Fog Carporte

Maj 2020



|  |  |
| --- | --- |
| **Datamatiker 2. Semester, Hold E, Gruppe 666** | |
| **Alexander Pihl**  *Email:* [cph-as509@cphbusiness.dk](mailto:cph-as509@cphbusiness.dk)  *Github:* <https://github.com/AlexanderPihl> | **Morten Rahbek**  *Email:* [*cph-mr462@cphbusiness.dk*](mailto:cph-mr462@cphbusiness.dk)  *Github:* [*https://github.com/amazingh0rse*](https://github.com/amazingh0rse) |
| **Mick Larsen**  *Email:* [*cph-ml616@cphbusiness.dk*](mailto:cph-ml616@cphbusiness.dk)  *Github:* [*https://github.com/MivleDK*](https://github.com/MivleDK) | **Per Kringelbach**  *Email:* [cph-pk171@cphbusiness.dk](mailto:cph-pk171@cphbusiness.dk)  *Github:* <https://github.com/cph-pk> |
| **Jean-Poul Leth-Møller**  *Email:* [*cph-jl360@cphbusiness.dk*](mailto:cph-jl360@cphbusiness.dk)  *Github:* [*https://github.com/Jean-Poul*](https://github.com/Jean-Poul) | |

Indhold

[0. Projektformaliteter 3](#_Toc41590777)

[1. Indledning (Evt indrag problemformulering her) 4](#_Toc41590778)

[1.1. Baggrund 4](#_Toc41590779)

[1.3 Teknologivalg 4](#_Toc41590780)

[2. Krav 4](#_Toc41590781)

[2.1 Overordnet virksomhedsbeskrivelse 4](#_Toc41590782)

[2.2 Arbejdsgange der skal IT-støttes 4](#_Toc41590783)

[3 Diagrammer 4](#_Toc41590784)

[3.1 ER Diagram 4](#_Toc41590785)

[3.2 Navigationsdiagram 5](#_Toc41590786)

[3.3 Klassediagram 6](#_Toc41590787)

[4 Særlige forhold 6](#_Toc41590788)

[5 Udvalgte kodeeksempler 10](#_Toc41590789)

[5.1 Carport Calculation 10](#_Toc41590790)

[5.2 Price Calculation 13](#_Toc41590791)

[6 Status på implementation 14](#_Toc41590792)

[7 Test (Husk word fil fra Jon her og indrag v-model) 15](#_Toc41590793)

[8 Process 16](#_Toc41590794)

[8.1 Arbejdsprocessen faktuelt 16](#_Toc41590795)

[Sprint 1 16](#_Toc41590796)

[Sprint 2: 16](#_Toc41590797)

[Sprint 3: 17](#_Toc41590798)

[Sprint 5: 18](#_Toc41590799)

[8.1.1 SCRUM Userstories 18](#_Toc41590800)

[8.2 Arbejdsprocessen reflekteret 18](#_Toc41590801)

[9 Konklusion 19](#_Toc41590802)

[10 Perspektivering 19](#_Toc41590803)

# 0. Projektformaliteter

Produktets demovideo: a

Produktets website: a

Projektets kildekode: <https://github.com/Jean-Poul/Eksamensprojekt_Fog>

Projektets JavaDoc: a

# 1. Indledning (Evt indrag problemformulering her)

*Kort intro til hvad dette projekt omhandler. Formålet med indledning er at sætte en fagfælle i stand til at forstå resten af rapporten. For jer som studerende er en “fagfælle” en anden datamatiker studerende på 2. semester der er på samme niveau, men som ikke kender opgaven.*

## 1.1. Baggrund

*En kort beskrivelse af den virksomhed som skal bruge systemet*

*Hvilke krav kunden har til systemet forklaret i brede termer, f.eks. “kunden skal kunne bestille en cupcake, hvor man kan vælge både bund og top.” i modsætning til “der skal være en drop down menu med scrollbar i højre side med en liste over hvilke bunde der er og hvad de koster”.*

## 1.3 Teknologivalg

*En kort beskrivelse af hvilke teknologier der er brugt (jdbc, mysql, …). Her er det ikke meningen at I skal beskrive disse teknologier, men man skal sige hvilke der er brugt sådan at den der skal overtage projektet ved hvilken software der skal bruges. Der skal versions numre på (Netbeans 8.2, ikke blot “Netbeans”).*

# 2. Krav

*Der er to dele i dette afsnit:*

*- Hvad er firmaets håb med dette system*

*- scrum userstories (product backlog)*

## 2.1 Overordnet virksomhedsbeskrivelse

Beskriv hvad firmat Fog forventer og ønsker at få ud af dette nye IT-system

## 2.2 Arbejdsgange der skal IT-støttes

Dels skal afsnittet beskrive de overordnede arbejdsgange før og efter IT systemet.

# 3 Diagrammer

## 3.1 ER Diagram

*Det interesante ved denne domæne og database er at den langt hen af vejen er grundlaget for resten at systemet. Tabeller og relationer siger noget om hvad systemet arbejder med, ikke hvordan. Så det er godt sted at starte.*

*Som led i beskrivelsen af ER diagram skal man have følgende med:*

*Diagram over hele modellen. Det er vigtigt at få plads til alle tabeller og alle relationer. Det kan så betyde at man ikke kan få plads til alle attributter på de enkelte tabeller. Dem kan man slå op i databasen, så det er ikke så vigtigt*

*Hvis nogle af tabellerne ikke er på 3. normal form vil det være almindeligt at nævne det, og forklare hvorfor det er gjort (tidspres eller anden overvejelse).*

*Hvis der anvendes 1-1 relationer kan man beskrive hvorfor man ikke blot har en tabel.*

