

Cupcake – Sweet Retreat



Sofie Amalie Landt

E-mail: cph-sl307@cphbusiness.dk

Github: amalielandt

Amanda Juhl Hansen

E-mail: cph-ah433@cphbusiness.dk

Github: amandajuhl95

Klasse: A

Dato: 25/2 2019 – 17/3 2019

CUPCAKE
BOUTIQUE

Indledning

Denne rapport har til formål at beskrive programmet "cupcake". Programmet er en simpel webshop hvortil der er gjort brug af MySQL-database, Java servlets, jsp-sider som backend, samt html, css og JavaScript som frontend.

Webshoppen formål er at sælge cupcakes, som består af en valgfri kagebund samt topping. Kunder har mulighed for at lave en konto i webshoppen, for derved at kunne foretage en bestilling. Bestillinger kan kun laves hvis kunden er logget ind på sin konto, og har nok penge indsat. Betaling er ikke en del af dette program og varetages anden steds, men saldo håndteres stadig i databasen.

Baggrund

En aktuel kunde skal kunne registrere en ny bruger med et brugernavn, kodeord og en e-mail, eller logge ind med brugernavn og kodeord på en allerede eksisterende bruger. Kunden skal have mulighed for at vælge en cupcake bestående af en valgfri kagebund og topping, samt det ønskede antal af sammensætningen. Der skal desuden være mulighed for at se den igangværende ordre i indkøbskurven, hvoraf den samlede pris på ordren også fremgår. Det skal i indkøbskurven også være muligt at fjerne elementer fra ordren, samt afslutte ordren. Ordren skal kun kunne afsluttes hvis der er nok penge indestående på kontoen. Kunden skal derfor også have mulighed for at sætte flere penge ind på sin konto. Kunden skal også have mulighed for at se en liste af sine tidligere ordrer, samt at kunne klikke på disse for at få yderligere information om den enkelte ordre. Kunden kan eventuelt have mulighed for at se og rette egne informationer, samt at kunne logge ud af sin konto, når det ønskes.

Der skal desuden indgå en admin bruger til indehaveren af webshoppen, som har mulighed for at se alle ordrer der er bestilt, samt hvem der har bestilt dem.

Teknologi valg

Vi har udviklet webshoppen i Netbeans IDE 8.2
Data til og fra webshoppen håndteres i MySQL Workbench 8.0.13
For at tilgå databasen er der anvendt JDBC
Til frontend har vi brugt: HTML (CSS) – JSP – JavaScript

Domæne diagram – ER diagram

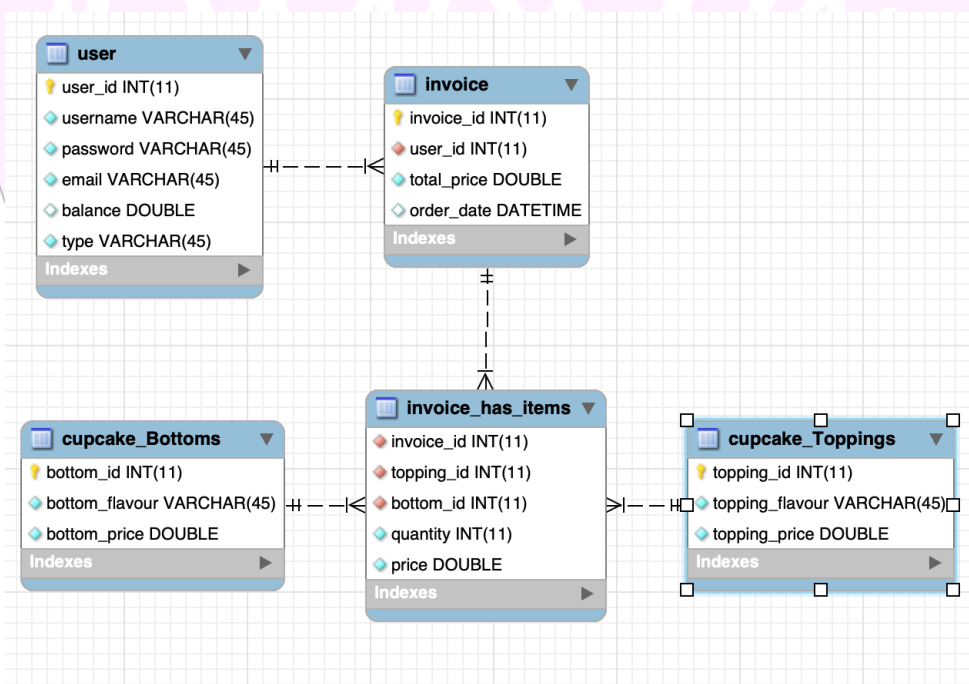
Alle tabeller i databasen er på 3. normalform. Derudover fremgår det af relationerne at alle tabellerne er 1 : mange.

