

# Johannes Fog - Carport Case - Rapport

May 29, 2018

Jacob Borg - cph-jb308@cphbusiness.dk - Jack-Borg  
Daniel Lindholm - cph-dl110@cphbusiness.dk - Hupra  
Nikolaj Thorsen Nielsen - cph-nn134@cphbusiness.dk - NikolajX4000  
Stephan Marcus Duelund Djurhuus - cph-sd115@cphbusiness.dk - Stephan-MDD

Link til GitHub repository: <https://github.com/NikolajX4000/Sem2Exam>  
Link til deployet sitet: websitet  
Link til javadoc: <https://nikolajx4000.github.io/Sem2Exam/>

# Contents

<b>1</b>	<b>Indledning</b>	<b>3</b>
1.1	Baggrund . . . . .	3
1.2	Teknologivalg . . . . .	3
<b>2</b>	<b>Krav</b>	<b>3</b>
2.1	Overordnet beskrivelse af virksomheden . . . . .	4
2.2	Arbejdsgange der skal IT-støttes . . . . .	4
2.2.1	As-is . . . . .	4
2.2.2	To-be . . . . .	4
2.3	Scrum userstories . . . . .	4
<b>3</b>	<b>Domæne model og ER diagram</b>	<b>6</b>
<b>4</b>	<b>Navigationsdiagram</b>	<b>6</b>
<b>5</b>	<b>Sekvensdiagrammer</b>	<b>6</b>
<b>6</b>	<b>Særlige forhold</b>	<b>7</b>
6.1	Session . . . . .	7
6.2	Exceptions . . . . .	7
6.3	Brugerinput validering . . . . .	7
6.4	Login sikkerhed . . . . .	8
6.5	Brugertyper i databasen . . . . .	8
<b>7</b>	<b>Udvalgte kodeeksempler</b>	<b>8</b>
<b>8</b>	<b>Status på implementation</b>	<b>8</b>
<b>9</b>	<b>Test</b>	<b>8</b>
<b>10</b>	<b>Process</b>	<b>9</b>
10.1	Arbejdsprocessen faktisk . . . . .	9
10.2	Arbejdsprocessen reflekteret . . . . .	9
<b>11</b>	<b>Appendix</b>	<b>10</b>
11.1	Appendix A: Diagrammer . . . . .	11
11.1.1	Domæne model . . . . .	11
11.1.2	ER diagram . . . . .	12
11.1.3	Klassediagram . . . . .	13
11.1.4	Navigationsdiagram . . . . .	14
11.1.5	Aktivitetsdiagrammer . . . . .	15
11.1.6	Sekvensdiagrammer . . . . .	18
11.2	Appendix B: User Stories . . . . .	22
11.2.1	Implementeret . . . . .	22
11.2.2	Ikke implementeret . . . . .	22
11.3	Appendix C: Code Coverage . . . . .	23

# 1 Indledning

Vi er blevet kontaktet af Johannes Fog Værebros, der har spurgt om vi kan lave et program og tilhørende hjemmeside til deres salg af carporte. Deres nuværende system bygger på manuelt arbejde samt en ikke tilstrækkelig adgang til ændringer i både system og database. Dette er nogle af fokuspunkterne der er stillet i vores opgave.

## 1.1 Baggrund

Johannes Fog er en koncern der både har design & bolighuse og trælast & byggecenter. Vi vil i dette projekt fokusere på Fog Værebros, der har bedt os om at opdatere deres system. Fog Værebros sælger som de andre trælast & byggecentre, træ, byggematerialer og alt det du behøver til hus og have inden for f.eks. maling, bad og VVS, beslag, elartikler og lamper samt haveredskaber, grill og havemøbler, men derudover har de gjort det til deres varemærker at være specialister i carporte. Det er i den forbindelse at vi er blevet bedt om at udvikle et system, med tilhørende hjemmeside, til at erstatte deres nuværende system, da de har erkendt at det er outdated. Med systemet skal man kunne gå på hjemmesiden og bestille en carport, systemet skal kunne udregne en stykliste af materialer, en medarbejder skal kunne gå ind og give en kunde særlige tilbud, og en kunde skal kunne se sin ordre.

## 1.2 Teknologivalg

- Projektet er et Maven 3.1
- Projekt skrevet i Netbeans IDE 8.2
- mysql 5.1.39
- MySQL Workbench 6.3ce
- Ubuntu 16.04.3 x64 server
- Websitet er deployet vha. apache-tomcat-8.0.32
- Sprogene der er brugt i koden er java jdk 1.8.0\_141
- JAVAX 7.0
- HTML5
- JSTL 1.2 til at opbygge JSP sider
- JBCrypt 0.4 til salt og hashing af passwords

# 2 Krav

På websitet skal man som kunde kunne bestille en carport. Man skal kunne vælge om taget skal være med eller uden rejsning, og hvis man har valgt med skal man kunne vælge hvor meget hældning man vil have på taget. Man skal kunne vælge om man vil have et redskabskur, og hvor stort det skal være. Når man bestiller en carport skal man kunne indtaste sine oplysninger, navn, telefonnummer, email og adresse. Når man har bestilt en carport kan man se alle de ordre der er bestilt med den samme email. Man skal som bruger kunne se alle sine tidligere ordre ved at indtaste sin email. Som medarbejder skal man kunne se alle ordre. Man skal kunne ændre status og pris på en ordre. Man skal kunne navne og priser på de forskellige typer træ, skruer og beslag.

Systemet skal kunne udregne en stykliste ud fra en ordre og en vejledende pris ud fra styklisten, som en medarbejder kan bruge til at lave et tilbud til kunden.

