

CDIO del 3, April/Maj 2015

02324 Videregående programmering

Projektnavn: 16_CDIO3

Emne: Operatørkartotek med GUI

Afleveringsfrist: Mandag d. 11/05-2015, kl. 05:00

CDIO gruppenummer: 16

Gruppemedlemmer (studienummer, efternavn, fornavn):

s073629, Jabbari, Muhammad



s123157, Adamsen, Frederik



s144846, Eriksen, Robert



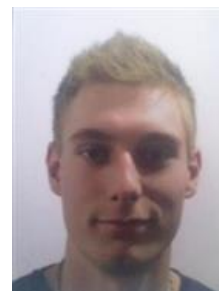
s144845, Ørnby, Victor



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 repository URL: https://github.com/nicklas669/16_CDIO3/commits/master

Indholdsfortegnelse

[Timeregnskab](#)

[Indledning](#)

[Analyse](#)

[Kravspecifikationer](#)

[Funktionelle krav](#)

[Ikke funktionelle krav](#)

[Use-case diagram](#)

[Design](#)

[Klassediagram](#)

[Implementation](#)

[Database](#)

[Password validering](#)

[Layout](#)

[Login](#)

[Vis operatører](#)

[Tilføj operatør](#)

[Ret operatør](#)

[Slet operatør](#)

[Konklusion](#)

Timeregnskab

16_02324_CDIO 3							
Time regnskab							
Dato	Gruppefælle	Analys e og design	Implementation	Tes t	Dokumentatio n	Ande t	I alt
22/04/2015	Nicklas	3	1		2	1	7
27/04/2015	Nicklas	1	3		2	1	7
01/05/2015	Nicklas		2	1			3
03/05/2015	Nicklas		4		2		6
04/05/2015	Nicklas		2	1	1,5		4,5
05/05/2015	Nicklas		5	1	1	1	8
06/05/2015	Nicklas		0,5				0,5
07/05/2015	Nicklas		4		0,5		4,5
08/05/2015	Nicklas		2		4		6
10/05/2015	Nicklas				2		2
						Total timer:	31, 5

Indledning

Projektet er tredje iteration af et gennemgående CDIO-tema på andet semester. Vi har tidligere i dette kursus udviklet et operatørsystem hvor man som systemadministrator kunne vise, oprette, slette og rette operatører i systemet, i en konsol applikation. Dette system skal nu opgraderes i dette projekt. Opgraderingen består i, at dette operatørsystem nu skal have en grafisk brugerflade lavet i GWT. Derudover, skal der også oprettes en database og heri skal operatørerne gemmes i. Der er altså tale om en opgave hvor der skal udvikles en hjemmeside der kan kommunikere med en database.

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.

Det nævnes kort her at der i applikationen kan logges ind som superbruger med operatørid: 31 og password: passNH31.

Analyse

Projektarbejdet startede, som altid i de iterative udviklingsmetoder vi har lært, med en analyse af hvad systemet skulle kunne, i henhold til Unified Proces.

Først arbejdede vi på at fastslå projektets rolle i forhold til det overordnede CDIO projekt på tværs af fagene. Denne manøvre gav det fornødne overblik til at bestemme hvilken rolle netop dette projekt skulle understøtte.

Derefter analyserede vi det udleverede oplæg samt det udleverede eksempel på et personkartotek i GWT. På baggrund af dette materiale lavede vi følgende analytiske elementer.

Kravspecifikationer

Kravspecifikationen er udarbejdet ud fra opgave oplægget.

Funktionelle krav

1. Alle felterne givet i oplægget for OperatoerDTO objektet er obligatoriske.
 - 1.1. En operatør skal have et "id" felt.
 - 1.1.1. En operatørs id skal være entydigt, altså en primær nøgle.
 - 1.1.2. En operatørs id skal være i området 1-99999999.
 - 1.2. En operatør skal have et "navn" felt.
 - 1.2.1. En operatørs navn skal være minimum 2 karakterer og maksimum 20 karakterer.
 - 1.3. En operatør skal have et "initialer" felt.
 - 1.3.1. En operatørs initialer skal være minimum 2 karakterer og maksimum 3 karakterer.
 - 1.4. En operatør skal have et "cpr" felt.
 - 1.4.1. En operatørs cpr skal være 10 karakterer.
 - 1.4.2.
 - 1.5. En operatør skal have et "password" felt.
 - 1.5.1. En operatørs password skal være minimum 7 karakterer og maksimum 8 karakterer.

