University College Nordjylland

Professionsbacheloruddannelsen i Datamatik

DMA-CSD-S211



Vejledere:

*Anita Lykke Clemmensen*

*Finn Ebertsen Nordbjerg*

*Torben Larsen*

*Bodil Nørgaard*

Gruppe 4:

*Ali Khalil Dbouk, Kavinsan Karunanithi, Malthe Morsing Larsen, Mark Panabe Bovbjerg, Pavlo Skyba*

*Github-repo: https://github.com/mmlucn/1.-Semester-Projekt*

Afleveringsdato: 17/12-2021

Abstract

In this project the group had been asked to develop a program for the company VestBjerg A/S. The company wants to be able to track sales records and have more control of all the stock they have in their warehouses. They also want to be able to search up sales assistants records as well as the costumers' records. Throughout the project the group shows how it solved the assignment with the help of different IT feasibility studying methods such as workflow analysis and mock-ups. Then the group worked out the demands for the system by using use cases and working out a domain model. The use cases were then analyzed and used to design communications diagrams, that were then used to build the projects design class diagram. The design class diagram is last step before implementation, and it gives an overview of the layout, that shows how the system is to be coded.

Indholdsfortegnelse

[Indledning 5](#_Toc90632377)

[Kode konvention 5](#_Toc90632378)

[Tidsplan 5](#_Toc90632379)

[Del-konklusion 6](#_Toc90632380)

[Business Case 6](#_Toc90632381)

[Organisationskulturen 6](#_Toc90632382)

[Porters five forces 8](#_Toc90632383)

[SWOT-analyse 10](#_Toc90632384)

[Systemets mål og vigtighed 10](#_Toc90632385)

[Costs and benefits 11](#_Toc90632386)

[IT-Forundersøgelse: 11](#_Toc90632387)

[Medarbejdermål tabel 11](#_Toc90632388)

[Workflow 11](#_Toc90632389)

[Mock-ups 15](#_Toc90632390)

[Resultat af Tænk-høj test 17](#_Toc90632391)

[Systemvision 18](#_Toc90632392)

[Ikke-funktionelle krav: 18](#_Toc90632393)

[Punkopstillet liste 18](#_Toc90632394)

[Funktionelle krav: 19](#_Toc90632395)

[Use-case Diagram 19](#_Toc90632396)

[Use-case Briefs 19](#_Toc90632397)

[Use-case prioriteringstabel 20](#_Toc90632398)

[Use-case fully dressed 20](#_Toc90632399)

[Fully dressed: Salg 21](#_Toc90632400)

[Fully dressed: Kasse salg 21](#_Toc90632401)

[Fully dressed: Udlån værktøj 22](#_Toc90632402)

[Fully dressed: Returner item 22](#_Toc90632403)

[Fully dressed: Returnere værktøj 23](#_Toc90632404)

[Informationskrav: 23](#_Toc90632405)

[Kandidat til klasse – vurderingstabel 23](#_Toc90632406)

[Domænemodel 24](#_Toc90632407)

[Analyse og design af use-cases: 25](#_Toc90632408)

[Systemsekvensdiagram (SSD) 25](#_Toc90632409)

[Use-case Salg 26](#_Toc90632410)

[Use-case kassesalg 27](#_Toc90632411)

[Use-case Udlån værktøj 28](#_Toc90632412)

[Use-case aflever værktøj 29](#_Toc90632413)

[Operationskontrakt 30](#_Toc90632414)

[Design: 31](#_Toc90632415)

[Interaktionsdiagram 31](#_Toc90632416)

[Design klasse diagram 36](#_Toc90632417)

[Programmet 37](#_Toc90632418)

[Gruppeevaluering 37](#_Toc90632419)

[Konklusion 38](#_Toc90632420)

[Referenceliste 38](#_Toc90632421)

[Bilag 38](#_Toc90632422)

[Bilag 1: 38](#_Toc90632423)

[Bilag 2: 39](#_Toc90632424)

[Bilag 3: 39](#_Toc90632425)

[Bilag 4: 40](#_Toc90632426)

[Bilag 5: 40](#_Toc90632427)

[Bilag 6: 41](#_Toc90632428)

[Bilag 7: 41](#_Toc90632429)

[Bilag 8: 42](#_Toc90632430)

[Bilag 9: 42](#_Toc90632431)

[Bilag 10: 43](#_Toc90632432)

[Bilag 11: 43](#_Toc90632433)

[Bilag 12: 44](#_Toc90632434)

[Bilag 13: 44](#_Toc90632435)

[Bilag 14: 45](#_Toc90632436)

[Bilag 15: 45](#_Toc90632437)

[Bilag 16: 46](#_Toc90632438)

[Bilag 17: 46](#_Toc90632439)

[Bilag 18: 47](#_Toc90632440)

[Bilag 19: 47](#_Toc90632441)

[Bilag 20: 48](#_Toc90632442)

[Bilag 21: 48](#_Toc90632443)

[Bilag 22: 49](#_Toc90632444)

[Bilag 23: 50](#_Toc90632445)

[Bilag 24: 51](#_Toc90632446)

# Indledning

I dette semesterprojekt har gruppen fået tildelt en case, som der skal arbejdes ud fra. Gruppen skal arbejde med firmaet VestBjerg Byggemarked A/S, som er et byggemarked hvor der en tømmer afdeling og en DIY-afdeling. Med en årlig omsætning 69 millioner DDK, lever deres gamle Windows vista software ikke op til deres forventninger. Formålet med dette projekt er at lave en dyb analyse af VestBjerg Byggemarked, derefter analysere, designe og implementere et software, som udfylder firmet behov. Et nyt system som skal kunne følgende ting, registre nye kunder, produkter og give informationer omkring leverandør. Det er ledelsen som skal sætte priserne for varerne, ledelsen vil også gerne have muligheden for at se, hvad den enkelte medarbejder sælger for. Der skal også være muligt at oprette en ordre for varer, bekræfte en ordre, leveringsnote og mulighed for at sende en faktura for ordren. Registrering og kontrol af varer skal kunne ses med stregkode. I forhold til lager, vil de gerne se hvor meget de har på lager, men også hvor varen ligger henne.

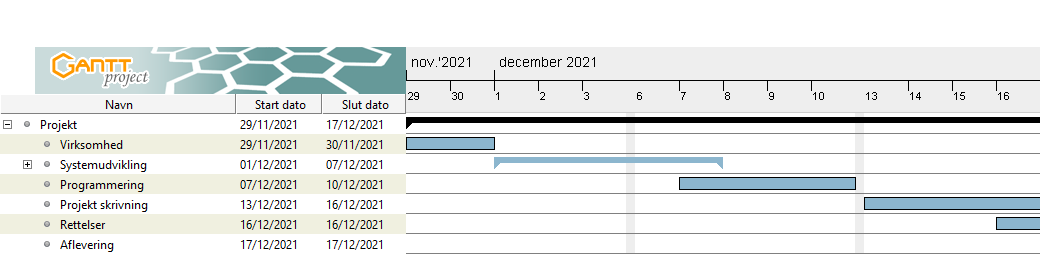
Der bruges unified process (UP) metode, som skabelon for projekt. UP metoden beskriver en udviklingsproces i form af en række opgaver, der skal udføres for at kunne forvandle brugerkrav til et IT-system. Der startes først med at lave en business analyse af VestBjerg Byggemarked, derefter kigges der på hvilket slags system der skal programmeres og hvordan det skal bygges op, ved hjælp af forskellige metoder og diagrammer.

# Kode konvention

Vi har igennem dette projekt stræbet efter at følge Javas kode konvention. Værd at nævne for Javas Kode Konvention er blandt andet at klasser skal følge CamelCase navngivningen f.eks. *DetteErEnKlasse* og at variabler skal starte med et småt bokstav – f.eks. detteErEnVariabel. Derudover skal en kodeblok (compound statement på engelsk) starte i slutningen af en sætning (statement på engelsk) og slutte på en linje for sig. If-statements skal også have en tilhørende compound statement - også selvom den kun indeholder én instruktion.[[1]](#footnote-2)

# Tidsplan

Vi har som noget af det første i dette projekt, udarbejdet en tidsplan ved hjælp af et Gantt-skema. Til det har vi brugt et program med navnet GanttProject[[2]](#footnote-3). Overordnet kører projektet – for vores vedkomne, fra d. 29-11-2021 til og med d. 17-11-2021, altså knapt 3 uger. Vi har delt vores tid op Gantt-skema op i 6 dele.

* Virksomhed  
  Vi analyserer vores case ved hjælp af de værktøjer vi har fået i virksomhedsfaget. Til det har vi afsat 2 dage, d. 29/11 og d. 30/11.
* Systemudvikling  
  Vi havde afsat 7 dage fra d. 01/12 til d. 07/12, til systemudvikling. Vi blev i gruppen enige om, at vi ville forsøge at gøres meget umage med netop denne del, til dels fordi den er svær, men selvfølgelig også for at gøre kodningen af programmet nemmere.
* Programmering  
  Til programmering har der været afsat 3 dage. Fra d. 7/12 til d. 10/12.
* Projekt-skrivning  
  De sidste 5 dage i projektet - fra d. 13/12 til d. 17/12 - afsatte vi til projektskrivning. Vi blev enige om at vi ville forsøge at blive færdige med første udkast til projektet til d. 16/12, altså dagen før aflevering. Så kunne vi bruge den sidste dag på at rette projektet samt at aflevere den.

## Del-konklusion

Eftersom at vi ikke har så meget erfaring endnu, har det også været svært at estimere hvor lang tid hver del tager. Set i bakspejlet kunne vi måske godt have brugt en dag eller to mindre, på systemudviklingen og tilføjet dem til programmeringen. Men overordnet set, er vi tilfredse med vores tidsplan.

# Business Case

## Organisationskulturen

Anders Olesen er lidt gammel dags når det kommer til udførelsen af arbejdet, og forstår godt at arbejdet ikke kan udføres ligesom i gamle dage. Derfor har han uddelegeret ansvaret til hans to sønner og nogle af de andre gamle ledere. Når det kommer til motivationen, gør han sit bedste for at motivere hans folk i hans ansvarsområde som de andre ledere og kontor. Den administrative afdeling og de ansatte har et højt ansvar og frihed til at arbejde, da det ikke er hans primære fokusområde.

Han er en meget synlig person i virksomheden som kender alle medarbejdere og har et godt forhold til dem. Han forventer at medarbejderne er hårdtarbejdende og positive, specielt mod kunderne, men også mod kollegaerne og management. Hvis han ikke er tilfreds med en medarbejders indsat og attitude vil han give feedback, så de har en chance for at forbedre. Hvis det ikke hjælper er han ikke bange for at fyre den medarbejder, men det sker ikke normalt da hans medarbejder har været i virksomheden i mange år.

Grundet hans alder og uddannelse er Thomas Olesen mere opmærksom på hvordan han behandler hans medarbejdere, da han har taget en uddannelse i ledelse på UCN T&B. Han holder af hver medarbejder og vil gerne have at de trives i virksomheden. Han har årlige interviews med medarbejderne for at sikre medarbejderne er værdsat for hvad de gør og give dem flere muligheder for udfordrende opgaver. Thomas er formand i personaleforening and gør meget for at skabe en god atmosfære i virksomheden. Han arrangerer kurser og arrangementer for at forbedre sammenholdet.

Casper Olesen har en mere praktisk uddannelse indenfor bygningsmarked og tømmer handel. Han har suppleret hans uddannelse med en deltids uddannelse i ledelse. Hans afdeling minder meget om Thomas’ men er mere detalje orienteret og kontrollerende i hans management stil på grund af hans praktiske viden indenfor tømmer industrien. Han vil meget gerne hjælpe hans medarbejder med deres opgaver så godt som muligt.

**Organisationsstruktur.**

Kigger vi på organisationsformen, kan vi konkludere at Vestbjerg Byggecenter har ”Den Basale form” da arbejdsopgaverne er meget enkle, men markedet er dynamisk.

I de forskellige kulturmetaforer er de meget fokuseret på personerne, så de ligger meget i ”Familien”

**Style of leadership.**

Ledelsesskabs begreberne består af Leadership, Coaching og Management.

Leadership er hvor lederne inspirerer og motivere deres medarbejdere, og samtidig skaber et miljø hvor de kan lærer og sørger for at medarbejderne stemme bliver hørt.

Coaching er hvor der er fokus på medarbejdernes udvikling og at de bliver anerkendt for deres arbejde. Samtidig med at de får jævnligt feedback.

Management er der fokus på planlæggelse af virksomhedens opgaver, og der kontrolleres og evalueres af faglighed

Disse 3 områder er vigtige indenfor ledelse.

Anders Olesen.

Kigger man på Blake & Moutons ledergitter, kan vi konkludere at Anders Olesen er en klar Holdleder (9,9), da han går meget op i at vise hensyn til hans medarbejder og produktionen.

Thomas Olesen

Kigger man på Blake & Moutons ledergitter, kan vi konkludere at Thomas Olesen er en Middle of the road ledelse (9,5) med større hensyn til medarbejderne. Grundet han er 9,5 er fordi han har årlige samtaler med medarbejderne og gør meget for at skabe et godt miljø. Samtidigt laver hans også arrangementer og kurser.

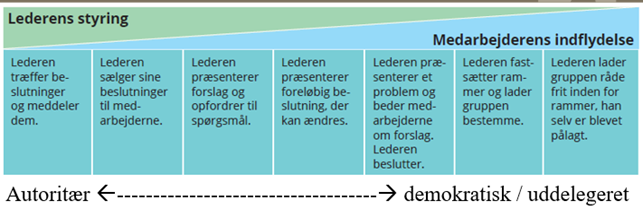
Casper Olesen

Kigger man på Blake & Moutons ledergitter, kan vi konkludere at Casper Olesen er en holdleder (9,9), da hans arbejdstilgang er ligesom Thomas’ men er mere detalje orienteret i hans management style.

Ifølge Jack Bobos ledelsesstile kendetegner ”Den Analytiske” ledelsesstil meget dem alle sammen.

Hvis vi skulle sætte medarbejderne i Mcgregors X- og Y-syn, er de meget Y-syn. Kigger man på medarbejderens forhold til deres arbejdstilgang er de meget interesseret og engageret, hvilket vil sætte dem i Y-syn kolonnen. Hvis man kigger på medarbejderes ønsker til arbejdet, har de en større frihedsgrad til at udvikle sig, hvilket vil sætte dem i Y-syn kolonnen. Man kan også se at medarbejder gerne vil have tryghed og sikkerhed i arbejdet, kan vi konkludere at de er i X-syn kolonnen. Medarbejderne er villige til at påtage sig er ansvar hvis det er meningsfyldt, hvilket vil sige at de er i Y-syn kolonnen.

Hvis vi skal vurdere en ledelsesform ud fra Lewins tre ledelse model vil vi sige at de ligger mod den autoritær side, da de har magt over.



## Porters five forces

De fem konkurrencekræfter (*Porters five forces*) er:

1. Konkurrencen imellem direkte konkurrenter i industrien

2. Nye spillere i industrien

3. Leverandørens forhandlingsmagt

4. Kundernes forhandlingsmagt

5. Trussel om substitution

Disse fem kræfter kan have en positive eller negativ indflydelse på konkurrencen i branchen, så længe at det er en profitabel branche. Ved at analysere de fem kræfter, kan vi danne os et billede af konkurrencen i industrien og dermed få et overblik over hvor attraktiv branchen er.

Kigger vi først på konkurrencen imellem de direkte konkurrenter i industrien, ved vi at Vestbjerg byggecenter A/S er i tæt samarbejde med andre virksomheder i samme branche og ønsker at forsætte med at arbejde med andre virksomheder. Vestbjerg byggecenter A/S er også en del af XL-byg. Dette er en fordel da de har et samarbejde, i stedet for at de er direkte konkurrence med XL-byg som er en af de store spillere i branchen.

