



Danmarks Tekniske Universitet

3-ugers projekt

02324 - Videregående programmering

Gruppe 11

25-06-2020



Mikkel Danielsen
s183913



Frederik Koefoed
s195463



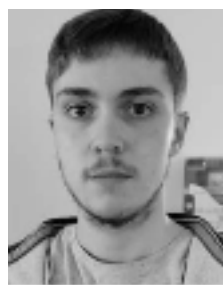
Muhammad Talha
s195475



Volkan Isik
s180103



Lasse Strunge
s195486



Mark Mortensen
s174881

Indholdsfortegnelse

Indledning	3
Kravliste - MOsCOW? JA!	4
Systemkrav	4
Analyse	6
UseCase Diagram	7
Use case beskrivelse	7
Fully dressed	8
Navigeringsdiagram	10
Domæne model	11
Systemsekvensdiagram	11
Design	12
Design Class Diagram	14
Dependency diagram over web	15
Implementering	16
Overvejelser	16
3-lags modellen	16
Database server	17
Demodata	18
Platform	18
Pakkediagram:	19
Brugervejledning	22
Test	22
Bilag	24

Indledning

.....
.....

Noter

- Vi har brugt onclick i buttons til at aktivere js i stedet for at via js at target de enkelte elementer. På den måde kan kode også nemt genbruges
- .getRole(String role)
- Default.css

Kravliste¹ - MOsCOW? JA!

Systemkrav

K?? M - Der skal være et web interface (GUI)

K?? M - Dataen skal gemmes i en database

K?? C - En bruger må aldrig slettes fra databasen, men kun "inaktiveres"

K?? S - Alle input til GUI skal valideres og hvis der gives et ugyldigt input skal systemet skrive at der er en fejl

K?? -

K?? -

¹ Bilag 1

Bruger administration

- K3 M - Det skal være muligt for en administrator at oprette en bruger
- K4 M - Når man opretter en bruger skal man kunne angive deres navn, initialer, cpr-nummer samt deres roller
- K5 S - Man skal kunne se data for alle oprettede brugere på en samlet liste
- K6 M - Brugerens ID skal være mellem 11 og 99
- K7 S - Man skal kunne ændre i brugerens oplysninger (Navn, initialer, cpr-nummer, password)
- K9 M - Der skal være 4 roller (Admin, Pharmacist, Foreman, Operator)
- K10 M - Brugernavnet skal være mellem 2 og 20 karakterer
- K11 M - Initialerne skal være mellem 2 og 4 tegn
- K12 M - Passwordet skal indeholde mellem 6 og 50 tegn
- K15 C - Når en ny bruger oprettes skal der genereres et password til dem
- K16 M - Når en ny bruger oprettes skal de have det laveste ledige ID

Råvare administration

- K?? M - *farmaceuten* skal kunne oprette råvarer
- K?? M - Råvarer skal oprettes med id, navn og leverandører.
- K?? M - En *råvare* defineres ved et råvareNr (**brugervalgt** og entydigt) og navn. (d.v.s. råvareNr skal IKKE autogenereres.)
- K?? S - Farmaceuten skal kunne se råvarer i en samlet liste
- K?? C - Farmaceuten skal kunne opdatere råvarer i den samlede liste
- K18 M - Systemet kan kunne definere *Råvare*: Kommer fra forskellige leverandører i *RåvareBatches* med given mængde af råvarer.

Recept-administration

- K?? M - En recept skal have et receptnummer og en eller flere receptkomponenter
- K?? M - En receptkomponent består af en råvare, en mængde og en tolerance
- K?? S - Råvaren skal findes i systemet for at være en del af en recept
- K?? M - En farmaceut skal kunne oprette en recept, hvori der bestemmes hvilke råvare der skal bruges samt mængden og tolerancen af disse.

Råvarebatch-administration

- K?? M - Produktionslederen skal kunne oprette et råvarebatch
- K?? S - Produktionslederen skal kunne vises alle råvarebatches
- K?? M - Et råvarebatch skal have et råvarebatchnummer, en mængde og en leverandør
- K?? C - Mængden i råvarebatchet skal automatisk opdateres når der bliver brugt noget af en råvare i et produktionsbatch

Produktbatch-administration

K?? M - Produktionslederen skal kunne oprette produktbatch

k?? M - Produktbatch skal oprettes med id, receptid, status, userid, råvarebatch id, tara i kilogram og netto vægt i kilogram

k?? M - Afvejningsresultater for de enkelte *receptkomponenter* gemmes som en *produktbatchkomponent* i *produktbatchen* i takt med at produktet fremstilles.

k?? S - Produktionslederen skal kunne opdatere en produktbatch

k?? C - Produktionslederen skal kunne printe produktbatch i et dokument.

Afvejning

K?? M - Systemet skal fortælle laboranten hvilken ingrediens, samt mængde og tolerancen før han begynder at veje

K?? M - Laboranten skal kunne skrive vægt ind på websiden

K?? M - Hvis vægten ikke er tæt nok på den vægt der er givet i recepten, skal websiden melde fejl og laboranten må veje igen

K?? C - Laboranten skal identificere sig selv før de kan bruge systemet

K?? M - Når laboranten indtaster produktbatch-id'et skal sytemet kunne finde det pågældende produktbatch og laboranten redigerer herefter i netop det produktbatch

K?? S - Laboranten skal kunne bestemme produktbatchets tara.