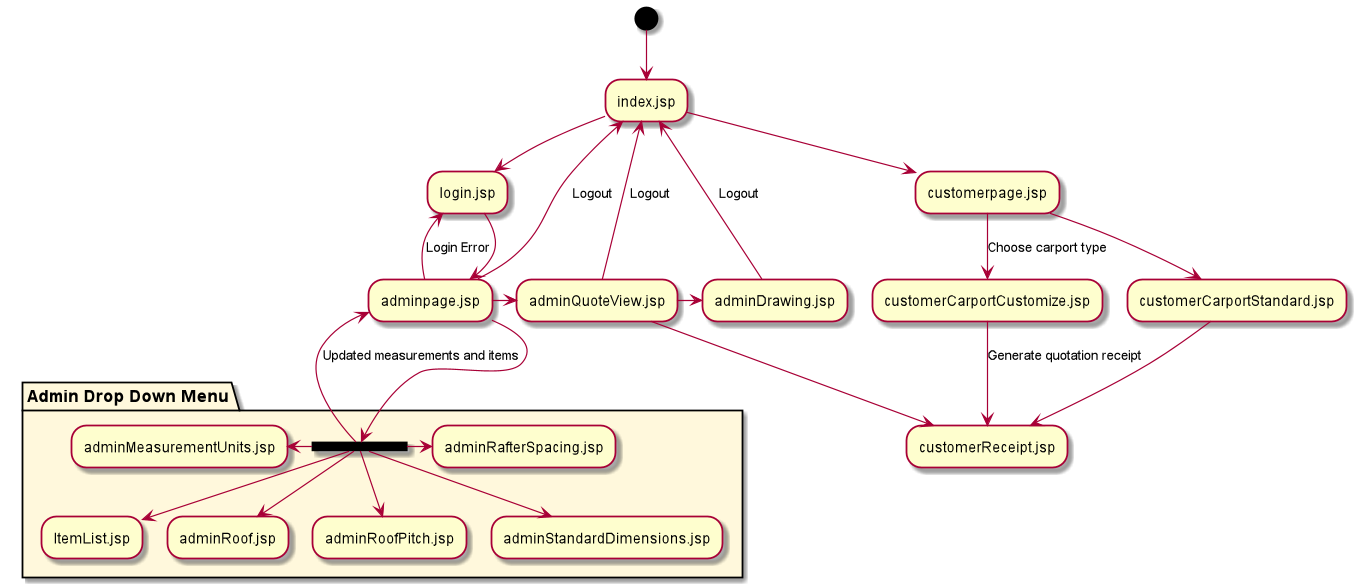
*Hvis nogle tabeller implementerer en mange-mange relation vil det være normalt at beskrive det.*

*Hvis der er flere veje at nå fra et sted til et andet vil det nemt gøre det svært at holde databasen konsistent. Hvis I har gjort det alligevel så skal I forklare hvorfor.*

*Hvis der er tabeller hvor man benytter andet end et automatisk generetet ID som nøgle skal man forklare det.*

*Der er interessant at beskrive hvilke overvejelser der ligger til grund for de konkrete valg der er i ER modellen (fremmednøgler, constraints, triggers, osv)*

## 3.2 Navigationsdiagram



Ovenstående diagram afspejler navigationen gennem Fogs hjemmeside, som enten kunde eller Fog medarbejder. Dette er beskrevet for at få en bedre forståelse af brugeroplevelsen på siden.

Brugeren starter på velkomstsiden index.jsp hvor der vises en navigation bar i toppen af siden. I denne navigation bar er det muligt for brugeren at trykke Home, Bestil eller Log ind. Trykker brugeren Home bliver siden redirected til velkomstsiden igen og fungere på samme måde på samtlige jsp sider.

Er brugeren en kunde har brugeren mulighed for at trykke på Bestil i navigationsbaren og sendes derved hen til customer.jsp hvor der er følgende to valgmuligheder. Den ene er “Bestil - pr. stk. 23.998,-” (customerCarportStandard.jsp) hvor det er tiltænkt at kunden skal kunne bestille carporte på standardmål. Denne mulighed er dog ikke implementeret da product owner ikke ønskede dette på nuværende tidspunkt.

Klikker brugeren derimod på “Byg selv” (customerCarportCustomize) sendes brugeren en til en side hvor det gør det muligt for brugeren at bestille en carport med specielmål samt tagtype- og farve. Efter kunden har tastes alle mål, valg, oplysninger og bemærkninger har kunden mulighed for at trykke på send. Derved sendes kunden hen til en kvitteringsside (customerReceipt.jsp) hvor kunden modtager en kvittering på at deres forespørgelse er gennemført. Denne kvittering kan også printes hvis kunden vil det.

Går vi tilbage til udgangspunket index.jsp og lader nu en Fog medarbejder logge ind på hjemmesiden, bliver Fog medarbejderen sendt hen til adminpage.jsp hvor der gives en række valgmuligheder. Navigationsbaren ændre sig og indeholder nu Forespørgelse Oversigt samt en drop down menu. Hvis Fog medarbejderen vil kigge, rette eller slette en bestemt forespørgelse kan han trykke på knapperne “Åben” for at se og rette eller Fjern for at fjerne en forespørgelse.

Trykker Fog medarbejderen på “Åben”, sendes medarbejderen hen til adminQuoteView.jsp hvor der kan rettes og slettes i de informationer kunden har indtastet. Derudover har Fog medarbejderen også mulighed for at trykke på knappen “Se tegning & stykliste”, som viser tegningen af den pågældenden carport. Her sendes medarbejderen hen til adminDrawing.jsp hvor tegningerne og stykliste bliver genereret og vist.

Hvis Fog medarbejderen vil se samme kvittering som kunden kan han/hun trykke på “Se tilbud” og bliver dermed sendt til customerReceipt.jsp.

Til sidst har Fog medarbejderen mulighed for at gøre brug af drop down menuen hvor der kan rettes følgende ting: Måleenheder (adminMeasurementUnits.jsp), Vareliste (ItemList.jsp), Rafterafstand (adminRafterSpacing.jsp), Tagbeklædning (adminRoof.jsp), Taghældning (adminRoofPitch) og Standardmål (adminStandardDimensions.jsp).