Attributten, username, har vi valgt at gøre unik i tabellen user, da vi ikke ønsker flere brugere med samme brugernavn. Ligeledes har vi valgt at sætte attributterne, password og e-mail, til "not null", da det er en selvfølge at have en adgangskode til at logge ind med, samt en e-mail til videre håndtering af ordren. Attributten, type, har vi også sat til "not null", da det er relevant at vide om vi har med en administrator eller en kunde at gøre når vi bevæger os igennem webshoppen. Attributten, balance, har en default værdi på 0, da der ikke er indsat penge på en nyoprettet konto, dette må kunden selv gøre når de logger ind.

I tabellen invoice, har vi derudover gjort user_id til en fremmednøgle, der forbindes på user_id i tabellen user, da alle ordre er tilknyttet en specifik bruger. Vi har valgt at sætte order_date default til dags dato, således at vi har dato for den angivende bestilling.

Vi har valgt at bottom-price, bottom_flavour, topping_price og topping_flavour er "not null", men kan konkludere at det ikke er nødvendigt da vi aldrig skriver til disse tabeller i databasen. Vi kan ligeledes konkludere at det er unødvendigt at have "not null" på attributterne quantity og price i tabellen invoice, da vi aldrig burde komme ud for et tilfælde hvor disse værdier er 0. Dette har vi håndteret anden steds.

Derudover har vi valgt at gøre invoice_id, topping_id og bottom_id til fremmednøgler i invoice_has_items, der hver især forbindes til attributterne af samme navn i de respektive tabeller. Dette skyldes at hver række i tabellen invoice_has_items består af en unik bottom, topping og invoice.