K?? M - Før afvejningen startes skal produktbatchets status være ikke påbegyndt

K?? M - Når den første afvejning er sket skal produktbatchets status gå fra ikke påbegyndt til under produktion

K?? M - Når alle receptkomponenter er afvejet skal produktbatches status gå fra under produktion til afsluttet

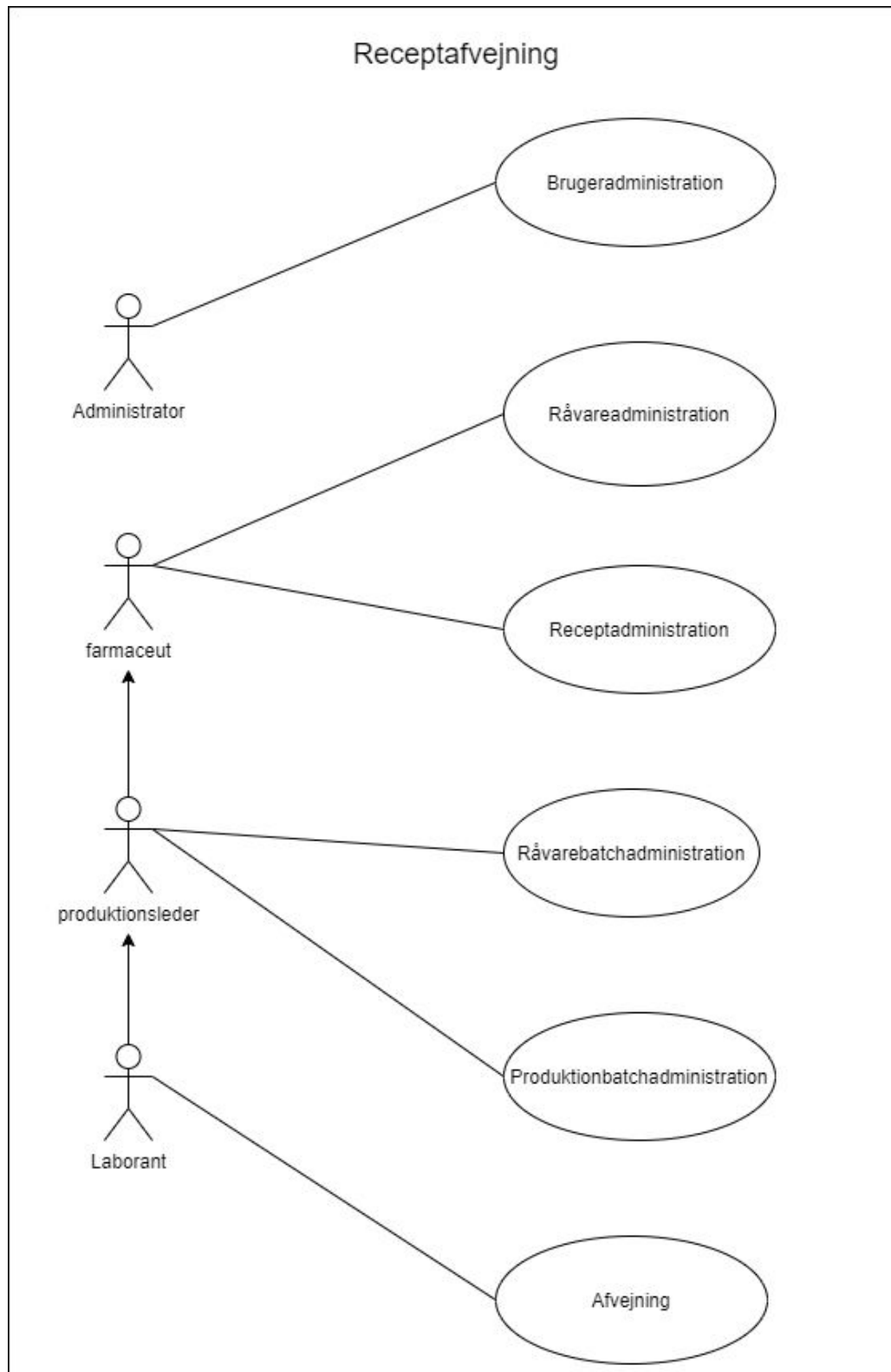
evt ikke funktionele krav

Rightbillede

Analyse

usecase liste og aktør tabel

UseCase Diagram



Use case beskrivelse

Fully dressed

Use Case Section	Comment
Navn	Opret bruger
Scope	CDIO final
Level	User-goal
Primær aktør	Bruger
Preconditions:	Hjemmesiden er startet og brugeren er logget ind.
Postconditions:	En ny bruger er oprettet
Hoved succes scenarie:	<ol style="list-style-type: none">1) Brugeren trykker på knappen "Opret Ny Bruger"2) Navn bliver indtastet.3) Initial bliver indtastet.4) CPR-nummer bliver indtastet.5) Password bliver indtastet.6) En rolle eller flere roller vælges fra menuen.7) Brugeren trykker på knappen "Opret".8) Brugeren bekræfter oprettelse ved at trykke på "ok".9) Brugeren bliver oprettet.
Alternative scenarier:	<p>2-6: Brugeren ikke indtaster på nogle af de felter, 7: Brugeren trykker på knappen "Opret". 8: Brugeren får fejlmeddelelse at den skal udfylde de tomme felter og bruger trykker på "ok", og udfylder de tomme felter. 7: Brugeren trykker på knappen "Opret". 8a: Brugeren bekræfter oprettelse ved at trykke på "ok". 8b: Brugeren trykker på "cancel", der sker ikke noget man bliver på samme side.</p> <p>2-6: Brugeren indtaster alle oplysninger 7: Brugeren trykker på knappen "Opret". 8a: Brugeren trykker på "cancel" og der sker ikke noget man bliver på samme side. 8b: Brugeren trykker på "ok" og brugeren bliver oprettet.</p>

Use Case Section	Comment
Navn	Vis brugere
Scope	CDIO 2

Level	User-goal
Primær aktør	Bruger
Preconditions:	Hjemmesiden er startet
Postconditions:	Liste af brugere bliver vist
Hoved succes scenarie:	1) Brugeren indtaster brugernavn og password. 2) Brugeren trykker på "login" 3) Brugeren er logget ind. 4) Brugeren kan nu se alle de oprettede brugere i systemet.

