CUPCAKE

Oktober 2018

Navn	**Cph-Email:**	**Github-Navn:**	**Klasse:**
Lea Gemzøe Nielsen	cph-ln240@cphbusiness.dk	Lea-Nielsen	2. Sem efterår 2018
Marko Kvasina	cph-mk524@cphbusiness.dk	Mkvasi	2. Sem efterår 2018
Morten Feldt	cph-mf227@cphbusiness.dk	spottieMF	2. Sem efterår 2018
Nikolai Rojahn	cph-nr87@cphbusiness.dk	NikolaiRojahn	2. Sem efterår 2018

Indholdsfortegnelse

Indledning

Systemet er udviklet til brug i en online cupcake shop. Der kan ved oprettelse af en bruger bestilles cupcakes, som efterfølgende kan afhentes på firmaets adresse.

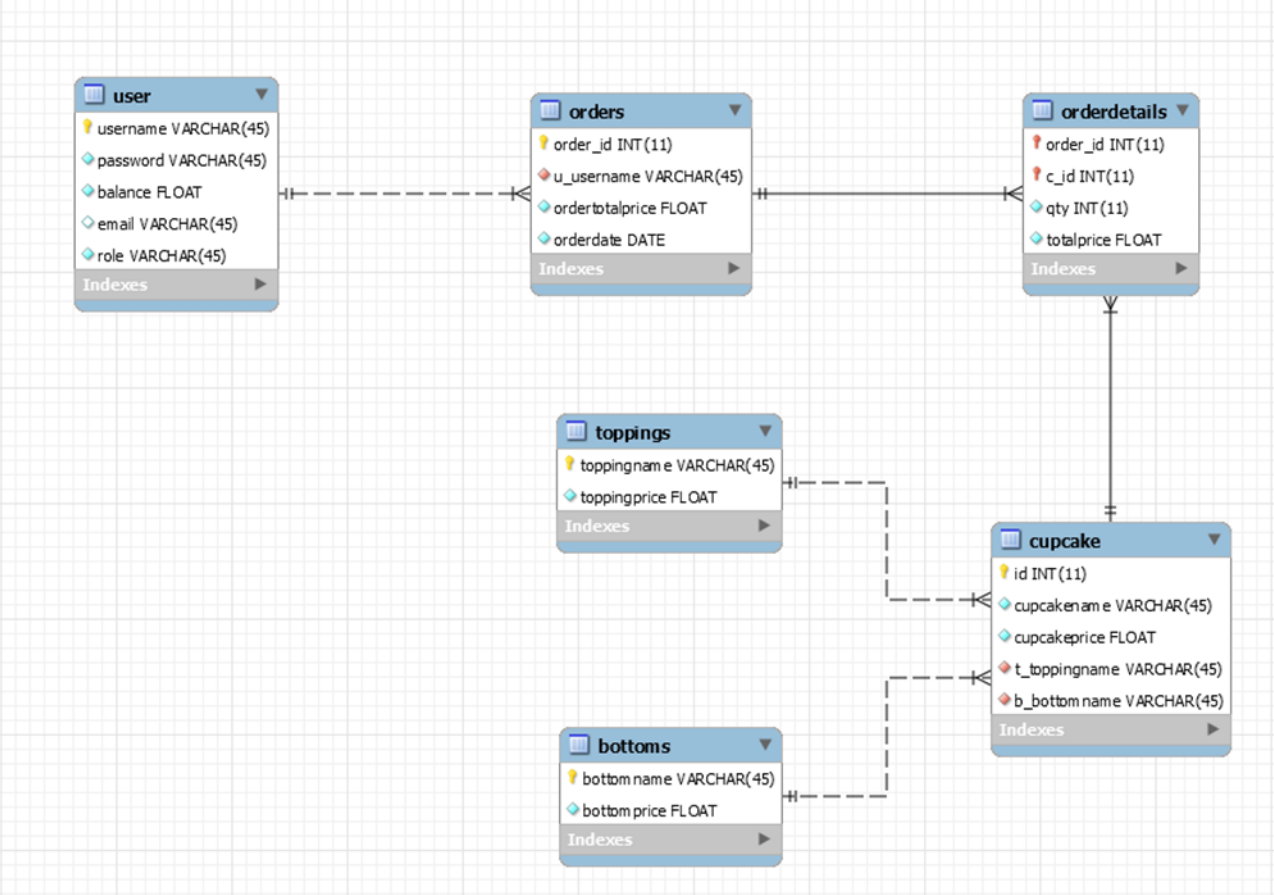
Baggrund

- Brugeren skal kunne oprette en bruger med tilknyttet saldo til køb af cupcakes.
- Brugeren skal logge ind og have nok kredit på sin saldo for at kunne gennemføre en ordre.
- En cupcake består af en bund, en top og et antal.
- Brugeren skal kunne se sine fakturerer fra tidligere bestillinger.
- Admin login, hvor alle brugernes fakturerer er listet, men også kan ses i detaljer.

Teknologi valg

- Netbeans IDE 8.2
- MySQL Connector JDBC 5.1.47
- MySQL Workbench 6.3 CE
- Visual Paradigm 15.1
- WEBserver (f.eks. Apache Tomcat)
- GitBash
- Internet browser (f.eks. Google Chrome)

