

Semesterprojekt - FOG carport



CPHbusiness Datamatiker Uddannelsen / 2. semester / Hold B / Gruppe 4

Afleveres: d. 24/5 - 2024

Udarbejdet af:

Sander Marcus Christensen / cph-sc296@cphbusiness.dk / Github:

sanderMarcusChristensen

Marcus P Hjorth Rasmussen / cph-mr634@cphbusiness.dk / Github: MarcusHjorth

Mateen Jan Rafiq / cph-mr645@cphbusiness.dk / Github: MateenCPH

Indholdsfortegnelse

1. Indledning	3
2. Baggrund	4
3. Virksomhed og Forretningsforståelse	4
3.1 Interessentanalyse	4
3.2 Risikoanalyse	6
4. Teknologivalg	8
5. Aktivitetsdiagram AS/IS	9
5.1 TO/BE (Swimlane)	10
6. User Stories and A/C	11
7. Domæne Diagram	13
8. ERD Diagram	14
9. Første klasse diagram	15
10. Navigationsdiagram	16
11. Figma - Mockups	17
12. Valg af Arkitektur	18
13. Særlige forhold	20
14. Udvalgte kodeeksempler	22
14.1 Fragments og header	22
14.2 Carport SVG	23
15. Status på implementering	24
16. Kvalitetssikring	25
16.1 Automatiserede tests	25
16.2 User Acceptance Test	26
17. Proces	29
17.1 Arbejdsprocessen faktuelt	30
17.2 Arbejdsprocessen reflekteret	31
1. Referenceliste	32
2. Nice to have User-stories	33
3. Dagbog	34

1. Indledning

Dette er vores eksamsprojekt på Datamatikeruddannelsen, 2. semester på CPHbusiness. Opgaven er en gruppeopgave, der løses i grupper af 2-5 personer. Vi studerende har fået til opgave at forbedre/løse nogle af de problemer, som byggemarkedet Johannes FOG havde med deres IT-system. Til opgaven har vi fået en "interview"-video, hvor vi møder vores kunde "Martin," som fortæller og forklarer om deres problemer og ønskede løsninger til deres nye IT-system. Selvom dette var et problem, som FOG havde tilbage i 2016 og nu er blevet løst, er det stadig en god udfordring for os som studerende, og det er meget virkelighedsnært i forhold til problemstillinger, man kan møde som færdiguddannet datamatiker.

Vi studerende skal selv gennemgå interview-videoen, analysere hvad der skal laves, og hvordan det skal laves. FOG har stort fokus på deres "byg-selv carporte," hvor en kunde skal kunne indsætte sine egne mål og få tilsendt en tilsvarende carport. På grund af den skarpe deadline vil gruppen tage FOG's fokus som udgangspunkt og derfor kun lave "byg-selv"-delen af systemet. Alle de normale køb af standardcarporte som FOG tilbyder, er ikke med i denne version. Alle andre køb samt standardcarporte vil blive løst ved at linke produktet til FOG's nuværende hjemmeside.

Rapporten skrives af datamatikerstuderende til datamatikerstuderende. Rapporten henvender sig til personer med lignende erfaring og viden. Læserne skal kunne forstå vores tanker bag projektet og forstå, hvorfor vi har truffet de beslutninger vi har.

2. Baggrund

FOG er et trælast og byggecenter, der sælger alt fra carporte til hegner. FOG ønsker nu en bedre webløsning, der kan gøre det muligt og nemmere for deres kunder at selv bestille en carport ud fra deres egne mål, og derved lette presset for deres sælgere.

3. Virksomhed og Forretningsforståelse

3.1 Interessentanalyse

Med udgangspunkt i videoen "På besøg hos kunden" og yderligere brainstorm har vi udarbejdet en interessentanalyse, der har til formål at stå klar med en masse input til, hvordan vi håndterer vores vigtigste interessenter, hvis/når der skulle dukke nogle bump op på vejen. Se figur 3.1.

Eksterne interessenter:

Lagermedarbejderne hos FOG har ikke nødvendigvis særlig stor indflydelse på projektet, men de kan skabe en konflikt, da vi ændrer i IT-systemet, som de også gør brug af. På den anden side kan de også bidrage positivt ved at komme med forbedringsforslag.

Pressen spiller en vigtig rolle i analysen, da de kan påvirke projektets offentlige opfattelse og videreforsmide information til en bred målgruppe. Dette kan både være positivt og negativt.

Kunderne er ikke en del af selve organisationen eller projektet, men de spiller en stor rolle i, hvor godt projektet klarer sig. Deres tilfredshed, behov og feedback er meget vigtige for, om projektet bliver en succes.

Gidsler:

Sælgere hos FOG har en stor medvirken i projektet. De er direkte påvirket af det nye IT-system, da det vil ændre deres arbejdsproces. Deres daglige brug af systemet og erfaring gør deres feedback meget værdifuldt, hvilket hjælper med at gøre systemet mere brugervenligt og bedre til at opfylde praktiske behov. Man kunne argumentere for at placere dem i kategorien "Ressourcepersoner", men da vi allerede har direkte kontakt til salgschefen Martin, betyder det, at han har ansvaret for at opsamle deres feedback og videreforsmide det til os.

IT-ansatte i FOG er nødvendige for gennemførsel af projektet. De besidder den nødvendige viden, der er afgørende for implementeringen samt vedligeholdelse af projektet. De spiller en central rolle i at identificere og løse tekniske problemer, der kan opstå under og efter implementeringen af systemet.

Ressource-personer:

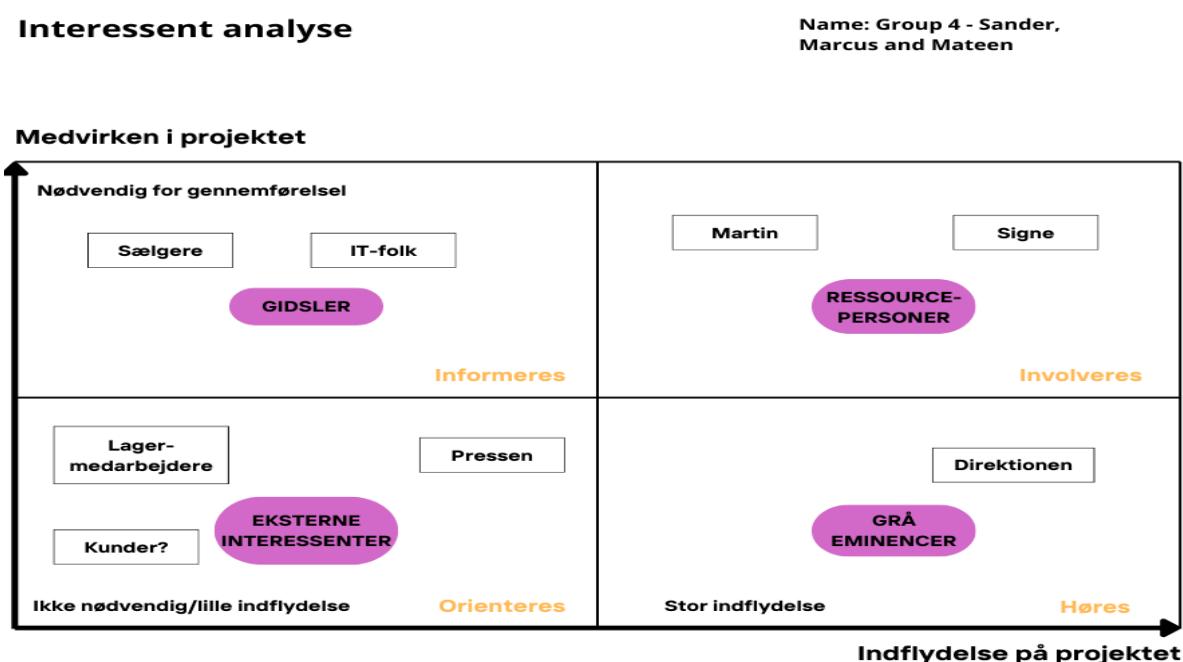
Martin er salgschefen for FOG. Han spiller en aktiv og vigtig rolle i projektet, da han fungerer som det primære bindeled mellem salgsteamet og os. Han har ansvaret for at indsamle og videreforsmilde feedback til os, hvilket sikrer, at deres behov bliver adresseret. Derudover besidder han dybdegående viden om salgsteamets daglige arbejdsgange og udfordringer, hvilket giver os værdifuld indsigt i at udvikle et brugervenligt og effektivt system.

Signe er vores primære kontaktperson ift. vejledning af projektet. Som vores lærer og vejleder besidder hun afgørende viden inden for området, hvilket gør hende til en værdifuld kilde til rådgivning. Hun har hjulpet med at facilitere diskussioner og beslutningsprocesser, hvilket har givet stor medvind til projektets fremgang.

Grå eminencer:

Direktionen for FOG har ikke nogen medvirken i projektet, men de har en stor indflydelse på en mere indirekte måde uden nødvendigvis at være direkte involveret i projektets daglige drift. De træffer strategiske beslutninger, der kan påvirke projektets rammer, ressourcer og retning. Deres beslutninger kan have afgørende betydning for projektets fremgang.

(Figur 3.1 billede af interesseanalyse



3.2 Risikoanalyse

Figur 3.2 repræsenterer vores risikoanalyse. Vi har vurderet potentielle risici og mulige konsekvenser af hændelser, som har indflydelse på vores projekt.

Sygdom i udviklingsteamet er ret sandsynligt, men det tolereres. Risikoniveauet er medium, da vi mister to hænder, og derfor bliver vi nødt til at have en buffer for arbejdsmængden, således at vi optimerer arbejdsmængden til 2,7 personer i stedet for 3.