Det er lidt svært at være en ny spiller i byggemarkedsindustrien, da folk allerede kender de store kæder som fx XL-byg eller Stark. Det betyder at de har mere tillid til disse store og “ældre” firmaer, hvilket gør dem til kundernes favorit. Skal et ny byggemarked på banen, skal den have godt styr på de vigtigste punkter indenfor byggemarkedsindustri – Kvalitet, pris og bæredygtighed. For at et byggemarked skulle kunne forblive i industrien, skal det have god kontakt med kunderne, og altid have en fornuftig løsning klar, for begge parter. Dette sørger firmaet for faste og loyale kunder, der højst sandsynligt kommer igen, hvis de skal bruge noget.

Ved leverandørens forhandlingsmagt kigger vi på, hvem har af de to parter som har forhandlingsmagten. Kigger vi på Vestbjerg Byggecenter ved vi at de har lidt over 1000 faste kunder, dette betyder så at de har brug for at få leveret varer ofte. Når det kommer til at få leveret disse varer, er der en række leverandør, som kan levere disse varer. Dette kan være en ulempe for leverandøren, da der er flere konkurrenter der kan levere det samme produkt. Hvilket giver Vestbjerg byggecenter en fordel, når det kommer til at få leveret varer til en lav pris. Men hvis der en begrænset antal af leverandører som kan levere varer, skifter forhandlingsmagten til leverandøren. Da de sælger et produkt, som der ikke mange der kan levere.  
Her vil vi kigge på hvilke andre muligheder kunderne har indenfor det samme marked som Vestbjerg Byggecenter, og kigge på prisfølsomheden i kunderne.  
Da VestBjerg byggecenter er lokaliseret nord for Aalborg, vil der være en større befolkning af kunder, men samtidigt er der også en del andre virksomheder såsom Bauhaus, Silvan, Jem og Fix, Stark og Harald Nyborg. Virksomheden har lige omkring 1000 faste kunder som har en konto i DIY-centeret, og er divideret ind i 22 grupper med hvert deres forskellige former af betaling og tilbud. Markedet er meget konkurrencedrevet, så virksomheden giver kunderne speciale tilbud og med muligheden for at give kunderne tilbud kan de konkurrere med de andre virksomheder og samtidigt beholde kundeloyaliteten.

Så kundernes forhandlingsmagt har en stor indflydelse i hvordan vi skal fastsætte vores priser, men samtidigt bliver vi også nødt til at ændre priser ift. Import og den internationale pris.

Ser man på truslen om substitution i byggebranchen, kan vi se at der er mange andre steder der tilbyder det samme som Vestbjerg gør, er der en trussel om substitution. Heldigvis er en af styrkerne for Vestbjerg at de har faste lokale kunder, men hvis Vestbjerg på et tidspunkt vælger at hæve priserne, vil kunderne nemt og hurtigt kunne skifte til en konkurrent.

* Analysis of the project stakeholders and their expectations to a new system

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stakeholder | Interest | Influence/  attitude | Contribution | Importance | Handling |
| Anders Olesen | Vækst og  vedligeholdelse | High/  Positive | Leadership/  Knowledge | High | Old fashioned, need to learn |
| Management | Vækst og  vedligeholdelse | High/  Positive | Leadership | High | Need to learn  the system |
| Medarbejder | Bedre kunde hjælp | Low/  Positive | Shop Maintenance | Medium | Need to learn the system |
| Aktionærer | Vækst | Medium/  medium | Money | Medium |  |
| Kunder | Større udvalg af produkter | High/  High | Money | High |  |

Forventninger fra stakeholderne beskrives nedenfor.

Anders Olesen.   
Forventer at systemet vil gøre det muligt at udvide kundegruppen og produkter for følgende år.

Management.  
 Et system hvor de kan lave salgsstatistikker over nye kunder, produkter, entreprenører og en statistik over medarbejderne. System skal også kunne lave en registrering for et tilbud. Der skal også være mulighed for at lave en ordre, bekræftelse af ordre, afsendelse af ordre, følgeseddel og faktura.  
 Det skal også være muligt at bruge stregkoder for inventar kontrol. På lageret skal det være muligt at se minimum lager og maksimum lager af produkter samtidigt med at det skal kunne bestille nye produkter når produktets lager når under minimumsbeholdning. Der skal også være et tjek om at kundebestilling af et produkt ikke overgår produktets beholdning.

Medarbejder.  
 Medarbejderne skulle kunne være i stand til at registrere nye kunder, produkter og information om entreprenørerne. Det er dog ikke muligt for alm. Medarbejdere at sætte priser og tilbud på varer, dette er kun tilgængeligt for medlemmerne af “management” teamet. Det skal være nemt at logge ind som medarbejder, det skal på samme tid dog også være sikkert.

* Considerations about management change

Vi tænker på at tilføje en IT-leder med et hold med 1-2 personer under sig til vedligeholdelse af det nye system, ellers tænker vi ikke at ændre noget på management.

Management behøves ikke at laves om da det er godt at have dedikeret folk til hver afdeling af køkken, baderum og deputy manager. Samtidigt behøves der ikke laves om i tømmerafdelingen, fordi vi tænker der ikke behøves en manager, da det ikke er så viden nødvendigt i forhold til køkken og baderum.

## SWOT-analyse

|  |  |
| --- | --- |
| **Styrker** | **Svagheder** |
| Effektiv personalegruppe  Familieejet virksomhed  Samarbejde med fx. XL byg  Få medarbejdere  Store tilbagevendende kunder som står for en stor del af indkomsten | Forældet teknologi  Familieejet virksomhed  Få medarbejdere  Holder ikke øje med hvor meget deres salgsassistenter sælger for |
| **Muligheder** | **Trusler** |
| Interesserede investorer  Udvidelse i form af afdelinger i andre byer  Flere store kunder | Eventuelle konkurrenter i samme by  De store kunder finder et andet sted at handle |

**Vision**   
Vores vision er at være kundernes nummer 1 valg, når det kommer til kvalitets byggemateriale. Vi ønsker at være det fornuftige valg en kunde foretager sig.    
Vi vil gerne skabe et godt samarbejdsmiljø for vores arbejdsgruppe, for dermed at have den bedst mulige arbejdsstruktur og motivation.

**Mission**

Virksomhedens mission er at give kunderne et udvalg af kvalitetsrige samt bæredygtige byggematerialer og specielle materialer, der kan bruges til at udføre kundernes ønsker indenfor byggerier og andre projekter.

**Strategiske mål:**

* Øge den årlige omsætningen med 10%
* At udvide udvalget af produkter og få flere kunder indenfor næste par år.

## Systemets mål og vigtighed

Målet er at løse de problemer og mangler som Vestbjerg A/S oplever fordi de ikke har et opdateret system, der kan give overblik over deres lagre og arbejdere. Systemet er vigtigt da deres nuværende system er gammelt, og et nyt system vil give virksomheden muligheden for at holde bedre styr på produkterne og arbejdspladsen, og dermed vil virksomheden også undgå at spilde for meget tid. Hvis arbejdet kører hurtigere, vil det også være nemmere at holde kunderne tilfredse.

## Costs and benefits

Der vil være et engangsbeløb for it-systemet, og så er det permanent hos virksomheden. Virksomheden vil så år efter år tjene flere penge og vokse mere som virksomhed med det nye system. Med et forbedret overblik over kunder, salg, produkter og medarbejdere, som virksomheden får gennem systemet, vil virksomheden hurtigt kunne tjene pengene tilbage som de har givet for systemet og derefter vokse nemmere som virksomhed.

Systemet vil for virksomheden være en langsigtet investering, som ikke kun vil gøre arbejdet nemmere for dem, men også hjælpe dem med at vokse som virksomhed, få flere kunder som de kan skabe et bedre forhold med, og dermed også give en vækst i omsætning.

Alt i alt vil systemet være en klog investering for virksomheden, da arbejdspladsen har brug for et nyt system, som kan hjælpe virksomheden med at vokse, og personalet er også blevet trætte af deres nuværende system.

# IT-Forundersøgelse:

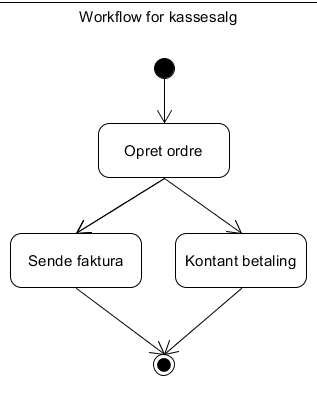
## Medarbejdermål tabel

|  |  |  |
| --- | --- | --- |
| Medarbejder | Mål | Trin |
| Sales Assistent | Opret ny ordre | Kunde bekræfter hvilke han skal have, og derefter laver S.A dem til en ordre |
| Sales Assistent | Registrer ny kunde | En ny kunde bliver registreret |
| Sales Assistent | Oprette et salg | Kunden vil købe nogle produkter, så S.A registrerer salget i systemet |
| Sales Assistent | Katalogisering | En ny vare skal sættes på en fast plads |
| Sales Assistent | Sender faktura | Sidst på måneden sendes en faktura ud til kunderne |
| Sales Assistent | Udlån af værktøjer | Værktøjet kan lånes ud via en registrering |
| Sales assistent | Registrer nyt produkt | Der skal registreres at bogen er blevet afleveret af låner, og så skal bogen sættes på plads |
| Sales assistent | Bestil produkter | Ringer leverandøren eller bestille varer igennem deres system |
| Sales assistent | Ændre pris | S.A skal nedsætte varer |

## Workflow

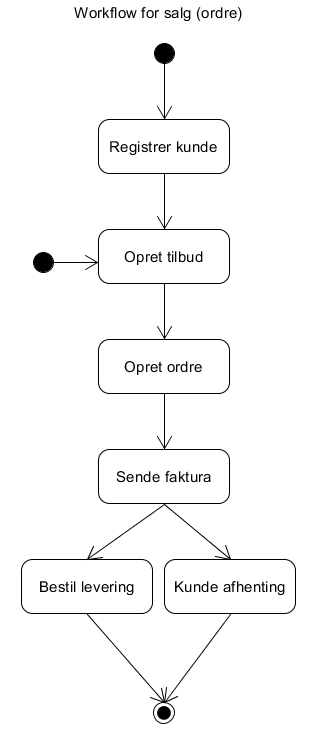
Et workflow bliver lavet ud fra de mål som er blevet lavet i medarbejdermål tabellen, og informationer i virksomheden.

Workflowet har med formål at vise en rækkefølge i mindre opgaver som skal løse et primært eller større opgave i virksomheden.



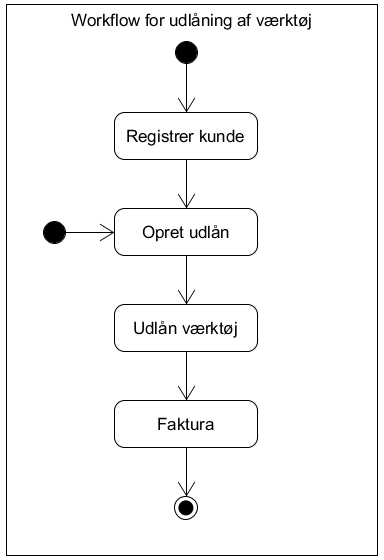
Workflow: Kassesalg

I kassesalget starter det med at oprette en ordre, hvorefter kunden kan vælge om de vil betale med kontant eller faktura, hvorefter ordren afsluttes.



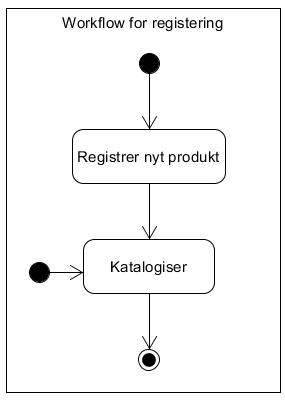
Workflow: Salg (Ordre)

Der er 2 starts punkter i dette workflow, da man kan være registeret som kunde eller ikke kunde. Når en kunde er registeret og har fundet varerne, kan vi oprette et tilbud til dem, hvorefter tilbudet bliver lavet om til en ordre. Hvis kunden allerede eksisterer i systemet kan vi gå direkte hen og oprette tilbuddet, som derefter bliver til en ordre. I dette salg betaler kunden med faktura, hvorefter de har mulighed for at bestille levering eller afhentning selv.



Workflow: udlån af værktøj

I dette workflow er der 2 startpunkter. Da man kan være ny kunde eller allerede være oprettet som kunde. Hvis vi starter med at oprette en kunde kan vi derefter starte med at oprette et lån. Hvis kunden allerede er oprettede starter vi med at oprette et udlån. I det næste skridt kan værktøjet lånes ud, hvis den er ledig. Efter udlån af værktøj sendes der en faktura ud til kunden.

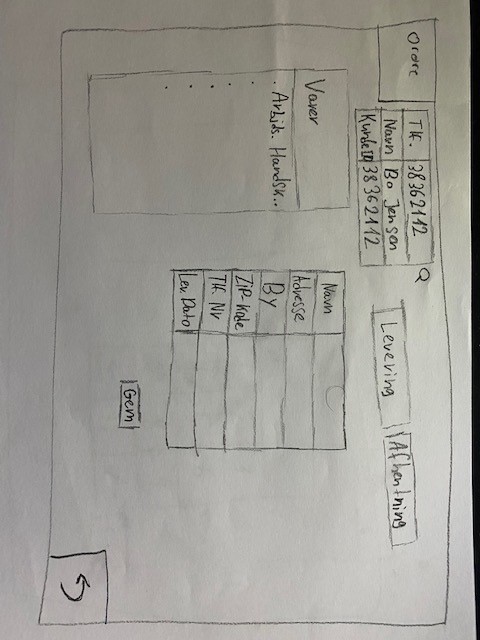
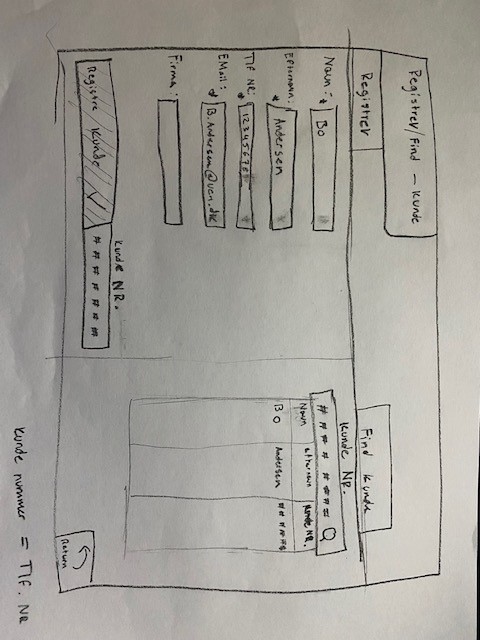
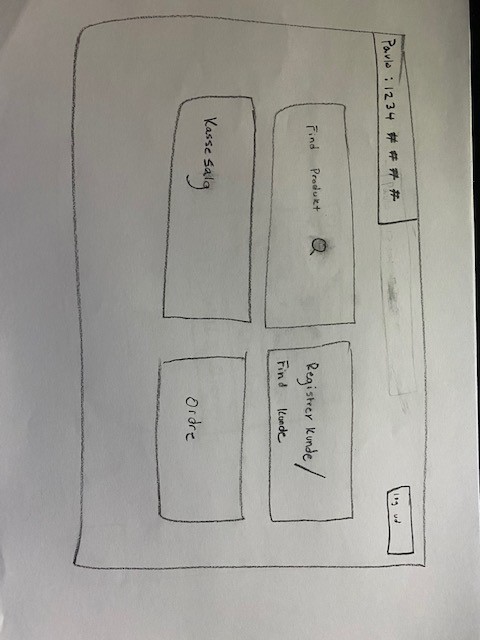
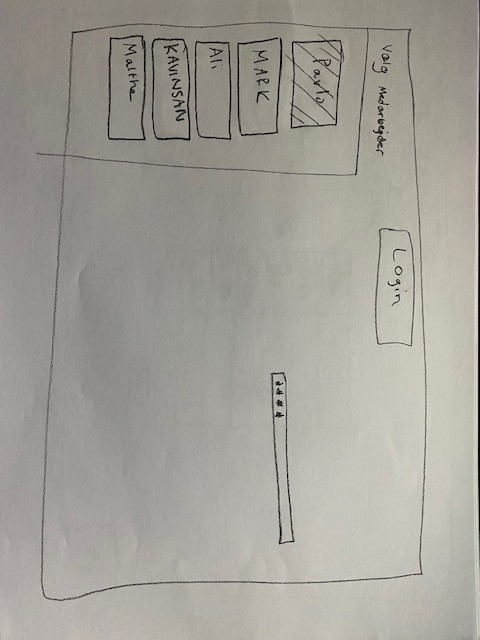
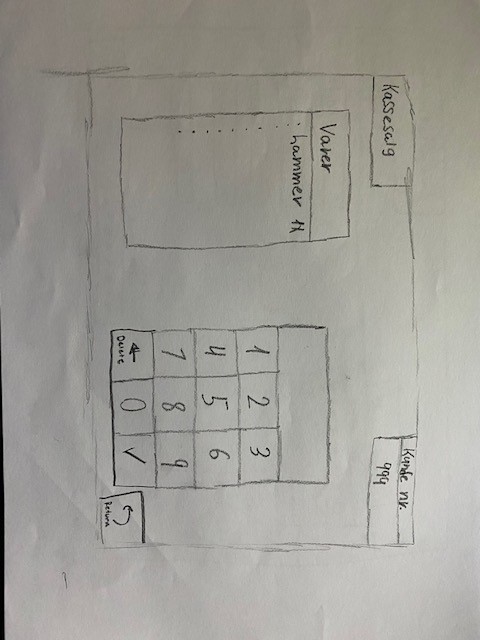
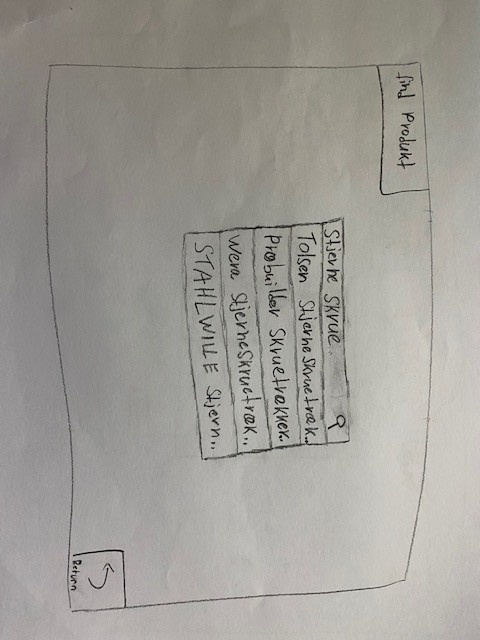


