

# CDIO del 2, forår 2015

*02325 Datakommunikation*

**Projektnavn:** CDIO\_datakom2

**Emne:** FTP-klient og WCU

**Afleveringsfrist:** Tirsdag d. 14/04-2015 18:00

## **Gruppemedlemmer (studienummer, efternavn, fornavn):**

s144830, Sepstrup, August



s123157, Adamsen, Frederik



s144846, Eriksen, Robert



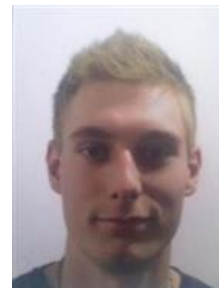
s144845, Ørnby, Victor



s144844, Thomsen, Mads



s144858, Hansen, Nicklas



Rapporten er afleveret elektronisk via Campusnet og der skrives derfor ikke under. Rapporten indeholder 18 sider inkl. denne side og evt. bilag.

Code repo. URL: [https://github.com/nicklas669/CDIO\\_datakom2/commits/master](https://github.com/nicklas669/CDIO_datakom2/commits/master)

# Indholdsfortegnelse

[Indholdsfortegnelse](#)

[Indledning](#)

[Analyse](#)

[Kravspecifikationer FTP-klient](#)

[Funktionelle krav](#)

[Kravspecifikationer WCU](#)

[Funktionelle krav](#)

[Tolkning af krav til opgave 2](#)

[Design](#)

[Klassediagram FTP-klient](#)

[ER Diagram for database til WCU](#)

[Flowchart for WCU](#)

[Login](#)

[Config](#)

[Service](#)

[Best case](#)

[Item kan ikke afvejes](#)

[Administration](#)

[Ny varetype](#)

[Ny beholder](#)

[Opfyldning](#)

[Implementation og tests](#)

[FTP](#)

[Sensorer](#)

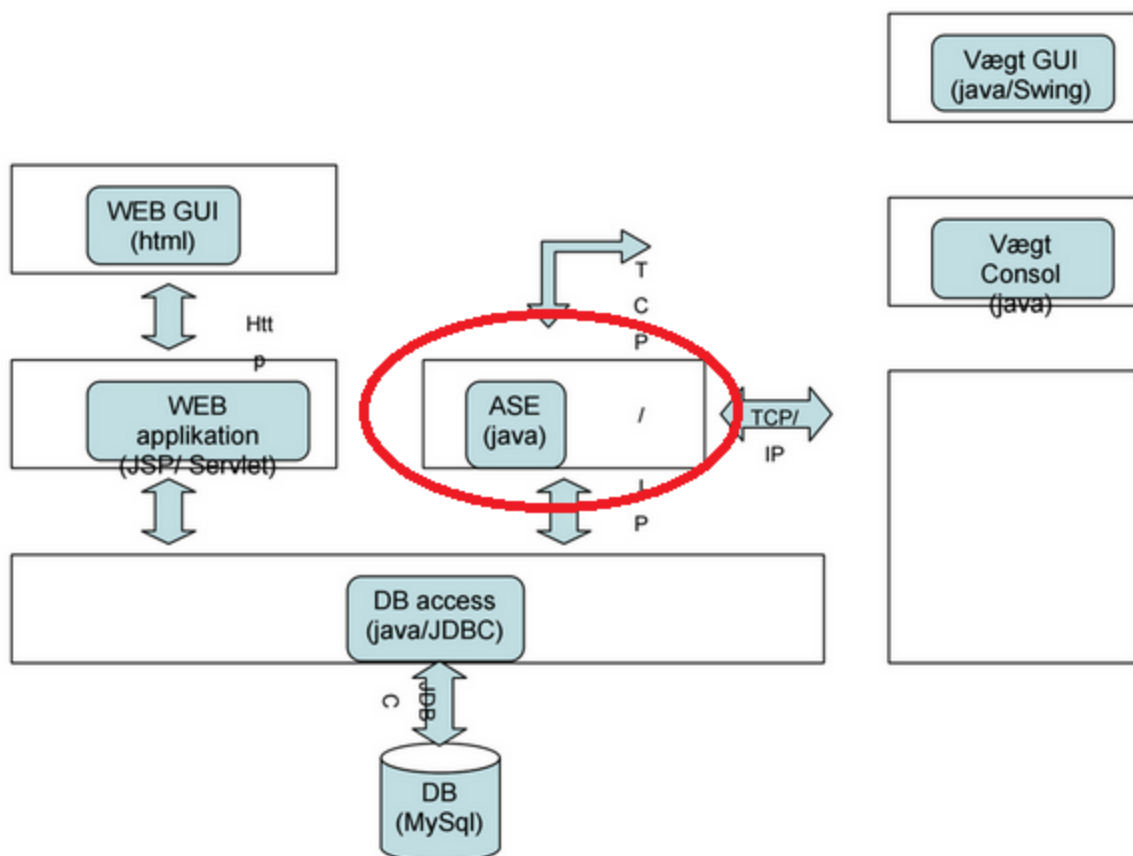
[Kilder/litteraturliste](#)

[Konklusion](#)

## Indledning

Projektet er lavet som CDIO del 2 i kurset 02325 Datakommunikation. I projektet skal afleveres en rapport, dog er det blevet tilkendegivet at rapporten blot bør være en "lille rapport". Opgaven bundet ud i to applikationer - en simplificeret FTP-klient og en såkaldt WCU - Weight Control Unit.

Dette system skal på sigt udbygges og integreres med dele af CDIO i andre kurser for til sidst at have en fuld applikation hvorfra man kan fjernbetjene en vægt og lave et recept-system med afvejning. Forneden ses et diagram over det samlede system hvoraf vi har udviklet nogle af delene i andre kurser allerede mens andre laves i fremtidige projekter. Det er markeret med den røde cirkel, hvilken del af systemet, dette projekt omhandler. Weight Control Unit (WCU) der laves i dette projekt svarer til ASE (Afvejnings Styrings Enhed) på diagrammet:



## Analyse

Projektene startede med en analyse af hvad systemet skulle kunne, i henhold til Unified Process. Først arbejdede vi på at fastslå projektets rolle i forhold til det overordnede CDIO projekt på tværs af fagene, som også beskrevet i indledningen.

Denne manøvre gav det fornødne overblik til at bestemme hvilken rolle netop dette projekt skulle understøtte. Derefter analyserede vi det udleverede oplæg og opstillede krav, der skulle opfyldes. Specielt for dette projekt var, at vi skulle udvikle to forskellige applikationer og ikke kun én. Der vil i rapporten derfor blive lavet UML-diagrammer og krav for begge projekter. På baggrund af denne analyse lavede vi følgende domænemodeller:

## Kravspecifikationer FTP-klient

FTP'en er blevet udarbejdet ud fra CDIO 2 oplægget, hvor det bliver forklaret at det første program skal være en FTP klient der skal kunne forbinde sig til en anden FTP server, hvor det så skal være muligt at hente filer fra den og sende kommandoer til den.

