

CPH Business

Datamatiker

B-Klassen



Cupcake - Webprojekt

Morten Bokkenheuser-Bæhr
cph-mb537@cphbusiness.dk
git: MortenBB

Michael Ejdal Lundsgaard
cph-ml510@cphbusiness.dk
git: Gonron

Johachim Raimund Gulstad
cph-jg133@cphbusiness.dk
git: Johachim

18. Marts 2018

Indledning 1	3
Baggrund 1.1	3
Teknologiske Valg 1.2	3
Klassediagram 2	4
EER Diagram 3	5
Navigationsdiagram 4	6
Sekvensdiagram 5	7
Særlige Forhold 6	8
Status på Implementation 7	8

Indledning 1

Formålet med dette webprojekt, cupcake, er at få oprettet en hjemmeside, som henter informationer og gemmer nyt data i en database som ligger på en droplet på digitalocean. Kravene til opgaven har vi fået af underviserne på Copenhagen Business Academy.

Baggrund 1.1

Meningen med hjemmesiden, er at en bruger kan gå ind på hjemmesiden, oprette en brugerprofil og derefter logge ind med den oprettede brugers brugernavn og password.

Efter brugeren har fået oprettet sig på hjemmesiden, kan vedkommende lave sine egne cupcakes, ved at vælge imellem forskellige bunde og toppe. Brugerens bestilling sendes herefter til en database så der kan holdes styr på alle bestillingerne.

Teknologiske Valg 1.2

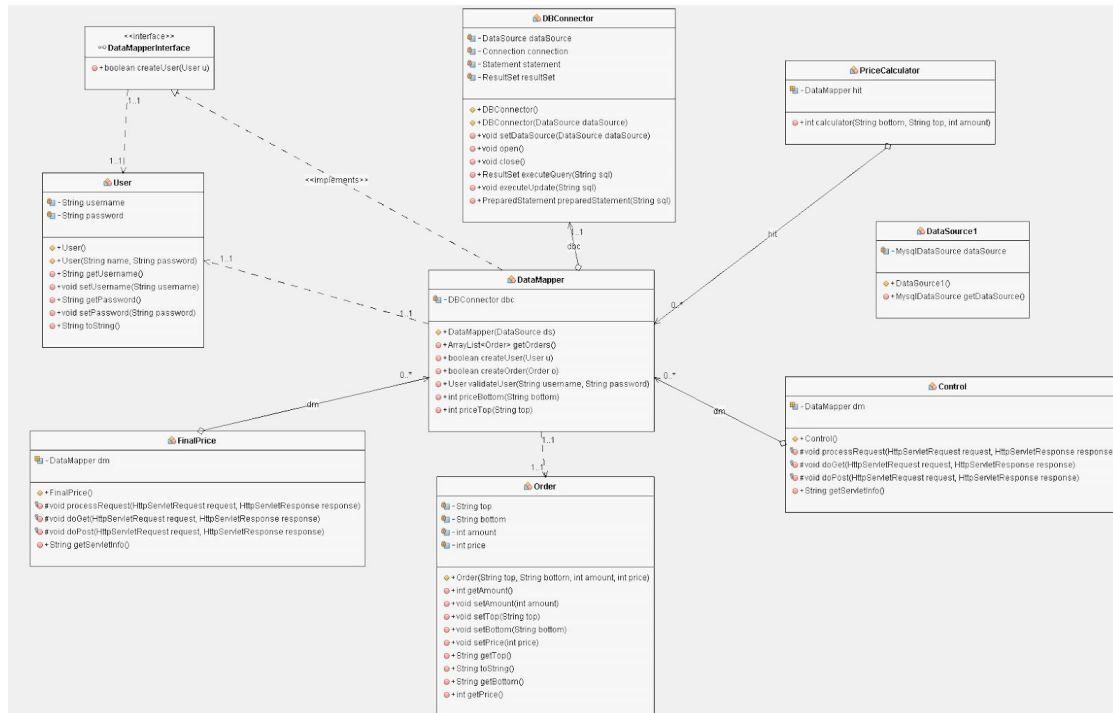
Til dette projekt har vi gjort brug af følgende, forskellige programmer.:

