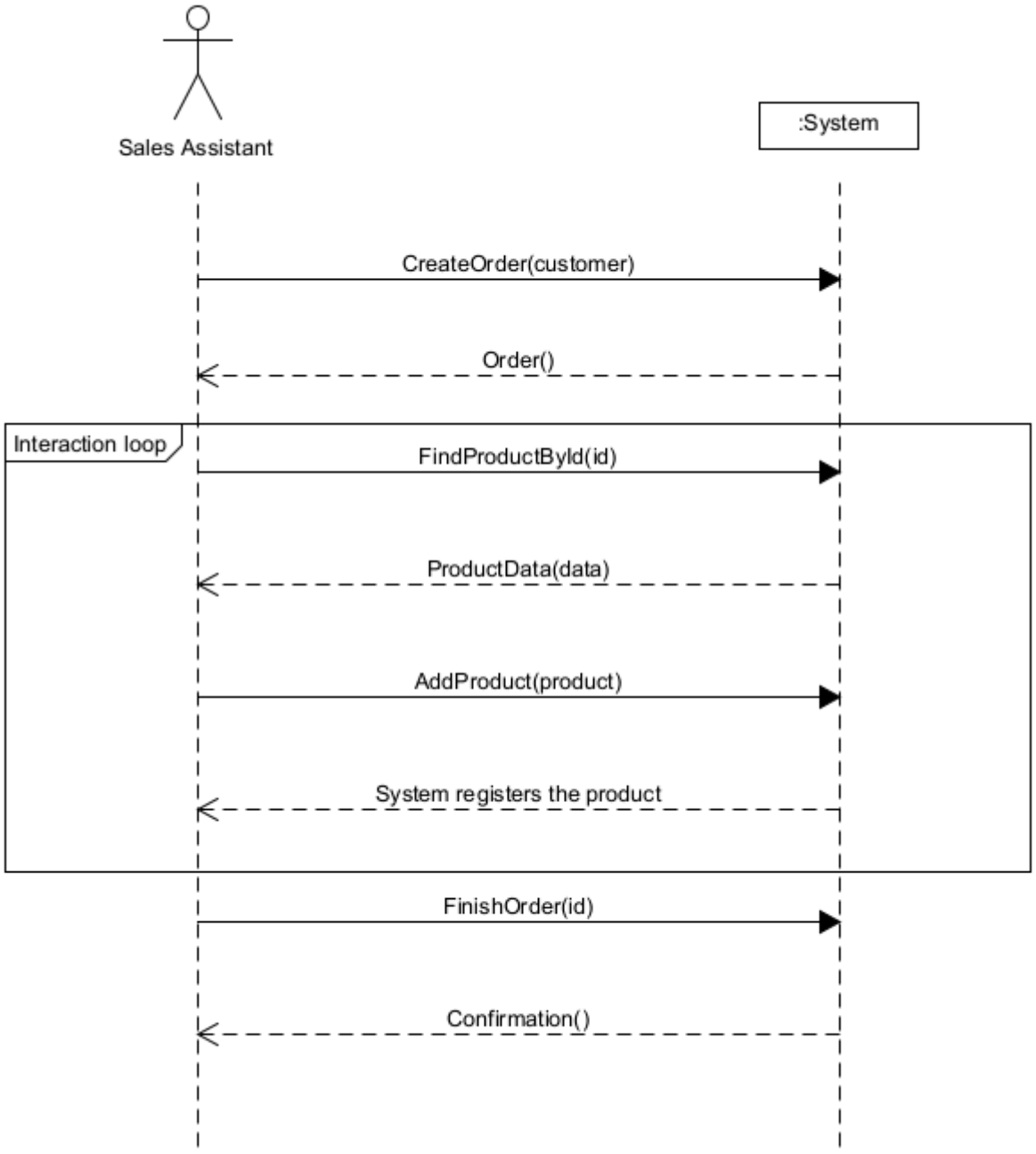
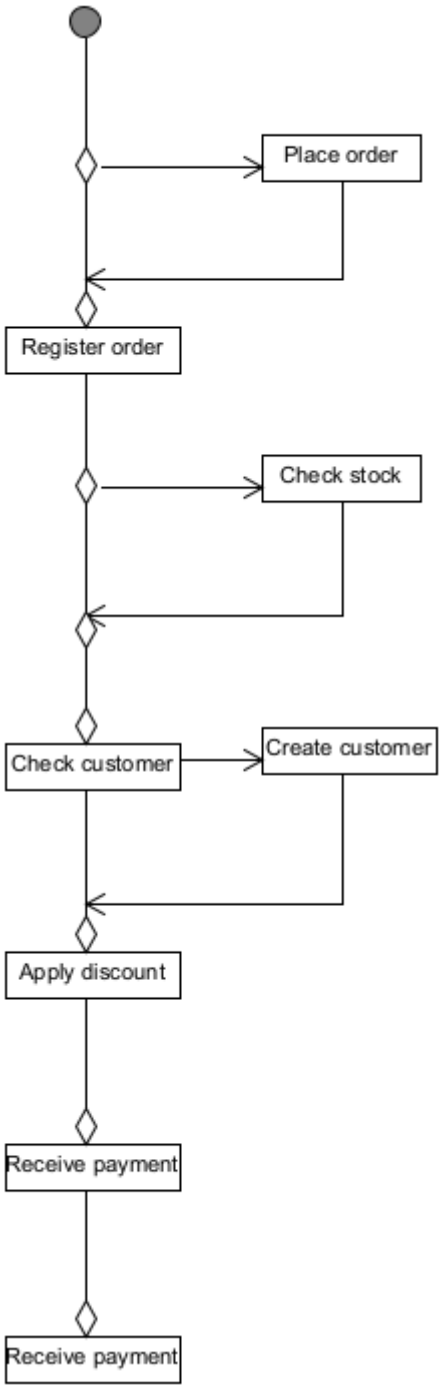


