

2. Semester Datamatiker. Afleveret d. 8. Oktober 2017

Cupcake - En Mini Webshop

Walter Hansen - cph-wh102@cphbusiness.dk

Bo Henriksen - cph-bh136@cphbusiness.dk

Benedikte Majbrink - cph-bm109@cphbusiness.dk



Indledning

I dette projekt skal vi udvikle en simpel webshop løsning, der bruger en database, java servlets og jsp sider som backend, og html, css, bootstrap og javascript til frontenden.

Baggrund

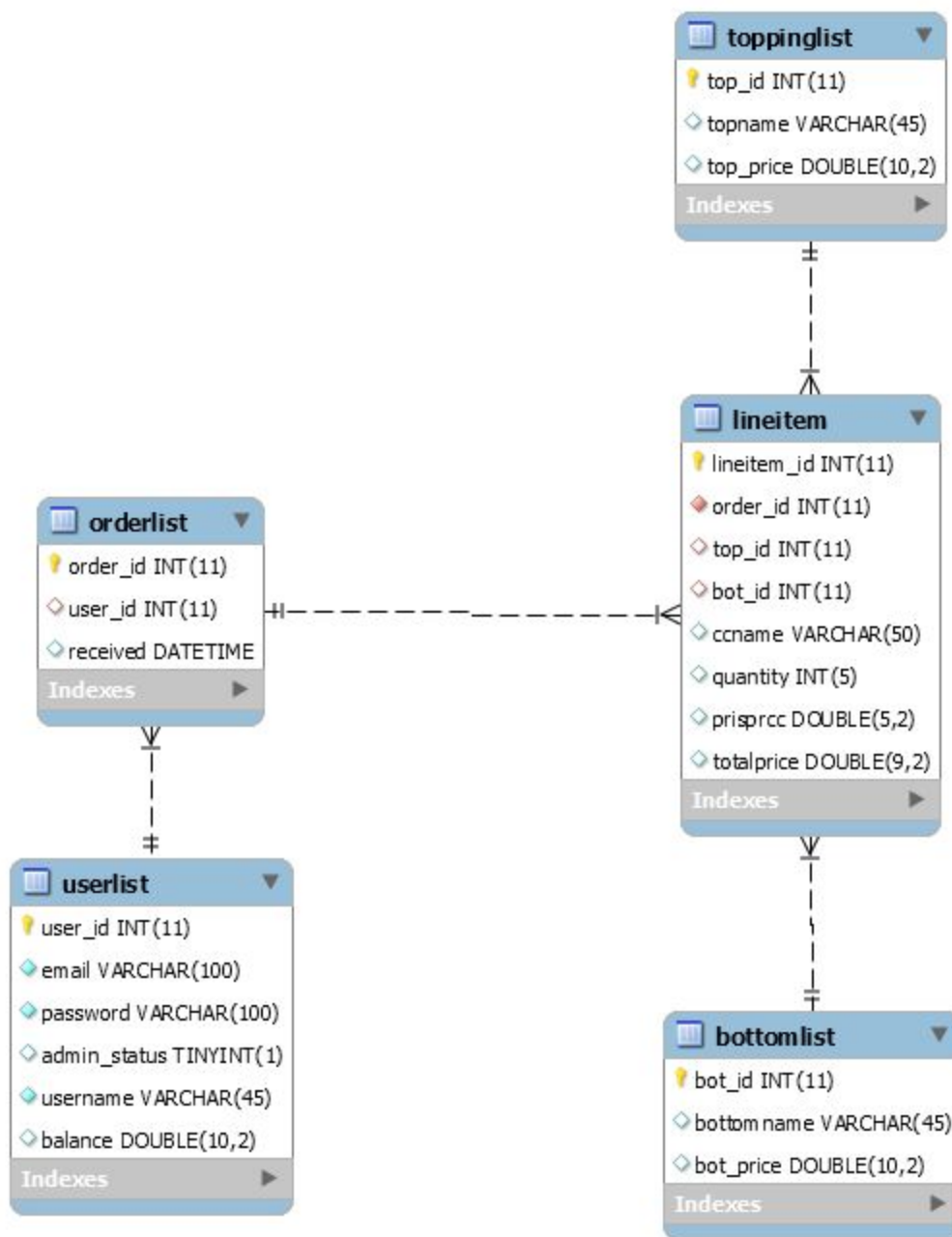
Vi har haft kontakt med et lille bageri, der producerer forskellige kager. De vil gerne tiltrække flere kunder ved hjælp af nemme online løsninger. Derfor ønsker de at få udviklet en hjemmeside som deres kunder kan finde information og bestille deres produkter fra. De har derfor hyret os til at lave en hjemmeside, der har et let og fint design, hvor deres kunder kan gå ind og finde information om deres produkter og bestille cupcakes. Det som de kan tilbyde er forskellige kagetoppe og forskellige kagebunde, som kan sammensættes efter kundens ønsker, og bestilles fra hjemmesiden.