FOG hacking er usandsynligt, da de har en masse firewalls og god IT-sikkerhed. Vi ønsker ikke, at de bliver hacked, da det kan have forstyrrelser i forhold til kommunikation, men det er en lav risiko for os, da det ikke rører projektet. Man kunne forebygge det ved at øge sikkerheden under projektarbejdet.

Computer breakdown er en mulig risiko, der kan ødelægge en hel dags arbejde. Det er en høj risiko, der tolereres, hvilket gør risikovurderingen til medium. Vi skal have teamet til at lave flere commits dagligt og pushe noget oftere.

FOG Kommunikationsfejl med FOG er en uønsket risiko med højt risikoniveau, da det kan have store konsekvenser for vores projekt. Risikovurderingen ligger på høj, og derfor er det en god idé at tage fat i salgschefen Martin og etablere en god strategi/metode for kommunikation.

Commits fejl har et medium risikoniveau, da man altid kan hente en ældre version af programmet frem. Det tolereres, og sandsynligheden for, at det sker, er mulig. Vores team har været virkelig gode til at forebygge dette vha. flere commits og samtlige commit-møder.

Database fejl er en uønsket risiko, som har et højt risikoniveau, hvis det sker. Sandsynligheden for, at det sker, er mulig. Derfor er det vigtigt, at vi som team opdaterer vores SQL-scripts løbende, således at vi under alle omstændigheder har en database liggende i skriftform.

FOG konkurs er ret alvorligt i forbindelse med vores projekt, og derfor er risikoniveauet sat til ekstremt. Sandsynligheden for, at det sker, er meget tæt på usandsynligt, men det er stadig muligt. Det er ikke noget, som vi kan forebygge, men derfor er det stadig en vigtig risiko at have med.

Martin sygdom er uønsket, da han er en vigtig ressourceperson i vores interessentanalyse. Derfor kan vi som team forebygge dette ved at bede ham om at udnævne nogle

repræsentanter i form af sælgere, således at hvis uheldet hænder, at der er nogle til at tage over for ham.

Kommunikationsfejl internt er uacceptabelt med højt risikoniveau og mulig sandsynlighed.

Det er vigtigt at forebygge dette ved at opretholde klar og god kommunikation med bekræftelse. Opnåelse af dette kan sikre en klar arbejdssproces med maksimal effektivitet, hvilket øger chancerne for at ramme vores mål med projektet samt øger arbejdsmoralen i teamet.

Strømafbrydelse hos FOG er uønsket, men har et lavt risikoniveau, da FOG har backup-generatorer klar. Risikoen er usandsynlig, da vi ikke oplever strømafbrydelser så ofte her i Danmark. Samtidig er vi gode til at håndtere afbrydelserne.

Fejl i oplæring af sælgere til nyt system er en risiko, som tolereres og er mulig.

Risikovurderingen er sat til medium, da det har en mærkbar effekt, men ikke på kritiske punkter. Vi skal som projektteam sørge for at stå til rådighed, hvis de har brug for hjælp, samt sørge for at udgive gode og klare instrukser til brug af programmet.

Nedskæringer af ressourcer fra FOG er en usandsynlig risiko, som er uønsket.

Risikoniveauet er sat til at være ekstremt, da det første, de ville skære ned på, er eksterne projekter. Det kunne fx være direktionen, som vi i vores interessentanalyse har sat til at være en grå eminence - altså har de ikke nogen medvirken i projektet, men de har en stor indflydelse. Risikovurderingen er sat til at være medium. Derfor er det vigtigt, at vi skærer lidt ned på vores projekt og minimerer omkostningerne.

(figur 3.2 billede af Risikoanalysen)

Baseret på skabelon fra altomledelse.dk						
Risikoanalyse						
Semesterprojekt Gruppe 4 (Sander, Marcus og Mateen)						
Risk ID	Risiko	Alvor	Sandsynlighed	Risikoniveau	Plan for forebyggelse/afværgning	
1	Sygdom i udviklingsteamet	Tolereres	Sandsynligt	Medium	Buffer for arbejdsmængde	
2	FOG Hacking	Uønsket	Usandsynlig	Lav	Bedre sikkerhed fra deres side	
3	Computer breakdown	Tolereres	Muligt	Høj	Upload på github / mange commits	
4	Kommunikationsfejl m. FOG	Uønsket	Muligt	Høj	God og klar komunikation m. bekræftelse	
5	Commits fejl	Tolereres	Muligt	Medium	Flere commits / branch protection / commit-møder	
6	Database fejl	Uønsket	Muligt	Høj	Oprette et SQL Script og løbende opdater dette / Backups	
7	Fog konkurs	Uacceptabelt	Muligt	Ekstrem		
8	Martin sygdom	Uønsket	Sandsynligt	Medium	Anskaffe flere repræsentanter (sælgere)	
9	Kommunikationsfejl m. team	Uønsket	Muligt	Høj	God og klar komunikation m. bekræftelse	
10	Strømafbrydelse hos FOG	Uønsket	Usandsynlig	Lav	Gentagende backups	
11	Fejl i oplæring af sælgere til nyt system	Tolereres	Muligt	Medium	Gode og klare instrukser, hjælp til rådighed	
12	Nedskæringer af ressourcer fra FOG	Uønsket	Usandsynlig	Ekstrem	Skær ned på vores projekt, minimer arbejde + omkostninger	

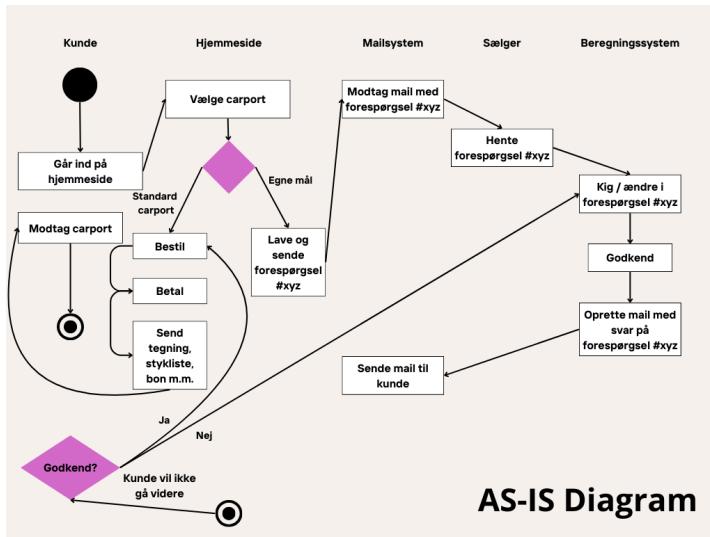
4. Teknologivalg

Som valg af teknologier bruger vi IntelliJ IDE (v.2023.2.2) som hovedprogram til kodning. Databasen med tabeller er oprettet i PostgreSQL, og de grafiske mockups er blevet lavet i Figma. Selve projektets flow er dokumenteret i denne rapport. Til løsning af projektet er der også blevet anvendt et Trello-board samt GitHub's egen portal til løbende projektstyring. Nedenstående tabeller giver et visuelt billede af alle teknologier, værktøjer, biblioteker og buildtools, som er blevet brugt i dette projekt.

Tools	Java libraries	Buildtools
<ul style="list-style-type: none">• IntelliJ IDEA (2023.2.2)• PostgreSQL (42.7.2)• pgAdmin 4 version: 8.4• Github desktop version: 3.3.8 (x64)• Figma desktop• digitalocean droplet (ip: 209.38.202.233)	<ul style="list-style-type: none">• io.javalin.Javalin• maven.compiler.source 17• maven.compiler.target 17• javalin.version 6.1.3• javalin-rendering.version 6.1.3• thymeleaf.version 3.1.2• thymeleaf-extras.version 3.0.4• hikariCP.version 5.1.0• junit.version 5.10.2• postgresql.version 42.7.2	<ul style="list-style-type: none">• IntelliJ IDEA Maven

(Figur 4, tabel over teknologivalg)

5. Aktivitetsdiagram AS/IS



(Figur 5: as-is diagram)

For eksempel håndteres kundernes ønsker om "byg-selv carporte" ved, at kunden udfylder en formular, som derefter overføres manuelt mellem to eksterne systemer for at give en pris. Dette indebærer unødig manuelt arbejde og spildtid.

Martin føler, at sælgerne ikke kun fungerer som sælgere, men også som "mellemmænd" mellem systemerne og kunderne. I videoen har Martin meget stort fokus på, at kunderne ikke skal få adgang til styklisterne før købet af carporten:

"Vi ønsker ikke at dele en stykliste til en kunde, der får et tilbud. Han får styklisten i det øjeblik, han køber carporten" (Bilag: 1.Referenceliste - 1.4 interview video med Martin, minut 8:34).

Martin ønsker ikke at dele hvilke materialer FOG bruger, før købet af carporten, for at undgå, at kunderne køber materialerne hos konkurrenterne. Derfor kommer "styklisten" og hvem der skal se den til at være en af gruppens hovedpunkter, når det kommer til det nye system.

Gruppen skal derfor udvikle en plan for et nyt arbejdsflow og system, der kan samle alle Martins daglige opgaver. Systemet skal være overskueligt og reducere behovet for at skifte mellem forskellige systemer, så sælgerne kan fokusere mere på salg og mindre på dataoverførsel.

Figur 5 illustrerer AS-IS diagrammet. Et visuelt billede på, hvordan gruppen har forstået arbejdsgangen og system-flowet hos Fog - Carporte ud fra interview videoen med Martien.