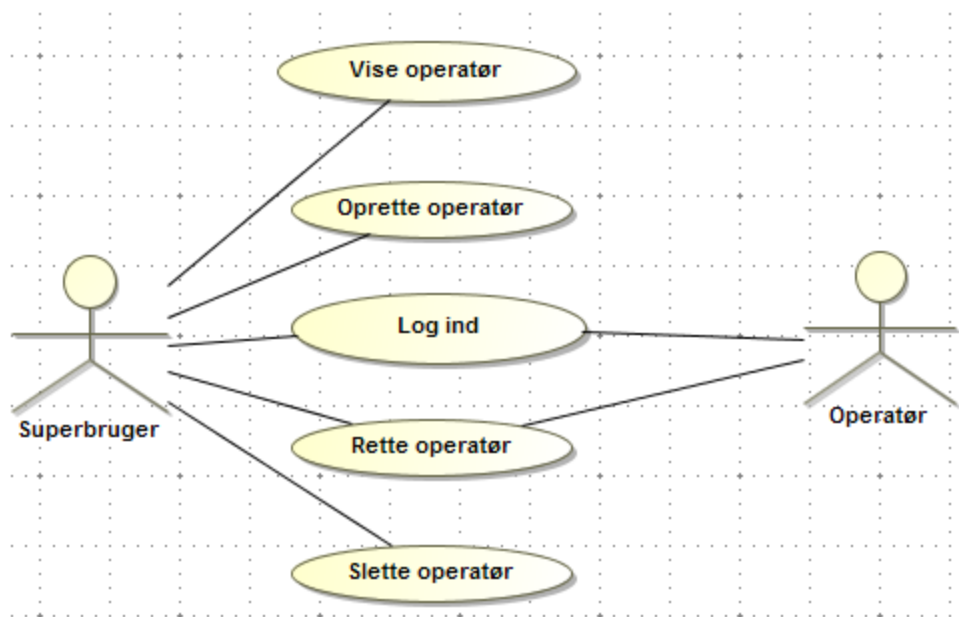
2. Der skal udvikles en webapplikation i GWT, der giver brugeren mulighed for at oprette, slette, rette og vise operatører.
3. Datalaget skal afkobles med et interface IOperatoerDAO, der indeholder definition af de nødvendige metoder.
4. Datalaget skal implementeres som en SQL database.
5. Det skal være muligt at logge ind som superbruger og herefter kunne udføre principperne i CRUD.
 - 5.1. En superbruger skal kunne vise operatører.
 - 5.2. En superbruger skal kunne tilføje operatører.
 - 5.3. En superbruger skal kunne rette operatører.
 - 5.4. En superbruger skal kunne "slette" operatører.
 - 5.4.1. Et krav til applikationen er, at en operatør aldrig kan slettes helt. Denne kan sættes til at være inaktiv men aldrig slettes af hensyn til mulighed for tilbagesporing.

Ikke funktionelle krav

6. Koden skal designes efter principperne i 3-lags modellen.
7. Programmet skal være udviklet i Java.
8. Programmet skal være udviklet i GWT.

Use-case diagram

Som en del af analyse delen af dette projekt er der blevet udviklet et use-case diagram. Nedenfor ses use-case diagrammet for dette projekt:



Det ses at en superbruger kan væsentlig flere ting end en “almindelig” operatør. De kan, og skal, begge logge ind før de kan gøre noget. Herefter er der dog funktioner tilgængelig alt efter hvilken type bruger der logges ind som.

En operatør kan kun logge ind og derefter rette, og dermed se, sine egne oplysninger. Denne kan ikke vise alle operatører, rette, oprette eller slette operatører.