Fully dressed

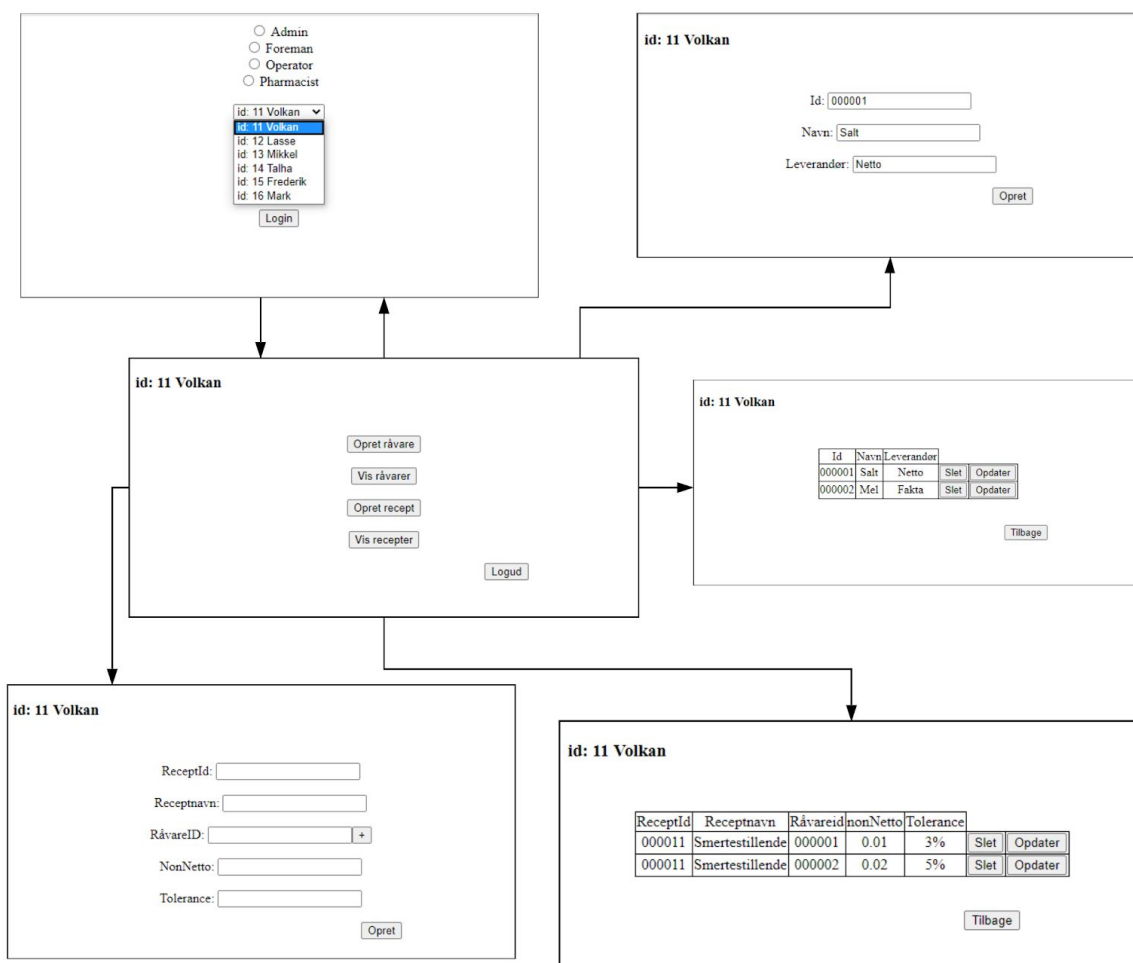
Use Case Section	Comment
Navn	Opdater bruger
Scope	CDIO 2
Level	User-goal
Primær aktør	Bruger
Preconditions:	Programmet startet og en database valgt
Postconditions:	En bruger bliver ændret
Hoved succes scenarie:	1) Brugeren trykker på knappen "update" bruger. 2) Brugeren indtaster ændringerne som skal ændres (Brugernavn, initialer, cpr, password, role). 3) Brugeren trykker på knappen "update". 4) Brugeren bekræfter sine ændringer og trykker "ok".
Alternative scenarier:	4.a: Brugeren trykker på "OK" og nu er brugeren har ændret sine oplysninger. 4.b: Brugeren trykker på "CANCEL" og opdater siden bliver gentaget igen.

