



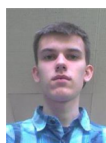
02324

VIDREGÅENDE PROGRAMMERING

CDIO - Del 2

Gruppe 17

s185118
Aleksander Lægsgaard Jørgensen



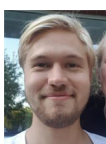
s175561
Andreas Østergaard Schliemann

s185096
Jonatan Amtoft Dahl



s164177
Josephine Weirøse

s185103
Søren Hother Rasmussen



s185121
Theodor Peter Guttessen

21. marts 2019

DTU Compute

Institut for Matematik og Computer Science

DTU Diplom

Center for Diplomingeniøruddannelse

Kode og Git: https://github.com/theogutt/17_CDIO2

Abstract

In the following report we describe our process behind designing and coding a program that can connect and interact with a scale. When run, it executes a predefined set of commands which are sent to the scale. These commands include asking for a user ID, batch number, tare etc. A user and a batch has been hardcoded into the program.

Indhold

1	Indledning	2
2	Timeregnskab	2
3	Krav	3
3.1	Vision	3
3.2	Kravliste	3
4	Analyse	4
4.1	Aktører	4
4.2	Use case	4
4.3	System sekvensdiagram	6
5	Design	7
5.1	Klassediagram	7
6	Implementering	7
7	Test	8
7.1	Positiv test	8
8	Projektplanlægning	8
8.1	Planlagt forløb	8
8.2	Reelt forløb	8
9	Konklusion	9
10	Bilag	10

1 Indledning

I dette projekt har vi lavet et program, der kan oprette en forbindelse til en vægt, fysisk eller virtuel, og derefter styre en afvejningsproces. Når vores program har forbindelse til vægten, kan det sende diverse kommandoer til den og modtage den værdi den vejer. Vi illustrerer også programmets struktur med diagrammer, såsom designklassediagram. Projektet er designet med fokus på at være objekt orienteret, og på at følge tre-lagsmodellen. Programmet er skrevet med Java som programmeringsprog og IntelliJ som udviklingsprogram.

2 Timeregnskab

Opgave	Aleksander	Andreas	Jonatan	Josephine	Søren	Theodor	Sum
Kode	4	1	5	0.5	1	3	14.5
Rapport	1	6	2	4.5	4	0	17.5
Sum	5	7	7	5	5	3	32

3 Krav

3.1 Vision

Kunden ønsker et afvejningssystem. Systemet skal guide en bruger igennem en afvejning på vægten, og skal registrere brugerID og batchID.

3.2 Kravliste

ID	Funktionelle krav
Must have	
M01	Systemet skal kunne åbne og lukke forbindelse til enten en virtuel eller en virkelig vægt, og disse skal kunne snakke frem og tilbage.
M02	Systemet skal kunne bruge kommandoerne S, T, D, DW, p111, RM20 8.
M03	Systemet skal kunne ud fra operatørnummre finde operatorens navn.
M04	Systemet skal kunne få data fra vægten og arbejde videre med den data.
Could have	
C01	Systemet skal kunne finde de forskellige operators navne samt deres ID fra en database.

ID	Ikke-funktionelle krav
Must have	
IF01	
IF02	

4 Analyse

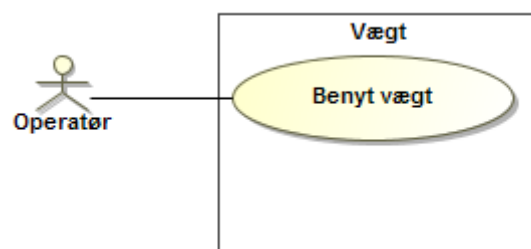
4.1 Aktører

I programmet er den eneste aktør operatøren/brugeren af vægten.

Aktørtype	Aktører
Primære	Bruger/operatør
Supporterende	-
Offstage	-

4.2 Use case

Den eneste use case der er i programmet er "benyt vægt".



Figur 1: Use Case diagram

Fully dressed

I arbejdet med use cases kan der benyttes tre forskellige grader af beskrivelse. Her er der tale om "brief", "casual" og "fully-dressed". Brief er den mindst omfattende og fully-dressed er den mest omfattende.

