CDIO 2 forår 2016

02324 Videregående programmering

Projekt Titel: CDIO 2

Gruppe nr: 24

Afleveringsfrist: Lørdag 09/04-2016 05:00

Denne rapport er afleveret via Campusnet

Denne rapport indeholder 17 sider inkl. denne side.

S153391,Wichmann Alexander

S153646 Einarsson, Kristófer

S153086 Jensen, Magnus

S144827 Jørvad, Morten

S154088 Hazara, Mohammad

S144826 Frantsen, Tobias

Indholdsfortegnelse

Timetabel 3
Indledning 3
Kravspecifikation 4
Analyse 5
Use case 5
BCE model 11
Klassediagram 12
Test 13
Diskussion 16
Konklusion 17

Timetabel

Dato	Deltager	Design	Impl.	Test	Dok.	I alt
09/04/2016	**Magnus Jensen**	3	8	2	5	**18**
09/04/2016	**Morten Jørvad**	2	8	2	5	**17**
09/04/2016	**Kristofer Mar Einarson**	1	2	2	5	**10**
09/04/2016	**Alexander Wichmann**	2	8	2	5	**17**
09/04/2016	**Tobias Frantsen**	0	3	2	5	**10**
09/04/2016	**Mohammad Hazara**	1	1	3	5	**10**
Sum		**9**	**32**	**13**	**30**	**82**

Indledning

Formålet med denne rapport er at skabe overblik over den vægt simulator, som ønskes implementeret.
Projektet er baseret på at få alle funktioner som er på den fysiske vægt, ind på den vægt simulator som ønskes laves. Derudover er der blevet lavet en grafisk brugergrænseflade, som minder om den rigtige vægt, således at det gør vægt simulatoren så realistisk som muligt. Systemet skal håndtere kommunikation mellem en server, hvor vægt simulatoren er serveren og en klient, som tilslutter sig til serveren igennem en specifik defineret port. Når forbindelsen mellem serveren og klient er blevet lavet, gøres det muligt for klienten at bruge vægten igennem TCP.
Vægtens data gemmes ikke nogle steder og dataen er lagret direkte i systemet, og bliver nulstillet hver gang der slukkes for serveren.

Kravspecifikation

Funktionelle Krav

1. Der skal implementeres en Vægtsimulator vha. en grafisk brugergrænseflade.
2. Vægtsimulatoren skal simulere en Mettler BBK vægt.
3. Der benyttes SISC-protokollen til kommunikation med vægten.
4. Følgende 8 kommandoer skal implementeres: S, T, D, DW, P111, RM20, B og Q.
5. Simulatoren skal lytte på port 8000, som er dens default-værdi. Denne værdi skal kunne overskrives med et selvvalgt portnummer, der angives i kommandolinjen ved opstart.
6. Man skal kunne taste tara på vægt-simulatoren, som nulstiller vægtens visning
7. Man skal kunne ændre brutto belastning på vægten.
8. Efter at vægten har modtaget RM20 8 ordren, skal betjeneren af GUIen have mulighed til at indtaste et svar.
9. Når programmet afsluttes, skal alt været lukket ned