## 2.1 Overordnet beskrivelse af virksomheden

Fog er en koncern med afdelinger i det meste af nordsjælland og en enkelt i Vordingborg. Af deres 10 afdelinger har de 1 afdeling med design og bolighus placeret i Lyngby, og 9 afdelinger med trælast og byggecenter placeret i Farum, Fredensborg, Helsingør, Herlev, Hørsholm, Kvistgård, Lyngby, Værebros og Vordingborg. Udover at være trælast og byggecenter, har Fog i Værebros gjort det til sit varemærke at være specialister i carporte.

Fogs arbejdsopgaver kan inddeles i følgende funktioner:

Salgsfunktion	Indkøbs- og lagerfunktion	Økonomifunktion
Kundebetjening	Valg af leverandør	Opstille økonomiske mål
Markedsføring	Varebestilling	Budget og regnskab
Vareopfyldning	Varemodtagelse	Bogføring og løn
Sætte kunde i kontakt med tømrer	Modtagekontrol	Budgetkontrol

## 2.2 Arbejdsgange der skal IT-støttes

### 2.2.1 As-is

Når en kunde laver en ordre bliver der genereret en email, som en medarbejder manuelt skal indtaste i IT-systemet. Fog vil gerne derefter kunne ringe og snakke med kunden da kunde service er meget vigtigt for Fog. Efter samtale med kunden har medarbejderen mulighed for at lave ændringer til styklisten og lave et tilbud til kunden.

### 2.2.2 To-be

Når en kunde opretter en ordre vil IT-systemet udregne en stykliste og en vejledende pris ud fra styklisten. En e-mail bliver sendt til fog for at underrette om den nye ordre i systemet, en medarbejder tjekker ordren og har mulighed for at lave rettelser i styklisten og derefter kan medarbejderen sende et tilbud til kunden.

Aktivitetsdiagram: Kunde bestiller en carport kan ses i Appendix A: Diagrammer: Figure 5.

Aktivitetsdiagram: Kunde ser sine ordre kan ses i Appendix A: Diagrammer: Figure 6.

Aktivitetsdiagram: Medarbejder ændre på tilgængeligt materiale kan ses i Appendix A: Diagrammer: Figure 7.

## 2.3 Scrum userstories

Som kunde vil jeg kunne bestille en carport så jeg kan få en carport.

how-to-demo:

- Bruger skal kunne vælge om carporten skal være med eller uden rejsning.
- Bruger skal kunne vælge størrelse mm. på en carport.

tasks:

- Opret order tabel i databasen.
- Opret shoppage med form til at angive størrelse på carport.

- Opret ordermapper.
- Opret user til databasen.
- Giv useren til databasen insert privilegier.
- Opret create metode i ordermapper.
- Opret order klasse.
- Command til bestilling.

**estimat:** Estimeret til en uges arbejde.

**Som kunde vil jeg kunne se en tegning af min carport så jeg kan se hvad det er jeg køber.**

**how-to-demo:**

- Brugeren skal kunne se en tegning af en carport med fladt tag fra oven.
- Brugeren skal kunne se en tegning af en carport med fladt tag fra siden.
- Brugeren skal kunne se en tegning af en carport med rejsning fra oven.
- Brugeren skal kunne se en tegning af en carport med rejsning fra siden.

**tasks:**

- Lav stykliste for carport med fladt tag fra oven.
- Lav stykliste for carport med fladt tag fra siden.
- Lav stykliste for carport med rejsning fra oven.
- Lav stykliste for carport med rejsning fra siden.
- Lav tabel i database med tilgængeligt materiale.

**estimat:** Estimeret til en dags arbejde.

**Som medarbejder vil jeg kunne udregne en stykliste så jeg ved hvilke materialer der skal bruges.**

**how-to-demo:**

- Styklisten stemmer overens med tegningerne.

**tasks:**

- Lav stykliste for carport med fladt tag.
- Lav stykliste for carport med rejsning.

**estimat:** Estimeret til en uges arbejde.

**Som medarbejder vil jeg kunne se en vejledende pris baseret på en styklisten så jeg har et udgangspunkt for at kunne give kunden et tilbud.**

**how-to-demo:**

- Medarbejderen kan se en pris der er beregnet ud fra styklisten.

**tasks:**

- Udregning for carport med fladt tag.

- Udregning for carport med rejsning.

**estimat:** Estimeret til en times arbejde.

**Som bruger af systemet vil jeg have et brugervenligt design så jeg kan finde rundt.**

**how-to-demo:**

- Sitet ser godt ud og er nemt at finde rundt i.

**tasks:**

- Formatér dato.
- Formatér pris.
- Styling.

**estimat:** Estimeret til et par timers arbejde.

### 3 Domæne model og ER diagram

Domæne model kan ses i Appendix A: Diagrammer: Figure 1.

ER diagram kan ses i Appendix A: Diagrammer: Figure 2.

Klassediagram kan ses i Appendix A: Diagrammer: Figure 3.

### 4 Navigationsdiagram

Navigationsdiagram kan ses i Appendix A: Diagrammer: Figure 4.

Når man kommer ind på websiden kommer man først til index siden.

Derfra har man mulighed for at komme til makeCarport siden, hvor man kan bestille en carport, derfra kommer man over på specificUserOrders siden, hvor man kan se alle sine ordre. Man kan også komme til specificUserOrders siden fra index siden ved at indtaste sin email.