Martin forklarer i videoen, at han og hans kollegaer spilder meget tid på "ikke-relaterede" opgaver.

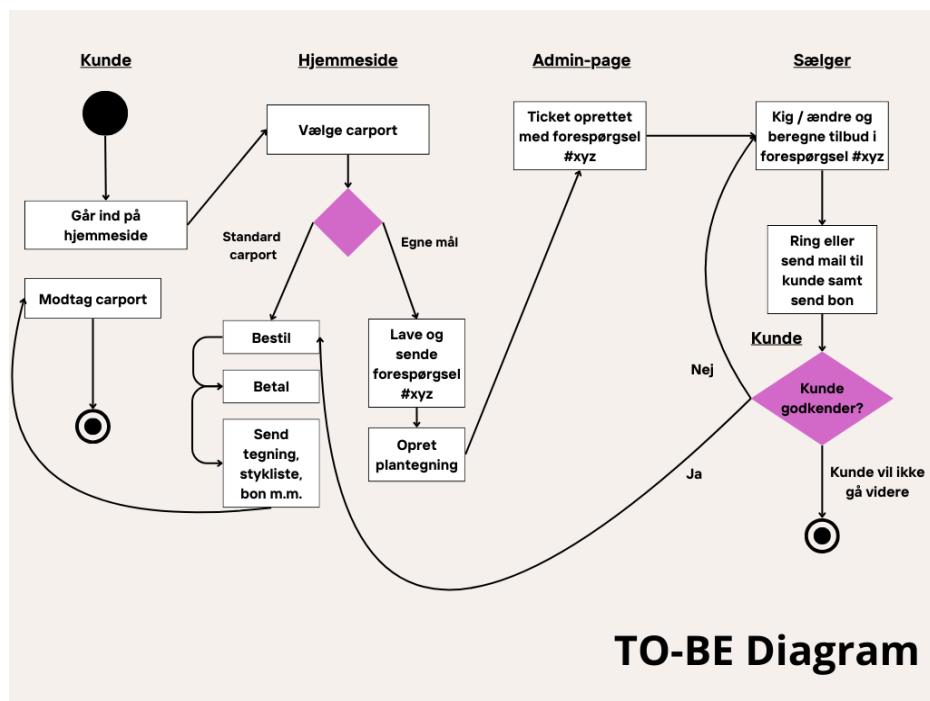
5.1 TO/BE (Swimlane)

I modsætning til "AS-IS" aktivitetsdiagrammet er "TO-BE" diagrammet en plan over, hvordan vi ønsker, at FOG's system skal se ud. Gruppens første tanke var at fjerne eller minimere antallet af programmer, der bruges til at lave et tilbud på en carport. Kigger man på figur 5, kan man se, at en sælger skal igennem et mail- og beregningssystem. Vi vil forsøge at samle det hele på FOG's hjemmeside, så sælgeren ikke manuelt skal flytte frem og tilbage med kundernes data.

Gruppen ønsker at ændre mailsystemet samt integrere alle beregningssystemets funktionaliteter på forskellige admin-konti. Vi håber, at dette kan give sælgerne mulighed for at fokusere mere på salget af carporten i stedet for at være "mellemmænd," som flytter kundernes data frem og tilbage.

Vi tænker ikke at fjerne hele mailsystemet, da det stadig har nogle funktionaliteter, som vi gerne vil beholde. Gruppen kan godt lide ideen om en "ordrebekræftelse," som bliver sendt på mail til kunderne, når de opretter en forespørgsel. Dog skal alle informationer omkring handlen foregå på hjemmesiden og fungere som et salgs-"ticket"-system. En forespørgsel skal ses som en "ticket," som en admin-konto kan se og ændre i.

Gruppen tænker og håber, at dette nye systemflow og design kan hjælpe FOG's sælgere og gøre deres arbejdsværelse nemmere.



6. User Stories and A/C

Disse "user-stories" har gruppen valgt at fokusere på gennem dette projektforløb. Vi vil så vidt muligt forsøge at implementere alle disse user-stories inden vores "code-freeze" d. 20/5-2024.

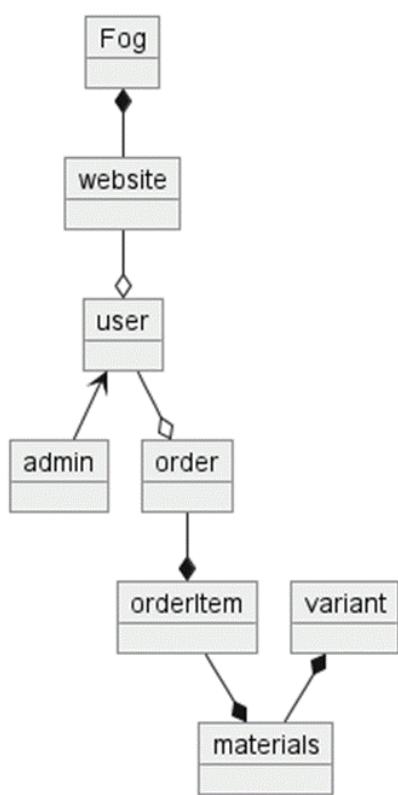
OBS: Vi har fået krav om kun at aflevere de 5 vigtigste user-stories og tilhørende A/C, men eftersom vi har lavet flere end 5, har vi valgt at aflevere følgende:
US-1, US-2, US-4, US-7 og US-10 samt alle tilhørende A/C.

User Stories	Acceptance Criteria
US-1 Som kunde skal jeg kunne se alle carporte på hjemmesiden med deres standard (STD) priser, og klikke ind på dem og få mere information	Givet jeg er på forsiden, vil jeg kunne se en liste med alle carporte fra databasen, som jeg så kan klikke ind på og få mere information
US-2 Som kunde skal jeg kunne bestille og betale for en standard (STD) carport uden at skulle igennem en sælger.	Givet jeg har fundet og valgt en carport jeg vil have, da trykker jeg på knappen "Køb nu" og bliver ledt hen til betalingsterminalen, hvor jeg derefter kan gennemføre købet. Herefter modtager jeg en ordrebekræftelse samt stykliste.
US-3 Som kunde skal jeg kunne indsætte mine egne mål på en carport og derved oprette en forespørgsel, som bliver sendt videre til en sælger.	Givet at jeg har fundet og valgt en carport jeg godt kan lide, kan jeg trykke på knappen "Bestil tilbud på carport i specialmål", hvor jeg kan indtaste alle mål og ønsker samt oplysninger på mig.
US-4 Som kunde skal jeg kunne se en plantegning på alle STD og custom carporte.	Givet at jeg har fundet en carport / har indtastet mine egne specialmål til en carport, da kan jeg trykke på knappen "Plantegning" og hermed få vist en plantegning over carporten.

US-5 Som kunde kan jeg betale for mine varer og modtage en ordrebekræftelse pr. mail samt styklisten.	Givet jeg har fundet eller selv har oprettet en carport, kan jeg betale for den, hvorefter jeg vil blive sendt en ordrebekræftelse pr. mail samt styklisten.
US-6 Som sælger skal jeg kunne logge ind på en “admin” bruger, og se en liste over kundeforespørgsler	Givet at jeg er logget ind som admin, når jeg klikker på knappen “Forespørgsler” og bliver ledt til ny side med alle kunde-forespørgsler.
US-7 Som sælger/admin skal jeg kunne vælge en specifik forespørgsel og se alle informationer der vedrører denne.	Givet jeg er logget ind som admin og har flere forespørgsler liggende i mit system, da kan jeg vælge en specifik forespørgsel fra listen af forespørgsler og se dens informationer.
US-8 Som sælger/admin skal jeg kunne ændre på den samlede pris på en forespørgsel (give et tilbud)	Givet jeg er logget ind som en admin, vil jeg kunne se en liste over forespørgsler, hvorefter jeg kan klikke ind på de forskellige forespørgsler og ændre priserne.
US-9 Som sælger/admin skal jeg kunne godkende / annullere en forespørgsel	Givet jeg er logget ind som en admin, kan jeg ‘efter en samtale med kunden over telefonen’, trykke på knapperne “Godkend” eller “Annuler” på hver enkelt forespørgsel.

(User-stories 6 : Tabeller med vores originale user-stories)

7. Domæne Diagram



Dette er, hvordan gruppen har forstået domænet hos Johannes Fog. FOG er ligesom mange andre byggemarkeder og butikker og har valgt at have en online hjemmeside, hvor kunder kan kigge på og købe carporte, som kan leveres til deres adresse.

Gruppen betragter relationen mellem FOG og hjemmesiden som en "stærk relation" i "HAS-A relationship"-klassen. FOG er "herre" over hjemmesiden og bestemmer derfor, hvad der sker på og med hjemmesiden. En hjemmeside kan dog godt eksistere uden nogen, der bruger den, så gruppen har valgt en "svag relation" i "HAS-A"-klassen for denne kontekst.

FOG ønsker en måde, hvorpå de kan se, hvilke ordrer der ligger i deres system efter køb af en carport, samt en måde at rette oplysninger på deres hjemmeside.

