



13-03-2019

Cupcake Shop



Klasse A

Dennis Hansen, cph-dh153@cphbusiness.dk, git: drh89

Niklas Brynfeldt, cph-nb167@cphbusiness.dk, git: nbryn

Indledning

Denne rapport omhandler en Maven webapplikation som er udarbejdet på 2. semester på cph-business Lyngby i år 2019. Rapporten omhandler projektet cupcakes.

Applikationens formål er at man som bruger skal kunne oprette en bruger, logge ind, og bestille / købe cupcakes.

Det er en simpel webshop hvor der er gjort brug af en SQL database, java, java-servlets og jsp-sider som backend, samt html og css som frontend.

Rapporten indeholder desuden et ER diagram, forskellige sekvensdiagrammer samt et navigationsdiagram over programmet.

Baggrund

En ny kunde der besøger websiden, skal kunne registrere sig selv ved at indtaste email, brugernavn og password. Derefter skal den nye kunde kunne logge sig selv ind med sin email og sit password. Når kunden er logget ind, skal kunden frit kunne sammensætte en cupcake ud fra udvalget af kagebunde og toppings, vælge det antal af den valgte cupcake som kunden ønsker, og derefter kunne tilføje det valgte antal cupcakes til sin indkøbskurv. På betalingssiden skal kunden kunne betale sine cupcakes, fjerne eventuelle uønskede cupcakes, samt kunne vælge at tilføje penge til sin konto. Kunden skal desuden være i stand til at tilføje flere cupcakes til sin indkøbskurv hvis det ønskes. Ydermere skal kundens ubetalte cupcakes i indkøbskurven gemmes, sådan at de stadig er der næste gang kunden logger ind.

Der skal være en admin side hvor man kan se alle ordre, samt dennes oplysninger.

