

Cupcake

Victor Kjær Lauritsen - cph-vl48@cphbusiness.dk - vl48 / Vics

Martin David Pilo Døi - cph-md273@cphbusiness.dk - fungler

Casper Kruse Olesen - cph-co130@cphbusiness.dk - casperolesen

Annika Bhagya Ehlers - cph-ae112@cphbusiness.dk - marshmallouws

Afleveringsdato: 17/03 2019

Indledning

Cupcakes er en webshop, som er udviklet til en kunde. Ønsket har været at have en hjemmeside, hvor kunder kan gå ind og sammensætte deres egne cupcakes ved at vælge en top og en bund. For at benytte webshoppen, skal kunden være oprettet som bruger samt logget på. Brugeren har en saldo i systemet, som denne kan benytte til at købe cupcakes. Såfremt brugeren ønsker flere penge på saldoen, skal brugeren kontakte en administrator, som kan lægge penge ind. Administratoren har ydermere til formål at se hvilke ordrer, der er lavet, som en bager så kan lave i butikken.

Webshoppen består af en MySQL database, som indeholder information om brugere, cupcakes (hvilket er delt op i top og bund) samt ordre. Forretningslogikken er udviklet i Java, som også bliver brugt til at hente og skrive til databasen. Til frontend delen har vi benyttet HTML, JSP og JavaScript for at gøre hjemmesiden dynamisk.

Krav

- Kunden skal kunne oprette en bruger og logge ind.
- Kunden skal kunne sammensætte en cupcake ud fra et valg af top og bund samt vælge et ønsket antal.
- Kunden skal kunne bestille flere forskellige slags cupcakes i én ordre ved at have en indkøbskurv.
- Kunden skal kunne se sine gamle ordre samt detaljer om denne.
- Administratoren skal kunne logge ind.
- Administratoren skal kunne se en liste over alle ordre og sortere denne ud fra navn eller dato.
- Administratoren skal kunne tilføje penge til en brugers saldo.

Specificering af teknologier

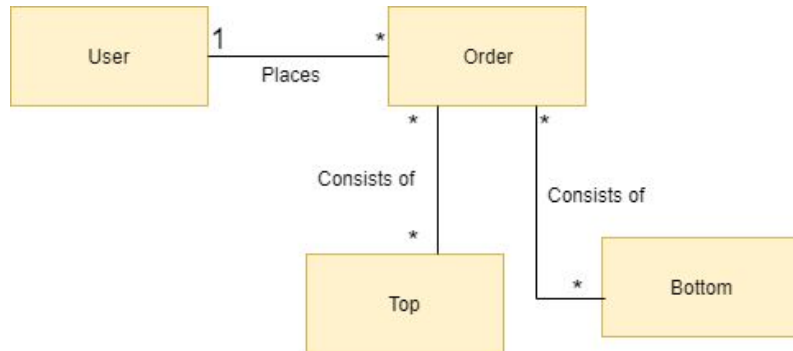
- MySQL 8.0.15
- Java 1.8.0_201
- Apache Tomcat 8.0.27
- Servlet 3.1
- JSP Version: 2.3
- HTML
- CSS
- JavaScript

Platforme:

- NetBeans 8.2
- MySQL Workbench 8.0 CE

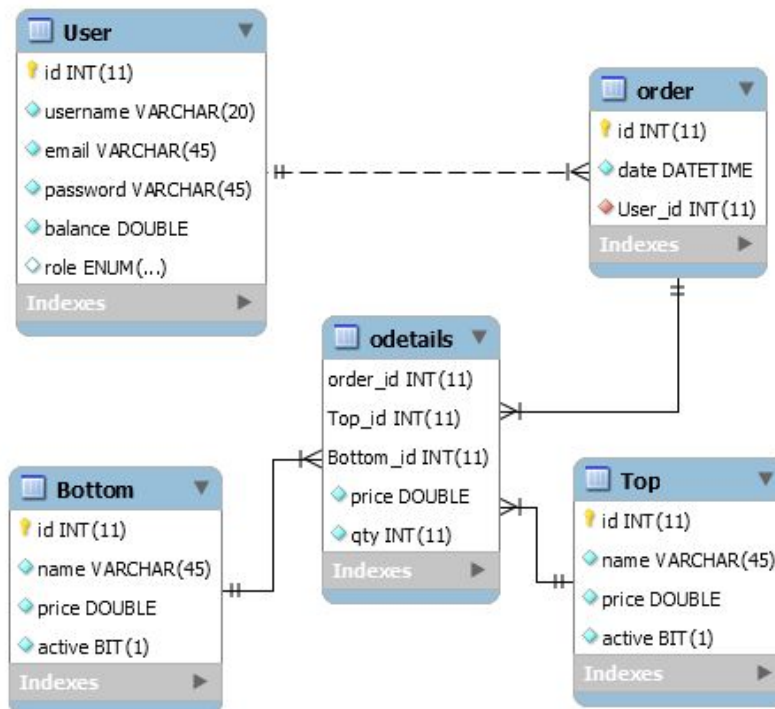
Domænemodel og ER-diagram

Domænemodel



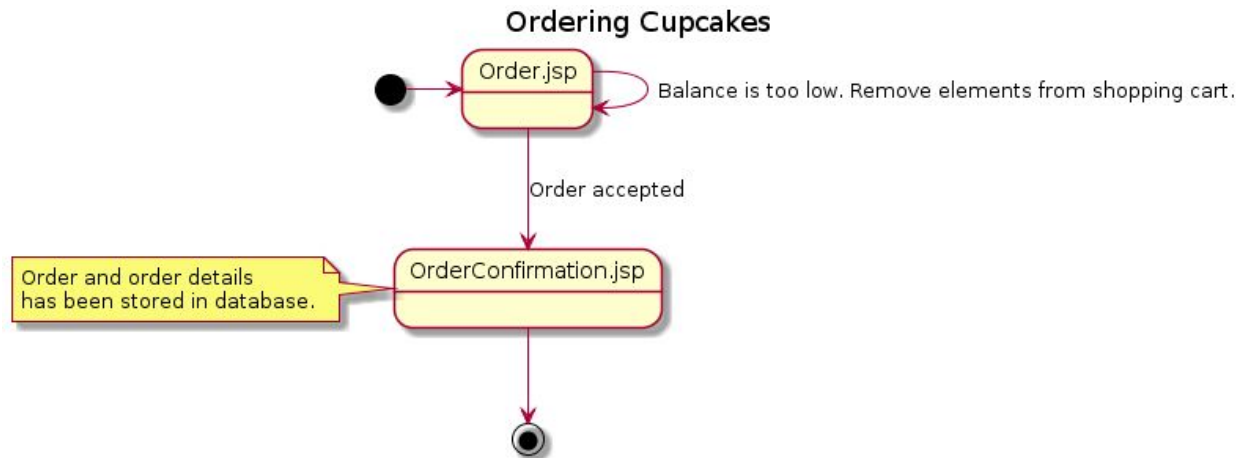
ER-diagram

ER-diagrammet er en visualisering af den færdige database. I tabellen *odetails* vises der ikke nøgler, men primærnøglen er en composit key som består af *order_id*, *Top_id* og *Bottom_id*.



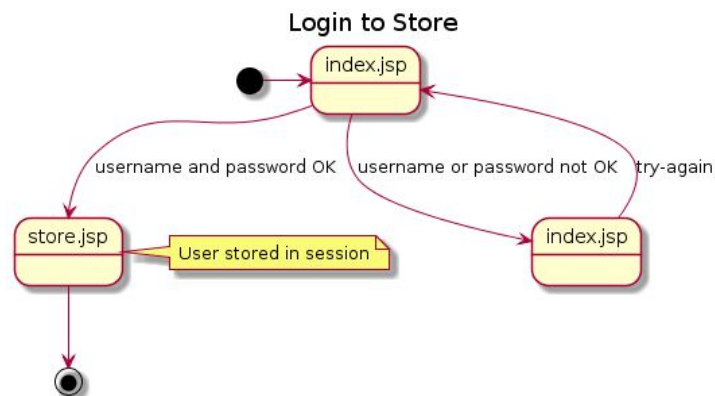
Navigationsdiagram

Navigationsdiagrammet viser en bruger, der ønsker at købe cupcakes. Såfremt brugeren ikke har penge nok på saldoen, vil Order.jsp vises med en fejlmeddelelse. Ellers bliver ordren godkendt, og brugeren bliver navigeret til OrderConfirmation.