Man har fra index siden også mulighed for at komme til login siden, hvor en medarbejder kan logge ind. Når man logger ind kommer man til allOrders siden, hvor man får en liste med alle ordre.

Vi bruger en header med links, så man kan fra alle siderne komme til index siden, makeCarport siden, og specificUserOrders siden, og hvis man er logget ind kan man også komme til allOrders siden og editMaterials siden, hvor man kan se tilgængeligt materiale, og ændre på priser og længder.

Vi har også en footer, hvor vi har et link til login siden, så man kan også komme dertil fra alle siderne.

### 5 Sekvensdiagrammer

Sekvensdiagram: Placer ordre kan ses i Appendix A: Diagrammer: Figure 8.

Sekvensdiagram: Se ordre kan ses i Appendix A: Diagrammer: Figure 9.

Sekvensdiagram: Opdatér ordre kan ses i Appendix A: Diagrammer: Figure 10.

Sekvensdiagram: Inspicér ordrekant kan ses i Appendix A: Diagrammer: Figure 11.

## 6 Særlige forhold

### 6.1 Session

Hvis man logger ind som en employee gemmer vi en reference til et employee objekt med information om den bruger. Denne information bruges til at se om man har adgang til forskellige sider.

### 6.2 Exceptions

Vi arbejder primært med to forskellige exceptions "CustomException" og "NoAccessException".

Hvis der opstår en anden slags exception f.eks.. en SQLException eller NumberFormatException vil den blive grebet, og der vil blive kastet en ny "CustomException" som man så giver en brugervenlig besked, som der vil kunne blive vist til brugeren på siden med hvad der gik galt.

CustomExceptions bliver grebet i vores forskellige Command klasser hvor de bliver lagt ned i en attribut ved navn "feedback", som så vil blive skrevet ud på siden.

NoAccessException kan opstå hvis man ikke har tilladelse til at tilgå den command man prøver at lave. Der vil så blive kastet en NoAccessException som bliver grebet på vores FrontController, der så vil sende brugeren hen til en anden side.

### 6.3 Brugerinput validering

Der bliver lavet brugervalidering både på front og backend.

På frontend bruger vi lidt forskellige ting, men primært HTML ting som 'required' hvis et felt skal være udfyldt.

Hvis man f.eks. skal indtaste et postnummer, er der blevet brugt regular expressions til at sikre det man taster ind er et gyldigt dansk postnummer: "`\d{4}`"

det ovenstående statement altså "`\d{4}`" siger at man kun vil modtage 4 digits, og ikke nogle bogstaver, tegn eller mindre/flere end 4 digits.

Hvis man kigger på denne expression som validere telefonnumre og prøver at bryde den ned så den er til at forstå: "`[+]?([\d][ -]?){4,19}`"

- `[+]?` siger at nummeret må starte med 0 eller 1 '+' tegn.
- `([\d][ -]?){4,19}`
  - `[\d]` siger der må være et digit
  - `[ -]?` siger der må være enten et space eller en bindestreg 0 eller 1 gang.
  - `{4,19}` siger at den sekvens der står før må forekomme 4-19 gange

Så den siger: Der må være 0 eller 1 '+' tegn, efterfulgt af 4-19 ganges: digit med mulighed for et space eller bindestreg efter.

For at sikre os at man ikke kan lave en ordre på et skur der ikke giver mening f.eks. 3x3 meter carport med 4x4 meter skur er der lavet javascript(jQuery) der hver gang man ændre størrelsen af skuret sørger for at man ikke kan lave et skur der er for stort. Hvis dette skulle gå galt bliver det også tjekket på backenden.

Fordi vi validere på frontenden antager vi at det data som bliver sendt over til vores backend er acceptabel, derfor prøver vi bare at bruge den data som er sendt med, hvis af en eller anden grund det data vi får med ikke er ordentligt vil der blive kastet en 'CustomException' og aktionen vil blive stoppet.

## 6.4 Login sikkerhed

Sikkerhed i forbindelse med login er lavet med BCrypt, som både står for salt og hashing af passwords.

Hvis man prøver at logge ind, vil vores program se om det indtastede brugernavn findes i databasen, hvis det gør det sammenligner man så det indtastede password med det hashed password fra databasen.

## 6.5 Brugertyper i databasen

Vi har to forskellige brugertype i vores database.

Den første en den vi bruger på selve sitet ved navn: "Fog" denne bruger har kun de rettigheder som den har brug for og ikke andet.

Vores anden bruger er kaldt: "fogTest" denne bruger kan gøre alt og har ikke nogen restriktioner

# 7 Udvalgte kodeeksempler

# 8 Status på implementation

Som det kan ses i Appendix B: User Stories er der user stories vi ikke har implementeret, dette er da PO har valgt at nedprioritere dem i forhold til de andre user stories. Vi har derfor ikke lave metoder til at kunne tilføje og fjerne materialer, da det var en del af en af disse user stories.

Vi har lige nu kun sat op så der sendes en email til fog med at der er lavet en ny ordre, vi kunne godt tænke os at der blev sendt flere emails med opdatering så som, at når man lavede en ordre også modtog en bekræftelse der sagde noget i stil med "Tak for din ordre du vil blive kontaktet snarest af en medarbejder". Eller når status bliver opdateret af en medarbejder at man som kunde også modtager en email der fortæller status af ens ordre er blevet opdateret.

# 9 Test

Code Coverage kan ses i Appendix C: Code Coverage: Figure 12.