## 3.3 Klassediagram

Klassediagrammet er taget med i rapporten for at skabe et overblik over systemets klasser, alle deres metoder og packages. Klassediagrammet er blevet udarbejdet i plantUML og er vedlagt som bilag.

Klasserne er inddelt i forskellige packages i forskellige lag af systemet. Grunden til at vi har inddelt i underpackages inde i de 3 lag (DBAAccess, FunctionLayer & PresentationLayer) af command pattern skabelonen, er for at skabe en struktureret, overskuelig opbygning og læsning af koden. Koden er som sagt designet efter en command patteren designskabelon der gør brug af *servlets*, *frontcontroller* samt en *command* klasse. Dette er visualiseret ved hjælp af et klassediagram. Dette ses blandt andet ved de røde pile på klassediagrammet, der binder de enkelte klasser sammen og viser deres afhængighed af hinanden.

# 4 Særlige forhold

*Hvilke informationer gemmes i session, Hvordan håndterer man exceptions, Hvordan man på har valgt at lave brugerindput validering, Hvordan man har valgt at lave sikkerhed i forbindelse med login, Hvilke brugertyper der er valgt i databasen, og hvodan de er brugt i jdbc, andre elementer i Fog projektet kan det være: Tegning, Stykliste beregner*

Gruppen har fra starten arbejdet med følgende forudsætning for øje:

*”Vi vil ikke levere mindre funktionalitet, end Fog har i dag”.*

Derfor blev det tidligt besluttet, at komme så tæt på de rigtige (Og direkte anvendelige) beregninger fra start. Dette medførte et forholdsvis stort research-arbejde og mundede ud i konkrete udregninger programmet skal foretage når carporten skal beregnes.

Gevinsten heraf er at alle generede SVG-tegninger er dynamiske, ligesom størstedelen af carportens dimensioner og stykliste tager højde for statikberegninger samt brugerens ønsker.

De få ting som ikke er dynamiske er baseret på realistiske antagelser fra Fog’s eget produktkatalog. Det drejer sig bl.a. om størrelser på forskellige tagsten og trapezplader. Disse er undladt da andre beregninger og opgaver blev prioriteret frem for disse.

For Carport kalkulationen dokumenteres udregningerne samt et eksempel derfor, i dette afsnit. Udvalgte sektioner af den tilhørende, funktionelle kode vil blive gennemgået i afsnit omkring udvalgte kodeeksempler.

Figur 1 - Noter og brainstorming ifbm. udregninger

**Om dimensionering**

Når man bygger en carport skal de bærende søjler samt tagspærene dimensioneres. Parameteret for dimensionering udgøres af tagkonstruktionens vægt, ofte regnet i kilo newton [kN].

En tommelfingerregel[[1]](#footnote-1) siger at:

* Et let tag vejer maks. 25 kg / m2
* Et tungt tag vejer maks. 45 kg / m2

Typen af tagkonstruktionen falder derfor indenfor kategorien ”Let” eller ”Tung”.

For Fogs udbudsmateriale er tilfældet typisk, at en carport med ”Rejst” tag falder i kategorien ”Tung” og at en carport med et fladt tag falder i kategorien ”Let”.

Dimensionering af søjler

Afhængigt af, om taget er let eller tungt anvender man desuden forskellige søjler og spær dimensioner.

For søjler kan man slå op i en dimensioneringstabel[[2]](#footnote-2) for at finde passende dimensioner til sit byggeprojekt.

Til dette projekt anvendes følgende søjler:

* Til let tag: 100x100x25000 mm
* Til tungt tag: 125x125x25000 mm

Alle søjler er 2,5 m høje, nøjagtigt ligesom fogs udbud. Søjlernes dimension betyder desuden, at de er dimensioneret til hhv. let og tungt tag, inkl. en sikkerhedstolerance.

Standardstørrelserne Fog udbyder er magen til de størrelser programmet kalkulerer med. Det betyder at en Carport ikke kan blive større end maksimalt 750 x 780 cm – Denne størrelse kan med de, dimensionerede søjler understøttes af fire stolper. Af samme grund er programmet bygget op således, at en carport altid vil bestå af fire søjler medmindre, at kunden har tilvalgt skur – I så fald vil carporten bestå af otte søjler.

Dimensionering af spær

At dimensionere spær er en proces der kræver flere trin.

Afhængigt af tages hældning udregnes spærets længde vha. simpel trigonometri, som afhænger af kundens specifikationer. Disse udregninger resulterer i en spærlængde. Spærafstanden skal herefter beregnes for at sikre et holdbart tag. Til dette ganges spærets længde med en hældningsfaktor hvorefter resultatet af dette anvendes til at slå op i en tabel som både angiver den nøvendige spærafstand samt spærets dimensioner. Heri skal den korteste spærafstand vælges, uden at være lavere tabelopslaget. Et eksempel på anvendelsen af tabellerne findes sidst i dette afsnit.

