

Navn:

CPH-mailadresse

Githubnavn:

Klasse:

Christian Ambjørn Kehr:

cph-ck209@cphbusiness.dk

Christian-A-Kehr

B

Mads-Ulrik Korsgaard Hansen:

cph-mh752@cphbusiness.dk

MukHansen

B

Claus Mikkelsen Findinge:

cph-cf124@cphbusiness.dk

ClausFindinge

B

Cupcake



Afleveret 17.03-2018

Navn:	CPH-mailadresse	Githubnavn:	Klasse:
Christian Ambjørn Kehr:	cph-ck209@cphbusiness.dk	Christian-A-Kehr	B
Mads-Ulrik Korsgaard Hansen:	cph-mh752@cphbusiness.dk	MukHansen	B
Claus Mikkelsen Findinge:	cph-cf124@cphbusiness.dk	ClausFindinge	B

Indholdsfortegnelse

INDLEDNING OG BAGGRUND	3
TEKNOLOGIVALG	4
DIAGRAMMER OG MODELLER	5
DOMÆNEMODEL OG ER-DIAGRAM	5
NAVIGATIONS-DIAGRAM	6
SEKVENSDIAGRAM	9
Registration	9
Login	10
Shop (old version)	12
STATUS PÅ IMPLEMENTATION	13
UTILSIGTEDE MANGLER	13
BEVIDSTE VALG OG FRAVALG	13
TESTS	14

Navn:	CPH-mailadresse	Githubnavn:	Klasse:
Christian Ambjørn Kehr:	cph-ck209@cphbusiness.dk	Christian-A-Kehr	B
Mads-Ulrik Korsgaard Hansen:	cph-mh752@cphbusiness.dk	MukHansen	B
Claus Mikkelsen Findinge:	cph-cf124@cphbusiness.dk	ClausFindinge	B

INDLEDNING OG BAGGRUND

Formålet med opgaven var at programmere og dokumentere et onlinesystem, som kan håndtere en cupcake bagers ordre. Dette skal optimere bagerens virksomhed, ved at indføre et automatiseret system. Bageren ønsker at systemet kan; register bruger, log ind bruger ind på en personlig shop, hvor brugeren kan vælge cupcakes ud fra en liste eller ligne. Derefter gemmes online bestillingers samt bruger registreringer i databasen.

Gruppen har valgt at lave en servlet, benævnt frontcontroller, som håndtere en række JSP, herunder registrering, login, shop og shoppingcart. Dog skal det tilføjes, at login er styrende for om klienten får adgang til shop og shoppingcart.

Fremgangsmåden har været at indsætte nogle test-brugere, sådan at programmet kunne testes løbende. Når kunden var kommet ind på siden, skulle vedkommende mødes af et menukort. Her skulle kunden kunne krydse af, hvilke cupcakes vedkommende ønskede at købe. For hver gang kunden bestilte en cupcake, var hensigten, at denne skulle tilføjes kundens personlige kurv, så det var muligt at danne sig et overblik over hele bestillingen. I tilgift var ønsket, at kunden ikke automatisk ville miste sin bestilling, hvis vedkommende lavede et request til en anden side. Ønskede brugeren flere af samme slags cupcake, skulle det være muligt at gange op fra start i feltet "quantity", så vedkommende ikke skulle bestille den samme cupcake flere gange.

En cupcake består af en topping og en bottom. I stedet for at lade kunden vælge en topping og en bottom separat fra hinanden, ville vi liste alle kombinationsmuligheder op, så kunden kunne krydse af, hvilken kombination, vedkommende ønskede.

Et sådan system ville gøre det lettere for kunden at bestille cupcakes, samtidig med at det ville spare bageriet for en masse tid ressourcer på at tage i mod bestillinger. Hermed var der mulighed for at lægge mere energi i at producere cupcakes, samtidig med at kunden ville opleve en mere overskuelig og brugervenlig betjening.

I forhold til rapportens opbygning gør vi først rede for hvilke programmer og teknologier, vi har gjort brug af i forhold til at skabe systemet. Herefter følger en række diagrammer, der bl. a. har til opgave at danne et overblik over, hvordan vi har valgt at bygge systemet op. Til sidst vil vi redegøre for implementationen, og

Navn:	CPH-mailadresse	Githubnavn:	Klasse:
Christian Ambjørn Kehr:	cph-ck209@cphbusiness.dk	Christian-A-Kehr	B
Mads-Ulrik Korsgaard Hansen:	cph-mh752@cphbusiness.dk	MukHansen	B
Claus Mikkelsen Findinge:	cph-cf124@cphbusiness.dk	ClausFindinge	B

hvor vi befinder os i forhold til at besvare den opgave, vi er stillet. Vi vil også berøre årsagen til vores mangler i koden. I forlængelse heraf vil vi redegøre for de tests, der kunne have været relevante at lave i forhold til systemet.

