

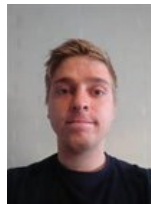
**DTU Compute**  
Institut for Matematik og Computer Science

## CDIO 22\_del2

### Gruppe 22



Kasper Baun (s172483)



Hans Peter Byager (s185132)



Anders Frandsen (s170197)



Rikke Galløe (s185015)



Jakob Mortensen (s185134)

DTU Compute

Institut for Matematik og Computer Science

Danmarks Tekniske Universitet

Matematiktorvet

2800 Kongens Lyngby, Denmark

Phone 45 4525 3031

[compute@compute.dtu.dk](mailto:compute@compute.dtu.dk)

[www.compute.dtu.dk](http://www.compute.dtu.dk)

# Log & Timeregnskab

---

Fase	Rikke	Kasper	Anders	Jakob	Hans
Planlægning	2	3	2	2	2
Krav	2	2	2	2	2
Design + Implementering	6	6	7	8	6,5
Rapport	6	3,5	2	4	2,5
Total	14	14,5	13	16	13

# Forord

---

Formålet med dette projekt er at få designet og udviklet et softwareprogram som benytter en vægt til afvejning af materiale og gemmer denne data samt hvem der har afvejet det.

I vores analyse og design af problemområdet har vi valgt at benytte Interfaces for let udskiftelighed af de forskellige klasser fx hvis man senere udskifter vægten har man stadig behov for de samme metoder.

Vi er lykkedes med at fremstille et velfungerende afvejningssystem program der gemmer data persistent i en .txt fil der ligger lokalt i samme mappe som projektet.

Kongens Lyngby, 23. marts 2019

Kasper Baun (s172483)  
Hans Peter Byager (s185132)  
Anders Frandsen (s170197)  
Rikke Rubin Galløe (s185015)  
Jakob Ole Mortensen (s185134)

# Indhold

---

<b>Forord</b>	<b>ii</b>
<b>Indhold</b>	<b>iii</b>
<b>1 Kravspecifikation</b>	<b>1</b>
1.1 Kravliste . . . . .	1
<b>2 Analyse</b>	<b>2</b>
2.1 Use Cases . . . . .	2
2.2 Rigt billede . . . . .	3
2.3 System Sekvens Diagram . . . . .	3
<b>3 Design</b>	<b>5</b>
<b>4 Konklusion</b>	<b>7</b>
<b>Bibliografi</b>	<b>8</b>

# KAPITEL 1

# Kravspecifikation

---

## 1.1 Kravliste

Vi lavede en prioritering af kravene ved hjælp af en MoSCoW-analyse.

### 1.1.1 Must have krav

1. Brugeren skal kunne logge sig på vægten som godkender brugeren via sit operatør ID og skriver navnet på vægten.
2. Vægten accpeterer et batch ID.
3. Operatøren skal kunne kvitterer til vægten for at indtastede oplysninger er korrekte.
4. Vægten skal kunne tareres.
5. Alle afvejninger skal kunne gemmes uanset om de kasseres eller accepteres.

### 1.1.2 Should have krav

1. Programmet bør have input validering i TUI.
2. Programmet lagre persistent i en database.

### 1.1.3 Could have krav

1. Programmet viser et informationsfelt ved behov for eksempel med informationer om materialet.

### 1.1.4 Won't have krav

1. Systemet kunne have et loop så man kun logger ind én gang - evt. med en log ud knap eller automatisk log ud ved afslutning af afvejningen.

# KAPITEL 2

## Analyse

### 2.1 Use Cases

Vi fandt kun en meget omfattende use case i dette projekt, nemlig BenytVægt.

#### 2.1.1 Brief use case

ID	UC01
Navn	Benyt Vægt
Beskrivelse	En bruger logger ind på vægten og afvejer et produkt.