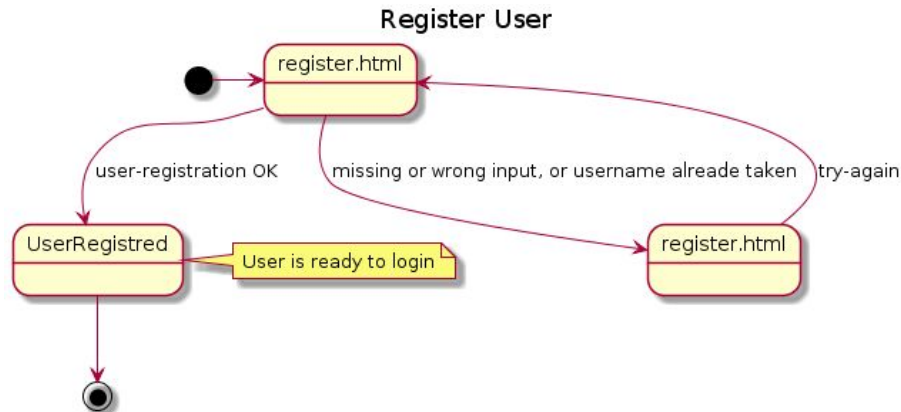
Login

Hvis indtastede brugernavn og password ikke matcher med en bruger i databasen, vil brugeren få vist en fejl på siden, og kan herefter forsøge at logge ind igen.



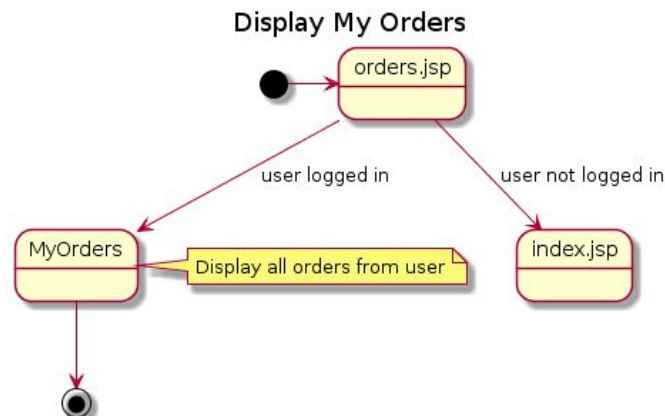
Register ny bruger

For at oprette en bestilling, kræves en registreret bruger. Der vælges et brugernavn samt password, som ved succesfuld oprettelse straks kan bruges til at logge ind med. Hvis det ønskede brugernavn er optaget, skal der vælges et andet.



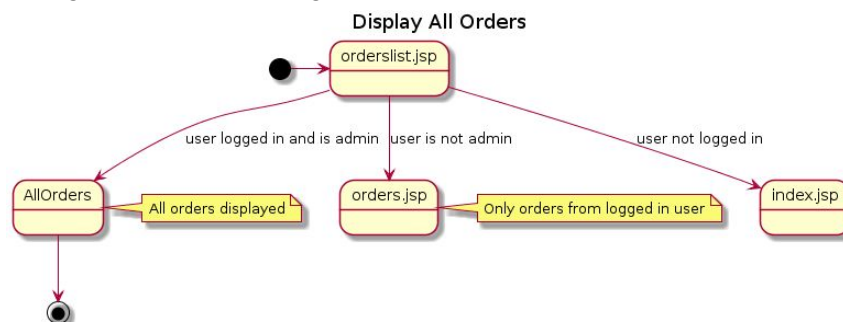
Vis brugerens ordrer

Det er muligt at se en oversigt over tidligere ordrer, blot man er logget ind. Tilgår man siden uden at være logget på, får man vist siden hvor dette kan gøres.