TEKNOLOGIVALG

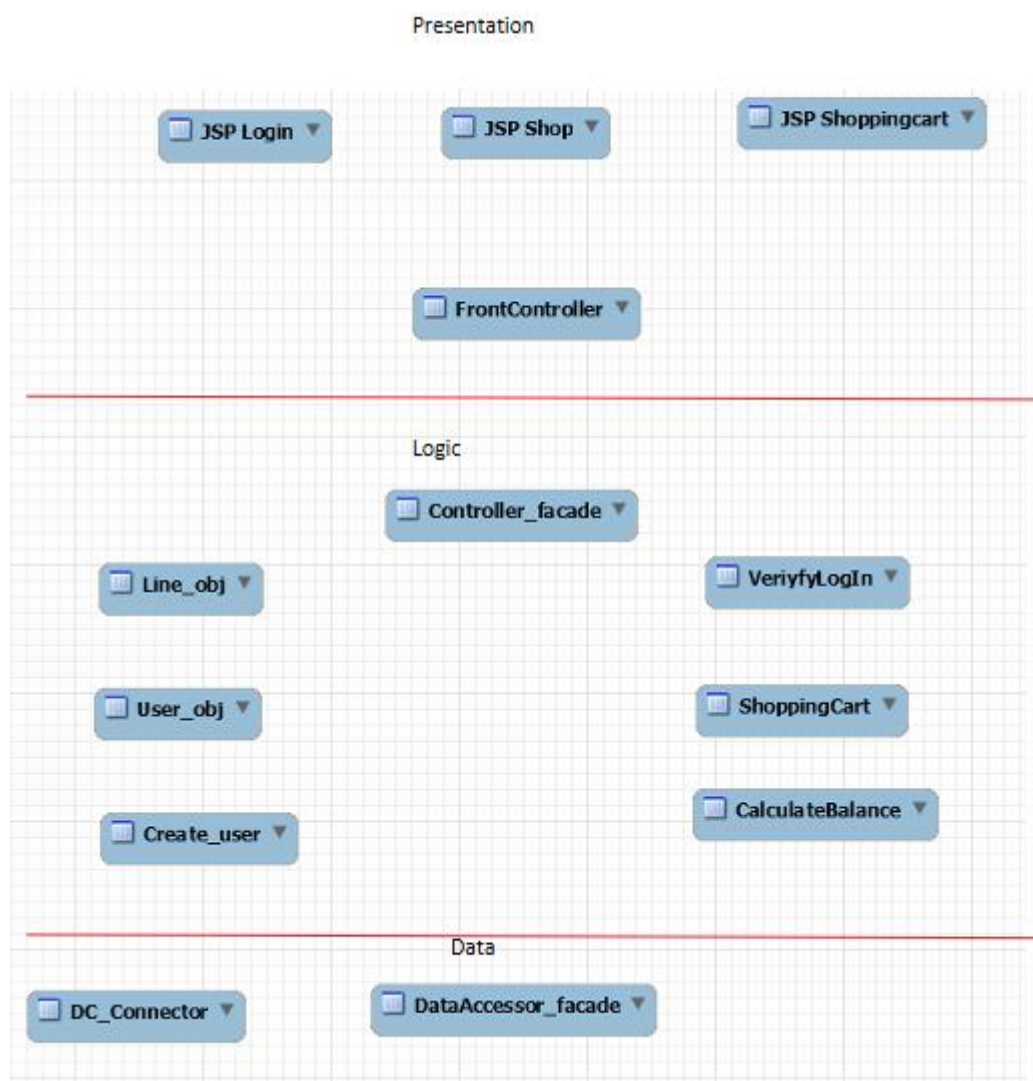
Som programmeringssprog til at kode logikken bag vores system og til forsøgsmæssigt at hente data fra vores database, gør vi brug af Java. I forhold til at skabe selve designet til vores system bruger vi HTML (Hyper Tekst Markup Language) og CSS (Cascading Style Sheet). Vores HTML- og CSS-kode laver vi i flere JSP-dokument, så der er mulighed for at inkludere Java-kode. Programmet vi bruger til at skrive vores Java-HTML- og CSS-kode er Netbeans IDE 8.2.