De tre tabeller[[3]](#footnote-3) er angivet nedenfor:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Spærafstand ved spærvidde (Let tagkonstruktion)** | | | | |
| **Bjælkedimension i mm** | **0,4 m** | **0,6 m** | **0,8 m** | **1,0 m** | **1,2 m** |
| 45 x 120 | 2,81 | 2,48 | 2,26 | 2,10 | 1,98 |
| 45 x 195 | 4,52 | 4,02 | 3,68 | 3,44 | 3,24 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Spærafstand ved spærvidde (Tung tagkonstruktion)** | | | | |
| **Bjælkedimension i mm** | **0,4 m** | **0,6 m** | **0,8 m** | **1,0 m** | **1,2 m** |
| 45 x 120 | 2,43 | 2,13 | 1,93 | 1,79 | 1,68 |
| 45 x 195 | 3,94 | 3,48 | 3,18 | 2,95 | 2,78 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Gangefaktor for spærafstand** | | | | | | | |
| **Hældning** | **<=15°** | **20°** | **25°** | **30°** | **35°** | **40°** | **45°** |
| Faktor | 1 | 0,97 | 0,94 | 0,89 | 0,84 | 0,79 | 0,72 |

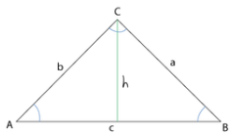
Med antallet af spær og søjler, er de mest komplicerede udregninger nu foretaget, og programmet kan udregne resten. Specifikt udregner programmet følgende:

|  |  |  |  |
| --- | --- | --- | --- |
| **Vare**  (Vareteksten er simplificeret) | **Udregningsmetode**  **(Dimensioner)** | **Udregningsmetode**  **(Stk. antal)** | **Tilhører**  **(Rejst, fladt, skur)** |
| Diverse beslag til hele carporten | Antaget | Antaget | Alle |
| 38x73 taglægte T1 | Udregnet | Udregnet | Alle |
| 25x200 Bræt til stern trykimpr. | Udregnet | Udregnet | Alle |
| 45x195 Rem ubh. | Udregnet | Udregnet | Alle |
| Skruepakke til universalbeslag + toplægte | Antaget | Antaget | Alle |
| 45x120 Tagspær trykimpr. | Udregnet | Udregnet | Alle |
| 100x100 Stolpe trykimpr. | Udregnet | Udregnet | Alle |
| 109 x 240 tagplade | Udregnet | Udregnet | Fladt tag |
| Skruer til tagplader | Antaget | Antaget | Fladt tag |
| Skruepakke til taglægter | Antaget | Antaget | Rejst tag |
| 20,4 x 23,6 tagsten | Udregnet | Udregnet | Rejst tag |
| Komplet dør sæt til skuret | Antaget | Antaget | Skur |
| Skruepakke til mont. af bræt ved beklædning | Antaget | Antaget | Skur |
| Skruepakke til mont. af bræt ved beklædning | Antaget | Antaget | Skur |
| 45x95 mm ubh. Løsholte | Udregnet | Udregnet | Skur |
| 19x100 Beklædning af skur | Udregnet | Udregnet | Skur |

Ovenstående linjer anvendes til at generere stykliste, pris samt dynamiske SVG-tegninger.

Betragter man carportens tag forfra vil man se at den er udformet som en ligebenet trekant.

Derfor kan længden af spær regnes ud ved at anvende formler der gælder for en ligebenet trekant.

Kunden oplyser selv følgende mål:

* Tagets hældning (Vinkel) illustreret nedenfor i vinkel “A” og “B”.
* Carportens, og dermed tagets, bredde illustreret nedenfor med “c”.

Programmet skal derfor udregne:

Figur 2- Ligebenet trekant

* Længden på spær illustreret nedenfor med hhv. “b” og “a”. Vi ved at trekanten er ligebenet hvorfor “a” må svare til “b”.
* Højden på den samlede konstruktion illustreret nedenfor med “h”.

Tagets kip (Vinkel C) udregnes:

Spærets længde:

Tagkonstruktionens højde:

Med disse formler kan vi nu udregne det væsentligste.

**Udregnet eksempel**

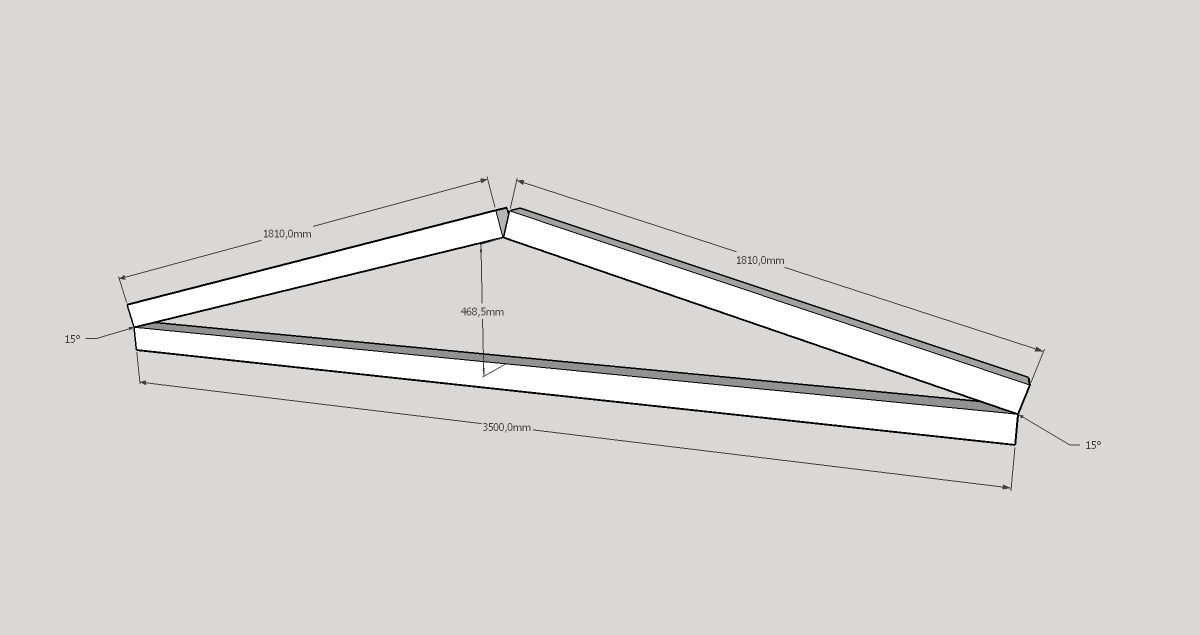
Lad os antage at kunden bestiller en carport med målene 350 x 510 cm, rejst tag på 15° samt et skur på 270 x 240 cm.