Som udvikler er test af programmet en vigtig del af processen. Lever koden ikke op til de givne test må den revurderes og testes igen, indtil alle testene går igennem. En måde at sikre at alle metoder gennemføre deres test er at bruge Test Driven Development (TDD). TDD er en god måde at sikre signe test på, da man starter med testen og derefter udarbejder metoden til at løse den. Desværre lærte vi først denne udviklingsmetode midvejs gennem forløbet hvilket gjorde den svær at implementere, da det er en metode der skal implementeres i starten af projektet.

Vores testmiljø består af JUnit test og et plugin som hedder TikiOne JaCoCoverage. Vores JUnit Mapper test er koblet til vores aktuelle database, hvilket vi ved ikke er den optimale måde at teste på. Der er andre måder, som f.eks. at oprette diverse test tables i databasen som er identiske til de aktuelle og som kun har forbindelse til vores test run. Dette vil kun sikre at vores test ikke skabte konflikt i vores program senere hen, men vil ikke reducere forbindelsen mellem program og server. De resterende JUnit test er blevet udarbejdet på den traditionelle whitebox måde, da der ikke er



nogle specielle faktore der spiller ind. Whitebox tetsing er en test metode der giver person som tester mulighed for at følge med i kode contra Blockbox testing hvor der ikke er adgang til koden. Vi har brugt et pugin der heder TikiOne JaCoCoverage, som skaber at overblik over hvilke metoder der er blevet testet.

## **10 Process**

### **10.1 Arbejdsprocessen faktuel**

### **10.2 Arbejdsprocessen reflekteret**

## 11 Appendix

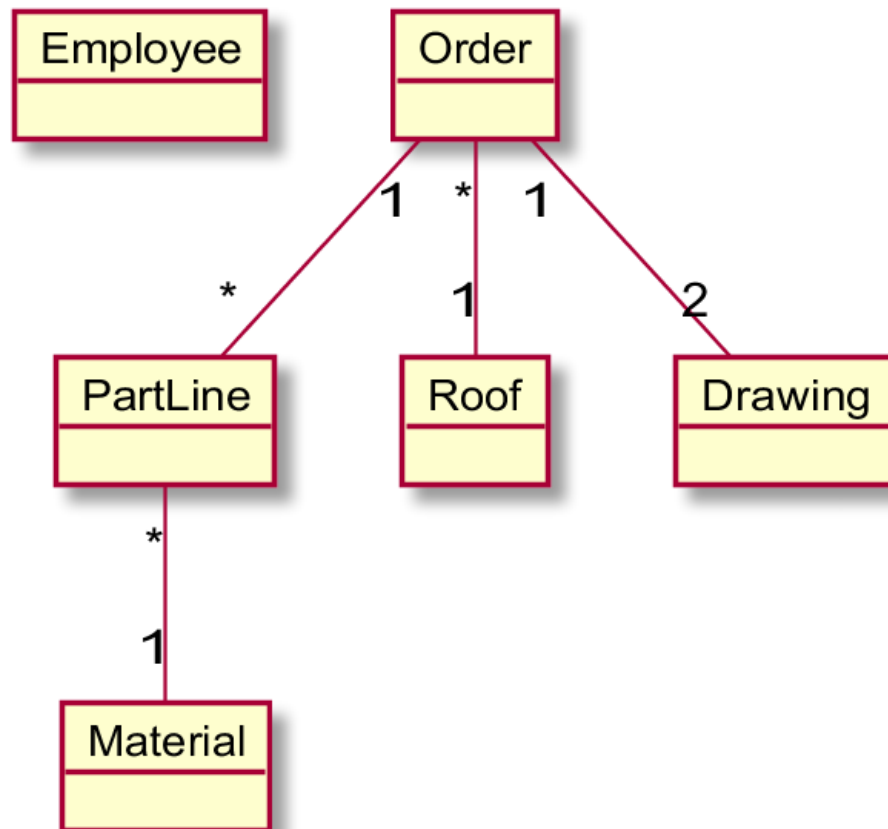
### List of Figures

1	Domæne model . . . . .	11
2	ER diagram . . . . .	12
3	Klassediagram . . . . .	13
4	Navigationsdiagram . . . . .	14
5	Aktivetsdiagram: Kunde bestiller en carport . . . . .	15
6	Aktivetsdiagram: Kunde ser sine ordre . . . . .	16
7	Aktivetsdiagram: Medarbejder ændre på tilgængeligt materiale . . . . .	17
8	Sekvensdiagram: Placer ordre . . . . .	18
9	Sekvensdiagram: Se ordre . . . . .	19
10	Sekvensdiagram: Opdater ordre . . . . .	20
11	Sekvensdiagram: Inspicer ordre . . . . .	21
12	Code Coverage . . . . .	23