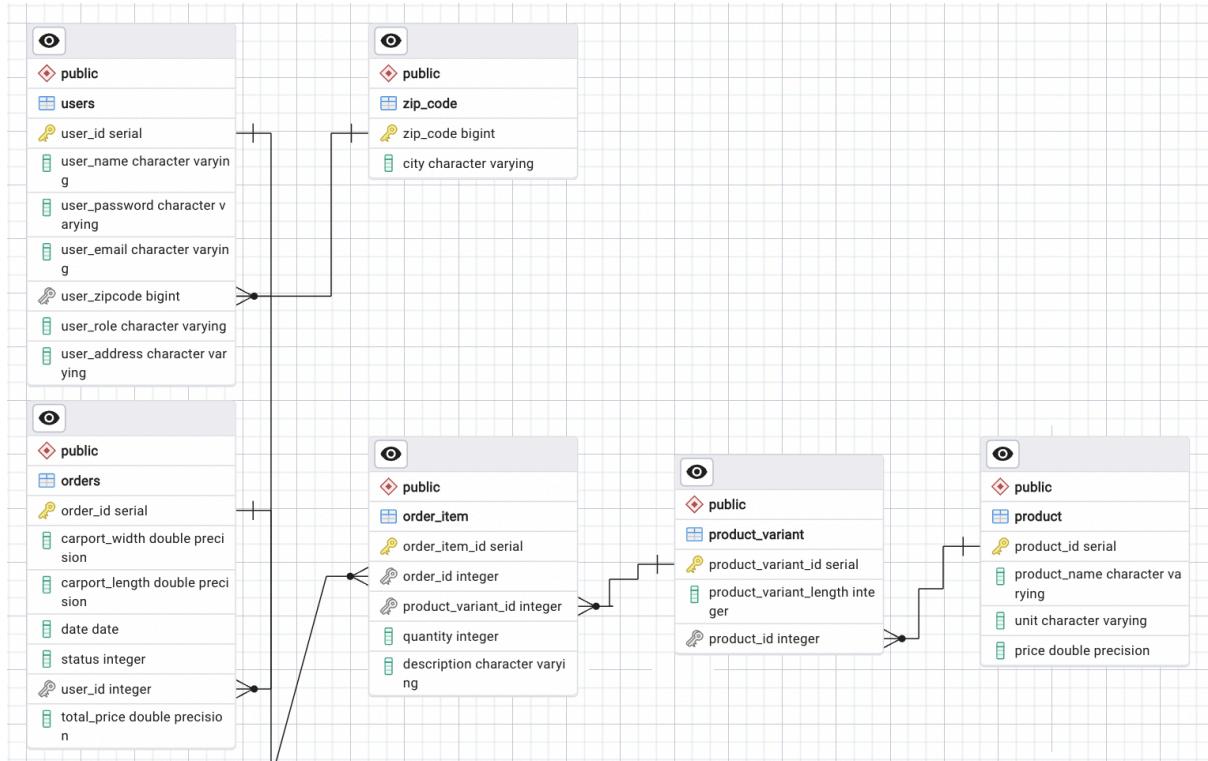
(Figur 7 billede af domæne diagrammet)

Dette kunne for eksempel være at udskifte sortimentet eller håndtere udsolgte varer i en periode.

Gruppen har derfor valgt, at en "admin" - som er forskellig fra en "bruger" - skal kunne udføre de ønskede opgaver. Relationen mellem bruger og admin er "IS-A," fordi ikke alle brugere skal kunne oprette en admin-bruger. Der skal være nogle "specielle" konti, som medarbejderne hos FOG kan logge ind på for at få den ønskede information.

Relationen mellem "bruger" og "ordre" er "svag," fordi man ikke skal oprette en ordre for at have en konto. Alle andre relationer er "HAS-A," fordi de skal bruge hinanden for at eksistere. Man kan ikke oprette en ordre (carport) uden at skulle bruge "carport-dele." Carport-dele består af forskellige materialer og har forskellige typer (træ, stål, tagplader osv.).

8. ERD Diagram



(Figur 8 billede af vores ERD diagram)

Vores database består af 6 tabeller: users, zip_code, orders, order_item, product_variant og product. Vi har valgt at lave både en ordre og en ordrelinje for at fremtidssikre det bedst muligt, da det gør det nemmere at tilføje nye produkter til databasen uden at skulle ændre i den eksisterende database. Derudover har vi valgt at lave en product_variant for at gøre databasen mere overskuelig, da vi ved at have den reducerer redundans, eftersom vi ikke længere vil have mange af de samme produkter i produkt-tabellen, men med forskellige længder.

Vi har implementeret det således, at en user har en én-til-mange-relation til orders og zip_code, så en bruger kun kan have et postnummer, og en ordre kan have én user. Et postnummer kan høre til mange brugere, og en bruger kan have mange ordrer. En ordre har en mange-til-mange-relation til order_items. Derudover har vi en én-til-mange-relation mellem order_items og product_variant, da vi igen kan have mange varianter af produkter i et order_item. Det samme gør sig gældende mellem product_variant og product.

9. Første klasse diagram

Figur 9 viser, hvordan vi i gruppen har valgt at starte dette projekt. Dette diagram skal ikke ses som den endelige, fastsatte plan for programmet, men mere som et 'uddelegering værktøj'. Da dette projekt og diagrammerne er lavet af os studerende på 2. semester med minimal erfaring, bør man forvente, at planen ændrer sig.

Diagrammets formål er at give os et 'startpunkt' for projektet og dets klasser. Da vi endnu ikke kan forudsige, hvordan vi når vores 'definition of done' for projektet, vil planen samt klassediagrammet sandsynligvis ændre sig. Selvom vores endelige klassediagram nok ikke kommer til at se sådan ud ved slutdatoen d. 24/5 - 2024, er det en god måde for os som studerende at vise vores udvikling og fremskridt i projektet.

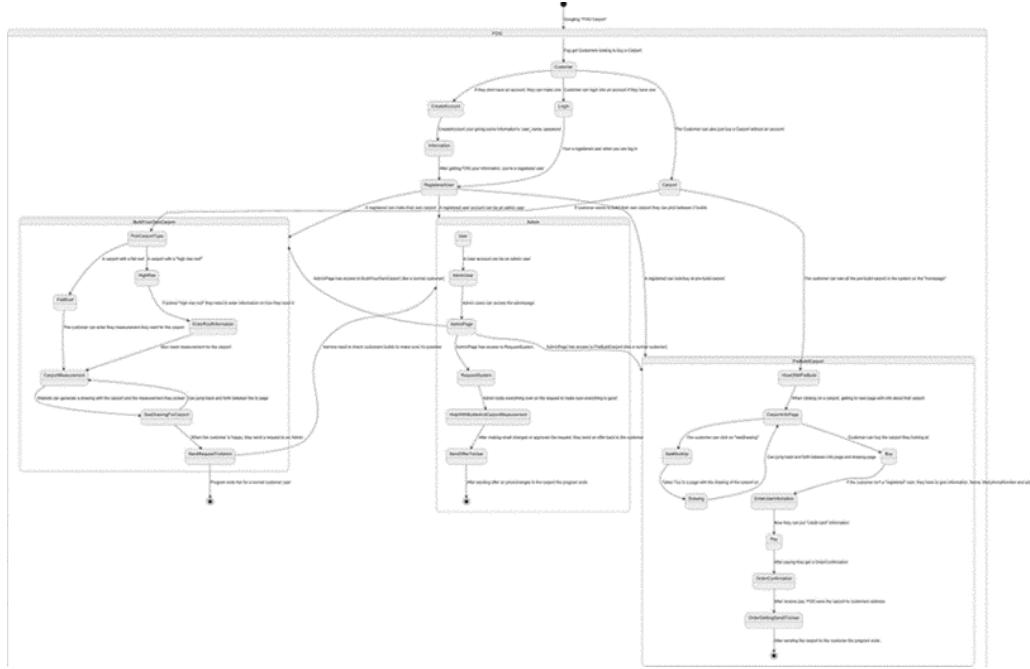


(Figur 9 første klassediagram)

Forklaringen på de forskellige klasser og deres formål står i [“12. valg af arkitektur s.17”](#). Samt kan man finde projektet sidste og endelige klassediagram i “doc”-mappen under diagrammer, navn: Final KlasseDiagram.

10. Navigationsdiagram

På grund af tidsbegrænsninger og instruktioner fra vores lærer må vi erkende, at vi ikke kan lave det hele. Derfor fokuserer vi på "Build Your Own Carport" og "Admin"-delen af hjemmesiden.



(Figur 10: Navigationsdiagram)

FOG tilbyder forskellige typer af carporte, man kan for eksempel lave: "carporte med fladt tag, høj rejsning, uden tagplader, med plasttrapezplade, med eller uden skur." Da vi ikke har tid til at sætte os ind i alle de forskellige farve-begreber og konstruktioner, har gruppen valgt at fokusere på en "standard carport" – en carport med fladt tag, uden tagplader og uden skur. Fra forsiden skal det være muligt at logge ind eller oprette en konto hos FOG, samt starte en forespørgsel på en carport. Knappen "Design din egen carport" starter "FOG's spørgeskema," hvor man bliver spurgt om mål og andre ønsker til sin carport. Herfra skal man kunne se en tegning af sin carport med de valgte mål og dernæst indtaste sine informationer. Når man er tilfreds og alt er som man vil have det, oprettes en forespørgsel på en carport. Efter oprettelsen af forespørgslen på carporten skal en person med adgang til en admin-konto kunne logge ind og se forespørgslen på vores "admin-page." Derefter skal en admin-konto have muligheden for at ændre pris i på ordenen. Og hvis der er tid, skal admin også kunne ændre i materialerne på ordren

11. Figma - Mockups

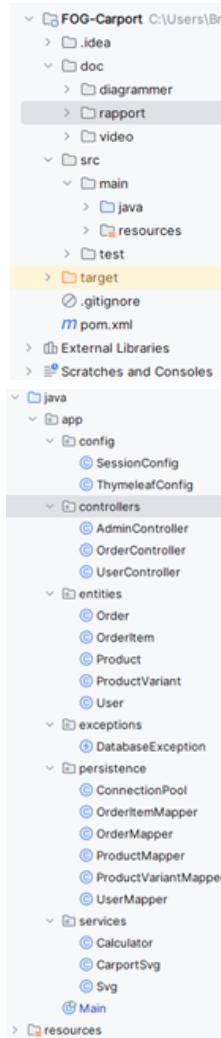
Gruppen har arbejdet i designprogrammet Figma og skabt flere udkast til FOG's nye hjemmeside. Disse mock-ups fungerer som inspiration og tillader kreativ frihed. Selvom vi stræber efter at følge vores Figma-designs, kan der komme uforudsete ændringer som kræver ændring i planen.

