

Johannes Fog - Carport Case - Rapport

May 25, 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](#)

[Link til websitet](#)

[Link til Javadoc](#)

Contents

1	Indledning	3
1.1	Baggrund	3
1.2	Teknologivalg	3
2	Krav	3
2.1	Overordnet beskrivelse af virksomheden	4
2.2	Arbejdsgange der skal IT-støttes	4
2.3	Scrum userstories	4
3	Domæne model og ER diagram	5
4	Navigationsdiagram	5
5	Sekvensdiagrammer	5
6	Særlige forhold	5
7	Udvalgte kodeeksempler	5
8	Status på implementation	5
9	Test	5
10	Process	6
10.1	Arbejdsprocessen faktuel	6
10.2	Arbejdsprocessen reflekteret	6
11	Appendix	7
11.1	Appendix A: Diagrammer	7
11.2	Appendix B: User Stories	12
11.2.1	Implementeret	12
11.2.2	Ikke implementeret	12

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.

1.1 Baggrund

Johannes Fog er en koncern der både har design & bolighuse og trælast & byggecentre. 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om at udvikle et system, med tilhørende hjemmeside, til at erstatte deres nuværende system, da de har erkendt at det er outdated.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 et 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. Databasen, mysql 5.1.39, er lavet i MySQL Workbench 6.3ce, og ligger på en Ubuntu 16.04.3 x64 server, hvor websitet er deployet vha. apache-tomcat-8.0.32. Sprogene der er brugt i koden er java jdk 1.8.0_141, HTML5 og javax 7.0.

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 udfra en ordre og en vejledende pris udfra styklisten.

2.1 Overordnet beskrivelse af virksomheden

2.2 Arbejdsgange der skal IT-støttes

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.

estimat: Estimeret til et par timers arbejde.

3 Domæne model og ER diagram

4 Navigationsdiagram

5 Sekvensdiagrammer

6 Særlige forhold

7 Udvalgte kodeeksempler

8 Status på implementation

9 Test

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 sig 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 skaber 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 faktorer der spiller ind. Whitebox testing er en test metode der giver personen som tester mulighed for at følge med i koden kontra Blackbox testing hvor der ikke er adgang til koden. Vi har brugt et plugin der hedder TikiOne JaCoCoverage, som skaber et overblik over hvilke metoder der er blevet testet.