Til at gemme systemets data gør vi brug af databaseserveren MySQL og benytter SQL som databasesprog. Programmet hvori vi gemmer og kalder vores data er Workbench 8.0 CE. Til at forbinde vores Java-kode med vores database bruger vi (JDBC Java Database Connectivity) som softwaregrænseoverflade.

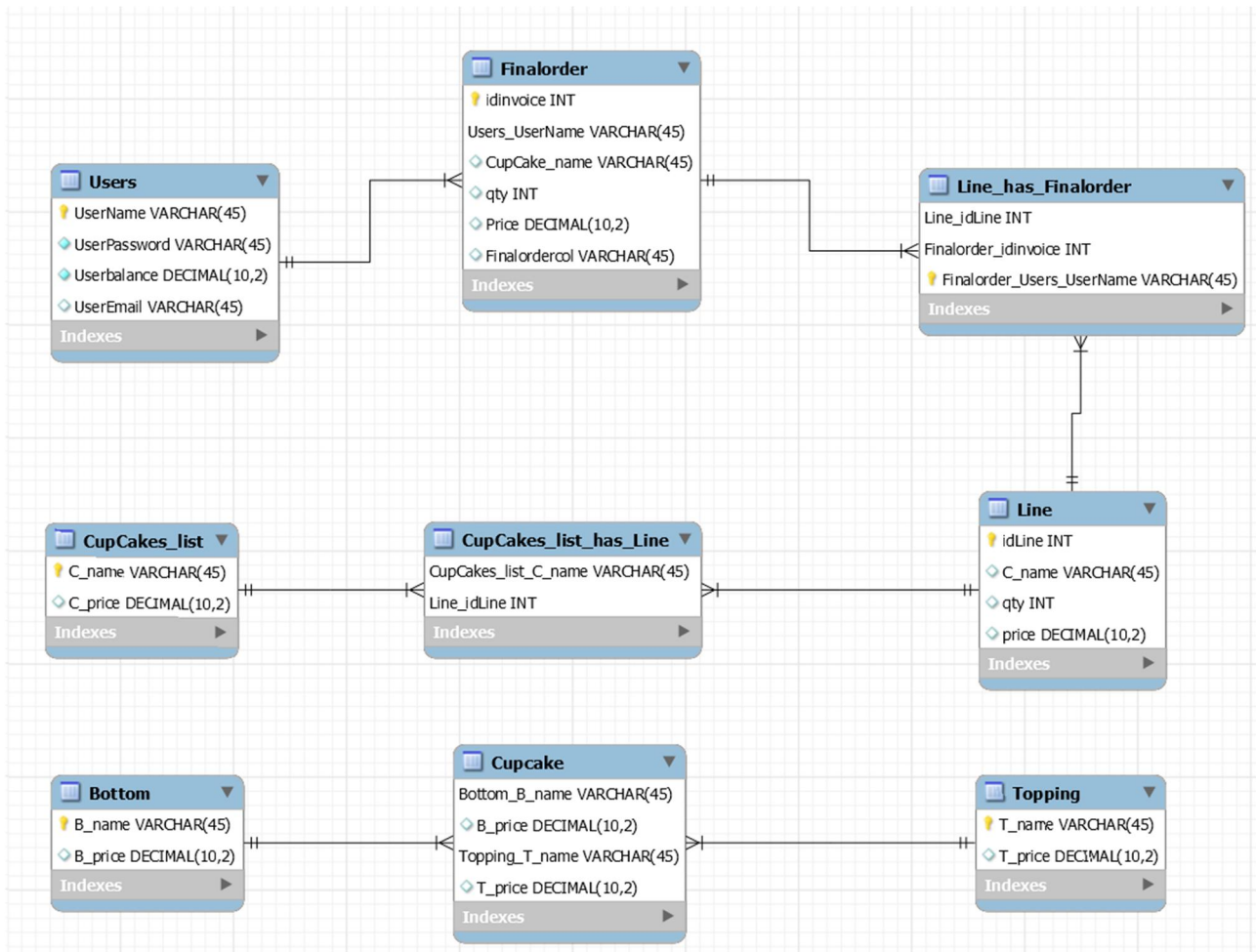
Navn:	CPH-mailadresse	Githubnavn:	Klasse:
Christian Ambjørn Kehr:	cph-ck209@cphbusiness.dk	Christian-A-Kehr	B
Mads-Ulrik Korsgaard Hansen:	cph-mh752@cphbusiness.dk	MukHansen	B
Claus Mikkelsen Findinge:	cph-cf124@cphbusiness.dk	ClausFindinge	B