Workflow: registrering af produkt

I dette workflow er der 2 startpunkter, da man allerede kan have produktet eller skal til at registrere produktet. Vi starter med at registrere produktet i vores system, derefter katalogiser vi den, altså vi vælger hvor dens plads er på hylden.

## Mock-ups

Mockups viser gruppens første sketch af hvordan man forstiller sig systemets interface skal se ud, lavet ud fra det gruppen indtil videre ved om virksomhedens ønsker til systemet og de generelle krav. Gruppens mockups skal så bruges til en tænke højt test, som laves på en mulig bruger af systemet.



Vi fik lavet mockups af vores system, dem lavede vi sådan som vi tror det burde se ud hvis programmet så dagslys. Man starter med at logge ind, måden vi har valgt at gøre det på er: Du finder dit navn blandt medarbejder-navnene til venstre, og skriver din unikke kode. Herefter vælger man den opgave man skal til at gå i gang med, man kan blandt andet finde et bestemt produkt og dens information, registrere/finde kunde. Oprette et kasse-/ordresalg på stedet.

## Resultat af Tænk-høj test

Ud fra tænkehøjt testen har gruppen lært mere om hvordan systemet ønskes, testen fungerede egentlig godt, og i mange tilfælde var testpersonen tilfreds. Der er selvfølgelig plads til forbedringer, som gruppen også er gjort opmærksomme på, men de ønskede funktioner for programmeret er med.

## Systemvision

**Formål og afgrænsning**

Formålet med systemet er at gøre medarbejdernes arbejdsproces nemmere end den allerede er med det gamle system. Systemet skal kunne være i stand til at oprette nye kunder som brugere, tilføje varer til en ordre, administrer tilbud på varer, oprette udlånskontrakter på værktøj mv, ændre i kundernes information, slette kunder fra arkivet og afslutte ordrer.

**Problem Statement**

Både management teamet og de almindelige medarbejdere er trætte af deres nuværende system. Det der f.eks. mangler i det nuværende system, er salgsstatistikker over kunderne, produkterne og entreprenørerne. Man skal også kunne se statistikker over enhver medarbejder.

**Interessenter og Brugere**

De største interessenter er de almindelige medarbejdere og management teamet. Brugerne kommer til at være de almindelige medarbejdere, det er dog stadig muligt for management teamet at udføre de samme opgaver.

**Teknologi**

Vi regner med at systemet kommer til at køre på de lokale computere i byggemarkederne. Disse lokale computere skal have adgang til et fælles datasystem, som også bliver delt med XL-Byg, det giver firmaet muligheden for at fortælle kunderne om hvilke varer firmaet har på lager.

**Features**

* **Systemet skal kunne oprette ordrer.**
* **Systemet skal kunne oprette nye kunder som brugere.**
* **Systemet skal kunne give medarbejderne muligheden for at slette kundernes profiler efter behov.**
* **Systemet skal give medarbejderen mulighed for at ændre i kundernes oplysninger.**
* **Systemet skal være pålideligt.**
* **Systemet skal kunne give en oversigt af alle varer, om de er på lager, hvor mange der er eller hvornår de ca. kommer på lager.**

# Ikke-funktionelle krav:

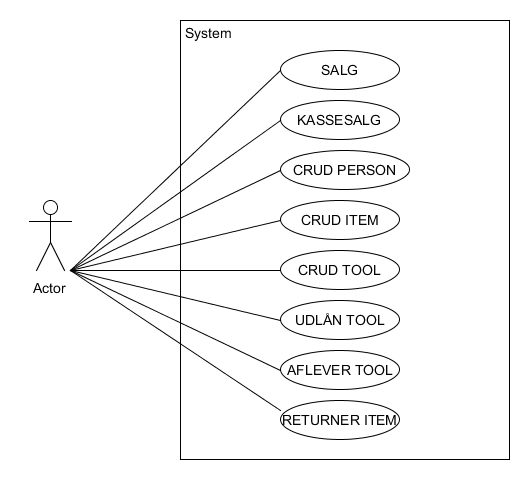
Før man begynder at arbejde på de funktionelle krav, altså use-cases, skal man have defineret nogle generelle ikke-funktionelle krav, som systemet skal kunne overholde. Altså ting man forventer af systemet, som ikke har noget med systemets mål eller funktion at gøre, men mere at gøre med vedligeholdelsen og andre interne ting.

## Punkopstillet liste

* Systemet skal være overskueligt og nemt finde rundt i.
* Systemet skal være effektivt og leve op til formålet.
* Systemet skal være brugervenligt.
* Systemet skal testes hyppigt, for at sikre et velkørende system.
* Systemet skal have en hurtig responstid, så arbejdspladsen ikke bliver forsinket.
* Systemet må ikke have en for lang opstartstid.

# Funktionelle krav:

## Use-case Diagram



## Use-case Briefs

Use-case briefs er en hurtig forklaring med nok detaljer til at forstå hvad aktøren gør.

***Use-case: Opret ordre (kassesalg)****:*  
En kunde (ikke oprettet) vil købe et eller flere produkter. I dette tilfælde behøver kunden ikke at være oprettet i systemet, man kan derfor oprette en ordre uden at tilknytte en kunde til salget.

***Use-case: Opret ordre (salg):***

En kunde vil købe et eller flere produkter. Hvis personen er registreret som kunde i systemet, kan ordren blive lavet med det samme, ellers skal personen registreres som kunde først.

***CRUD Kunde:***  
En kunde bliver oprettet i systemet med navn, telefon, email og adresse. Ellers kan kunden blive fjernet fra systemet.

***CRUD Item:***

CRUD-item handler om at tilføje nye produkter til systemet, og at kunne opdatere og slette dem.

***CRUD Tool***

CRUD tool skal bruges til at oprette nye værktøjer som virksomheden låner ud. Den kan også bruges til at redigerer eksisterende værktøjer eller slette dem.

***Use-case: Udlån værktøj:***

En kunde henvender sig til en salgsassistent for at låne et eller flere værktøj. Salgsassistenten opretter et udlån, giver værktøj og reserver dem til en given dato.

***Use-case: Aflever værktøj:***

En kunde aflever værktøjet, salgsassistenten registrer at værktøjet er afleveret. Værktøjet er klar til at blive udlånt igen.

***Use-case: Returner produkt:***   
Hvis en kunde har købt et produkt og ombestemmer sig, har han muligheden for at returnere produktet. Det skal så registreres i systemet at produktet er blev returneret.

## Use-case prioriteringstabel

Der laves en prioriteringstabel for alle use casene i projektet, tabellen bruges til at prioritere de vigtige opgaver programmet skal kunne.

|  |  |  |
| --- | --- | --- |
| Priority | Use Case Name | UP phase |
| 1 | Salg | Elaboration |
| 2 | Kassesalg | Elaboration |
| 3 | Udlån værktøj | Construction |
| 4 | CRUD Produkt | Construction |
| 5 | CRUD Kunde | Construction |
| 6 | CRUD Tool | Construction |
| 7 | Aflever værktøj | Construction |
| 8 | Returner produkt | Construction |

## Use-case fully dressed

Fully dressed er en detaljeret use-case hvor man beskriver, aktører, præbetingelse, postbetingelse, frekvens og flow of events.  
Aktører er personer som er involveret i use-casen, og det kan være alt fra en til mange, salgsassistent til kunde.  
Præbetingelse beskriver hvilke forhold skal være til stede inden use-case handlingen sker, det kan f.eks være at der skal være ting tilgængelse.  
Postbetingelse beskriver hvilke handlinger er sket og skal være opfyldt efter use-casen er gennemført, det kan f.eks. være at når et salg et færdig skal ordren være gemt.  
Frekvens beskriver hvor mange gange denne use-casen sker gennem dagen, ugen eller måneden, og det kan ske flere gange på samme dag, uge eller måned som f.eks. 10+ gange om dagen eller 2+ gange om måneden.  
Flow of events beskriver hvilke handlinger vores aktør gør og hvilke handlinger systemet vil give tilbage, når aktøreren har gjort en handling. Hver handling i flow of events har et sekvensnummer som viser hvornår handlingen sker.  
Alternative flows beskriver flows som kan ske hvis en handling i Flow of Events ikke opfylder eller mangler noget. Det kan f.eks. i Flow of Events henter vi et telefonnummer på en kunde, men kunden findes ikke. Så vi springer ind i et alternative flow hvor vi opretter kunden og derefter fortsætter på Flow of Events

### Fully dressed: Salg

|  |  |  |
| --- | --- | --- |
| Use case navn | **Salg** | |
| Aktører | Salgsassistent | |
| Præbetingelse | De ønskede produkter er tilgængelige og salgsassistenten er logget ind. | |
| Postbetingelse | Ordren er gemt og betalingsmåden er valgt | |
| Frekvens | 20+ gange jævnt fordelt over dagen | |
| Flow of events | **Aktør** | **System** |
| 1. Flowet starter med at en kunde vil købe nogle produkter. |  |
| 1. Assistenten opretter en ordre |  |
| 1. Assistenten tilføjer kunden til ordren. | 1. Systemet gemmer kunden under ordren |
| 1. Assistenten tilføjer de ønskede produkter | 1. Systemet tilføjer produkterne til ordren. |
| 1. Step 5–6 Gentages. |  |
|  | 1. Salgsassistenten sætter en leverings- eller afhentningsdato | 1. Systemet gemmer leveringsmetoden og datoen under ordren |
|  | 1. Ordren afsluttes og gemmes. | 1. Systemet gemmer ordren |
| Alternative flow | 3a: kunden er ikke registreret i systemet.  1: Salgsassistenten opretter i stedet en ny konto for kunden, og fortsætter som normalt. | |

### Fully dressed: Kasse salg

|  |  |  |
| --- | --- | --- |
| Use case navn | **Kasse salg** | |
| Aktører | Salgsassistent og kunde | |
| Præbetingelse | De ønskede varer er tilgængelige også i antal | |
| Postbetingelse | Ordren er gemt og betalingsmåden er valgt | |
| Frekvens | 20+ gange jævnt fordelt over dagen | |
| Flow of events | **Aktør** | **System** |
| 1. Use-casen starter med at en kunde/ salgsassistenten finder den/de varer de nu ønsker at købe, og tager dem til kassen. |  |
| 1. Salgsassistenten scanner varen ind. | 1. System tilføjer varen til salg(?) |
| 1. Step 2-3 gentages til der ikke er flere varer. |  |
| 1. Salgsassistenten afslutter ordren, og det totale beløb bliver betalt. | Systemet gemmer og afslutter orden |
| Alternative flow |  | |

### Fully dressed: Udlån værktøj

|  |  |  |
| --- | --- | --- |
| Use case navn | **Udlån værktøj** | |
| Aktører | Salgsassistent og Kunde | |
| Præbetingelse | At værktøjet er frigivet til udlån | |
| Postbetingelse | Et lån er blevet registeret og associeret med en kunde og produkter. | |
| Frekvens | 1-5 gange om ugen | |
| Flow of events | **Aktør** | **System** |
| 1. Use-casen starter med at en kunde henvender sig for at låne et eller flere værktøj. |  |
| 1. Salgsassistenten opretter et udlån. | 1. Systemet opretter et udlån. |
| 1. Salgsassistenten tilføjer værktøjer. | 1. Systemet tilføjer værktøj til udlån. |
| 1. Step 4-5 gentages til kunden er færdig. |  |
| 1. Salgsassistenten vil tilføje låne periode. | 1. Systemet opretter en periode for udlån. |
| 1. Salgsassistenten giver/sender kunden fakturaen. | 1. Systemet sender eller udskriver faktura. |
| Alternative flow | ?a: Der er ingen ledige værktøjer.  1: Systemet vil ikke kunne tilføje værktøj. | |

### Fully dressed: Returner item

|  |  |  |
| --- | --- | --- |
| Use case navn | **Returner item** | |
| Aktører | Salgsassistent og kunde | |
| Præbetingelse | Kunde skal have et produkt returneret, produktet er i systemet | |
| Postbetingelse | Produktet er returneret, produktet sættes ind i systemet igen, hvis det er god kvalitet | |
| Frekvens | 1-5 gange om ugen | |
| Flow of events | **Aktør** | **System** |
| 1. Use-casen starter med at en kunde henvender sig for at returnere et produkt ved kassesalg eller salg |  |
| 1. Salgsassistenten tjekker om produktet stadig er i systemet | 1. Systemet sender info tilbage omkring produktet |
| 1. Salgsassistenten skanner produktet ind, så produktet kan returneres | 1. Systemet bekræfter at produktet er skannet ind og er klar til at blive returneret |
| 1. Step 2-5 gentages til kunden er færdig. |  |
| 1. Salgsassistenten giver pengene tilbage for produktet/produkterne |  |
| Alternative flow | ?a: produktet er i dårlig stand og kan derfor ikke returneres. | |