En superbruger kan, efter at logge sig ind, gøre alt det ovenstående. Denne kan vise alle operatører i hele systemet, også de, der er inaktive. En superbruger kan også oprette en operatør, rette en hvilken som helst operatør eller slette enhver operatør.

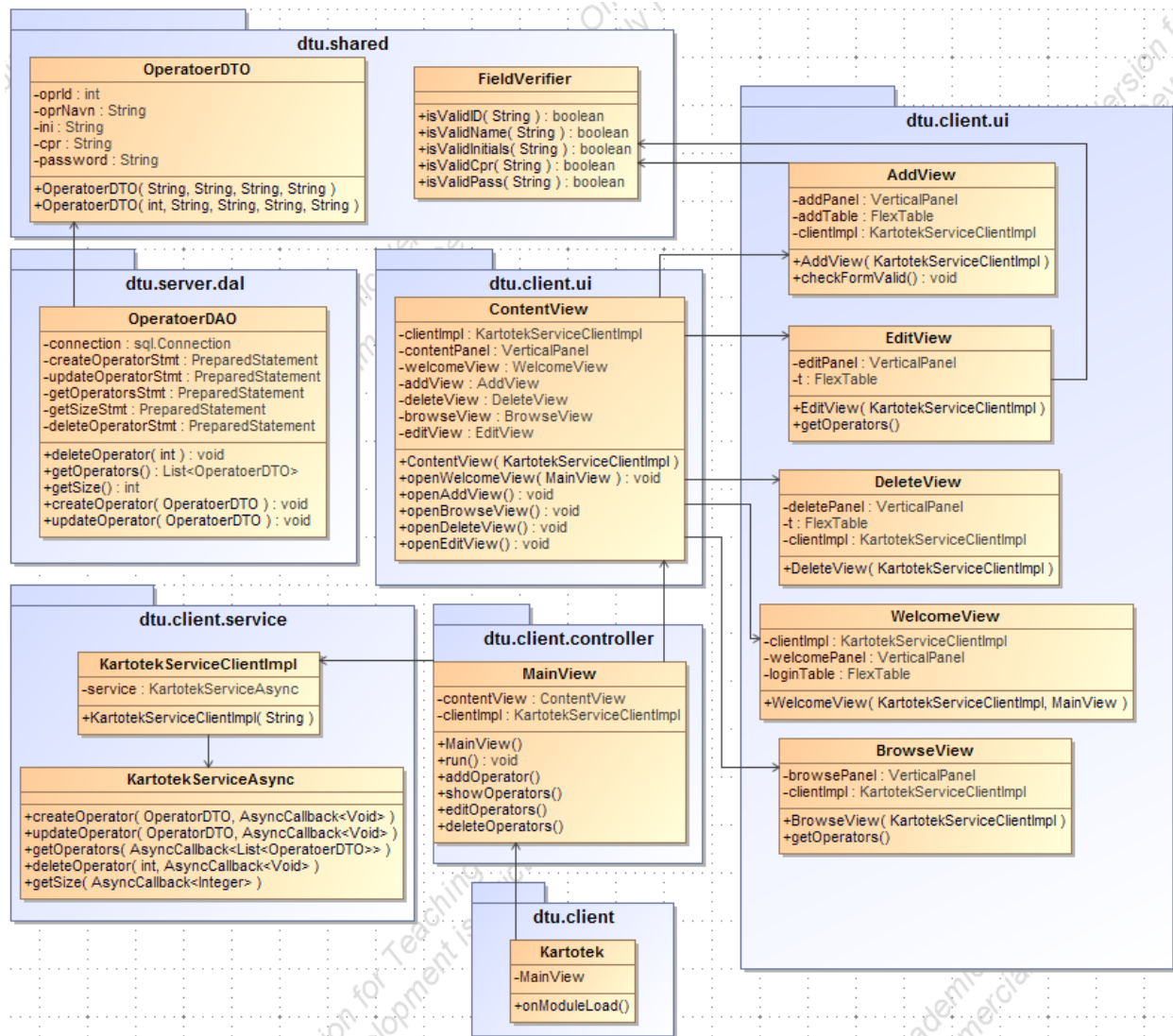
Design

I forbindelse med at projektopgaven blev stillet, blev vi introduceret til et eksempel på hvordan et personkartotek kunne laves i GWT. Vi fik udleveret kildekoden til dette eksempel og vi har baseret vores implementation af operatørkartoteket på dette eksempels kildekode. Vi har altså fundet inspiration i Personkartotek V2¹ af Mads Nyborg og brugt dele af dette til at forme basen af vores kartotek.