Udregning af tagets kip:

Udregning af spærets længde:

Udregning af tagkonstruktionens højde:

Eksemplet kan bl.a. verificeres ved at tegne spærkonstruktionen i målfast 3D



Figur 3 - Målfast 3D tegning

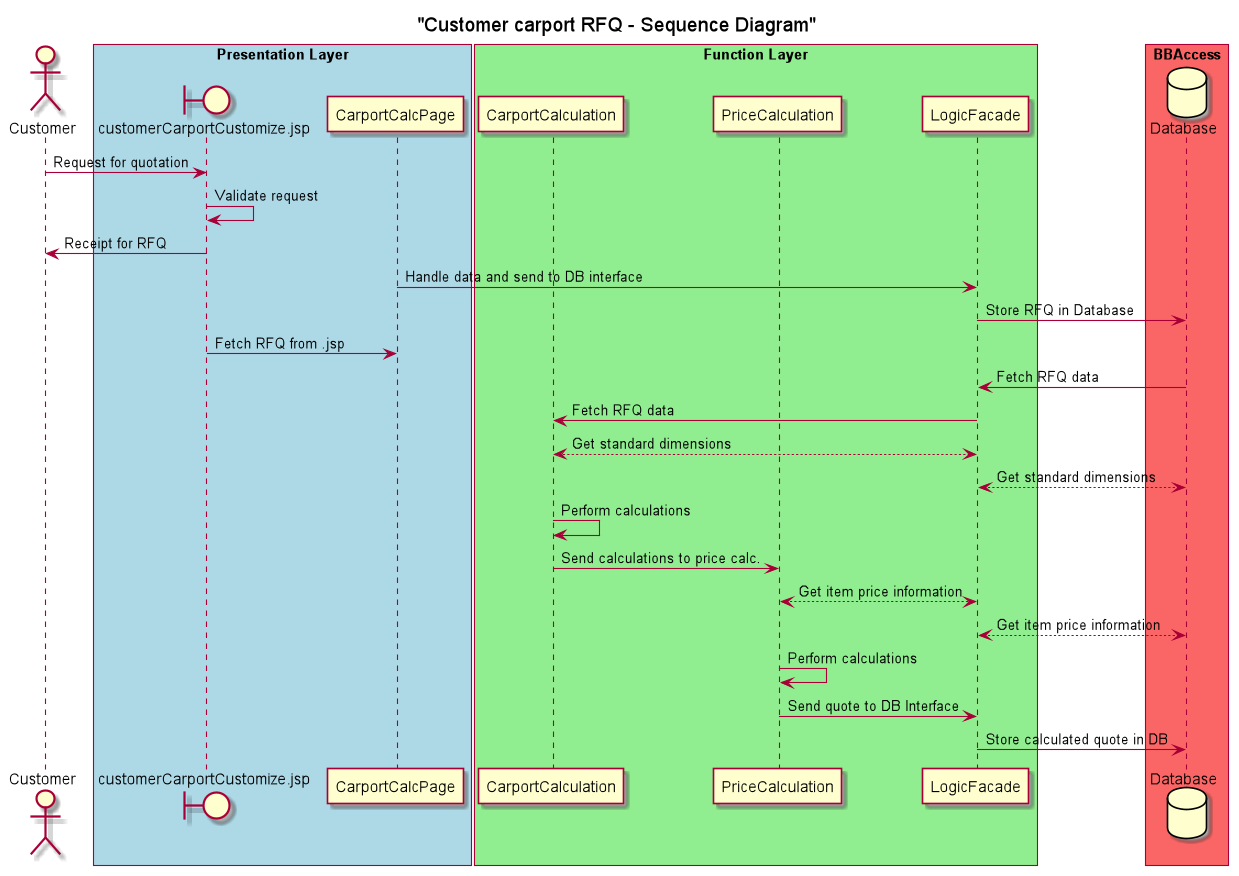
De øvrige mål for carporten er forholdsvis simple, sammenlignet med de ovennævnte udregninger hvorfor de ikke gennemgås i rapporten.

# 5 Udvalgte kodeeksempler

*Det kommer til at virke særligt overbevisende hvis den kode man vælger ud indgår som led i et af sekvensdiagrammerne.*

## 5.1 Carport Calculation

Teorien bag ”CarportCalculation” klassen blev gennemgået i afsnit 8.1 og vil blive yderligere forklaret i, kodemæssigt i dette afsnit.

BESKRIV DIAGRAMMET HER

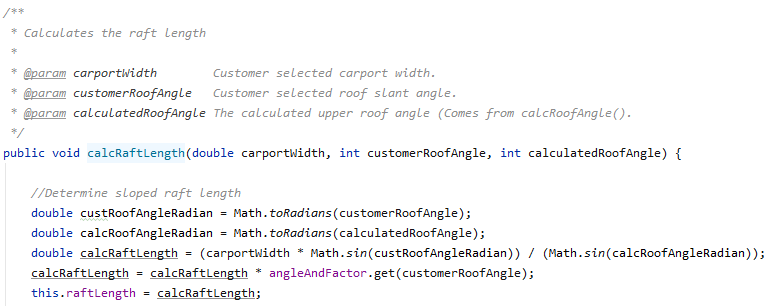
**Standardmål**

Tabellerne i afsnit 8.1 med standardmålene er oprettet som tabeller i databasen, således at programmet kan slå op i disse for at generere en korrekt, dynamisk beregning. Der er bl.a. oprettet én tabel til spærafstande og en til ”Vinkelfaktoren” (Også gennemgået i afsnit 8.1).

Disse tabeller blev oprettet med følgende SQL:

|  |  |
| --- | --- |
| Figur 4 - tabel ”rafter\_spacing” | Figur 5 – Tabel ”roof\_pitch” |
| Figur 6 - Spærafstand tabel data (Udsnit) | Figur 7 - Vinkelfaktor tabel data |

Når systemet har udregnet spærlængde og determineret om der er tale om et ”Let” eller ”Tungt” tag, dikterer det byggetekniske spærafstanden. Denne undersøges ved at gange spærlængden med vinkelfaktoren som beskrevet i afsnit 8 og lave et opslag i tabellen ”. Programmet skal finde en *rafter\_length* som kommer tættest på den udregnede spærlængde, uden at være under denne.



