cupcake

Oktober 2018

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

Indholdsfortegnelse

[Indledning 2](#_Toc526507002)

[Baggrund 2](#_Toc526507003)

[Teknologi valg 2](#_Toc526507004)

[Domæne model og ER diagram 3](#_Toc526507005)

[Navigationsdiagram 5](#_Toc526507006)

[Sekvensdiagrammer 6](#_Toc526507007)

[Særlige forhold 7](#_Toc526507008)

[Status på implementation 8](#_Toc526507009)

# 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.

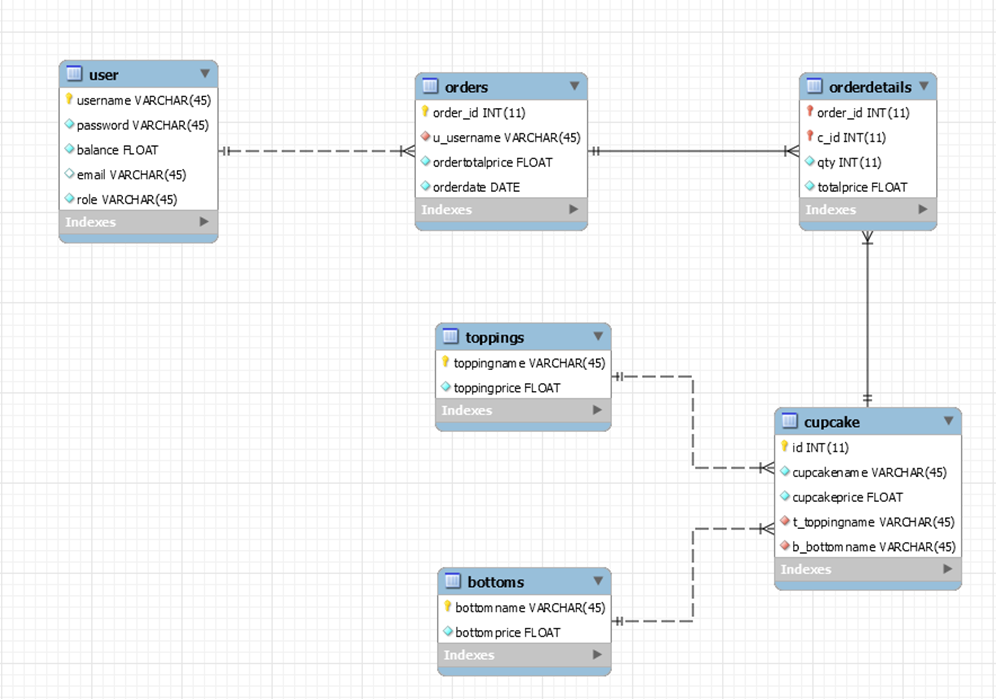
## Baggrund

* 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

# 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.

Cupcake tabellen 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 vælge deres salgsvarer, og brugeren skal kunne forstå indholdet af valgt bund eller top, er disse primærnøgler ikke autogenereret. Dette forhindre 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 forhindre oprettelse af identiske brugernavne.

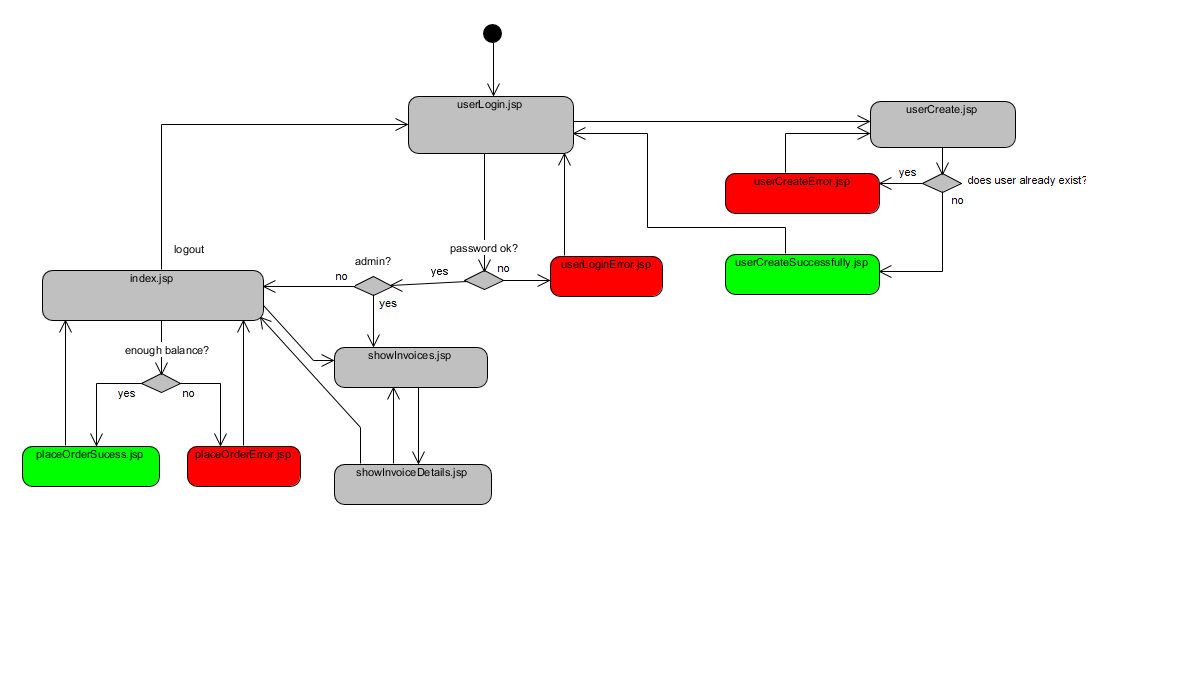
**Beskrivelse af fremmednøgler:**

Attributterne ***order\_id*** og ***c\_id*** er begge fremmednøgler til hhv. ***orders*** og ***cupcake*** tabellen. 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.

