Cupcake

Daniel Lindholm	cph-dl110@cphbusiness.dk	Hupra
Jacob Borg	cph-jb308@cphbusiness.dk	Jack-Borg
Nikolaj Thorsen Nielsen	cph-nn134@cphbusiness.dk	NikolajX4000
Stephan Marcus Duelund Djurhuus	cph-sd115@cphbusiness.dk	Stephan-D-CBA

Klasse: A

18-03-2018

Indledning

Dette java web projekt giver brugeren mulighed for at oprette en bruger/logge ind og sammensætte forskellige former for cupcakes efter eget ønske ved at vælge topping, bottom og antal.

Man vil så kunne tilføje de cupcakes man har sammensat til sin kurv hvorefter man vil kunne købe dem og hente dem i vores fiktive butik.

Hvis ens bruger har admin rettigheder er det muligt at tilføje nye cupcake toppings og bottoms til udvalget. Det er også muligt at gå igennem alle de forskellige køb sidens brugere har lavet for at se hvad de har købt og redigere dem hvis nødvendigt.

Baggrund

“You are to develop a simple web-shop using MySql database, java servlets and jsp pages on the backend and html, css and javascript on the frontend.

The web shop sells cupcakes, but only as pick-up. Customers can use the web shop to place an order and then show up in person to pick up the cupcakes. Sending cupcakes in the mail has shown to be a bad idea as they turn up with bite marks if at all.

The bakery has a very fast cupcake-machine, so the instant the order is placed the cupcakes are ready for pickup.

The cupcakes have a bottom and a topping which can be combined in many ways, but a cupcake must always have both a bottom and a topping.

Customers each have an account with the shop and orders can only be placed if the account hold enough money to cover the price. Payment is handled by another system and as of now deposits will have to be handled manually in the database, but withdrawals happen when cupcakes are ordered.

In order to pay with their account the customers will have to use a username and a password to login before placing an order.”[1]

Teknologivalg

Projektet er et Maven 3.1 projekt, som vi har skrevet i NetBeans IDE 8.2. Databasen, mysql 5.1.39, er lavet i MySQL Workbench 6.3ce, og ligger på en Ubuntu 16.04.3 x64 server, hvor websitet er deployet vha. apache-tomcat-8.0.32. Sprogene der er brugt i koden er Java jdk1.8.0_141, HTML5 og Javax 7.0.

[1] Opgaveformulering

Krav

Dette projekt skal tage udgangspunkt i en webshop med bruger registrering, login, shop og administrerende brugere. Projektet bygger meget på forbindelse mellem webshop og database, derfor skal databasen også tilgås med forskellige rettigheder, og i forskellige situationer.

Bruger registreringen forbinder brugeren til vores 'users' tabel i vores database, hvor han eller hun bliver tilføjet.

Efter registreringen vil brugeren nu få mulighed for at logge ind med brugernavn og kode. Dette vil omdirigere brugeren til vores shop hvor man kan lave sine cupcakes.

Vores cupcakes er lavet af tre dele topping, bottoms og en sammenlagt pris af de to individuelles. Disse informationer henter vi fra vores tabeller 'toppings' og 'bottoms'. dette gør at vores liste af valg vil være identisk med vores tabel på databasen.

Når brugeren er færdig med at tilføje cupcakes i kurven og klikker checkout, han kurvens indhold blive tilføjet som en ordre i vores ordre tabel på databasen. Derefter kan man som bruger nu se den bestilling man har lavet sammen med de tidligere bestillinger, hvis man har nogen.