Figur 8 - metode "calcRaftLength"

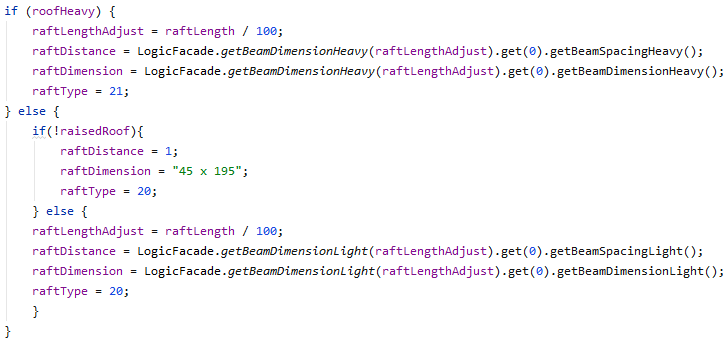
Metoden *calcRaftLength* tager tre parametre. Carportens længde, samt hhv. kundens hældningsvinkel og den udregnede vinkel i tagets kip.

For at anvende sinus i Javas *Math* klasse skal vinklerne konverteres til radianer.

Herefter ganges spærlængden med hældningsfaktoren som er hentet fra databasen og gemt i Javas Collection class, *”Map”*.

Ved at ”Matche” kundens hældningsvinkel med tilsvarende i dette map, findes vinkelfaktoren nemt.

Programmet skal nu finde spærafstanden i databasetabellen *rafter\_spacing­*.



Figur 9 - determinering af spærafstand

Programmet har undersøgt om der er tale om en tung eller let tagkonstruktion på dette tidspunkt.

I Fogs tilfælde, vil udbuddet af carporte med flade tage, altid udgøre lette tagkonstruktioner.

**Fladt tag**

Er der tale om et let, fladt tag er det givet, at spærafstanden svarer til carportens bredde.

Spærafstanden sættes til 1m og spæret dimensioneres til et standardmål der passer til samme.

*raftType* svarer direkte til spærets varenummer i databasen

**Rejst tag**

Er der tale om et rejst tag skal programmet kigge i databasen efter den rigtige spærafstand.

Først divideres spærlængden med 100 for at konvertere den til meter så den svarer til databaseformatet. Herefter laves opslaget igennem *LogicFacade* som indeholder den SQL der håndterer søgningen, og returneringen fra databasen. Der kaldes to metoder som returnerer hhv. spærets dimension og spærafstanden.



Figur 10 - SQL statement til spær

I ovenstående figur ses et SQL opslag i kategorien ”Let” (Metoden er ens for kategorien ”Tung”).

Metoden søger, som tidligere beskrevet, efter den spærlængde som kommer tættest på den udregnede, uden at være under. Resultatet, i form af spærdimension og spærlængde gemmes i en *List* som returneres til CarportCalculation klassen igennem LogicFacade.

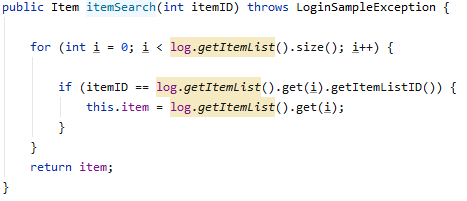
På denne måde returneres den rette spærafstand altid.

## 5.2 Price Calculation

Priskalkulationsklassen foretager forholdsvis simple udregninger. Carportkalkulationen har på forhånd udregnet antal og / eller mængder af de forskellige, nødvendige materialer. Klassen ganger derfor, afhængigt af varen, antal, mængde og størrelse sammen med pris per enhed (Eksempelvis 14,95 kr. pr. meter) for alle varer hvilket resulterer i en totalpris. Der regnes moms på totalprisen og efterfølgende en dækningsgrad, fastsat af Fog medarbejderen.

Samtlige priser hentes altid fra databasen og som nemt opdateres i programmets tilhørende CRUD system.

Et væsentligt aspekt ved priskalkulationen er måden hvorpå varen findes i databasen for efterfølgende at blive anvendt. Til dette er der skrevet en ”Søgefunktion”.



Figur 11 - PriceCalculator search

Metoden modtager et varenummer, som er fastsat i carportkalkulationsklassen.

Den søger herefter i en liste som består af samtlige varenumre, hentet via LogicFacade i databasen.

Når varenummeret matcher returneres hele varen, *item*, fra listen som indeholder informationer om pris, navn mm. Varen kan nu bruges til udregninger.

# 6 Status på implementation

*Dette afsnit skal liste hvor langt man er nået med implementationen. Typiske ting man kan have sprunget over er:*

*Man har ikke nået at lave alle de jsp sider man har med i navigationsdiagrammet.*

*Man har ikke nået at lave alle CRUD metoderne til alle tabellerne*

*Man har ikke fået stylet sine sider*

*Man har fundet en fejl “i sidste øjeblik”, men har ikke haft tid til at rette det. - F.eks. at man har brugt session forkert sådan at man på en af siderne kan komme ind uden at være logget in.*