Klassediagram

Vi har i forbindelse med projektet udviklet et design klassediagram. Klasserne i diagrammet er inddelt i pakker, ligesom de er i Java projektet. Dette er gjort for at skabe et bedre overblik over hvordan disse klasser afhænger af hinanden samt hvor de hører til. Forinden ses et design klassediagram for projektet:

¹ <https://drive.google.com/open?id=0B-pPbZ8YwfkFaHowSTFBMUx4MU0&authuser=0>, Personkartotek V2



For at knytte et par ord til ovenstående figur, så viser det hvilke klasser der findes i projektet samt hvordan de gør brug af hinanden. I GWT skal angives en Entry point klasse der i dette projekt er Kartotek klassen. Når denne bliver loaded, opretter den en instans af MainView, der er controller for hele GUI'en.

MainView instantierer et KartotekService objekt som sendes med videre rundt til alle views der oprettes, således at de kan gøre brug af kommandoer til og fra server delen af applikationen.

Implementation

Vi vil i dette afsnit komme ind på hvilke funktionaliteter vi har implementeret, hvordan de virker, hvordan vi har gjort dette samt eventuelt hvad vi ikke har fået implementeret.

Database

Vi startede med at implementere datalaget som et transient datalag, hvor vi blot brugte en ArrayList til at holde vores data i. Det havde den ulempe at datalaget blev "nulstillet" hver gang man genindlæste en side på hjemmesiden. I opgaven blev det stillet, at vi siden hen skulle udvikle applikationen så den kunne arbejde sammen med en SQL database. Til dette behøvede vi en database at kommunikere med.

Vi designede derfor en database, kaldet kartotek, med en enkelt tabel i, operatoer. Denne database er noget simplificeret i forhold til den database vi skal gøre brug af i forbindelse med 3-ugers projektet men til denne opgave kan den sagtens klare jobbet.

Forneden ses et billede over tabellen operatoer's attributter samt deres datatyper:

Column	Type	Default Value	Nullable
◇ id	int(11)		NO
◇ navn	varchar(20)		NO
◇ ini	varchar(3)		NO
◇ cpr	varchar(10)		NO
◇ pass	varchar(8)		NO
◇ active	tinyint(1)		NO
◇ level	tinyint(2)		NO

Attributter og datatyper er designet med inspiration fra opgaveoplægget, der definerede nogle obligatoriske fields som en operatør skulle have. Det bemærkes at der dog er to attributter der ikke er defineret i opgaveoplægget, nemlig "active" og "level". Dette er kolonner vi selv har valgt at indføre.

Active attributten bruges til at angive om en given operatør er aktiv eller ej. I praksis betyder dette, at en inaktiv operatør er "slettet". Vi kan dog ikke tillade os at droppe en operatør fuldstændig netop på grund af den omtalte database der skal bruges senere hen. Den database har nemlig operatørers id som fremmednøgle og man kan derfor ikke blot fjerne rows som man vil. Derfor bruger vi active attributten til at holde styr på om en operatør er aktiv eller ej.

Level attribut bruges til at holde styr på en operatørs "niveau" i systemet. I dette system forefindes kun to typer af brugere, nemlig superbruger og operatør.

Scriptet til at oprette den udviklede database er vedlagt i det afleverede Java projekt som "operatoerkartotek_dump_08-05-15.sql".