Fully dressed

Use Case Section	Comment
Navn	Slet bruger
Scope	CDIO 2

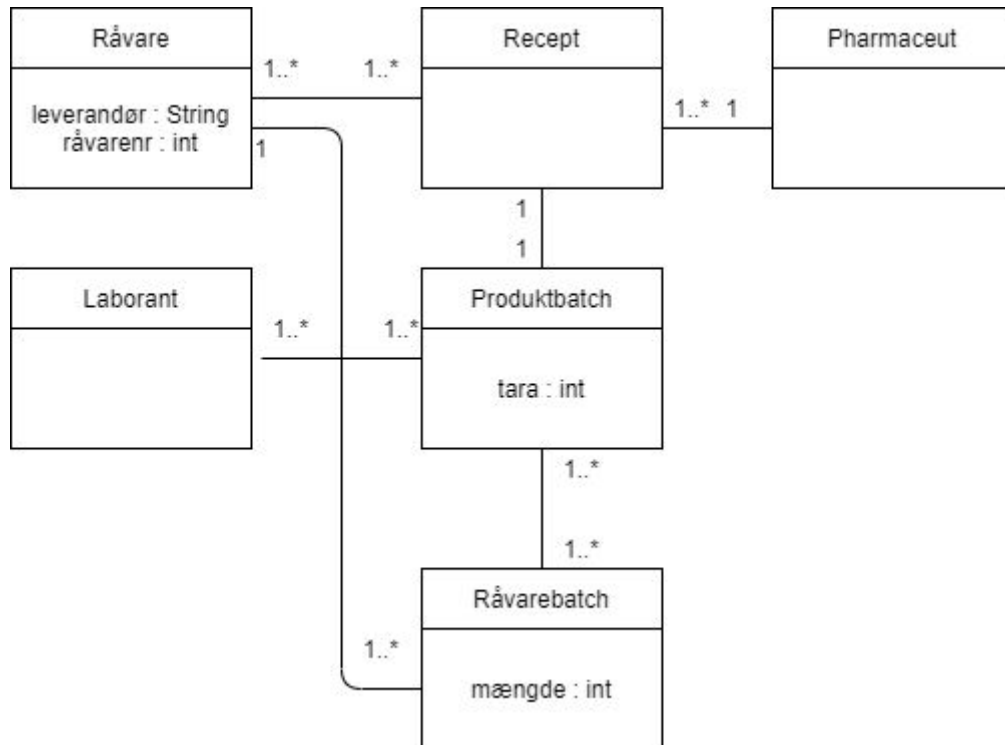
Level	User-goal
Primær aktør	Bruger
Preconditions:	Hjemmesiden startet og en brugeren er logget ind
Postconditions:	En brugere er blevet slettet
Hoved succes scenarie:	<ol style="list-style-type: none"> 1. Brugeren trykker på "slet" knappen. 2. Brugeren får en besked på toppen af skærm, om de sikker på at slette den bruger. 3. Brugeren trykker på "ok" 4. Den valgte bruger bliver slettet fra systemet
Alternative scenarier:	<ol style="list-style-type: none"> 1. Brugeren følger trin 1-2 fra hoved succes scenarie. 3.a: Brugeren trykker på "cancel" og der sker ikke noget 3.b: Brugeren trykker på "ok", og brugeren er slettet fra system.

Navigeringsdiagram



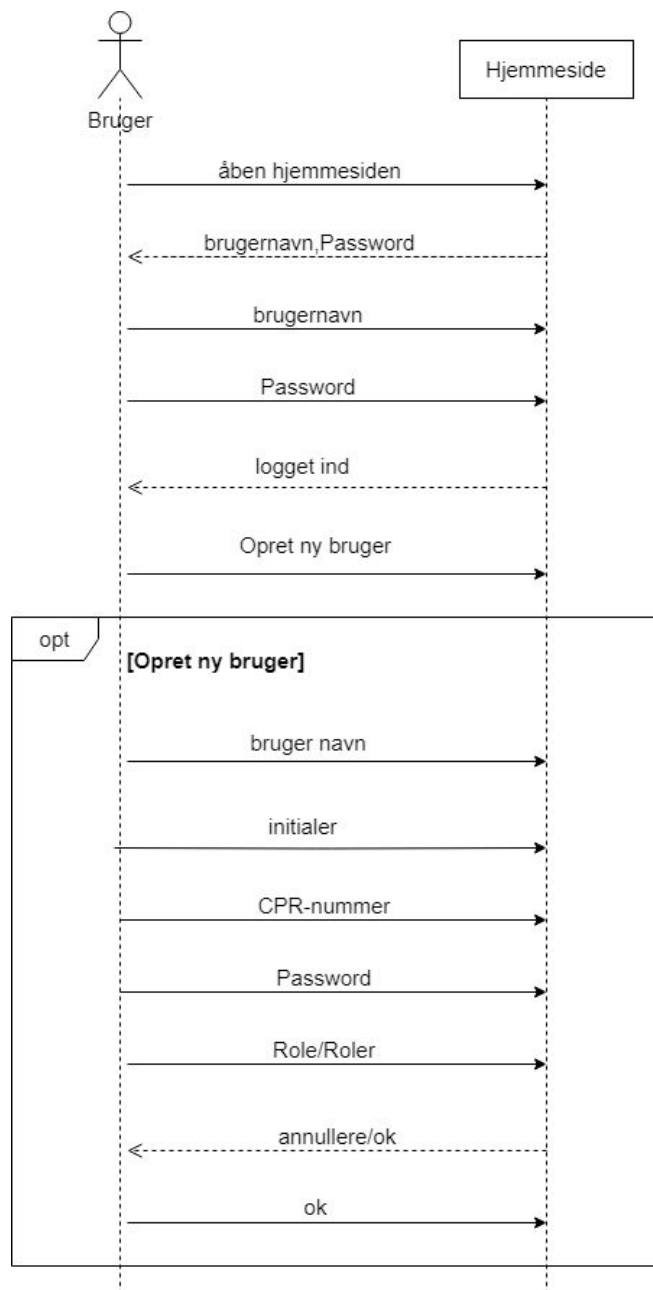
Aktivitetsdiagrammer

Domæne model



Systemsekvensdiagram

Nedenfor ses et systemsekvensdiagram over use casen "opret bruger":



Design

Klassebeskrivelser

Functionality: Logikken for validering af input og **andre** funktioner.