*tests der fejler på afleveringstidspunkt*

*… andre mangler*

# 7 Test (Husk word fil fra Jon her og indrag v-model)

|  |  |
| --- | --- |
| **Gruppenavn** |  |
|  |  |
| **Spørgsmål** | **Teststrategi** |
| Virker vores beregner? | Unit testing: White-box |
| Passer antallet af stolper? | Unit testing: White-box |
| Passer antallet af spær? | Unit testing: White-box |
| Andre beregninger? | Unit testing: White-box (Alle relevante metoder I beregner) |
| SVG-motor |  |
| Kan jeg tegne et rektangel med en vis størrelse? Virker vores metode? | Unit testing: White-box |
| Kan jeg tegne en pil som f.eks. er 300px lang? | Unit testing: White-box |
| Datamappere |  |
| Er der overhovedet hul igennem til databasen? | Integrationstest: Black box |
| Kan jeg indsætte en ny bruger? | N/A |
| Kan jeg hente et produkt / liste af produkter? | Integrationstest: Black box |
| Bliver en ordres status gemt i databasen når man opdaterer den på jsp siden? | Integrationstest: Black box |
| Hvad med vores brugergrænseflade? |  |
| Kan en kunde finde ud af at bestille en carport? Hvordan afgør vi det? | Testet på potentielle kunder |
| Kan de ansatte hos Fog finde ud af at bruge systemet? | Testet på ”Ikke-softwarekyndige”. |
| Hvordan håndterer vi input validering? | jQuery på front-end. Diverse validering på back-end. |
| Generelt |  |
| Er vores kode skrevet, så vi automatisk kan teste den? | Ingen selenium tests, men god struktur og base for unit- og integrationstests. |
| Hvordan sikrer vi os en ensartet kodekvalitet? | God dialog og daglig præsentation af gårsdagens arbejde. |
| Hvordan sikrer vi os at vi ikke tjekker fejlagtig kode ind på vores master- eller developerbranch i GitHub? | Ved altid at bruge *git pull* inden et commit og den daglige dialog. |

*Der skal være lavet test. Du kan dokumentere tests ved at beskrive i tabel form:*

*Hvilke klasser er testet, Hvilke metoder der er testet, Dækningsgrad af dine tests for de valgte metoder og klasser*

*Desuden kan du beskrive hvordan i systematisk har arbejdet med at teste koden før den er blevet gjort til en del af master branch.*

# 8 Process

Der skal være et afsnit hvor I beskriver jeres arbejsprocess i projekt perioden. Der skal dels være et faktuelt afsnit og et reflektions afsnit.

## 8.1 Arbejdsprocessen faktuelt

**Hvordan i afholdt jeres daglige standup møder.**

**Hvornår i holdt retrospectives.**

Under projektet har vi arbejdet med 5 sprints. Vi valgte at uddele Scrum Master titlen således at vi alle kunne få erfaring med dette. Udover scrum-masterens normale ansvar blev han også tildelt ansvaret for at føre en log over ”daily scrum” møder. De daglige scrum møder tog alt mellem 15 minutter og op til flere timer. Det skal dog tilføjes at de lange møder, oftest var grundet at de endte ud i et samarbejde om en specifik problemløsning.

Under de daglige møder blev statussen på userstories, task, eventuelle rettelser, problemer og tilføjelser gennemgået samt uddelegeret hvis dette var nødvendigt.

Alle userstories, tasks og vores daily scrum log kan findes i appendix # og appendix #

### Sprint 1

Scrum Master:Mick

Sprint 1 var fokuseret på at få forespørgsels delen af projektet op at køre.

Da dette var en af de grundlæggende ting i projektet, hvis funktionalitet skulle benyttes til mange af de efterfølgende sprint, var det for os en vigtig ting at få op at køre som det første.

Sprintet indeholdt følgende userstories:

**#21 Kunde: Forespørgsel på spec. carport (basic)**

**#22 Kunde: Forespørgsel på spec. carport (basic m. tag)**

**#23 Kunde: Forespørgsel på spec. carport (basic, tag + skur)**

**#34 Kunde: Forespørgsel på materialetyper**

### Sprint 2:

Scrum Master: Jean-Paul

Sprint 2 var fokuseret omkring Fogs mulighed for at se en kundes forespørgsel, samt få systemet til at kunne genere en standard tegning af en carport.

Sprintet indeholdt følgende userstories:

**#5 Fog: Se forespørgsler**

**#15 Fog: Generering af tegning**

Under sprint 2 havde vi desuden både et scrum review samt et scrum planning møde med Product owner, følgende er taget fra vores daily scrum log, som set i apendex #:

*30-04-2020 - Scrum review med P.O*

Vi havde lidt misforståelse med product owner, som ikke troede vi havde fået lavet det som var planlagt, men det viste sig, at vi var længere end antaget og kunne fremvise en tegning af vores carport. Gruppen mener vi kan nå i mål med de fleste user stories, men vil mødes igen til aften for at få et bedre overblik over eventuelle mangler.

”Se forespørgsler” og ”slet forespørgsler” er de to user stories, som mangler mest arbejde og vi kan blive nødsaget til at skubbe dem til næste sprint. Alt afhænger af hvad gruppen når at få kodet i løbet af dagen.

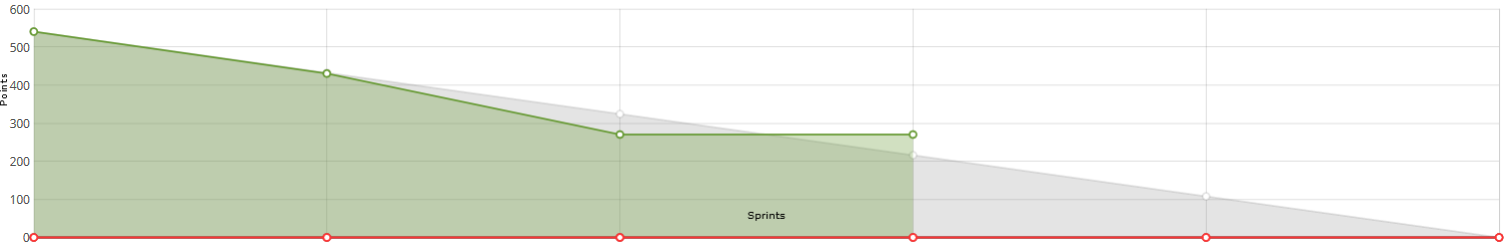
Der skal desuden sættes point på de næste user stories. Dette vil blive fremlagt til scrum planning mødet med vores P.O i morgen.