Password validering

Det blev oplyst i opgaveoplægget at en operatørs password skulle være 7-8 karakterer langt. Da dette var implementeret, valgte vi at forsøge at genbruge vores password validering fra CDIO1 i 02324. I det projekt skulle vi nemlig lave password validering i forhold til DTUs regler for passwords. Vi lånte således vores kode fra CDIO1² og brugte dele af den til vores password validering i dette projekt.

Passwords håndteres igennem hele applikationen som plaintext passwords, hvilket selvfølgelig ikke er særlig hensigtsmæssigt i forhold til sikkerhed. En ondsindet person kan forholdsvis nemt opsnappe kodeordet hvis denne er på samme netværk. Det er et spørgsmål om tid at lave hashing af passwords og det var heller ikke et krav til opgaven og vi har derfor valgt ikke at gå mere i dybden med dette.

Layout

Da vi havde fået applikationen til at virke som den skulle og opfyldt alle krav, mente vi, at vi ville gøre lidt ekstra ud af designet og ikke blot bruge det layout der blev lagt op til i det udleverede kartotek eksempel.

Vi begyndte derfor at designe vores eget bud på hvordan hjemmesiden kunne se ud men fandt dog ud af at det ville være nemmere blot at finde en hjemmeside template på internettet og så få flettet vores GWT sammen med denne.

Vi fandt således en skabelon på internettet³ som vi synes var en forbedring af det gamle og samtidig forholdsvis simpelt at konvertere til vores behov. Det gav dog en smule problemer i forhold til at få HTML, CSS og GWT til at spille helt sammen men i sidste ende lykkedes det og resultatet kan ses på de kommende billeder af applikationen i brug. Vi har ladet links til skaberen af skabelonen stå i footeren, da dette var hans eneste ønske hvis man gjorde brug af den.

² https://github.com/nicklas669/16_del1, password validering, CDIO1 af CDIO gruppe 16

³ <http://www.bryantsmith.com/template/>

Login

I opgaveoplægget stod at der skulle laves samme testprogram som i CDIO1, hvilket indebærer at man skal kunne logge ind som superbruger og teste programmets funktioner.

Når man kommer ind på forsiden af operatøradministrationshjemmesiden, mødes man af følgende skærm:



The screenshot shows a web application interface for 'Operatørkartotek CDIO gruppe 16'. The title is in large white text on a dark background. Below the title, a message reads: 'Velkommen til operatør kartotek af CDIO gruppe 16. Du skal logge ind for at komme videre!'. There are two input fields: 'Operatør id:' and 'Password:'. A 'Log ind' button is positioned below the password field. At the bottom, small text reads: 'web development by bryant smith, edited by CDIO group 16'.

Det bemærkes at navigationsmenuen ikke er tilgængelig når man ikke har logget ind. Når der logges ind, får man en besked om dette og navigationsmenuen bliver tilgængelig for brugeren. Der kan f.eks. logges ind som superbruger med operatørid: 31 og password: passNH31. Efter log ind med denne bruger ses følgende skærm:



The screenshot shows the dashboard after a successful login. The title remains the same. Below the title, there is a navigation bar with four links: 'Vis operatører', 'Tilføj operatør', 'Ret operatør', and 'Slet operatør'. A message in the center reads: 'Du er nu logget ind og kan bruge navigationsmenuen foroven.'. The same footer text 'web development by bryant smith, edited by CDIO group 16' is present at the bottom.

