CDIO Del 3

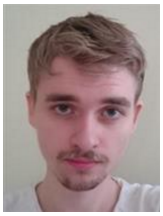
GWT
Gruppe 15

Lars Quaade
s144850

Lukas Bek
s153475

Magnus Haakonson
s153947

Sebastian Hoppe
s154306



William Wiberg
s153356

Danmarks Tekniske Universitet
Softwareteknologi Diplom
Videregående programmering 02324
Stig Høgh
07. Maj 2016

Time Regnskab

Navn	Dok.	Design	Impl.	Test	Andet	I alt
Lukas Bek	6	8	18	0	8	40
Sebastian Hoppe	6	4	20	7	4	41
Magnus Haakonsson	6	8	18	0	7	39
Lars Quaade	8	10	14	4	2	40
William Wiberg	8	10	10	1	8	37

Indhold

Indledning

Vi er en gruppe studerende, der i dette projekt har fået til opgave at lave en viderebygning på vores CDIO1 projekt.

Projektet ønskes udvidet med et web interface med back-end til operatøradministration. Yderligere ønskes der at datalaget - udover at blive implementeret i container klasser, også bliver implementeret som en database i MySQL.

Opgaven

Opgavebeskrivelsen kan findes på klassens fællesdrev i google-docs 02324 Tilgængeligt på CN:

`https://docs.google.com/document/d/1l0mSHTJmUcMbL-GQX_dN_ImHSpVwLR1Jxt8Acry_a0E/edit`

Kravspecifiering

Da dette projekt er en viderebygning på det tidligere projekt CDIO1, er mange af de tidligere krav stadig gældende for dette projekt med undtagelse af kravet for TUI, som i dette projekt erstattes med et Web interface.

Funktionelle

1. Systemet tilgås gennem en Web-applikation i GWT (Google Web Toolkit).
2. Web interface med et back-end som kan administreres af operatørene.
3. En superbruger som kan udføre CRUD operationer på operatørerne (Create, Read, Update og Delete, dog er der blevet nævt at Delete ikke skal føjes til i dette projekt).
4. Operatører skal indeholde følgende felter:
   (a) oprID: personens ID (Skal være entydigt)
   (b) oprNavn: operatørens navn
   (c) ini: Initialer
   (d) CPR nummer
   (e) password