- Netbeans IDE 8.2
- MySQL 6.3
- Tomcat 8
- Haveged 1.9.2
- Git 2.16.2
- Droplet, digitalocean
- Filezilla 3.31.0
- Notepad ++ 7.5.5 *eller* Notepad 1709
- PuTTY 0.7
- Java SE Development Kit 8

De programmer der skulle installeres på vores droplet var:

- Tomcat 8
- MySQL 6.3
- Haveged 1.9.2
- Java SE Development Kit 8

Klassediagram 2

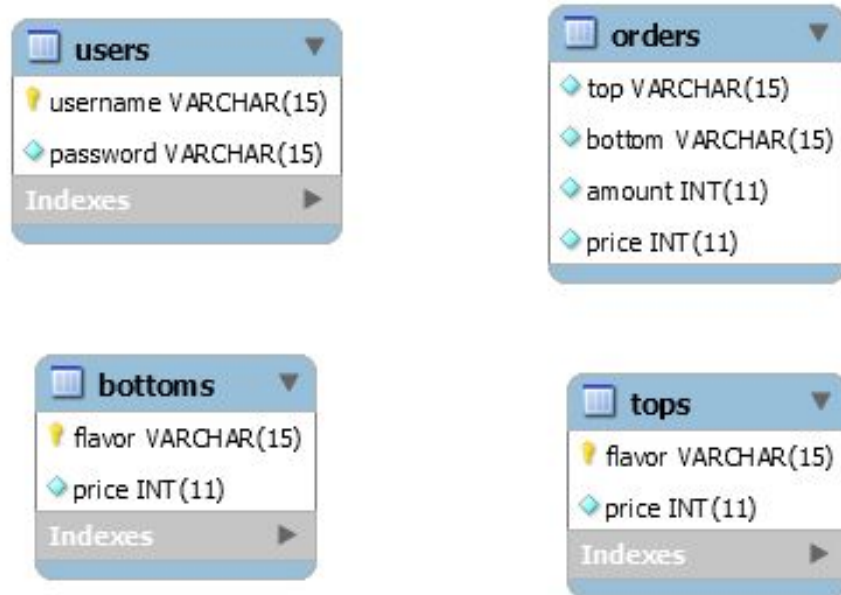


Et klassediagram over vores java klasser <https://goo.gl/wZKVnM>

link da det er et meget lille billede

I klassediagrammet kan man se sammenhængen mellem klasserne i cupcake projektet. I vores DataMapper klasse har vi alle metoderne der bruger databasen på den ene eller den anden måde.