Use case: Benyt vægt
ID: UC01
Kort beskrivelse: Brugeren benytter programmet til at registrere data fra vægten og operatøren.
Primære aktøre: Operatør
Preconditions: Forbindelse til vægten er etableret.
Main flow: <ol style="list-style-type: none"> 1. Vægten spørger om operatørnummer 2. Operatøren indtaster brugernummer 3. Ved reelt operatør nummer, findes Operatørens navn og vises på vægten 4. Operatøren kvitterer for at det tilsvarende navn er korrekt

5. Vægten beder om, at operatøren indtaster batch nummer
6. Operatør indtaster batch nummer
7. Vægten instruerer om at vægt ikke må være belastet
8. Operatøren kvitterer
9. Vægten tareres
10. Operatøren instrueres om, at placere tom beholder på vægten
11. Operatøren kvitterer
12. Tara's vægt registreres
13. Vægten tareres
14. Operatøren instrueres i at placere beholder samt indehold på vægten
15. Operatøren kvitterer
16. Nettovægt registreres
17. Vægten tareres
18. Vægten instruerer om at fjerne brutto fra vægten
19. Operatøren kvitterer
20. Negativ bruttovægt registreres
21. Der udskrives OK
22. Operatøren kvitterer
23. Vægten tareres

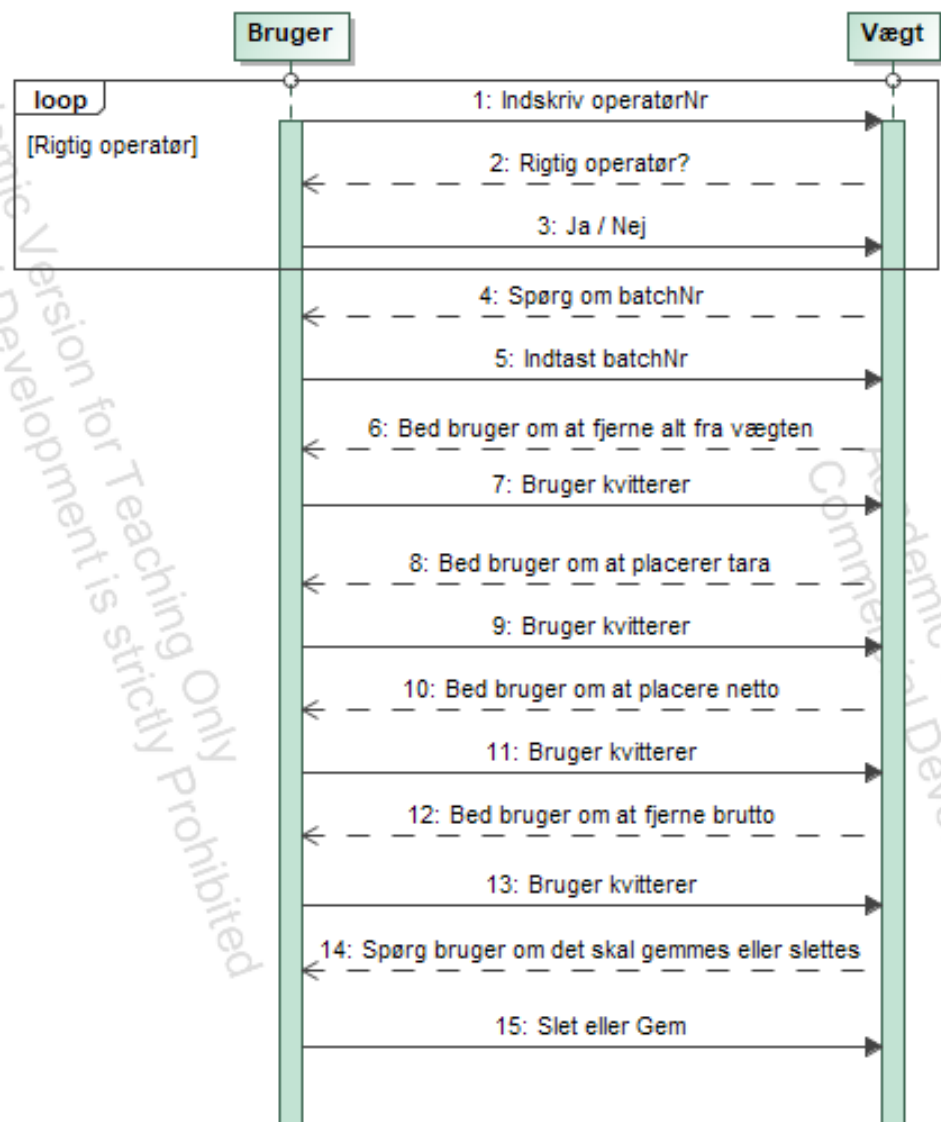
Postconditions: Objekt afvejet.