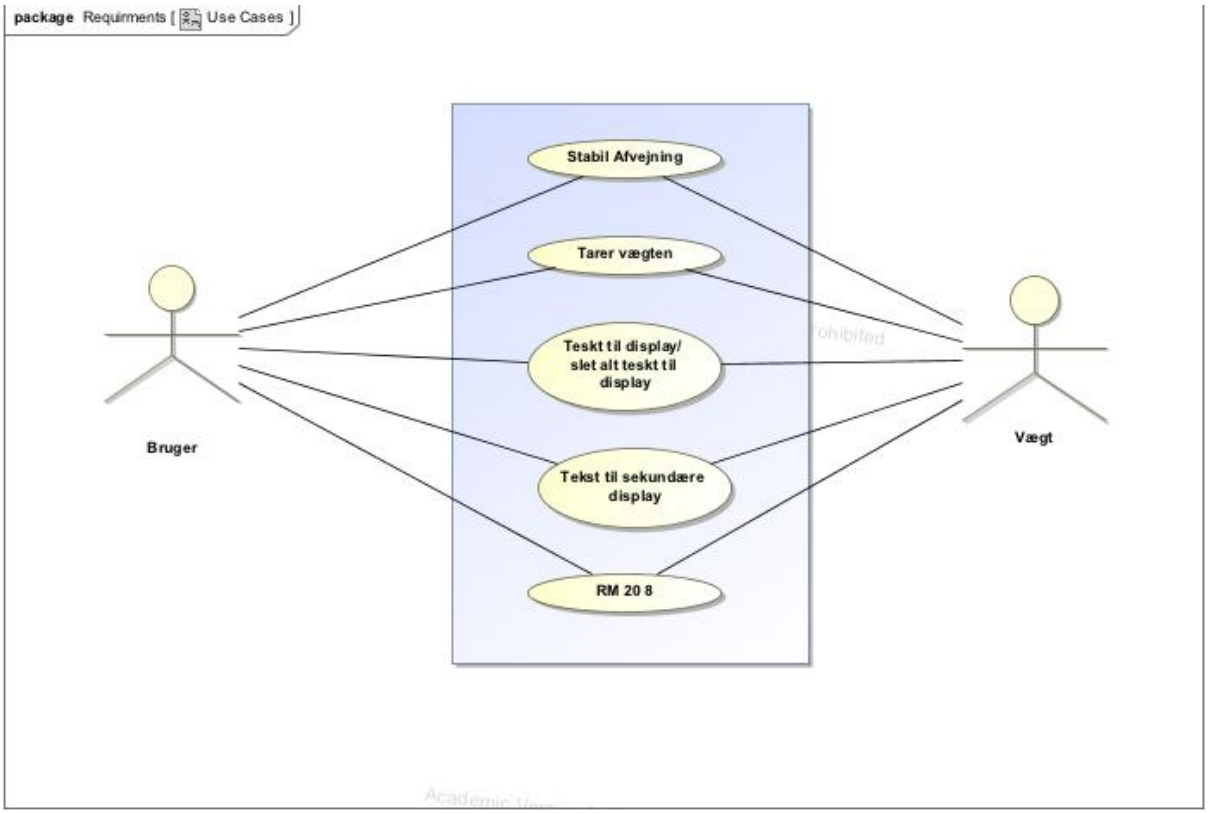
Non-funktionelle krav

10. Menu sendt til klient, således at bruger kan se mulighederne for vægten

Analyse

Use case

Herunder ses en oversigt over forskellige scenarier, en bruger af systemet kan støde på.



Denne usecase beskriver hvordan en stabil afvejning bliver foretaget i vores system.

Use Case:	**Stabil afvejning**
ID:	1
Brief Description	Brugerne bliver præsenteret for et vindue, hvor det er gjort muligt at indtaste de forskellige kommandoer der kan anvendes på vægten. Systemet sender kommandoer over en socket forbindelse på en selvvalgt port. Vægten modtager kommandoen for stabil afvejning og der sendes resultatet tilbage til vinduet og bliver præsenteret for brugeren.
Primary actors:	Operatøren
Secondary actors:	Vægten
Preconditions:	Systemet er startet og operatøren har oprettet forbindelse til vægten.
Main flow:	Operatøren indtaster sin kommando for stabil afvejning (S cr lf) Vægten modtager kommandoen og sender sit svar tilbage. Den stabil afvejning bliver præsenteret for operatøren.
Postconditions:	Til slut er der blevet taget en afvejning på vægten, som er blevet præsenteret for operatøren.
Alternative flows:	Hvis operatøren er kommet til at skrive noget forkert i punkt 1, skal der komme en meddelelse fejl op, og han skal prøve at indtaste sin kommando igen.

Denne usecase beskriver hvordan tarer funktionen fungerer i vores system.

Use Case:	**Tarer vægten**
ID:	2
Brief Description	Brugerne bliver præsenteret for et vindue, hvor det er gjort muligt at indtaste de forskellige kommandoer der kan anvendes på vægten. Systemet sender kommandoer over en socket forbindelse på en selvvalgt port. Vægten modtager kommandoen for at tarere vægten og der sendes resultatet tilbage til vinduet og bliver præsenteret for brugeren.
Primary actors:	Operatøren
Secondary actors:	Vægten
Preconditions:	Systemet er startet og operatøren har oprettet forbindelse til vægten.
Main flow:	Operatøren indtaster sin kommando for Tarer vægten (T cr lf) Vægten modtager kommandoen og sender sit svar tilbage. Vægten på emballagen bliver præsenteret for operatøren.
Postconditions:	Til slut er der blevet taget en afvejning af emballagen på produktet som er på vægten, som er blevet præsenteret for operatøren.
Alternative flows:	Hvis operatøren er kommet til at skrive noget forkert i punkt 1, skal der komme en meddelelse fejl op, hvor der evt. kan stå hvilke kommandoer der kan anvendes i dette program, og han skal prøve at indtaste sin kommando igen.

Denne usecase beskriver hvordan man kan ændre eller slette.

Use Case:	**Tekst til display / slet alt tekst i display**
ID:	3
Brief Description	Brugerne bliver præsenteret for et vindue, hvor det er gjort muligt at indtaste de forskellige kommandoer der kan anvendes på vægten. Systemet sender kommandoer over en socket forbindelse på en selvvalgt port. Vægten modtager kommandoen for at skriv i vægtens display og der sendes resultatet tilbage til vinduet og bliver præsenteret for brugeren.
Primary actors:	Operatøren
Secondary actors:	Vægten
Preconditions:	Operatøren har forbundet sig til vægten
Main flow:	1. Operatøren indtaster sin kommando for at skrive i displayet (D “tekst” cr lf) 2. Vægten modtager kommandoen og sender sit svar tilbage. 3. Display teksten er blevet ændret og bliver præsenteret for operatøren.
Postconditions:	Hvis der anvendes kommandoen D “tekst” cr lf så er der blevet ændret i display teksten på vægten. Modsat hvis følgende kommando DW cr lf, er blevet anvendt sletter den hele displayet på vægten
Alternative flows:	1. Hvis operatøren er kommet til at skrive noget forkert i punkt 1, skal der komme en meddelelse fejl op, hvor der evt. kan stå hvilke kommandoer der kan anvendes i dette program, og han skal prøve at indtaste sin kommando igen. 2. Hvis operatøren vil slettet vægtens display, skal der sendes følgende kommando DW cr lf, i stedet for D “tekst” cr lf.