### Funktionelle krav

1. FTP-klienten skal kunne forbinde til en FTP-server på Zybo-boardet og derfra hente en eller flere filer.
  - 1.1. Der skal implementeres LIST kommando til at printe working directory's indhold.
  - 1.2. Der skal implementeres RETR kommando til at hente en fil fra serveren.
  - 1.3. Der skal eventuelt implementeres flere kommandoer alt efter hvordan serveren er sat op.
2. Når en kommando sendes, præsenteres svaret for brugeren.
3. kunne sende kommandoer og modtage svar fra Zybo-boardet og sende kommandoer til Zybo-boardets sensorer.
4. Have en menu hvor brugeren kan vælge hvad denne vil gøre.

## Kravspecifikationer WCU

Kravspecifikationerne er givet i opgaveoplægget og er gentaget her:  
WCU, Weight Control Unit, er et java-program der fungerer som vægts klient. Klienten skal kommunikere med vægten via TCP-socket

### Funktionelle krav

- Der skal være et identifikations program for afvejning processen
- Der skal være mulighed for at indtaste varenummer
- Der skal være mulighed for at bruge varenummer til at slå op i en database og udskrive varenavn.
- Operator skal kunne identificeres ved hjælp af et id.
- Man skal kunne tare, påfylde vægt, afmåle nettovægt.
- Der skal være mulighed for at afskrive forbrugt råvare i databasen og indfør afvejning i loggen.
- Der skal være mulighed for at quite programmet og starte processen forefra.

## Tolkning af krav til opgave 2

Dette projekt vil udover at opfylde de opstillede krav fra opgavebeskrivelsen, så har vi som gruppe udarbejdet yderligere krav til projekts form og indhold. Overordnet er disse krav inddraget for at gøre projekt mere erhvervsorienteret, og dermed sætte et personligt præg på programmet. Følgende afsnit vil give en kort beskrivelse af de nye krav.

Det første krav vi havde til projektet var at gøre databasen online. Skiftet fra lokal til online database går i naturlig forlængelse af et andet CDIO projekt vi arbejder med i Indledende Databaser.

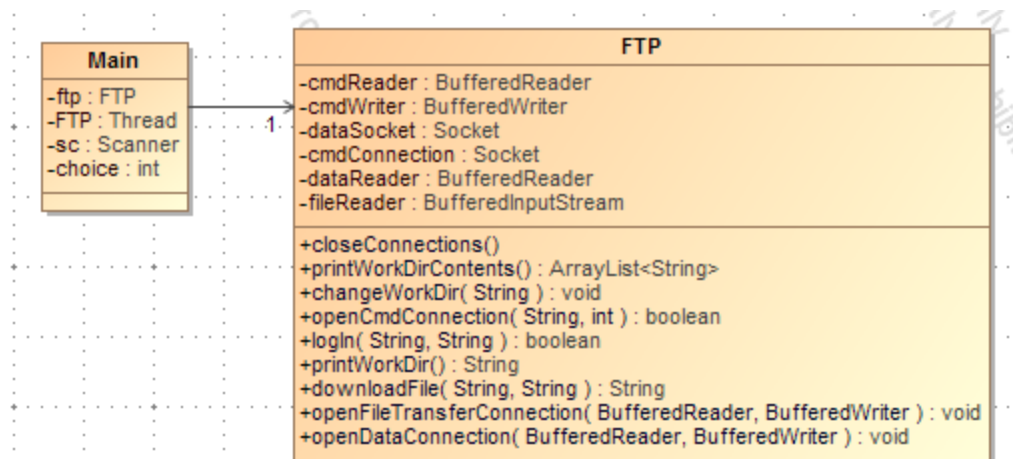
Et andet krav var at gøre programmet mere erhvervsorienteret. Tanken om at programmet skulle bruges udenfor et lukket laboratorium, var en filosofi som prægede alt fra konstruktion af databasen til udformningen af programmets grafiske interface. Bl.a har vi inddraget at der er en levetid på varen, i varelageret. Formålet med denne var at give brugeren de bedste rammer for varelagerhåndtering.

Det var tanken at vi ville implementere en tabellen Price til databaser, hvor prisen for de forskellige varer ville være opbevaret. Dette ville have muliggjort at man meget præcist kunne beregne profitten, samt mulighed for at sammensætte livstid med evt. tilbud.

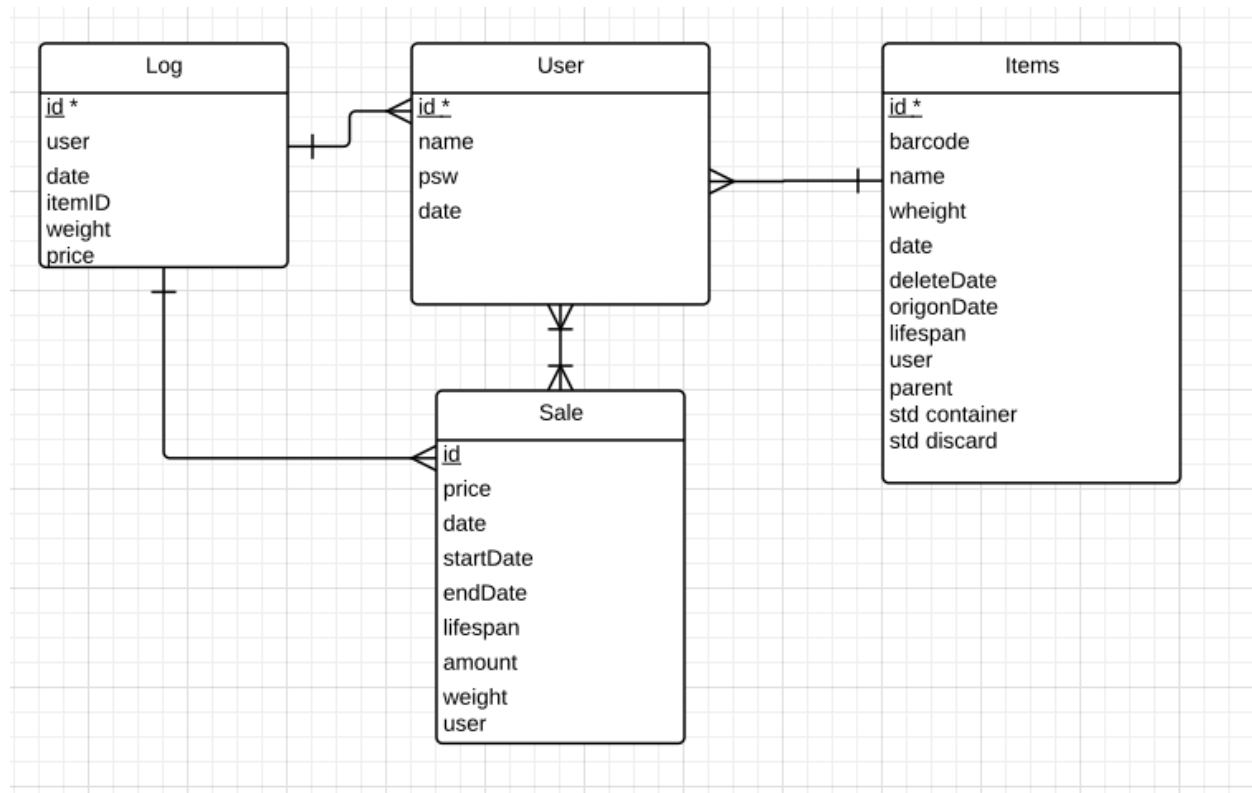
Desværre har vi ikke nået at få implementere prissætning af varen til nærværende iteration af projektet. Dog står ideen stadig som en testamente til et mere erhvervsorienteret CDIO projekt, hvor afvejning stadig kan bruges i et laboratorium, men også ville passe til en grønthandler.