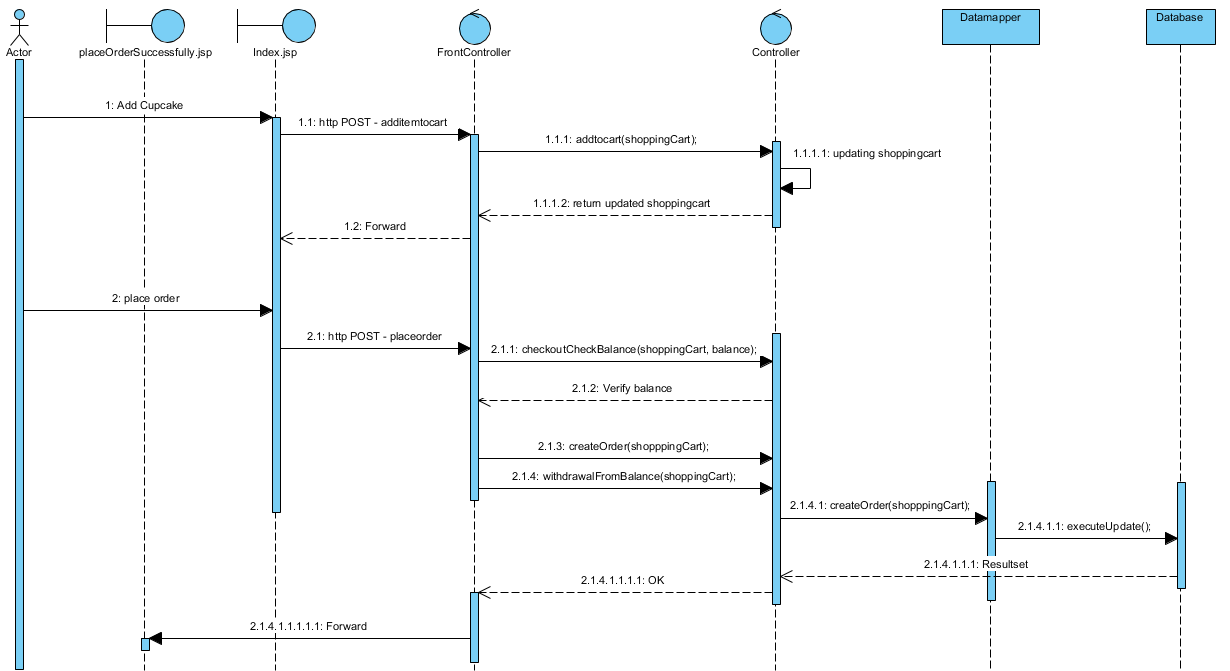
Attributten ***b\_bottomname*** i tabellen ***cupcake*** er fremmednøgle til tabellen ***bottoms***, dette er gjort således, at der ikke er risiko for oprettelse af en cupcake med en top, som ikke eksisterer ***bottoms*** tabellen.

# Navigationsdiagram



Navigationsdiagrammet illustrerer brugerens og admin kan bevæger sig rundt på hjemmesiden. Frontcontrolleren er den servlet, som bringer brugeren fra en side til den næste. Som det fremgår af diagrammet, har både admin og bruger adgang til showInvoices.jsp. Siden bliver dog generet ift. om det er hhv. en admin eller en bruger der et logget ind. Brugeren får kun adgang til sine egne fakturaer, mens adminen har adgang til at se alle brugeres fakturaer.

# Sekvensdiagrammer



**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 Controlleren om at køre metoden ”addToCart” med argumentet ”ShoppingCart”. Herefter opdateres ShoppingCart, som ligger i sessionen til at inkludere den nye cupcake og returnere tilbage til FrontController, som laver en 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 Controlleren om at tjekke om balancen er stor nok til at orderen kan gennemføres. Controlleren sender svaret retur til FrontController, som tjekker svaret. Hvis svaret er godkendt og balancen er god nok til at gennemføre ordren, giver FrontControlleren besked til Controlleren om at opdatere balancen ved at trække det samlede ordrebeløb fra, og sender en opdateret balance tilbage til FrontController.

Herefter giver FrontController besked til Controller om at udføre metoden ”createOrder” med ShoppingCart fra sessionen som argument, hvorefter Controlleren giver besked til DataMapper om at udføre metoden createOrder med ShoppingCart fra sessionen som argument. DataMapperen udføre en forespørgsel til databasen, indsætter ordren i tabellerne og returnerer et resultSet til DataMapperen, som sender videre til Controlleren. Controlleren giver besked til FrontController om at placeringen af ordren er OK, hvorefter FrontController laver en Forward retur til index.jsp.

# Særlige forhold

**Brugere i database og JDBC:**

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

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

**Hvilke informationer gemmes i sessioner:**

* Username:
  + Den bruger som er logget ind gemmes username på, da username skal bruges til flere metoder i systemet.
* Balance
  + Balance gemmes på den bruger som er logget ind, således kan vi nemt vise brugerens balnce 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.
* Listen på Toppings
  + Når en bruger logger ind gemmer vi alle Toppings fra databasen, således vi kan vise dem 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 gemmer vi alle Bottoms fra databasen, således vi kan vise dem 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 bestillger en cupcake genereres en ”ShoppingCart” i sessionen, og denne opdateres hvergang man bestiler nye cupcakes. Dette er for at undlade at man laver forespørgsel til databasen ved hver bestilling af en cupcake.

# Status på implementation

* Systemet håndtere 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