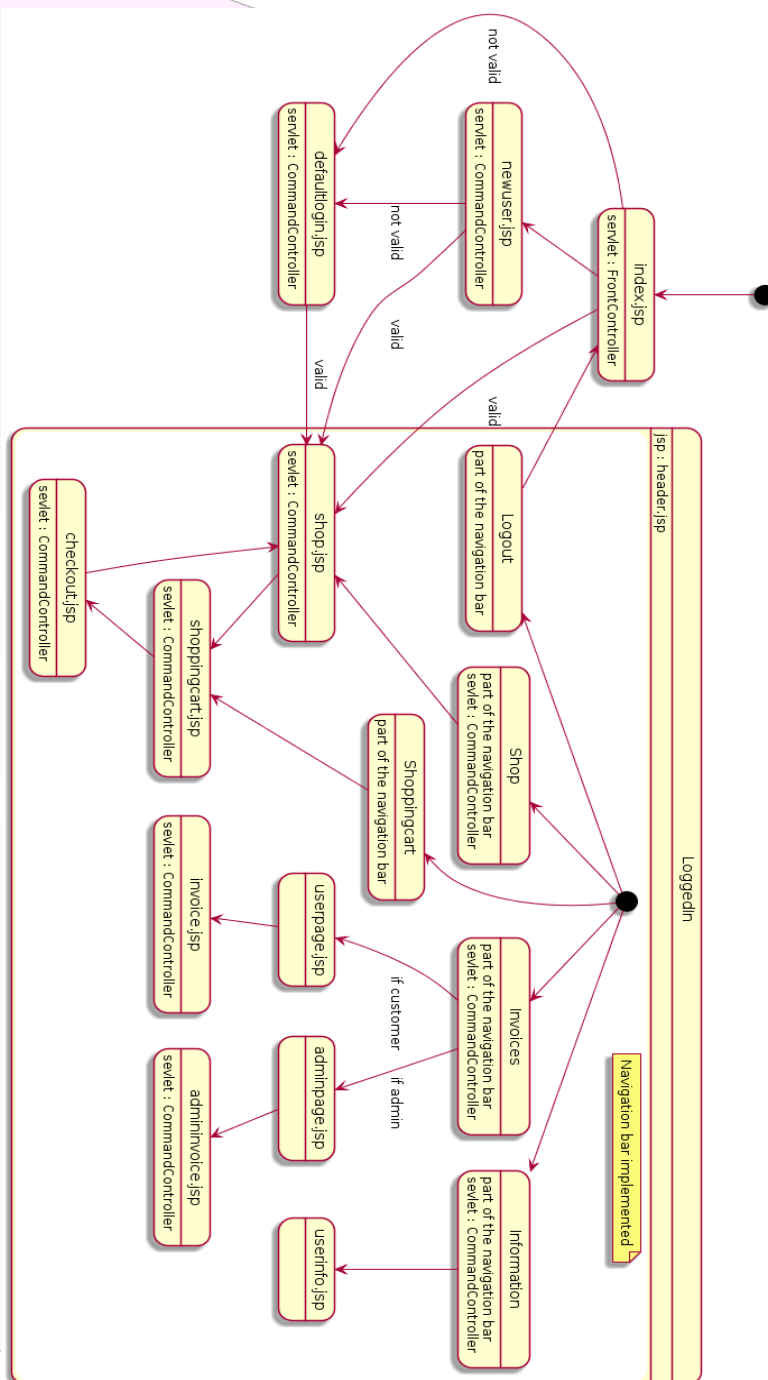


Navigations diagram

Når man som ny kunde, går ind på webshoppen bliver man ført ind i index.jsp, hvor man kan få lov til enten at logge ind på en allerede eksisterende bruger eller oprette en ny bruger. Hvis det indtastede er ugyldigt, bliver man ført videre til defaultlogin.jsp. Hvis der oprettes en bruger, føres kunden til newuser.jsp, hvorefter de har mulighed for efterfølgende at logge ind.

Efter at have indtastet et validt brugernavn og en valid adgangskode, logges man ind og kommer direkte ind på shop.jsp. Desuden kommer der en navigationsmenu frem, som er at finde på alle websider, så længe man er logget ind. Navigationsmenu henviser til henholdsvis shoppen, indkøbskurven, brugerinformation og tidligere ordre. Det fremgår yderligere af diagrammet at det henholdsvis kun er kontotypen admin der kan nå adminpage.jsp samt admininvoice.jsp og at kontotypen customer kun kan få adgang til userpage.jsp og invoices.jsp.

Når der logges ud af kontoen føres kunden tilbage til index.jsp og kan herfra på ny logge ind, hvis dette ønskes.