Bageriets kunder skal kunne registrere sig som kunde samt logge ind og bestille cupcakes, hvis de har nok penge på deres konto.

Teknologivalg

- Java NetBeans IDE 8.2
- Linux Ubuntu Server 16.04.3
- MySQL 5.7.19
- MySQL JDBC 5.1.3.9
- Tomcat8 Server
- MySQL Workbench 6.1 / 6.3 (Windows XP / Windows 10 kompatibilitet)
- Git Bash v. 2.10.0 commando tool

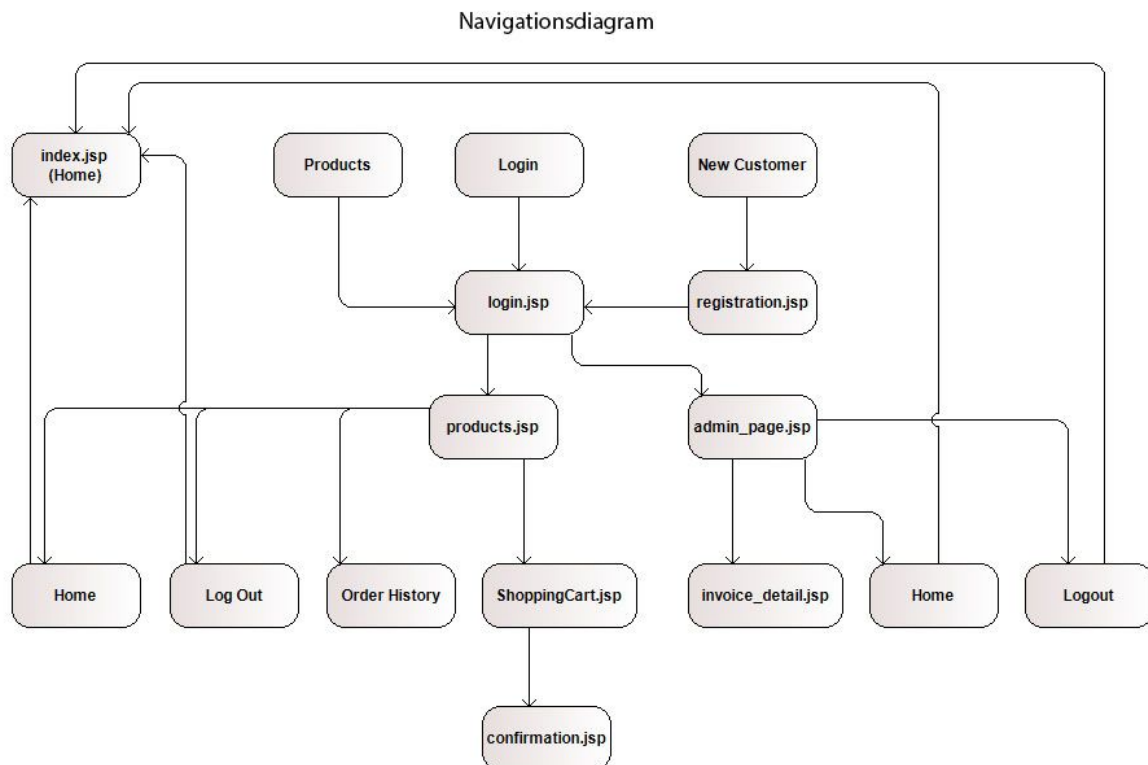
Domaine og ER model



Vores ER-diagram har ændret sig en del undervejs. Ovenstående ER-diagram viser, hvordan databasen ser ud pt.