EER Diagram 3



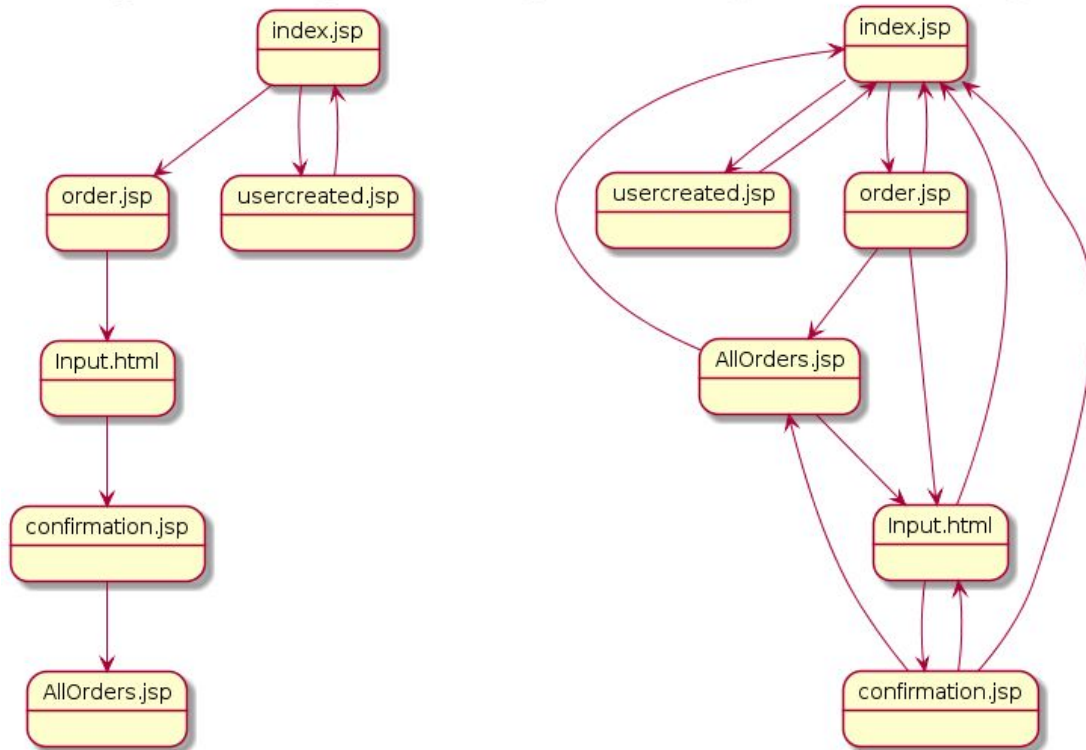
Vores EER diagram

Vores users har username sat til primary key så man ikke kan oprette flere brugere med samme brugernavn. Orders tabellen har ingen primary key fordi man skal kunne bestille flere af den samme ordre.

Vi har valgt at lave vores database uden brug af foreign keys. Konsekvensen af dette er at man lige nu ikke kan se hvilke bruger der har bestilt ordrene. Hvis man skulle optimere vores database skulle man gemme et bruger ID i ordrene og give vores tops og bunde et ID nummer så det er nemmere at holde styr på ordrene, det kommer vi mere ind på senere i rapporten.