Fig 3

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

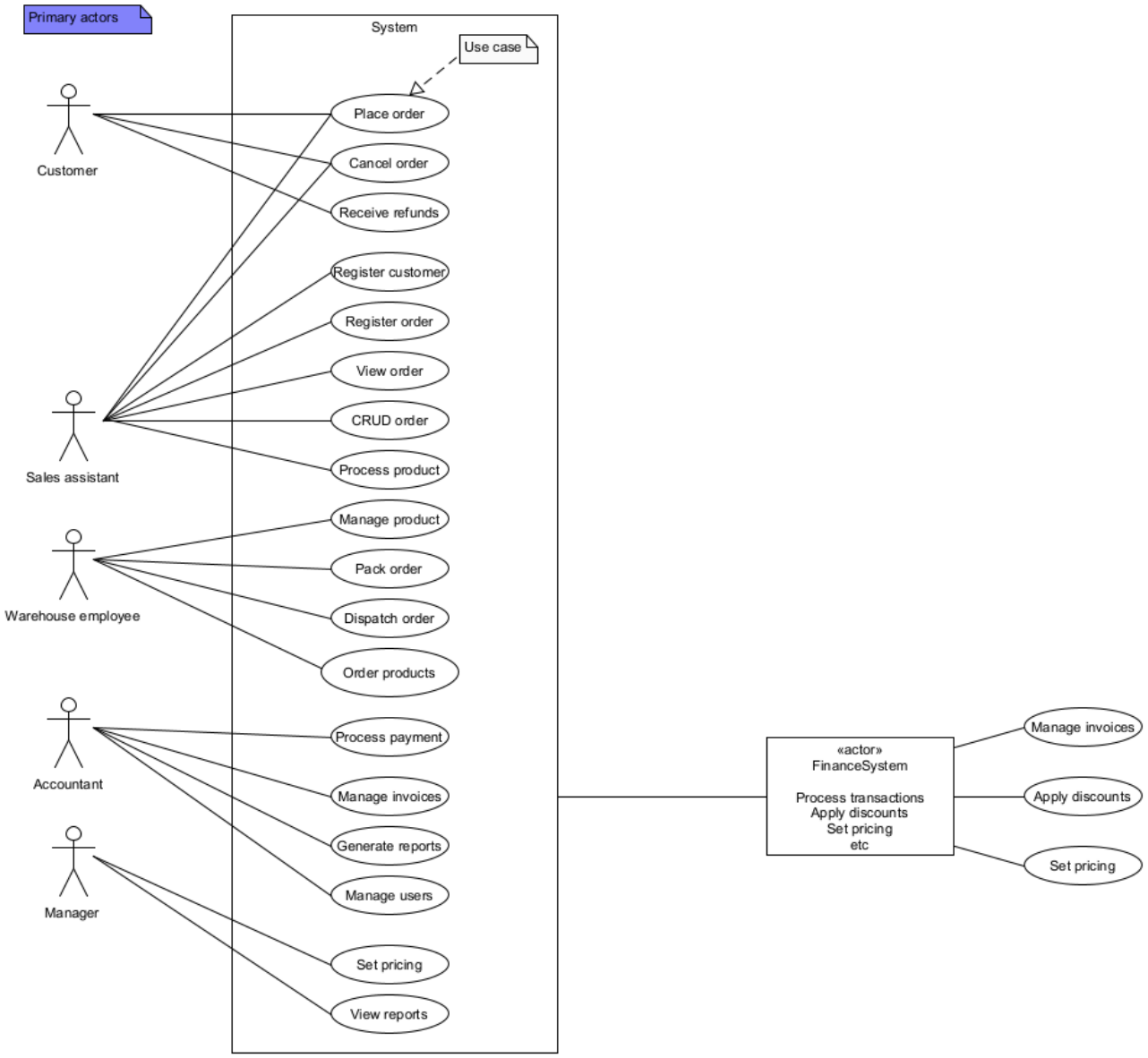


University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV



1st Semester Projekt, 2025.
Matias Munk Persson, Computer Science.

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV



Mockups
Use case diagram
Briefs and casuals

1st Semester Projekt, 2025.
Matias Munk Persson, Computer Science.

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

Use case prioritet tabel

Use case	High value business	Technical risk	Coverage	Total score	Rank
Place order	5	2	5	12	1
Cancel order	2	1	5	8	4
Receive refunds	2	1	5	8	4
Register customer	5	3	4	12	1
Register order	3	3	3	9	3
View order	2	2	2	6	6
CRUD order	2	5	5	6	1
Process product	3	1	3	6	6
Manage product	3	1	3	7	6
Pack order	4	1	2	6	6
Dispatch order	4	1	3	7	5
Order products	3	1	3	7	6
Process payment	2	1	5	7	5
Manage invoices		1	5	7	5
Generate reports	2	1	5	10	5
Manage	2	1	5	9	6

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

users					
Set pricing	4	1	5	10	2
View reports	3	1	5	9	3

Klasse Kandidatliste
Domænemodel
Elaboration
Fully dressed på use case

Use case	**Create place order**	
Actor	Sales assistant, customer	
Pre-conditions	Customer is in store, selected a product and ready to proceed with payment.	
Post-conditions	Order is recorded in the system and an order confirmation is generated.	
Frequency	Daily	
Flow of events	1. Customer wishes to purchase a product	
	2. Actor greets the customer and creates new order with customer	3. System creates new order
	4. Actor specifies ID of the product	5. System returns product data, subtotal and total
	6. Actor adds product to the order	7. System registers the product

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

	Step 4 to 7 is repeated until all products are added	
	9. Actor finish the order	10. System saves the order
Alternative flow	**Condition**	**Alternative steps**
1	No product available	1. System shows an "Out of Stock" message. 2. Actor suggests alternative products.
2	Customer is not found	1. System displays an error message 2. Actor creates a new profile
3	Payment fails	1. System shows error message. 2. Actor requests alternate payment method.
4	Customer changes mind mid-process	1. Actor cancels the current transaction. 2. System logs cancellation for analytics.

Operations Kontrakter

Operation contract	
Operation	CreateOrder(customer)

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

Use-cases	Når en kunde starter en ny ordre via brugergrænsefladen. Systemet skal oprette og gemme en ny ordre, som tilknyttes kunden.
Precondition	Kunden eksisterer i `CustomerContainer`. Der må ikke være en åben ordre for kunden (valgfri regel).
Post conditions	En `Order`-instans er oprettet og tilføjet til `OrderContainer`. Ordren er associeret med kunden og har tomme ordrelinjer og status "pending".