En af de constraints vi bruger er "not null". F.eks. Bruger vi det i userlist tabellen. Vi har sat username, email og password til "not null". Det betyder, at der ikke kan oprettes en ny bruger med mindre alle 3 felter er udfyldt. Vi kan dermed sikre os, at vi får både username, password og email på alle nye brugere. Et andet sted vi bruger "not null" er i vores lineitem. Her er det order_id, der har constraint "not null". Det er vigtigt, at vi får ordre_id'et med over i cupcakelisten, når der indsættes en ordre, da vi så kan finde alle de kager, der er på ordren baseret på ordre_id.

Navigationsdiagram



NB. Starten af navigationsdiagrammet er den øverste linje, der består af (index.jsp, Products, Login og New Customer. Den øverste linje repræsenterer hovedmenuen på startsiden.

Cupcake shoppen benytter sig af en "navigationsbar" i toppen. Dog er der nogen menuer, som kun kan tilgås, alt efter om du er logget ind som kunde eller administrator, samt der er en logout knap i toppen når man er logget ind.

Når en kunde er logget ind har kunden mulighed for at vælge menupunktet "Order Cupcakes" og "Shopping Cart".

En administrator har muligheden for at vælge menuen "Customers Order", når vedkommende er logget ind.

Cupcake shoppen består af følgende jsp sider og servletter der bringer en til den pågældende side.

index.jsp - (LoginServlet henviser til denne side i tilfælde af, at man logger forkert ind.) Denne jsp side henviser bare til andre jsp sidder uden selv at benytte sig af nogen servlet.

login.jsp - Der er ingen servletter, som henviser til siden. Login.jsp tilgås fra andre jsp sider eks. via index.jsp.

products.jsp - LoginServlet henviser til denne side, når en bruger succesfuldt logger ind. Derudover henviser NewProductControlServlet også til siden.

admin_page - LoginServlet henviser til denne side, når en administrator succesfuldt logger ind.

confirmation.jsp - NewProductControlServlet henviser til denne side.

error_page.jsp - InvoiceDetailServlet henviser til denne side.

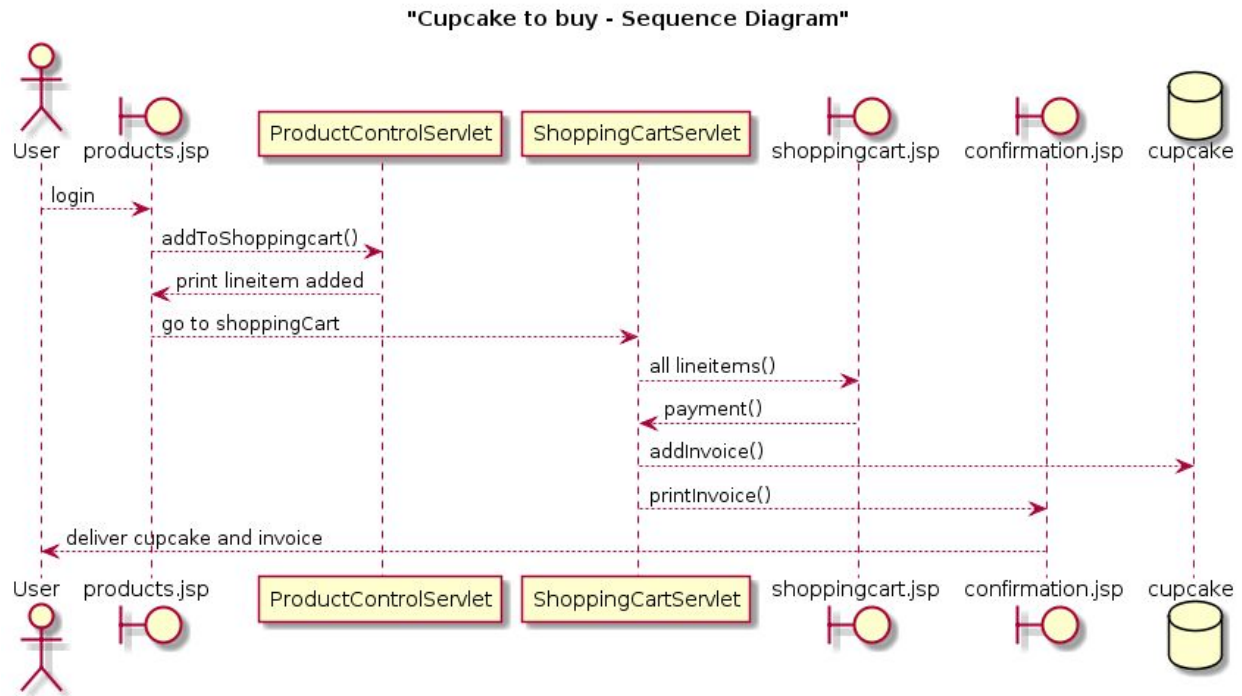
invoice_detail.jsp - InvoiceDetailServlet henviser til denne side.