Domæne model og ER diagram



ER diagrammet illustrerer måden hvorpå systemet er sammensat af entiteter og relationer. Når en bruger afgiver en ordre, vil valgte toppe og bunde blive omdannet til cupcake objekter, som indsættes i databasen, til brug i tabellen orderdetails, som repræsentere brugerens faktura.

Cupcaketabellen er indsat således, at tidligere fakturas angivne priser ikke påvirkes ved en evt. prisændring senere.

Beskrivelse af primærnøgler:

Attributterne ***bottomname*** og ***toppingname*** er primærnøgler af datatypen VARCHAR. Ift. at virksomheden selv skal kunne angive deres salgsvarer, er disse primærnøgler ikke autogenereret. Dette forhindrer oprettelse af identiske navne.

Attributten ***username*** er af hensyn til brugeren også af datatypen VARCHAR og autogenreres ikke, da det kan være til gene for brugeren ikke selv at kunne vælge sit brugernavn. Dette forhindrer oprettelse af identiske brugernavne.

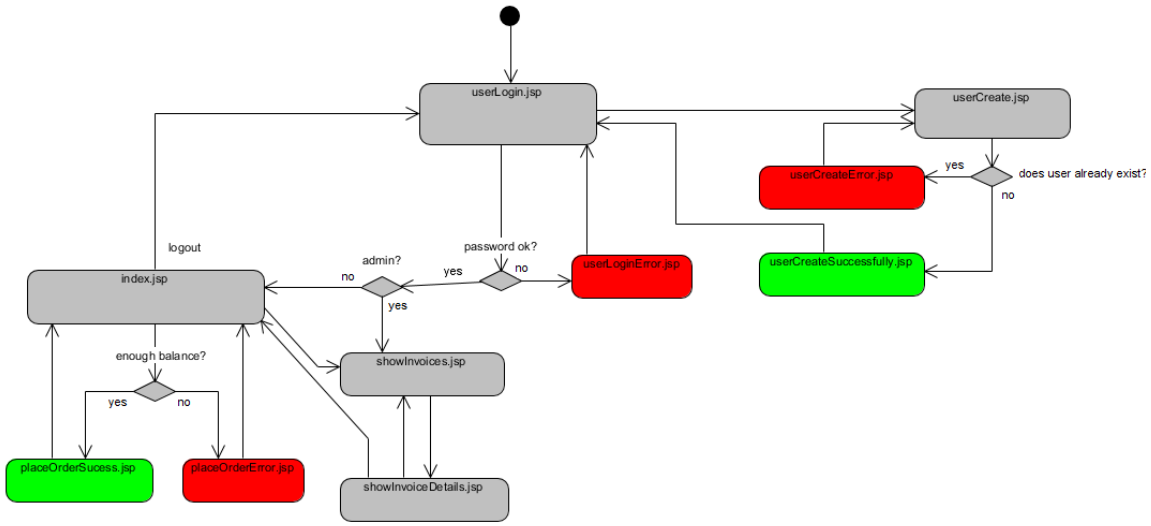
Beskrivelse af fremmednøgler:

Attributterne ***order_id*** og ***c_id*** er begge fremmednøgler til hhv. ***orders*** og ***cupcake***. De to fremmednøgler anvendes som en sammensat primærnøgle for ***orderdetails*** tabellen.