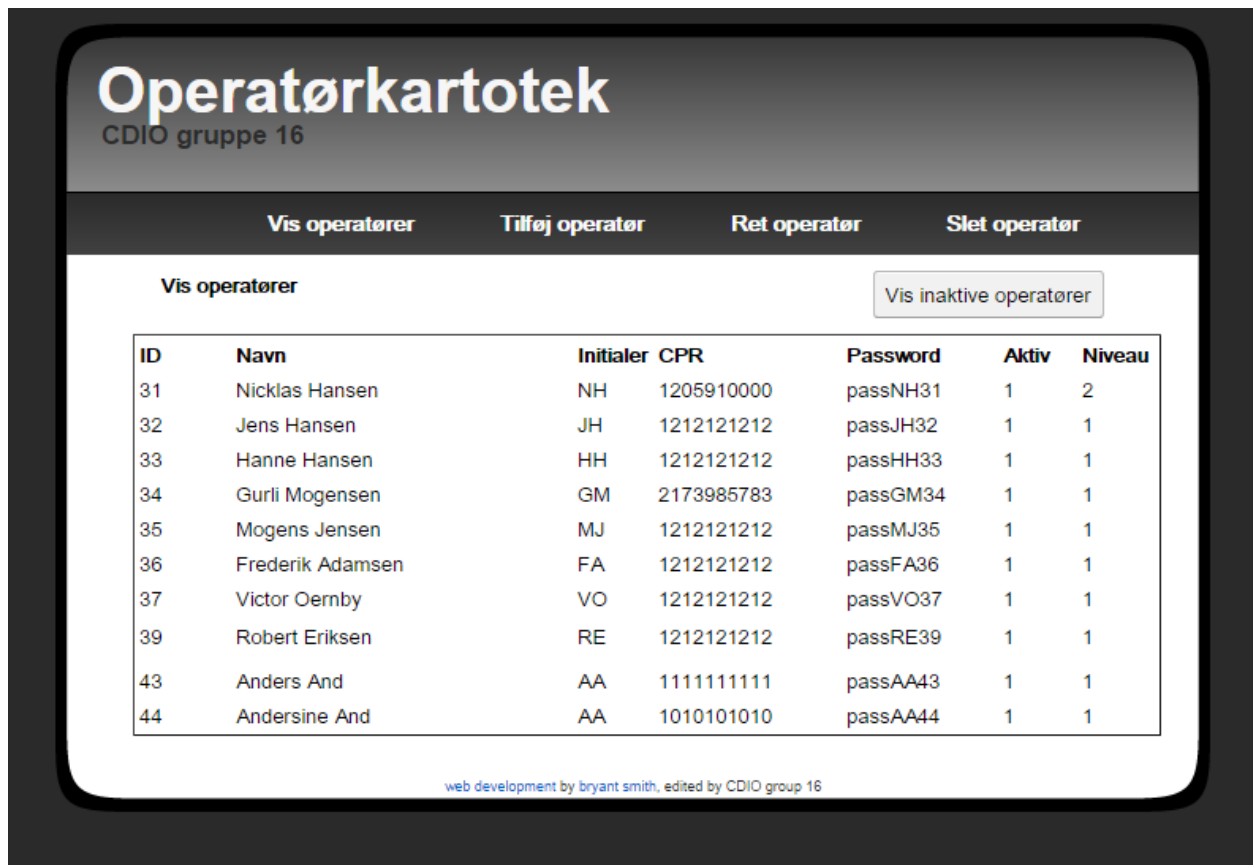
For at lave login funktionaliteten, har vi været nødt til at oprette en ny metode i interfacet KartotekService, kaldet login, der tager et id og et password som argument. Da vi gjorde

dette, var vi også nødsaget til at oprette funktionen login i KartotekServiceAsync. Dette har vi også gjort og den returnerer en AsyncCallback<Boolean>, som kan antage true eller false alt efter om et login forsøg var successfuldt eller ej.

Det er ikke blevet implementeret så et brugerniveau har en reel forskel på ens views. Det vil sige, at en operatør og en superbruger faktisk kan udføre de samme funktioner. Dette er selvfølgelig beklageligt men at lave et fuldt funktionelt login system var noget der ikke var tid til i gruppen. Man kan logge ind med de rigtige id og passwords men derfra er der ikke forskel på de to brugertyper.

Vis operatører

Forneden ses et billede af vores "Vis operatører" side:



The screenshot shows a web application titled "Operatørkartotek" with the subtitle "CDIO gruppe 16". The interface has a dark header with four navigation buttons: "Vis operatører", "Tilføj operatør", "Ret operatør", and "Slet operatør". Below the header, there is a section titled "Vis operatører" with a button "Vis inaktive operatører" to its right. A table displays a list of operators with the following columns: ID, Navn, Initialer, CPR, Password, Aktiv, and Niveau. The table contains 10 rows of data. At the bottom of the application, there is a small footer text: "web development by bryant smith, edited by CDIO group 16".