AppConfig: Definition af applikation stien.

UserAPI: Indeholder Rest webservice med Jersey for UserDTO

RåvareAPI: Indeholder Rest webservice med Jersey for RåvareDTO

RåvareBatchAPI: Indeholder Rest webservice med Jersey for RåvareBatchDTO

ReceptAPI: Indeholder Rest webservice med Jersey for ReceptDTO

ProduktbatchAPI: Indeholder Rest webservice med Jersey for ProduktBatchDTO

SQLDatabaseIO: Kommunikation af SQL databasen (query/manipulation)

UserDTO: UserDto er objektet der bliver transmitteret igennem vores system

UserController: Manipulerer og henter data fra UserDAO SQL ved hjælp af UserAPI

UserDAO SQL: Manipulerer og henter data fra fjerner fra User tabel i mySQL database

RåvarerDTO: RåvarerDTO er objektet der bliver transmitteret igennem vores system

RåvareController: Manipulerer og henter data fra RåvareDAO SQL ved hjælp af RåvareAPI

RåvarerDAO SQL: Manipulerer og henter data fra fjerner fra Råvare tabel i mySQL database

RåvarebatchDTO: RåvarebatchDTO er objektet der bliver transmitteret igennem vores system

RåvarebatchDAO SQL: Manipulerer og henter data fra fjerner fra Råvarebatch tabel i mySQL database

RåvarebatchController: Manipulerer og henter data fra RåvarebatchDAO SQL ved hjælp af RåvarebatchAPI

ReceptDTO: ReceptDTO er objektet der bliver transmitteret igennem vores system

ReceptController Manipulerer og henter data fra ReceptDAO SQL ved hjælp af ReceptAPI

ReceptDAO SQL: Manipulerer og henter data fra fjerner fra Recept tabel i mySQL database

ReceptkompDTO??

ProduktbatchDTO:: ProduktbatchDTO er objektet der bliver transmitteret igennem vores system

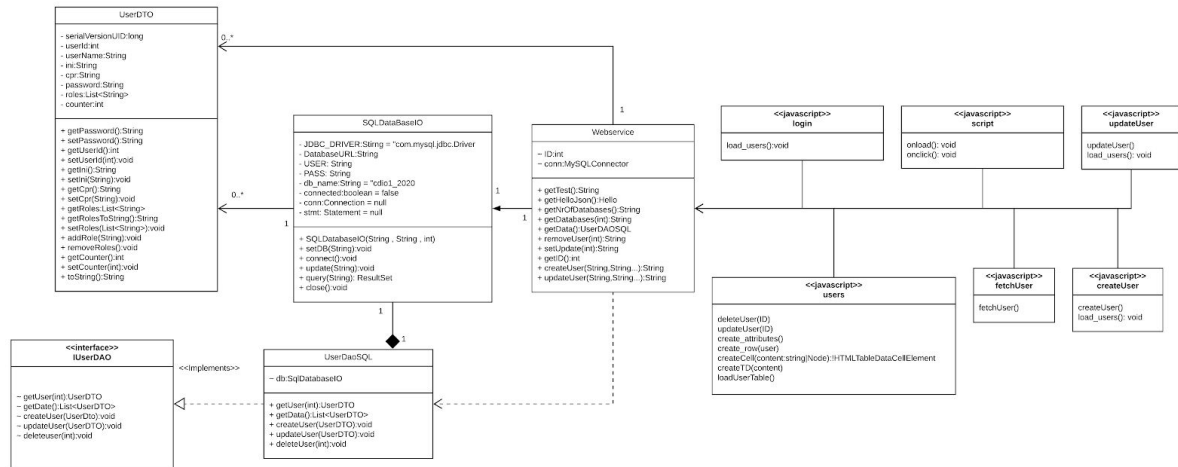
ProduktbatchController: Manipulerer og henter data fra ProduktbatchDAO SQL ved hjælp af ProduktbatchAPI

ProduktbatchDAO SQL: Manipulerer og henter data fra fjerner fra Produktbatch tabel i mySQL database