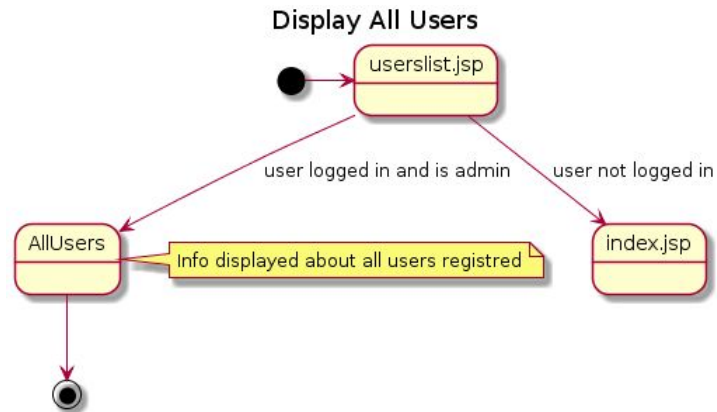
Vis alle ordrer

Hvis du har en bruger med 'admin' adgang, kan du se en side med alle kunders ordrer. Her kan en specifik ordre søges frem via et brugernavn.



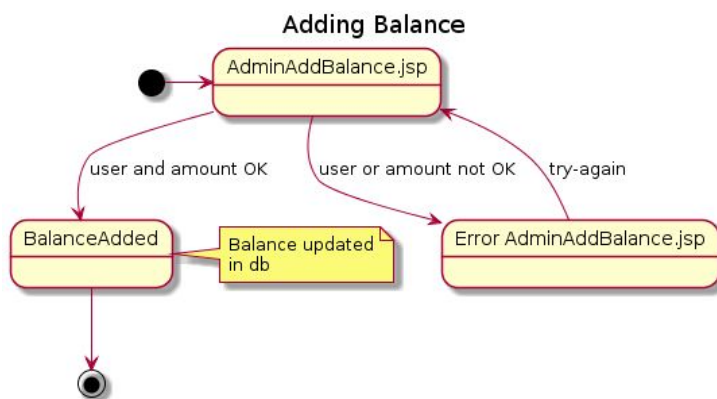
Vis alle registrerede brugere

En bruger indtaster nogle informationer ved oprettelse, som navn og email. En tabel med alle registrerede bruger og deres oplysninger kan tilgås med admin adgang via menuen.