DIAGRAMMER OG MODELLER

DOMÆNEMODEL OG ER-DIAGRAM



Navn:	CPH-mailadresse	Githubnavn:	Klasse:
Christian Ambjørn Kehr:	cph-ck209@cphbusiness.dk	Christian-A-Kehr	B
Mads-Ulrik Korsgaard Hansen:	cph-mh752@cphbusiness.dk	MukHansen	B
Claus Mikkelsen Findinge:	cph-cf124@cphbusiness.dk	ClausFindinge	B



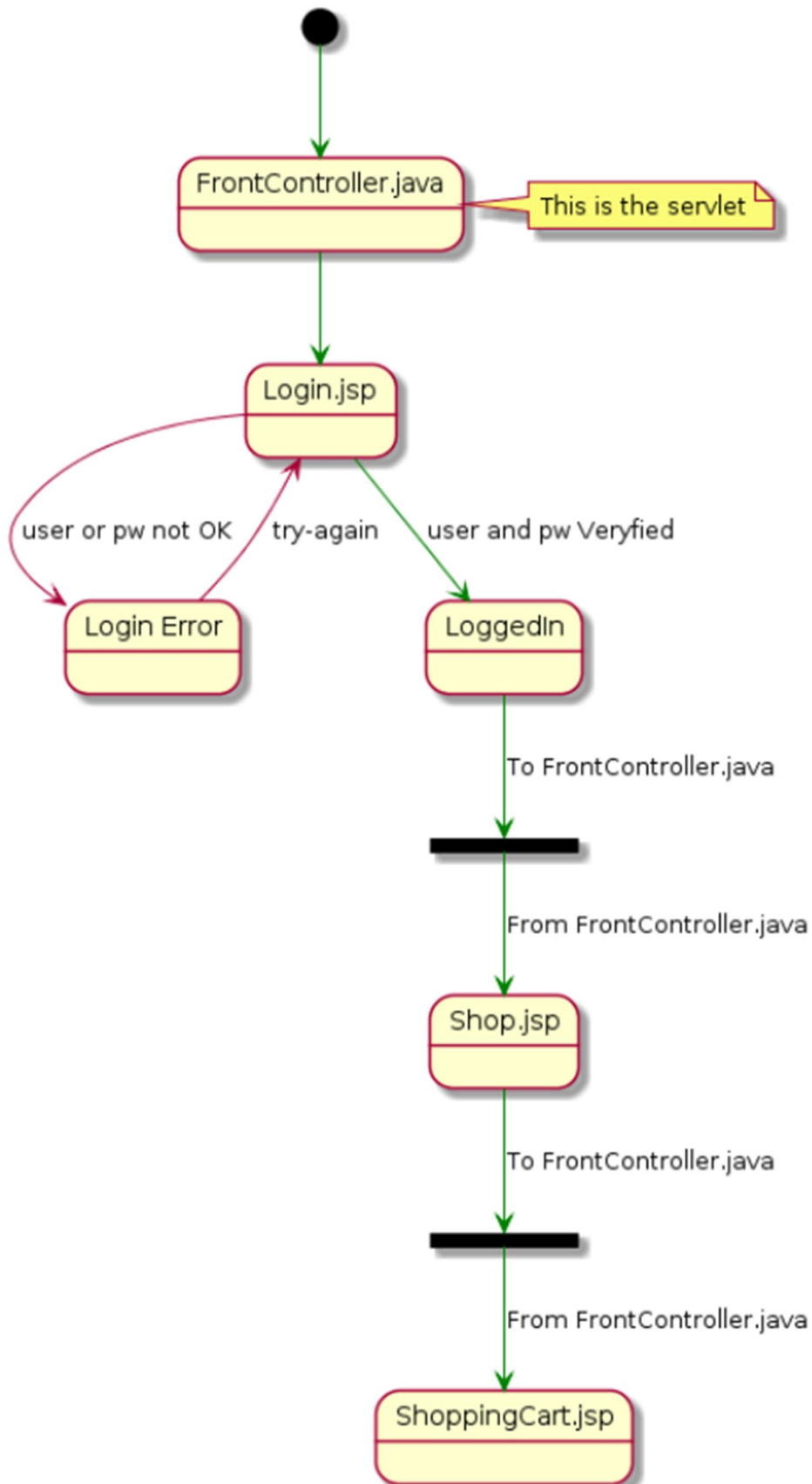
NAVIGATIONS DIAGRAM

Formålet med vores navigationsdiagram er at skabe en let og overskuelig illustration over, hvordan man navigerer og finder information på vores planlagte cupcake-hjemmeside. Når man åbner siden sendes en

