

Cupcake webshop

Deltagere:

Mark Dyrby Denner - cph-md254@cphbusiness.dk -
<https://github.com/froscoldt>

Mark Jespersen - cph-mj679@cphbusiness.dk -
<https://github.com/Markjesp>

Mads Wulff Nielsen - cph-mn492@cphbusiness.dk -
<https://github.com/Wulffn>

Christian Falk Moustesgård - cph-cm267@cphbusiness.dk -
<https://github.com/jenzuffer>

Klasse A

Tidspunkt: Oktober 2018

Indledning :

Projektet omhandler en cupcake webshop som skal kunne bruges af virksomheder så kunder kan købe produkter igennem webshoppen. Webshoppen består af en frontend og en backend og en database. Hvor at frontenden præsenterer vores webshop, i form af JSP filer, hvor at JSP filerne bliver håndteret af vores backend, som snakker sammen med en database. Frontenden er bare et template for hvad brugeren skal se. Der vil løbende komme nogle diagrammer som fremviser hvordan det hele hænger sammen.

Baggrund:

Webshoppen er beregnet til virksomheder og ikke selve slutbrugeren, dermed er den i B2B imens virksomhederne som anvender webshoppen kan være B2C eller B2B afhængig af om de vil sælge cupcakes til det ene eller andet eller begge og i hvilken mængde. Virksomhederne som skal anvende webshoppen har brug for at deres kunder kan sætte en cupcake sammen også bestille den.

Teknologi valg:

Mysql udgave 5.1.471, Netbeans 8.2 javaee with Apache Tomcat 8.0.27[1], Maven project med .JSP filer, HTML filer og Java dependencies JDK 1.8, mysql workbench[2].

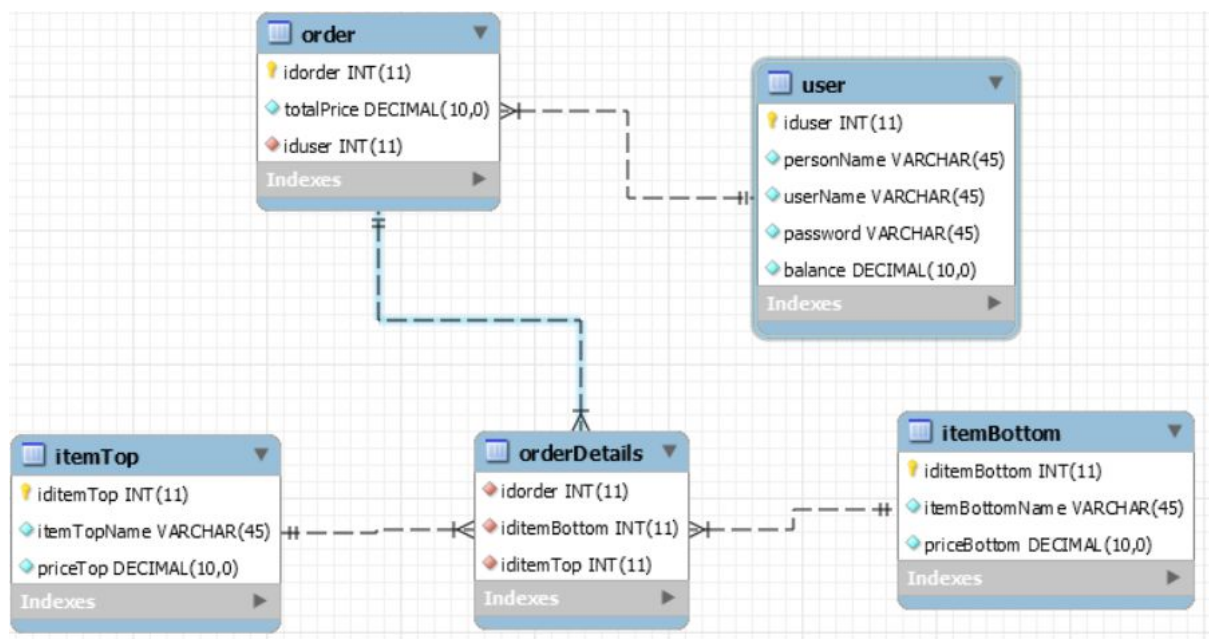
t

[1] <https://netbeans.org/downloads/?pagelang=EN>

[2] <https://www.mysql.com/products/workbench/>

ER Diagram:

Dette ER Diagram generet med MYSQL Workbench viser Opsætningen af Databasen:



Applikationens E/R diagram er sat i tredje normalform og indeholder fremmednøgler for at holde styr på databasestrukturen. En orderDetail sættes sammen, når en order bliver kreeret med en fremmednøgle til den unikke bruger. Itembottom og itemtop sættes sammen på baggrund af brugerens køb og bliver sat sammen i orderdetails på kundens order ID.