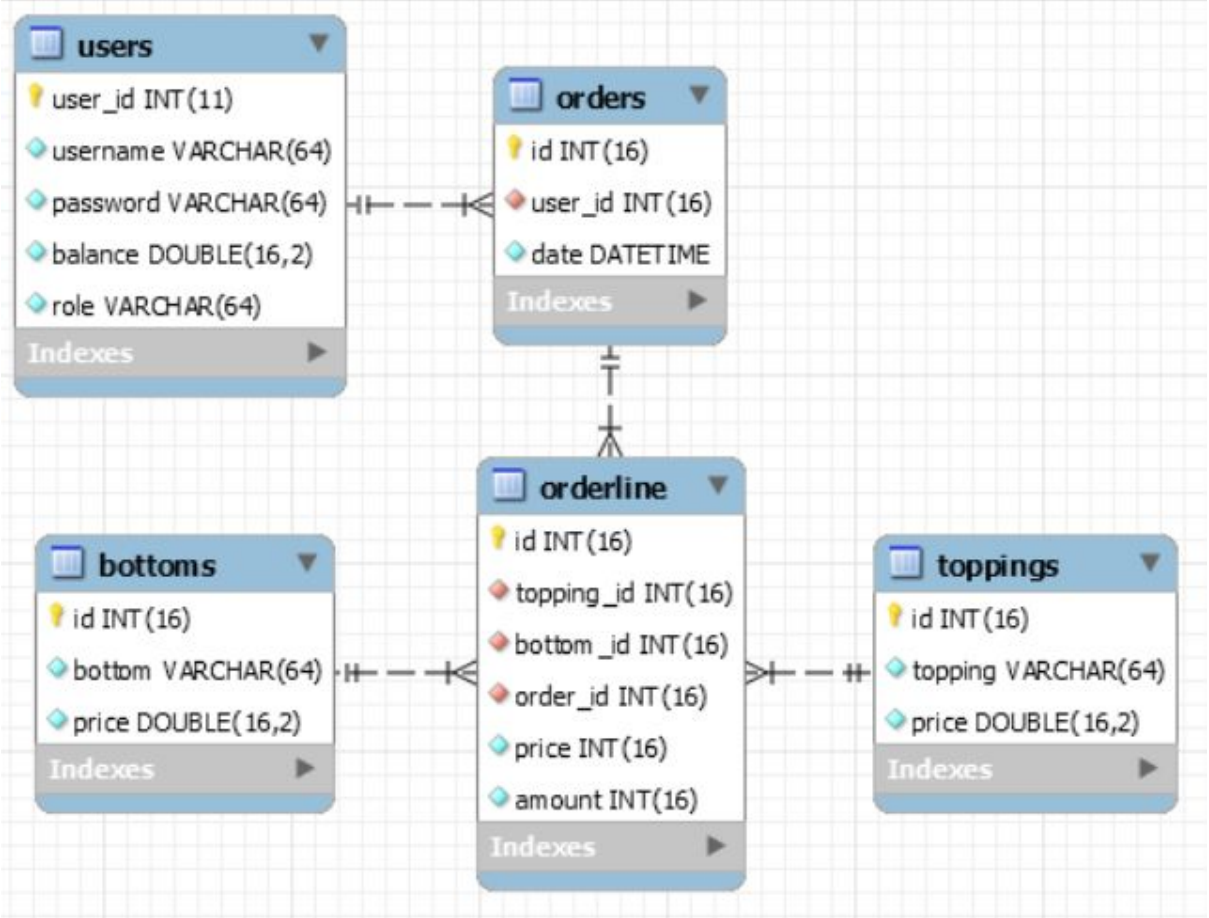
Som administrerende bruger har man mulighed for at se og ændre alle ordre samt tilføje nye toppings og bottoms. Som admin får du en ekstra side kaldet 'admin', som er synlig men ikke tilgængelig for andre brugere. På denne side kan man se alle ordre samt hvem der har lavet dem og hvornår de blev lavet.

Hver ordre henviser til en mere detaljeret beskrivelse, hvor man kan se totalpris og hvilke cupcakes der er blevet bestilt. Her har administratoren en mulighed for at ændre antallet af bestilte cupcakes. Efter ændring af antallet vil dataen blive sendt til vores database og opdatere vores 'orders' tabel.

For at tilføje nye varianter af cupcakes skal man også være administrator. Denne funktion skaber forbindelse til vores 'toppings' og 'bottoms' tabeller, hvor man kan indsætte de nye varianter. Dette gøres ved at give dem et navn, en pris og definere om det er en topping eller bottom.