10 Process

10.1 Arbejdsprocessen faktuel

10.2 Arbejdsprocessen reflekteret

11 Appendix

11.1 Appendix A: Diagrammer

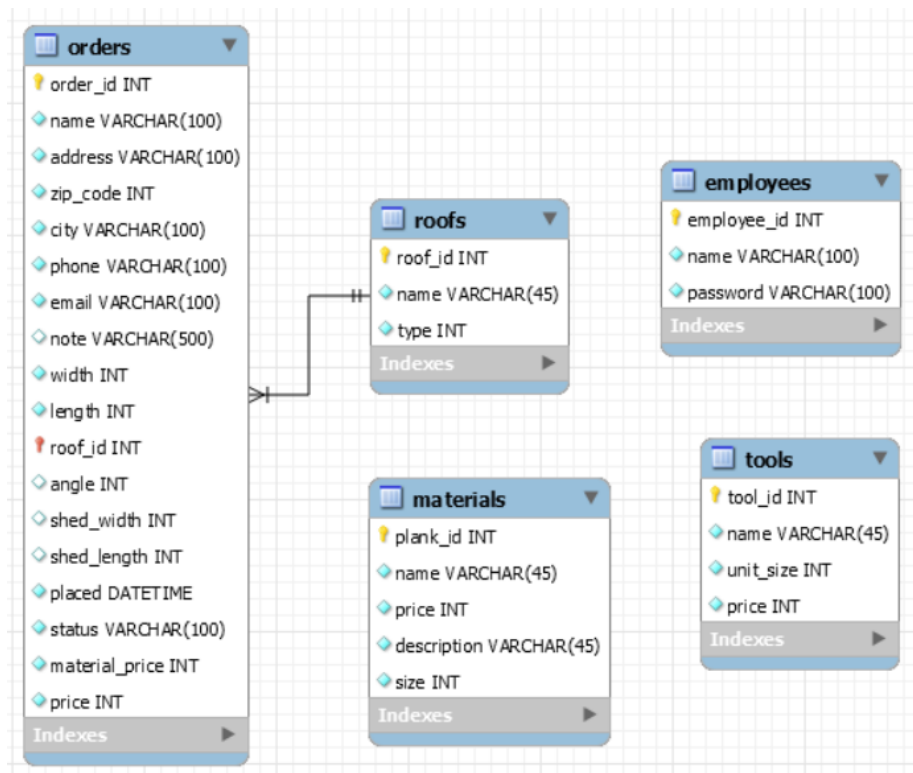


Figure 1:
ER diagram

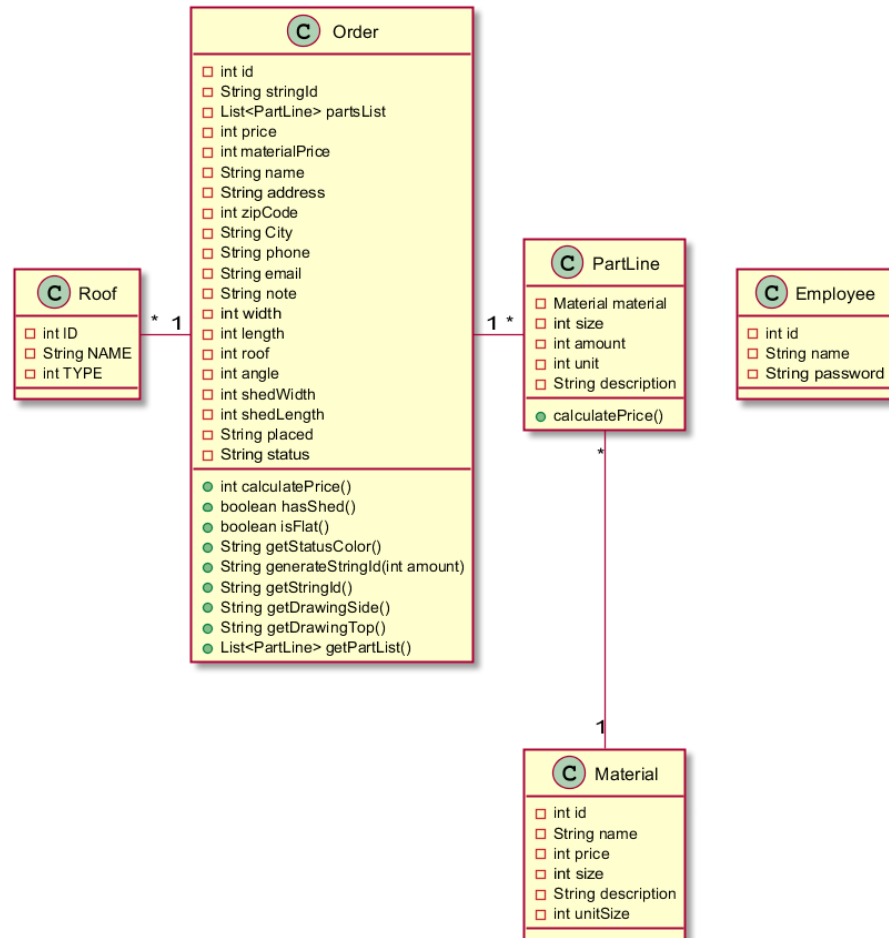


Figure 2:
Klassediagram

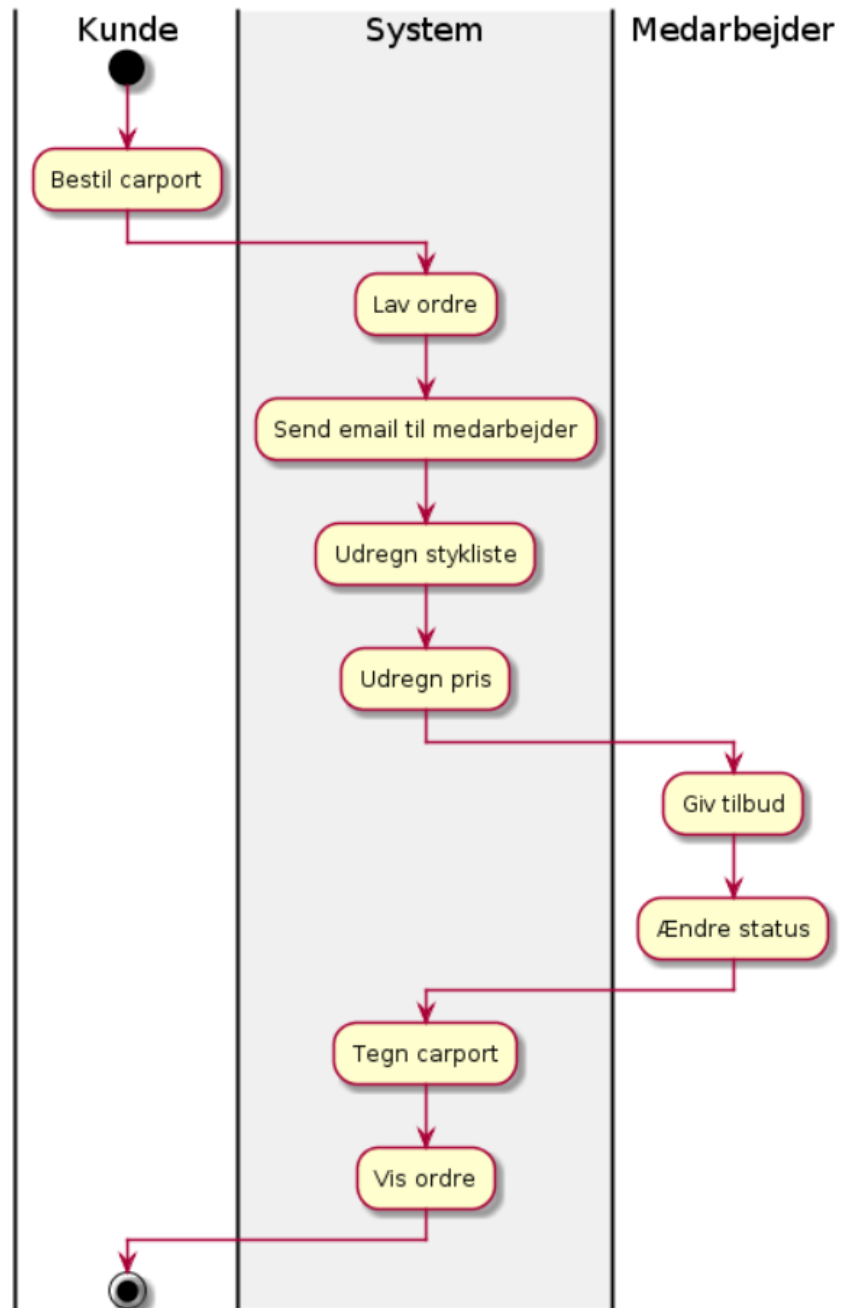


Figure 3:
Aktivitetsdiagram: Kunde bestiller en carport

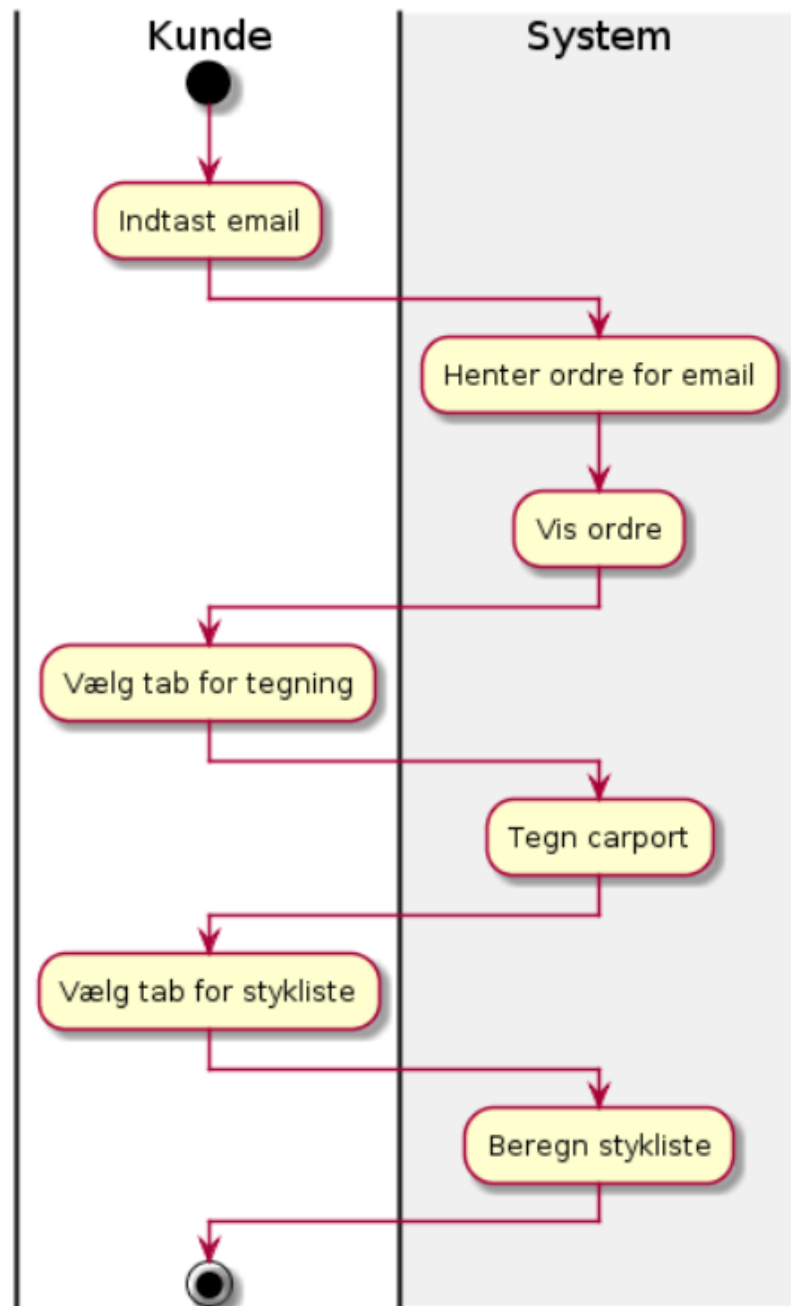


Figure 4:
Aktivitetsdiagram: Kunde ser sine ordre