Attributten ***t_toppingname*** i tabellen ***cupcake*** er fremmednøgle til tabellen ***toppings***, dette er gjort således, at der ikke er risiko for oprettelse af en cupcake med en top, som ikke eksisterer ***toppings*** tabellen.

Attributten ***b_bottomname*** i tabellen ***cupcake*** er fremmednøgle til tabellen ***bottoms***, dette er gjort således, at der ikke er risiko for oprettelse af en cupcake med en top, som ikke eksisterer ***bottoms*** tabellen.

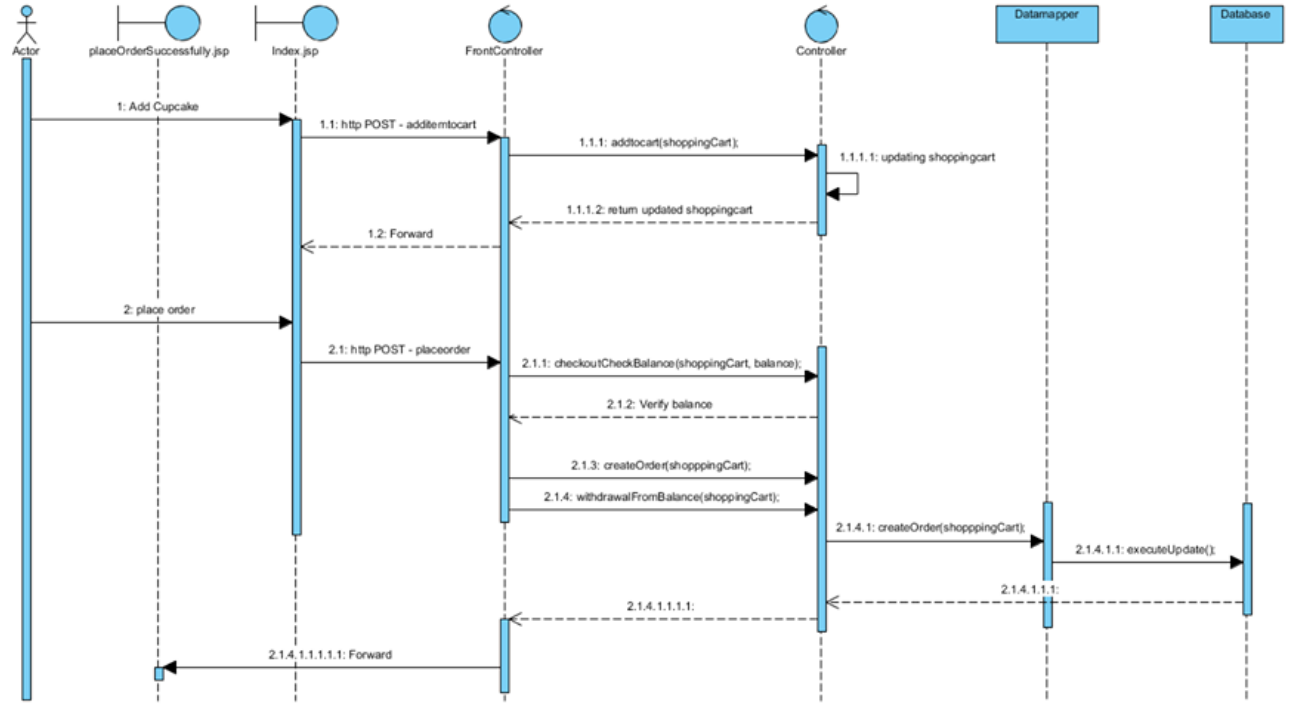
Navigationsdiagram



Navigationsdiagrammet illustrerer hvordan brugeren/admin kan bevæge sig rundt på hjemmesiden. Frontcontrolleren er den servlet, som bringer brugeren/admin fra en side til den næste.