## Design

### Klassediagram FTP-klient



## ER Diagram for database til WCU



Overstående ER diagram viser database strukturen for projektet. Primary keys er vist ved understregning, og \* betyder at der er auto inkrementering.

Som nævnt i kravspecifikationen afsnittet adskiller dette projekts database sig fra projektoplægget ved at være en online database. Vi vil ikke i denne rapport udlede de mange forskelle mellem en lokal og online database. Dog er det værd at bemærke at en online database gør informationen lettere tilgængelig, da den kan tilgås online. Dette åbner naturligvis op for en række sikkerheds perspektiver hvad angår opbevaringen af data. Projektet har ikke videre taget højde for disse sikkerhedsaspekter, andet end at kræve password og id, ved opkobling til databasen.

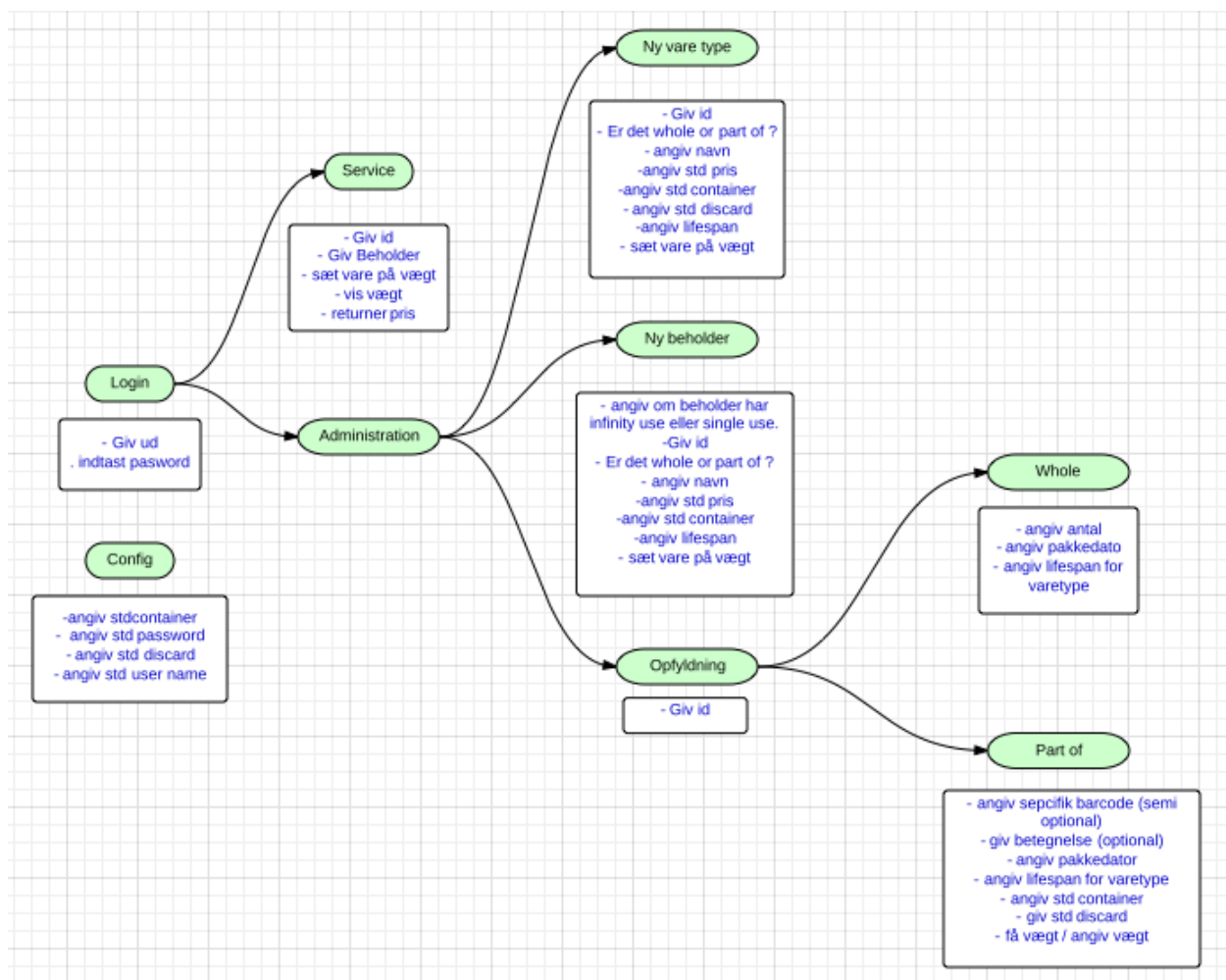
Databasen er yderligere blevet udvidet til at inkludere en tabel for Sale. Selvom Sale delen er desværre ikke blevet implementeret tids nok til indgå i denne iteration af CDIO.

I Items tabellen har vi inkluderet data om hvor gammel en vare kan blive (lifespan). Dertil er pakkedatoen for varen også inkluderet (origonDate). Fælles for ovenstående key's er at de gør varelager håndtering lettere.

Item tabellen fremkommer også key'en std container. Tanken var at have en standard beholder man kunne bruge til at afveje (fx en kop). Vi havde så yderligere defineret hvorvidt denne container kunne genbruges (en porcelæns kop) eller den var single user only (en pap kop). Desværre stødte vi ind i nogle problemer, ift. at opfyldning med std container. Derfor er denne funktion ikke fuldt ud implementeret til denne iteration.

Overordnet er databasen lavet for at skabe meningsfuld data omkring varerne og dets funktioner, der i sidste ende kan skabe værdi for brugeren. Desværre er flere af disse tabeller ikke fuldt ud implementeret i dette projekt.