Alle brugere har muligheden for at logge ud, dette sker ved at omdirigere til vores login side og nulstille sessionen.

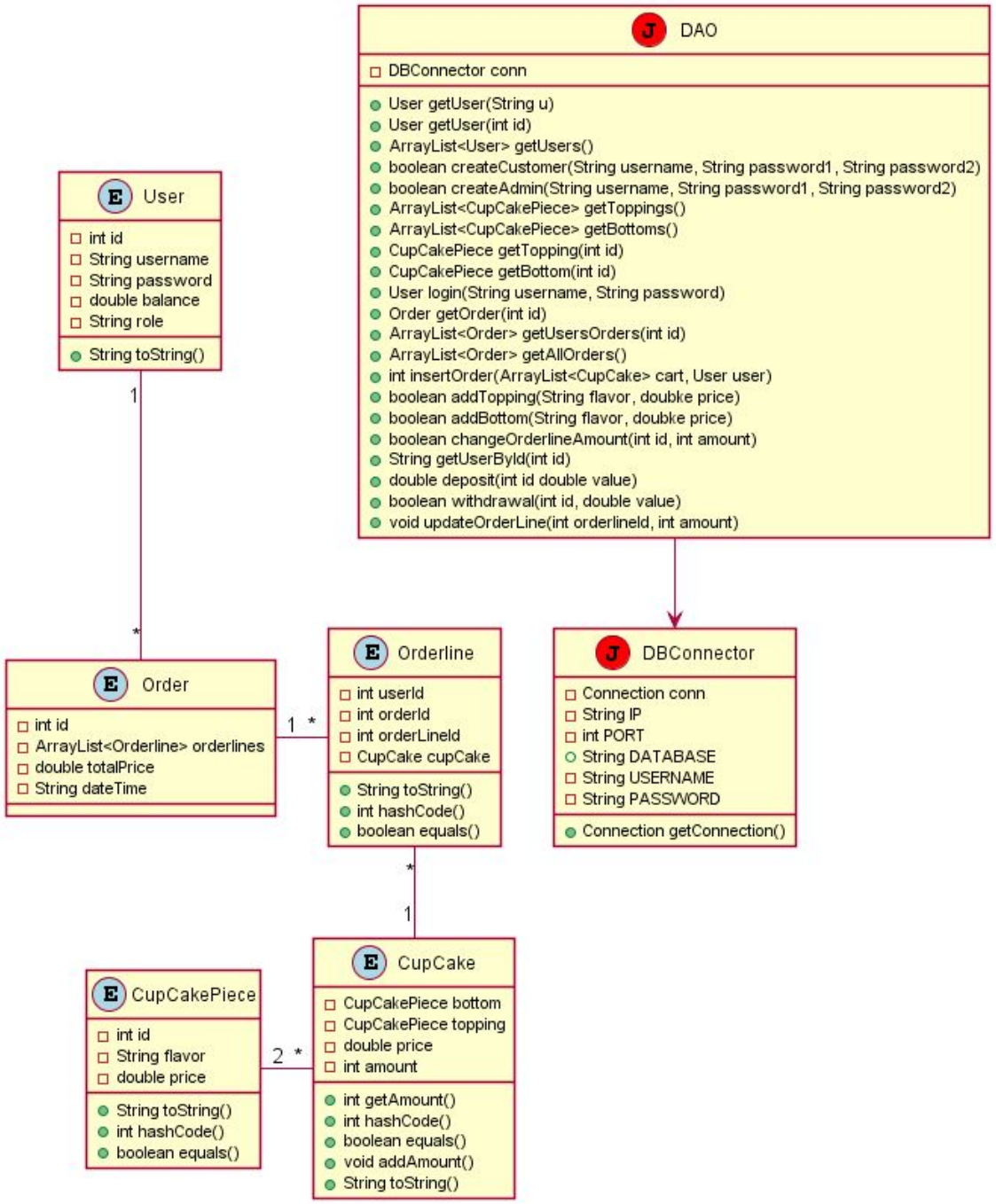
Domæne model og ER diagram



I vores “users” tabel opbevare vi vores bruger data.