Teknologivalg

Webapplikationen er udviklet i NetBeans IDE 8.2

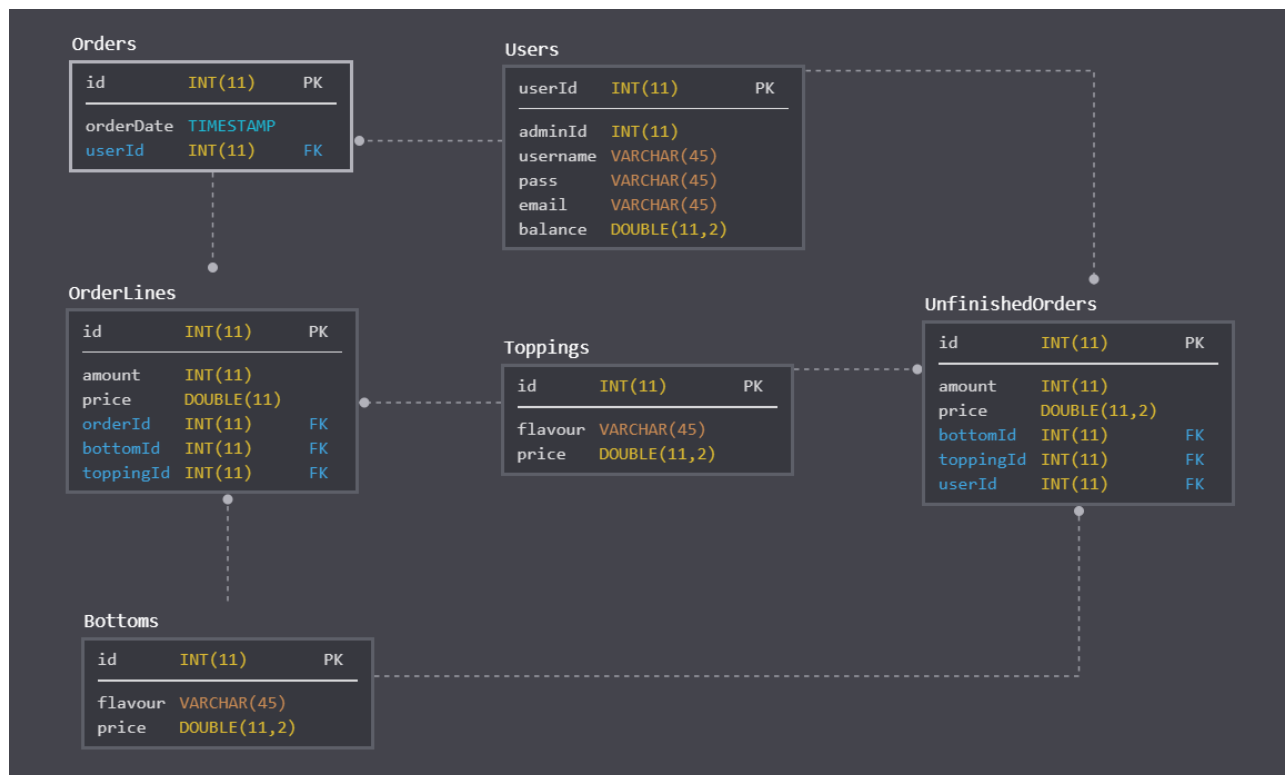
Data håndteres i MySql Workbench 8.0.0

Der er anvendt JDBC til at tilgå databasen

Frontend: HTML(CSS) og JSP.