The image displays four Figma mockups of the FOG website:

- Carport Selection Page:** Shows a grid of carport models under the heading "Enkelt Carport". Options include "DOBBELT CARPORT 6,00x4,80M" (pris 21.490,-), "DOBBELT CARPORT 6,00x6,20M" (pris 25.005,-), "ENKELT CARPORT 3,90x7,80" (pris 35.998,-), "ENKELT CARPORT 4,90x8,80" (pris 32.990,-), "DOBBELT CARPORT 6,50x3,80M" (pris 47.908,-), and "DOBBELT CARPORT 6,50x8,60M" (pris 30.000,-). Each item has a "Bestil" button.
- Carport Design Request Page:** Titled "Lad os tegne din carport drøm - bestil et Quick-Byg tilbud". It includes a large image of a car parked under a wooden carport, and two smaller images below labeled "Carport med fladt tag → (link)" and "Carport med høj rejsning → (link)". A yellow callout box says "Vælg nedenfor, om du ønsker et bud på carport med fladt tag eller carport med høj rejsning".
- Carport Configuration Page:** Titled "Bestil Quick-Byg tilbud - carport med fladt tag". It shows input fields for "Carport bredde", "Carport længte", and "Carport trapettag*". Below is a large text area with the placeholder "Få vist prefix af plantegning". Buttons for "Tilbage" and "Forsiden" are at the bottom.
- Login Page:** Titled "Log in". It features fields for "Brugernavn" and "Kodeord", and buttons for "Sign in" and "Læs forsægelse".
- Quote Request Page:** Titled "Forespørgsler". It lists recent messages from users like "Marcus #5323", "Sander #441234", "Christ #123445", and "Malene #966452". A large form on the right contains sections for "Adresse", "Længe/Areal/Årelængde", "Bredde", "Højde", "Taktype", "Material", "Dør", "Fönster", "Tilføj til kundekonto", and "Send". A summary at the bottom shows "13.998,50 prakt + evt. fragt" and "16.998,50 pr. vsk. + evt. fragt".

12. Valg af Arkitektur

Vores projekt anvender en MVC-inspireret arkitekturstil til vores kode. MVC står for "Model-View-Controller" og er et designmønster, der adskiller og opdeler koden i overskuelige sektioner.



Til start har vi vores "Idea" og "doc" mapper, som indeholder alle vores XML-filer og dokumentation for projektet, såsom denne rapport, diagrammer og demovideo.

Vores "src" mappe er opdelt i to dele: "main", hvor al koden ligger, og "test" mappen, hvor alle vores test ligger. I "main" mappen er koden yderligere delt op i backend og frontend. Backend kaldes "java", og frontend kaldes "resources".

Java-mappen indeholder klassen "Main", som fungerer som vores "start/stop"-knap til programmet. Samtidig indeholder den "app"-mappen, forkortet for "applikation", hvor al koden til vores backend ligger.

I "Config"-mappen ligger klasserne ThymeleafConfig og SessionConfig, som er to ud af de fire klasser, der er blevet givet til os på forhånd af projektet. ThymeleafConfig bruges som bindeleddet mellem backend og frontend, mens SessionConfig gemmer data fra den nuværende sæson.

I "Controller"-mappen findes controller klasser, der navigerer brugerne mellem de forskellige HTML-sider og kontrollerer, hvilke data og informationer der bliver vist til brugerne.

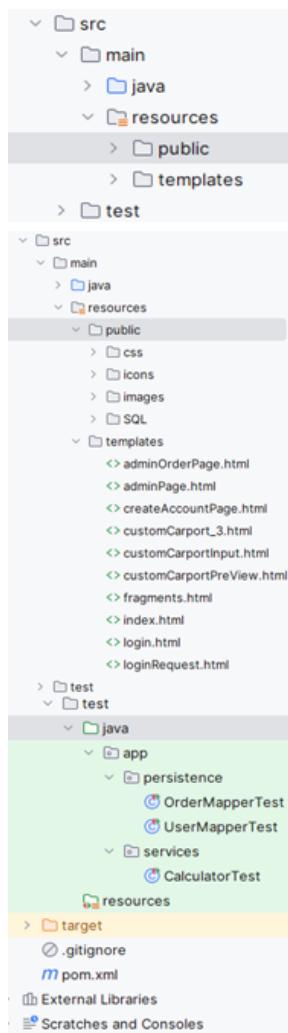
(Figur 12 arkitektur)

Entities-mappen indeholder forskellige klasser, der repræsenterer de forskellige objekter/domæner, vi arbejder med. Disse klasser fungerer som byggesten for objekter som "brugere", "materialer" og "ordrer".

I "Exceptions"-mappen ligger DatabaseExceptions, som er den næstsidste klasse, der er blevet givet til os på forhånd. Den håndterer alle fejl i forhold til databasen.

Persistence-mappen indeholder klassen ConnectionPool, den sidste af de fire klasser, der er blevet givet på forhånd. ConnectionPool opretter og vedligeholder forbindelsen til vores database. Derudover indeholder mappen forskellige "mapper"-klasser, som fungerer som broen mellem programmet og databasen.

I "Services"-mappen findes klasserne SVG, som genererer tegningen af carporten ud fra de valgte mål, og Calculator, som udregner antallet og typen af materialer, der er nødvendige til carporten.



Vores "resources" mappe er opdelt i "public" og "templates".

I "Public"-mappen findes al CSS, ikoner og billeder, der anvendes på alle vores HTML-sider. Derudover indeholder den en SQL-mappe med et SQL-script, der kan bruges til at generere en identisk database (med data) lokalt på computeren i pgAdmin.I "Templates"-mappen ligger al vores HTML-kode og de forskellige sider, man møder i vores program, samt "fragments", som forklares i ["14. udvalgte kodeeksempler"](#).

Til sidst har vi vores "test"-mappe, hvor vi placerer alle vores test for de forskellige klasser. Her er der lavet unit-tests på vores Calculator og integrationstests på nogle af vores "Mapper"-klasser.

(Figur 12.1 arkitektur)

13. Særlige forhold

Dette afsnit bruges til at beskrive særlige forhold der benyttes i programmet. Det kan f.eks. være:

- Hvilke informationer gemmes i application, session, og request variabler?

I **application** (app) har vi til at starte med gemt alle vores static files som ligger i /public mappen. Derudover gemmer vi alle routings, som gør det muligt at navigere rundt på hjemmesiden.

I **session** gemmer vi den indloggede bruger som “currentUser”, samt alle tilhørende attributter såsom id, navn og email. Udo over det gemmer vi også det input som brugeren indtaster i forespørgslen som session attributter.

I request-variabler gemmes mange informationer. Bruger-input fra login/opret bruger-siderne, bruger-input fra opret forespørgsel-siden og admin-input fra admin-siden. De informationer der gemmes, er relateret til siden/handlingen.

- Hvordan håndterer man exceptions og logger?

Vi håndterer exceptions vha. klassen DatabaseException og try-catch blokke. På den måde kan vi bedre styre hvordan fejl håndteres, og give mere meningsfulde fejlmeldelser. Dette gør vores program mere robust og lettere at fejlfinde.

- Hvordan har man valgt at lave brugerinput validering?

Når man gerne vil oprette en bruger, bliver man bedt om at indtaste sit password to gange. Hvis begge passwords stemmer overens, bliver brugeren oprettet og smidt ned i databasen. Modsat hvis der er forskel på de to passwords, bliver brugeren ikke oprettet. Brugeren skal også indtaste tegnet "@" i feltet e-mail.

- Hvordan har man valgt at lave sikkerhed i forbindelse med login?

I forbindelse med login kontrollerer vi i første omgang om e-mailen eksisterer i databasen. Hvis den eksisterer, kontrollerer vi, om den indtastede kode er korrekt. Dernæst bliver brugeren logget ind. Hvis der er noget der ikke stemmer overens, bliver der spyttet en besked ud til brugeren afhængig af hvad der ikke stemmer.

- Hvilke brugertyper (roller) der er valgt i databasen, og hvordan de er brugt i jdbc

Vi har valgt to brugertyper: admin og user. Man behøver ikke at logge ind for at få adgang til hjemmesiden, og se nogle standard carporte. Hvis man vil oprette en forespørgsel, kræver det at man er logget ind, logger ind eller opretter en bruger. Når man er logget ind som admin, bliver der tilføjet en ekstra knap i headeren, hvor man kan se alle forespørgsler og ændre i dem ift, pris, længde, bredde, status osv.

- SVG-tegning

Tegningen bliver genereret ud fra en bredde og en længde, en bruger indtaster. De forskellige dele af tegningen bliver først oprettet som attributter i "SVG" classen hvor tegningen derefter bliver brugt til at oprette selve tegningen i "carportSVG" ved at samle de forskellige dele ud fra de mål brugerne indtastede.

- Stykliste beregner

Al matematik og alle beregninger går igennem klassen "Calculator". Calculator har to opgaver: at beregne, hvilke materialer kunden skal bruge til sin carport (inklusive længde), og at finde ud af, hvor meget FOG skal sende til kunden. Efter alle beregninger bliver det hele samlet i en liste, som vises på admin-siden "Order".