## 11.1 Appendix A: Diagrammer

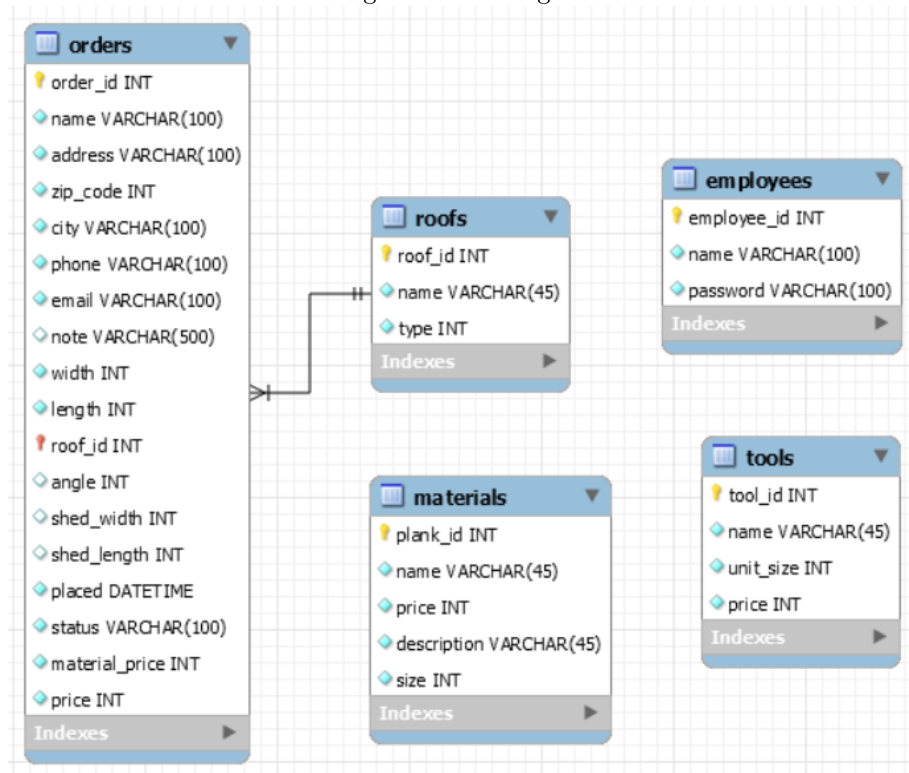
### 11.1.1 Domæne model

Figure 1: Domæne model



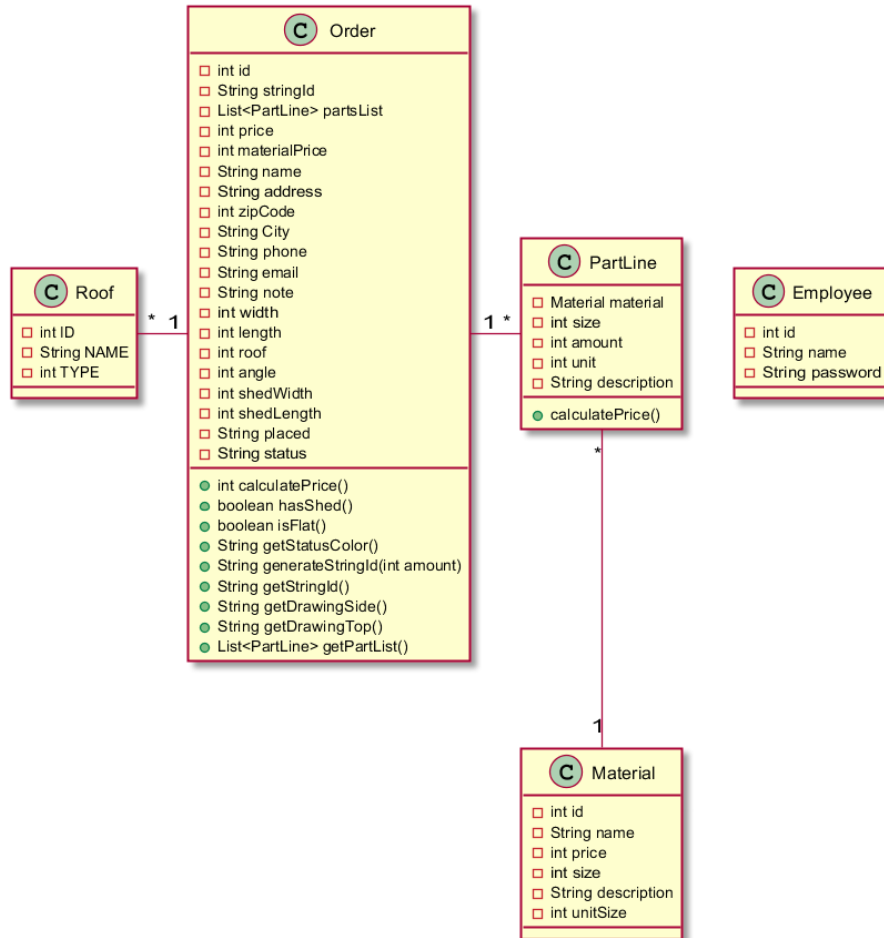
### 11.1.2 ER diagram

Figure 2: ER diagram



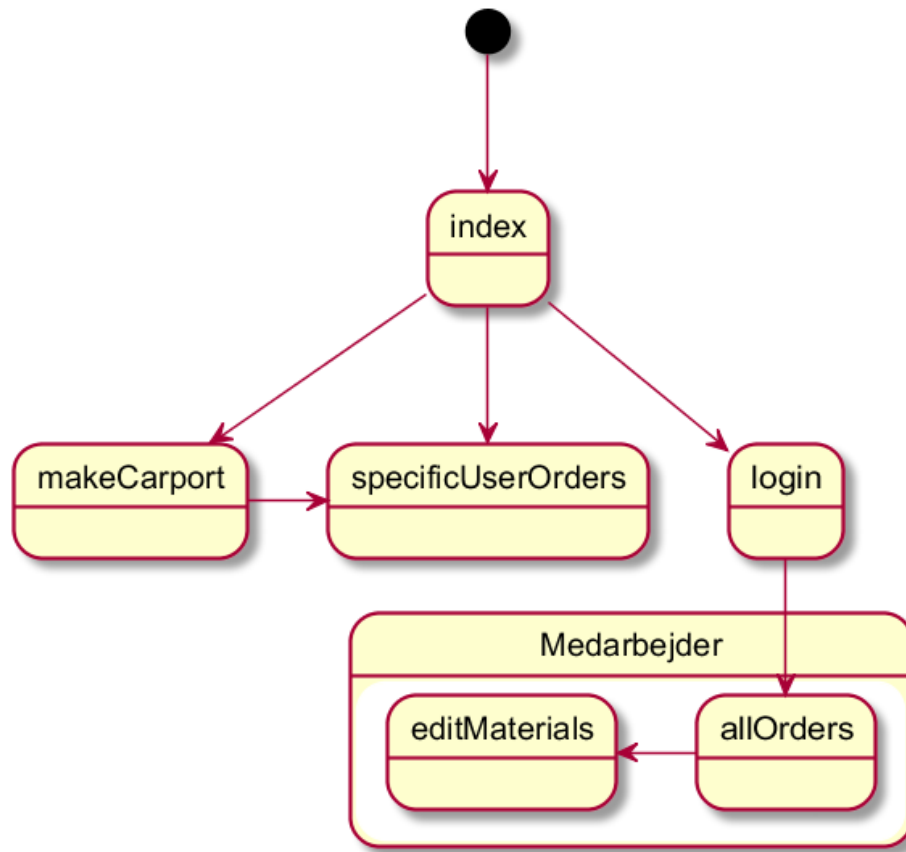
### 11.1.3 Klassediagram

Figure 3: Klassediagram



#### 11.1.4 Navigationsdiagram

Figure 4: Navigationsdiagram



### 11.1.5 Aktivitetsdiagrammer

Figure 5: Aktivitetsdiagram: Kunde bestiller en carport

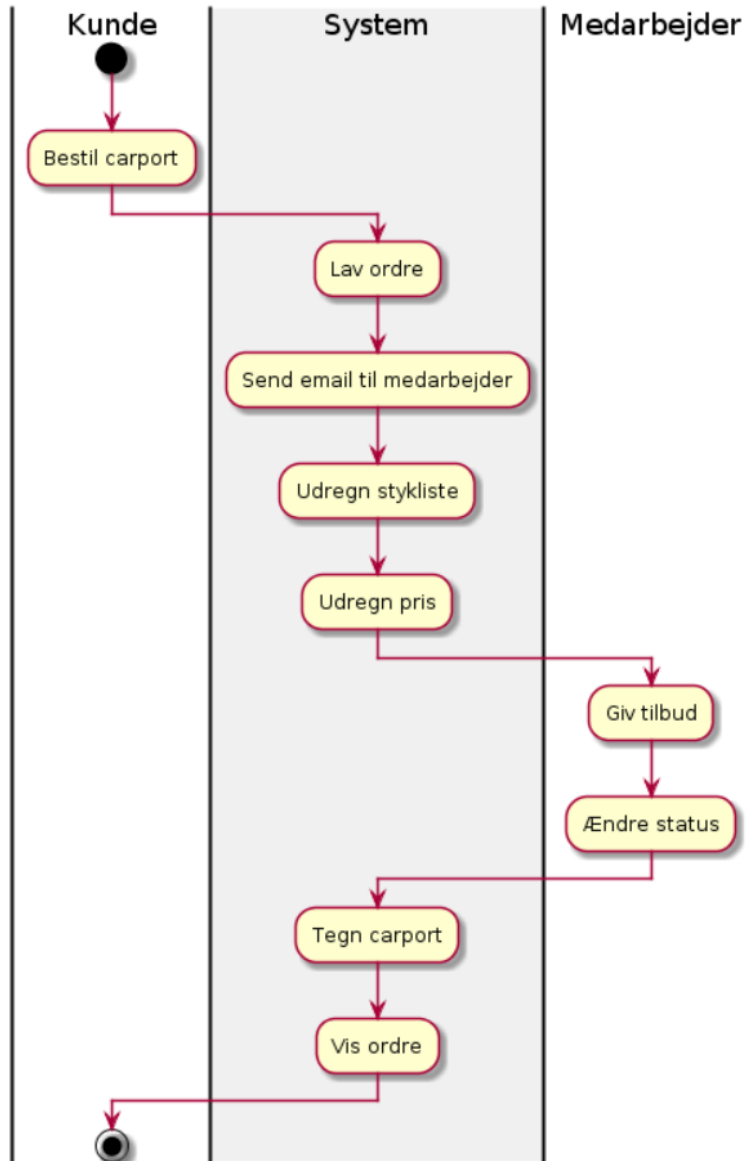


Figure 6: Aktivitetsdiagram: Kunde ser sine ordre

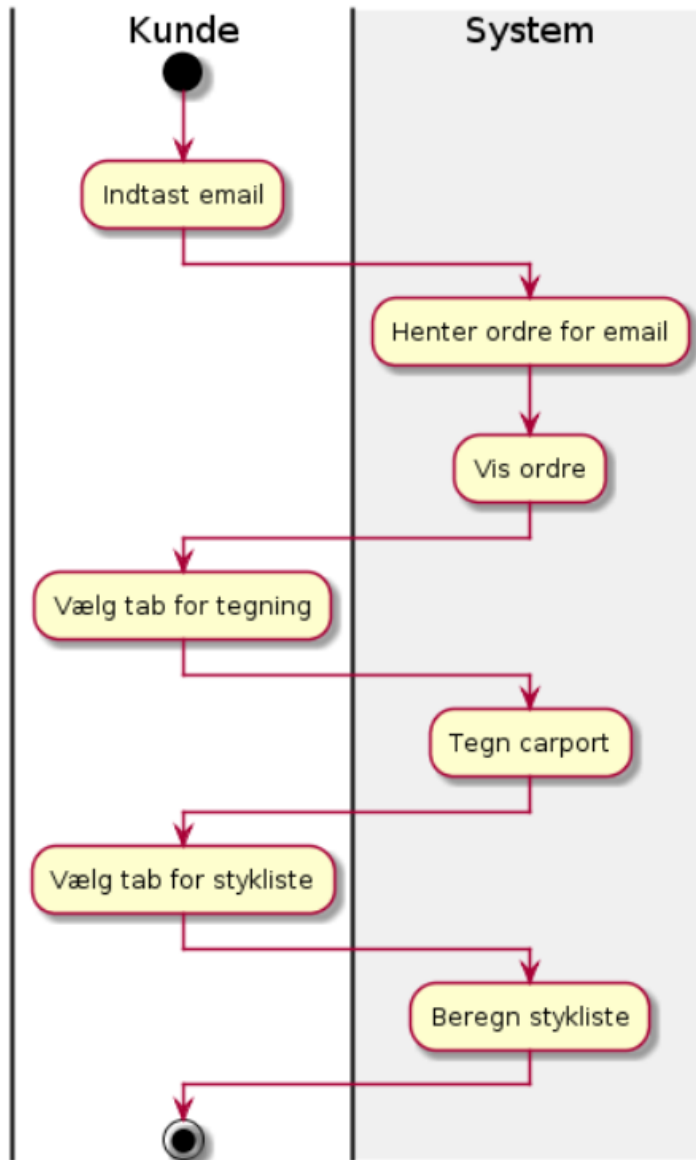




Figure 7: Aktivitetsdiagram: Medarbejder ændre på tilgængeligt materiale



### 11.1.6 Sekvensdiagrammer

Figure 8: Sekvensdiagram: Placer ordre

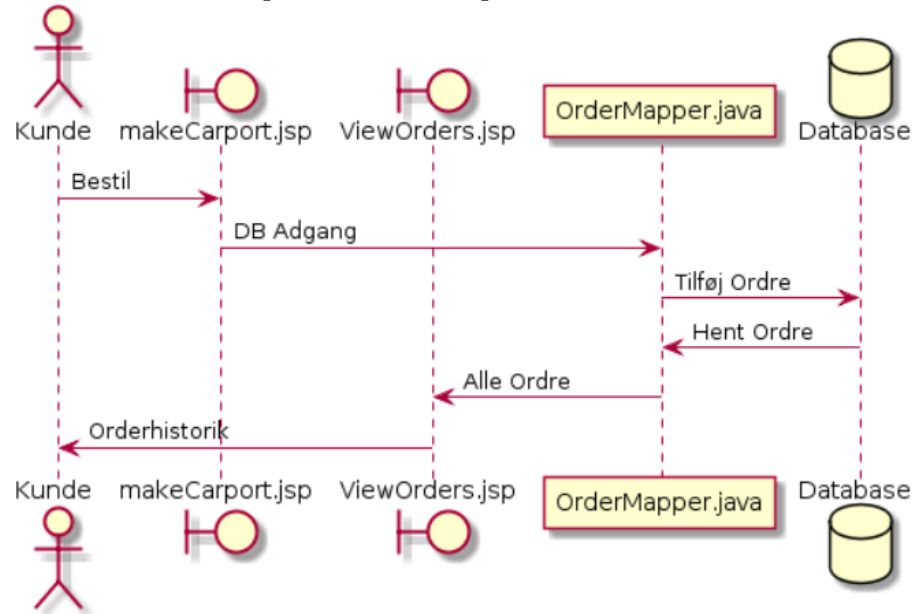


Figure 9: Sekvensdiagram: Se ordre