En “orders” indeholder en foreign key til den “users” som har lavet den.

En “orderline” har foreign key “order_id” til den “orders” den tilhøre. den har bottom_id og topping_id til dens “bottoms” og “toppings”.



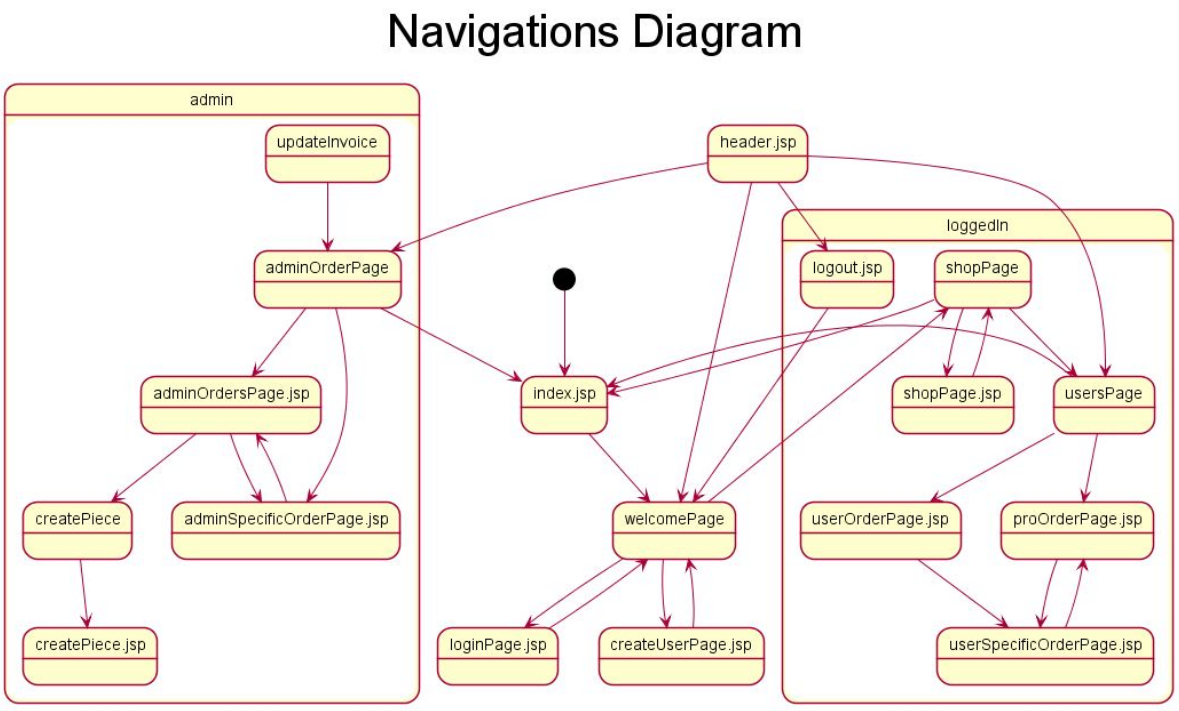
User til Order er 1-til-mange. En bruger kan have mange ordrer. En ordre kan tilhører én bruger.

Order til Orderline er 1-til-mange. En ordre kan indeholde mange linjer. En linje kan være på én ordre.

CupCake til Orderline er 1-til-mange. En cupcake kan være på mange linjer. En linje kan indeholde én type cupcake.