Alternative flows:

- 2.1 Operatøren indtaster ugyldigt brugernummer
- 2.1.1 Operatøren starter ved 2 igen indtil brugernummer er gyldig og forsætter så til 3.
- 4.1 Der er givet forkert navn.
- 4.1.1 Operatøren afviser og starter ved 2 igen.
- 6.1 Operatøren indtaster ugyldigt batch nummer
- 6.1.1 Operatøren starter forfra ved 6.

4.3 System sekvensdiagram

Dette system sekvensdiagram beskriver hvordan brugeren arbejder med vægten. Det meste der forgår, er at vægten beder brugeren at gøre noget, og brugeren trykker ok, når det er gjort. Systemet venter på at brugeren taster "ok" før at systemet går videre i programmet.

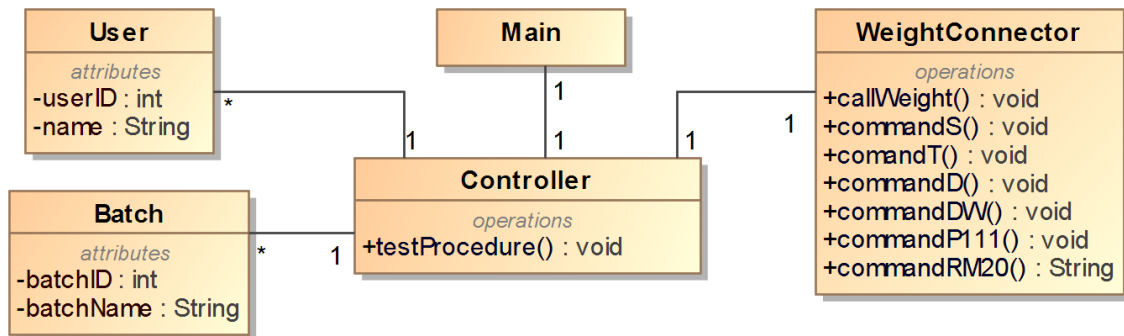


Figur 2: System sekvensdiagram

5 Design

5.1 Klassediagram

I vores designklassediagram kan man få et overblik over hele vores program. I diagrammet kan man se, hvordan klasserne relaterer til hinanden, deres metoder og attributter.



Figur 3: Designklassediagram

Se bilag for forstørret version.

6 Implementering

Strukturen i programmet har hovedsageligt taget udgangspunkt i tre-lagsmodellen, hvor vi har fokuseret på at holde de tre grupper for sig. Vores grænseflade i projektet har vi ikke selv lavet, det bliver nemlig lavet af den virtuelle vægt, da dette er det som brugeren kan interagere med. Funktionalitetslaget er vores to controllere, hvor klassen "controller" er klart den største del. Denne klasse sender de rigtige kommandoer på de rigtige tidspunkter, og det er denne klasse, som tager i mod alt data fra brugerfalden. Til sidst har vi datalaget og her er vores bruger og batch klasser. Hvis programmet bliver større vil selve database som vi bruger blive til datalaget, og de to klasser vil fungere mere som controllere.

7 Test

7.1 Positiv test

Det er blevet testet, at programmet følger de krav der er stillet. Dette er gjort ved at køre programmet fejlfrit flertallige gange.