Navigationsdiagram 4

Navigationsdiagram Navigationsdiagram med navigations-bar

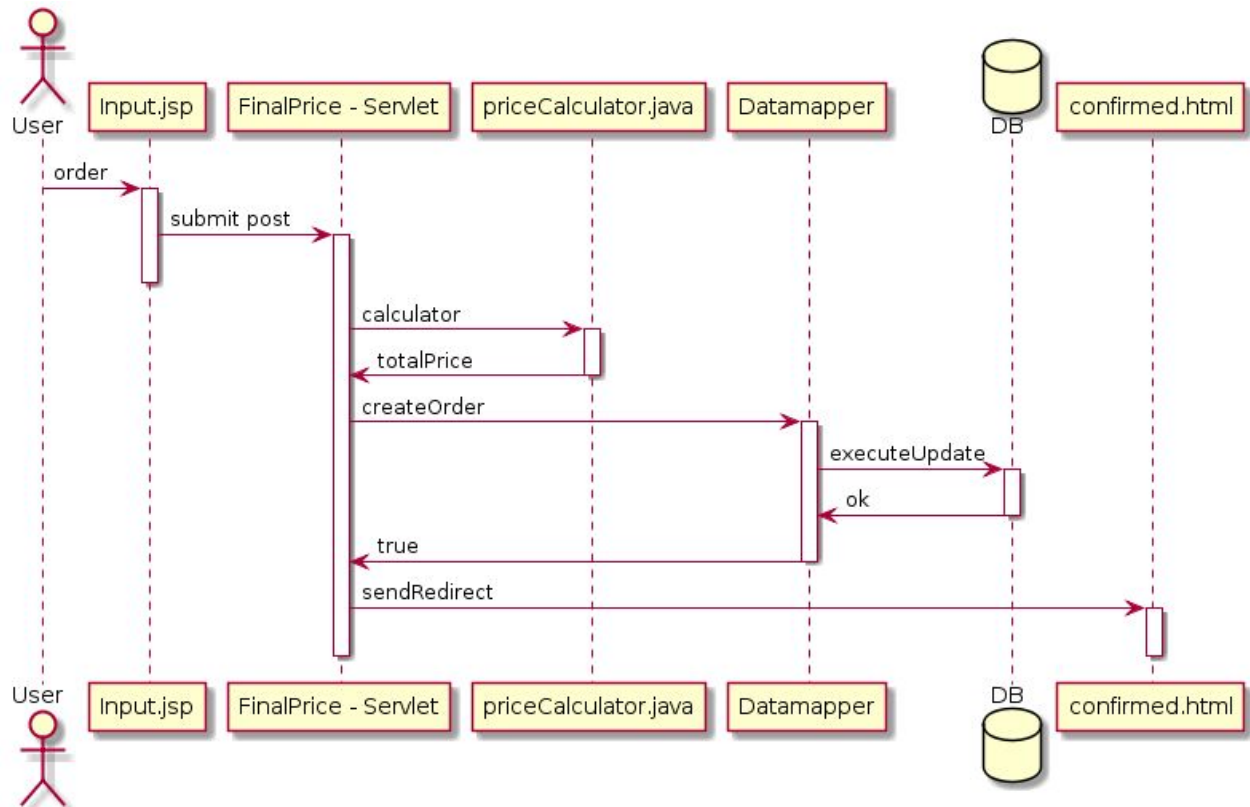


Vores navigationsdiagram. Bemærk at vi i vores projekt benytter navigations diagrammet uden navigations-bar.

Som der kan ses er vores navigationsdiagram næsten lineært. Kun når man skal oprette en ny bruger, bliver man sendt tilbage til forsiden, dette gør det nemt for brugeren at navigere på siden fordi de bliver tvunget til at gå videre til næste side i stedet for selv at skulle finde rundt på siden. En af ulemperne ved ikke at have en navigations-bar er, at man skal indlæse siden igen hvis man kommer til taste forkert på loginsiden, dette ville have været unødvendigt hvis man havde en navigations-bar fordi man bare kunne gå til loginsiden igen.

Hvis vi havde implementeret navigationsbaren havde der været større mulig for at brugeren var "fare vild" på siden, til gengæld havde der været større frihed fordi brugeren havde mulighed for at gå tilbage og bestille flere ordre efter at de havde fuldført en bestilling og også at skifte bruger hvis det var nødvendigt.

Sekvensdiagram 5



Dette er et sekvensdiagram over vores cupcake projekt.

I vores sekvensdiagram har vi taget udgangspunkt i et scenarie hvor kunden allerede har valgt hvilke og hvor mange cupcakes han/hun skal have. Kunden klikker på 'order' hvorved vores servlet tager over derfra. I servleten laver vi en calculator der udregner total prisen og så laver vi en ny ordre med top, bund, antal og pris, og gemmer den i databasen. Til sidst bliver brugeren redirected til confirmation siden.

Særlige Forhold 6

Når man logger ind sørger vores servlet for at oprette en ny session for brugeren. Det vil vi bruge til at associere ordrene med den bruger der har afgivet ordren.

Status på Implementation 7

Databasen vi har udarbejdet har ikke alle delene af CRUD (Create, Read, Update, Delete) metoden. Vores database kan på nuværende tidspunkt kun udføre 'Create' og 'Read'.

Af og til når projektet køres, sker det at nogle af siderne ikke er stilet, selvom koden bag burde have implementeret stylingen. Dette kan dog hurtigt løses ved at 'refreshe' siden via ctrl+f5.

Vi har ikke nogen form for sikkerhed på hjemmesiden lige nu, det gør ikke så meget lige nu da der ikke er nogen forskel på brugerne og man bliver logget ind som gæst hvis brugeren ikke findes. Det vil alligevel være en god ide at implementere noget sikkerhed senere i projektet når man skal til at lave administrator brugere der kan se alle ordre, slette og opdatere ordre.

Der er nogle mangler i dette projekt, som vi gerne ville have fået lavet hvis vi havde haft mere tid. For det første, er der ingen sikkerhed på vores database. Når en bruger opretter sig, bliver passwordet bare gemt i databasen. Her ville vi gerne implementere en metode hvor passwordet bliver krypteret.

Derudover kan vi lave vores database struktur meget bedre. tage brug af fremmede nøgler, give vores cupcake-toppe og bunde hver deres ID.