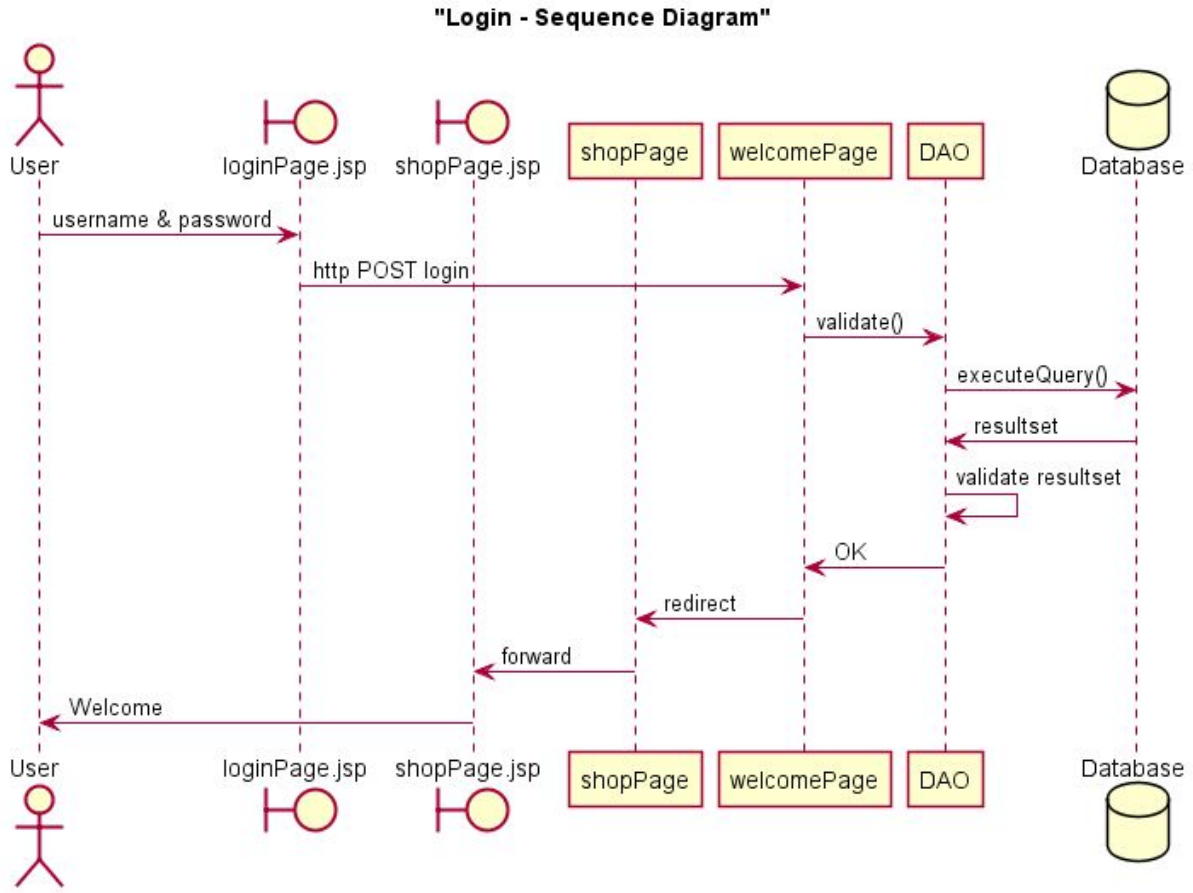
CupCakePiece til CupCake er 2-til-mange. En cupcake består af to cupcake dele. En type cupcake del kan være i mange cupcakes.

Navigationsdiagram



Der er ikke nogen naturlig sidste side, da alle siderne er forbundet, da header.jsp og footer.jsp er included på alle sider.