Som det fremgår af diagrammet, har både admin og bruger adgang til showInvoices.jsp. Siden generes efter om det er hhv. en admin eller en bruger der et logget ind. Brugeren får kun adgang til sine egne fakturaer, mens admin har adgang til at se alle fakturaer.

Sekvensdiagram



Add Cupcake:

Brugeren vælger en cupcake bestående af top og bund og trykker på ”Add to cart”. Index.jsp sender en http POST til FrontController med parameter ”addItemToCart”, hvorefter FrontController giver besked til Controller om at køre metoden ”addToCart” med parameteret ”ShoppingCart”. Herefter opdateres ShoppingCart, som ligger i sessionen til at inkludere den nye cupcake og returnerer til FrontController, som laver forward videre til index.jsp.

Place Order:

Brugeren trykker på ”Place order”. Index.jsp sender en http POST til FrontController med parameter ”placeOrder”, hvorefter FrontController giver besked til Controller om at tjekke om balancen er stor nok til at ordren kan gennemføres. Controller sender svaret retur til FrontController, som tjekker svaret. Hvis balancen er stor nok til at gennemføre ordren, giver FrontController besked til Controller om at opdatere balancen ved at trække det samlede ordrebeløb fra, og sender en opdateret balance tilbage til FrontController.
Hvis balancen ikke er stor nok, vises en side til brugeren om at der ikke er balance nok til at gennemføre ordren. Herefter giver FrontController besked til Controller om at udføre metoden ”createOrder” med ShoppingCart fra sessionen som parameter, hvorefter Controlleren giver besked til DataMapper om at udføre metoden createOrder med ShoppingCart fra sessionen som parameter. DataMapper udfører en forespørgsel til databasen, indsætter ordren i tabellerne og returnerer et resultSet til DataMapper, som sendes videre til Controller. Controller giver besked til FrontController om at placeringen af ordren er OK, hvorefter FrontController laver forward videre til index.jsp.

Særlige forhold

Brugere i database og JDBC:

På databasen er oprettet en bruger som hedder ”connect” med passwordet ”connect”, denne bruger har alle rettigheder, den er oprettet for ikke at bruge ”root” brugeren.

Ovenstående bruger er også brugt i JDBC, som laver connection til databasen gennem applikationen.

Hvilke informationer gemmes i session:

- Username:
    - Usernamet på den bruger som er logget ind gemmes i sessionen, da username skal bruges til flere metoder i systemet.

- Balance
    - Balance gemmes på den bruger som er logget ind, således kan der nemt vises brugerens balance, selvom der f.eks. navigeres til sider hvor denne information ikke er vist.

- Status om det er admin eller en bruger
    - Der er forskel på hvilket indhold der skal vises alt efter om man er ”admin” eller almindelig ”bruger”, derfor gemmer vi en indikation af om den bruger som er logget ind er ”admin” eller almindelig ”bruger” i sessionen.

- Listen på Toppings
    - Når en bruger logger ind, gemmes alle Toppings fra databasen, således kan de vises på siden uden yderligere forespørgsler til databasen, hvis der navigeres til sider uden disse lister, f.eks. invoice mellem bestilling af cupcakes.

- Listen på Bottoms
    - Når en bruger logger ind, gemmes alle Bottoms fra databasen, således kan de vises på siden uden yderligere forespørgsler til databasen, hvis der navigeres til sider uden disse lister, f.eks. invoice mellem bestilling af cupcakes.

- ShoppingCart
    - Når en bruger bestiller en cupcake, genereres en ”ShoppingCart” i sessionen, og denne opdateres hver gang man bestiller nye cupcakes. Dette er for at undgå at man laver forespørgsel til databasen ved hver bestilling af en cupcake.

Status på implementation

- Systemet håndterer ikke catch ved en exception.
- Det er kun muligt at logge ud af systemet fra index-siden.
- Place order vises selvom der ikke er bestilt en cupcake.
- Der vises jsp-sider med tekstbeskeder i stedet for diskret fejl besked.
- Siden der viser liste over invoices er ikke stylet.
- Ustruktureret login-case i FrontController f.eks. hentes for mange User Objekter og listerne på Toppings og Bottoms hentes før login er valideret.
- Username på invoices tages fra sessionen og ikke fra databasen.