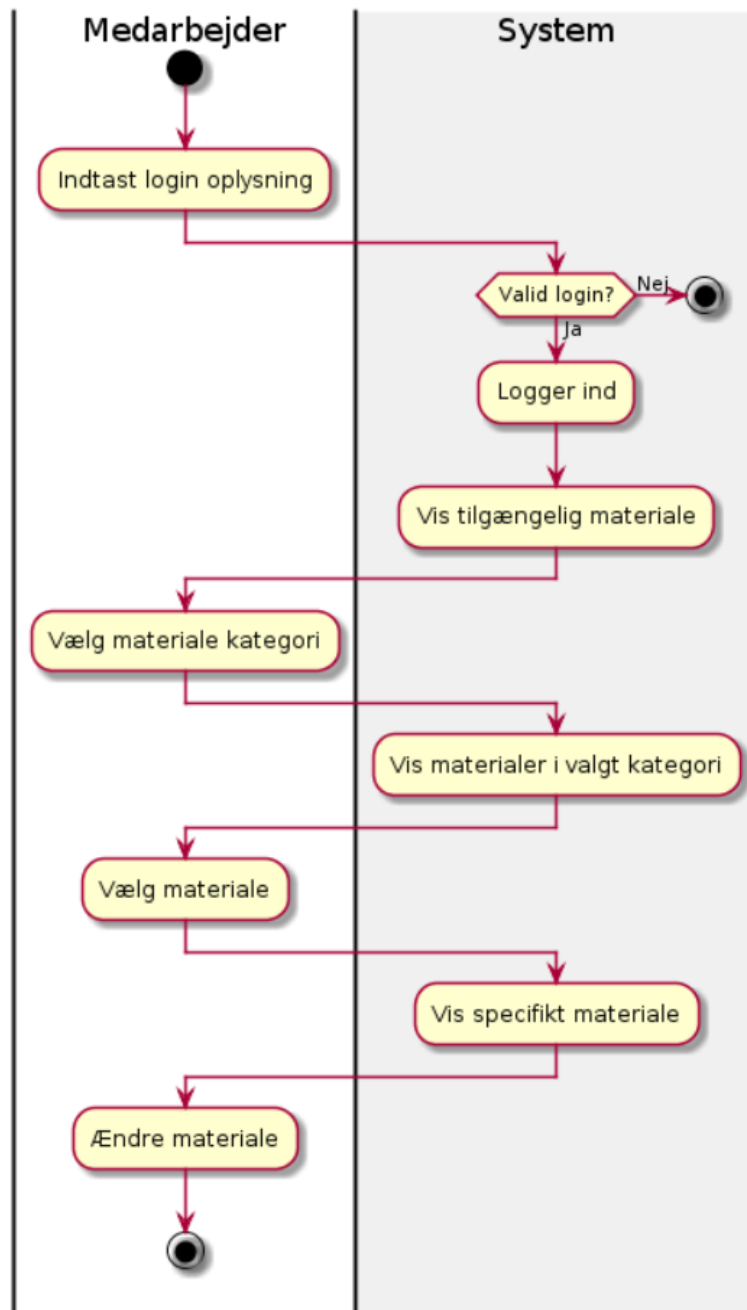


Figure 5:
Aktivitetsdiagram: Medarbejder ændre på tilgængeligt materiale

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 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 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 se prisen på min carport så jeg ved om jeg har råd til den
- 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 medarbejder vil jeg kunne udregne en stykliste så jeg ved hvilke materialer der skal bruges
- 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 være i stand til at ændre på tilgængelige materialer så jeg ved hvilke materialer jeg har
- Som medarbejder vil jeg være i stand til at annullere en ordre så det er muligt at annullere sin ordre
- Som medarbejder vil jeg have mulighed for at ændre status for en ordre så kan holde kunden opdateret
- 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

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