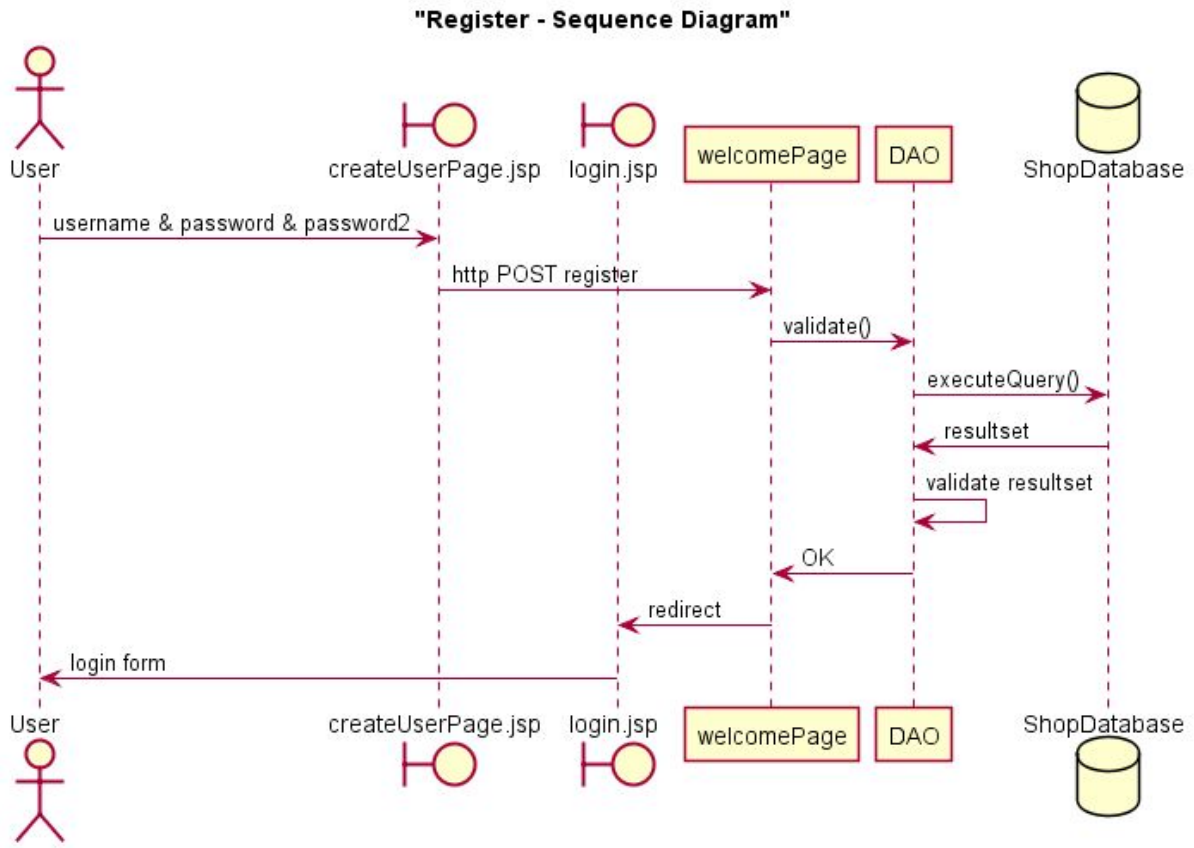
Sekvensdiagrammer



Brugeren indtaster sit username og password på loginPage.jsp.

Det bliver sendt til welcomePage(servlet) som bruger det som parameter til Data Access Object metoden.

Vi antager at oplysningerne er valide, så vi bliver sendt hen til shopPage(servlet), der forwarder til shopPage.jsp.