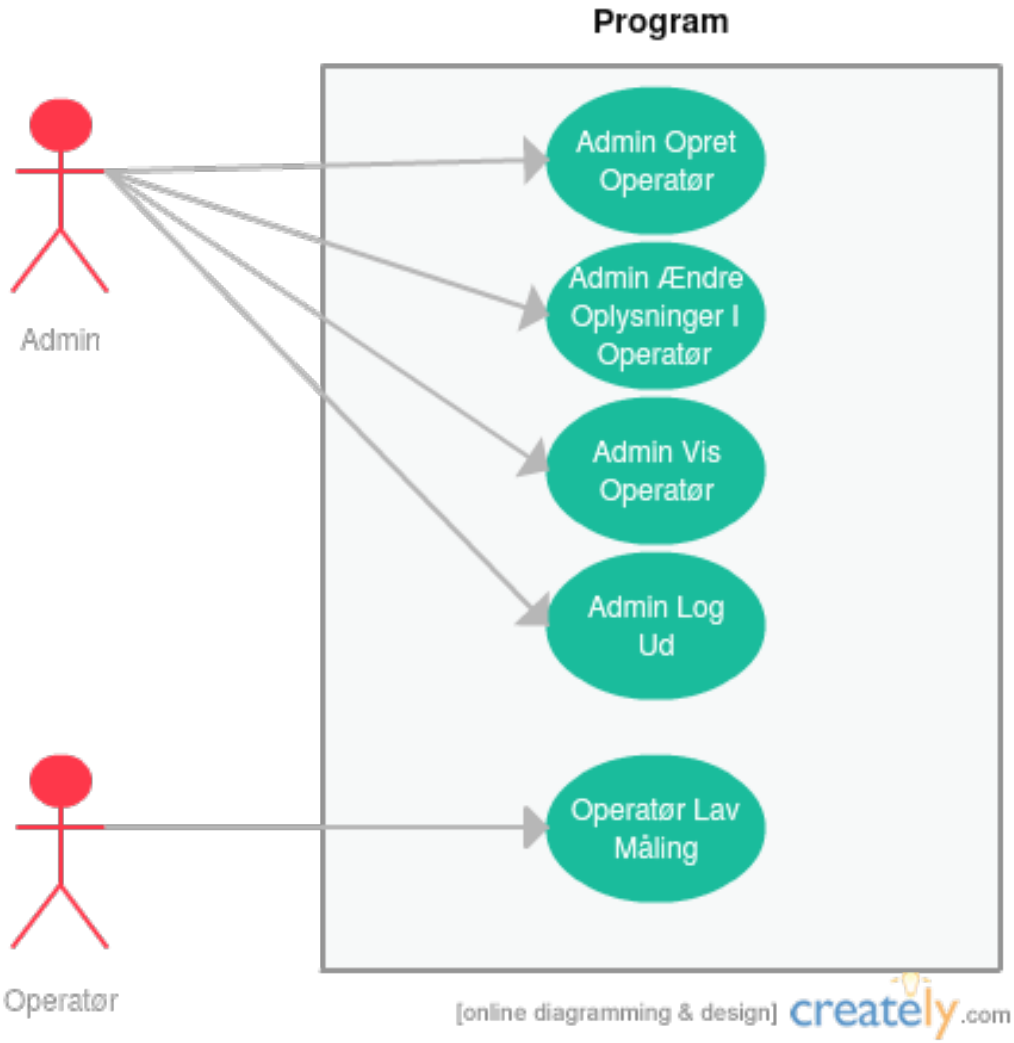
Non Funktionelle

1. Opbygges efter 3-lags modellen.
2. Datalag som både transient lager, ved brug af container klasser, og som en database i MySQL.
3. Brug af UML til illustration af design.
4. Skal kodes i Java, HTML og CSS og simuleres via Eclipse og en Web Browser.

Design

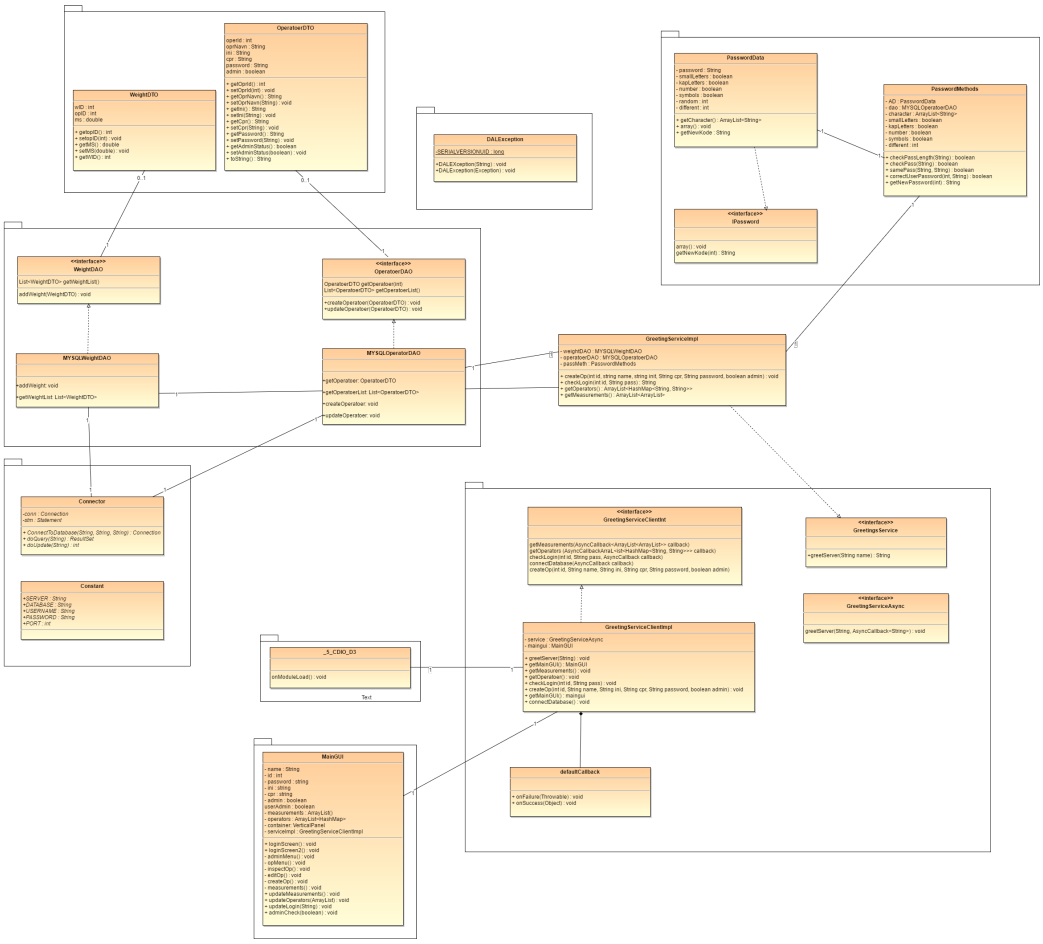
Designet af systemet tager udgangspunkt i de nye, samt de gamle krav fra CDIO1 og er her beskrevet med UML diagrammer og Use Cases.
Ønskes der at se og prøve programmet selv, kan der logges ind som admin med:
Username: 10
Password: Abc02324
Normal operatør:
Username: 2
Password: password