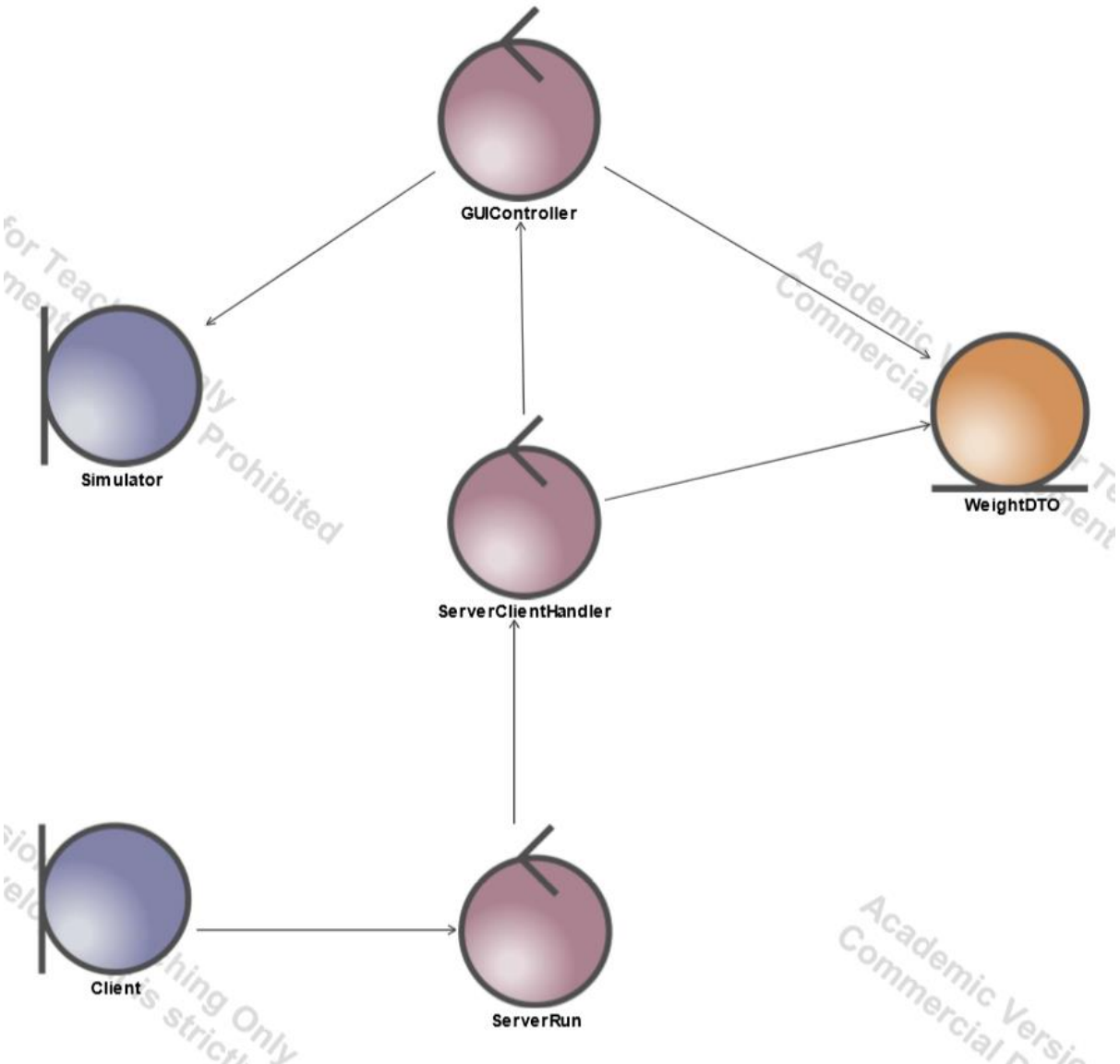
Denne usecase beskriver flowet når operatøren gerne vil skrive en tekststreng som skal stå på vægtens sekundære display.

Use Case:	**Tekst til sekundære display**
ID:	4
Brief Description	Brugerne bliver præsenteret for et vindue, hvor det er gjort muligt at indtaste de forskellige kommandoer der kan anvendes på vægten. Systemet sender kommandoer over en socket forbindelse på en selvvalgt port. Vægten modtager kommandoen for at skrive i vægtens sekundære display og der sendes resultatet tilbage til vinduet, som bliver præsenteret for brugeren.
Primary actors:	Operatøren
Secondary actors:	Vægten
Preconditions:	Systemet er startet, og operatøren har oprettet forbindelse til vægten.
Main flow:	1. Operatøren indtaster sin kommando for at skrive i det sekundære displayet (P111 “tekst” cr lf) 2. Vægten modtager kommandoen og sender sit svar tilbage. 3. Display teksten er blevet ændret og bliver præsenteret for operatøren.
Postconditions:	Der er blevet skrevet en tekst til det sekundære display på vægten
Alternative flows:	1. Hvis operatøren er kommet til at skrive noget forkert i punkt 1, skal der komme en meddelelse fejl op, hvor der evt. kan stå hvilke kommandoer der kan anvendes i dette program, og han skal prøve at indtaste sin kommando igen.

Denne usecase beskriver funktionen “RM20 8” kommandoen, når operatøren gerne vil spørge serveren og noget og have et svar baseret på spørgsmålet.