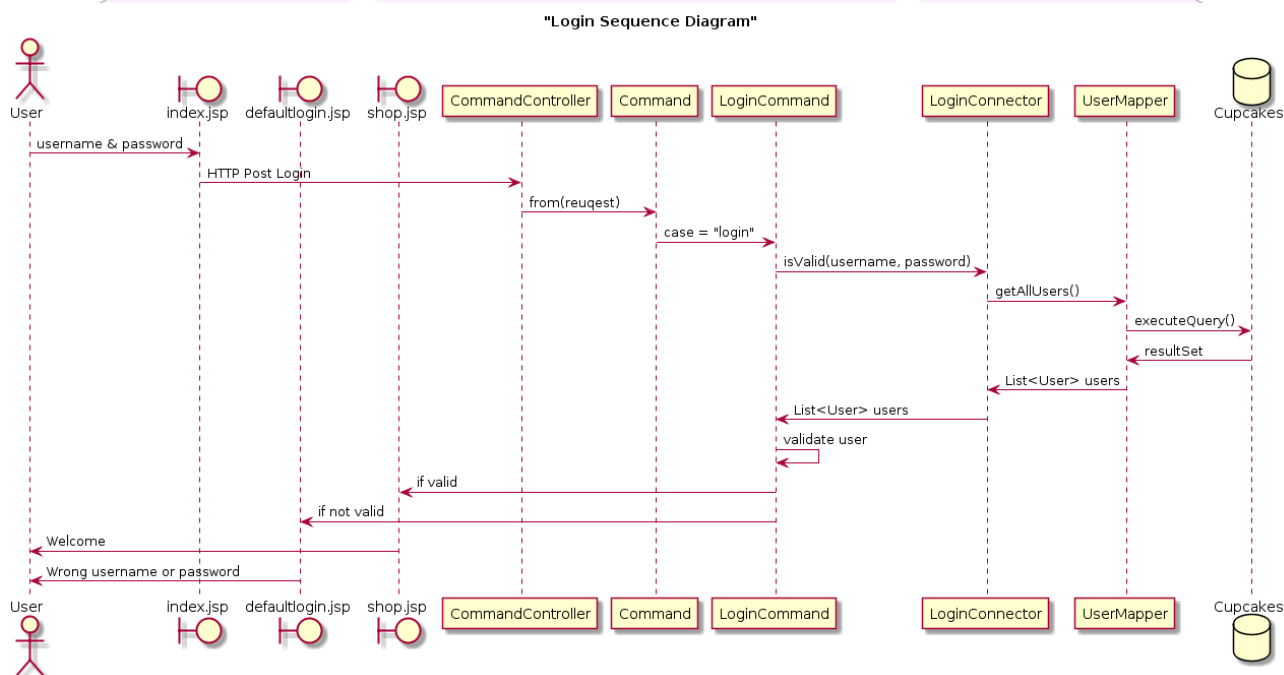


Sekvensdiagram

Login Sequence Diagram

Det nedenstående sekvensdiagram viser hvad der sker når en bruger forsøger at logge ind på webshoppen med en bruger der allerede er oprettet i databasen.

Når brugeren indtaster sit brugernavn og adgangskode på index.jsp sendes der et request til CommandController og derfra videre til Command, som identificerer dette request som "login", ved brug af path. For at LoginCommand kan validere brugernavn og adgangskode, skal der hentes en liste af de brugere som findes i databasen. Listen med alle brugerne sendes tilbage til LoginCommand, som sammenligner brugerens input med de kombinationer af et brugernavn og adgangskode der er i databasen. Hvis kombinationen af det indtastede brugernavn og adgangskode findes i databasen, bliver brugeren succesfuldt logget ind, og føres til shop.jsp, hvis kombinationen ikke findes i databasen, føres brugeren til defaultlogin.jsp og kan herfra forsøge igen, men en ny kombination.