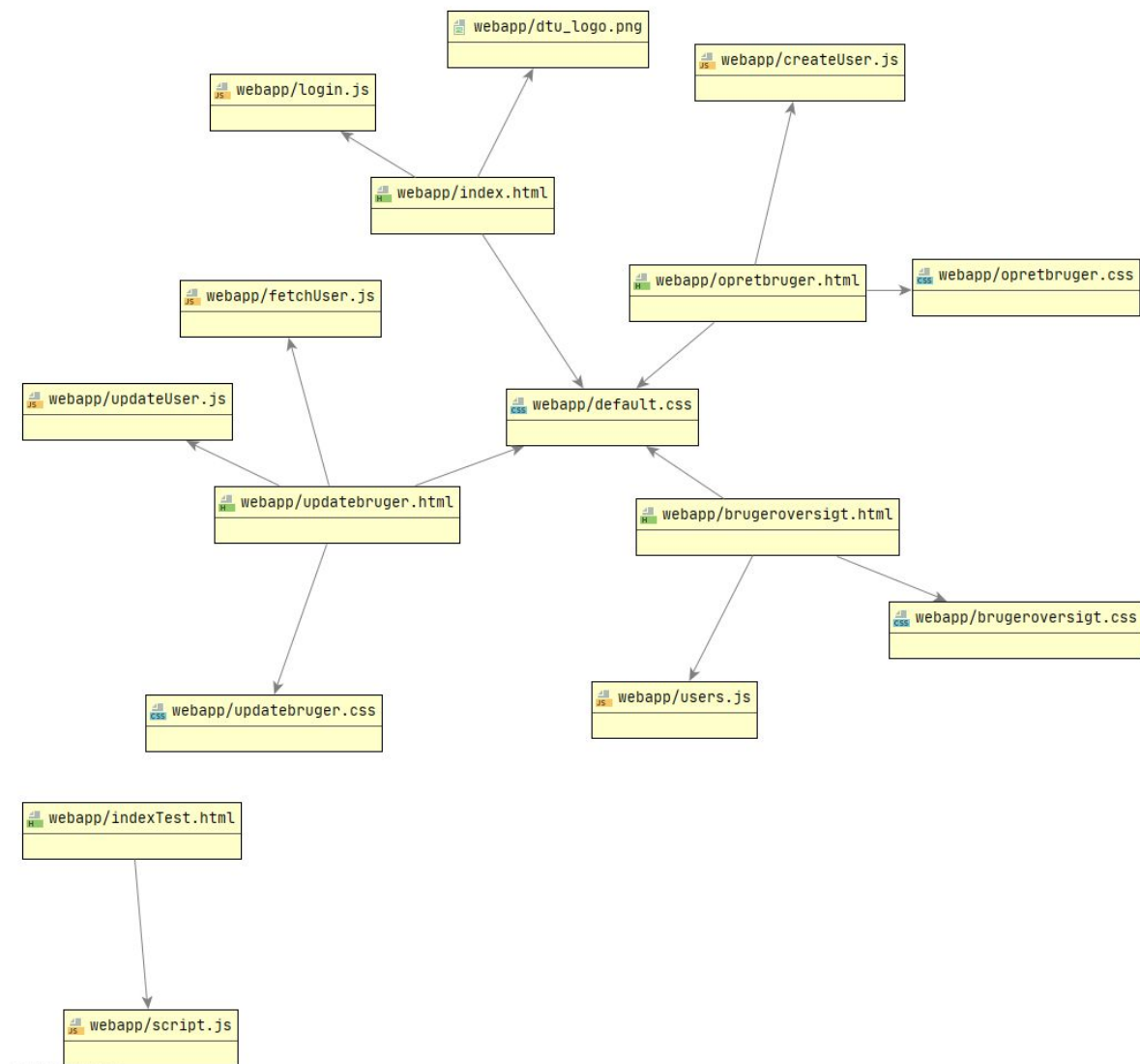
ProduktbatchkompDTO:??

Design Class Diagram

Design
Klassediagram - CDIO 1



Dependency diagram over web



Powered by [yarn](#)

Implementering

Overvejelser

3-lags modellen

Vi har i vores projekt valgt at benytte os af trelagsmodellen, for at gøre systemets struktur mere overskuelig og de enkelte moduler nemmere at forstå. Samtidig bidrager modulariseringen til et system der er mere genanvendeligt, nemmere at vedligeholde samt lettere at udvide i fremtiden, da de enkelte delelementer kan betragtes isoleret.²

Vores grænseflade findes i webapp mappen og består af en række .html, .css og .js filer. HTML og CSS styrer strukturen og layoutet på hjemmesiden og JavaScript står for kommunikation med de forskellige webservices. Vi har desuden brugt JQuery biblioteket for at manipulere og kommunikere med backenden.

Vores controllerlag indeholder de forskellige webservices, der er dekorerede med vha. Jersey til at definere forskellige resources og actions. Vha. Jackson kan vi nemt oversat JSON til klassiske Java-objekter og omvendt.

Til sidst har vi datalaget, som controllerlaget arbejder tæt sammen med. Her finder vi de nødvendige klasser for at oprette en forbindelse til og kommunikere med vores database. Bemærk hvordan vi til at overføre information mellem de forskellige lag benytter en såkaldt DTO (data transfer objekt), til at overskueliggøre processen.