Operation contract	
Operation	FindProductById(Id)
Use-cases	Bruges når brugeren søger efter et specifikt produkt, fx ved oprettelse af ordre eller visning af produktinformation.
Precondition	Produktcontaineren er initialiseret og indeholder mindst ét produkt.
Post conditions	Returnerer det produkt-objekt med det givne ID, hvis det findes. Ellers returneres null/fejlmeddelelse.

Operation contract	
Operation	AddProduct(product)
Use-cases	Bruges til at tilføje et valgt produkt til en eksisterende ordre. Typisk kaldt efter produktvalg i UI.
Precondition	En aktiv ordre findes og er under oprettelse. Produktet findes og er ikke allerede tilføjet.

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

Post conditions	Produktet tilføjes som en ny ordrelinje i den aktuelle ordre. Ordren opdateres med ny totalpris og antal.

Operation contract	
Operation	FinishOrder(Id)
Use-cases	Bruges når en ordre er færdig og klar til afslutning – fx efter betaling eller godkendelse.
Precondition	Ordren med det angivne ID eksisterer og har mindst ét produkt. Den må ikke allerede være markeret som "afsluttet".
Post conditions	Ordren ændrer status til "completed" eller "finished". Den kan ikke længere ændres. Systemet kan trigge fakturering eller forsendelse.

Kommunikations Diagram

Design Klassediagram

Analysis

Key use cases include registering customer/product/contractor, creating offers, managing orders and invoices, generating reports, and barcode-based stock control.

Design

Modular design med frontend, backend og moduler for administration, workflow for salg, lager og rapportering.

UML diagrammer i referencer.

Implementation

Software udvikles i Java med indlæsning af produkter fra konfigurationsfil i form af alm.

1st Semester Projekt, 2025.
Matias Munk Persson, Computer Science.

tekst, key value pair. Del af implementationen omfatter registrering og salgsstart.

Kode eksempler

Fig 1 viser entry point for programmet uden nogle command-line arguments.

Der bliver instantieret nogle controllere og forudindlæst nogle produkter som der er mulighed for at bruge i programmets main loop.

```
public static void main(String[] args) {

    // Initialize controllers and scanner
    scanner = new Scanner(System.in);
    customerController = new CustomerController();
    productController = new ProductController();
    orderController = new OrderController(customerController, productController);

    // Preload some products
    preloadProducts();

    boolean running = true;

    while (running) {
        System.out.println("\n=== ERP Order System ===");
        System.out.println("1. Create new order");
        System.out.println("2. Add product to current order");
        System.out.println("3. Remove product from order");
        System.out.println("4. Apply discount to order");
        System.out.println("5. View current order");
        System.out.println("6. Finalize order");
        System.out.println("7. Cancel order");
        System.out.println("0. Exit");

        System.out.print("Choose an option: ");
        String input = scanner.nextLine();

        switch (input) {
            case "1" -> createOrder();
            case "2" -> addProductToOrder();
            case "3" -> removeProductFromOrder();
            case "4" -> applyDiscountToOrder();
            case "5" -> viewCurrentOrder();
            case "6" -> finalizeOrder();
            case "7" -> cancelOrder();
            case "0" -> running = false;
            default -> System.out.println("Invalid option, please try again.");
        }
    }

    System.out.println("Exiting ERP System.");
}
```

Fig 1

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

Fig 2 viser forudindlæsning af produkter

```
private static void preloadProducts() {
    productController.addProduct(new Product("Hammer", "ACME", "H100", "1111", 10, 100, "Heavy hammer", "Tools", "SKU001", 79.95, 0, 49.95, 59.95));
    productController.addProduct(new Product("Cement Bag", "BuildWell", "C250", "2222", 20, 500, "50kg Cement", "Materials", "SKU002", 149.00, 0, 89.50, 99.95));
    productController.addProduct(new Product("Screw Pack", "ScrewIt", "S300", "3333", 30, 300, "200 screws", "Hardware", "SKU003", 35.00, 0, 15.00, 25.00));
}
```