### Fully dressed: Returnere værktøj

|  |  |  |
| --- | --- | --- |
| Use case navn | **Returner værktøj** | |
| Aktører | Salgsassistent og kunde | |
| Præbetingelse | Kunde skal aflevere et værktøj, værktøjet er i systemet, der er blevet betalt for værktøjet i den aftalte periode | |
| Postbetingelse | Værktøjet er afleveret, værktøjet sættes ind i systemet igen, hvis det er god kvalitet | |
| Frekvens | 1-5 gange om ugen | |
| Flow of events | **Aktør** | **System** |
| 1. Use-casen starter med at en kunde henvender sig for at aflevere et værktøj |  |
| 1. Salgsassistenten tjekker om værktøjet stadig er i systemet | 1. Systemet sender info tilbage omkring værktøjet |
| 1. Salgsassistenten skanner værktøjet ind, så det kan afleveres | 1. Systemet bekræfter at værktøjet er skannet ind og er klar til at blive afleveret |
| 1. Step 2-5 gentages til kunden er færdig. |  |
| 1. Salgsassistenten tager imod værktøjet og afslutter lånet |  |
| Alternative flow | ?a: værktøjet er i dårlig stand og kunden skal derfor betale  for en ny eller betale for reparationerne for værktøjet.   1. Værktøjet er leveret i dårlig stand (det kan ikke bruges igen). | |

# Informationskrav:

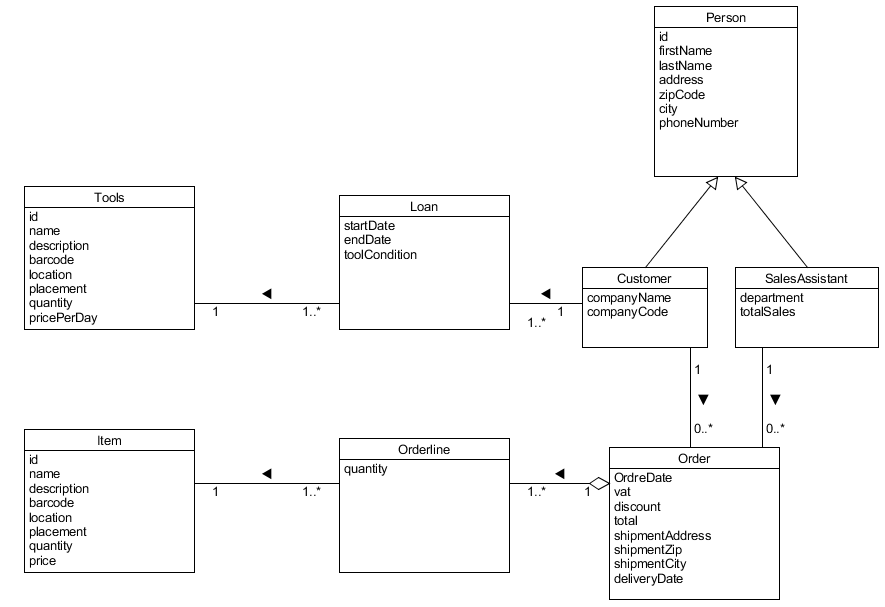
## Kandidat til klasse – vurderingstabel

Kandidat til klasse - tabellen bruges til at vurdering hvorvidt, hvor mange klasser der skal være i programmeret. De konceptuelle klasser kan findes på flere forskellige måder, bl.a. ved at gå igennem use case beskrivelserne og notere alle navneord og derefter vurderes der om de burde medtages som klasser.

|  |  |  |
| --- | --- | --- |
| Kandidat til klasse | Vurdering | Medtages (ja/nej) |
| Produkt | Holder information om produkt | ja |
| Kunde | Holder information om kunden | ja |
| Salgsassistent | Holder information omkring salgsassistenten | ja |
| Værktøj | Liste over værktøjer | ja |
| Managerteamet | Holder information om managers | nej |
| Tømmer center | Holder information om Tømmer-centeret | nej |
| DIY-center | Holder information om DIY-centeret | nej |
| Ordre | Holder en liste over ordren | ja |
| Levering | Information omkring leveringsadresse | ja |
| Placering | Lokaliteten af produkterne | nej |
| Bathroom |  | nej |
| Kitchen |  | nej |
| Levering | Information om leveringsadresse | ja |
| lån | Information om lånet | ja |

## Domænemodel

Domænemodellen er en visuel repræsentation af IT systemets informationskrav for problemområdet. Det primære i en domænemodel er identificeringen af problemområdets klasser og deres attributer samt associationer mellem klasserne. Klasserne til domænemodellen, findes ud fra en vurderingstabel (kandidat til klasse tabel), som ses ovenover.



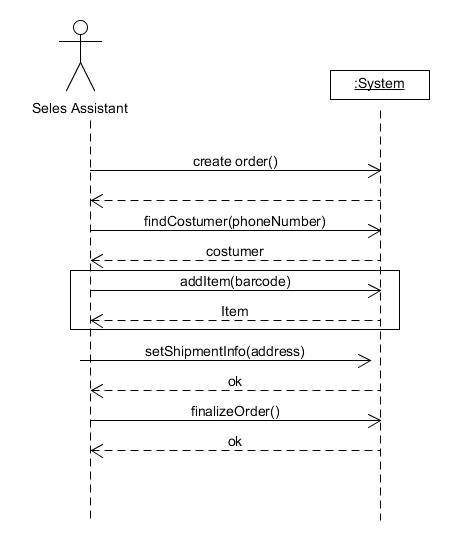
Domænemodellen indeholder 6 klasser, en super klasse og en sub klasse. En customer klasse, en salesassistent klasse, en order klasse, en loan klasse, en tools klasse, en item klasse og en orderline sub klasse. Der er også en superklasse, hvor customer og salesassistent arver fra. Mellem de forskellige klasser er der associeringer, og fra customer klassen og salesassistent klassen er der to tomme pile (inheritence) op til personklassen, som indikerer at Person klassen er en super klasse. Mellem Order klassen og orderline er der en aggregering, som indikerer at der kan ikke orderline ikke kan eksistere uden order klassen. Mellem de andre klasser, kan vi se at der er associeringer.

# Analyse og design af use-cases:

## Systemsekvensdiagram (SSD)

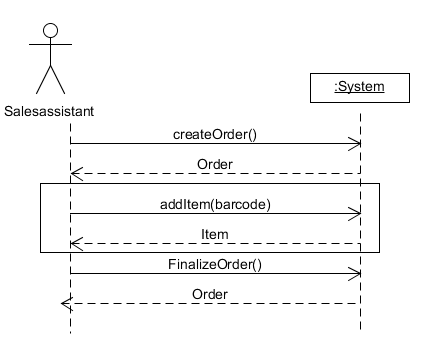
Man bruger systemsekvensdiagrammet til at vise hvordan aktøren interagerer med systemet, og hvordan systemet svarer tilbage på aktørernes handling. Dette diagram er godt til at give os overblik over hovedprocessen. Pilen der peger mod højre, er en handling aktøren foretager. Den stiplede pil der peger imod venstre, viser systemets svar, i dette tilfælde er det hvad systemet returnerer.

### Use-case Salg

(Salg)

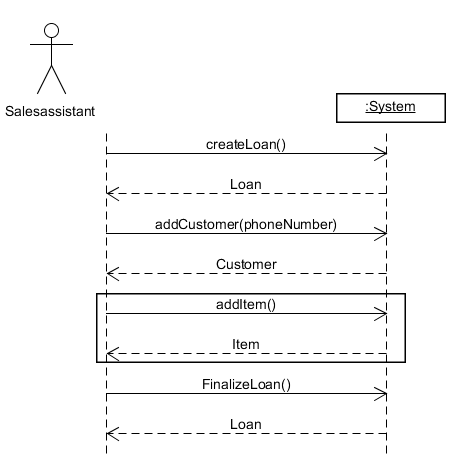
Diagrammet er for et salg, viser hvordan det fungerer når kunden skal kobles til salget. Det bruges enten når kunden vil have sendt en faktura i slutningen af måneden i stedet for at betale kontant, eller hvis kunden vil have leveret sin ordrer hjem, eller til en bestemt adresse, hvis nu ordren er for stor. Kunden kan selvfølgelig også altid vælge at betale med det samme, i stedet for at få sendt en faktura, og kunden kan også vælge en dato som varerne skal afhentes på, eller om han/hun vil have dem med det samme.

### Use-case kassesalg

(Kassesalg)

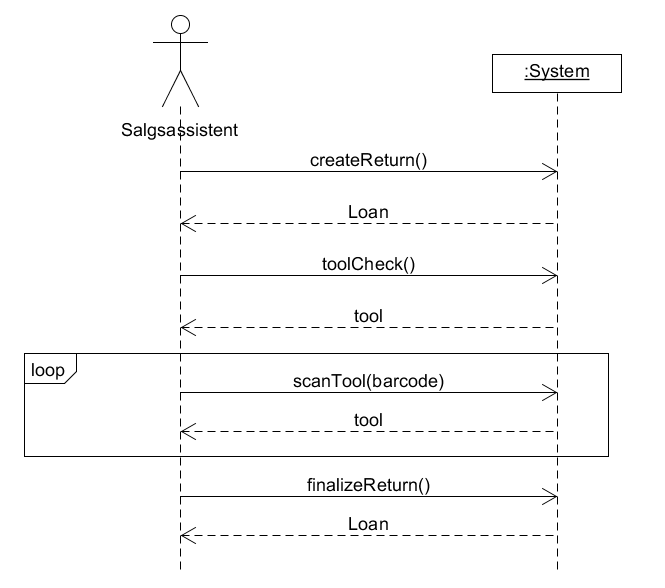
På kassesalg diagrammet kan man se interaktionen imellem en salgsassistent og systemet. Vores mål i denne instans er at oprette en ordre, og så sammenkoble det til en kunde (customer). Vi navngiver alle objekterne det samme som de metoder vi har i koden, samt parametre. Dette gør det nemmere for os som programmører, da vi altid kan forstå hvad det drejer sig om og hvilken metode det er vi har med at gøre. Hvis kunden er en kontantkunde og ikke har en brugerprofil eller gerne vil have en, skriver man 999 i telefonfeltet, så de bliver registreret som kontantkunde.

### Use-case Udlån værktøj

(Udlån af værktøj)

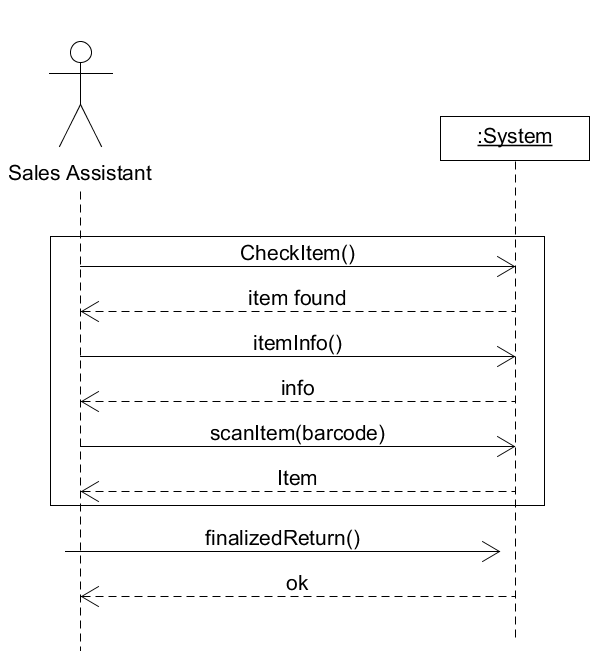
På udlån af værktøj diagrammet kan man se at interaktionen imellem en salgsassistent og systemet. Vores mål i denne instans er at oprette et lån, og så sammenkoble det til en kunde (customer). Vi navngiver alle objekterne det samme som de metoder vi har i koden, samt parametre. Dette gør det nemmere for os som programmører, da vi altid kan forstå hvad det drejer sig om og hvilken metode det er vi har med at gøre.

### Use-case aflever værktøj

(Aflevere værktøj)

Diagrammet er for casen “Aflevere værktøj”, som viser hvordan det fungerer når kunden skal aflevere værktøj tilbage efter et lån. Først opretter man en “return”, hvor systemet finder lånet og viser den frem til aktøren. Derefter fortæller man systemet om værktøjet er i okay stand. Det svarer systemet på, for at vise at den har registreret dette. Herefter kan man scanne værktøjet, og her vil systemet så svare med enten “barkoden blev fundet” eller “barkoden blev ikke fundet”. Til sidst afslutter man retur processen.

### Use-case returnere produkt

(returnere produkt)

På returner produktdiagrammet i denne instans er målet at salgsassistenten kan bruge systemet til at returnere et produkt en kunde alligevel ikke vil have. Den første metode bruger man for at tjekke om produktet st6adig findes i systemet. Systemet svarer enten om den er der, eller om den ikke er der. Derefter beder man systemet om produktinformation, det får man så retur af systemet. Herefter scanner man produktets stregkode og får produktet tilføjet på returneringslisten. Til sidst afslutter man ordren.

## Operationskontrakt

Operationskontrakt er en aktørs handling i systemet, hvor der sker en ændring i systemet. Operationskontrakter er lavet ud fra handlinger i en SSD eller domænemodel. Der er 2 forhold der definer de nødvendige handlinger for at køre operationen. Først er der en precondition eller forudsætning som beskriver hvilke forhold, skal være tilstede for at køre denne operation. Postcondition beskriver hvilke handlinger som er blevet udført efter operationen er kørt.

|  |
| --- |
| Operationskontrakt: scanTool(barcode) |
| Use case: Aflever værktøj  Preconditions: Et lån objekt findes i systemet og værktøjet eksisterer.  Postcondition:   * Tool t er lavet. * Tool t bliver tilføjet til returneringsordren. |
| Operationskontrakt: addItem(barcode) |
| Use case: Salg  Preconditions: En ordre instans-objekt er lavet og item’et eksisterer.  Postcondition:   * Produktet p er lavet. * Produktet p bliver tilføjet til ordren. |

|  |
| --- |
| Operationskontrakt: findCostumer(phoneNumber) |
| Use case: Salg  Preconditions: En ordre instans-objekt er lavet og kunden eksisterer.  Postcondition:   * Kunden er sat til c. * C er tilføjet til ordren. |

|  |
| --- |
| Operationskontrakt: finalizeOrder() |
| Use case: Kasse salg  Preconditions: En ordre instans er lavet.  Postcondition:   * En ordre er lavet. * En person er associeret til ordren. * En eller flere item objekter er tilføjet til ordren. |

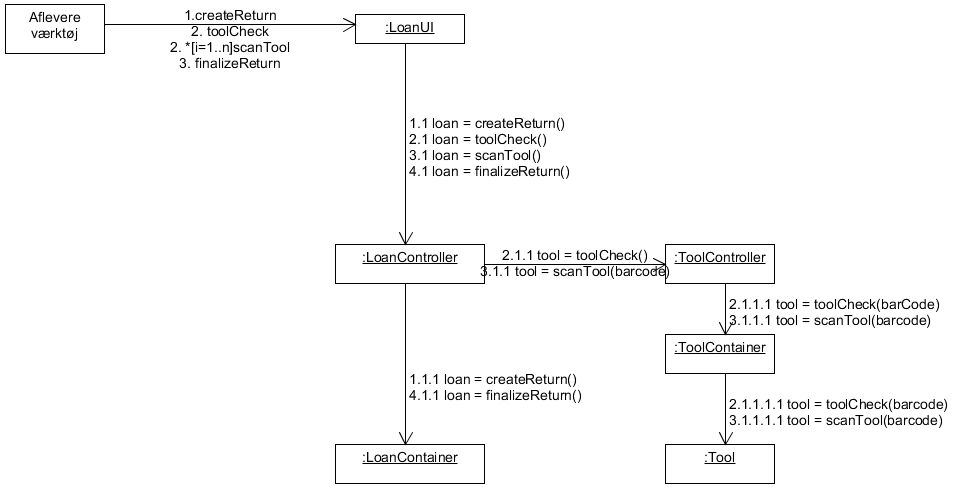
|  |
| --- |
| Operationskontrakt: finalizeLoan() |
| Use case: Udlån værktøj  Preconditions: En lån instans er lavet.  Postcondition:   * En dato og periode er associeret til lånet. * En lån er lavet. * En person er associeret med lånet. * En eller flere værktøjer er tilføjet til lånet. |

|  |
| --- |
| Operationskontrakt: finalizeReturn() |
| Use case: Aflevere værktøj  Preconditions: Et lån med værktøj er lavet.  Postcondition:   * Et eller flere værktøj er blevet returneret tilbage i vores systemet. * Lånet er blevet fjernet fra vores system. |