Funktionelle krav	Positiv test	Bestået
M01	Systemet skal kunne åbne og lukke en forbindelse til en virtuel eller virkelig vægt, disse skal kunne snakke frem og tilbage.	Bestået
M02	Systemet skal kunne bruge kommandoerne S, T, D, DW, p111, RM20 8.	Bestået
M03	Systemet skal kunne ud fra operatørnummre finde operatorens navn.	Bestået
M04	Systemet skal kunne få data fra vægten og arbejde videre med den data.	Bestået
C01	Systemet skal kunne finde de forskellige operators navne samt deres ID fra en database.	Ej Bestået

8 Projektplanlægning

8.1 Planlagt forløb

Vores plan var at starte med at finde ud af, hvilke klasser, der kunne være i vores program. Derefter skulle vi lave et simpelt program, som kunne oprette forbindelse til vægten. Når dette var lavet, ville vi udbygge programmet til også at kunne sende kommandoer og modtage information fra vægten. Mens programmet ville blive arbejdet på, ville vi også arbejde på at lave sekvensdiagram, klassediagram med mere.

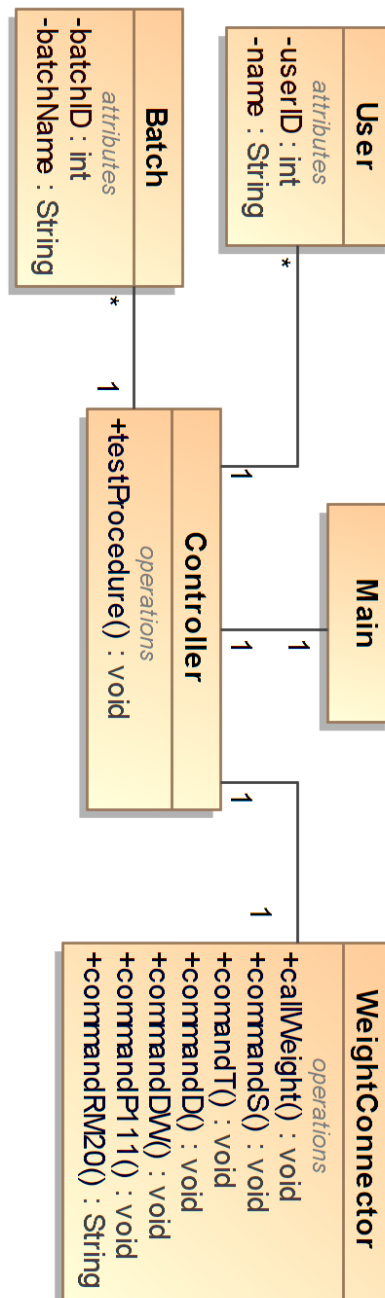
8.2 Reelt forløb

Da selve projektet var relativt småt, i forhold til andre CDIO projekter. Dette havde den konsekvens at det var markant nemmere at holde sig til den originale plan. Vi startede først med at få et program, som igennem konsollen i IntelliJ kunne sende kommandoer til vægten og modtage svar. Herefter begyndte dele af gruppen at lave rapport, og resten lavede det resterende kode, så programmet den rigtige samtale med en bruger. Vi fik både startet og afsluttet projektet i god tid, og processen blev ikke ramt af større problemer såsom sygdom.

9 Konklusion

Vi kan hermed konkludere at vores produkt blev mere eller mindre som vi havde forventet. Vores program kan have de samtaler med vægten som opgavebeskrivelsen kræver, og vi har lavet loops til at give brugerne flere chancer for at indskrive data såsom operatorkode. Projektet kan også forholdsvis nemt blive opgraderet til at bruge databaser eller andre opgraderinger. Processen har som sagt været ret smertefri, og vi har overholdt alle de dele som vi satte os for til start. I forhold til det forrige CDIO projekt i dette semester, så har dette projekt nok været mere planlagt, og vi har haft bedre tid til at lave det.

10 Bilag



Figur 4: Designklassediagram