Use case diagram



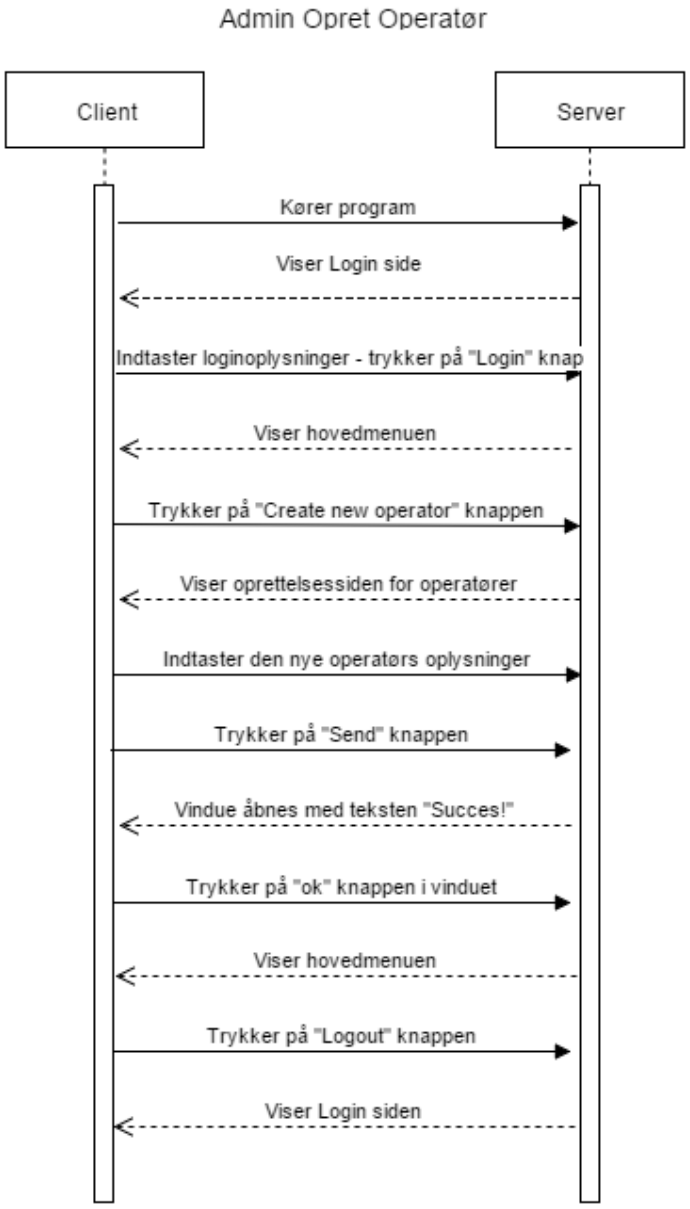
Ovenover ses et Use Case diagram. Use Case diagrammet illustrerer Admin og Operatør som aktører og de forskellige Use Cases som de udfører, indkapslet af programmet.