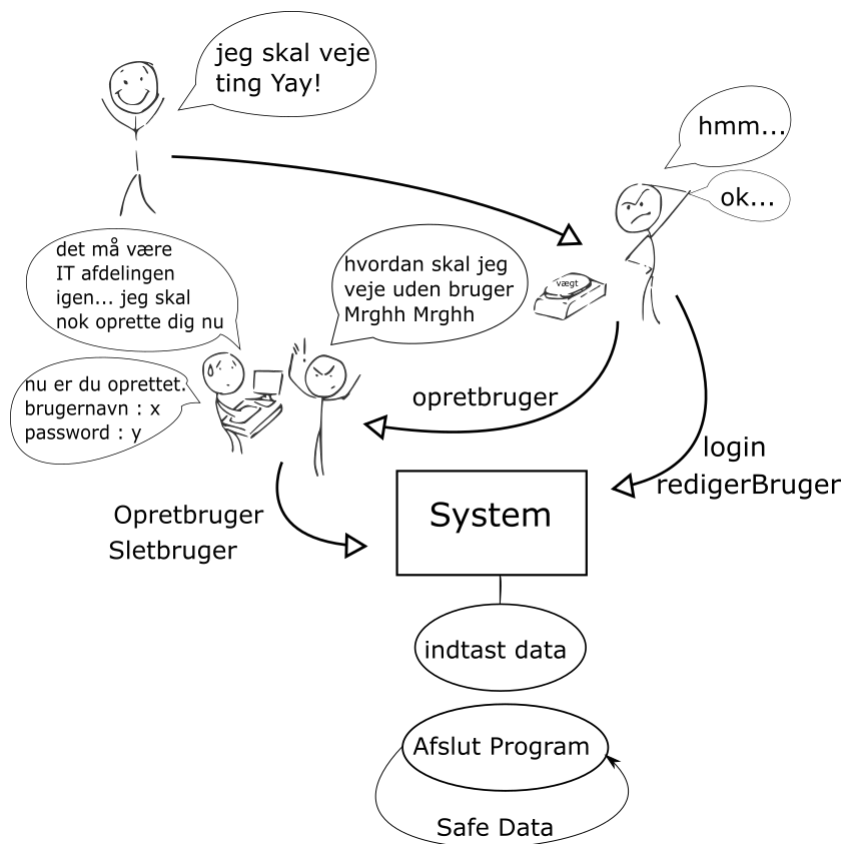
#### 2.1.2 Fully dressed use cases

Vi udbyggede den til at være fully dressed for at gennemtænke de forskellige scenarier.

Use Case Benyt Vægt	
ID	UC01
Brief description	En bruger logger ind på vægten og afvejer et produkt.
Primary actors	Bruger
Secondary actors	Leder og andre i virksomheden som har behov for at se hvem der har afvejet et bestemt materiale.
Preconditions	Systemet gemmer afvejningen. Brugeren logges af.
Main flow	Brugeren logger på vægten, indtaster batchnr. , afvejer produktet og godkender det. Dataen logges og brugeren logger ud.
Postconditions	Systemet har nu gemt brugeren så næste gang denne vil tilgå systemet bliver adgang givet ved indtastning af ID og password.
Alternative flow	1. Brugers log ind virker ikke og vægten kan derfor ikke benyttes. 2. Brugeren kan ikke godkende når vægten beder om det og afvejning kan ikke foretages. 3. Brugeren kan benytte vægten men det logges ikke når brugeren logges af.
Specielle krav	-

## 2.2 Rigt billede

Rige billeder har den fordel at man visuelt opnår en forståelse for samspillet mellem aktørerne i systemet og de involverede elementer.