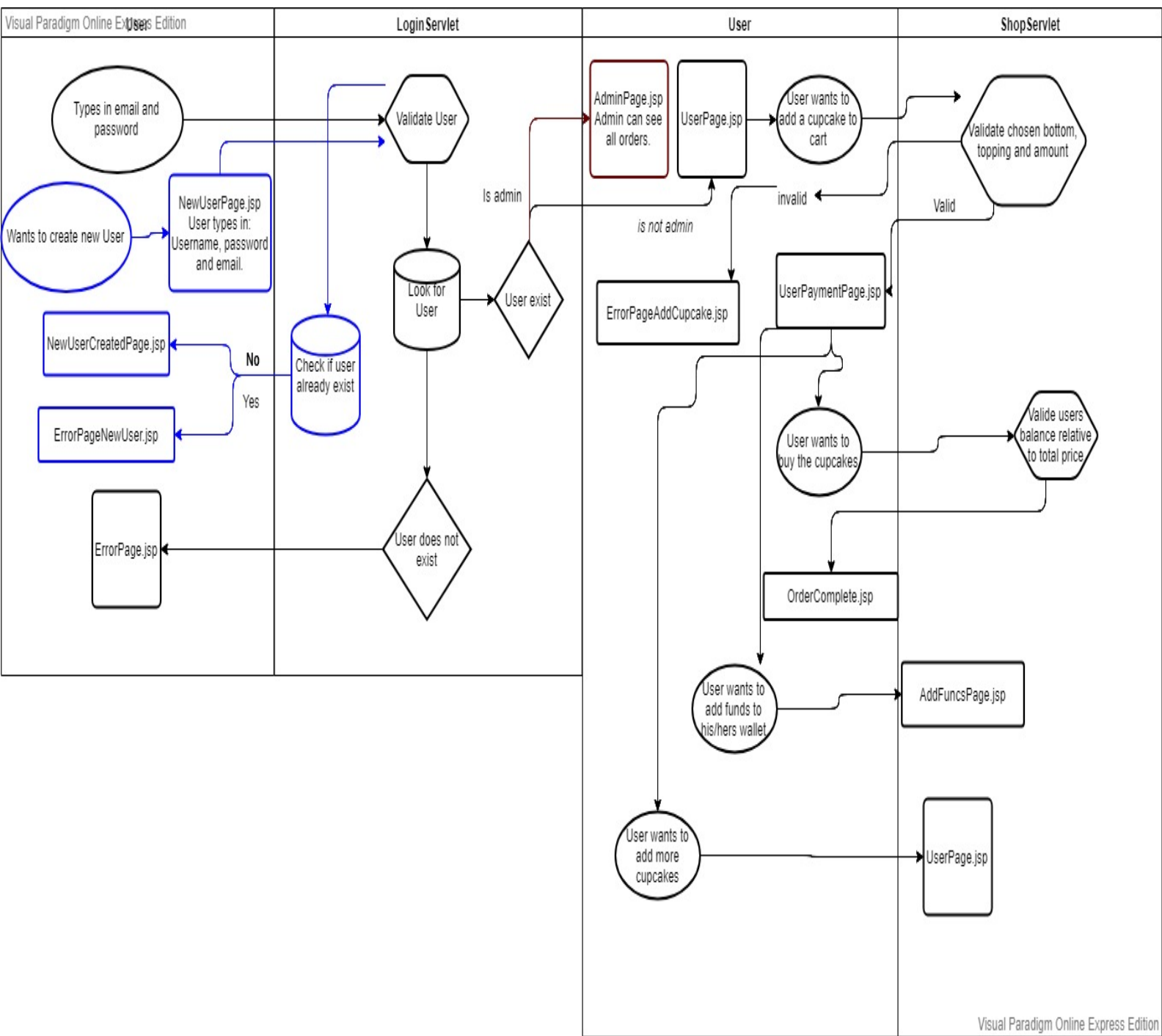
ER-diagram

Alle tabellerne i databasen er på 3. normalform. Desuden er relationerne, tabellerne imellem, alle 1 : mange. Alle tabellerne har et autogenerated id som også er primær nøgle, som bruges til at skabe relationer tabellerne imellem som foreign keys. Derudover bruges dette id eksempelvis til at checke om den bruger der tilgår siden har et id, har brugeren ikke et id, er brugeren ikke registreret i databasen, og tanken var at brugeren så skulle dirigeres til forsiden. Der kan ikke skrives ned til Toppings eller Bottoms i programmet. Tabellen Users indeholder desuden attributten adminId som meget enkelt fortæller om brugeren er admin eller ej. Er adminId 1 er man admin. Man kan ikke oprette admins fra programmet og skal derfor gøres direkte i databasen.



Navigationsdiagram

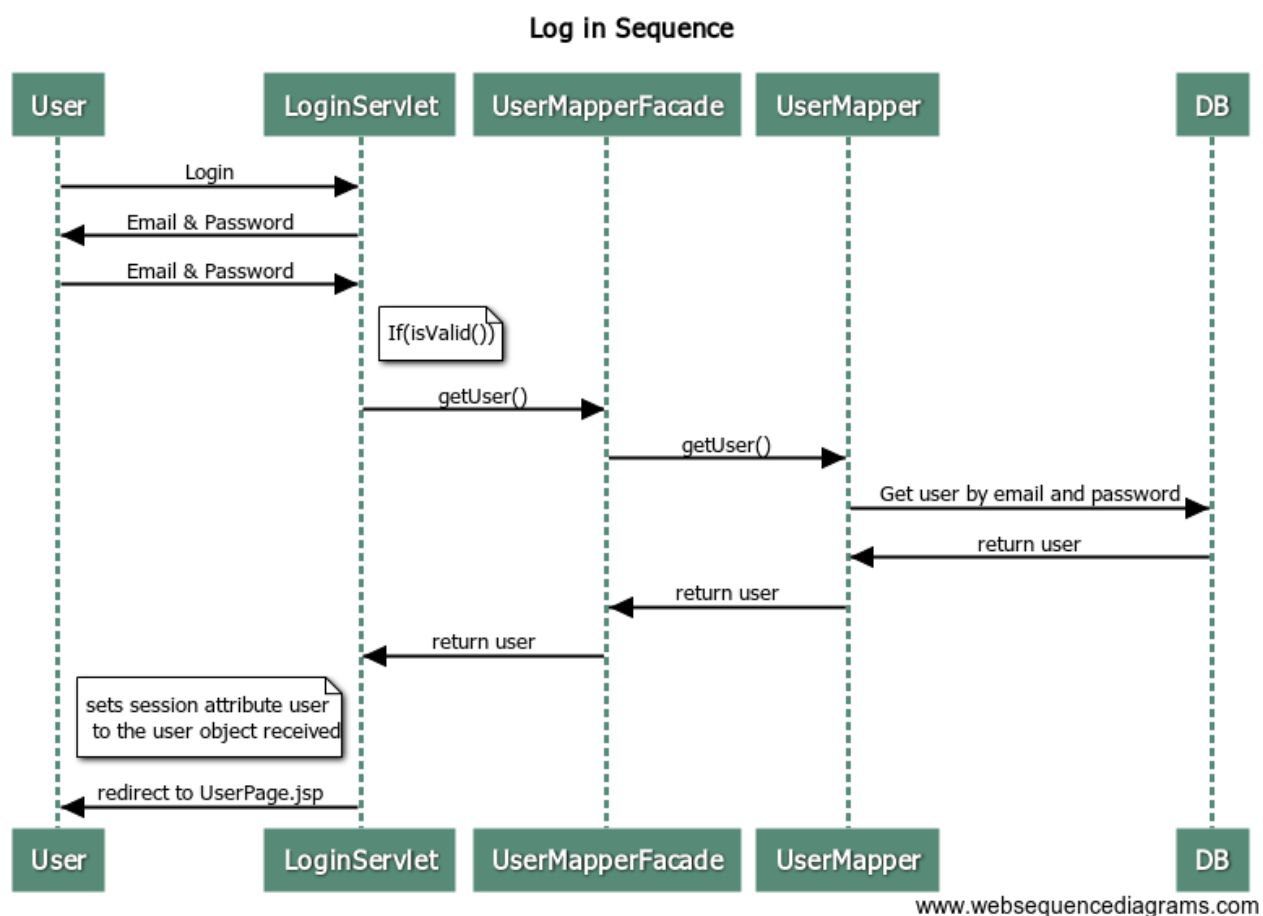
Shop servleten sender brugeren videre til `ErrorPageAddCupcake` ved fejlinput eller manglende input på `UserPage.jsp` og `UserPaymentPage.jsp`, `ErrorPageAddCupcake` viser en sessions attribut som er kaldt "errorMessage". Attributten "errorMessage" sættes alt efter hvilken fejl der er tale om, eks. Hvis brugeren har glemt at vælge en bund eller topping, eller hvis brugeren prøver at fjerne en cupcake fra indkøbskurven uden at have valgt en cupcake at fjerne. Når brugeren logger ind gemmes det User objekt der laves i sessionen som attributten "user". Den bruges til at sikre at den user der er logget ind rent faktisk eksistere i databasen ved at tjekke userens id, som først bliver lavet når useren oprettes, hver gang brugeren kommer med et request. Når en bruger logger på laves et `ShoppingCart` objekt som indeholder en liste med `orderLines`, denne liste tilføjes straks efter de ordre som brugeren tidligere har valgt men endnu ikke købt (`getUnfinishedOrderLines()`). `ShoppingCart` objektet gemmes så i session som attributten "cart". Når en bruger er logget ind og tilgår en fejlside eller en konfirmationsside vil der blive lavet to hidden parametre med hjælp fra sessions attributten "user", her gemmes userens email som "email" og password som "password", de to parametre bruges til at logge brugeren ind automatisk når han vælger at trykke på tilbage knappen.