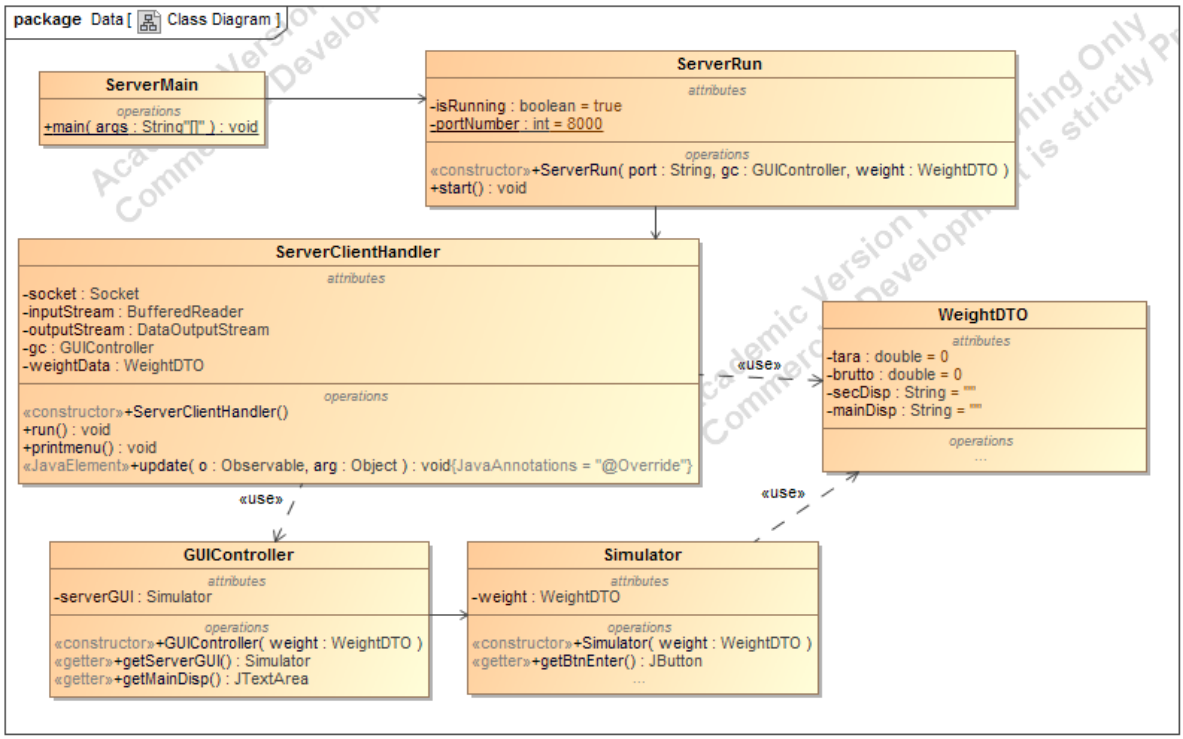
Use Case:	**RM 20 8**
ID:	5
Brief Description	Brugeren sender kommandoen RM20 8” *tekst”,* den fungerer således, at efter brugeren har sendt en besked til vægten (serveren), så skal man vente indtil han modtager en besked fra vægten (serveren) i mellemtiden kan brugeren ikke anvende andre kommandoer.
Primary actors:	Operatøren
Secondary actors:	Vægten
Preconditions:	Systemet er startet og operatøren har oprettet forbindelse til vægten.
Main flow:	1. Operatøren sender kommandoen RM20 8 efterfulgt af en besked til vægten. 2. Operatøren venter til at vægten svarer, i mellemtiden kan operatøren ikke anvende nogle andre kommandoer 3. “Brugeren” af vægten indtaster en besked på vægten. 4. Operatøren modtager beskeden
Postconditions:	Der er blevet sendt og modtaget en besked mellem klienten og serveren
Alternative flows:	1. Hvis operatøren er kommet til at skrive noget forkert i punkt 1, skal der komme en meddelelse fejl op, hvor der evt. kan stå hvilke kommandoer der kan anvendes i dette program, og han skal prøve at indtaste sin kommando igen.

BCE model



Ovenfor ses vores BCE-model, i hvilken vi har angivet vores klassers roller i forhold til hinanden. Som vores to boundaries ses hhv. Simulator og Client. I vores tilfælde repræsenterer Client en forbindelse oprettet vha. Putty. Simulator repræsenterer den GUI, vi har valgt at implementere. Da både klienten og serveren skal kunne kommunikere med *ServerClientHandler*-klassen, der håndterer, hvad der skal ske med de commands, der modtages fra enten serveren eller klienten. *ServerClientHandler* indeholder et objekt af typen *WeightDTO*. Dette bliver oprettet i main klassen siden passet ned i ServerClientHandlers constructor.