14. Udvalgte kodeeksempler

14.1 Fragments og header

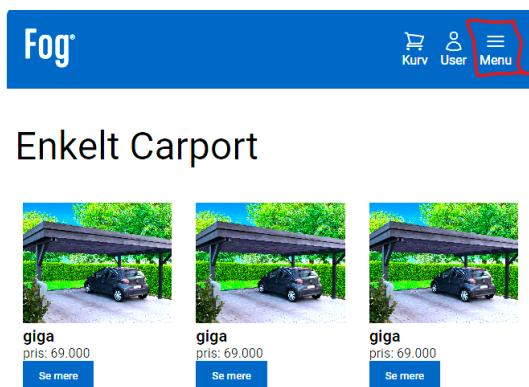
```
3   <head>
4     <title>Frontpage</title>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <link href="../public/css/styles.css" th:href="@{/css/styles.css}" rel="stylesheet"/>
8   </head>
9
10  <body>
11    <header th:fragment="header" class="header-container">
12      <a href="index.html" th:href="@{homepage}">
13        
14      </a>
15
16      <nav class="navbar">
17        <input type="checkbox" id="sidebar-active">
18        <label for="sidebar-active" class="open-menu">
19          <svg width="28" height="28" viewBox="0 0 24 24" fill="none" xmlns="http://www.w3.org/2000/svg">
20            <path d="M3.75 6.75H20.25M3.75 12H20.25M3.75 17.25H20.25" stroke="white" stroke-width="1.5"
21              stroke-linecap="round" stroke-linejoin="round"/>
22        </svg>
23        <span>Menu</span>
24      </label>
25    </header>
```

(billede 14.1)

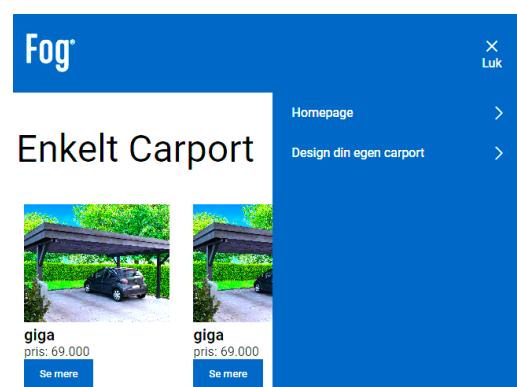
Dette er et billede af vores “fragments.html” side.

Dette har vi brugt til vores fordel ved at oprette vores header og footer som “fragments”, hvilket gjorde det muligt for os at undgå redundans og kun at lave dem én gang.

Vores header fragment har en toggle menu funktion som kun bliver vist, hvis man tilgår siden med en enhed på 700 pixels eller under som vist på billede 14.1.1 og 14.1.2. Den har til fordel at gemme alle vores links fra navigationen, for at undgå at de bliver mast på smallere enheder.



(billede 14.1.1)



(billede 14.1.2)

14.2 Carport SVG

```
3  public class Svg {
4      1 usage
5      private final static String SVG_TEMPLATE = "<svg version=\"1.1\"\n" +
6          "    x=\"%d\" y=\"%d\"\n" +
7          "    viewBox=\"%s\" width=\"%s\" \n" +
8          "    preserveAspectRatio=\"xMinYMin\" \n" +
9          "    class=\"carport-svg\" >";
```



```
10     1 usage
11     private static final String SVG_ARROW_DEFS = "<defs>\n" +
12         "    <marker id=\"beginArrow\" markerWidth=\"12\" markerHeight=\"12\" refX=\"0\" refY=\"6\" orient=\"auto\">\n" +
13         "        <path d=\"M0,6 L12,0 L12,12 L0,6\" style=\"fill: #000000;\" />\n" +
14         "    </marker>\n" +
15         "    <marker id=\"endArrow\" markerWidth=\"12\" markerHeight=\"12\" refX=\"12\" refY=\"6\" orient=\"auto\">\n" +
16         "        <path d=\"M0,0 L12,6 L0,12 L0,0 \" style=\"fill: #000000;\" />\n" +
17         "    </marker>\n" +
18     1 usage
19     public static final String SVG_RECT_TEMPLATE = "<rect x=\"%2f\" y=\"%2f\" height=\"%2f\" width=\"%2f\" style=\"%s\" />";
```

Dette er et billede af, hvordan nogle af templates først bliver oprettet som attributter i SVG klassen. Hvert template har nogle forskellige “placeholder værdier” i form af et procenttegn. Et “s = String, d = decimal, .2f = floating point” (der vises kun 2 decimaltal)

```
27     4 usages  ▲ Marcus Rasmussen
28     public Svg(int x, int y, String viewBox, String width){
29         svg.append(String.format(SVG_TEMPLATE, x, y, viewBox, width));
30         svg.append(SVG_ARROW_DEFS);
31     }
32
33     10 usages  ▲ Marcus Rasmussen
34     public void addRectangle(double x, double y, double length, double width, String style){
35         svg.append(String.format(SVG_RECT_TEMPLATE, x, y, length, width, style));
36     }
37
38     public CarportSvg(int width, int length){
39         this.width = width;
40         this.length = length;
41         mainCarport = new Svg(x: 0, y: 0, viewBox: "0 0 "+(length+offset)+" "+(width+offset), width: "75%");
42
43         carportSvg = new Svg(offset, y: 0, String.format("0 0 %d %d", (length+offset), width), width: "100%"); // :
44         addArrow();
45         addText();
46         addInnerSvg();
47     }
48
49     ~~~~~
50     public void addText(){
51         mainCarport.addText( style: "text-anchor: middle; transform: rotate(-90deg);", -(width / 2), y: 12, width); // left side
52         mainCarport.addText( style: "text-anchor: middle;", x: (length / 2) + offset, y: width + offset, length); // bottom
53     }
54
```

Attributterne bliver sendt videre ned i en ny metode. I en string formater, samt med diverse parametre til koordinater, størrelser og styling.

CarportSVG klassen er hvor SVG'en bliver oprettet og får tilføjet de forskellige metoder fra SVG klassen med de korrekte mål, ud fra hvad brugeren har indtastet. For at kunne give tegningen pile og tal til, at indikere længde og bredde på tegningen, har vi været nødt til at oprette den som “to svg'er”. Det vil sige vi har en overordnet svg som er “mainCarport” som får initialiseret den overordnede længde og bredde. Derudover får “mainCarport” tilføjet pilene, teksten og derefter “carportSVG” som er en ny SVG af selve carporten.

15. Status på implementering

Som udgangspunkt er vi nået godt igennem projektet, men vi har stadig nogle hængepartier og mangler. Vi har lavet alle de planlagte sider, som kan ses i Figma-mockupsne. Dog har vi ændret på vores "admin-page", da vi ikke havde tid til at lave den som oprindeligt planlagt. Vi har også fundet ud af, at vi mangler en "user-page" til at se status på ordrer og forespørgsler.

Alle sider er gjort responsive og inkluderer en toggle-menu til mobiltelefoner, med undtagelse af admin-siden. Admin-siden er ikke gjort responsiv, dels på grund af tidsmangel og dels fordi vi forventer, at sælgerne hos FOG sidder ved en computer. Gruppen har vurderet, at en "responsiv admin-side" er "nice to have".

Vores database indeholder alle de nødvendige tabeller og kolonner, som programmet kræver. Vi har også tilføjet ekstra kolonner i "Users"-tabellen for at få nogle ekstra "nice-to-have" informationer om brugerne. Derudover har vi oprettet en "zip_code"-tabel, som skal hente byen baseret på postnummeret, men dette har vi ikke nået at implementere på grund af tidsmangel. Vi har sat en "unique key" på e-mail-kolonnen, så det ikke er muligt at oprette flere brugere med samme e-mail. Databasen er IP-sikret, så kun brugere fra de tilladte IP-adresser kan tilgå den.

Calculator-klassen har plads til mange forbedringer. Koden er ikke særligt fremtidssikret og indeholder meget "hardcoded" indhold. Da vores hjemmeside ikke tillader forespørgsler på en carport over 7 meter i længde og bredde, er alle materialer fastsat. Koden kræver desuden, at man har kendskab til databasen og materialernes placering i den. Dette er noget, vi vil arbejde videre med efter aflevering, da vi ikke har tid til at forbedre det nu.

CarportSVG-klassen kunne også have brugt en ekstra kærlig hånd. Som det ser ud lige nu er stolperne hardcoded og der vil altid kun blive vist 4 stolper på svg'en tegningen. Dette er ikke helt optimalt da carporten kan ende med at bruge flere stoler afhængigt af længden på carporten. Man får dog det rigtige antal stolper i styklisten og det krævede bare at vi havde noget ekstra tid til at få implementeret regnestykket i metoden til udprintningen af stolperne.

16. Kvalitetssikring

Kvalitetssikring i projekter handler ikke kun om at køre programmet og se, om det virker. Det indebærer specifikke tests af metoder med høj fejlrisko, hvilket hjælper med fejlfinding.

I dette projekt er der lavet tre typer tests, inddelt i to hovedkategorier: automatiserede tests (unit- og integrationstests udført af IntelliJ) og bruger-acceptance-tests, hvor kunden sikrer, at programmet lever op til de aftalte brugerhistorier.

16.1 Automatiserede tests

Figur 12.2 i afsnittet “12. Valg af Arkitektur” viser, hvilke tests der er lavet og deres position i vores program.