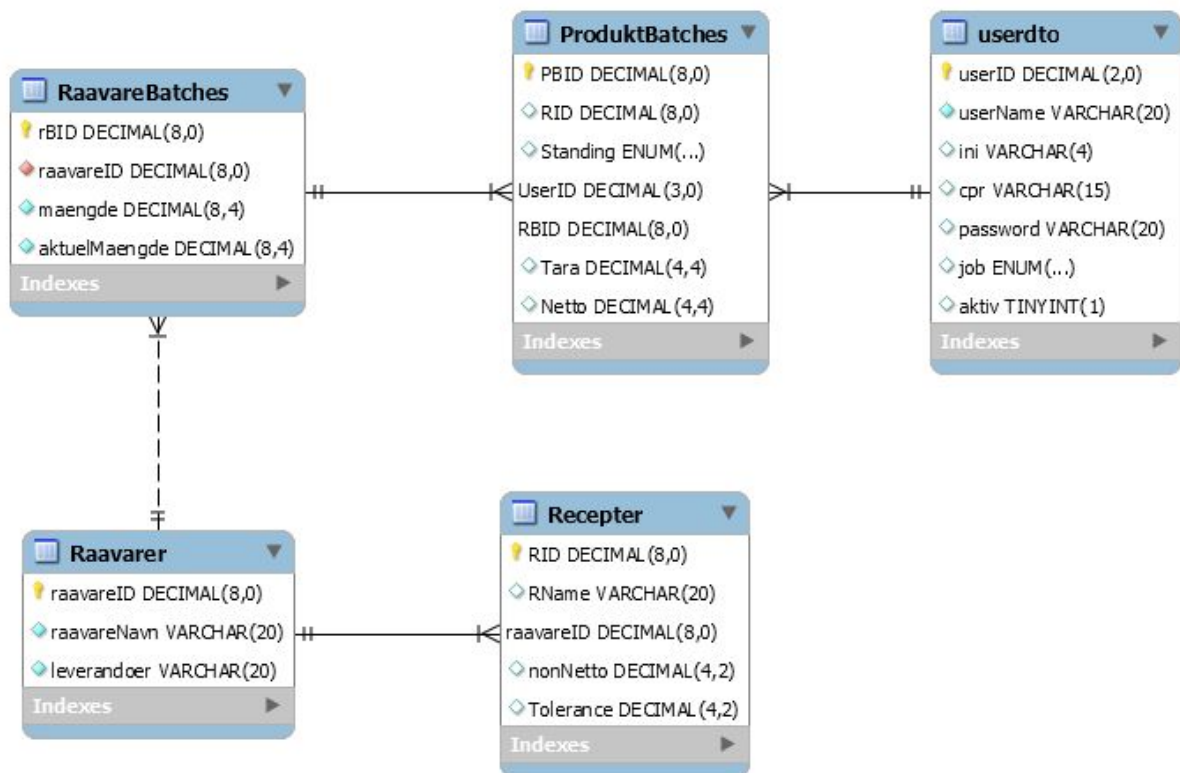
Som det kan ses, indeholder datalaget forskellige interfaces , hvilket vi med fordel også kunne have anvendt i de andre lag. Dette ville have forstærket effekt af modulariseringen yderligere og gjort vores software mere fleksibel.

Da der i vores løsning kun forekommer rene datamanipulationsopgaver, var der ikke har behov for et funktionalitetslag. Skulle vi også have implementeret f.eks. kodeordsgodkendelse, havde det været en fordel med et funktionalitetslag. Bemærk også, hvordan vi benytter en såkaldt DTO (data transfer objekt) til at overføre information mellem de forskellige lag, for at overskueliggøre processen.

² "3 lags modellen" af Mads Nyborg og Uffe Kofoed, 2013, s. 1-2.

Database server

EER diagram



i virkligheden har man mange til mange relation i mellem råvarer og recepter.(En recept indeholder mange råvarer og en råvare kan være i mange recepter)

Samme problemer har vi også med produktbatches og userdto.

vi har efterfulgt opgavebeskrivelsen!!! hvad gør vi her. Er det derfor man har receptKomp classer i opgavebeskrivelse ?

Vi har i vores løsning valgt at benytte en online MYSQL server som database, for at undgå at brugeren skal have en lokal database server liggende. Databasen er opsat med tre tabeller: userdto (brugeren), roles (roller) og userdto_roles (Hvilke userdto brugere har hvilke roller).

userdto

userID	userName	ini	cpr	password
--------	----------	-----	-----	----------

roles

role_name

userdto_roles

userID

role_name

Demodata

Nedenfor ses den demodata som alle systemets databaser er blevet initialiseret med