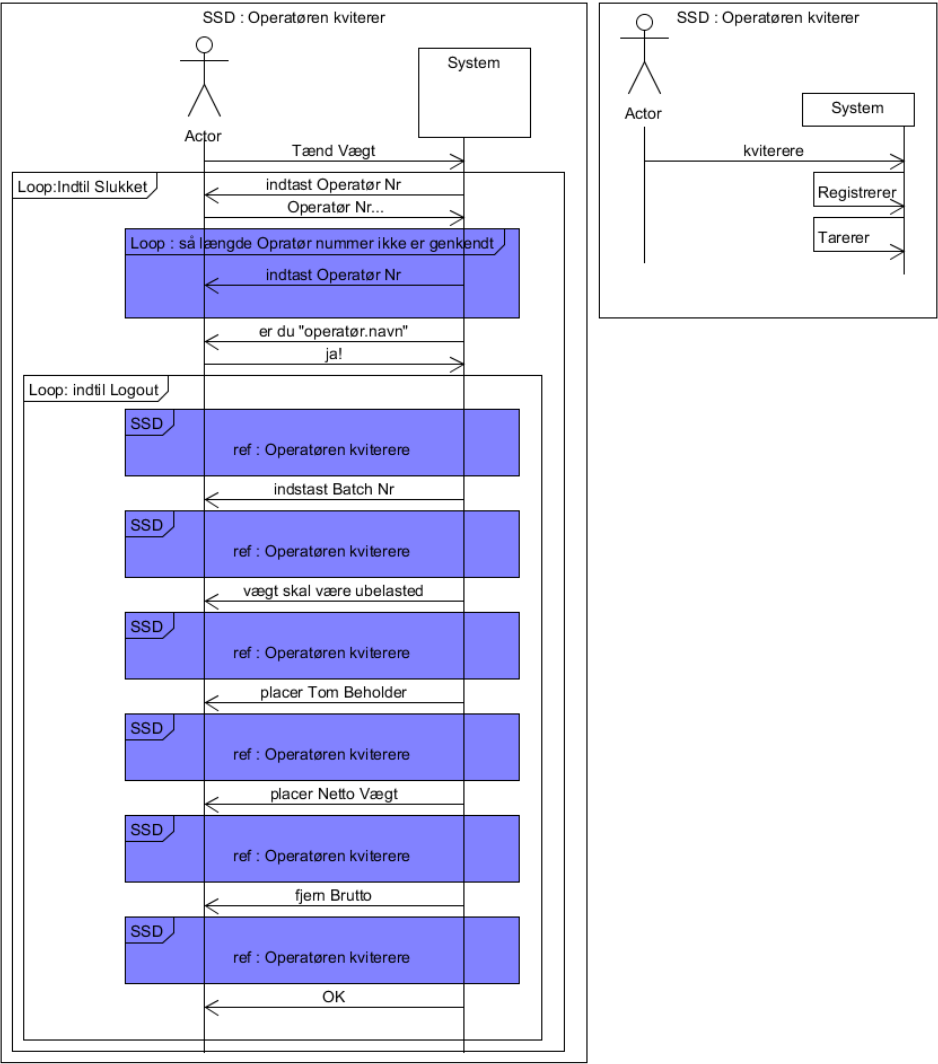
**Figur 2.1:** Rigt billede.

Dette billede viser tydeligt hvilke interessenter der er og hvordan samspillet mellem dem fungerer.

## 2.3 System Sekvens Diagram

Vi har lavet et system sekvens diagram for at få overblik over interaktionen mellem vægten og brugeren.





Figur 2.2: System Sekvens Diagram.