## Flowchart for WCU

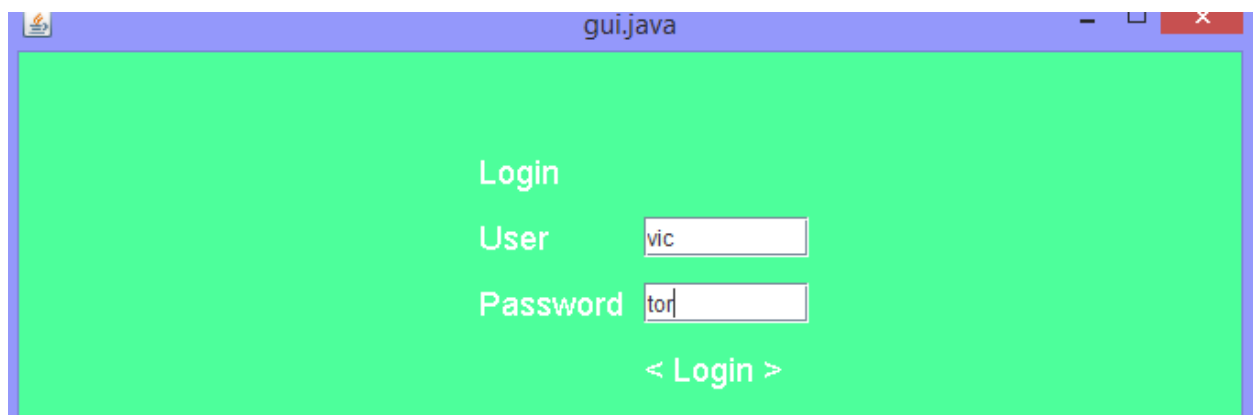




Overstående flowchart viser hvilke valg brugeren har til rådighed ved opstart af WCU'en. De grønne kasser er en dynamisk menu for brugeren, og den tilhørende tekstboks er de funktioner der tilknytter sig det respektive menupunkt. Følgende afsnit vil kort gennemgå hvert punkt og tale om fordele og ulemper ved implementering af disse.

## Login

I login menuen skal man indtaste et korrekt id og et password for at tilgå funktionaliteten. Ved indtastning af forkert id og password, bliver felterne blanke. I den nuværende form er der ikke specifikke krav til sikkerheden af password. Dette vil dog blive ændret ved 3 ugers projektet.



## Config

Config er den anden mulighed fra start menuen. Man behøver ikke password for at tilgå funktionaliteten. Baggrunden for dette er at programmet er designet til at bruges lokalt, og man kan som sådan ikke ødelægge databasen, ved at tilgå config. I Config kan man justere på std container og std discard.

## Service

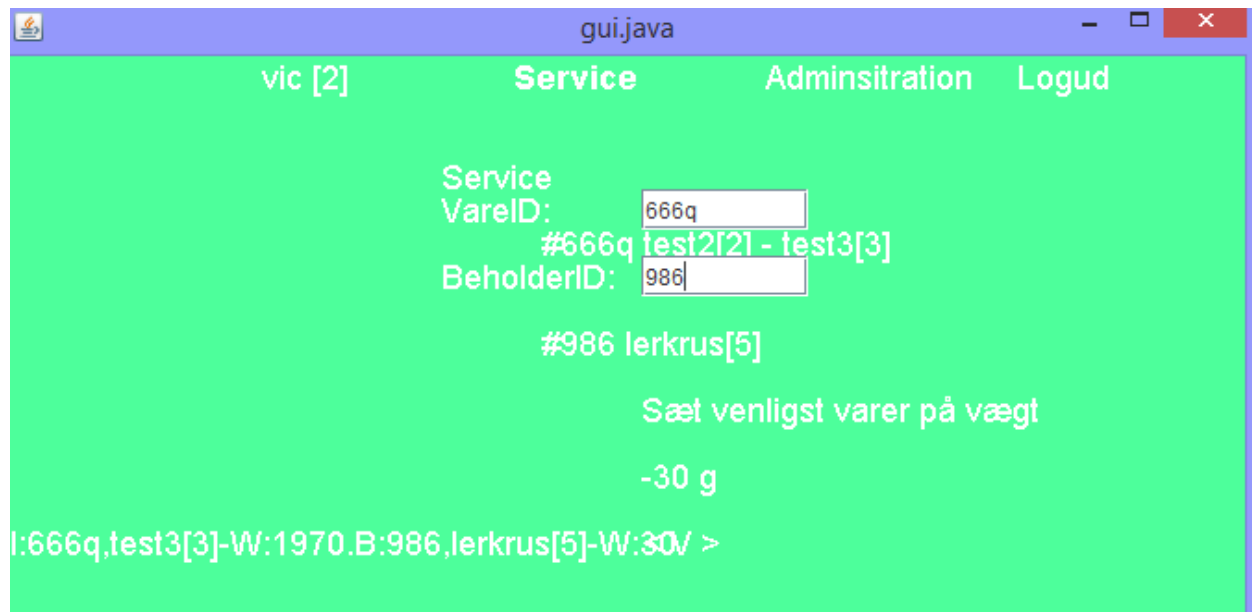
Service er til afvejning af vægt. Programmet antager at operatør har kendskab til samtlige vare ID. Der kan overordnet ske 2 forskellige scenarier for operatøren ved brug af service menuen.

### Best case

Det første scenarium kan kategoriseres som et best case scenarium. I denne case indtaster operatøren et vare ID som kan afvejes fx id 666q. Grunden til at nogle af varene ikke kan vejes er at de er kategoriseret i container, dvs. enheder, og

derfor giver det ikke mening at veje. Eksempelvis giver det langt bedre mening, at afskrive containeren "en dåse cola" i databasen fremfor at veje dåsen og derefter afskrive vægten.

Nedenfor har vi gennemgået et eksempel for indtastning ved id 666q. Man kan aflæse af GUI'en at programmet automatisk tarer for beholderen der knytter sig til denne vare her vist ved - 30 gram.



For at afveje varen, skal man åbne vægt simulatoren der blev lavet i forbindelse med CDIO 1.

For at veje en vare indtaster man b på Vægt simulatoren efterfulgt af vægten på varen.

Vi har desværre haft problemer ved denne operation, da programmet connecter med vægt simulatoren en gang i sekundet. Det besværliggør i den grad indtastning af vægt, hvilket især er problematisk hvis operatøren skal indtaste gram.