Klassediagram

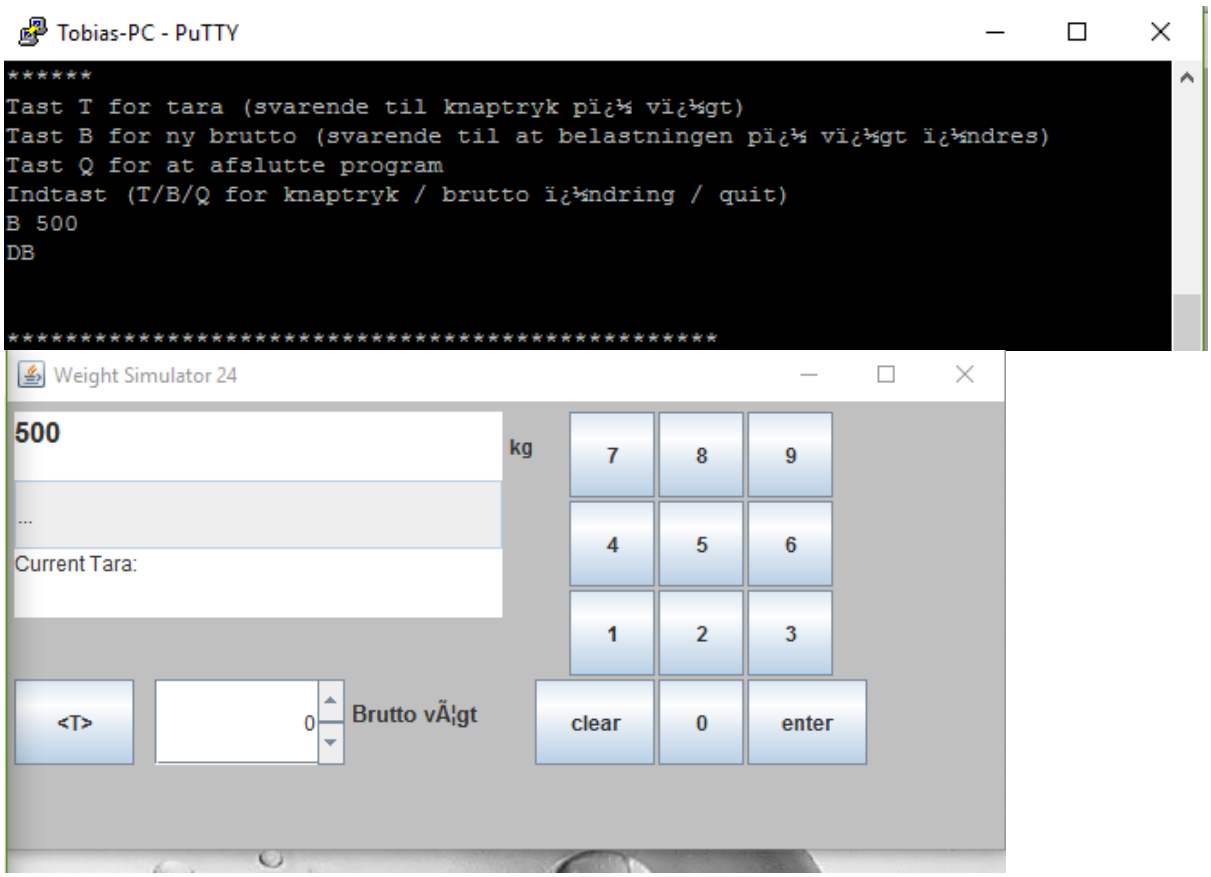


Ovenstående figur viser vores klassediagram, som er blever udbygget på baggrund af kravspecifikation og usecases. På klassediagrammet kan det ses, at der er blevet valgt at lave en GUI, dette er blevet valgt for at visualisere din fysiske vægt, i forhold til at køre den fra en konsol.

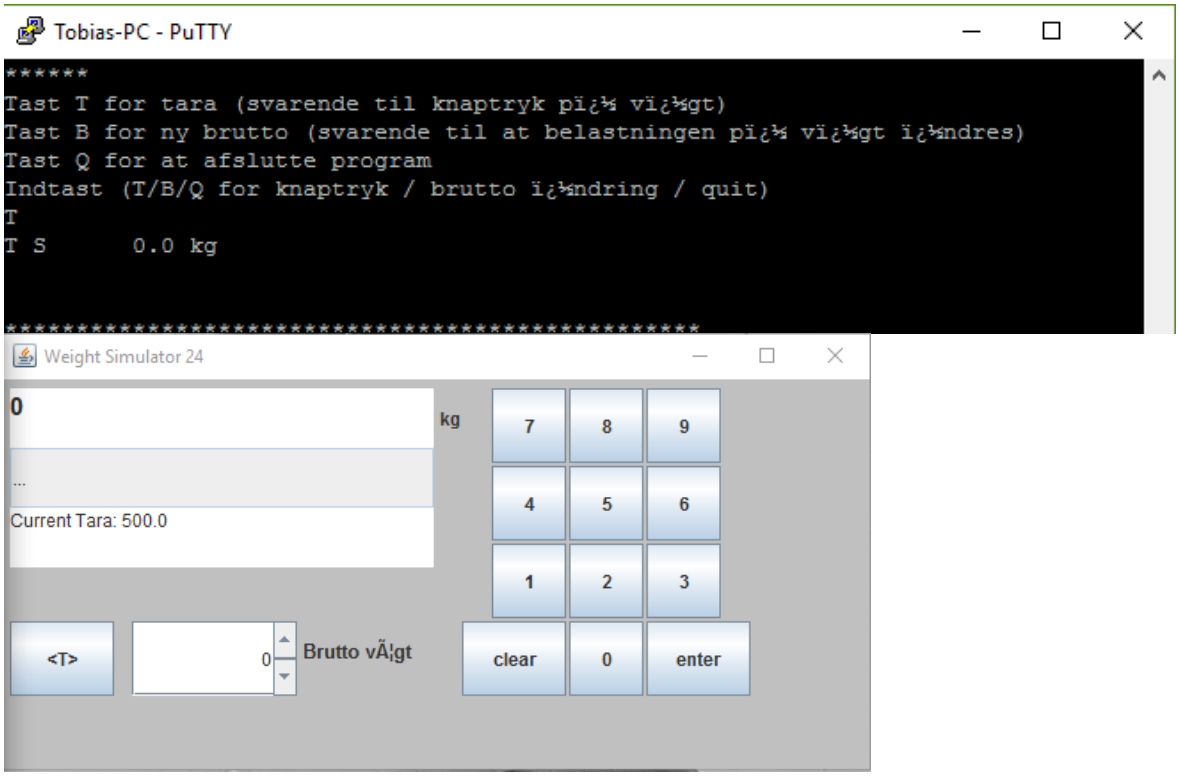
Test

Under dette afsnit, kan det ses hvilke metode der er blevet implementeret i vores system. Udklippene viser hvordan vi har testet alle kald mellem vores client og vores simulator. Vi har også implementeret to metoder B og Q som ud over standard kaldene.