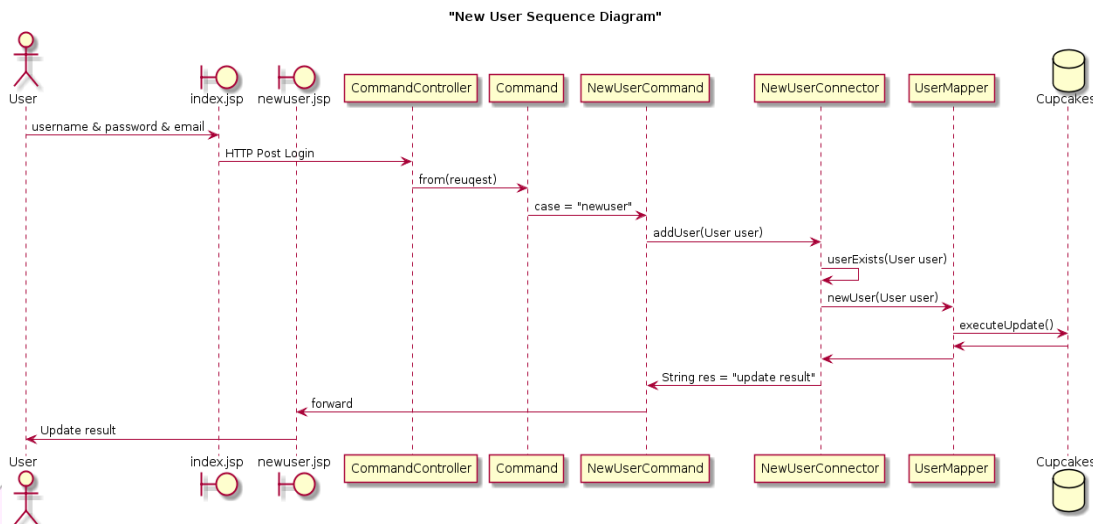


New User Sequence Diagram

Dette sekvensdiagram viser hvad der sker i programmet når en ny bruger oprettes på webshoppen. Når en nytilkommet bruger indtaster sit ønskede brugernavn, adgangskode og e-mail, sendes et http request til servlet CommandController og derefter videre til Command, der identificerer dette request som "new user".

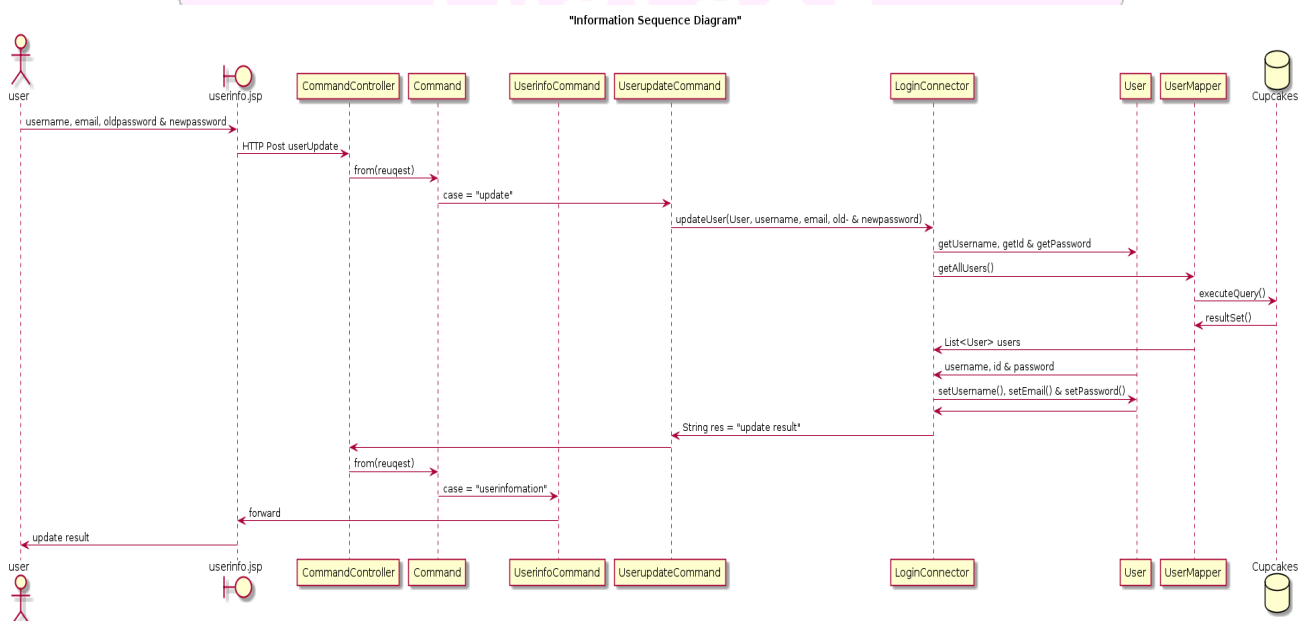
NewUserConnector anvender to metoder, den ene undersøger om både brugernavn, e-mail og adgangskode, ikke er null og ikke er tomme, mens den anden metode undersøger om en

bruger med samme brugernavn allerede eksisterer, ved at kigge i databasen. Resultatet returneres som en String, som fortæller den nye bruger, om kombinationen af brugernavn, adgangskode og e-mail allerede eksistere eller om brugeren har glemt at skrive et brugernavn eller adgangskode som ikke opfylder de specifikke krav (f.eks. for kort brugernavn). Hvis oprettelsen var en succes, vil dette også fremgå af den returnerede String.