# Design:

## Interaktionsdiagram

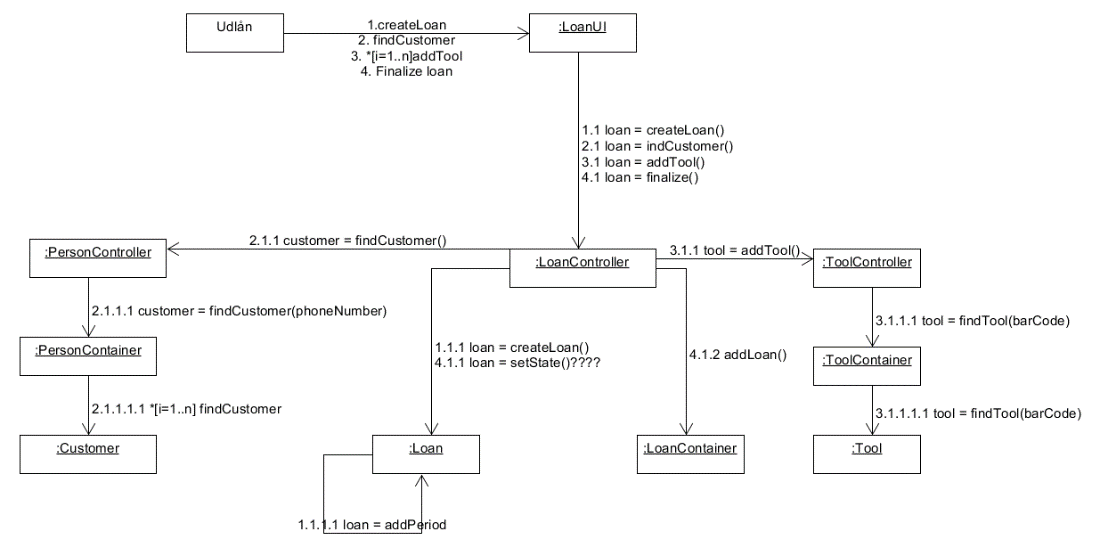
Formålet med et interaktionsdiagram er at visualisere handleringerne der sker i systemet, typisk for en use-case. Diagrammet vil vise vores objekt klasser samtidig med at vise hvilke handlinger eller beskeder der sker mellem de forskellige klasser og hvilke handlinger sker først med sekvensnummeret.



Interaktionsdiagram: Aflevere værktøj.

Her i diagrammet kan vi se hvilke objekter og hvilke metoder bliver brugt for at ”Aflevere værktøj”. Det bliver taget brug af LoanUI, LoanController, LoanContainer, ToolController, ToolContainer og Tool objekter.

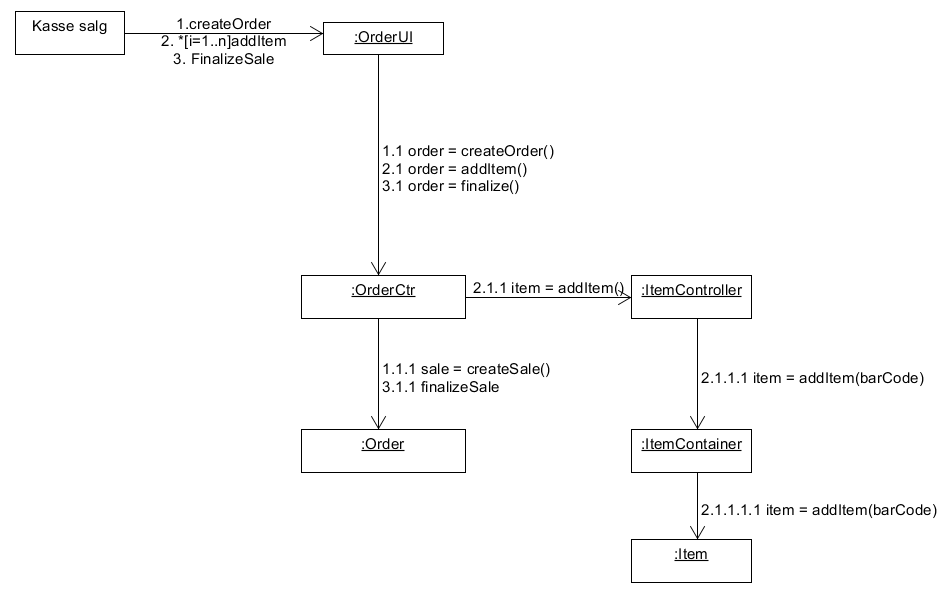
Der er 4 handlinger i dette interaktionsdiagram: CreateReturn, LoanCheck, ScanTool og finalizeReturn, som hvert beskriver deres handlinger i en sekvens.



Interaktionsdiagram: Udlån.

Her i diagrammet kan vi se hvilke objekter og hvilke metoder bliver brugt for at ”Udlån”. I dette interaktionsdiagram bliver der brugt: PersonController, PersonContainer, Customer, LoanUI, LoanController, LoanContainer, Loan, ToolController, ToolContainer og Tool, som hvert har deres egne metoder for at kommunikere med det næste objekt.

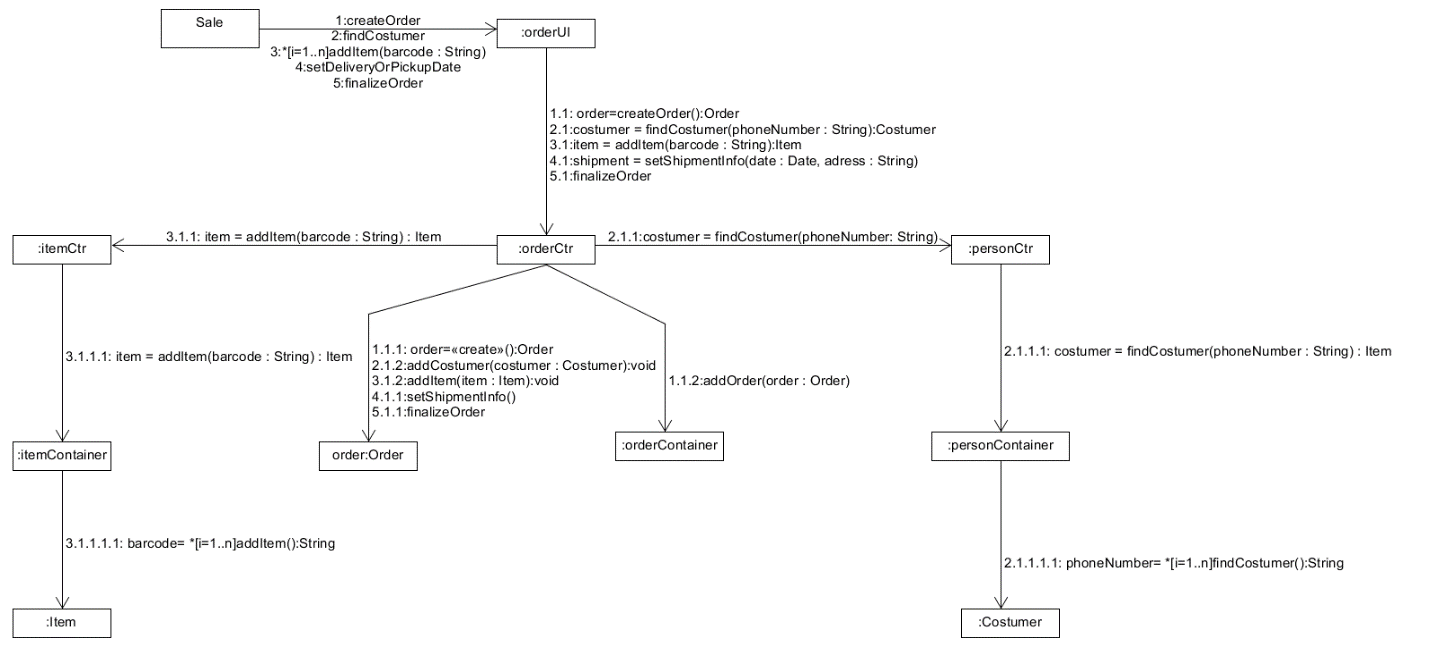
Der er 4 handlinger i dette interaktionsdiagram: CreateLoan, findCustomer, addTool og FinalizeLoan, som hvert beskriver deres handlinger i en sekvens.



Interaktionsdiagram: Kassesalg.

Her i diagrammet kan vi se hvilke objekter og hvilke metoder bliver brugt for at ”Kassesalg”. Det bliver taget brug af OrderUI, OrderController, Order, ItemController, ItemContainer og Item.

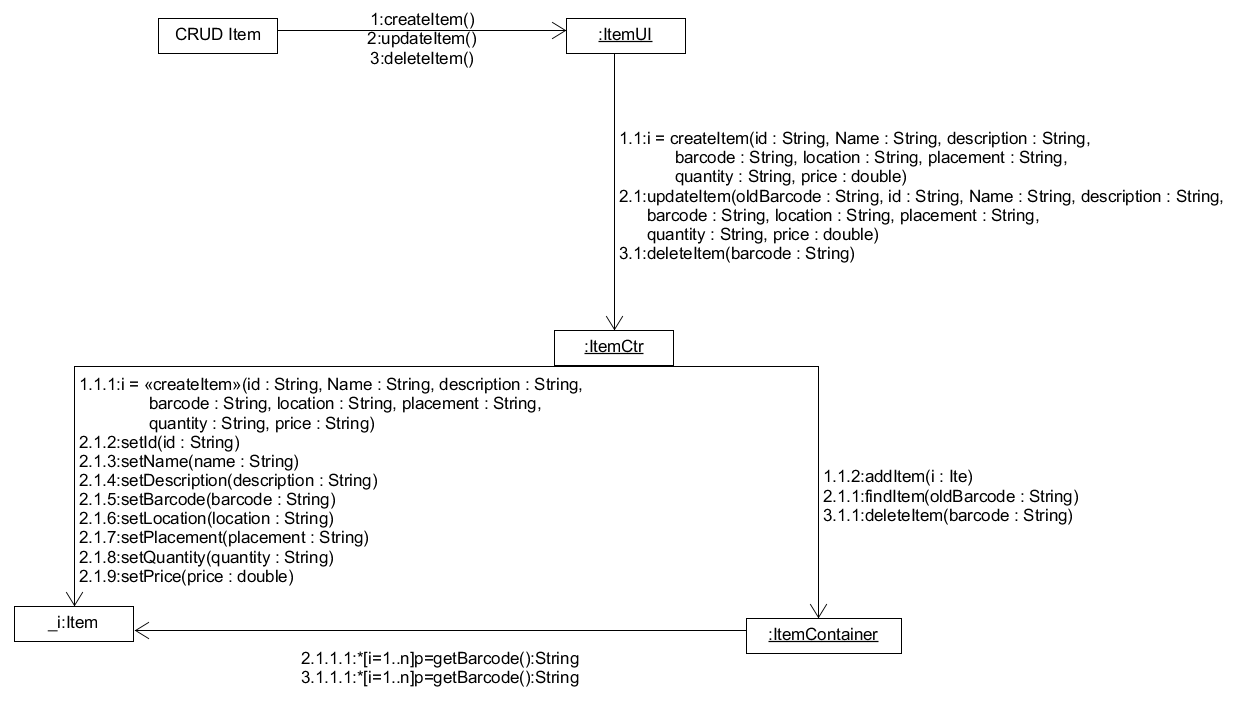
Der er kun 3 handlinger i dette interaktionsdiagram: createOrder, addItem og finalizeSale, som hvert beskriver deres handlinger i en sekvens.



Interaktionsdiagram: Salg.

Her i diagrammet kan vi se hvilke objekter og hvilke metoder bliver brugt for at ”Salg”. Her bliver der tager brug af OrderUI, OrderController, OrderContainer, Order, ItemController, ItemContainer, Item, PersonController, PersonContainer og Customer.

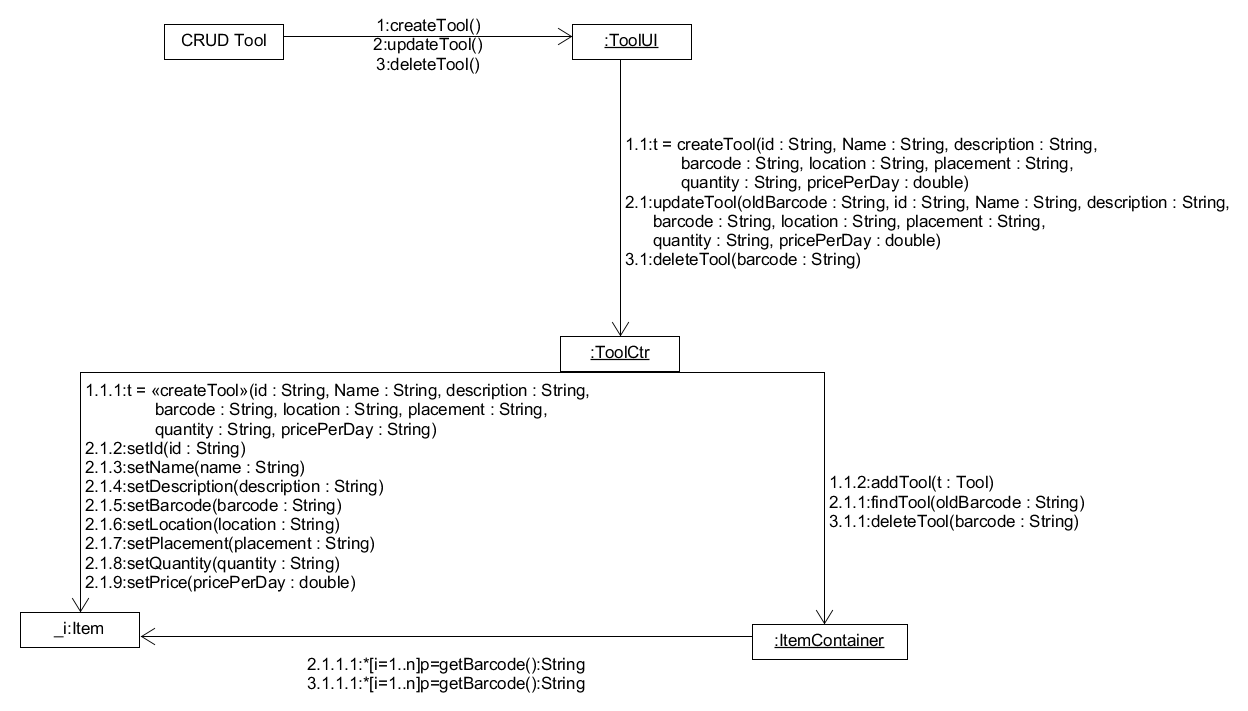
Der er 5 handlinger i dette interaktionsdiagram: createOrder, findCustomer, addItem, setDeliveryOrPickupDate og finalizeOrder, som hvert beskriver deres handlinger i en sekvens.



Interaktionsdiagram: CRUD Item.

Her i diagrammet kan vi se hvilke objekter og hvilke metoder bliver brugt for at ”CRUD Item”. I dette interaktionsdiagram er objekterne ItemUI, ItemController, ItemContainer og Item.