Navn:	CPH-mailadresse	Githubnavn:	Klasse:
Christian Ambjørn Kehr:	cph-ck209@cphbusiness.dk	Christian-A-Kehr	B
Mads-Ulrik Korsgaard Hansen:	cph-mh752@cphbusiness.dk	MukHansen	B
Claus Mikkelsen Findinge:	cph-cf124@cphbusiness.dk	ClausFindinge	B

request til frontcontrolleren, der sørger for, at man kommer til login-siden. Her skal man indtaste et brugernavn og et password. Matcher det brugernavn og password med et tilsvarende, der befinder sig i databasen, sendes en request til frontcontrolleren, der sørger for at sende brugeren videre til shoppen. Matcher brugernavn og password til gengæld ikke, får man en loginerror, og det er ikke muligt at komme videre. Er man først nået siden med shoppen, er det muligt at komme videre til bestillingssiden (shoppingcart). Her sendes endnu engang et request til frontcontrolleren, der responderer ved at sende brugeren videre til bestillingen.

CupCake Navigations Diagram



Navn:

CPH-mailadresse

Githubnavn:

Klasse:

Christian Ambjørn Kehr:

cph-ck209@cphbusiness.dk

Christian-A-Kehr

B

Mads-Ulrik Korsgaard Hansen:

cph-mh752@cphbusiness.dk

MukHansen

B

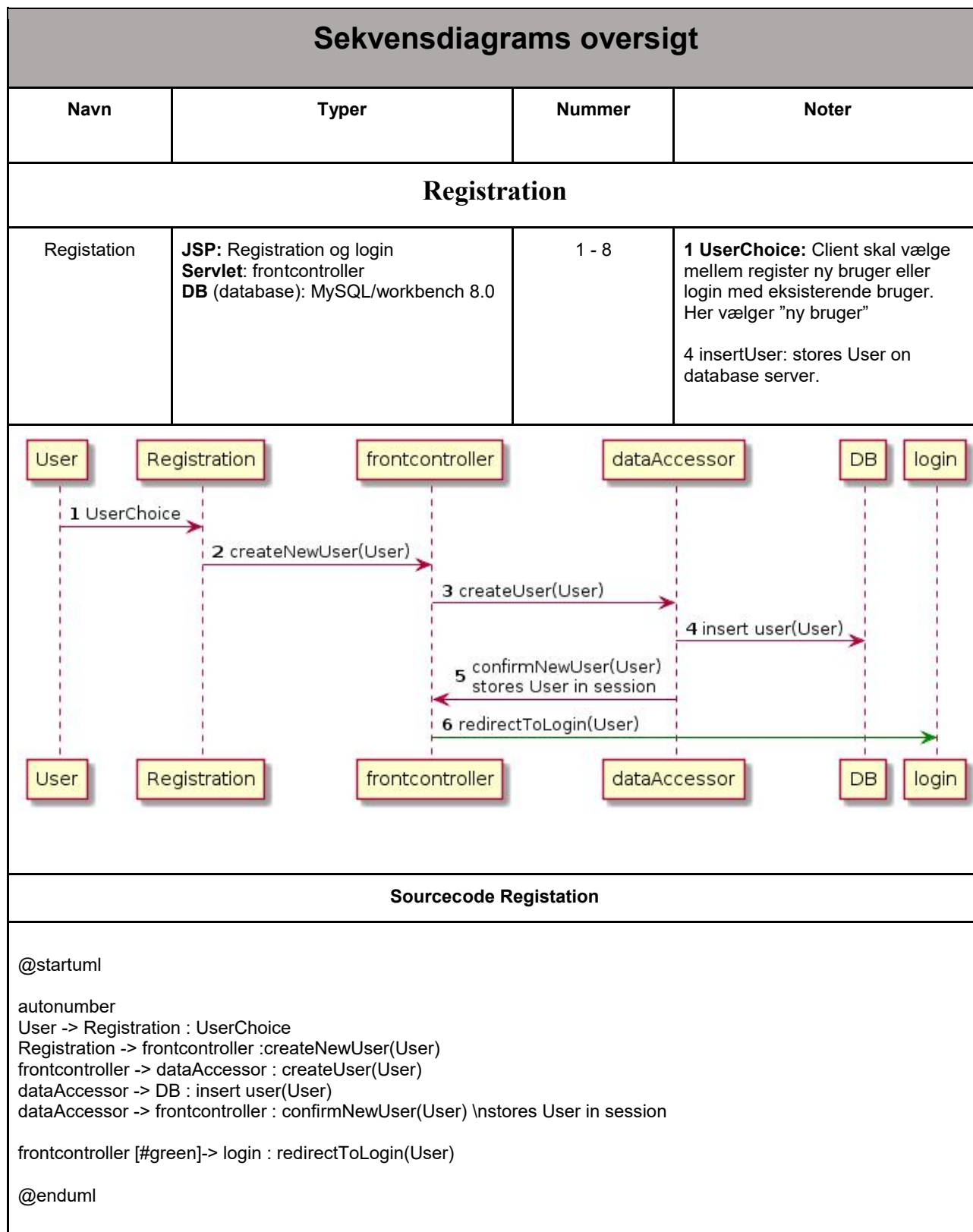
Claus Mikkelsen Findinge:

cph-cf124@cphbusiness.dk

ClausFindinge

B

SEKVENSDIAGRAM



Navn:

CPH-mailadresse

Githubnavn:

Klasse:

Christian Ambjørn Kehr:

cph-ck209@cphbusiness.dk

Christian-A-Kehr

B

Mads-Ulrik Korsgaard Hansen:

cph-mh752@cphbusiness.dk

MukHansen

B

Claus Mikkelsen Findinge:

cph-cf124@cphbusiness.dk

ClausFindinge

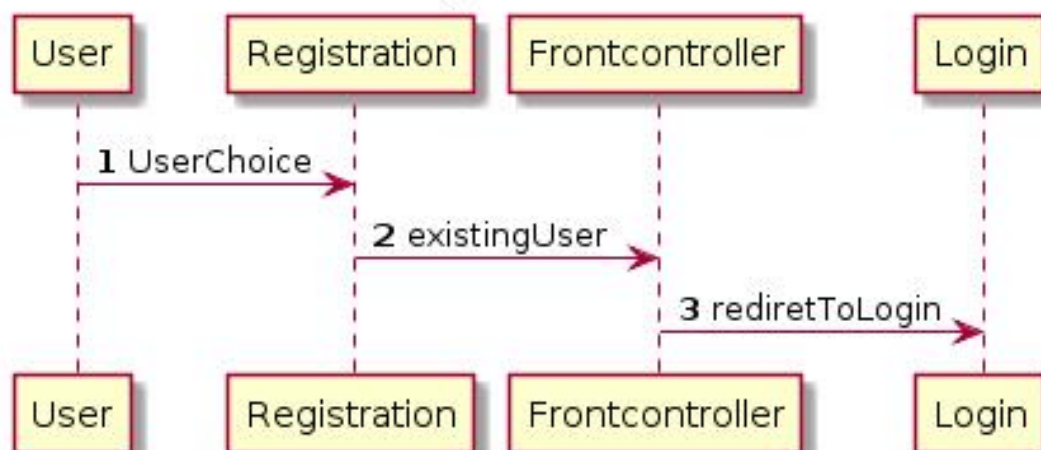
B

Sekvensdiagrams oversigt

Navn	Typer	Nummer	Noter
Login			
Login	JSP = login & shop Servlet : frontcontroller DB (database): MySQL/workbench 8.0	1 - 11	<p>Afsnit opdeles i to dele for overskuelighedens skyld.</p> <p>model 1: Clients starts side frem til login. Model 2: Login proces.</p> <p>Model 1: 1 UserChoice: Client skal vælge mellem register ny bruger eller login med eksisterende bruger. Her vælges "eksisterende bruger".</p> <p>Model 2: if statement i verify (3/6): sammenligner user Input med DB User og returnere boolean.</p>

Login part 1 (Registration - Login)

Login Model 1



Navn:

CPH-mailadresse

Githubnavn:

Klasse:

Christian Ambjørn Kehr:

cph-ck209@cphbusiness.dk

Christian-A-Kehr

B

Mads-Ulrik Korsgaard Hansen:

cph-mh752@cphbusiness.dk

MukHansen

B

Claus Mikkelsen Findinge:

cph-cf124@cphbusiness.dk

ClausFindinge

B

Sekvensdiagrams oversigt

Navn	Typer	Nummer	Noter
Login part 2 login - shop			
<pre> sequenceDiagram participant User participant Login participant Frontcontroller participant DBAccessor participant shop User->>Login: redireted via model 1 Login->>User: 1 requestUserLogin() User->>Login: 2 enterLoginInfo(name,password) Login->>Frontcontroller: 3 verifyUser (name,password) Frontcontroller->>DBAccessor: 4 getUser(name,password) DBAccessor->>Frontcontroller: 5 response Frontcontroller->>Login: 6 verifyFalse (false) Frontcontroller->>shop: 7 Shop(User) stores User in session and go to shop (true) </pre>			
Source code Login			
Model 1: @startuml title Login Model 1 autonumber User -> Registration : UserChoice Registration -> Frontcontroller : existingUser Frontcontroller -> Login : rediretToLogin @enduml		Model 2: @startuml title Login Model 2 User -> Login : redireted via model 1 autonumber Login -> User : requestUserLogin() User -> Login : enterLoginInfo(name,password) Login -> Frontcontroller : verifyUser (name,password) Frontcontroller -> DBAccessor : getUser(name,password) DBAccessor -> Frontcontroller : response Frontcontroller ->x Login : verifyFalse (false) Frontcontroller [#green]-> shop : Shop(User) stores User in session and go to shop (true) @enduml	

Navn:

CPH-mailadresse

Githubnavn:

Klasse:

Christian Ambjørn Kehr:

cph-ck209@cphbusiness.dk

Christian-A-Kehr

B

Mads-Ulrik Korsgaard Hansen:

cph-mh752@cphbusiness.dk

MukHansen

B

Claus Mikkelsen Findinge:

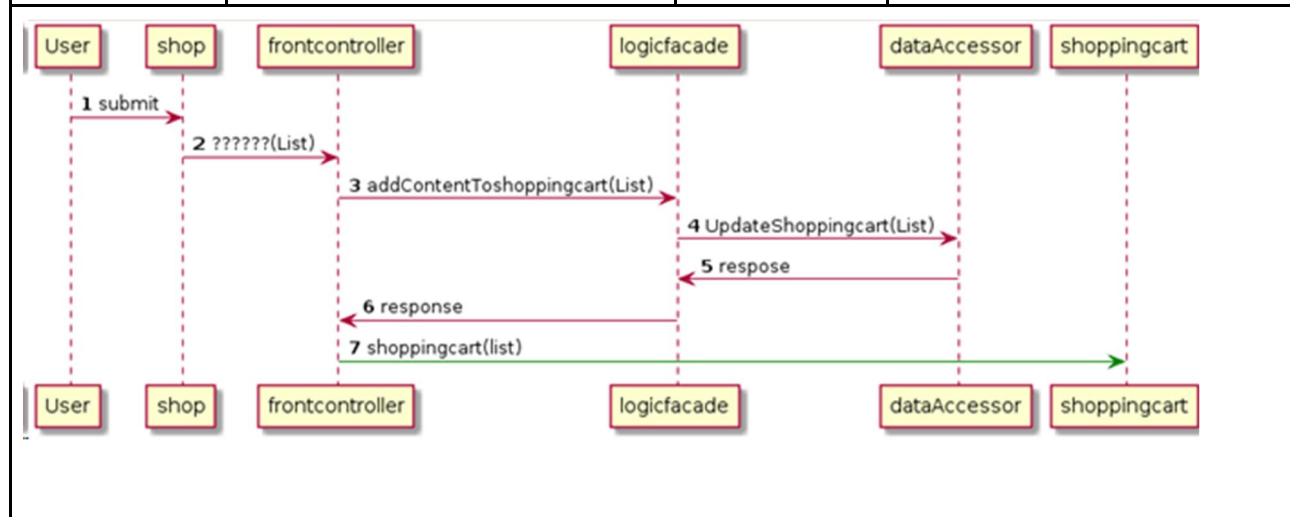
cph-cf124@cphbusiness.dk

ClausFindinge

B

Sekvensdiagrams oversigt

Navn	Typer	Nummer	Noter
Shop (old version)			
Shop	JSP = shop, shoppingcart Servlet : frontcontroller DB (database): MySQL/workbench 8.0	1-7	



Source code shop

@startuml

autonumber

User -> shop : submit

shop -> frontcontroller : ??????(List)

frontcontroller -> logicfacade : addContentToshoppingcart(List)

logicfacade -> dataAccessor : UpdateShoppingcart(List)

dataAccessor -> logicfacade : response

logicfacade -> frontcontroller : response

frontcontroller [#green]-> shoppingcart : shoppingcart(list)