Navigationsdiagram

<https://www.planttext.com/>

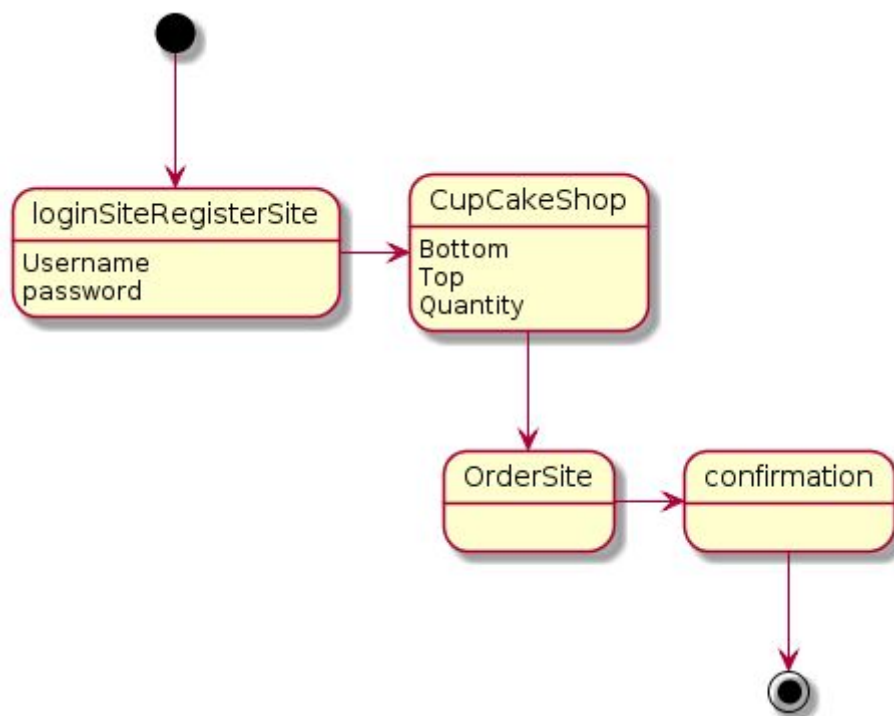
@startuml

[*] --> loginSiteRegisterSite

```

loginSiteRegisterSite : Username
loginSiteRegisterSite : password
loginSiteRegisterSite -> CupCakeShop
CupCakeShop --> OrderSite
CupCakeShop : Bottom
CupCakeShop : Top
CupCakeShop : Quantity
OrderSite -> confirmation
confirmation -->[*]
@enduml

```



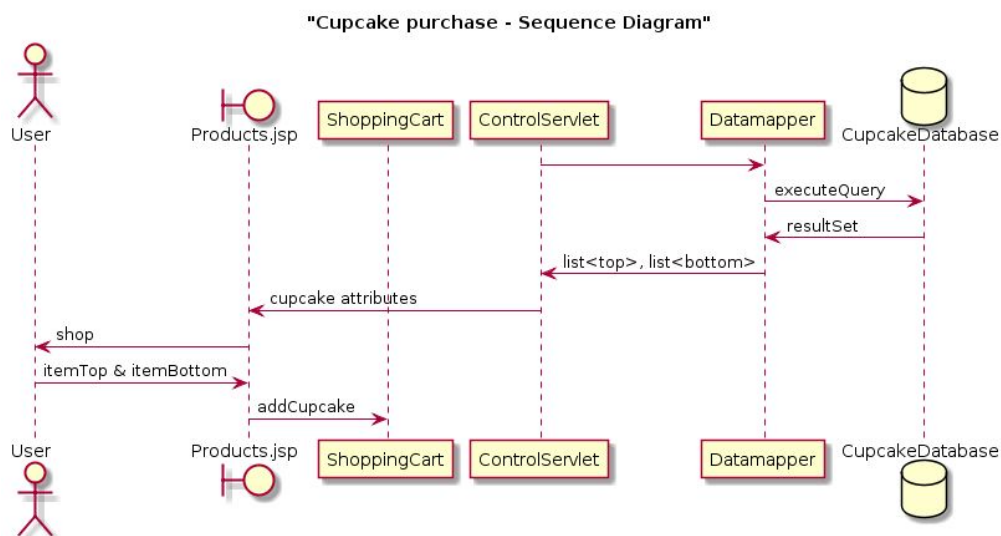
Applikationen starter med at bruger logger ind eller at registrere sig hvis brugeren ikke allerede har en konto. Uden en aktiv logged in i Session kan man ikke bevæge sig videre på shoppen. Vores Control servlet redirecter klienten over til login.jsp hvorfra af man kan gå videre til registration på samme side eller igennem et hidden parameter kan blive videresendt til products.jsp hvilket der er vores CupCakeShop. I products.jsp kan en klient sammensætte en cupcake ud fra de muligheder som blev sat sammen i databasen. Når klienten har sammensat en cupcake og antallet og klikker addtocart bliver de pt redirectet til login siden. Brugerens shoppingcart gemmes i sessionsobjektet og indeholder cupcakes

samt kvantitet sådan at når man logger ind igen er cupcaken opbevaret og man kan tilføje flere forskellige variationer med forskellige mængder på denne i products.jsp.