Klassediagram



Her ses et design-klasse-diagram over de forskellige klasse i programmet. Det kan være svært at se noget ud fra dette billede, men det er høj opløsning, så hvis der ønskes et nærmere kig, så kan der zoomes ind.

Systemsekvensdiagram



Systemsekvensdiagrammet ovenfor viser sammenspillet mellem klienten og serveren, når en af de primære Use Cases, "Admin Opret Operatør", udføres i web interfacet.

Google Web Tools (GWT)

I dette projekt er der anvendt GWT (Google Web Toolkit) til at bygge hjemmesiden og dens funktioner. GWT er smart til brug i dette projekt, da der kan anvendes almindelig java kode til at holde styr på systemet og der bruges java kode til at køre SQL scripts til databasen. GWT kan konvertere al koden til den JavaScript version der passer til den pågældende browser. GWT er også smart, da det indeholder en masse forskellige funktioner, som ville være svært eller krævende at lave ved brug af enten Java eller JavaScript. Websider kan laves rimelig nemt med GWT grundet dens API, som indeholder bl.a. Java biblioteker til nemt at kunne lave komponenter som fx. tekstbokse, indtastningsfelter og knapper.

System

Systemet bruger RPC og AsyncCallback til kommunikation mellem klient og server.

RPC

Et Remote Prosedure Call (RPC), bruges på klient siden til at få en metode på server siden til at starte. Alt efter om der anvendes Sync- eller Async-Callback, vil klienten hhv. vente eller ikke vente på det svar der kommer fra serveren. I dette projekt anvendes kun AsyncCallback.

AsyncCallback

Et AsyncCallback er en måde at kommunikere med en server fra klientens side. Til samme formål findes også SyncCallback. Den bedste måde at forklare disse - er ved en analogi. Hvis en person (i dette tilfælde klienten) ringer til en anden person (i dette tilfælde serveren), og stiller et spørgsmål til denne person, eksempelvis hvad er 2 + 2? Så virker Sync- og Async-Callback på to forskellige måder. SyncCallback vil stille spørgsmålet - "hvad er 2 + 2?"Og så vente på at personen i den anden ende svare: "Det er 4."Hvorefter klienten kan gøre med denne information hvad den nu skulle.

AsyncCallback virker fundamentalt på samme måde. Der er igen to instanser, hvor den ene fungerer som klient og den anden som server. Klienten kommer igen med det samme spørgsmål til serveren: "Hvad er 2 + 2?"Men i dette tilfælde siger den så også: "Kontakt mig når du har svaret."Nu venter klienten

ikke på at få svar fra serveren, men regner bare med at der kommer et svar på et tidspunkt i fremtiden. Dette gør at klienten kan gøre andre ting imens at serveren laver sit arbejde. Det er smart hvis klienten beder om en større bunke data fra en tabel, der skal arrangeres på en speciel måde. Klienten kan så anmode om dette, og mens at serveren sætter det sammen til klienten, kan klienten eksempelvis begynde at sætte GUI op til den data der kommer senere.

Database

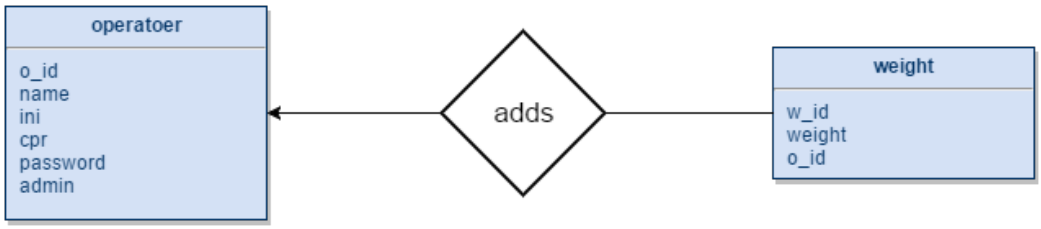
Til dette projekt er der opsat en database, til at opbevare den data, som skal gemmes. Databasen er forholdsvis lille, da der som sådan ikke er særlig meget data, men der er dog stadig nok til at der skal laves en. Databasen indeholder to tabeller: En med operatører kaldet "operatoer"og en med vægtmålinger kaldet "weight".
Operatør tabellen indeholder den data der haves om operatørerne:

- oprID: Operatørens ID
- oprNavn: Operatørens navn
- ini: Initialer
- CPR nummer
- password

Weight tabellen indeholder den data der haves om vejningen:

- wID: Et unikt ID som alle vægtmålinger fås
- opID: Operatørens ID
- ms: Objektets vægt

De kan her ses udtrykket i et ER-diagram (Entity-relationship):



Databasen kan findes på adressen:
`ec2-52-30-89-247.eu-west-1.compute.amazonaws.com:3306`
Login er grp15 både i login og password. Databasen hedder også grp15.

Test

Dette projekt er en viderebygning af CDIO-D1, så der er derfor ikke lavet test af hvad der allerede virkede i dette projekt. Her er det primært datalaget, som er opsat efter 3-lagsmodellen, der ikke er blevet testet. Den eneste tilføjelse til datalaget er, at selve dataet er blevet lagt i en database. De MySQL-statements, som er blevet tilføjet til projektet, er blevet testet med nogle manulle testmetoder, der kan findes i projektet under mappen "test". De MySQL test påviste at alle de ønskede funktioner virkede.

Hvad der mangler at blive testet er hele GWT modulet, men da der ikke var tid, blev det prioriteret fra. Der er dog stadigvæk blevet lavet de basis test, som påviser at de forskellige funktioner virker.

Viderebygning

Der er visse steder i dette projekt, hvor der kunne ønskes forbedringer eller viderebygning. Grundet tidpres til slut i projektet, er der visse steder, hvor en simpel løsning er valgt for at prøve at få alle ønskede krav stillet.

Operatøren

Som det er lige nu, er den eneste funktionalitet som operatøren har, at kunne indtaste nogle målinger, som bliver lagret i databasen. Det er som sådan alt hvad der egentlig blev ønsket at han kunne, men der kunne dog menes at han burde have adgang til sine egne oplysninger. Herunder ønskes der at der blev implementeret en funktion, som gav operatøren mulighed for at ændrer de oplysninger personen har.

Layoutet

Det tog lang tid før at en ordentlig forståelse af GWT var opnået, så derfor halter layoutet og den designmæssige del af siden. Der er stort set ikke brugt CSS kodning for at opsætte siden, da der blev prioriteret at få alle de ønskede funktionaliteter med.

Test

Der er ikke lavet direkte test af programmet, kun nogle få test af MySQL statements. Der kunne godt have været ønsket nogle flere test, herunder blackbox/whitebox testing med negative inputs, men der var ikke tid til dette i projektets ende. Dog er selve datalaget, som der burde testes på, blevet testet i CDIO-D1, så der kan her argumenteres for at det virker. Det eneste der er blevet tilføjet er at dataet ligger i en database og de MySQL statements der hører hertil er som nævnt blevet testet.

Konfigurations- og versionsstyring

Udviklings- og produktionsplatformen brugt til at udvikle dette projekt er ens og kan beskrives som en samlet platform. Den inkluderer følgende:

- OS: Windows 10
- Java Version 8 update 91
- Eclipse Version: Mars 2 Release (4.5.2) ...

Produktets udvikling er dokumenteret og versioneret, ved hjælp af GitHub, og kan ses på nedenstående link:
`https://github.com/Hoppe2808/15_CDIO_D3`

Konklusion

Ud fra rapporten kan det konkluderes at kravene til opgaven er blevet opfyldt og der er lavet et web interface med funktionerne fra CDIO 1, med tilhørende design i form af beskrivelser og UML (med få mangler ved funktioner til operatøren). Web interfacet giver en superbruger mulighed for at oprette, læse, opdatere og slette andre operatører.