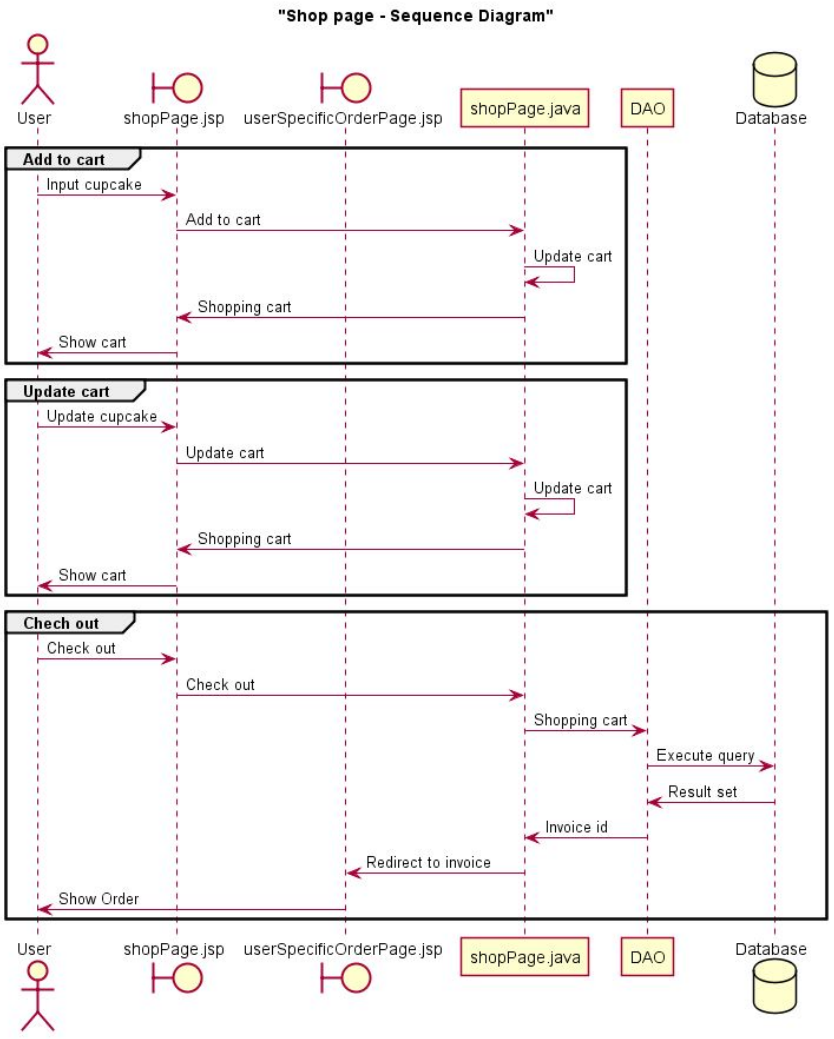


Brugeren starter med at indtaste et brugernavn, password og gentag password på createUserPage.jsp.

Det bliver så sendt til welcomePage(servlet) som så bruger det som parameter til Data Access Object metoden.

Vi antager at oplysningerne er valide og brugeren så bliver oprettet i databasen.

Derefter redirecter welcomePage(servlet) til login.jsp hvor brugeren vil kunne logge sig ind.



Add to cart

Brugeren indtaster en cupcake og antal på shopPage.jsp og trykker “add to cart”, den nye cupcake bliver sendt til shopPage(servlet) og lagt i kurven som er gemt i session.

Derefter bliver der forwarded tilbage til shopPage.jsp.

Update cart

Brugeren ændre antal på en eller flere cupcakes på shopPage.jsp og trykker på “update cart” som så bliver bliver sendt til shopPage(servlet), hvor kurven bliver opdateret og derefter opdateret i session.

Derefter bliver der forwarded tilbage til shopPage.jsp.

Check out

Når brugeren trykker på “check out”, på shopPage(servlet) bliver kurven via DAO’en sendt til databasen som returnerer id på den order, så redirecter shopPage(servlet) til den invoicen af den ordre.

Særlige forhold

Vores indkøbskurv bliver gemt i session så den hele tiden er opdateret. Vi gemmer også vores user i session så vi har adgang til objektet fra alle sider blandt andet for at kunne se om brugeren er logget ind

Exceptions bliver håndteret i vores Data Access Object hvor de bliver logged.

Vi har lavet både front og backend validering. Som frontend validering har vi brugt HTML validering såsom required, type osv. og vi har så lavet backend validering til hvis man prøver at snyde frontend valideringen eller der sker en fejl.

Vi beskytter brugerens data ved at lave login formen som POST, og beskytter vores database ved at bruge Prepared Statements for at undgå SQL-injection. Vi har også sikret mange af vores sider ved at tjekke om man er logget ind og ellers smider vi folk tilbage på forsiden/login-siden.

Der er blevet lavet en bruger i databasen som kun har adgang til det skema som vi bruger i projektet.

Status på implementation

Vi mangler at:

- implementere deposit, så brugeren kan sætte penge ind.
- implementere withdrawal, så der kan trækkes penge fra brugerens konto.
- implementere et check på om brugeren har penge nok til at købe sin ordre
- skjule admin knappen for bruger der ikke er admin