Der er blevet lavet 2 unit-tests til Calculator-klassen på metoderne:

"calcPostQuantity" og "calcRaftersQuantity".

Desuden er der blevet lavet 3 integrationstests til OrderMapper på metoderne: "getAllOrders", "getOrderById" og "insertOrder".

Endvidere er der blevet lavet 2 integrationstests til UserMapper på metoderne "insertUser" og "getUserById".

Gruppen har primært fokuseret på funktionaliteten af vores program, og har derfor ikke haft tilstrækkelig tid til at lave flere tests. Dette har også resulteret i, at vores test-dækningsgrad ikke er tilstrækkelig.

Element ^	Class, %	Method, %	Line, %	Branch, %
app.persistence	50% (3/6)	40% (11/27)	35% (118/332)	9% (18/190)
ConnectionPool	100% (1/1)	71% (5/7)	73% (22/30)	50% (2/4)
OrderItemMapper	0% (0/1)	0% (0/1)	0% (0/17)	0% (0/10)
OrderMapper	100% (1/1)	30% (4/13)	42% (79/187)	10% (13/120)
ProductMapper	0% (0/1)	100% (0/0)	100% (0/0)	100% (0/0)
ProductVariantMapper	0% (0/1)	0% (0/1)	0% (0/20)	0% (0/6)
UserMapper	100% (1/1)	40% (2/5)	21% (17/78)	6% (3/50)

(Figur 16.1.1)

16.2 User Acceptance Test

Meningen med "User Acceptance Test" er, at alle kunder (i dette tilfælde Martin) skal kunne afprøve og teste de forskellige brugerhistorier, der er blevet aftalt. Dog kan CPH-business ikke tilkalde Martin hvert semester, da dette er en problemstilling fra 2016.

Derfor har gruppemedlemmet Marcus fået en af sine roommates til at teste vores program og gennemgå forskellige brugerhistorier.

User Stories	Acceptance Criteria	Resultat
US-1 Som kunde skal jeg kunne se alle carporte på hjemmesiden med deres standard (STD) priser, og klikke ind på dem og få mere information	Givet jeg er på forsiden, vil jeg kunne se en liste med alle carporte fra databasen, som jeg så kan klikke ind på og få mere information	Delvist godkendt. Vi fik at vide vi ikke skulle lave den del efterfølgende, så vi har valgt at lave en forside med nogle carporte som føre brugeren vidre til FOG's side med mere information om den valgte carport
US-2 Som kunde skal jeg kunne bestille og betale for en standard (STD) carport uden at skulle igennem en sælger.	Givet jeg har fundet og valgt en carport jeg vil have, trykker jeg på knappen "Køb nu" og bliver ledt hen til betalingsterminalen, hvor jeg derefter kan gennemføre købet. Herefter modtager jeg en ordrebekræftelse samt en stykliste.	Delvist godkendt. Ligesom i US-1 blev vi informeret om det ikke var nødvendigt at lave så brugeren bliver sendt videre til FOG's hjemmeside med mere information hvor de kan købe carporten der.
US-3 Som kunde skal jeg kunne indsætte mine egne mål på en carport og derved oprette en forespørgsel, som bliver sendt videre til en sælger.	Givet at jeg har fundet og valgt en carport jeg godt kan lide, kan jeg trykke på knappen "Bestil tilbud på carport i specialmål", hvor jeg kan indtaste alle mål og	Godkendt.

	ønsker samt oplysninger på mig.	
US-4 Som kunde skal jeg kunne se en plantegning på alle STD og custom carporte.	Givet at jeg har fundet en carport / har indtastet mine egne special-mål til en carport, da kan jeg trykke på knappen "Plantegning" og hermed få vist en plantegning over carporten.	Godkendt.
US-5 Som kunde kan jeg betale for mine varer og modtage en ordrebekræftelse pr. mail samt styklisten.	Givet jeg har fundet eller selv har oprettet en carport, kan jeg betale for den, hvorefter jeg vil blive sendt en ordrebekræftelse pr. mail samt styklisten.	Godkendt
US-6 Som sælger skal jeg kunne logge ind på en "admin" bruger, og se en liste over kundeforespørgsler	Givet at jeg er logget ind som admin, når jeg klikker på knappen "Forespørgsler", da bliver jeg ledt hen til en ny side med alle kunde-forespørgsler.	Godkendt
US-7 Som sælger/admin skal jeg kunne vælge en specifik forespørgsel og se alle informationer der vedrører.	Givet jeg er logget ind som admin og har flere forespørgsler liggende i mit system, da kan jeg vælge en specifik forespørgsel fra listen af forespørgsler og se dens informationer.	Godkendt
-----	-----	-----

US-8 Som sælger/admin skal jeg kunne ændre på den samlede pris på en forespørgsel (give et tilbud)	Givet jeg er logget ind som en admin, vil jeg kunne se en liste over forespørgsler, hvorefter jeg kan klikke ind på de forskellige forespørgsler og ændre priserne.	Godkendt
US-9 Som sælger/admin skal jeg kunne godkende / annullere en forespørgsel	Givet jeg er logget ind som en admin, kan jeg 'efter en samtale med kunden over telefonen', trykke på knapperne "Godkend" eller "Annuler" på hver enkelt forespørgsel.	Godkendt

(Figur 16.2)

17. Proces

Vores arbejdsform og samarbejde har været rigtig godt siden det forrige projekt "Cupcake 2024" (et andet større projekt på Datamatikeruddannelsen). Derfor valgte gruppen at tage imod denne udfordring og samles omkring semesterprojektet. Denne gang har vores tilgang til arbejdsprocessen været lidt anderledes. På grund af arbejde og andre forpligtelser har gruppen ikke haft en fast plan for arbejdstiden. Den har ændret sig fra dag til dag. Nogle dage har mødetiden været fra 9.30 til omkring 15, og andre dage startede vi først omkring kl. 17.30 til omkring 21.00. Mange gange blev der også arbejdet 2-3 timer, både lørdage og søndage. Den eneste undtagelse har været den sidste uge af projektet. Her har gruppen haft en fast arbejdstid fra 10-16.

I sidste projekt fandt vi en arbejdsform, som fungerede godt for os, så den har vi også brugt denne gang. Vores arbejdsform/motto er:

"Alle skal lave lidt af alt".

Det ville være for kedeligt og ikke særligt lærerigt, hvis man kun arbejdede med det, man allerede forstod og var god til. Derfor har vi indført mottoet "alle skal lave lidt af alt". Alle skal lave mindst én HTML-side med CSS, bidrage med noget kode til backenden, uanset om det er mapper, klasser eller controllere. Derudover skal alle skrive noget til rapporten og være enige om projektets diagrammer. Dette har gjort, at vi alle skulle "søge ny information", hvad enten det har været via Google eller hos hinanden.

Alle har bidraget med lidt af alt og har været en del af både backend og frontend. Dog ville det være en løgn at sige, at alle har lavet lige meget af alt. For at vi kunne nå i mål, har vi været nødt til at uddelegere arbejdet efter vores styrker og svagheder. Vores gode motto: "Alle skal lave lidt af alt" er blevet opfyldt, dog med nogle få undtagelser.

17.1 Arbejdsprocessen faktuelt

- Arbejdsprocessen og dens faser kan opdeles i forskellige uger, som vi har haft gennem projektet.
- Gruppen har brugt et Trello-board til at holde overblik over arbejdsprocessen.
- Til hver vejledning har vi haft vores lærer, Signe. Alle gruppemedlemmer har skrevet spørgsmål ned i løbet af de forskellige uger for at være forberedt til vejledningen.
- Alt arbejde er dokumenteret i vores dagbog, som kan findes i bilaget.
- Kommunikationen er foregået via Discord, hvor alle gruppemedlemmer har mødtes hver dag for at holde et kort statusmøde: "Hvordan går det? Skal nogen have hjælp til noget? "Har du nogen gode ideer?"
- Code freeze dato: 20/5-2024.

Den første uge af projektet gik med planlægning, diagrammer, risikoanalyser, Figma-mockups osv. Gruppen havde stort fokus på at have en god opstart på projektet. Vi så og analyserede interview-videoen endnu en gang sammen, for at sikre en god fælles forståelse af opgaven. Signe gennemgik alt vores forarbejde ved første vejledning og gav feedback til ændringer og forbedringer.

Anden uge var den første uge med kodning. Her fik vi lavet skelettet til projektet: oprettet et GitHub-repository, indsat alle klasserne med metoder, vi troede, vi skulle bruge, og startede på de simple HTML-sider samt de forskellige mapper-klasser. Til vejledningen havde vi mange forståelsesspørgsmål til Signe:

"*Hvad gør vi ved XML-filer? Ser vores ProductMapper okay ud? Hvad skal vi fokusere på nu?*" (Se: 12. Valg af arkitektur).

Den tredje uge blev kodningen mere seriøs. Vi havde fået instruktioner om at starte på SVG-koden og begynde på Calculator-klassen, da det er det mest essentielle i denne opgave. Marcus startede på SVG, Sander startede på Calculator-klassen, og Mateen fik styr på de sidste mapper og controller-klasser. Vejledningen med Signe handlede om spørgsmål angående rapporten og kode, da code freeze-datoen nærmede sig.

I den sidste uge af forløbet havde gruppen code freeze, og der blev ikke rigtigt kodet. Vi deployede vores hjemmeside til "<https://fog.mateenrafiq.dk/>" på Google. Vi har pushet al vores kode til vores master-branch og færdiggjorde rapporten.

17.2 Arbejdsprocessen reflekteret

Hvad gik godt?