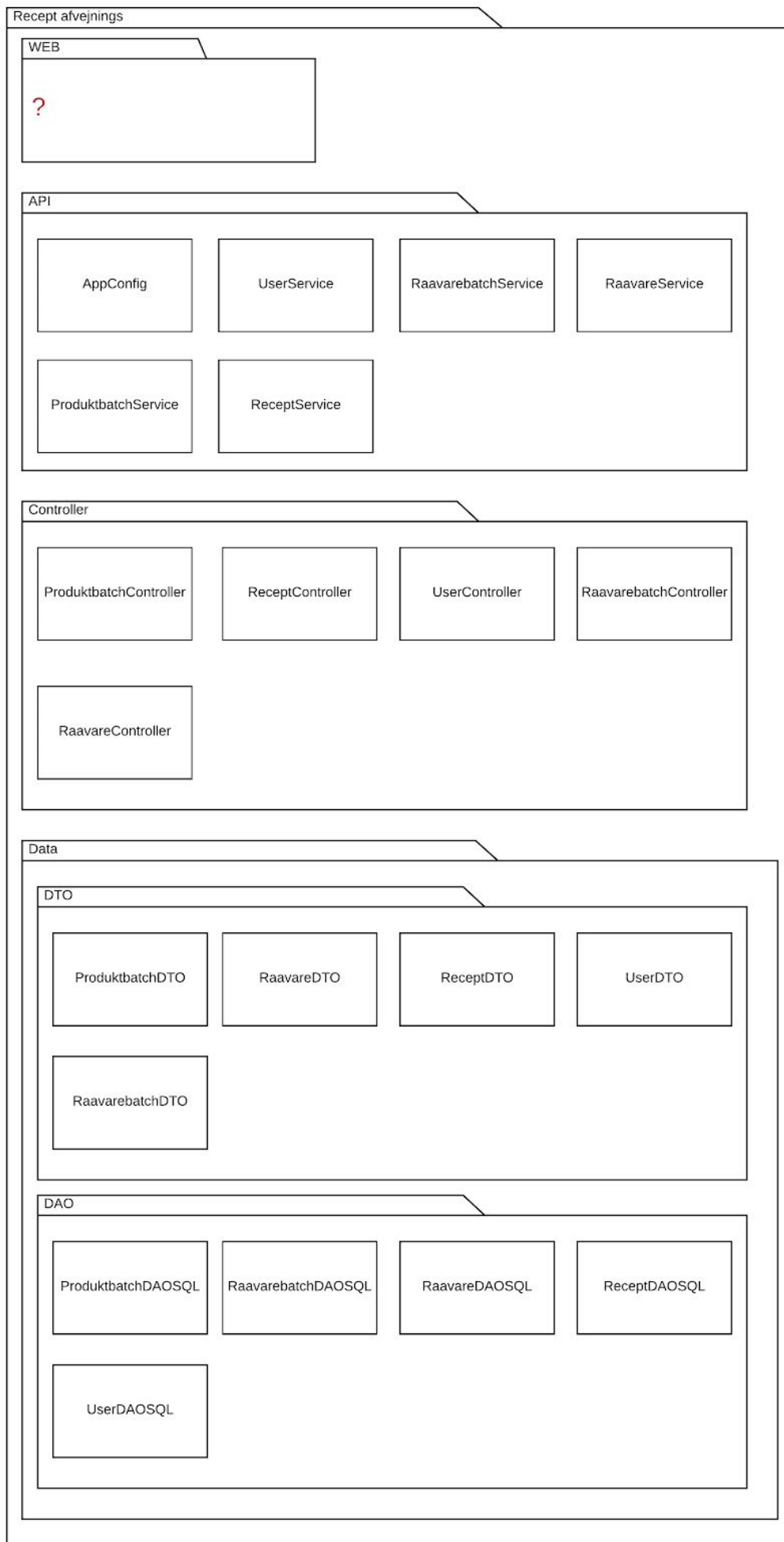
ID	Navn	Ini.	CPR	Kodeord	Roller
11	Volkan	VOL	0912941235	SkiftMig!	Admin
12	Mikkel	MIK	3007902345	SkiftMig!	Pharmacist
13	Mark	MAR	1603783455	SkiftMig!	Foreman
14	Talha	TAL	2301974565	SkiftMig!	Operator
15	Frederik	FRE	0501025675	SkiftMig!	Pharmacist, Operator
16	Lasse	LAS	1104966785	SkiftMig!	Admin, Pharmacist, Foreman, Operator

Platform

Operativsystemer testet på	Windows 10 Version 1903 Build 18362.476 MacOS 10.15.1 Ubuntu 19.10
Java	Java 1.8.0_161
IntelliJ	IntelliJ IDEA 2019.2.3 (Ultimate Edition) Build #IU-192.6817.14, built on September 24, 2019

Matadorgui	matadorgui-3.1.7.jar
------------	----------------------

Pakkediagram:



Brugervejledning

Installation og kørsel af CDIO1.jar

- Programmet kræver Java 8.
- Programmet er bygget til at køre på en Windows 10 maskine.
- Apache Tomcat 8.5.54 Core downloades [her](#)³ og pakkes ud.
- Udpak 11_del2.zip.
- Mappen udpakket åbnes som projekt i IntelliJ Ultimate edition 2020.1.1+
- Mavenfilen pom.xml genimporteres
- Øverst til højre vælges "Edit Configuration"
- Tryk på plus tegnet øverst til venstre i vinduet og Og tomcat > local vælges.
- Application server Configureres ved at vælge mappen apache-tomcat-8.5.54
- Hold alt andet standard of tryk på "fix" nederst til højre.
- Her vælges artifaktet "BoilerPlate:war exploded" og der trykkes "Apply" og "OK"
- Programmet køres ved tryk på den grønne pil øverst til højre.

³ <https://tomcat.apache.org/download-80.cgi>

Test

Test case ID	TC01
Summary	Tester om data på websiden bliver oprettet korrekt
Procedure	Klik opret ny bruger Skriv brugernavn: Bo Peter Skriv initialer BP Skriv CPR nummer 0102030405 Skriv password:test123 Vælg rolle Admin Tryk opret Kontrollér at det rigtige data står i de rigtige felter
Data	Bo Peter BP 0102030405 Test123 Admin
Status	Passed
Created by	Lasse Strunge
Creation date	10/05/2020
Last tested	10/05/2020
Environment	IntelliJ IDEA 2019.2.3 (Ultimate Edition) Build #IU-192.6817.14, built on September 24, 2019 on Windows 10 Enterprise Version 1903 OS build 18362.418 Google Chrome

Test case ID	TC02
Summary	Tester om slettefunktionen virker
Procedure	Klik slet bruger Tryk OK Kontrollér at brugeren ikke er der længere
Data	
Status	Passed
Created by	Lasse Strunge
Creation date	10/05/2020

Last tested	10/05/2020
Environment	IntelliJ IDEA 2019.2.3 (Ultimate Edition) Build #IU-192.6817.14, built on September 24, 2019 on Windows 10 Enterprise Version 1903 OS build 18362.418 Google Chrome

Ordbog

Konklusion

Bilag

Bilag 1: CDIO-D1 Rapport:

https://docs.google.com/document/d/1a9eWnNCtYzhWIhI8TpB49kwtZdvwW_2mMQBvW_alCTg/edit?usp=sharing