ID	Navn	Initialer	CPR	Password	Aktiv	Niveau
31	Nicklas Hansen	NH	1205910000	passNH31	1	2
32	Jens Hansen	JH	1212121212	passJH32	1	1
33	Hanne Hansen	HH	1212121212	passHH33	1	1
34	Gurli Mogensen	GM	2173985783	passGM34	1	1
35	Mogens Jensen	MJ	1212121212	passMJ35	1	1
36	Frederik Adamsen	FA	1212121212	passFA36	1	1
37	Victor Oernby	VO	1212121212	passVO37	1	1
39	Robert Eriksen	RE	1212121212	passRE39	1	1
43	Anders And	AA	1111111111	passAA43	1	1
44	Andersine And	AA	1010101010	passAA44	1	1

Det ses, at ikke alle operatører vises til at starte med. Det kan via knappen oppe til højre vælges om man også ønsker at se de inaktive operatører. Forneden ses et billede af "Vis operatører" hvor der også vises de inaktive operatører:

Operatørkartotek

CDIO gruppe 16

Vis operatører

Tilføj operatør

Ret operatør

Slet operatør

Vis operatører

Skjul inaktive operatører

ID	Navn	Initialer	CPR	Password	Aktiv	Niveau
31	Nicklas Hansen	NH	1205910000	passNH31	1	2
32	Jens Hansen	JH	1212121212	passJH32	1	1
33	Hanne Hansen	HH	1212121212	passHH33	1	1
34	Gurli Mogensen	GM	2173985783	passGM34	1	1
35	Mogens Jensen	MJ	1212121212	passMJ35	1	1
36	Frederik Adamsen	FA	1212121212	passFA36	1	1
37	Victor Oernby	VO	1212121212	passVO37	1	1
38	Mogen Svendsen	MS	1212121212	passMS38	0	1
39	Robert Eriksen	RE	1212121212	passRE39	1	1
40	Mogens Inaktiv	MI	1212121212	passMI40	0	1
41	Mogens Inaktiv2	MI	1212121212	passMI41	0	1
42	Mogens Inaktiv3	MI	1212121212	passMI42	0	2
43	Anders And	AA	1111111111	passAA43	1	1
44	Andersine And	AA	1010101010	passAA44	1	1

web development by bryant smith, edited by CDIO group 16

Tilføj operatør

En superbruger skulle også kunne oprette en ny operatør. Forneden ses et billede af vores implementation af denne funktionalitet:

The screenshot shows a web application titled "Operatørkartotek" by "CDIO gruppe 16". It features a navigation bar with four links: "Vis operatører", "Tilføj operatør", "Ret operatør", and "Slet operatør". The "Tilføj operatør" link is active. Below the navigation bar is a form titled "Tilføj operatør" with the following fields and validation rules:

Field	Validation Rule
Navn:	(min. 2 og max. 20 karakterer)
Initialer:	(min. 2 og max. 3 karakterer)
CPR:	(10 karakterer)
Password:	(7-8 karakterer og skal overholde DTU's regler for password.)
Aktiv:	(1 = aktiv, 0 = inaktiv)
Niveau:	(1 = operatør, 2 = superbruger)

Below the fields is a "Tilføj" button. At the bottom of the page, it says "web development by bryant smith, edited by CDIO group 16".

Alle tekstboksene er udstyret med felt validering, via en KeyUp handler der kalder en statisk metode i FieldVerifier. Kravene for et gyldigt felt er skrevet til højre for hver tekstboks. Tilføj knappen bliver først gyldig og klikbar når alle felter er gyldige samtidigt.

Ret operatør

Man skal også kunne rette operatører. Dette view har en blanding af funktionalitet fra “Vis operatører” og “Tilføj operatør”, idet at man både kan se operatører men også ændre i dem, hvor hver tekstboks har validering af felterne.