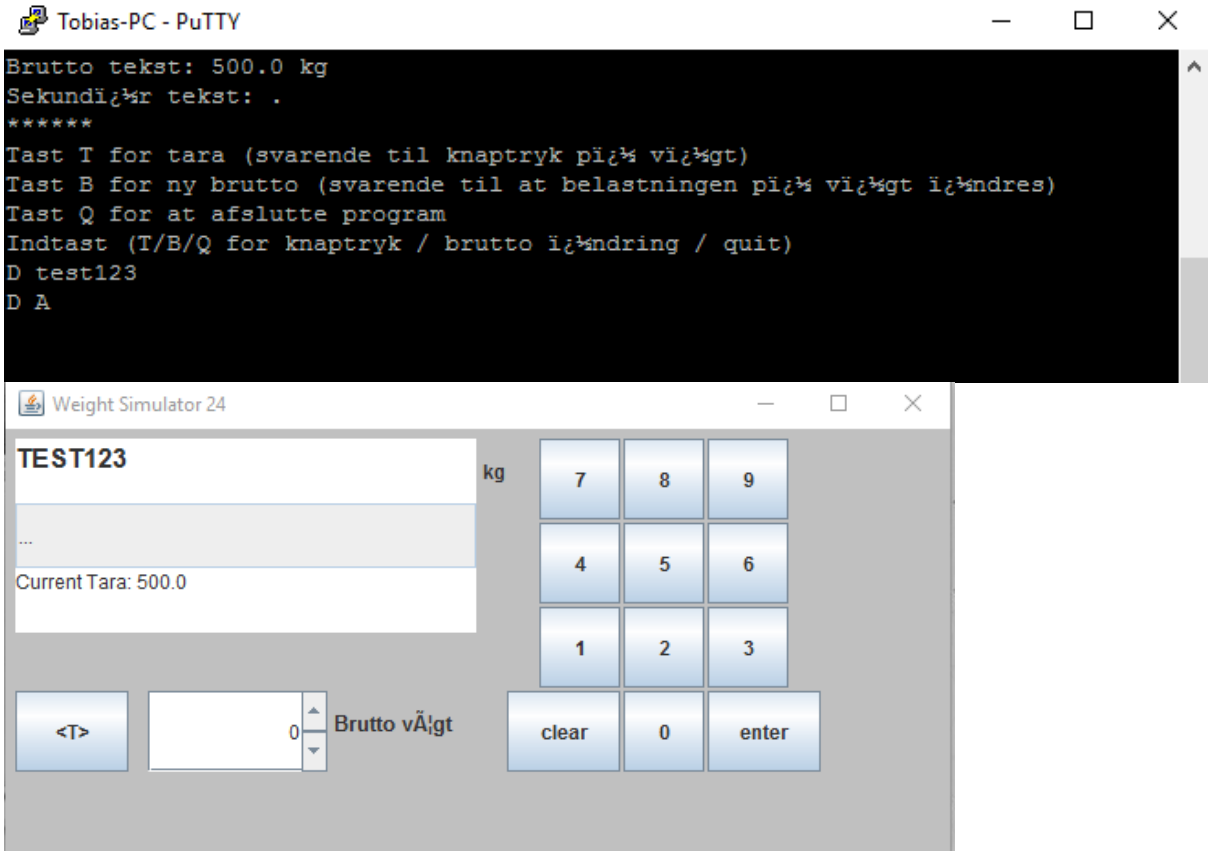
B:



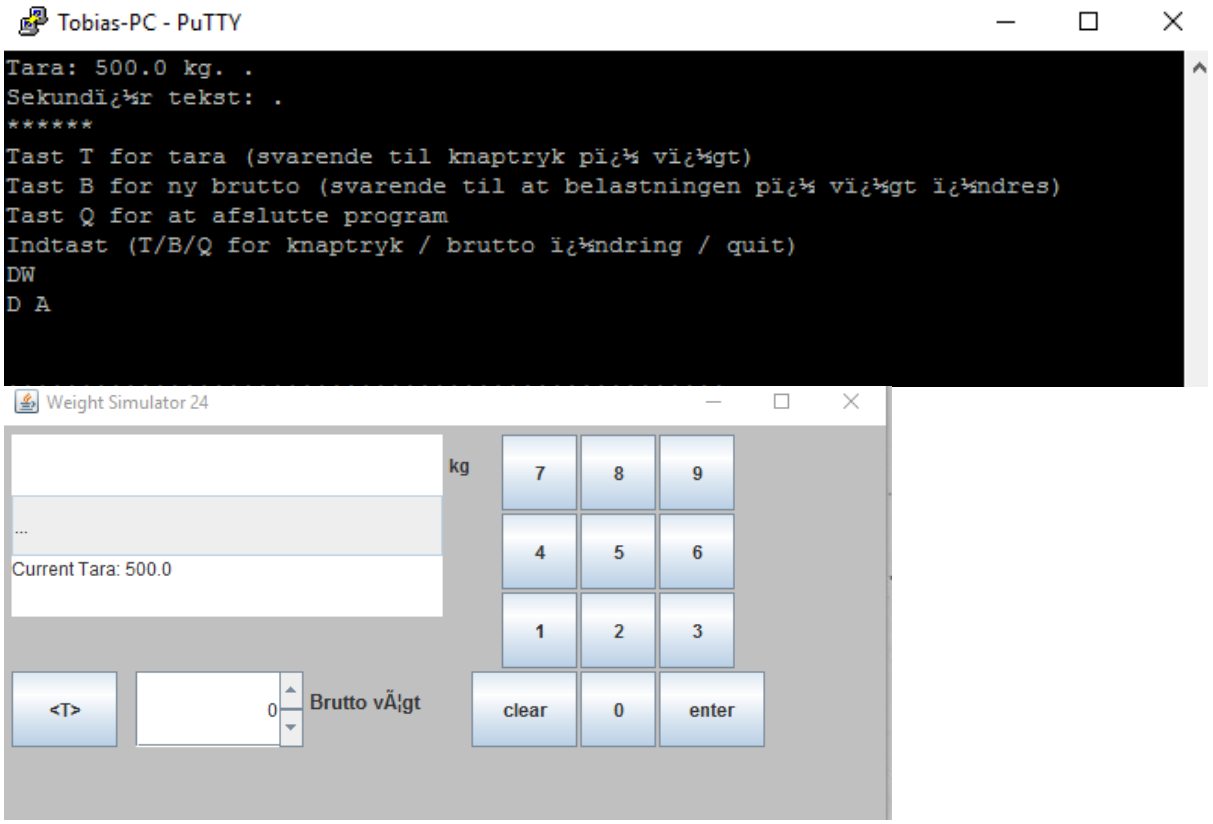
T :