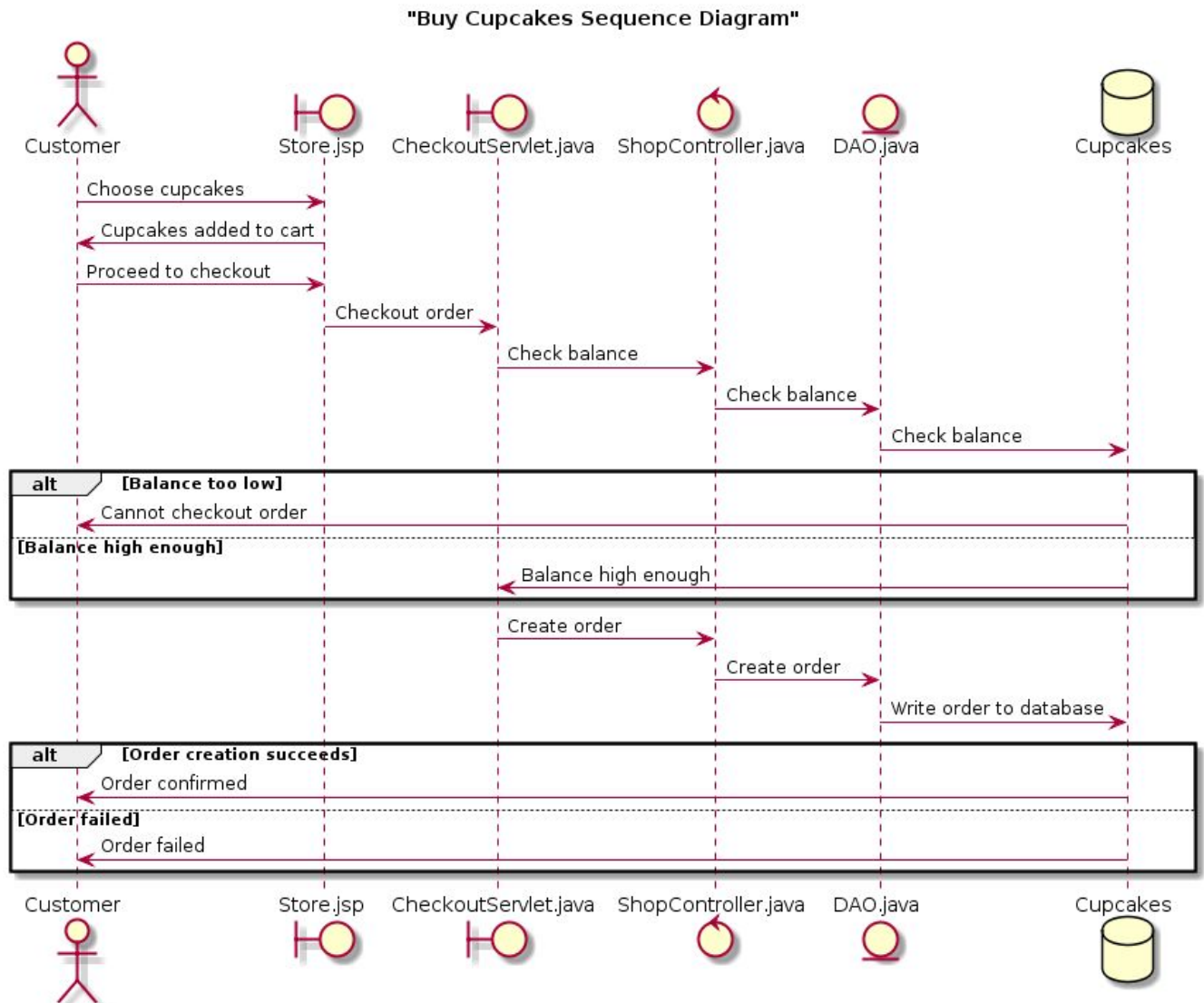
Tilføj balance til bruger konto

Fra tabellen med bruger-information, kan der for hver bruger vælges at indsætte et beløb til dennes konto. Hvis der vælges en bruger som ikke findes eller beløbet f.eks. Er negativt, får man besked på at prøve igen. Det er kun muligt for en admin at tilføje penge til en brugers konto.



Sekvensdiagram

Sekvensdiagrammet viser en brugers interaktion med systemet, når denne skal bestille cupcakes.



Særlige forhold

Session:

Indkøbskurven bliver lagt i session, ikke i databasen, under "cart" som en arraylist af objektet Odetails. Session bliver yderligere brugt til at holde styr på hvilken user der er logget ind, via "user" som objektet User.

Validering:

Der tjekkes om input er null eller empty, samt om user balance er høj nok til at foretage køb. Hvis der foretages string input hvor int forventes, bliver fejlen caught når der skal parses, så der ikke skabes problemer.

Status på implementation

Mangler:

Store.jsp

Det er i øjeblikket muligt at komme igennem til shoppen og forsøge at placere en ordre uden at være logget ind. Her skal store.jsp tjekke om man er logget ind.

Login / Register

Det kunne foreslås at forbedre login med mere validering, samt at validere register input.

Her burde man validere at email er valid og at username samt password har et bestemt antal tegn.

Register.html burde også laves om til en .jsp, så den også kan håndtere fejlmeddelelser.

Tilføjelse af balance til user

Hvis useren er logget ind, skal de i øjeblikket logge ind på ny for at se sin opdaterede balance.