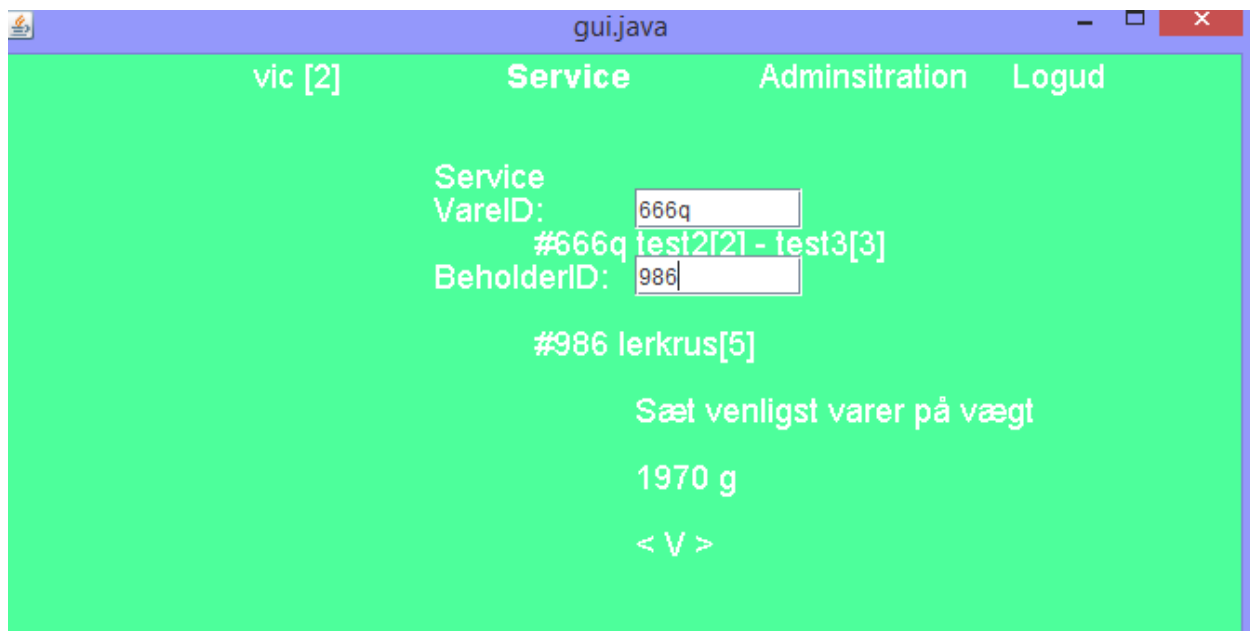
Heldigvis knytter dette problem sig kun til brug af vægt simulatoren, og ikke hvis programmet blev opkoblet til en fysisk vægt. Her ville man blot lægge varen på vægten og dette ville blive afspejlet på den GUI'en.

```

=====Weight Simulator =====
Display text:
Weight: 0.000 (<+0.000 tara> kg)
Hit 'B' - Brutto update <Simulating a mass is placed>
Hit 'Q' - Quit program <"Q" on wire>
Remote socket disconnected or in error
=====Weight Simulator =====
Display text:
Weight: 0.000 (<+0.000 tara> kg)
Hit 'B' - Brutto update <Simulating a mass is placed>
Hit 'Q' - Quit program <"Q" on wire>
Connected to IP: 127.0.0.1, port: 40968
Remote socket disconnected or in error
Remote socket disconnected or in error
Connected to IP: 127.0.0.1, port: 40969
=====Weight Simulator =====
Display text:
Weight: 0.000 (<+0.000 tara> kg)
Hit 'B' - Brutto update <Simulating a mass is placed>
Hit 'Q' - Quit program <"Q" on wire>
Remote socket disconnected or in error
Connected to IP: 127.0.0.1, port: 40970
=====Weight Simulator =====
Display text:
Weight: 0.000 (<+0.000 tara> kg)
Hit 'B' - Brutto update <Simulating a mass is placed>
Hit 'Q' - Quit program <"Q" on wire>
Remote socket disconnected or in error
=====Weight Simulator =====
Display text:
Weight: 0.000 (<+0.000 tara> kg)
Hit 'B' - Brutto update <Simulating a mass is placed>
Hit 'Q' - Quit program <"Q" on wire>
Connected to IP: 127.0.0.1, port: 40971
Remote socket disconnected or in error
Connected to IP: 127.0.0.1, port: 40972
Remote socket disconnected or in error
=====Weight Simulator =====
Display text:
Weight: 0.000 (<+0.000 tara> kg)
Hit 'B' - Brutto update <Simulating a mass is placed>
Hit 'Q' - Quit program <"Q" on wire>
Remote socket disconnected or in error
Connected to IP: 127.0.0.1, port: 40973

```

Vægten bliver afspejlet på GUI'en. Vi indtastede i dette tilfælde 2 kilo, hvilket automatisk fratrækker de 30 gram fra beholderen til 1970 gram på displayet. Vægten på varen bliver automatisk fratrullet fra varelager.



### Item kan ikke afvejes

Det andet scenarium er at operatør skriver et ID som er opgjort i beholder og ikke i vægt. Man ville normalvis prompte operatøren for at afvejning dermed ikke er mulig, men denne funktion har vi ikke fået implementeret til denne iteration af

programmet. Det samme gør sig gældende for ugyldigt item id (I dette tilfælde vil den simpelt bare arbejde videre, med sidst gyldige ID, eller slet ikke, hvis dette ikke eksisterer - indtil brugeren indtaster et, som eksisterer.



Når items er blevet tilføjet (< V > eller enter), vil de blive tilføjet en arraylist, som fungerer som "indkøbsvogn".

Ved tryk på <tjek varer ud> bliver arrayet clearet, og rowsne bliver opdateret. Hvis beholder er relevant (B:) bliver denne også. deletedDate er indikator i tabellen for, hvorvidt instancen er "forbrugt".

Der bliver tillige tilføjet til log.

Brugeren bliver returneret til starten af 1.1.1, efter updates, insert's er færdige.