Tobias-PC - PuTTY
******
Tast T for tara (svarende til knaptryk pï¿½ vï¿½gt)
Tast B for ny brutto (svarende til at belastningen pï¿½ vï¿½gt ï¿½ndres)
Tast Q for at afslutte program
Indtast (T/B/Q for knaptryk / brutto ï¿½ndring / quit)
T
T S 0.0 kg
Weight Simulator 24
0
kg
...
Current Tara: 500.0
7
8
9
4
5
6
1
2
3
<T>
0
Brutto vÃ¦gt
clear
0
enter

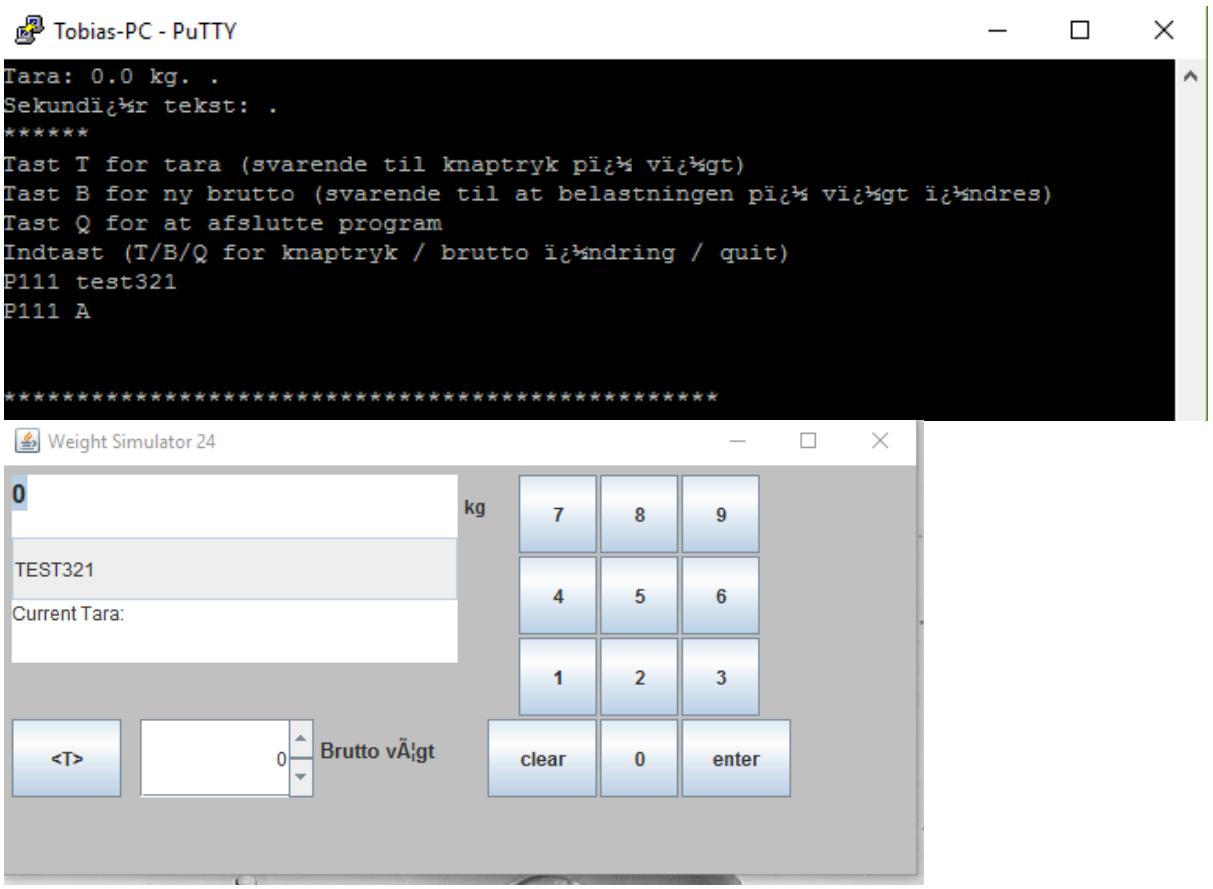
D:


Tobias-PC - PuTTY
Brutto tekst: 500.0 kg
Sekundï¿½r tekst: .
******
Tast T for tara (svarende til knaptryk pï¿½ vï¿½gt)
Tast B for ny brutto (svarende til at belastningen pï¿½ vï¿½gt ï¿½ndres)
Tast Q for at afslutte program
Indtast (T/B/Q for knaptryk / brutto ï¿½ndring / quit)
D test123
D A
Weight Simulator 24
TEST123
kg
...
Current Tara: 500.0
7
8
9
4
5
6
1
2
3
<T>
0
Brutto vÃ¦gt
clear
0
enter