# KAPITEL 3

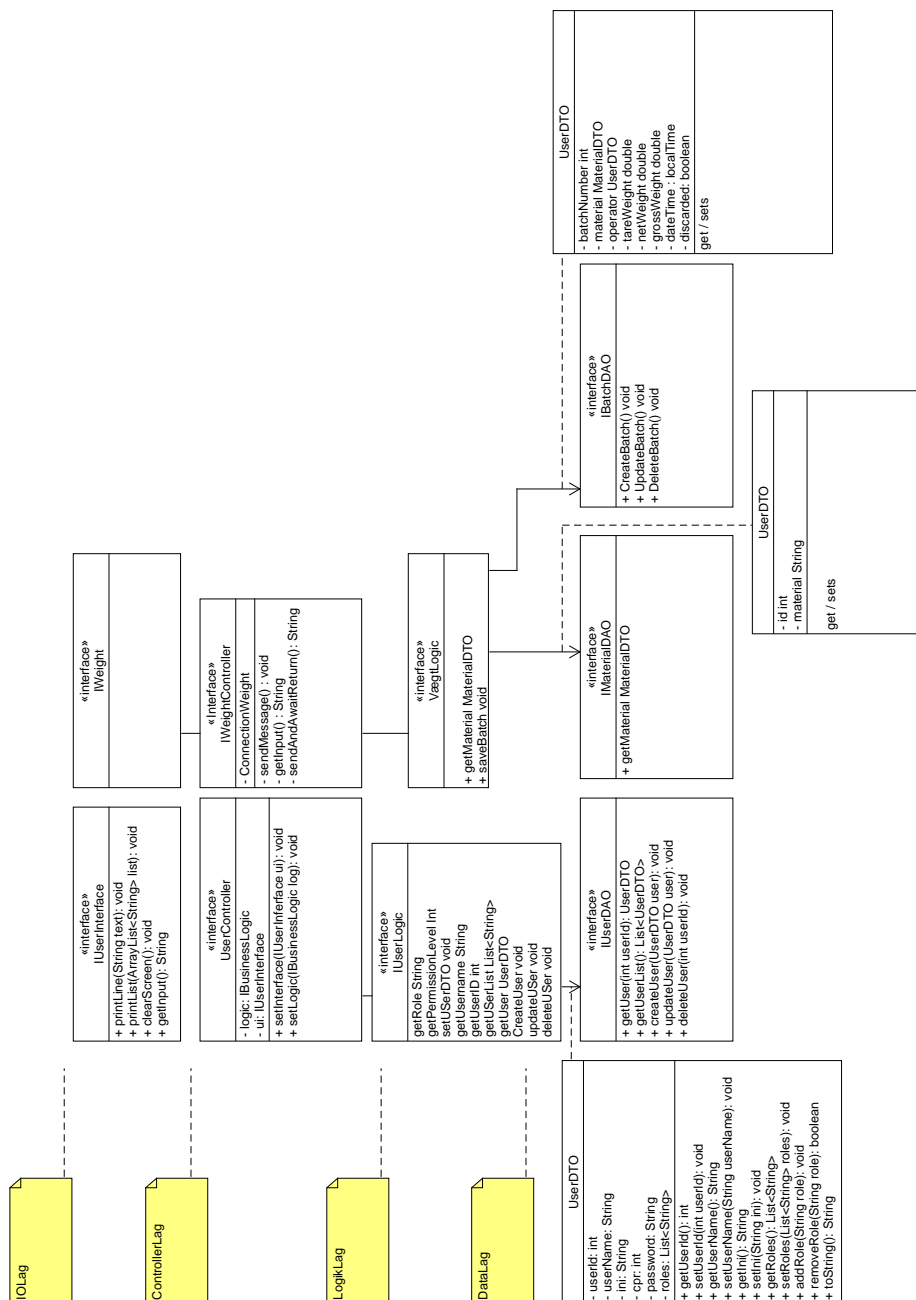
## Design

---

Vi byggede videre på arkitekturen fra CDIO1 med 3-lags modellen. Vi har således bevaret grænsefladelaget med TUI, et funktionalitetslag med program logikken samt et datalag med vores DTO og DAO. Vi har derudover indsat en controller som står for koordinere samspillet mellem grænsefladelaget og funktionalitetslaget.

Vi har i denne opgave udbygget CDIO1 programmet med at tilføje vægten i både controllerlaget, funktionalitetslaget og datalaget så vi bevarede vores arkitektur da vi har gennemtænkt arkitekturen for at gøre det nemt at vedligeholde og udvikle i fremtiden.

Ved at have fulgt 3-lags modellen har vi sørget for at optimere muligheden for at udskifte de enkelte dele uden at hele programmet går i stykker. Dette er yderligere underbygget ved at bruge interfaces så hvert lag kun taler med de andres interface lag. Vi sikrer på denne måde en lav kobling samt at alle udviklere tager stilling til de definerede metoder i interfacet.



**Figur 3.1:** Pakkediagram.

# KAPITEL 4

## Konklusion

---

Vi identificerede kun en use case i dette projekt nemlig Benyt Vægt. Denne analyse-rede vi både som brief og fully dressed. Derefter lavede en prioriteret kravliste ved hjælp af MoSCoW analysen. Vi fik implementeret alle vores Must-krav men ikke nogle af de andre krav pga. tidsmangel.

Vi fortsatte med et unified process forløb med inkrementel udvikling hvor vi så dette projekt som iteration 2. Vi indsatte vægt-klasser og -interface så det passede i den allerede designede arkitektur.

Vi skrev rapporten løbende undervejs og læste korrektur på hinandens afsnit.

Denne version af programmet gemmer dataen som objekt i en txt-fil. Men vi har forberedt at det er muligt at koble det til en database i en senere version.

Vi har bygget videre på CDIO1 i denne opgave og det har fungeret rigtig godt. Det grundlæggende arbejde i CDIO1 med at designe arkitekturen med tanke for videreudbygning har virket efter hensigten.

# Bibliografi

---

- [2219] Gruppe 22. *CDIO<sub>1</sub>*. <https://github.com/KasperBaun/Gruppe22>. 2019.
- [Bae] Baeldung. *Generate a Secure Random Password in Java*. URL: <https://www.baeldung.com/java-generate-secure-password>.