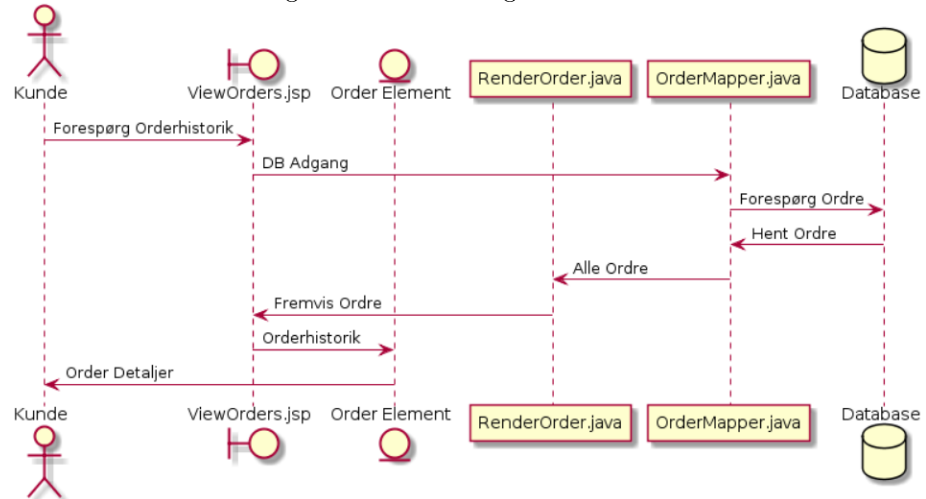


Figure 10: Sekvensdiagram: Opmåler ordre

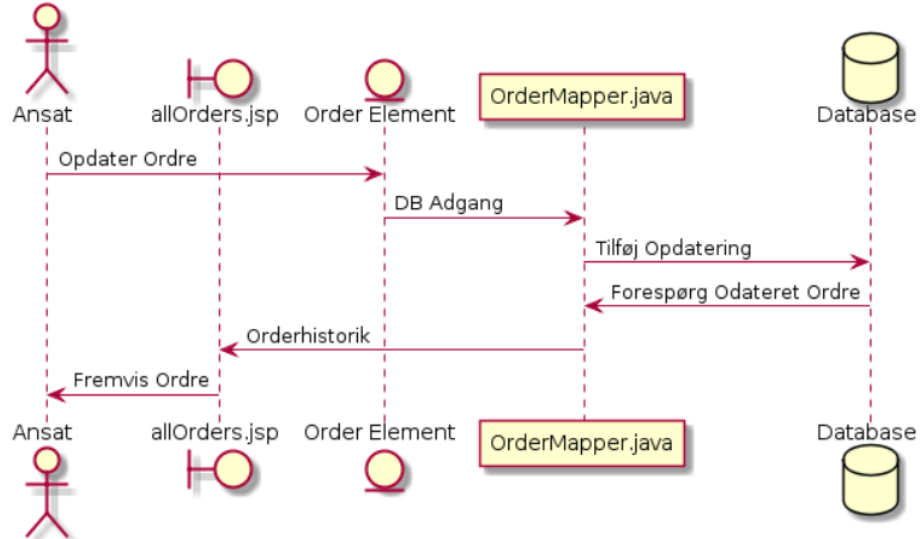
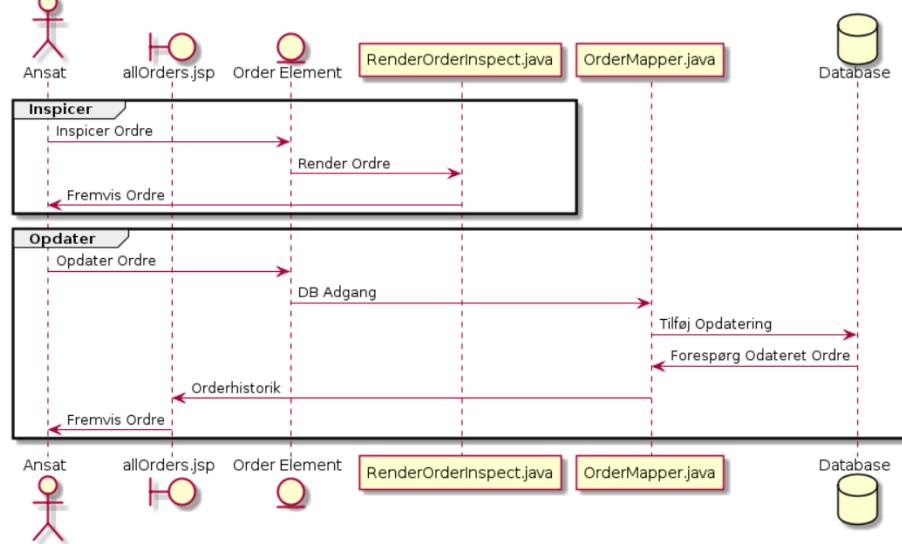


Figure 11: Sekvensdiagram: Inspicér ordre



## 11.2 Appendix B: User Stories

### 11.2.1 Implementeret

- Som bruger af systemet vil jeg have et brugervenligt design så jeg kan finde rundt
- Som kunde vil jeg have mulighed for at kontakte en medarbejder så jeg kan få rådgivning
- Som kunde vil jeg have mulighed for at kunne se mine ordre og deres status så jeg kan holde mig opdateret
- Som kunde vil jeg kunne bestille en carport så jeg kan få en carport
- Som kunde vil jeg kunne se en tegning af min carport så jeg kan se hvad det er jeg køber
- Som kunde vil jeg kunne se hvornår jeg har afgivet min ordre i et læseligt format så jeg kan se hvornår jeg har bestilt min carport