DW:

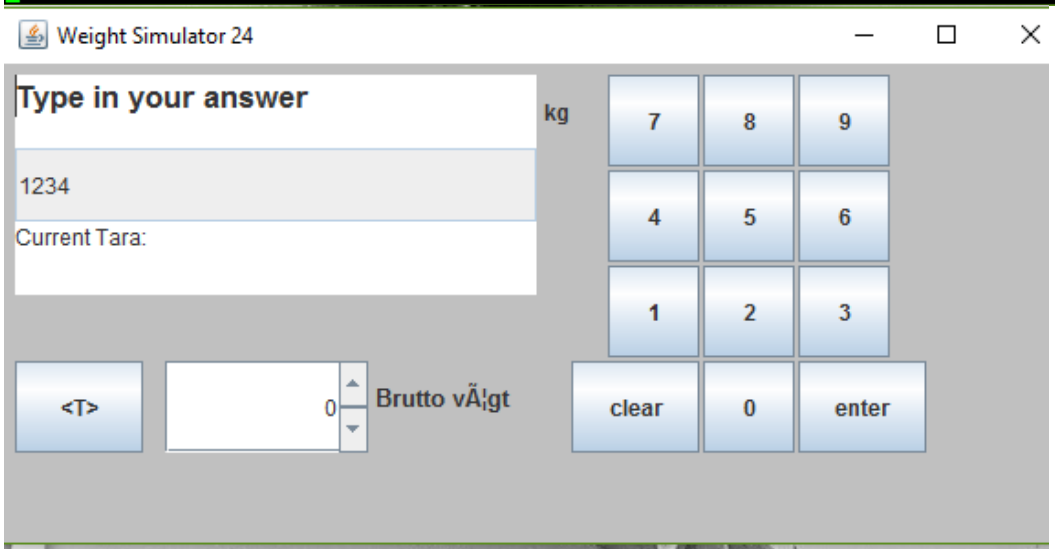

Tobias-PC - PuTTY
Tara: 500.0 kg. .
Sekundï¿½r tekst: .
******
Tast T for tara (svarende til knaptryk pï¿½ vï¿½gt)
Tast B for ny brutto (svarende til at belastningen pï¿½ vï¿½gt ï¿½ndres)
Tast Q for at afslutte program
Indtast (T/B/Q for knaptryk / brutto ï¿½ndring / quit)
DW
D A
Weight Simulator 24
kg
7
8
9
4
5
6
1
2
3
...
Current Tara: 500.0
<T>
0
Brutto vÃ¦gt
clear
0
enter

P111:


Tobias-PC - PuTTY
Tara: 0.0 kg. .
Sekundï¿½r tekst: .
******
Tast T for tara (svarende til knaptryk pï¿½ vï¿½gt)
Tast B for ny brutto (svarende til at belastningen pï¿½ vï¿½gt ï¿½ndres)
Tast Q for at afslutte program
Indtast (T/B/Q for knaptryk / brutto ï¿½ndring / quit)
P111 test321
P111 A
****
Weight Simulator 24
0
kg
7
8
9
4
5
6
1
2
3
TEST321
Current Tara:
<T>
0
Brutto vÃ¦gt
clear
0
enter

RM20 8 :

```
**************************************************
Denne vï¿½gtsimulator lytter pï¿½ ordrene
S, T, D, DW, RM20 8 .... , B og Q
******
Brutto: 500.0 kg
Tara: 500.0 kg. .
Sekundï¿½r tekst: .
******
Tast T for tara (svarende til knaptryk pï¿½ vï¿½gt)
Tast B for ny brutto (svarende til at belastningen pï¿½ vï¿½gt ï¿½ndres)
Tast Q for at afslutte program
Indtast (T/B/Q for knaptryk / brutto ï¿½ndring / quit)
RM20 8 1234
```




```
Tobias-PC - PuTTY
Brutto tekst: 0.0 kg
Sekundï¿½r tekst: .
******
Tast T for tara (svarende til knaptryk pï¿½ vï¿½gt)
Tast B for ny brutto (svarende til at belastningen pï¿½ vï¿½gt ï¿½ndres)
Tast Q for at afslutte program
Indtast (T/B/Q for knaptryk / brutto ï¿½ndring / quit)
RM20 8 1234
RM20 A 56789


**************************************************
```

Diskussion

Der har været mange udfordringer og mange løsningsforslag i gruppen om, hvordan vi skulle løse denne opgave. Vi diskuterede en del om hvad der skulle være klient og hvad der skulle være server. Vi fandt så ud af, at vægten var server og klienten så var brugeren af vægten. Vi valgte også i vores opgave at oprette en grafisk brugergrænseflade som har givet et mere visuelt billede af hvordan den fysiske vægt rent faktisk ser ud.

Konklusion

Ud fra arbejdet igennem dette projekt kan der konkluderes, at det er gjort muligt at få konstrueret et system, der kan simulere en fysisk vægt. Brugeren kan få adgang til dette produkt igennem en socket forbindelse på en specifik IP og port. Efter brugeren er tilsluttet til vægten er det således muligt at lave de samme funktioner på vægt simulatoren, som på den fysiske vægt.
Derudover er der lavet specielle funktioner som skal simulere at vægt bliver lagt på vægten. Denne funktion kan også tilgås gennem en grafisk brugerflade som er blevet lavet, for at efterligne den fysiske vægt så meget som muligt.