Sekvensdiagrammer

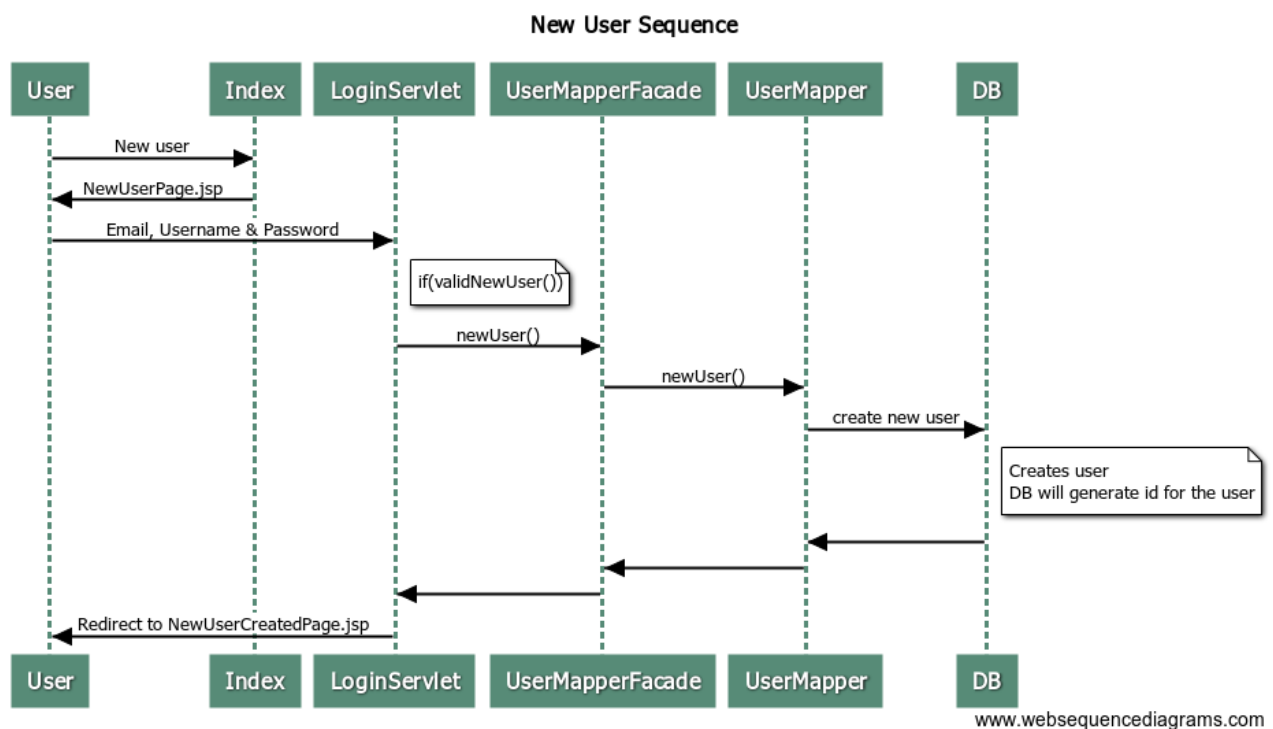
Log ind sekvens:

Brugeren indtaster email og password. Login servletten kalder så metoden isValid fra UserValidate klassen (fremgår ikke af billedet (note til højre for LoginServlet)) hvor den tjekker om Useren eksisterer nede i databasen, hvis ikke sendes brugeren til ErrorPage.jsp. Hvis Useren eksisterer i databasen kreeres et User objekt som sendes retur. Dette objekt gemmes i sessionen, og sender brugeren til UserPage.jsp.