*30-04-2020 - Scrum planning med P.O*

Andet sprint er gennemført og vi nåede i mål med det meste.

Vi havde en User story som lød på ”*Som Fog-medarbejder vil jeg have at systemet beregner en pris således at jeg kan klargøre et tilbud til kunden.”* Denne case måtte vi sande, at vi ikke kunne nå i mål med, da den var større end først antaget. Teamet har derfor flyttet den til næste uges sprint.

Vi kan derfor se et dyk i vores performance graf, men vil være stigende når vi får færdiggjort næste uges sprint.



Sprint 3 vil bestå af 210 point, i stedet for de 180 point, da vi som sagt har flyttet en user story over fra sprint 2. Efter gennemgang af user stories, task og en hel del mere erfaring med estimater, er vi, i teamet, enige om at vi kan nå i mål med Sprint 3. Det skal dog noteres at flere task kan opstå i det vi dykker ned i koden.

### Sprint 3:

Scrum Master: Alexander

Sprint 3 var fokuseret på et lidt bredere aspect. Et fokus var på at lave SVG tegningerne dynamiske, således at de kunne vise en tegning der passede til den enkelte forespørgsel.

Derudover var der fokus på at få yderligere håndtering af forespørgslerne til at virke, herunder at kunne slette dem, samt give en kvittering når en kunde sender en forespørgsel videre til Fog.

Til sidst var der fokus på, at få en validering af mål og valgmuligheder inkorporeret i drop Down menuerne. Således at en kunde ikke kan vælge en carport, hvis mål skaber en carport der ikke kan realiseres.

Sprintet indeholdt følgende userstories:

**#80 Dynamisk view a SVG**

**#82 Fog: Se forespørgsel**

**#8 Fog: Slette forespørgsler**

**#14 Validering af spec. løsninger**

**#109 Kunde: Kvittering ved forespørgsel**

**Sprint 4:**

Scrum Master: Morten

Sprint 4 var fokuseret på at CRUD. Herunder administrering af varekataloget samt forespørgsler, således at Fog kan redigere og tilføje varer, samt redigere i eksisterende forespørgsler direkte fra hjemmesiden af.

Sprintet indeholdt følgende userstories:

**#11 Fog: Administrering af varekatalog**

**#7 Fog: spec. carport (Redigering)**

### Sprint 5:

Scrum Master: Per

Sprint 5 var fokuseret på at lukket løse ender, fixe diverse fejl og få den sidste funktionalitet op at køre.

#171 Slutspurt var meget bred, og indeholdt mange vidt forskellige tasks, der kan ses i sprint back loggen i appendix #.

Sprint 5 gik også ud på at få de sidste dele af forespørgselsdelen af projektet til at virke, samt opdatere funktionaliteten og designet til administreringen af varekataloget.

Sidste del af sprintet var at få tilbudsprisen samt dækningsgrad til at fungere, således at Fog kan sende et tilbud til kunden.

Sprintet indeholdt følgende userstories:

**#171 Slutspurt**

**#6 Fog: spec. carporte (Modtaget tilbud)**

**#148 Fog: CRUD**

**#59 Fog: beregning af pris**

### 8.1.1 SCRUM Userstories

*Dette afsnit skal beskrive de user-stories der er aftalt med product-owner. Det er vigtigt at I har en håndfuld userstories som er lavet fuldt ud, dvs:*

*der er beskrevet how-to-demo*

*der er brudt ned i tasks*

*der er lavet et estimat*

*Den fulde produkt backlog kan ligge som appendix.*

## 8.2 Arbejdsprocessen reflekteret

Dette afsnit skal beskrive jeres overvejelser over hvilke dele der har fungeret godt og hvilke dele der måske er faldet lidt på gulvet. I kan f.eks. beskrive:

Om scrummaster rollen fungerede, hvilke problemer I så i den, og hvad I gjorde for at rette op på det.

Hvad der var de væsentligste emner på jeres retrospektiv møder

Om I havde problemer med at nedbryde user stories i tasks

Om I var spot-on med jeres estimeringer

Om der var problemer med vejledningen og PO møderne

Hvor langt inde i processen I fandt en rytme der var produktiv

Andre elementer der har at gøre med at forsøge at arbejde i et scrum team

# **9 Konklusion**

En konklusion er en sammenfatning af de resultater, man har fundet i forbindelse med udarbejdelse af rapporten. Dette skal dog ikke forstås som et resumé af, hvad man har skrevet, men som et svar på de spørgsmål man stillede i sin problemformulering. Det er meget vigtigt, at der bliver svaret på de spørgsmål, der blev stillet i problemformuleringen, men konklusionen skal kun indeholde de vigtigste pointer og påstande. Der må ikke inddrages nyt stof i en konklusion.

# **10 Perspektivering**

Ønsker man at inddrage andre perspektiver, kan man lave et afsnit efter konklusionen, der hedder ”Perspektivering”. Dette afsnit er ikke obligatorisk, men kan bruges til at beskrive, hvilket videre arbejde rapporten danner grundlag for, hvilke syn der kunne være interessante at inddrage i andre sammenhænge, hvilke paralleller der kan drages til andre områder, osv. Perspektiveringen er en diskussion af konsekvenserne af konklusionen i et større perspektiv.

1. <http://www.icopal.dk/~/media/UploadFolder/Products/DK/IcopalDK/ProductLibrary/Bitumen%20Membranes/SBS%20TopBase/6001altomtag0612.pdf> [↑](#footnote-ref-1)
2. <http://flexwood.dk/wp-content/uploads/2012/03/Flexwood-beregningstabel.pdf> [↑](#footnote-ref-2)
3. <https://www.ringstedspaer.dk/konstruktioner/bjaelkespaer> [↑](#footnote-ref-3)