Yderligere er der, som ønsket i datalaget, blevet implementeret en database i MySQL, ud over container klassen, til at opbevare dataet om både operatørerne og vægten, som begge indeholder de felter givet i kravene.

Opgaven har været meget omfattende, da vi både har brugt vores viden om GWT til opbygningen af vores Web applikation, samt vores viden fra et af vores andre fag, 'Databaser', til oprettelse af en database, samt tilsvarende connectors.

Bilag

Admin Opret Operatør

ID: 01
Kort Beskrivelse: Admin opretter en operatør
Primær Aktør: Bruger (admin)
Sekundære Aktører:
Forudsættelser: Admin har nødvendig info om operatør
Primært flow: 1. Admin logger ind 2. Admin anvender "Opret operatør" knappen 3. Admin indtaster nødvendig information 4. Admin committer 5. Admin afslutter oprettelsen
Efterfølge: En ny operatør med operatør privilegier er nu oprettet
Alternative Flows: 1. Admin logger ind 2. Admin anvender "Opret operatør" knappen 3. Admin indtaster nødvendig information 4. Admin committer 5. En fejl opstår op Admin bliver sendt tilbage til "Opret operatør" menu

Admin Ændre Oplysninger i Operatør

ID: 02
Kort Beskrivelse: Admin opdatere en operatør
Primær Aktør: Bruger (admin)
Sekundære Aktører:
Forudsættelser: Admin har ny info om operatør
Primært flow: 1. Admin logger ind 2. Admin anvender "Opdater operatør" knappen 3. Admin vælger hvilken operatør der skal opdateres 4. Admin indtaster ny information 5. Admin committer 6. Admin afslutter oprettelsen
Efterfølge: Operatøren er nu opdateret
Alternative Flows: 1. Admin logger ind 2. Admin anvender "Opdater operatør" knappen 3. Admin vælger hvilken operatør der skal opdateres 4. Admin indtaster ny information 5. Admin committer 6. En fejl opstår op Admin bliver sendt tilbage til "Opret operatør" menu

Admin Vis Operatør

ID: 03
Kort Beskrivelse: Admin opdaterer en operatør
Primær Aktør: Bruger (admin)
Sekundære Aktører:
Forudsættelser: Brugeren har Admin rettigheder
Primært flow: 1. Admin logger ind 2. Admin anvender "Vis operatør" knappen 3. Admin vælger hvilken operatør der skal vises 4. Operatøren bliver vist 5. Admin afslutter
Efterfølge: Admin er tilbage til start
Alternative Flows: 1. Admin logger ind 2. Admin anvender "Vis operatør" knappen 3. Admin vælger hvilken operatør der skal vises 4. Operatøren findes ikke 5. Programmet vender tilbage til "Vis operatør" menu

Admin Log Ud

ID: 04
Kort Beskrivelse: Admin Logger Ud
Primær Aktør: Bruger (admin)
Sekundære Aktører:
Forudsættelser: Admin er allerede logget ind og befinder sig i menuen
Primært flow: 1. Admin anvender "Log ud" knappen
Efterfølge: Admin er tilbage på log ind siden

Operatør Lav Måling

ID: 05
Kort Beskrivelse: Operatør foretager en måling og sender dataet til serveren
Primær Aktør: Bruger (Operatør)
Sekundære Aktører:
Forudsættelser: Brugeren har operatør rettigheder
Primært flow: 1. Operatør logger ind 2. Operatør indtaster sin måling i feltet "Indtast din måling" 3. Operatør trykker på "Send" knappen 4. Operatør får svar om at målingen er blevet tilføjet 5. Operatør trykker på "Logout" knappen
Efterfølge: Målingen er nu tilføjet og programmet er tilbage på login menuen
Alternative Flows: 1. Operatør logger ind og bliver rykket til målingsmenuen 2. Operatør indtaster sin måling i feltet "Indtast din måling" 3. Operatør trykker på "Send" knappen 4. Operatør får en besked om at der er opstået en fejl 5. Siden vender tilbage til målingsmenuen