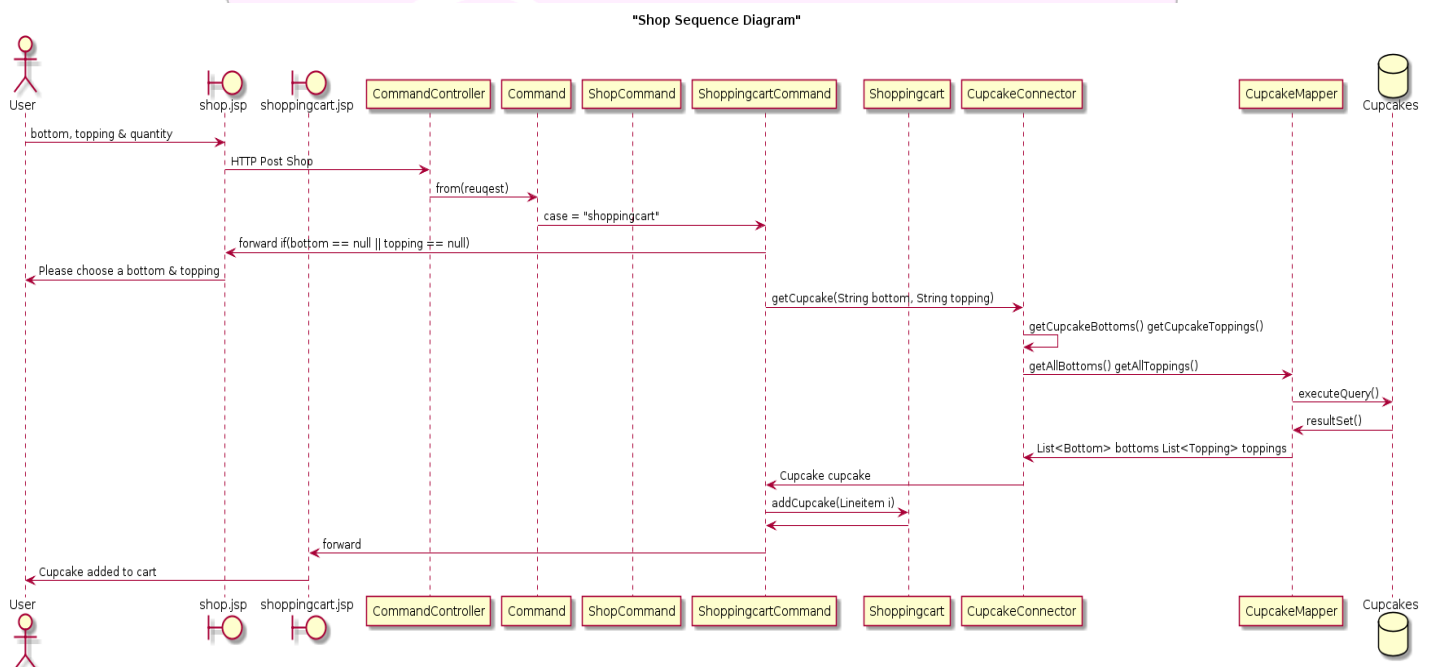
Information Sequence Diagram

Man kan opdatere sin brugers informationer, henholdsvis brugernavn, adgangskode og e-mail. Hvis dette gøres, sendes der et request til CommandController videre til Command, som indetificerer dette request som "update". For at kunne opdatere oplysninger på en bruger, skal listen af brugere hentes fra databasen, som det også ses på diagrammet. Metoden updateUser, som er i LoginConnector, undersøger hvorvidt det er muligt at opdatere de forskellige oplysninger. Hvis det ny indtastede brugernavnet allerede er i brug, eller hvis adgangskoden indtastet i feltet "Old Password" ikke stemmer overens med den gemte adgangskode i databasen, kan informationen ikke opdateres, og der sendes en string "res" tilbage til brugeren som fortæller dette. Hvis alle opdateringer er korrekte, og gemmes de nye informationer i databasen. Dette informeres brugeren også om via String "res". Brugeren føres herefter til userinfo.jsp med de opdaterede oplysninger.