Fig 2

Fig 3 viser hvordan vi opretter en ordre. Processen fungerer således at vi finder/opretter en kunde og derefter opretter ordren i kundens navn.

```
private static void createOrder() {
    System.out.print("Enter customer name: ");
    String name = scanner.nextLine();

    Customer customer = customerController.findCustomerByName(name);
    if (customer == null) {
        System.out.println("Customer not found. Creating new customer.");
        customer = new Customer(name);
        customerController.addCustomer(customer);
    }

    if (orderController.createOrder(customer.getId())) {
        System.out.println("New order created for " + customer.getName());
    } else {
        System.out.println("Error creating order.");
    }
}
```

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

Fig 4 viser hhv. hvordan vi kan fjerne produkt fra ordrer, tilføje rabat på ordrer, vise ordrer og afslutte ordrer.

```
private static void removeProductFromOrder() {
    System.out.print("Enter product ID to remove: ");
    int productId = Integer.parseInt(scanner.nextLine());

    if (orderController.removeProduct(productId)) {
        System.out.println("Product removed.");
    } else {
        System.out.println("Could not remove product.");
    }
}

private static void applyDiscountToOrder() {
    System.out.print("Enter discount percentage (0-100): ");
    double percent = Double.parseDouble(scanner.nextLine()) / 100.0;

    orderController.applyDiscount(percent);
    System.out.println("Discount applied.");
}

private static void viewCurrentOrder() {
    Order current = orderController.getCurrentOrder();
    if (current == null) {
        System.out.println("No active order.");
        return;
    }

    System.out.println("\n--- Current Order ---");
    System.out.println("Customer: " + current.getCustomer().getName());
    for (OrderLine line : current.getOrderLines()) {
        System.out.println(line);
    }
    System.out.println("Total: $" + current.calculateTotal());
}

private static void finalizeOrder() {
    if (orderController.validateOrder()) {
        Invoice invoice = orderController.finishOrder();
        System.out.println("Order finalized. Invoice generated:");
        System.out.println("Order ID: " + invoice.getOrder().getOrderId());
        System.out.println("Customer: " + invoice.getOrder().getCustomer().getName());
        System.out.println("Total: $" + invoice.getOrder().calculateTotal());
    } else {
        System.out.println("Cannot finalize. Order is invalid.");
    }
}
```

Fig 4

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

Fig 5 viser 2 metoder fra Order modellen og viser hvordan vi tilføjer og sletter produkter fra ordrer.

```
public boolean addProduct(Product product, int quantity)
{
    if (product == null || quantity <= 0)
        return false;

    for (OrderLine line : orderLines)
    {
        if (line.getProduct().getId() == product.getId())
        {
            line.addQuantity(quantity);
            return true;
        }
    }

    orderLines.add(new OrderLine(product, quantity));
    return true;
}

public boolean removeProduct(int productId)
{
    return orderLines.removeIf(line -> line.getProduct().getId() == productId);
}
```

Fig 5

Test

Enhedstest, integrationstest og simuleret brugeraccepttest ved hjælp af eksempeldata (Try Me)

Standard for Coding

Følg navnekonventioner, konsekvente indrykninger, 3-lags arkitektur, modulær kode.

Conclusion

Det nye system forslag adresserer aktuelle udfordringer ved at centralisere data og automatisere arbejdsgange. Teamet anvendte UP-metoder til at levere en skalerbar prototype og fik værdifuld erfaring fra den virkelige verden.

References

Larman, C. (2004). Applying UML and Patterns. Jacobson, I., Booch, G., & Rumbaugh, J. (1999). The Unified Software Development Process. Porter, M. E. (1979). How Competitive Forces Shape Strategy. Harvard Business Review. Lecture notes and project guidelines from ITFU.

University College Nordjylland
Technology & Business
Sofiendalsvej 60
9200 Aalborg SV

Appendix

Appendix A: Full Case Description. Appendix B: UML Diagrams. Appendix C: Prototype Screenshots.

1st Semester Projekt, 2025.
Matias Munk Persson, Computer Science.