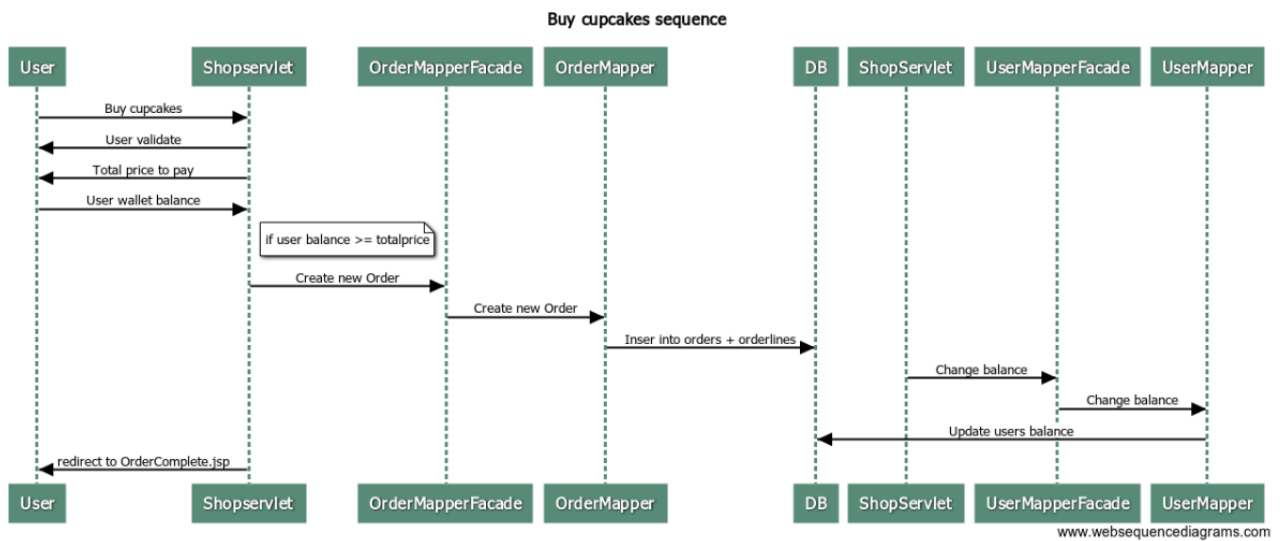
Ny bruger sekvens:

Her indtaster brugeren sit ønskede brugernavn og password samt sin email. Login servletten kalder så metoden isValid fra UserValidate klassen (fremgår ikke af billedet (note til højre for LoginServlet)) hvor den tjekker om Useren eksisterer nede i databasen, hvis ikke sendes brugeren til NewUserCreated.jsp. Hvis Useren eksisterer i databasen dirigeres brugeren hen til ErrorPageNewUser.jsp



Køb cupcake sekvens:

Brugeren har trykket på knappen "Buy cupcakes". Shop servleten tjekker så om brugeren har penge nok på sin konto, hvis han har det bliver ordren skrevet ned i databasen, brugerens indkøbskurv tømmes og brugerens balance opdateres nede i databasen. Brugeren bliver herefter sendt til OrderComplete.jsp



Særlige forhold

Session informationer:

User objektet der laves når en bruger logger ind gemmes i sessionen, mest for at få userens balance med hele vejen igennem programmet, men også for at få userens id med, for at tjekke om brugeren har lov til at være på siden, bruger objektet får først sit id, når der bliver logget ind, nede fra databasen.

ShoppingCart gemmes i sessionen fra Login servletten for at få den med over på User siden.

Derudover tages den med over på betalingssiden for at man kan se hvad man har bestilt.

Exceptions:

Håndterer nullpointer exception i shop servletten, hvis brugeren ikke har valgt en topping eller bund.

Status på implementation

Ville gerne have lavet det visuelt flottere, evt stylet det med bootstrap. Har ikke nået det grundet tidspres og gruppemæssige komplikationer.