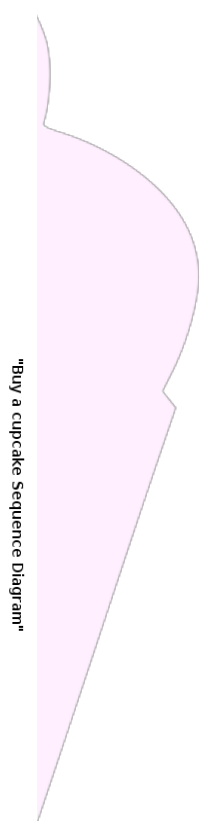
Shop Sequence Diagram

Når en bruger logger ind, bliver vedkommende henvist direkte til shop.jsp, hvor brugeren kan tilføje cupcakes bestående af en bund og topping, samt det ønskede antal af den valgte kombination, til sin indkøbskurv. Herfra vil der blive sendt en request til servlet CommandController og videre til Command, som indetificerer dette request som "shoppingcart". Hvis brugeren ingen bund eller topping har valgt sendes en string retur til brugeren, som giver brugeren besked herom, og brugeren bliver ført tilbage til shop.jsp, og har igen mulighed for at vælge en bund og topping til sin cupcake. Når brugeren har valgt en ønsket kombination af en bund og topping, samt det ønskede antal, hentes en liste af bunde og af toppings, for at sammensætte den ønskede cupcake. Når Cupcaken er fundet, tilføjes den som et Lineitem i brugerens Shoppingcart. Et Lineitem består af en Cupcake, antal og samlede pris for den kombinerede Cupcake. Brugeren føres herefter til shoppingcart.jsp, men har fortsat mulighed for at tilføje flere cupcakes på den samme ordre herfra, inden checkout.



Buy a cupcake Sequence Diagram

Når brugeren er færdig med at tilføje cupcakes til sin indkøbskurv, kan brugeren vælge at trykke "checkout", herfra vil der blive sendt en request til servlet CommandController og videre til Command, som identificerer dette request som "checkout". Hvis den totale pris af købet er lavere eller lig med balancen, vil ordren først blive sat ind i databasen, hvor den får tildelt et ordre id. Id'et sættes herefter på de enkelte Lineitem's i indkøbskurven, og disse indsættes også i databasen. Til sidst vil købeløbet blive trukket fra brugerens balance, og denne opdateres. Brugeren bliver dernæst ført tilbage til checkout.jsp, som hvis den returnerede String "checkout result", der fortæller hvorvidt ordren er gennemført eller ej. Hvis brugerens balance er lavere end den totale pris af indkøbskurven,



Særlige forhold

Informationer gemt i session

Under forløbet på webshoppens gemmes flere attributter i session. Nogle gemmes indtil der logges ud, mens andre slettes eller erstattes løbende.

User, Cart, Bottoms, Toppings, Invoices og Allinvoices gemmes igennem hele webshoppens indtil der logges ud. Vi har valgt at gemme Bottoms, Toppings, Invoices og Allinvoices i session for at opnå en hurtigere webshop, således at listerne ikke altid skal hentes fra databasen når der skiftes side. User og Cart gemmes igennem hele forløbet således at vi altid ved hvem der er logget og kunden altid kan se sin igangværende ordre. Cart slettes dog hvis ordren afsluttes. Bottoms og Toppings slettes når kunden logger ud af sin konto, dette har vi valgt så webshoppens ejer kan tilføje flere bottoms eller toppings til sin shop.