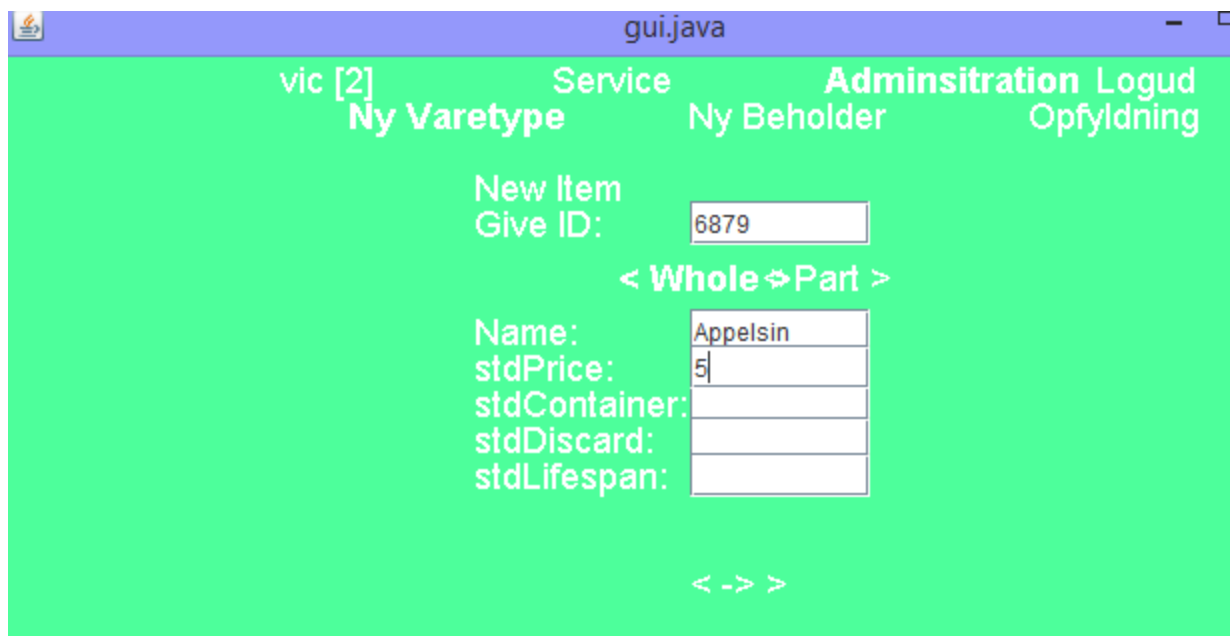
## Administration

Under administration eksistere der 3 forskellige valg for operatøren: Ny varetype, Ny beholder, Opfyldning.

### Ny varetype

Operatøren skal først indtaste et id, der skal være af typen int. Man kan med rette klandre menuen for ikke at kommunikere hvilken type man skal indsætte i de forskellige felter.

For dette projekt vil vi kun bruge Whole funktionen, da part of ikke er fuld ud implementeret. I dette eksempel opretter vi en appelsin som en ny varetype.



Som koden er nu, skal man blot udfylde et af felter for at gå videre til vægten. Dette er selvfølgelig ikke hensigtsmæssigt da det kan skabe tom/utilstrækkelig data i databasen. Optimalt ville det være hvis programmet insisterede på at x antal felter skulle udfyldes.

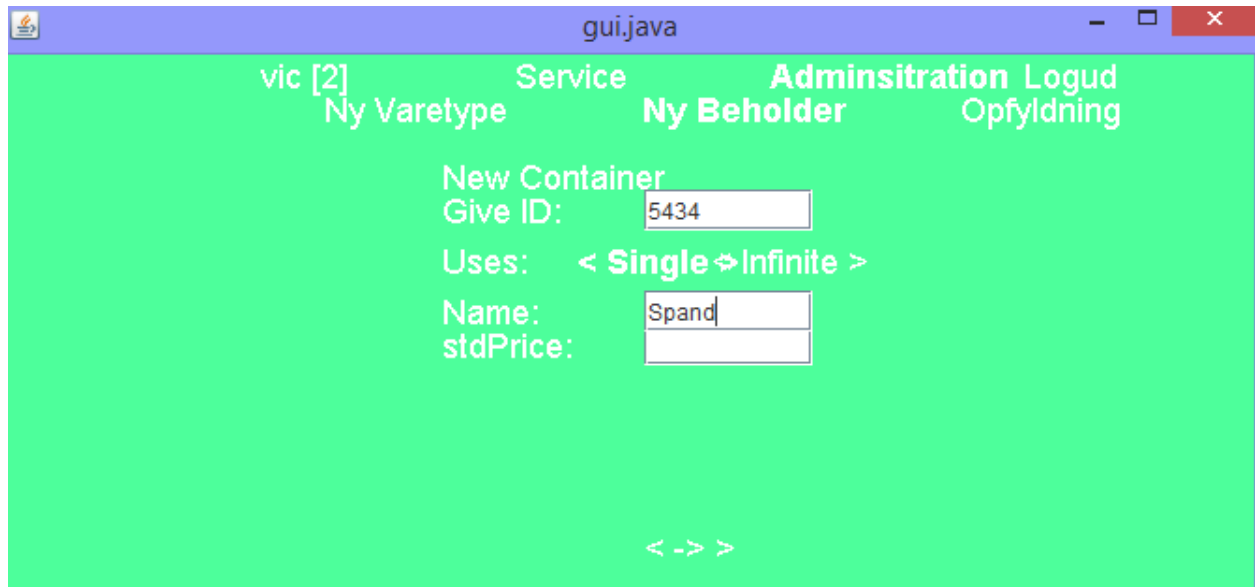
Alt den indtastede data, bliver gemt i databasen, dog bliver lifespan, std container, std discard og price ikke brugt til noget funktionelt i dette projekt.

Til sidst vejes den pågældende vare og det noteret og oprettet i databasen.

## Ny beholder

Ny beholder går i forlængelse af ideen om at gøre det mere erhvervsorienteret. Her kan man oprette beholdere som kasser, koppe og spande og bruge det som enhed til salg / afskrivning af vare. I dette eksempel indskrives vi et id, og derefter definere vi om container, kun kan bruges en gang (single use) fx en pap kop, eller infinite use fx. en porcelæns kop.

Herefter indtaster brugeren navn, og std. price for containeren.



## Opfyldning

I opfyldning er der to muligheder. Fælles for begge er at man starte med at indtaste id for den pågældende vare.

Derefter vælger operatøren om brugeren vil opfylde som "part of" eller "whole". Vælger operatøren "whole" skal man angive hvor mange enheder man øger lageret med.

Vælger operatøren i stedet "part of" har man flere muligheder. Bl.a. kan man give varen en betegnelse, lifespan, container etc. Den vigtigste funktion er at man vejer varen, og at dette opdatere hvor meget der er af varen i varelageret.

## Implementation og tests

### FTP

For at lave vores egen FTP-klient har vi været nødt til at lave omfattende research på omkring hvordan File Transer Protocol er opbygget. Vi har læst i RFC959 der omhandler FTP og her fundet ud af mere om hvordan protokollen fungerer.

Vi erfarede heri, at FTP gør brug af to forbindelser. Der findes en forbindelse kaldet kontrolforbindelsen, denne bruges til at sende kommandoer mellem

FTP-klient og -server. Dernæst findes der også en dataforbindelse, der bruges til at overføre filer mellem de to end hosts.

I passive mode står FTP klienten for initiering af begge forbindelser til serveren hvilket løser problemet med at klientens firewall filtrerer den indgående dataforbindelse fra serveren.

Når FTP forbindelser åbnes, åbner klienten to tilfældige lokale porte (port N (over 1023) samt port N+1). Den første port bruges til at kontakte serveren på dens port 21 og oprette en kontrolforbindelse. Klienten sender derpå en "PASV" kommando, der får serveren til at åbne en tilfældig port og serveren sender svar tilbage med portnummeret som svar på "PASV" kommandoen.

Klienten forbinder så fra sin åbnede port N+1 til den port, serveren har angivet som åben i sit svar. På denne måde er det klienten der initierer forbindelserne og derfor bliver indgående trafik ikke blokeret af firewallen.