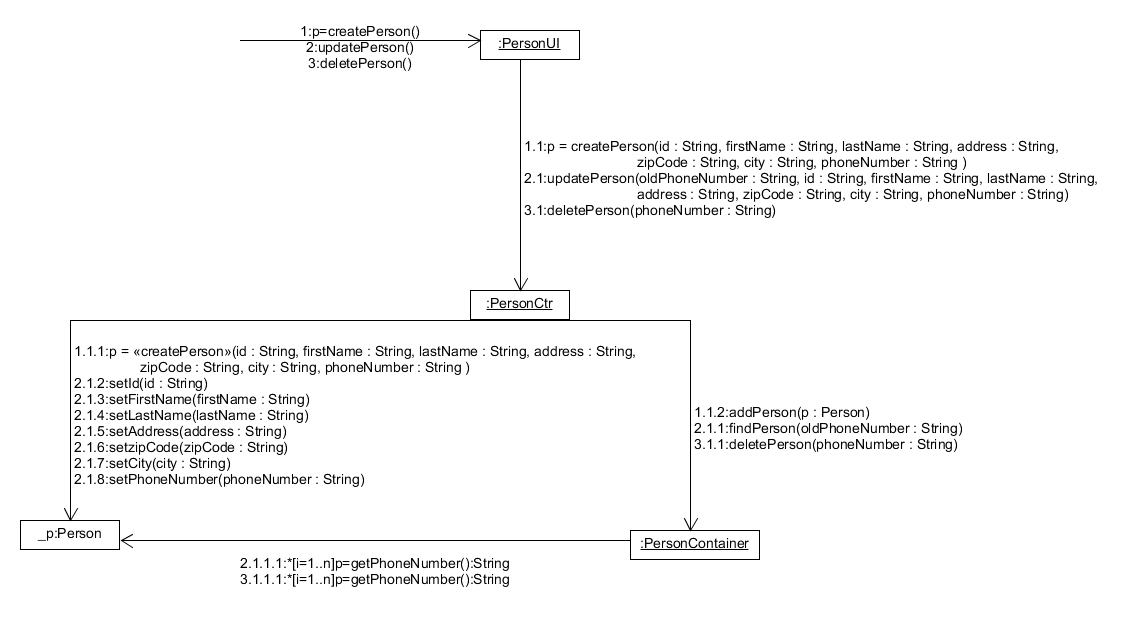
Der er kun 3 handlinger i dette interaktionsdiagram: createItem, updateItem og deleteItem, som hvert beskriver deres handlinger i en sekvens.



Interaktionsdiagram: CRUD tool.

Her i diagrammet kan vi se hvilke klasser og hvilke metoder bliver brugt for at ”CRUD tool”. I detter interaktionsdiagram er der ToolUI, ToolController, ItemContainer og Item.

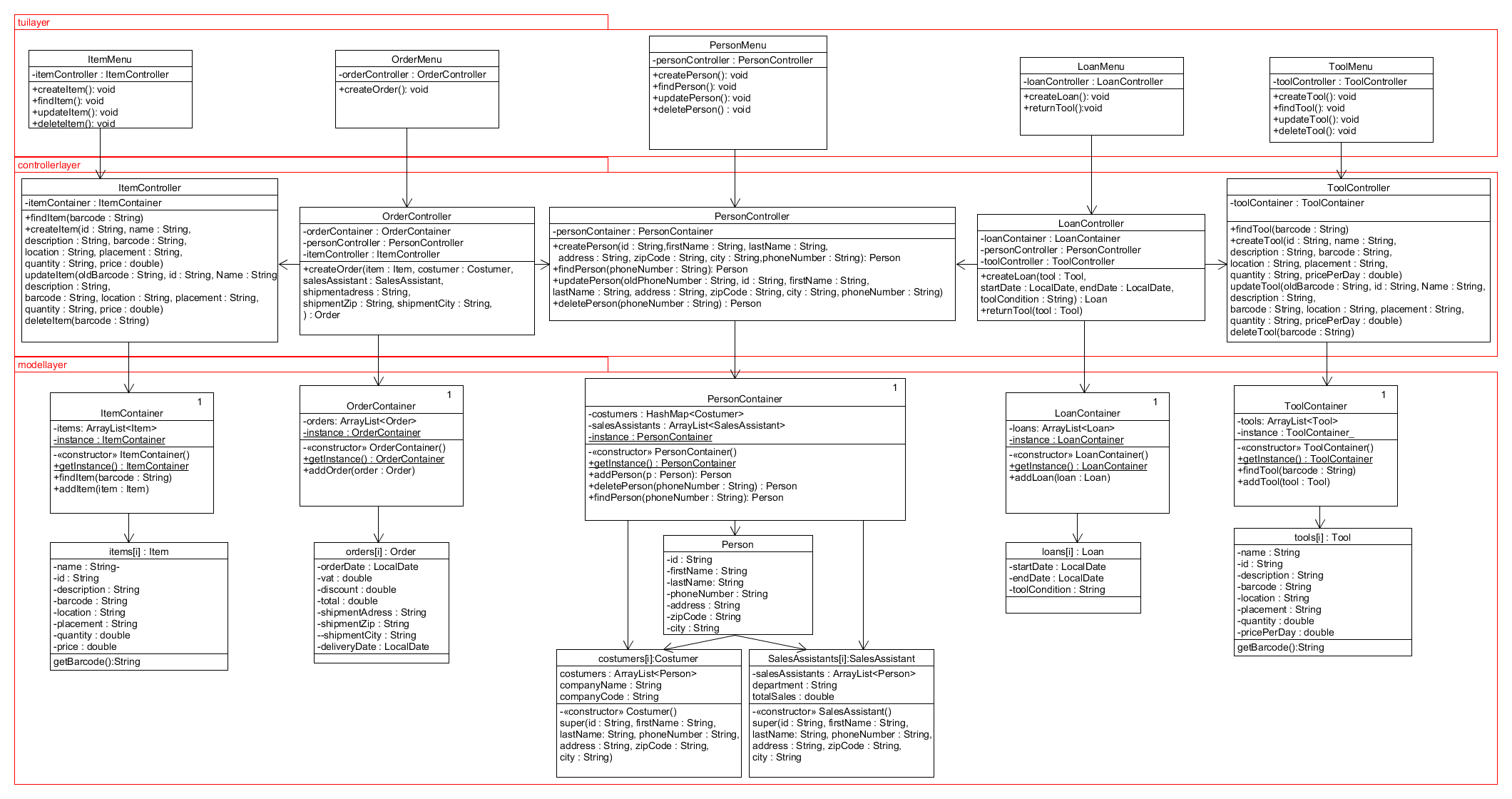
Der er kun 3 handlinger i dette interaktionsdiagram: createTool, updateTool og deleteTool, som hvert beskriver deres handlinger i en sekvens.



Interaktionsdiagram: CRUD Person

### Design klasse diagram

Design klassediagrammet kan laves ud fra kommunikationsdiagrammerne, og er det sidste diagram før man begynder at kode programmet. Diagrammet giver et overblik over metoder og klassernes indhold, delt ud i et 3 lags system, som består af et modellag, et kontrollag og et UI-lag.



# Programmet

I dette afsnit, vil vi komme ind på hvordan vi har løst vores case i form af et program.

Et billede, der indeholder tekst

Automatisk genereret beskrivelse

Når man i menuen har valgt at oprette en ny ordre, bliver metoden createOrder() kaldt. Heri bliver der først og fremmest erklæret nogle variabler, som knytter sig til den ordre der skal oprettes. F.eks. variablerne der starter med ”ship” som senere i metoden bliver initialiseret og tildelt en værdi fra brugerens input i konsollen. Meget essentielt for denne metode er while-løkken, som indeholder koden der tilføjer ordrelinjer til ordren, ud fra brugerens input. While-løkken kører indtil brugeren har indtastet en tom tekststreng. I linje 53 på overstående billede, har vi brugt ternære operatoren, til at styre hvorvidt konsollen skal printe ”produkt” i flertal eller ental. Ternære operatoren returnerer enten den ene eller den anden værdi, afhængigt af om udtrykket på venstre side er sandt eller falsk. Herefter bliver createOrder metoden kaldt, som er en metode i OrderControlleren der opretter ordren i systemet og dermed færdiggøre opgaven.[[3]](#footnote-4)

# Gruppeevaluering

Vi har som gruppe haft stor succes med at færdiggøre projektet. Til tider har der været problemer med at møde op til tiden, det kan dog skyldes bus/vejr. Alle i gruppen har vist lige så meget engagement for projektet, hvilket har gjort det nemt for hele gruppen at lave den. Alle fik tildelt en ting at skrive i rapporten, men diagrammerne var noget vi lavede i fællesskab. Alle var med til at skrive noget kode, skabe diagrammer og skrive et stykke rapport.

# Konklusion

Gruppen har gennem it forundersøgelsen, kravbeskrivelsen og analysen af systemets metoder og funktioner, kommet frem til et godt bygget system, der løser de fleste problemer som Vestbjerg har haft, og opfylder de mangler som de har oplevet med deres nuværende system. Der er dog nogle af funktionerne som gruppen ikke har fået løst endnu, som f.eks. den funktion der gør det muligt for Vestbjerg at holde styr på deres medarbejders salg. Gruppen har arbejdet med alle funktionerne, og lavet forarbejde på de fleste, men nogle af dem er dog ikke blevet implementeret i systemet endnu. Men alt i alt så har gruppen lavet et velfungerende system, der som sagt opfylder de fleste af Vestbjergs ønsker. Så efter at have udarbejdet et nyt system for virksomheden, kan vi konkludere at et nyt system tager længere tid, men ved hjælp at de forskellige diagrammer kunne vi bygge en bedre fremgangsmåde i vores arbejde. Ved at visualisere vores system i diagrammerne og de metoder vi har lært, blev det herefter nemmere at skrive koden og implementere det.

# Referenceliste

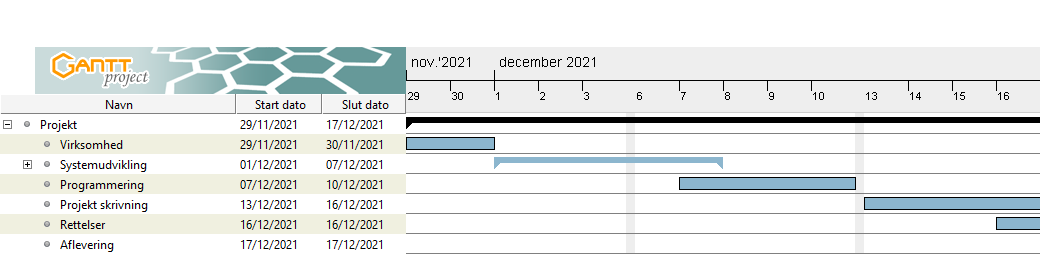
“Code Conventions for the Java Programming Language”. Set 14. december 2021. https://www.oracle.com/java/technologies/javase/codeconventions-introduction.html.

Srivastava, Shubhra. “Ternary Operator In Java | Baeldung”, 1. oktober 2018. https://www.baeldung.com/java-ternary-operator.

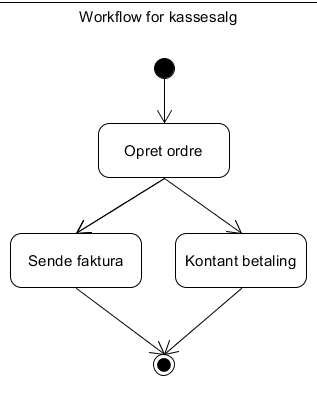
s.r.o, BarD Software. “GanttProject: Free Project Management Tool for Windows, MacOS and Linux”. GanttProject. Set 16. december 2021. https://www.ganttproject.biz.

# Bilag

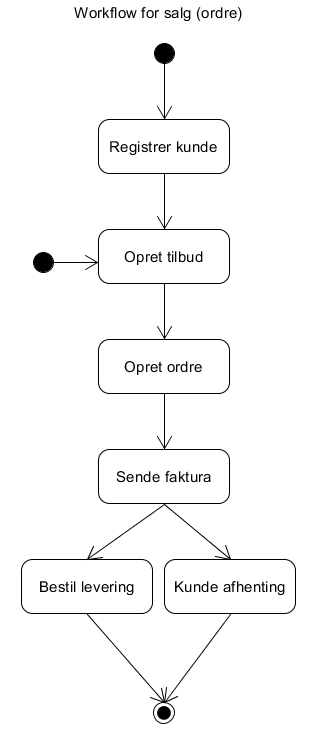
## Bilag 1:



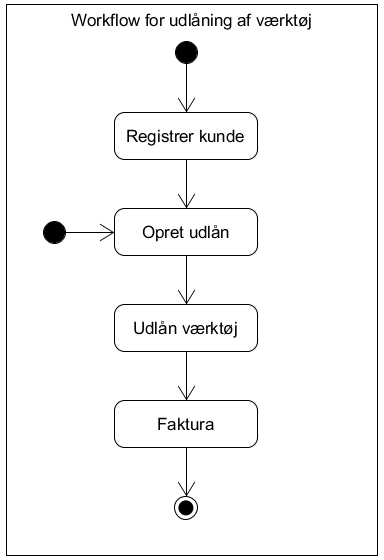
## Bilag 2:



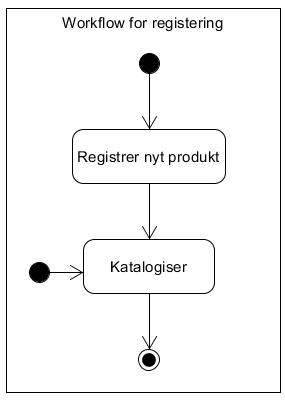
## Bilag 3:



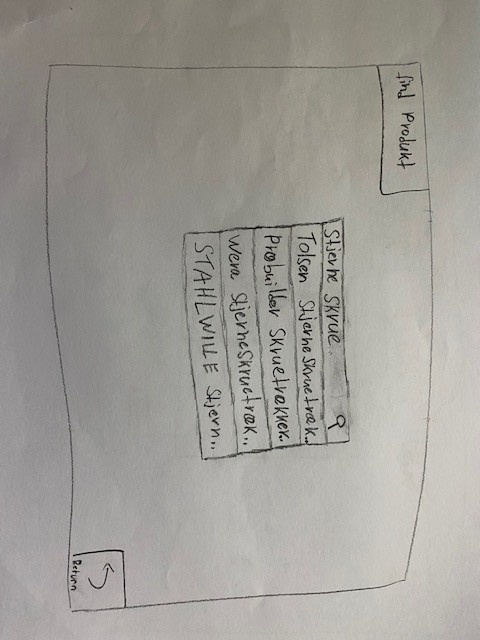
## Bilag 4:



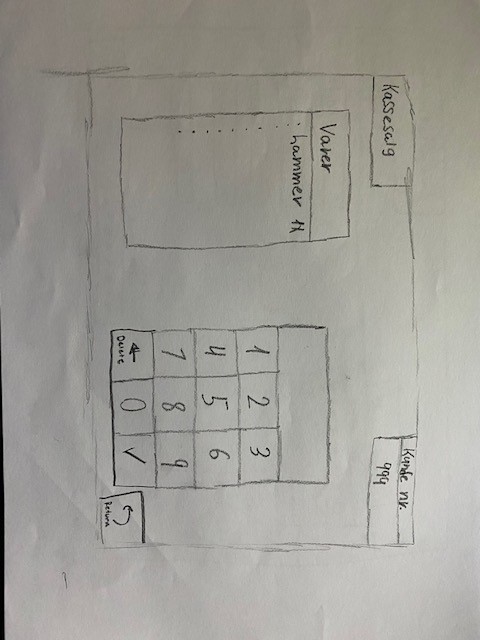
## Bilag 5:



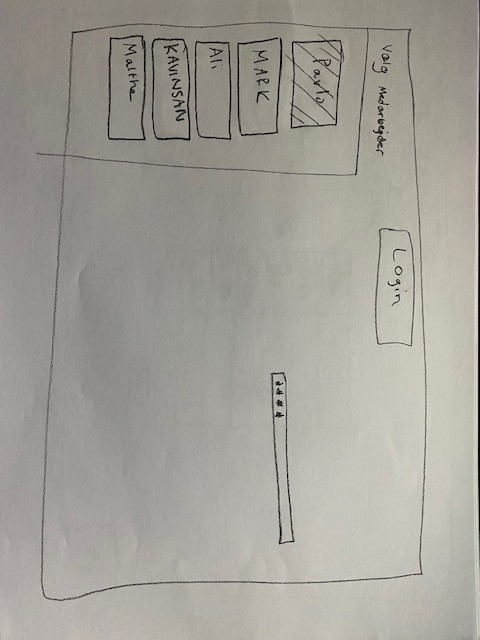
## Bilag 6:



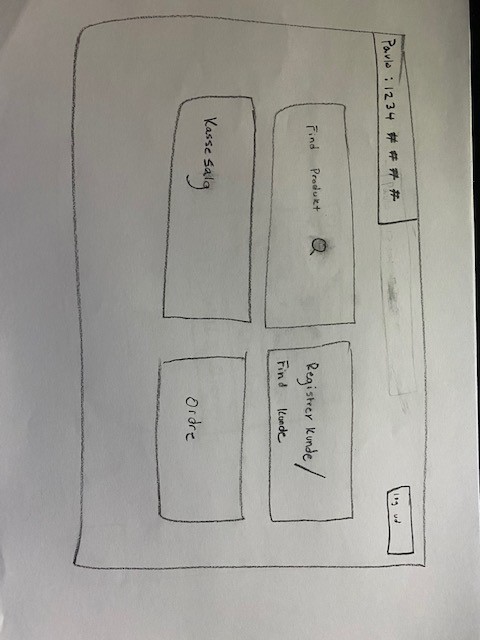
## Bilag 7:



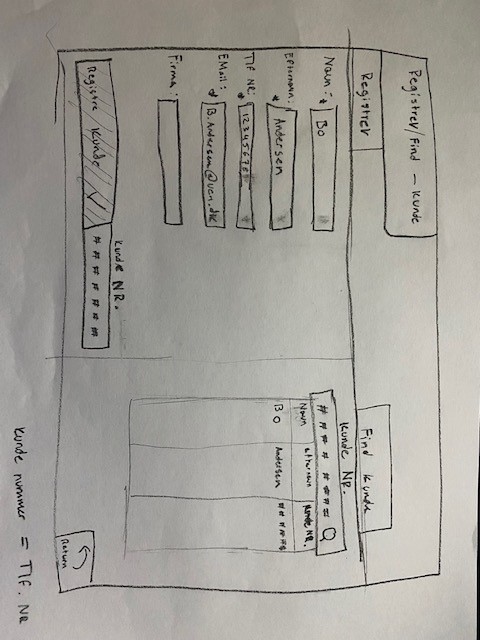
## Bilag 8:



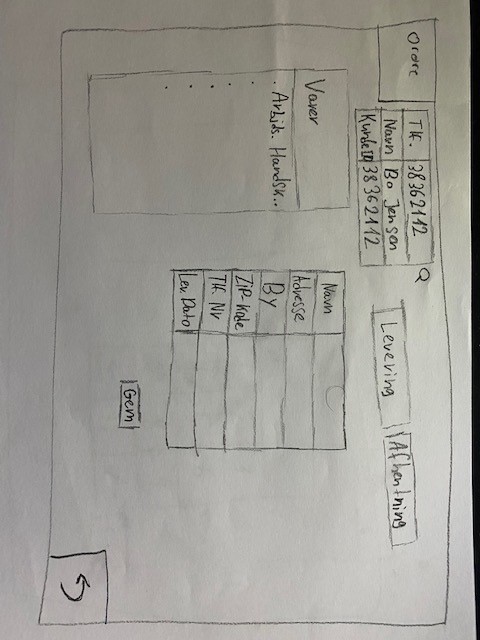
## Bilag 9:



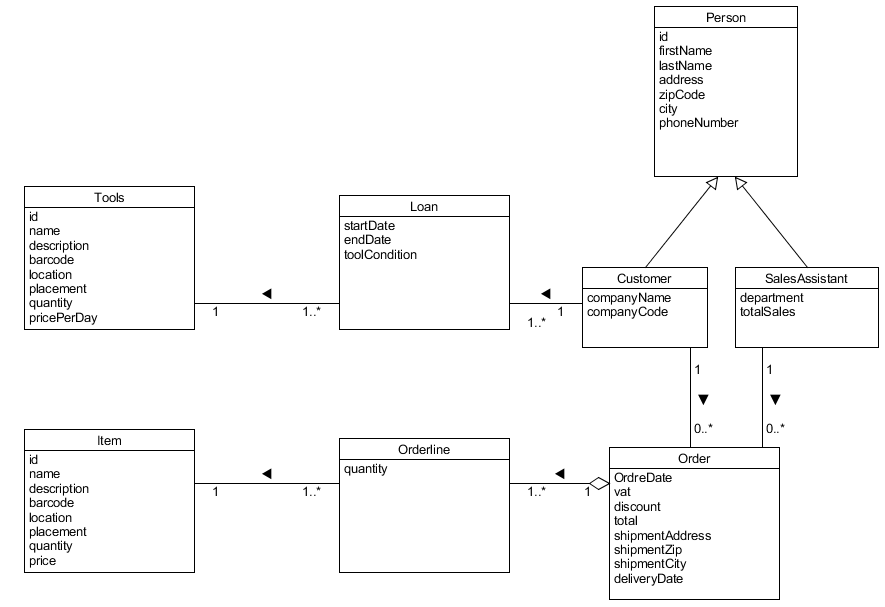
## Bilag 10:



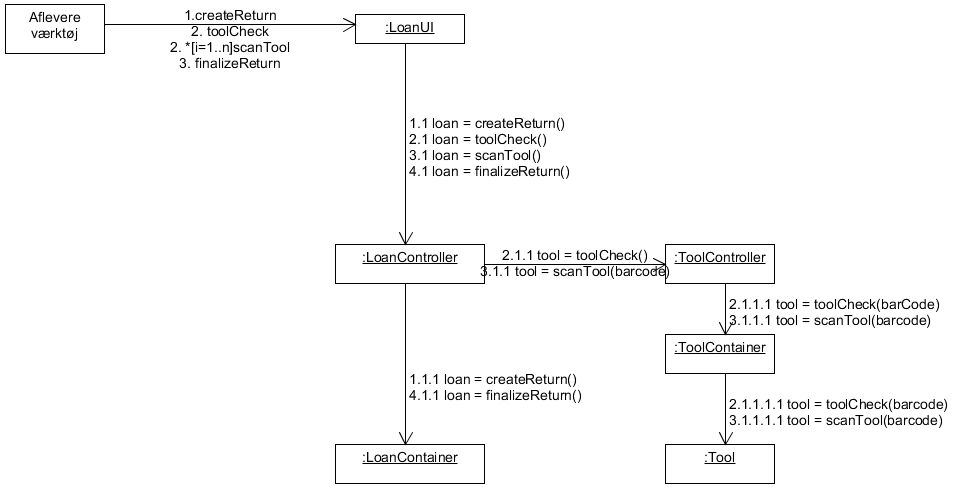
## Bilag 11:



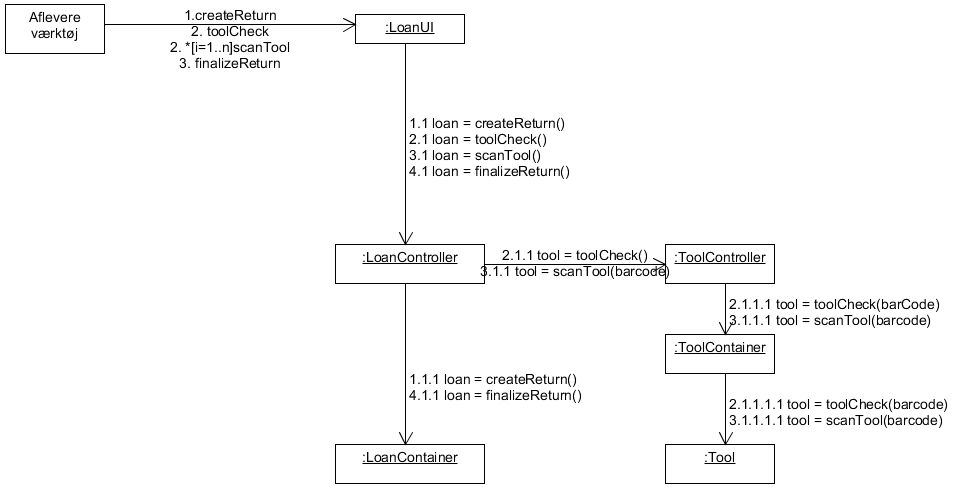
## Bilag 12:



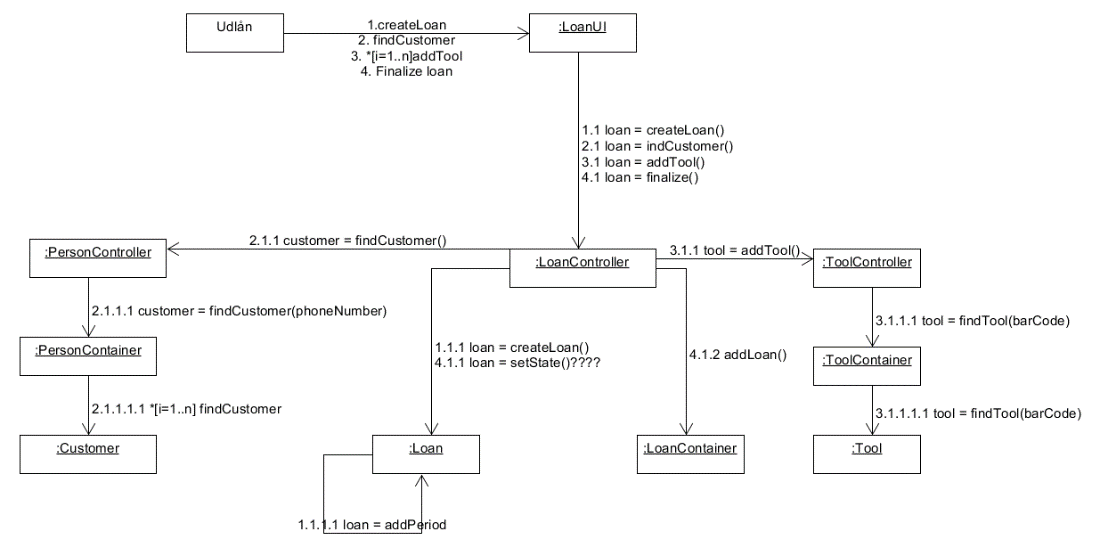
## Bilag 13:



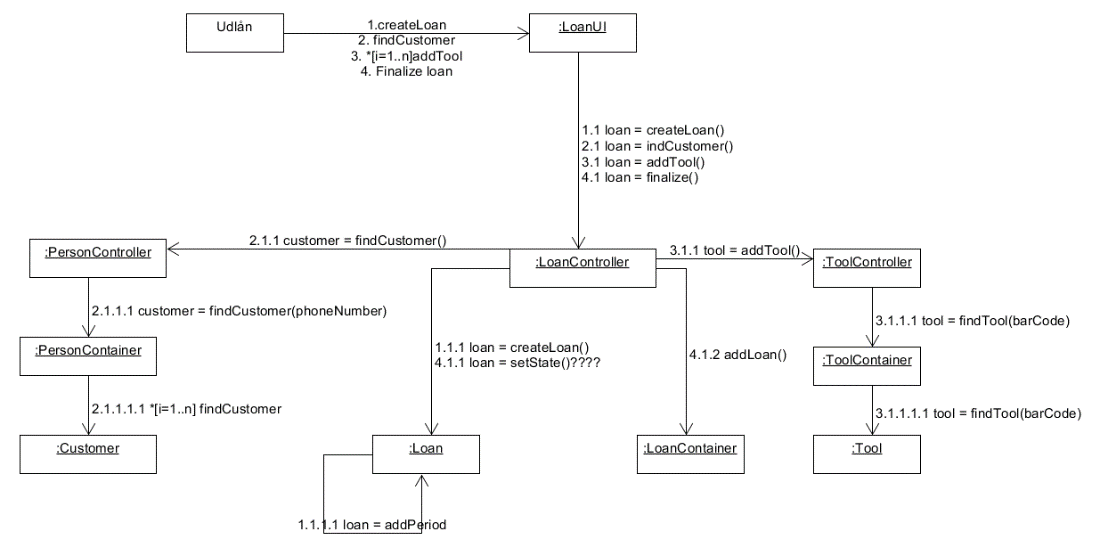
## Bilag 14:



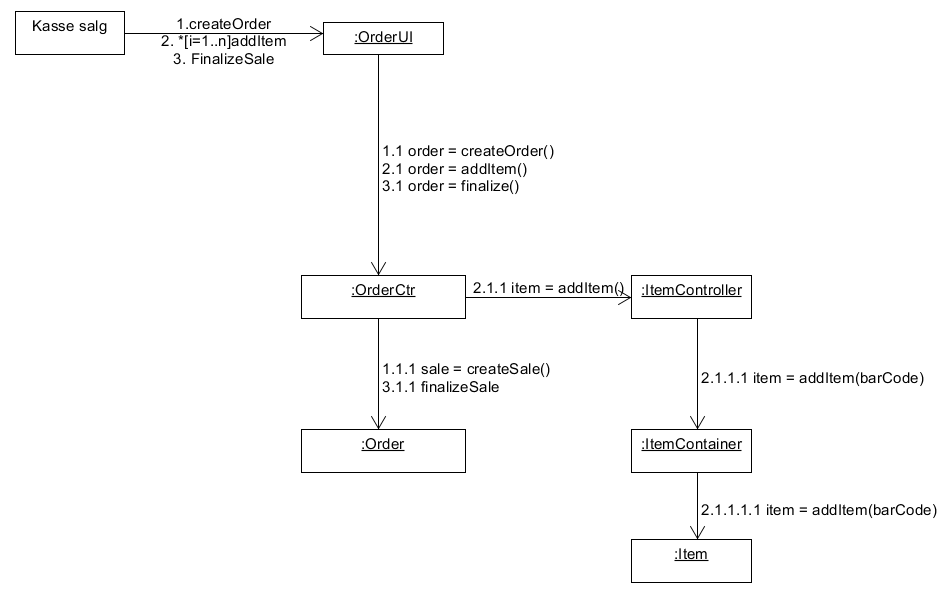
## Bilag 15:



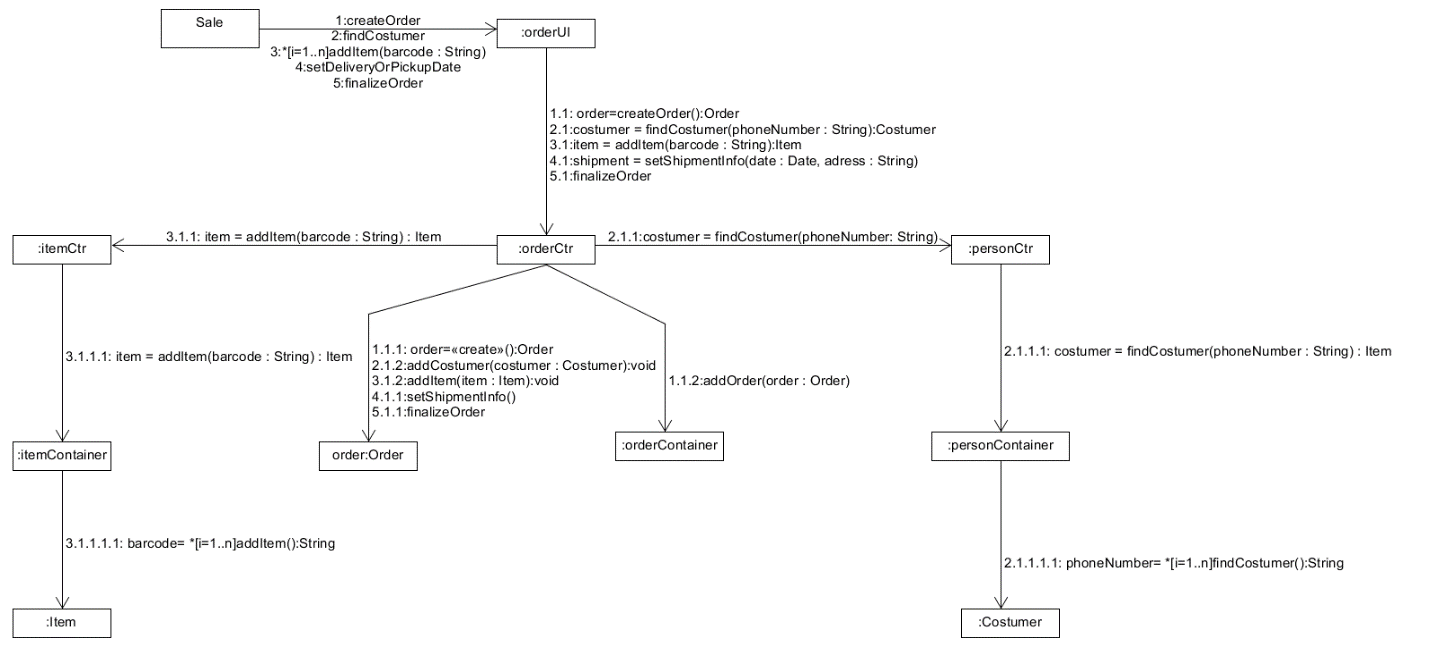
## Bilag 16:



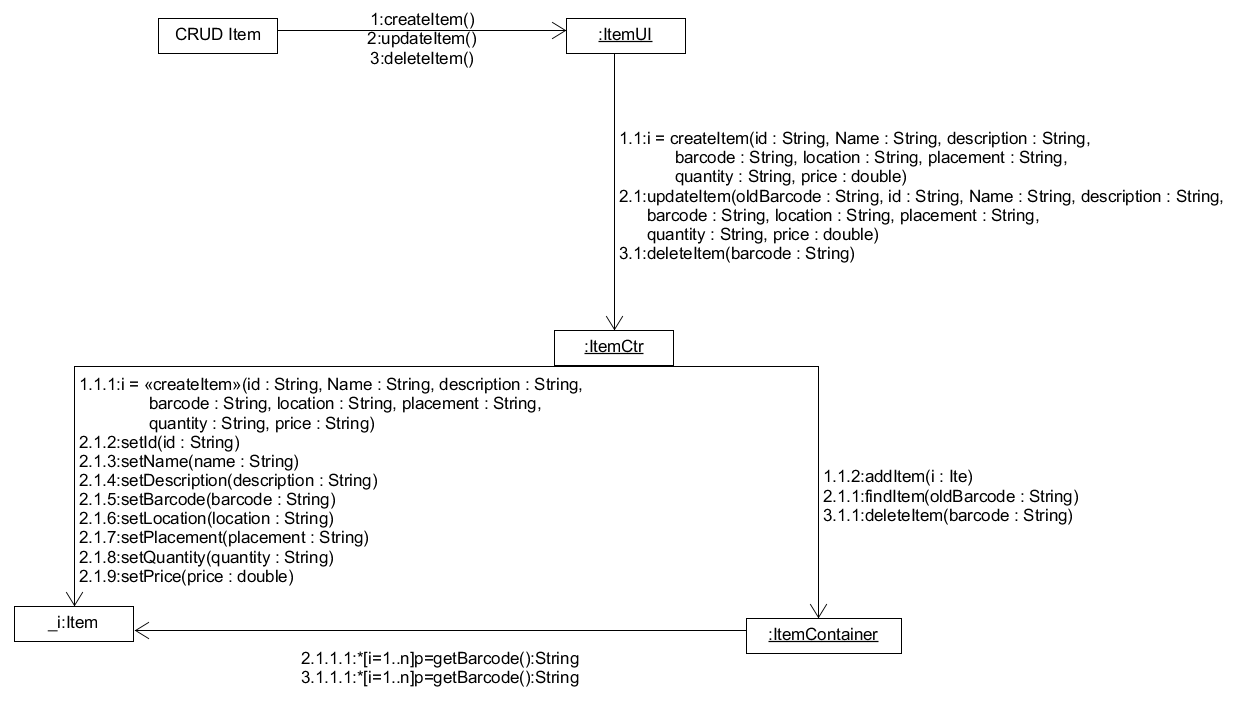
## Bilag 17:



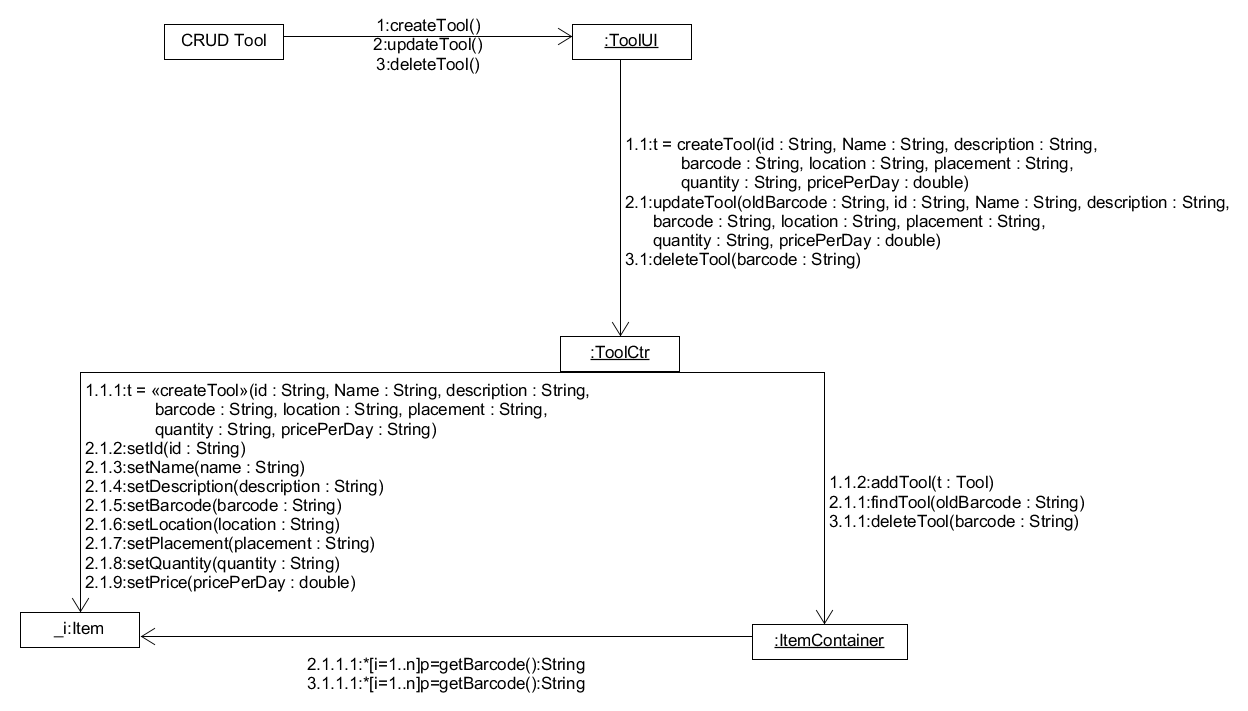
## Bilag 18:



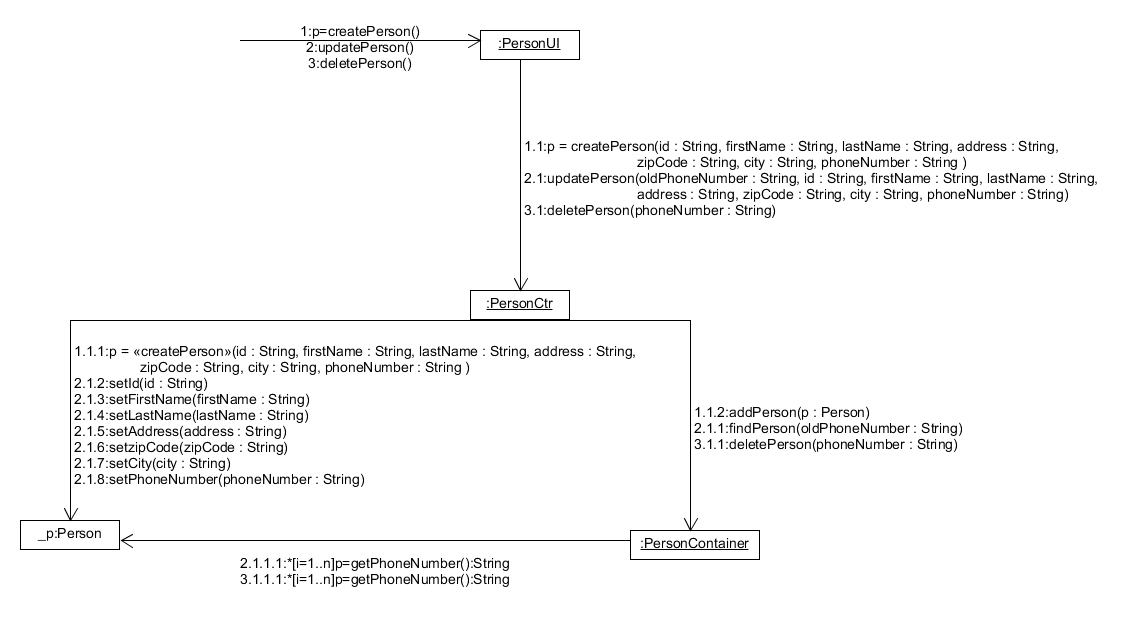
## Bilag 19:



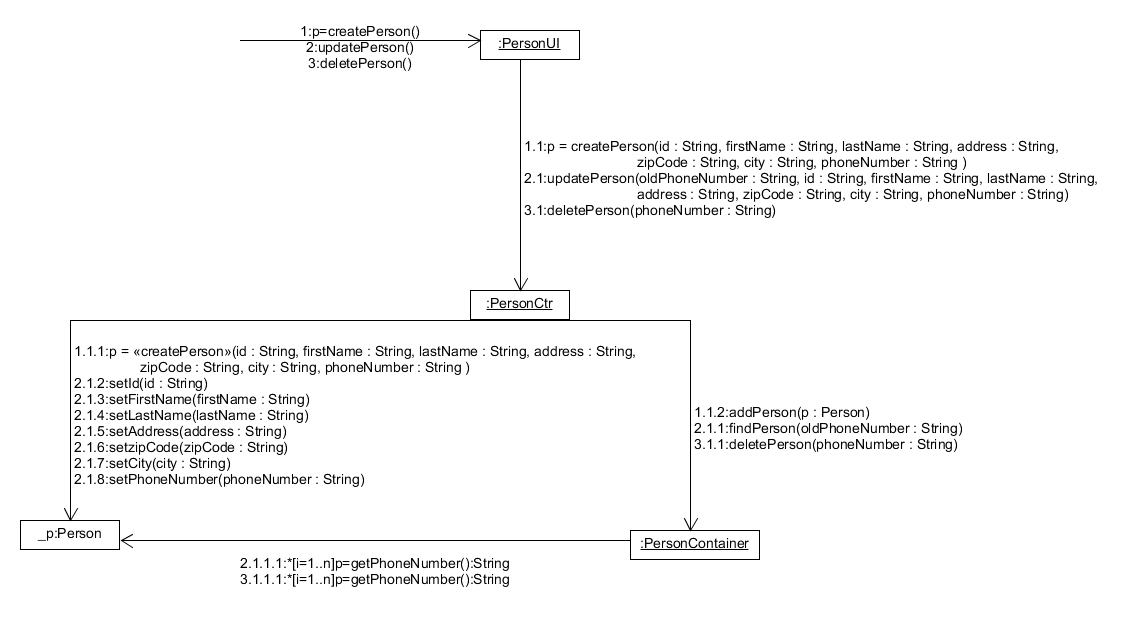
## Bilag 20:



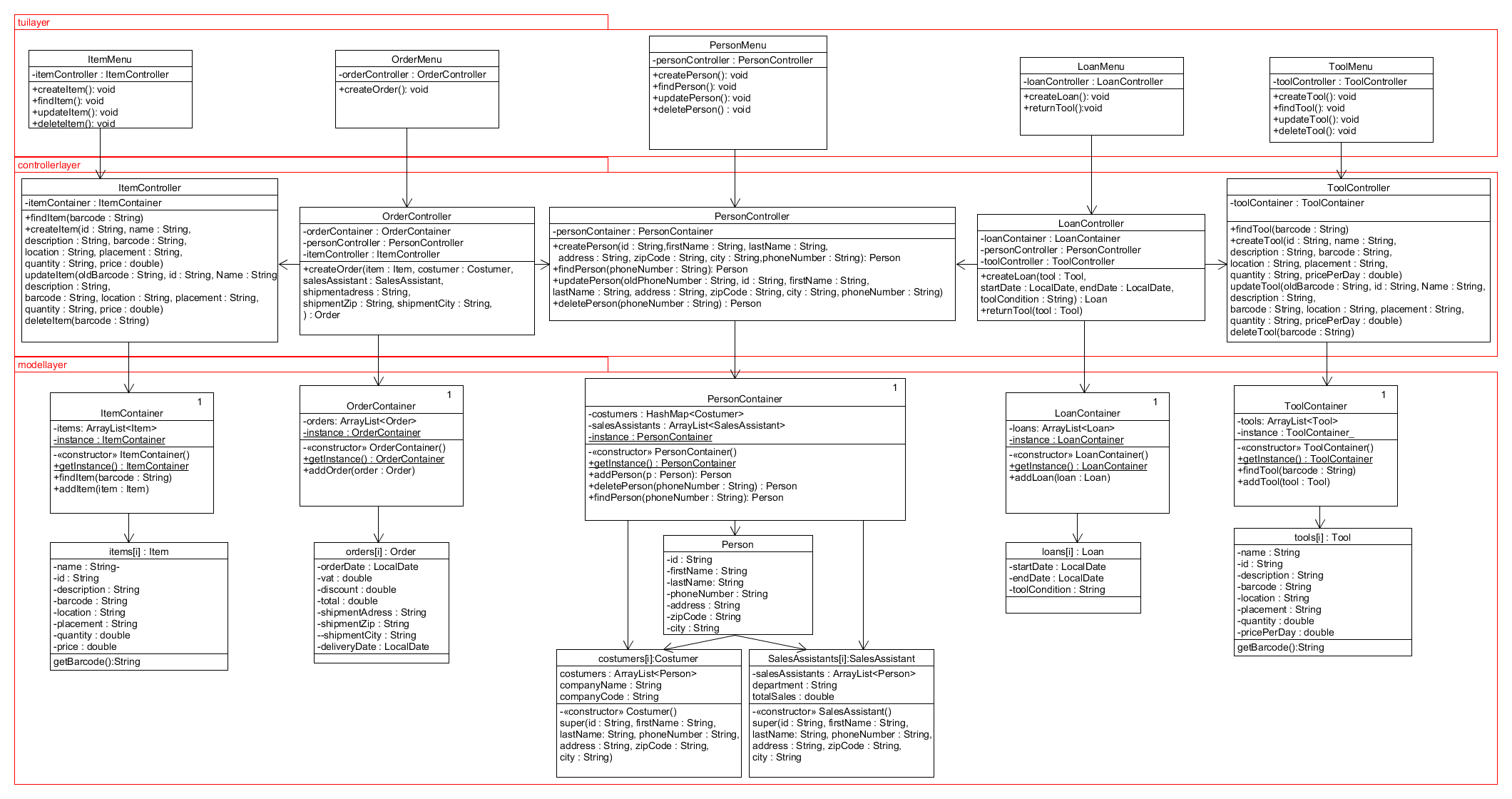
## Bilag 21:



## Bilag 22:



## Bilag 23:



## Bilag 24:

Et billede, der indeholder tekst

Automatisk genereret beskrivelse

1. “Code Conventions for the Java Programming Language”. [↑](#footnote-ref-2)
2. s.r.o, “GanttProject”. [↑](#footnote-ref-3)
3. Srivastava, “Ternary Operator In Java | Baeldung”. [↑](#footnote-ref-4)