Operatørkartotek

CDIO gruppe 16

[Vis operatører](#)[Tilføj operatør](#)[Ret operatør](#)[Slet operatør](#)

Ret operatører

Vis inaktive operatører

ID	Navn	Initialer	CPR	Password	Aktiv	Niveau	
31	<input type="text" value="Nicklas Hansen"/>	<input type="text" value="NH"/>	<input type="text" value="1205910000"/>	<input type="text" value="passNH31"/>	<input type="text" value="1"/>	<input type="text" value="2"/>	<div>okcancel</div>
32	Jens Hansen	JH	1212121212	passJH32	1	1	edit
33	Hanne Hansen	HH	1212121212	passHH33	1	1	edit
34	Gurli Mogensen	GM	2173985783	passGM34	1	1	edit
35	Mogens Jensen	MJ	1212121212	passMJ35	1	1	edit
36	Frederik Adamsen	FA	1212121212	passFA36	1	1	edit
37	Victor Oernby	VO	1212121212	passVO37	1	1	edit
39	Robert Eriksen	RE	1212121212	passRE39	1	1	edit
43	Anders And	AA	1111111111	passAA43	1	1	edit
44	Andersine And	AA	1010101010	passAA44	1	1	edit

web development by bryant smith, edited by CDIO group 16

Slet operatør

Som tidligere nævnt, under afsnittet "Database", skal det ikke være muligt at slette en operatør totalt fra systemet. Denne skal blot have ændret sin status fra aktiv til inaktiv (i praksis fra 1 til 0). Det er denne funktion "Slet operatør" siden har.

Af gode grunde vises kun operatører her, der er aktive. Det giver ikke mening at sætte en inaktiv operatørs status til at være inaktiv.

Forneden ses vores implementation af at slette en operatør:

Operatørkartotek

CDIO gruppe 16

[Vis operatører](#)[Tilføj operatør](#)[Ret operatør](#)[Slet operatør](#)

Slet operatør(er)

ID	Navn	Initialer	CPR	Password	Aktiv	Niveau		
31	Nicklas Hansen	NH	1205910000	passNH31	1	2	ok	cancel
32	Jens Hansen	JH	1212121212	passJH32	1	1	delete	
33	Hanne Hansen	HH	1212121212	passHH33	1	1	delete	
34	Gurli Mogensen	GM	2173985783	passGM34	1	1	delete	
35	Mogens Jensen	MJ	1212121212	passMJ35	1	1	delete	
36	Frederik Adamsen	FA	1212121212	passFA36	1	1	delete	
37	Victor Oernby	VO	1212121212	passVO37	1	1	delete	
39	Robert Eriksen	RE	1212121212	passRE39	1	1	delete	
43	Anders And	AA	1111111111	passAA43	1	1	delete	
44	Andersine And	AA	1010101010	passAA44	1	1	delete	

web development by bryant smith, edited by CDIO group 16

Konklusion

Det var opgaven i dette projekt at udvikle et operatørkartotek med grafisk brugerflade, hvor man skulle kunne logge ind som forskellige brugere og udføre elementer fra CRUD terminologien alt efter hvilken brugertype der logges ind som.

Vi har i løbet af projektet stiftet bekendtskab med GWT, Google Web Toolkit, og er blevet mere bekendt med hvordan dette værktøj virker. Vi har også fået en bedre forståelse af hvordan en hjemmeside er bygget om, navnlig begreberne HTML og CSS. Vi har også brugt MySQL til at opsætte en database så der i projektet forefindes et persistent datalag.

Vi har fået udviklet en applikation i Java og GWT der udgør det meste af den ønskede funktionalitet. Vi har fået designet en hjemmeside i GWT med bagvedliggende Java kode til server funktionaliteten samt en SQL database som et persistent datalag. Applikationen er ikke fuldstændig funktionel i det at login funktionen ikke helt tager højde for ens brugerniveau. Der skulle have været lavet forskellige views til forskellige brugere men dette har der ikke været tid til at udvikle færdigt.

Vi har så vidt muligt forsøgt at følge tankegangen bag Unified Process og tre-lags modellen hvor man deler koden op i boundary, function og data lag og hvor der kodes op mod interfaces.