Ved oprettelse af en ny bruger gemmes en string "res" i session. Denne string kommer fra metoden addUser i NewUserConnector og giver brugeren en besked om hvorvidt brugeren er oprettet, om de indtastede informationer er ugyldige eller om brugeren allerede eksisterer. "res" slettes aldrig, eftersom den altid overskrives.

String "reply" gemmes i session, således at den kan blive fremvist på shop.jsp, såfremt der ikke er valgt en gyldig kagebund eller topping. "reply" bliver slettet fra session efter fremvisning.

Ved checkout af en igangværende ordre gemmes en string "checkout" i session, denne string kommer fra metoden checkout(ShoppingCart cart) og giver brugeren en besked om hvorvidt ordren er afsluttet eller balancen var for lav til at afslutte ordren. "checkout" slettes aldrig, eftersom den altid overskrives.

Ved checkout gemmes også en shoppingcart "invoice" i session, denne shoppingcart er den netop afsluttede ordre. Den bruges til at fremvise ordren ved checkout. "invoice" slettes aldrig, eftersom den altid overskrives.

Ved opdatering af brugerinformationer gemmes en string "update" i session, denne string kommer fra metoden updateUser og giver brugeren en besked om hvorvidt brugerens informationer er opdateret og gemt, om brugernavn allerede er i brug, eller om det indtastede adgangskode i feltet "old password" er forkert. "update" bliver slettet fra session efter fremvisning.

Når der klikkes på en tidligere ordre, gemmes denne også i session som invoice. Dette bruges udelukkende til at fremvise detaljer fra en specifik ordre. Invoice slettes fra session næste gang man går ind på siden "invoices". Dette gælder både for admin og customer.

Validering af brugerinput

Ved registrering af en ny bruger er det ikke muligt at indtaste et brugernavn kortere end fire tegn, det samme gælder password. Desuden tager funktionen addUser højde for, om en bruger allerede er oprettet under samme brugernavn.

Ved login sammenlignes det indtastede brugernavn og password, med det brugernavn og password der er gemt i databasen, dette sker i funktion isValid. Hvis der ikke findes en bruger med den indtastede kombination i databasen, nægtes brugeren adgang til shoppen og henvises til at prøve igen.

På shoppen har vi valgt, at lave dropdown-list ved henholdsvis bottom og topping, således at brugeren kun har mulighed for at vælge validerede input fra databasen til sin cupcake. Det ønskede antal af en sammensat cupcake har en minimumsbegrænsning på 1 og et maksimum på 100.

Derudover er det ikke muligt at sætte et beløb lavere end 1 kr. og højere end 1000 kr. ind på sin konto.

Ved opdatering af sine informationer er de samme specifikationer gældende som ved oprettelse af en bruger. Brugernavn og adgangskode skal være minimum 4 tegn. I funktionen updateUser valideres det indtastede brugernavn, hvis det allerede er i brug, udføres opdateringen ikke. Det samme er gældende hvis feltet "old password" ikke stemmer overens med det allerede gemte password og opdateringen udføres heller ikke. Der udføres ingen validering af e-mail, heller ikke ved registrering af en ny bruger.

Sikkerhed af login

Ved login gemmes brugeren efter validering i session. Ved log ud slettes brugeren fra session. Vi har fravalgt at udføre yderligere sikkerhed omkring login, da vi ikke har kendskab til mere avanceret sikkerhed.

Brugertyper i databasen

Vi har to brugertyper: 1 admin og en række forskellige customers. Alle brugere der bliver oprettet, bliver automatisk oprettet som customer-type, mens der i forvejen er implementeret 1 admin bruger i databasen. Vi har fravalgt at gøre det muligt at oprette flere administratorer, da det kun er nødvendigt med en admin. Disse informationer bruges kun på brugerens egen side, for at bestemme hvor mange informationer og hvilke der skal vises.

Status på implementation

Vi har ikke fået "wrappet" database insert i en transaktion for at forsikre at enten hele ordren eller intet bliver gemt i databasen. Dette skyldes manglende kendskab til hvordan dette gøres.