FTP-klienten i dette projekt skulle kunne hente en fil fra en FTP-server. For at gøre dette skal man bruge kommandoen "RETR" men førend man kan bruge denne, er man dog nødt til at implementere en hel del andre kommandoer, dog afhængig af hvordan serveren er sat op. Vi har implementeret/brugt følgende kommandoer:

**USER** - bruges til at starte login-processen med det angivne brugernavn.  
**PASS** - sendes efter USER og bruges til at angive password for den bruger man vil logge ind som.  
**PWD** - bruges til at printe den nuværende arbejdsmappe på serveren.  
**CWD** - bruges til at skifte den nuværende arbejdsmappe på serveren.  
**PASV** - bruges til at fortælle serveren at den skal gå i passive mode.  
**LIST** - bruges til at printe en liste over filer og mapper i den nuværende arbejdsmappe på serveren.

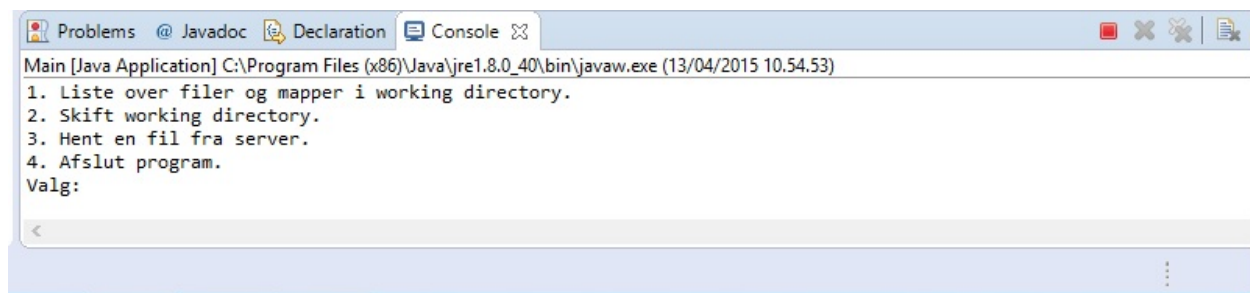
Til at finde disse kommandoer samt hvordan de bruges, har vi gjort brug af RFC959 men også en hjemmeside der har listet de forskellige kommandoer på et lidt mere behageligt layout. Der er links til både RFC og denne side med kommandoer sidst i rapporten i kildehenvisninger/litteraturliste. Ydermere har vi i forbindelse med svaret fra serveren når PASV sendes og IP og port skal dekodes, fundet inspiration fra en kilde til at gøre dette. Det drejer sig om at tokenize en string på den rigtige måde og vi henviser også til denne inspirationskilde i kildelisten til sidst.

Vi har udført mange blackbox tests undervejs i udviklingen af programmerne, for løbende at se, at programmerne virker ud fra et slut-bruger perspektiv.

Vi vil herunder vise screenshots for alle succesfulde forløb af muligheder for FTP-klienten. Vi har i løbet af vores udvikling af programmet gjort brug af en lokal FTP server. Vi har brugt FileZilla FTP server under udvikling og til tests men vi har også testet applikationen op mod FTP-serveren på Zyboboardet.

Når man først starter FTP-klienten skal man indtaste IP og port på den server, man vil forbinde til. Dernæst indtastes brugernavn og evt. password.

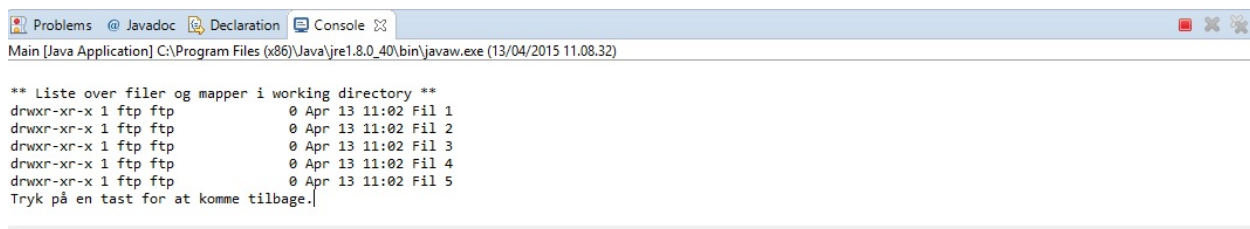
Dernæst kommer man til hovedmenuen:



```
Problems @ Javadoc Declaration Console
Main [Java Application] C:\Program Files (x86)\Java\jre1.8.0_40\bin\javaw.exe (13/04/2015 10.54.53)
1. Liste over filer og mapper i working directory.
2. Skift working directory.
3. Hent en fil fra server.
4. Afslut program.
Valg:
```

Her har man 4 valgmuligheder. For at lave testen har vi oprettet et work directory kaldet New Folder og oprettet 5 mapper inden i denne mappe.

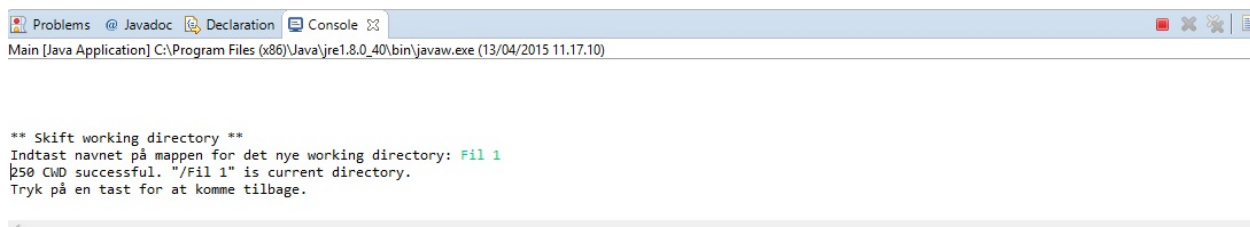
Trykker man 1, listes filer og mapper i working directory, hvilket giver følgende resultat:



```
Problems @ Javadoc Declaration Console
Main [Java Application] C:\Program Files (x86)\Java\jre1.8.0_40\bin\javaw.exe (13/04/2015 11.08.32)

** Liste over filer og mapper i working directory **
drwxr-xr-x 1 ftp ftp      0 Apr 13 11:02 Fil 1
drwxr-xr-x 1 ftp ftp      0 Apr 13 11:02 Fil 2
drwxr-xr-x 1 ftp ftp      0 Apr 13 11:02 Fil 3
drwxr-xr-x 1 ftp ftp      0 Apr 13 11:02 Fil 4
drwxr-xr-x 1 ftp ftp      0 Apr 13 11:02 Fil 5
Tryk på en tast for at komme tilbage.
```

Trykker man 2 kan man skifte arbejdsmappe (working directory). Man skal indtaste den nye mappe man ønsker som arbejdsmappe.