registration.jsp - UserServlet henviser til denne side.

shoppingCart.jsp - ShoppingCartServlet henviser til denne side. Desuden -

Bliver man henvist hertil fra NewProductControlServlet når der trykkes på <checkout> -knappen på products.jsp.

Sekvens diagram



Diagrammet er meget mangelfuldt i forhold til det beskrevne:

1. User har valgt en topping og en bottom ved at benytte radiobuttons på produktsiden.
2. Han vælger, hvor mange han vil have af den pågældende kage, ved at vælge et antal i "quantity" boksen.
3. Herefter trykkes der på knappen add to "shoppingcart", hvorved oplysningerne om top bund og antal bliver sendt videre til NewProductKontrolServletten.

-
4. Her bliver kagens navn, kagens pris og ordrens samlede pris udregnet og der bliver lavet et shoppingcart (ArrayList af LineItems), hvis det er den første kage kunden bestiller i denne session. Ellers bliver der bare added flere lineitems til shoppingcart-arraylisten, og når det er sket sendes kunden tilbage til products.jsp.
 5. Udover det forsøges der her at blive lagt et lineitem ned i databasen i tabellen lineitem, v.hj.a. En LineItemsMapper og en databaseConnector.
 6. Det forsøges også at oprette en Order som også skal ned i databasen. Her benyttes metoden addOrder fra InfoToAdminMapper.
 7. Kunden får at vide, hvilken kage han har valgt, antallet af den pågældende kage og hvad det koster.
 8. Kunden får endvidere vist, hvad han indtil nu har i sit shoppingcart.
 9. Der vises også, hvad hans resterende kredit på kontoen vil være, hvis han køber det han har i ShoppingCart. Dette udregnes også på NewProductControlServleten ved at lave en tempBalance. Totalprisen trækkes fra kundens balance, der er hentet fra databasen via en UserMapper.
 10. I det øjeblik han trykker på knappen "Checkout" bliver han henvist til shoppingcart servleten hvor han bliver stillet om til shoppingcart.jsp
 11. Her kan kunden vælge om han vil betale, hvis han trykker på "pay" eller om hans ønsker at annullere "cancel" (her vil han blive sendt tilbage til index.jsp". Kunden har også mulighed for at handle mere "get more cakes", hvor han sendes tilbage til products.jsp.
 12. I dette tilfælde vælger kunden at trykke på "Pay", og han bliver via shoppingcart servleten, hvor kundens balance i databasen bliver ændret til tempBalance via metoden: changeUserBalance();
 13. Kunden sendes videre til confirmation.jsp hvor han får vist sin ordre, med ordre id. ShoppingCartListen, den ny balance og et pænt farvel.
 14. Han kan så logge ud ved at trykke på "logout" i navigationsmenuen i toppen.

Javadoc

Vi har valgt at hoste javadoc på github pages. Det kan findes på dette link.

https://bomarconihenriksen.github.io/Cupcake_igen/

Særlige forhold

Information der gemmes i sessions

User bliver når han logger ind gemt i sessions, og dennes oplysninger kan så trækkes ud under hele sessionen indtil denne afbrydes ved at trykke på "pay" eller "logout".