@enduml

Navn:	CPH-mailadresse	Githubnavn:	Klasse:
Christian Ambjørn Kehr:	cph-ck209@cphbusiness.dk	Christian-A-Kehr	B
Mads-Ulrik Korsgaard Hansen:	cph-mh752@cphbusiness.dk	MukHansen	B
Claus Mikkelsen Findinge:	cph-cf124@cphbusiness.dk	ClausFindinge	B

STATUS PÅ IMPLEMENTATION

UTILSIGTEDE MANGLER

Det er desværre kun lykkedes os at få hul igennem til databasen fra vores shop, og derfor har vi kun i ringe grad fået systemet til at virke. Dette skyldes først og fremmest, at vi igennem hele processen har været tyngt af en basal manglende viden angående, hvordan vi skulle strukturere opbygningen af vores system. Vi har modtaget meget divergerende oplysninger til vores reviews (og når vi har søgt hjælp). Eksempelvis har vi i over en uge antaget, at LogikFacaden ikke var relevant for vores projekt, da vi har fået det fortalt til et review, for så at få det modsatte af vide til et efterfølgende review.

Vi har m.a.o. været i vildrede og ikke haft forudsætningerne for at vide, hvordan opgaven skulle håndteres og løses. Derfor er resultatet også blevet derefter.

Gruppen er opmærksom på at elev forudsætning fra det tidligere semester har ændret sig, derfor har vi som elever ikke haft den fornødne HTML-viden. Dette suppleres med en primær underviser, som ikke er forberedt, er ukoordineret, ikke åbner for læsestof før søndag aften (i forbindelse med frontend) og ikke tager styring ved review (Det kan umuligt være eleverne som skal vide hvad de ikke ved. Det må derimod være underviseren, som skal stille de ledende spørgsmål for at gøre elever opmærksom på evt. misforståelser og fejl)

Det skal her tilføjes, at dette selvfølgelig ikke er møntet på alle undervisere, men hele processen har grundet ovenstående været påvirket af frustration og forvirring.

BEVIDSTE VALG OG FRAVALG

Ønsket var helt fra start at lave et samlet menukort, hvor alle kombinationsmuligheder af toppings og bottoms er listet på forhånd. Dette i stedet for to separate lister, hvor kunden selv kunne skabe sin kombination af topping og bottom. Dette grundet overskueligheden og brugervenligheden.

I tilgift har vi lavet et Quantity-felt, hvor kunden markerer antallet af den enkelte cupcake, personen ønsker. Hermed behøver kunden ikke at trykke flere gange på den enkelte cupcake for at bestille flere af denne type.

Navn:	CPH-mailadresse	Githubnavn:	Klasse:
Christian Ambjørn Kehr:	cph-ck209@cphbusiness.dk	Christian-A-Kehr	B
Mads-Ulrik Korsgaard Hansen:	cph-mh752@cphbusiness.dk	MukHansen	B
Claus Mikkelsen Findinge:	cph-cf124@cphbusiness.dk	ClausFindinge	B

TESTS

I forhold til at sikre, at vores java-kode kunne fungere optimalt og ikke mindst for at kunne opspore eventuelle fejl, ville det være væsentligt at kunne foretage tests mellem systemets forskellige lag (præsentation-lag, logik-lag og data-lag). Vi ville således oprette tests, der afprøver hvorvidt vores DBConnector har forbindelse til vores database. Ligeledes ville vi teste, hvorvidt vores DBAccessor kalder vores DBConnector og hvorvidt det var muligt at hente data fra databasen gennem vores DBAccessors metoder.

Herefter ville vi afprøve forbindelsen mellem data-lag og logik-lag ved at teste, hvorvidt LogikFacaden har forbindelse til DBAccessoren, hvorpå vi testede forbindelsen mellem FrontControlleren i præsentationslaget og LogikFacaden. Dertil ville vi teste forbindelsen mellem FrontController og JSP-sider, hvor vores HTML-kode befinder sig.

Ved at lave disse tests og afprøve de forskellige lag og klassers forbindelse til hinanden ville det være lettere at opspore, hvor en evt. fejl i koden, der bremser systemets dynamik, befinder sig. Hermed bliver det også lettere at placere et ansvar i forhold til personen, der skal vedligeholde koden. Dette kunne eksempelvis være, hvis det brugernavn eller password vi indtaster i login-siden, ikke skabte forbindelse til shoppingcart-siden.