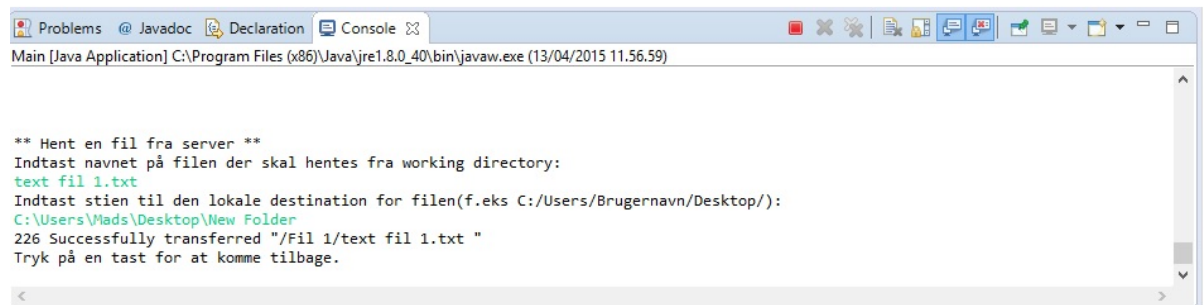


```
Problems @ Javadoc Declaration Console
Main [Java Application] C:\Program Files (x86)\Java\jre1.8.0_40\bin\javaw.exe (13/04/2015 11.17.10)

** Skift working directory **
Indtast navnet på mappen for det nye working directory: Fil 1
p50 CMD successful. "/Fil 1" is current directory.
Tryk på en tast for at komme tilbage.
```



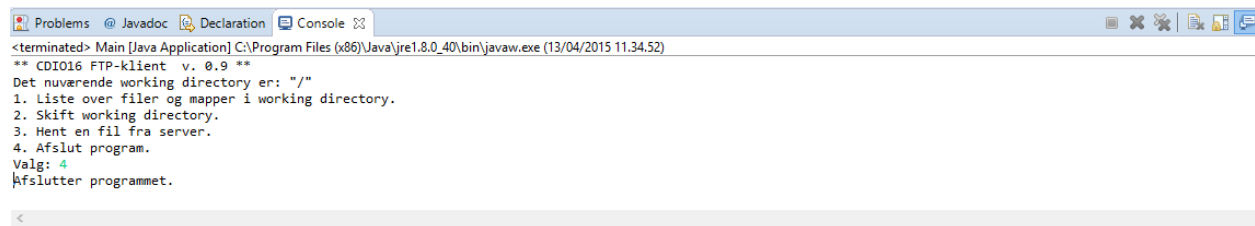
Trykker man 3 kan man hente en fil fra serveren. For at hente filen skal man indtaste navnet på filen, inklusiv filtype. Derefter skal man angive den lokale sti til filen, altså hvor man vil gemme filen lokalt på sin computer. Man kan kun hente filer enten fra arbejdsmappen eller undermappe af denne mappe. Det er ikke muligt at hente mapper, kun filer er en mulighed.



```
Problems @ Javadoc Declaration Console
Main [Java Application] C:\Program Files (x86)\Java\jre1.8.0_40\bin\javaw.exe (13/04/2015 11:56:59)

** Hent en fil fra server **
Indtast navnet på filen der skal hentes fra working directory:
text fil 1.txt
Indtast stien til den lokale destination for filen(f.eks C:/Users/Brugernavn/Desktop/):
C:\Users\Mads\Desktop\New Folder
226 Successfully transferred "/Fil 1/text fil 1.txt "
Tryk på en tast for at komme tilbage.
```

Trykker man 4 lukker man forbindelsen til serveren, og programmet afsluttes.



```
Problems @ Javadoc Declaration Console
<terminated> Main [Java Application] C:\Program Files (x86)\Java\jre1.8.0_40\bin\javaw.exe (13/04/2015 11:34:52)
** CDI016 FTP-klient v. 0.9 **
Det nuværende working directory er: "/"
1. Liste over filer og mapper i working directory.
2. Skift working directory.
3. Hent en fil fra server.
4. Afslut program.
Valg: 4
Afslutter programmet.
```

## Sensorer

I opgaveoplægget var det meget sparsommeligt beskrevet at programmet også skulle kunne styre nogle sensorer på Zyboboardet. Vi har ikke hørt om sådanne sensorer før eller siden og har derfor været rimelig uvidende omkring denne del af projektet. Det er mundet ud i, at vi blot har opdigtet en simpel temperatursensor som man kan styre via klientens terminal, ligesom FTP-kommandoerne. Man kan aflæse en sensor, ændre dens måleinterval og starte og stoppe sensorens måling.

## Kilder/litteraturliste

[http://en.wikipedia.org/wiki/File\\_Transfer\\_Protocol](http://en.wikipedia.org/wiki/File_Transfer_Protocol)  
<http://www.nsftools.com/tips/RawFTP.htm>  
<http://www.java2s.com/Code/Java/Network-Protocol/ImplementsaJavaFTPclientfromsocketandRFC.htm>

## Konklusion

I projektet er der udviklet to programmer. Ét der indeholder en simplificeret FTP-klient og ét der indeholder en WCU.

Arbejdet med WCU'en har været præget af gode intentioner ift. udvidelse af krav, samt mere kompleks funktionaliteten til programmet. Det erhvervsmæssige perspektiv har været et gennemgående tema i alt fra arbejdet med online databasen til flere muligheder for håndtering af vare. Projektet har til dels implementeret flere af disse koncepter i programmet, ved online database, containers, ny varetype og så videre.

Med et mere kritisk perspektiv kan man også konkludere, at projektet måske har slået et for stort brød op ift. hvad vi "kunne bage". Programmet opfylder kravspecifikationen fra opgavebeskrivelsen, men gennemgående problemer med GUI'en i Java og den tilhørende kommunikation med vægt simulatoren gør desværre at mange af de gode intentioner bliver overskygget af en besværlig user interface. Især kommunikationen med vægt simulatoren er svær at praktisere, heldigvis gør dette problem sig ikke gældende ved brug af den fysiske vægt.

Vi har i denne rapport prøvet at uddybe hvilke faldgrupper der ligger i vores user interface for operatøren, og hvordan det overordnet set fungerer bl.a. ved vedhæftning af en manual.

Vi har fået udviklet en applikation der kan forbinde til en FTP-server og modtage svar fra denne. FTP-klienten kan også downloade filer fra serveren, som specificeret i kravene i oplægget. Vi har fået implementeret alle de ønskede kommandoer og har dermed en funktionel FTP-klient. Sensordelen af Zybo-klienten har vi haft problemer med at forstå da begrebet sensor har været meget abstrakt, da vi ikke har lært noget som helst om dette. Dette har gjort det problematisk at udvikle men vi har forsøgt at lave en meget simpel temperaturføler i sidste øjeblik.