



02324

VIDREGÅENDE PROGRAMMERING

CDIO - Del 3

Gruppe 17

s185118
Aleksander Lægsgaard Jørgensen



s175561
Andreas Østergaard Schliemann

s185096
Jonatan Amtoft Dahl



s164177
Josephine Weirsøe

s185103
Søren Hother Rasmussen



s185121
Theodor Peter Guttesen

8. maj 2019

DTU Compute
Institut for Matematik og Computer Science

DTU Diplom
Center for Diplomingeniøruddannelse

Kode og Git: https://github.com/aleksanderlj/17_CDIO3

Indhold

1	Indledning	2
2	Timeregnskab	2
3	Krav	3
3.1	Kravliste	3
3.2	Domænemodel	3
4	Analyse	4
4.1	Use case	4
4.2	Traceability matrix: Use cases og krav	5
5	Implementering	5
6	Test	6
7	Konklusion	6

1 Indledning

Der ønskes en webbrowser, hvor det er muligt at oprette en bruger i en database og se en brugerliste. Det er meningen, at webbrowseren taler sammen med Java og MySQL. I dette projekt har vi implementeret disse koncepter, og fået lavet et program som kan tale på tværs af alle disse platforme. Dog som nævnt senere i opgaven løb vi ind i et problem som var markant mere indviklet end først forventet.

2 Timeregnskab

Opgave	Josephine	Aleksander	Søren	Jonatan	Andreas	Theodor	Sum
Rapport	10	1	0	10	10	10	41
Kode	0	80	56	0	0	20	156
Sum	10	81	56	10	10	30	197

3 Krav

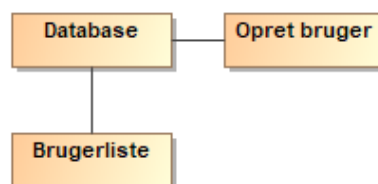
3.1 Kravliste

ID	Funktionelle krav
Must have	
M01	Alle med adgang til webbrowserne skal kunne oprette alle typer brugere
M02	Systemet skal understøtte de 4 brugertyper: Admin, Pharmaceut, Produktionsleder, Laborant
M03	Systemet skal kunne oprette bruger
MO4	Systemet skal kunne vise en brugerliste

ID	Ikke-funktionelle krav
Must have	
IF01	Systemet skal være implementeret som en Single page application med en REST-backend og en HTML/CSS/JavaScript baseret frontend.
IF02	Data skal lagres i en persistent database mellem sessioner.
IF03	Der skal sendes til/fra Rest-backenden serialiseret på en måde (eks. JSON) så andre kan anvende servicen
IF04	Modulet skal kunne tilgås via. en webbrowser

3.2 Domænemodel

Dette er en domænemodel, som viser, hvordan de forskellige klasser taler med hinanden. Lige i denne model vises, hvordan databasen snakker med de to forskellige dele af hjemmesiden. I virkeligheden vil der ligge noget java kode mellem databasen og hjemmesiden. Samtidigt vil der også være grænsekode mellem hjemmesiden og java koden, dette vil f.eks. være gennem JSON-objekter.



Figur 1: Domænediagram

4 Analyse

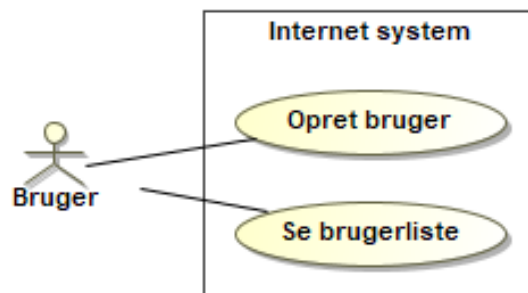
4.1 Use case

Use case: Opret bruger
ID: UC01
Kort beskrivelse: En bruger oprettes med rolle og login.
Primære aktører: Administrator
Sekundære aktører: Bruger
Preconditions: Systemet er åbent
Main flow: 1. Roller vælges 2. Brugernavn og initialer vælges
Postconditions: Brugeren er oprettet.
Alternative flows:

Use case: Se brugerliste
ID: UC01
Kort beskrivelse: En bruger ser alle brugere i databasen.
Primære aktører: Administrator
Sekundære aktører: Bruger
Preconditions: Systemet er åbent
Main flow: 1. Bruger åbner hjemmeside
Postconditions: Brugeren kan se alle brugere i databasen.
Alternative flows:

Use case diagram

Vi har valgt at kun sætte en aktør, som er brugeren. Vi har valgt dette, grundet, at der i opgavenbeskrivelsen ikke er nogle kunder eller lignende, som ellers potentielt ville være endnu en aktør.



Figur 2: Use case diagram

4.2 Traceability matrix: Use cases og krav

I følgende tabel ses sammenhængen mellem use cases og krav. De grønne er dem, der er implementeret, de røde er endnu ikke implementeret.

Krav/Use Case	UC01	UC02
M01	x	
M02	x	
M03	x	
M04		x

5 Implementering

Vi har som opgaven krævede implementeret to data lag. Et transient og et persistent. Det transiente lag består af UserDTO objekter, der er sat ind i en arrayliste. Det persistente er en MySQL database. Det største udfordring vi skulle igennem var et link mellem hjemmeside og den bagved liggende java kode. Vi havde svært ved at få sendt JSON objekter fra javascript og over til java koden. Derfor kan vi sige at det sværere ved dette projekt var at få linker mellem bruger grænsefladen og funktionalitets laget.

6 Test

Funktionelle krav	Positiv test	Bestået
M01	Alle med adgang til webbrowseren skal kunne oprette alle typer roller	Bestået
M02	Systemet skal understøtte de 4 brugertyper	Bestået
M03	Systemet skal kunne oprette bruger	Bestået
M04	Systemet skal kunne vise en brugerliste	Bestået

7 Konklusion

Til sidst blev vores projekt godt, og vi er tilfredse med det projekt vi fik lavet til sidst. Selve processen gik meget forventet og var den største udfordring. Vi havde først estimeret at vi skulle bruge et par dage, og være færdig med hele opgaven på den tid. Dog viste det sig hurtigt at dette var en stor undervurdering af opgaven. Vi løb ind til problemer med forkerte versioner af java, tomcat og flere. Da vi havde flere opgaver måtte vi derfor tage en stor del af holdet af projektet, og have nogle få fokusere på CDIO. Efter lang slidning fik Aleksander lavet et succesfuld program, som kunne det det skulle og fungerede godt.

I forhold til forrige CDIO projekter, har dette projekt været det projekt hvor mængden af arbejde har været mest overraskende. På grund af at processen blev så anderledes end forventet, så har dette CDIO projekt været meget anderledes end forrige, projektet har været meget mere splittet op, og det er også første gang, at vi har splittet gruppen op, for at fokusere på forskellige andre projekter.