Håndtering af Exceptions:

Dette vender vi tilbage til senere i semesteret.

Valg af brugerinput:

På index.jsp (eller Home) kan man kun komme videre via navigationsmetoden ved at trykke på menupunkterne. Hver af menupunkterne er blot et almindeligt link.

Ved registrering af en bruger indtastes brugernavn i et almindeligt tekstfelt. Email indtastes i et andet tekstfelt af typen email. Password og Password repeat felterne er af typen "password" som skjuler de tegn brugeren indtaster på siden.

Der befinder sig også en checkbox på siden, som bruges til at brugeren kan gemme navn og password på siden.

Når brugeren har indtastet oplysningerne trykker han på en submit knap som ligger inden for en form hvor post-metoden benyttes til at stille om til den servlet er sat som action.

Valg af toppe og bunde sker v.h.j.a radio buttons og antallet vælges i en nummer-vælger med et minimum på 1.

Ellers er der submit knapper af forskellig slags, der ligger inde i en form, som henviser til servletter, hvor de forskellige oplysninger kan bearbejdes.

Cancel knappen er en knap med et link, der viser tilbage til index.jsp.

Sikkerheden i forbindelse med login:

Når brugeren taster sit password fremstår bogstaverne som cirkler, så det ikke er muligt at se det indtastet password. En mulighed kunne være at tjekke for længden, og kompleksiteten af passwordet. Dette er dog ikke noget, vi har implementeret. En anden mulighed er, at implementere https, så den information, som sendes mellem kundens browser og databasen er krypteret.

Valg af brugertyper i databasen og hvordan er de brugt i jdbc:

Der findes to typer af brugere:

En kunde bruger som kan logge ind, vælge en cupcake kombination, bestille, betale kager og få en ordrebekræftelse på skærmen.

En admin bruger der kan gå ind, og se de kunder der er oprettet og deres ordrer. Ordrene kan desuden også ses i MySQL databasen.

Disse brugere ligger i den samme tabel i databasen og det der adskiller dem er om en boolean er true eller false. 1 for admin.

Status på implementation

I opgaveformuleringen er der nogle ting der skal være opfyldt, og så er der andre ting hvor vi selv har haft en forventning om hvad vi gerne ville have med både ud for hvert af de næste menupunkter, men også mere generelt.

Generelt:

Implementeret:

- Vi har lavet en pæn hjemmeside v.hj.a. Et bootstrap-tema.
- Lavet en registreringsformular ud fra de stillede krav om farvevalg og centrering af formular.
- Man kan navigere rundt på hjemmesiden på de fleste menupunkter.

Mangler:

- Når den almindelige bruger er logget ind på sin konto kan han endnu ikke komme til en side, hvor han kan se "Order History" selvom menupunktet er der. Ligeledes ved at trykke på menupunktet "ShoppingCart" i navigationsmenuen kommer han heller ikke nogen vegne, hvad enten han har tilføjet noget til shopping-cart eller ej.
- Når admin bruger er logget ind kommer denne heller ingen vegne ved tryk på "Customers Order".

Registrering:

Implementeret:

- Man kan tilføje en ny bruger i databasen.
- Man får at vide hvis registrerings felterne ikke er udfyldt
- Indtastning af email skal være: x@xx.xx

Mangler:

- Eventuelt at der er nogle større krav til længden af password og kompleksiteten.

Login:

Implementeret :

- Man skal være registreret for at komme ind på produktsiden products.jsp, hvilket giver lidt sikkerhed.
- Man kan hente brugeroplysninger fra databasen.
- Man kan kun logge sig ind som administrator, hvis man har administrator rettigheder.
- Login virker og der bliver skabt en session, som gemmer en user.

Mangler:

- Bedre håndtering af hvad der skal ske hvis man forsøger at logge ind med en bruger, der ikke findes eller et forkert password. Sådan som den er nu, kommer der ingen meddelelse til brugeren, og han bliver bare stillet om til index-siden.