Vi har alle forbedret vores "kodeskills", ikke kun i Java, men også i HTML og CSS, hvilket har givet os en bedre forståelse for "full-stack development". Gruppens samarbejde er også blevet styrket, med bedre kommunikation og bedre strukturering af projektet. Desuden har vi forbedret vores evne til at lave diagrammer og planlægge vores arbejde.

Hvad gik knap så godt?

Vi har haft udfordringer med at fastholde brugen af kanban-boards og har ikke udnyttet vores vejledningsmøder optimalt, da de primært har handlet om forståelse frem for kodning. Dette har ført til usikkerhed omkring kvaliteten af vores kode og dens fremtidssikring, samt at vi som gruppe er lidt usikre på, om vores kode er "god nok". *Har vi fået kodet det godt nok, kunne det gøres smartere, kunne vi have lavet det mere fremtidssikret?*

Disse er alle ting som skal arbejdes på og forbedres til næste projekt.

Hvad tager vi med os videre?

Mateen: Jeg har fået en bedre forståelse for vigtigheden af forberedelsen. At lave diagrammer, analysere interview-videoen grundigt og lave nogle gode user-stories har gjort projektet nemmere og mere overskueligt at gå i gang med. Dog er der stadig plads til forbedring.

Marcus:

Jeg har fået en bedre forståelse for hvordan svg-tegninger bliver lavet, samt fået en viden om hvordan man kan generere dem ud fra forskellige mål.

Sander:

Jeg har lært ikke at undervurdere alle de forskellige "test" der skal laves til sådan et projekt. Integrationstestene på de forskellige mapper-metoder var ikke så "ligetil" som før antaget. Testen af "dækningsgrad" har for mig, spillet en stor rolle og er noget som jeg skal arbejde på.

De sidste tanker:

Selvom der er områder, hvor vi kunne have gjort det bedre, synes vi at projektet er en succes. Vi er tilfredse med vores indsats, selvom der er småting, der mangler at blive finpudset. Samlet set føler vi, at vi har gjort vores bedste og brugt arbejdstiden fornuftigt og til bedste evne.

Bilag

1. Referenceliste

- 1.1 link til github - eksamensprojekt:

<https://github.com/sanderMarcusChristensen/FOG-Carport>

- 1.2 link til projektet :

<https://fog.mateenrafiq.dk/>

- 1.3 link til opgavebeskrivelse:

<https://cphbusiness.mrooms.net/mod/book/view.php?id=654353&chapterid=33087>

- 1.4 link til Fog's carport hjemmeside:

<https://www.johannesfog.dk/have-fritid/carporte/enkelt-carporte>

- 1.5 link til interview video med Martin:

<https://cphbusiness.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=41de9cde-b343-48b4-bd3d-ad1701664bf4>

- 1.6 link til Demo video:

https://youtu.be/W0G_uAmECiQ

2. Nice to have User-stories

Dette er alle user-stories, som gruppen godt kunne tænke sig, var med i programmet, men som samtidigt ikke er nødvendige. Derfor har gruppen valgt ikke at have dem som "Nice to have".

Hvis alle andre user-stories er lavet og godkendt, og gruppen har tid til overs, er disse de første ekstra ting gruppen vil implementere.

US 1. Som kunde skal jeg kunne oprette eller logge på en bruger, som kan gemme navn, betalingskort og leveringsadresse	Givet at jeg gerne vil oprette en bruger, når jeg kommer ind på forsiden, da kan jeg trykke på knappen "Opret bruger", indtaste mine oplysninger og til sidst trykke "Opret".
US-2 Som registreret kunde vil jeg nulstille min adgangskode	Givet at jeg har en aktiv account hos FOG, men ikke kan huske mit brugernavn eller kodeord. Kan jeg trykke på knappen "glemt kodeord"/"glemt brugernavn", hvor jeg derefter får tilsendt en mail med et link til oprettelsen af et nyt kodeord eller brugernavn.
US 3 Som admin skal jeg kunne ændre på de bestilte materialer og ikke kun prisen.	Givet at jeg er på admin-siden med "Forespørgsler", når jeg klikker ind på en specifik forespørgsel, da kan jeg efter en dialog med kunden ændre på de ønskede materialer som kunden har valgt.
US 4. Som "indlogget" bruger skal jeg have en side, hvor jeg kan se "min bon" på min forespørgsel	Givet at jeg er logget ind som en bruger, når jeg trykker på knappen "Mine køb" da kan jeg blive ledt hen til en ny side, hvor mine køb er listet.
US 5. Som Bruger kan jeg se en "status" på hvor langt min ordrer er i systemet.	Givet jeg er logget ind som normal bruger, kan jeg inde på min "user-page" se hvilken "status" min ordre har. Altså, hvor langt min ordre er kommet i systemet.

3. Dagbog

Vi som gruppe har valgt at skrive dagbog til dette projekt. Vi skriver som gruppe og ikke individer, da vi næsten altid arbejder sammen. Dette gør også at vi kan skiftes til at skrive lidt dagbog fra dag til dag og ikke alle skal skrive lidt. Dagbogen skal ses som en huskeliste til når vi skriver om vores "proces". Så vi kan gå tilbage og se, hvilke ting vi startede på, hvornår og hvilke udfordringer vi har haft igennem processen.

Dagbog bliver lagt i "bilag", da den ellers ville fyldte hele rapporten. Derfor bliver der kun skrevet mellem 3-5 linjer hver dag for at kunne huske de vigtigste ting. Efter datoén d. 20/5 - 2024, bliver der ikke skrevet mere dagbog. Her har gruppen valgt at lave "code freeze", det vil sige vi ikke "koder" mere på projekt og fokusere på rapportskrivningen.

29. april

Vi genså videoen, hvor Martin bliver interviewet for at være sikker på hvad han ville have. Lavet nogle nye user-stories ud fra vores forståelse, samt "nice to have" user-stories. Vi startede på nogle nye AS-IS og TO-BE diagrammer.

30. april

Vi startede på rapporten med en indledning, baggrund samt fik sat forsiden og indholdsfortegnelsen op. Vi fik lavet et domæne diagram samt lavet et udkast til et ERD diagram.

1. maj

Vi fik sat alle vores user-stories fast, samt fik vi færdiggjort vores "AS-IS" og "TO-BE". Vi startede på vores klassediagram, vores database og ERD-diagram. Til sidst fik vi brainstormet på hvordan hjemmesiden skulle se ud og gik i gang med figma modellen.

2. maj

Vi har fået oprettet projektet, samt færdiggjort vores design i figma. fået hjælp til vores ERD diagram til vores database og fået planlagt vores klassediagram.

3. maj

Vi har rettet vores domæne diagram og fået sat de rette relationer op, samt oprettet nogle af de basale klasser. Vi har lavet et ekstra par sider i figma efter en samtale med Jon vi havde slut på dagen før. Til sidst fik vi startet på det første klasse diagram.

5. maj

Vi har fået udfyldt det sidste på vores første klassediagram og lavet vores navigationsdiagram .

6. maj

Vi oprettede alle klasser og fik lavet de første par metoder på de forskellige "Mapper-klasser". Samt lavet alle de overordnet "entities" såsom "Order, materialer, user" osv.

Marcus fik startet forsiden i HTML hvor man kan se alle de forskellige carporte som FOG tilbyder.

7. maj

Dagen bød på forskellige udfordringer, da alle i gruppen havde forskellige tidsplaner denne dag. Dog fik vi lavet udkast til de sidste metoder i "variantMapper" klassen, kigget mere på HTML og skrevet dagbog. Fik også skrevet en forklaring til vores domæne diagram. En forklaring på, hvordan vi som gruppe har forstået domænet hos FOG og hvordan vi har forstået opgaven.

8. maj

For det meste gik dagen med at lave forskellige HTML sider og samle spørgsmål da vi havde vejledningen med vores lærer kl.15.30. Vi fik lavet vores "fragments-html" side færdig. En side hvor vi genbruger vores "header" så det er ens på alle sider. Samtidig giver den en lille "menu" til hvis man tilgår hjemmesiden fra en telefon (se billede under bilag).

9.maj

Vi fik endelig sat vores database op på vores droplet på digitalocean, men hjælp fra en klassekammerat. Samtidige fik lavet Login og createaccount siderne i html (mangler dog lidt css). vi startede også på create-carport siderne og gået i gang med at se Jons videoer omkring "SVG-kode" og hvordan man laver de forskellige udregninger til calculator klassen.

10.maj

Vi har fået opdateret UserMapper klassen og databasen således, at programmet nu kan registrere en bruger når man indtaster sine oplysninger under en forespørgsel. Derudover har vi også rettet et par småting i koden.

12.maj

Der er blevet skrevet en side tekst til "as-is" samt lavet generel oprydning i rapporten.

13 - 17 .maj

Gruppen har desværre glemt at skrive dagbog for disse dage, da arbejde og andre forpligtelser har spillet ind. Dog har gruppen fået udrettet meget på disse dage. Marcus har fået genereret "Svg" tegninger på alle ordrer. Mateen har fået en af admin-siderne på og kører, og Sander har fået lavet "Calculator- klassen" samt testede på calculator og nogle forskellige Mapper-Klasser.

18 - 19 .mai

Selv om det er weekend, arbejder gruppen stadig et par timer om dagen. Gruppen mangler kun de sidste rettelser i svg tegningerne og den sidste admin-side i HTML. Gruppen vurderer at de "sagtens" kan nå at blive færdig til kode-freez.. Men de nævnte klasser har også fået nye navne, da det var mere overskueligt for os.

20.mai

Sidste kode-dage inden "code freeze". Alle har fået afsluttet deres hængepartier, det eneste vi mangler er at deploye vores hjemmeside og lave "code-clean up".