Products.jsp (cupcakeshop) mangler stadig en endegyldig knap til at redirecte klienten til "ordersite", derfor er der ikke en ordersite.jsp. Der mangler fortsat implementation af korrekt metode i Datamapperen for at opbevare klientens endegyldige ordre i databasen.

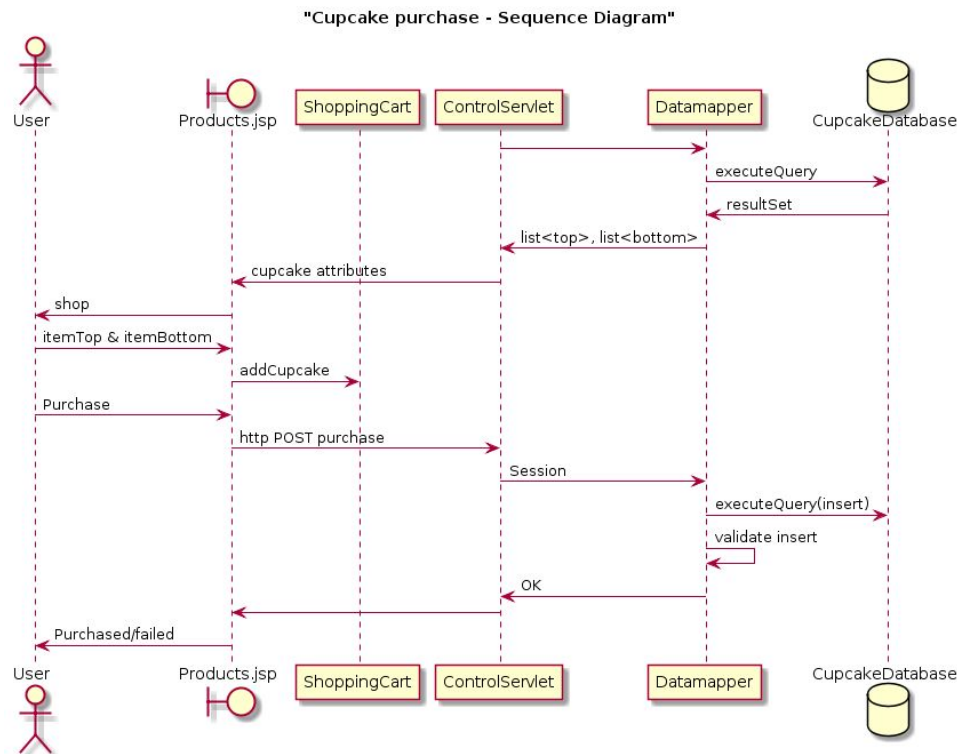
OrderSite.jsp er en side, hvor brugeren vil have mulighed for at se sine tidligere ordrer, denne er ikke implementeret i applikationen endnu.

Sekvensdiagram



Det nuværende sekvensdiagram indeholder, efter login til shoppen, et kald til databasen der returnerer to lister med topping og bottoms. Datamapperen har adgang til databasen og sender listerne til ControlServleten, der ved et forward til Products.jsp sætter listerne ved hjælp af request.setAttribute. Listerne hentes i Products.jsp og looper disse ud i to option lister. Bruger

en kan vælge hvilken bottom og hvilken top denne vil købe og tilføje kombinationen, som en hel Cupcake i et Lineltem til indkøbskurven, som er gemt i sessionsobjektet. Hvert Lineltem er loopet ud i Products.jsp således, at brugeren kan se hvilke Lineltems Shopping carten indeholder.



Det ønskede system ville se således ud, havde der været flere ressourcer til rådighed.

Særlige Forhold:

I sessionsobjektet opbevares applikationens shopping cart, hvor denne indeholder de forskellige cupcake variationer, priser og mængder som brugeren har sammensat. Hvad angår handling af exceptions er der på nuværende tidspunkt brugt forskellige fremgangsmåder i datamapperen. Nogle metoder indeholder en try & catch og andre bruger throws exception.

Hvad angår validering af brugerens credentials laves der test af brugeren før denne tilføjes som currentUser i sessionen. Der er brugt en metode som returnerer true eller false, alt efter brugerens oplysning sammenholdt med data fra databasen.

sekvensdiagram kode

@startuml

title "Cupcake purchase - Sequence Diagram"

actor User

boundary Products.jsp

participant ShoppingCart

participant ControlServlet

participant Datamapper

database CupcakeDatabase

ControlServlet -> Datamapper

Datamapper -> CupcakeDatabase : executeQuery

CupcakeDatabase -> Datamapper : resultSet

Datamapper -> ControlServlet : list<top>, list<bottom>

ControlServlet -> Products.jsp : cupcake attributes

Products.jsp -> User : shop

User -> Products.jsp : itemTop & itemBottom

Products.jsp -> ShoppingCart : addCupcake

User -> Products.jsp : Purchase

Products.jsp -> ControlServlet : http POST purchase

ControlServlet -> Datamapper : Session

Datamapper -> CupcakeDatabase : executeQuery(insert)

Datamapper -> Datamapper : validate insert

Datamapper -> ControlServlet : OK

ControlServlet -> Products.jsp

Products.jsp -> User : Purchased/failed

@enduml