- Som kunde vil jeg kunne se prisen på min carport så jeg ved om jeg har råd til den
- Som medarbejder vil jeg have besked når en ny ordre er blevet placeret så jeg hurtigt kan hjælpe kunde med at få sin carport
- Som medarbejder vil jeg have medarbejderspecifikke operationer gemt væk bag et login så kunder ikke har adgang
- Som medarbejder vil jeg have mulighed for at ændre status for en ordre så kan holde kunden opdateret
- Som medarbejder vil jeg kunne se en vejledende pris baseret på en styklisten så jeg har et udgangspunkt for at kunne give kunden et tilbud
- Som medarbejder vil jeg kunne udregne en stykliste så jeg ved hvilke materialer der skal bruges
- Som medarbejder vil jeg være i stand til at annullere en ordre så det er muligt at annullere sin ordre
- Som medarbejder vil jeg være i stand til at ændre på tilgængelige materialer så jeg ved hvilke materialer jeg har

### 11.2.2 Ikke implementeret

- Som kunde vil jeg kunne opdatere mine informationer så jeg kan få min carport leveret til den rigtige adresse
- Som kunde vil jeg kunne se et estimat for hvornår jeg kan få min carport så jeg ved hvornår jeg får min carport
- Som medarbejder vil jeg kunne lave et nyt kodeord hvis jeg glemmer mit kodeord så jeg stadig kan arbejde hvis jeg glemmer mit kodeord
- Som medarbejder vil jeg kunne slette og tilføje materialer så jeg kan tilpasse mig ændringer i tilgængeligt materiale
- Som medarbejder vil jeg kunne ændre en stykliste så jeg kan være sikker på at den er rigtig

## 11.3 Appendix C: Code Coverage

Figure 12: Code Coverage