Products:

- Vi har tænkt en løsning, hvor vi kunne hente ordren fra databasen ud fra de enkelte tabeller og sammensætte et view af de oplysninger. Fordelen ved denne løsning er, at administrator vil kunne lave statistik på kage salget.
- Vi havde også tænkt en anden løsning, hvor vi skrive ordrebekræftelsen ned på en tekststreng i ordre databasen. Den model forlod vi.

Implementeret:

-
- Brugeren får vist navn vha sessions og hans konto balance, når han bliver logget ind.
 - Man kan få en liste over toppe og bunde, og vælge en af hver.
 - Man kan vælge et antal af den enkelte cupcake.
 - Ved at trykke på "add to Shopping Cart" kan man få vist, at man har lagt et antal af en bestemt kage ned i kurven, og hvad det koster både pr. Kage og den samlede pris.
 - Man får også vist, hvor mange penge, der er tilbage på kontoen hvis man køber det man har i shoppingcart.
 - Når man trykker på check-ud-knappen, bliver man stillet om til shoppingCart.jsp, hvor man får vist username, balance, og hvad der er i ens shoppingcart.
 - en LineItem database, som nogenlunde svarer til entitets klassen LineItem.
 - Der er lavet en metode hvormed man kan lægge et lineitem i databasen.
 - Der er lavet en metode til at lægge ordrer ned i databasen

Mangler:

- Der var en formodning om, at der skulle laves en cupcake-class i projektet, men i stedet har vi valgt blot at generere et cupcake navn og pris ud fra de valgte toppings og bottoms navn og pris. Denne måde at designe programmet på vil gøre det lettere for vores kunde på et tidspunkt at tilføje en topping eller en bottom uden at skulle tilføje et større antal cupcakes til databasen.
- Der sættes ikke nogen rækkefølge på listen af bestillinger. Hverken alfabetisk eller bestillingsmæssigt.
- Hvis kunden vælger flere af den samme topping bottom kombination at de bare bliver lagt til i det lineitem der allerede har den pågældende kombination i denne session.

-
- Den metode hvor man kan lægge et lineitem i databasen virker nogle gange og andre gange ikke.
 - En metode hvor man kan slette lineitems.
 - At få lagt ordrer ned i databasen.
 - metode der kan samle user, lineitems og den samlede pris for bestillingen fra databasen.
 - print.out linjen være mere læsevenlig.

ShoppingCartServlet og jsp:

Implementeret:

- Det var tænkt, at vi skulle lave en ShoppingCart klasse. Det gjorde vi faktisk også, men i stedet valgte vi blot at bruge en ArrayListe, da denne klasse allerede har de metoder som er nødvendige til at udføre de operationer, der skulle benyttes undervejs i programmet.
- kunden kan se, hvad han har i sit shoppingcart, og vælge om han vil købe mere.
- Og igen hvad konto balancen vil være, hvis han køber det der er i shoppingcart.
- Når kunden trykker på buy bliver han stillet om til en bekræftelses side.
- Man kan hente det højeste ordrenummer i ordre tabellen i databasen.

Mangler:

- at man ikke kan trykke på buy knappen hvis kredit er for lav.
- At det bliver pænt stillet op.

Confirmation.jsp:

Implementeret:

-
- På siden får kunden en bekræftelse på, hvad han har købt, hvad det koster og hvor meget han har tilbage på kontoen efter købet, samt et invoice id.
 - Konto balancen er ændret i databasen til balancen minus det, det kostede at købe det der var i shoppingcart.

Mangler:

- En logout knap i bunden af siden. På nuværende tidspunkt skal kunden op i navigationsmenuen for at logge ud.

Admin:

Implementeret:

- Når man er logget ind på admin siden kan man få vist en liste over de invoicenumre der er i databasen.

Mangler:

- Invoice details på kunderne.
- Hvis man trykker på et nummer skal admin stilles om til invoicedetail.jsp.

Kundesiden:

Mangler:

- Vise alle kundens fakturaer.
- Vise kundens navn på siden.
- Hvis kunden klikker på en faktura, skal kunden kunne se detaljerne for den pågældende faktura.